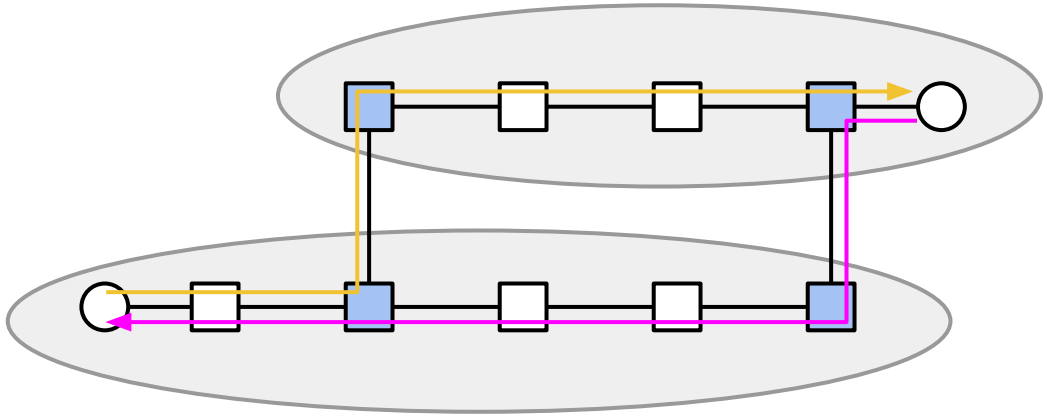


Lecture 10 (Routing 6)

BGP Implementation and IP Header

CS 168, Spring 2025 @ UC Berkeley

Slides credit: Sylvia Ratnasamy, Rob Shakir, Peyrin Kao

External BGP and Internal BGP

Lecture 10, CS 168, Spring 2025

BGP Implementation

- **External BGP and Internal BGP**
- Multiple Links Between ASes
- Message Types and Route Attributes
- Issues with BGP

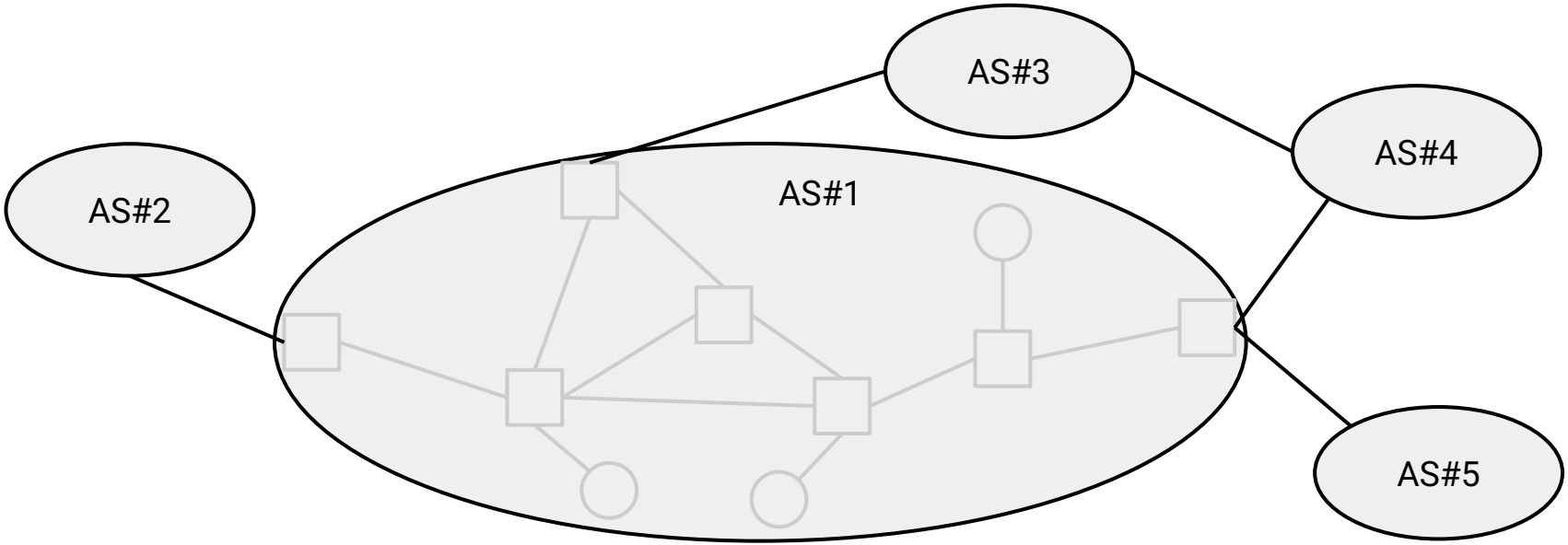
IP Header

- IPv4 Header Fields
- IPv6 Changes
- Security

Combining Intra-Domain and Inter-Domain Routing

So far, we've shown BGP in terms of ASes talking to each other.

An AS advertises routes to its neighbor ASes.

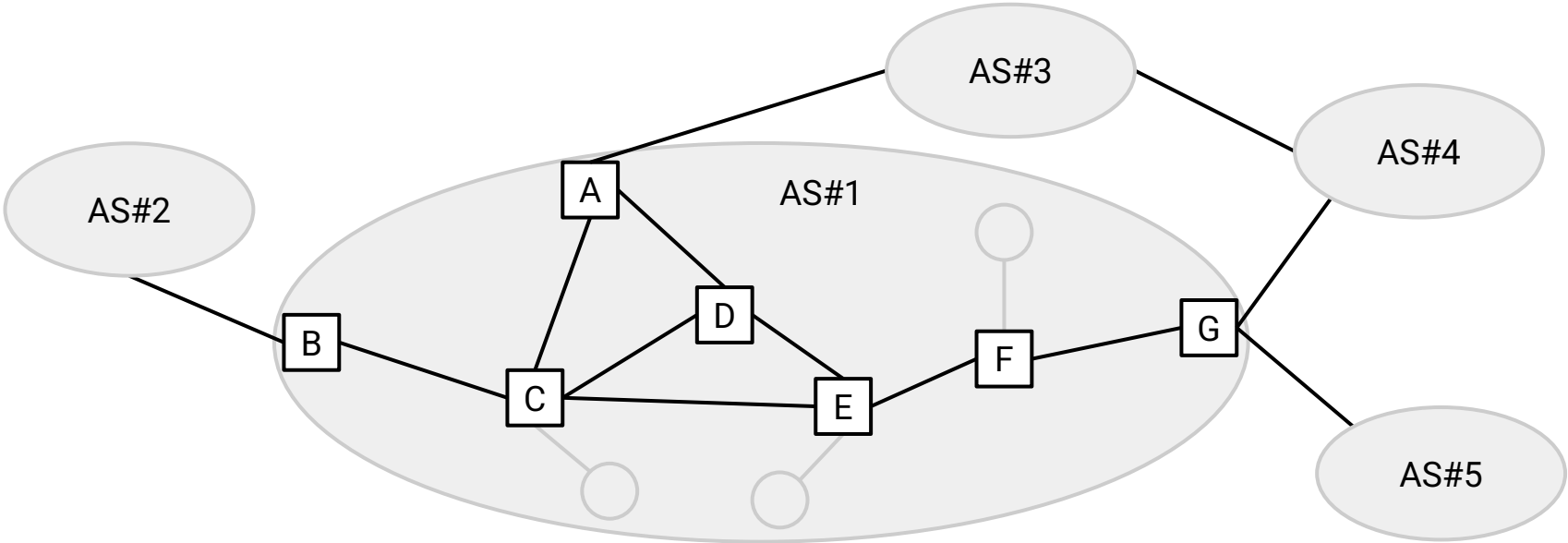


Combining Intra-Domain and Inter-Domain Routing

In reality, an AS is made up of a bunch of routers.

We have to ensure the routers "act as one."

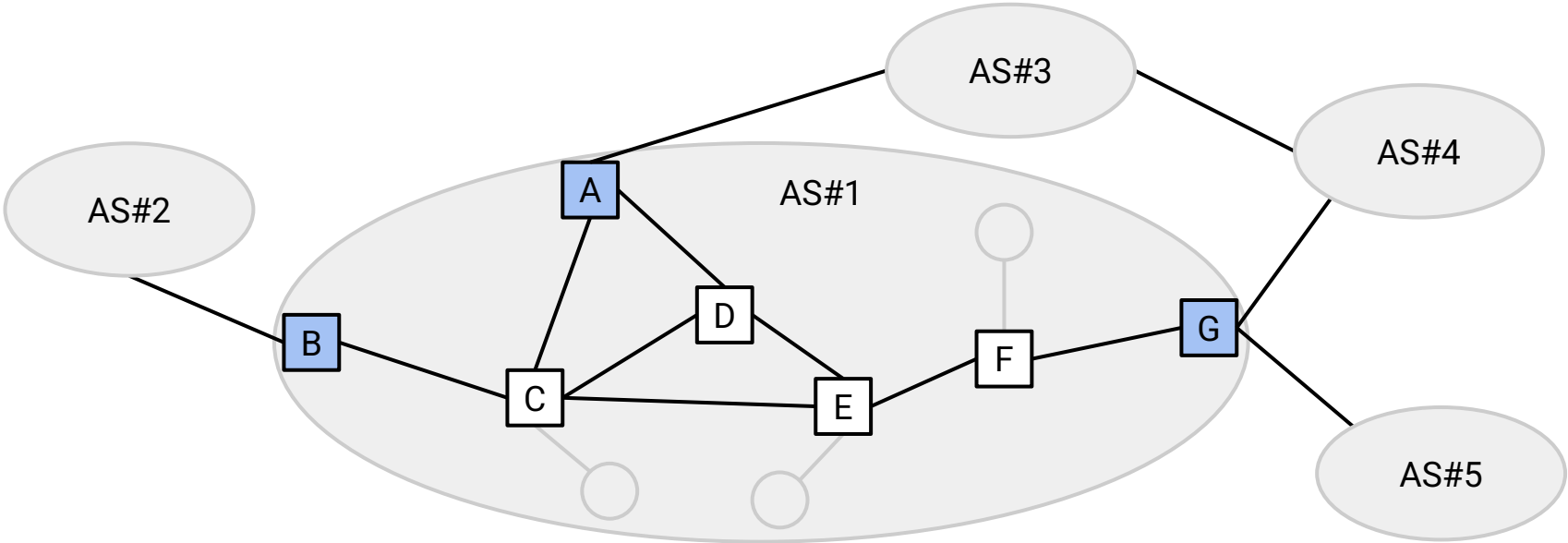
Today, we'll also see how BGP interacts with intra-domain routing.



Border Routers and Interior Routers

Border routers: At least one link to a router in a different AS. (A, B, G)

Interior routers: Only linked to other routers in the same AS. (C, D, E, F)

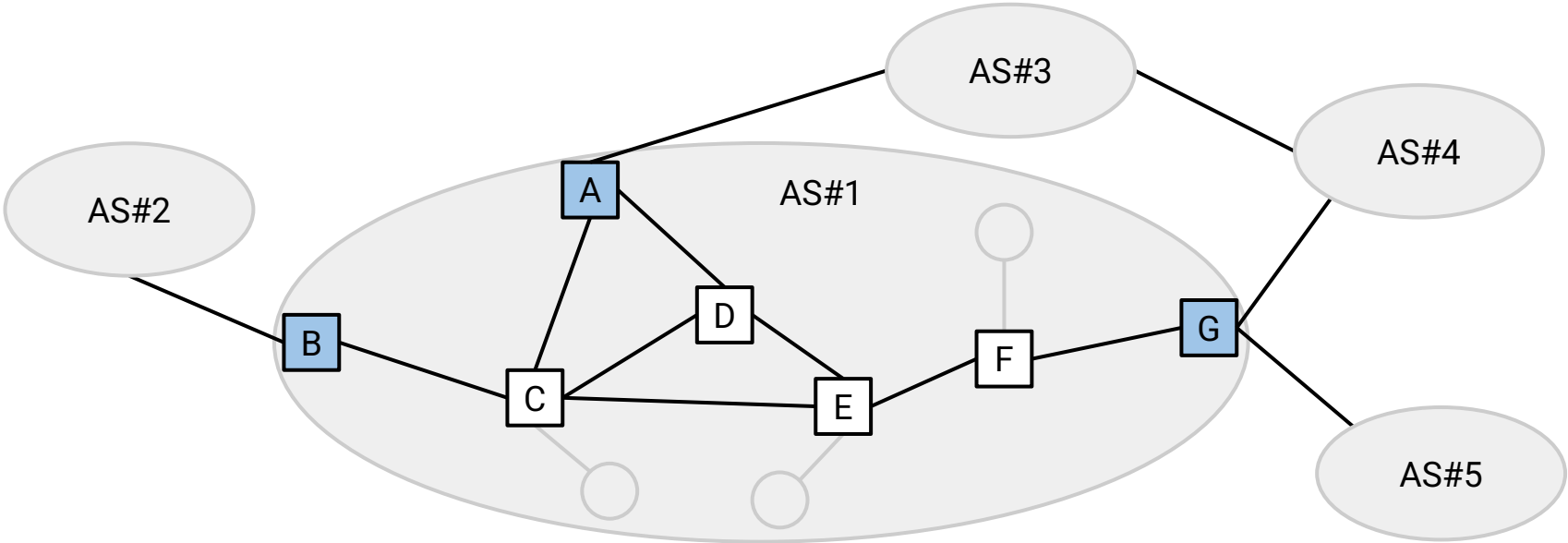


Border Routers and Interior Routers

BGP speakers: The routers that advertise BGP paths to other ASes.

Who speaks BGP? All the border routers inside each AS.

Speakers must understand BGP syntax and semantics.

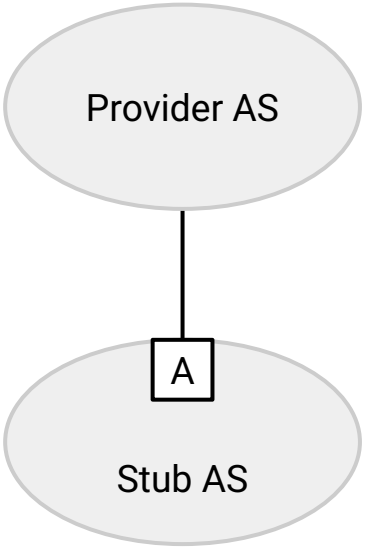


Border Routers and Interior Routers

BGP speakers: The routers that advertise BGP paths to other ASes.

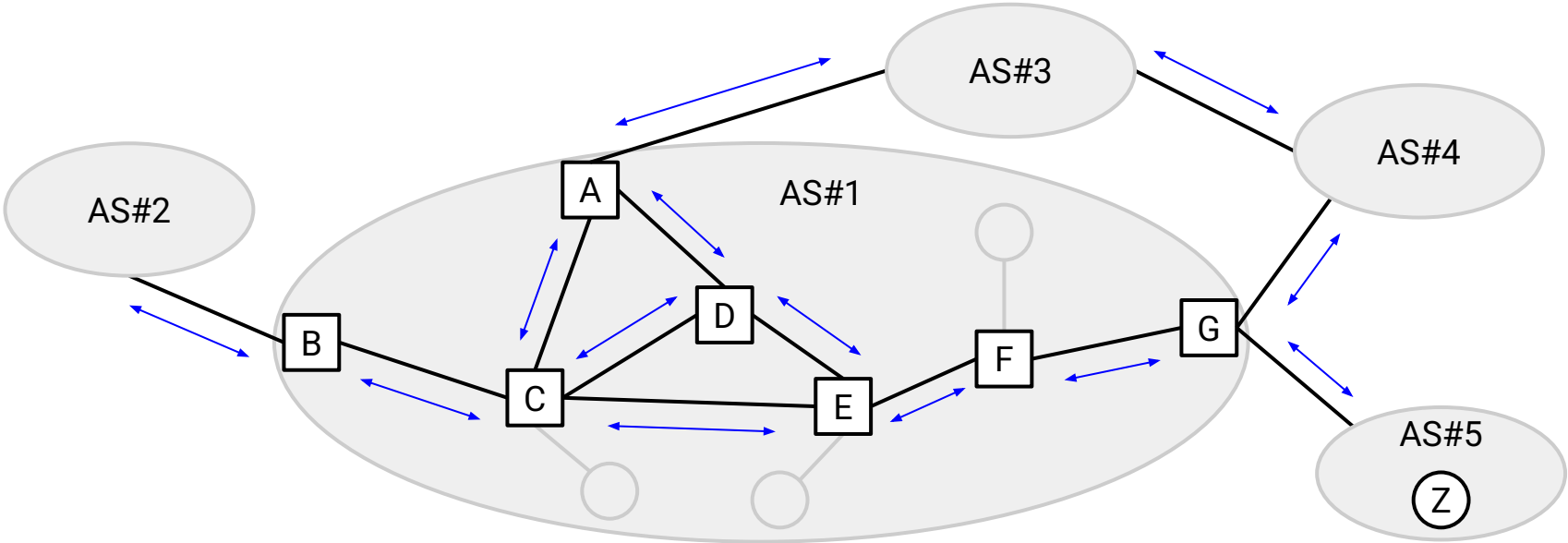
Recall: Stub ASes don't need to speak BGP.

The border router can default-route everything to the provider AS.



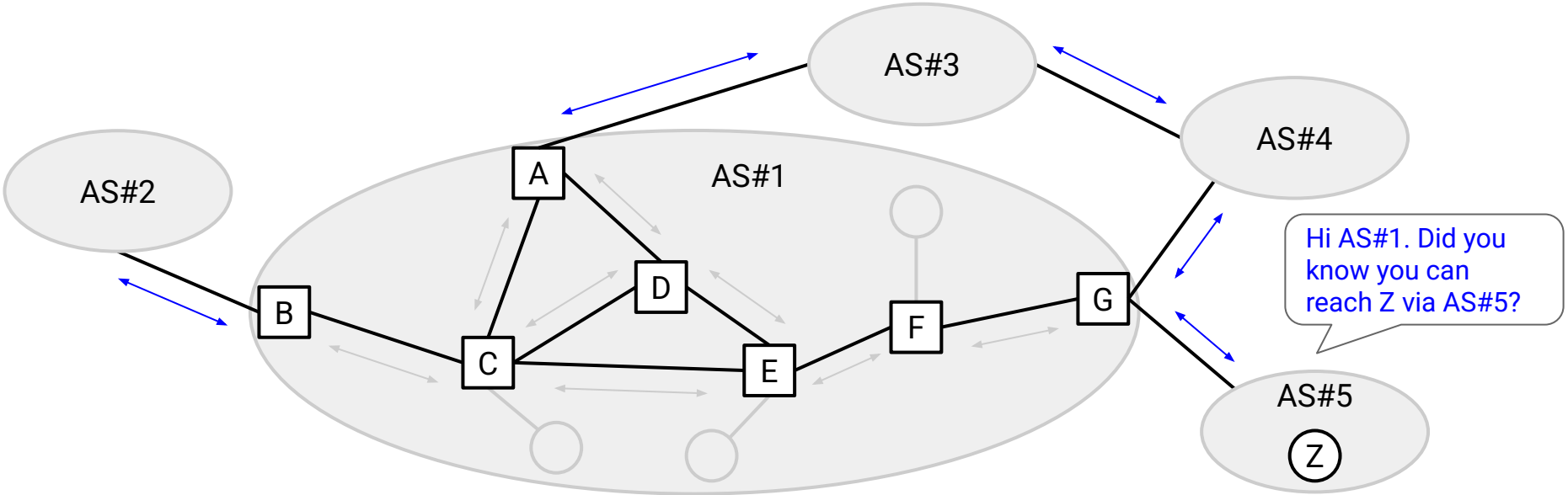
External BGP, Internal BGP, and IGP

BGP session: Two routers exchanging BGP information between each other.



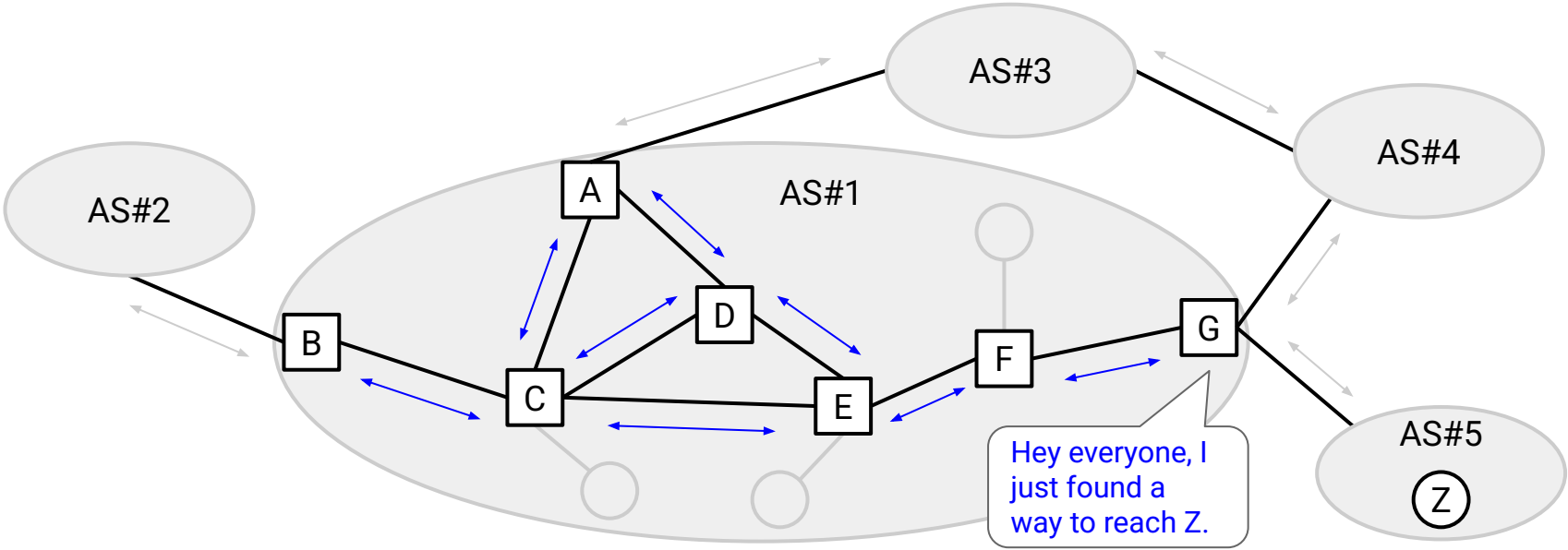
External BGP (eBGP) session: Between two routers in *different* ASes.

Border routers use eBGP to exchange inter-domain routes.



Internal BGP (iBGP) session: Between two routers in the *same* AS.

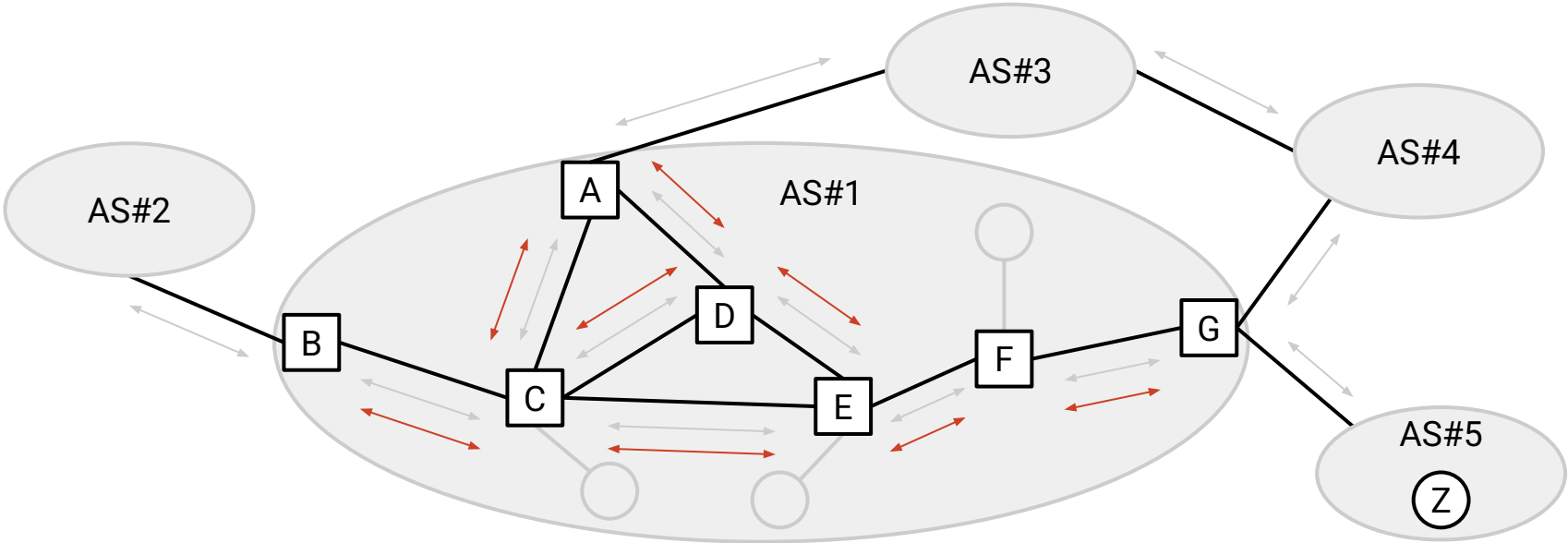
Border routers use iBGP to distribute the routes it discovers to everyone else in the AS.



External BGP, Internal BGP, and IGP

Interior gateway protocol: Between two routers in the *same* AS.

Used to find paths inside the AS (e.g. distance-vector, link-state).



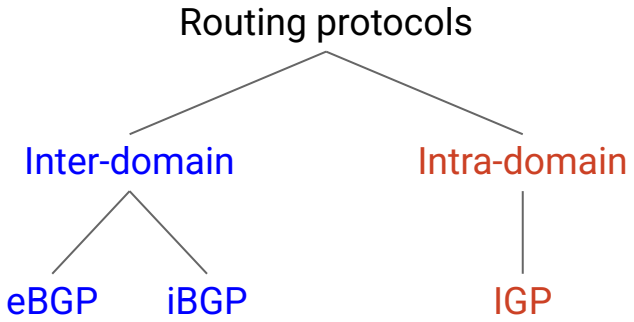
External BGP, Internal BGP, and IGP

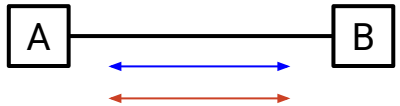
Three types of communications between routers:

- **eBGP**: Routers in different ASes exchanging routes.
- **iBGP**: Router distributing externally-learned routes to interior routers.
- **IGP**: Intra-domain routing protocols for internal reachability.

Be careful not to confuse iBGP and IGP.

- Both are between interior routers.
- iBGP is for external routes, and IGP is for internal routes.



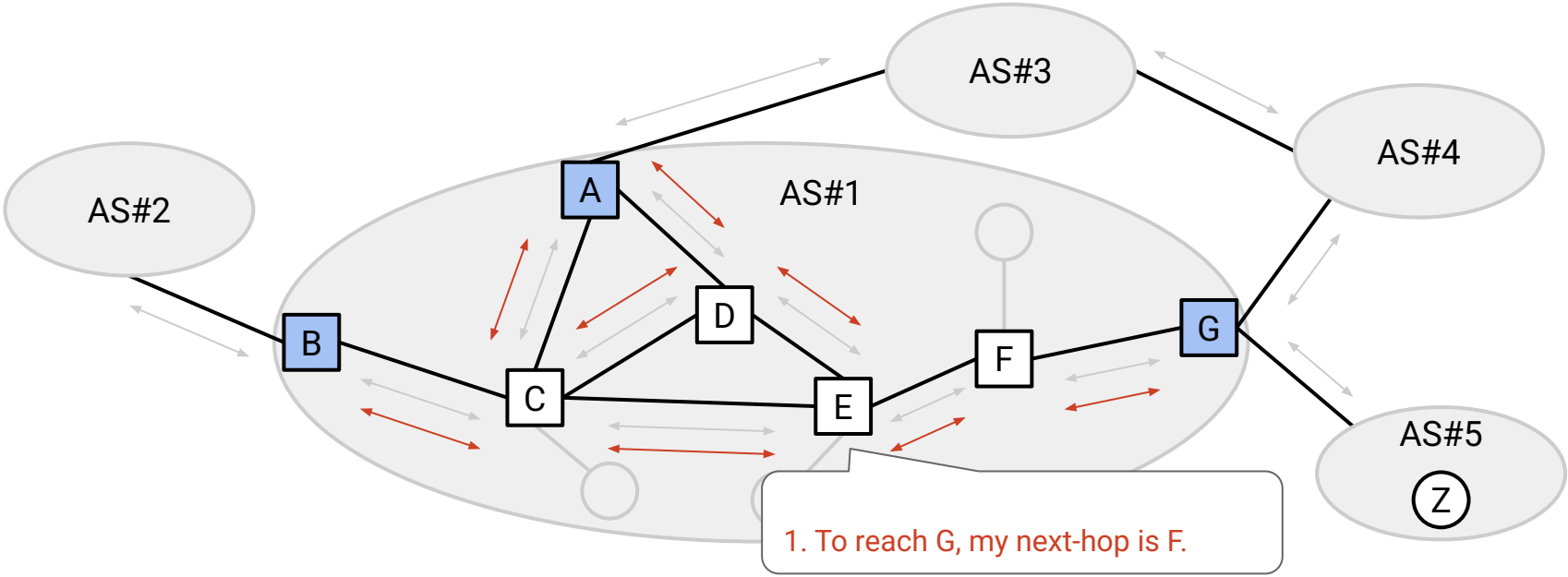


iBGP: "I can reach AS#5 via [AS#7, AS#9]."

IGP: "I can reach C with cost 3."

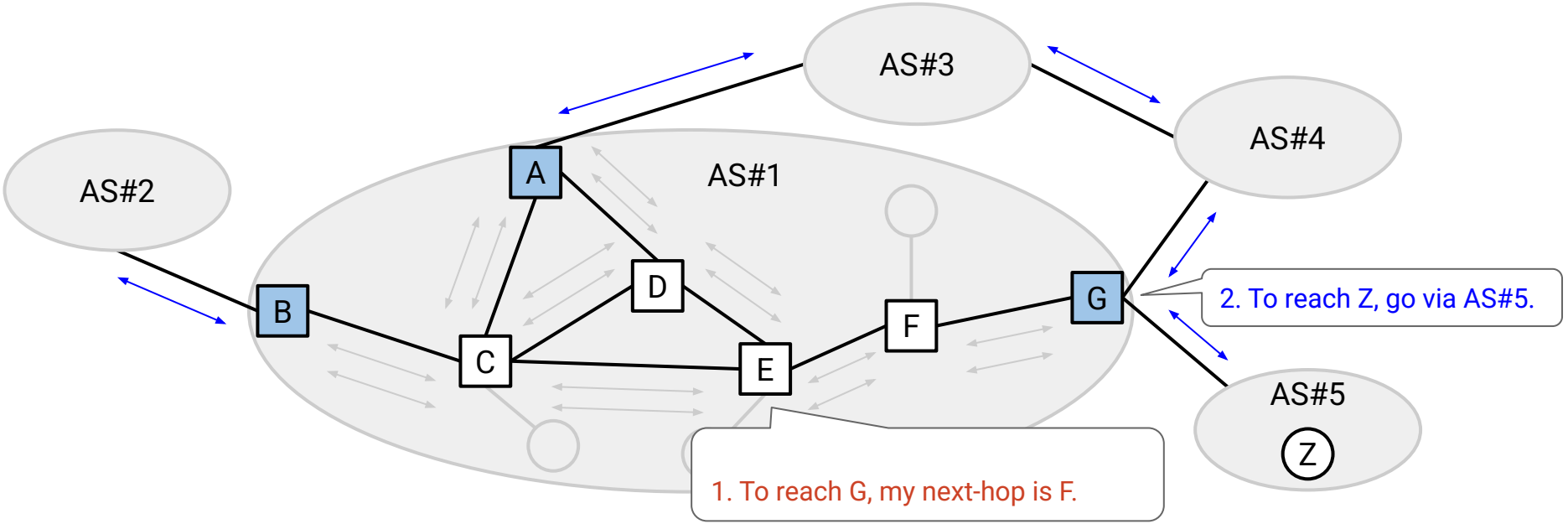
Combining Intra-Domain and Inter-Domain Routing

1. Use IGP to learn *internal* routes to others in the AS.



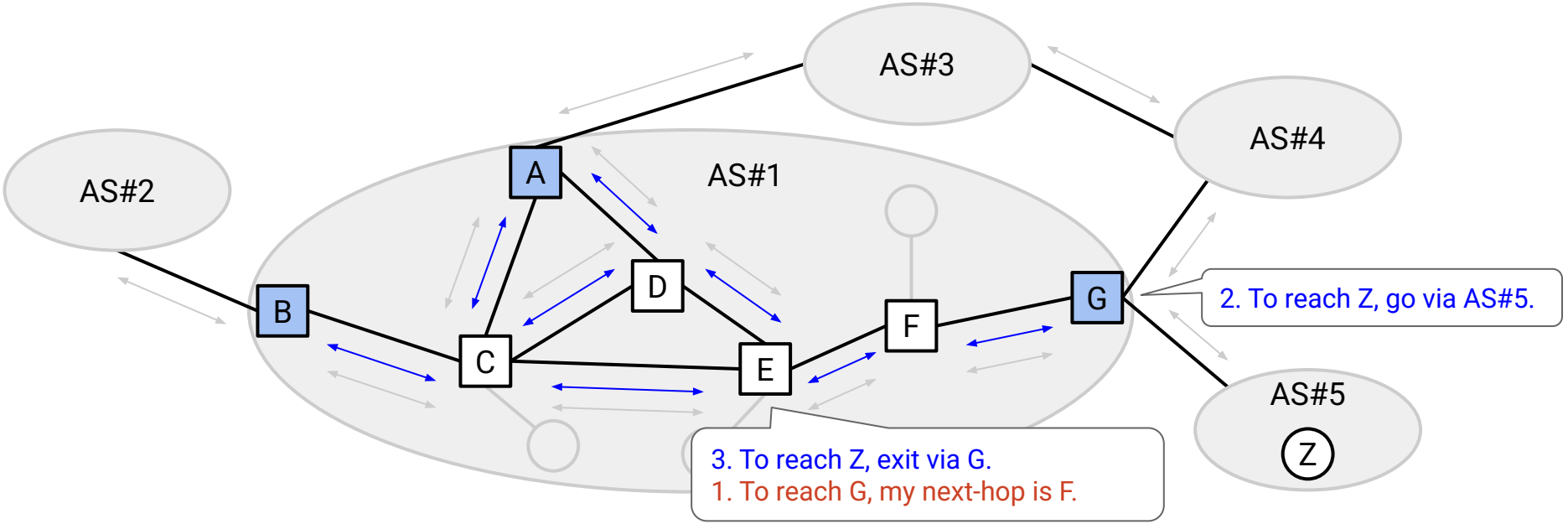
Combining Intra-Domain and Inter-Domain Routing

1. Use IGP to learn *internal* routes to others in the AS.
2. Use eBGP to learn *external* routes to other ASes.



Combining Intra-Domain and Inter-Domain Routing

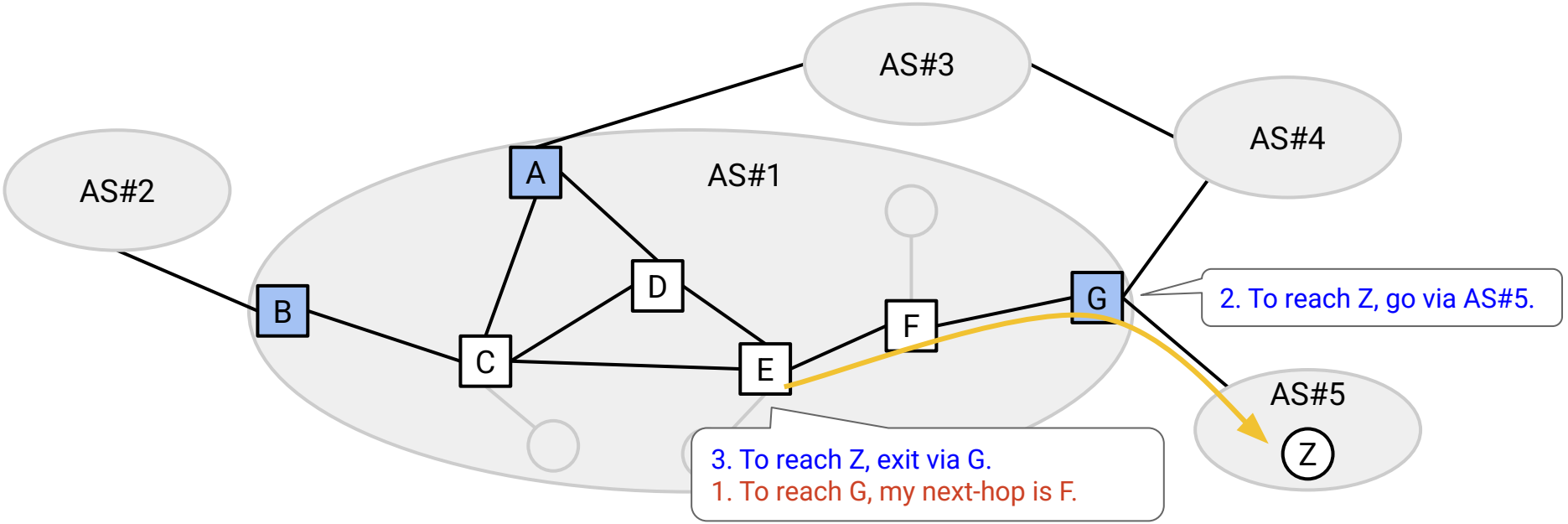
1. Use IGP to learn *internal* routes to others in the AS.
2. Use eBGP to learn *external* routes to other ASes.
3. Use iBGP to distribute *external* routes to others in the AS.



Combining Intra-Domain and Inter-Domain Routing

An **egress router** for a destination is a border router who can reach that destination.

- Example: Inside AS#1, G is an egress router for Z.
- G uses iBGP to tell local routers: "If you have packets for Z, send them to me."



BGP and IGP Routing Tables

Every router now has two routing tables:

- IGP table: Maps each *internal* destination to a next-hop.
- BGP table: Maps each *external* destination to an egress router.
  - The egress router is not necessarily a direct neighbor.

The two tables can be used to find the next hop for any destination.

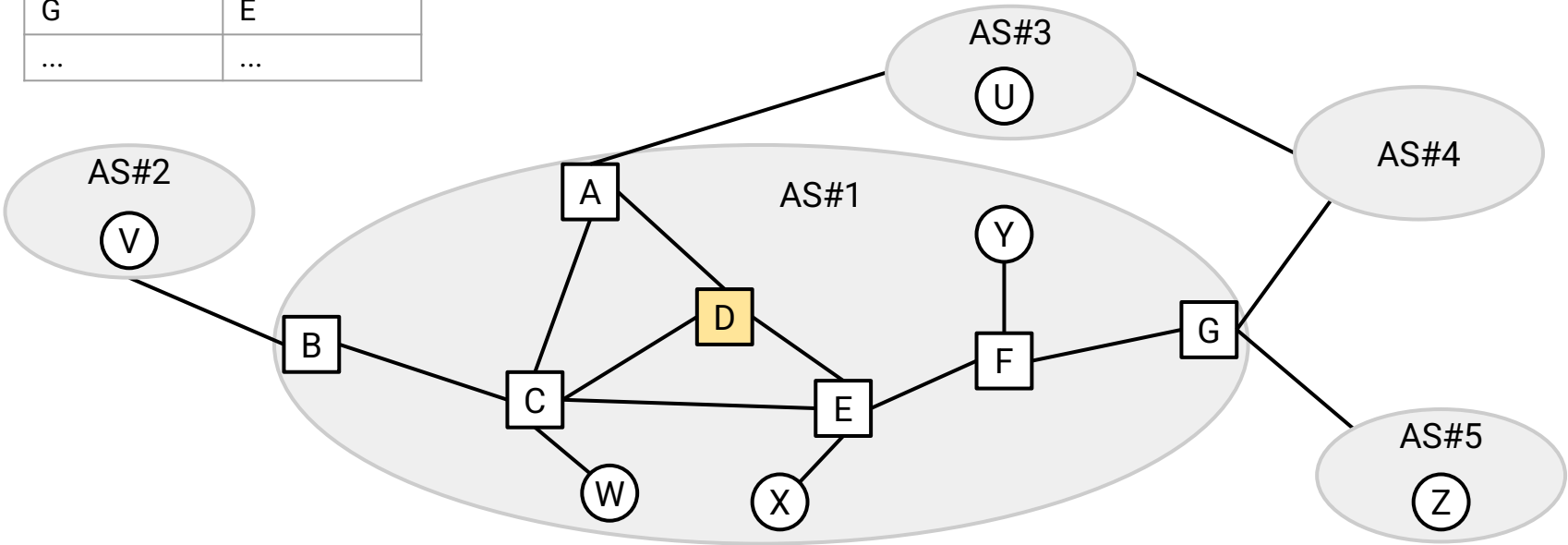
- Internal destinations: Just use the IGP table to find the next hop.
- External destinations:
  - First, use the BGP table to find the egress router.
  - Then, use the IGP table to find the next-hop to the egress.

BGP and IGP Routing Tables

Internal Destinations	
Destination	Next Hop
B	C
W	C
X	E
Y	E
G	E
...	...

External Destinations	
Destination	Egress via
U	A
V	B
Z	G

← D's two routing tables.



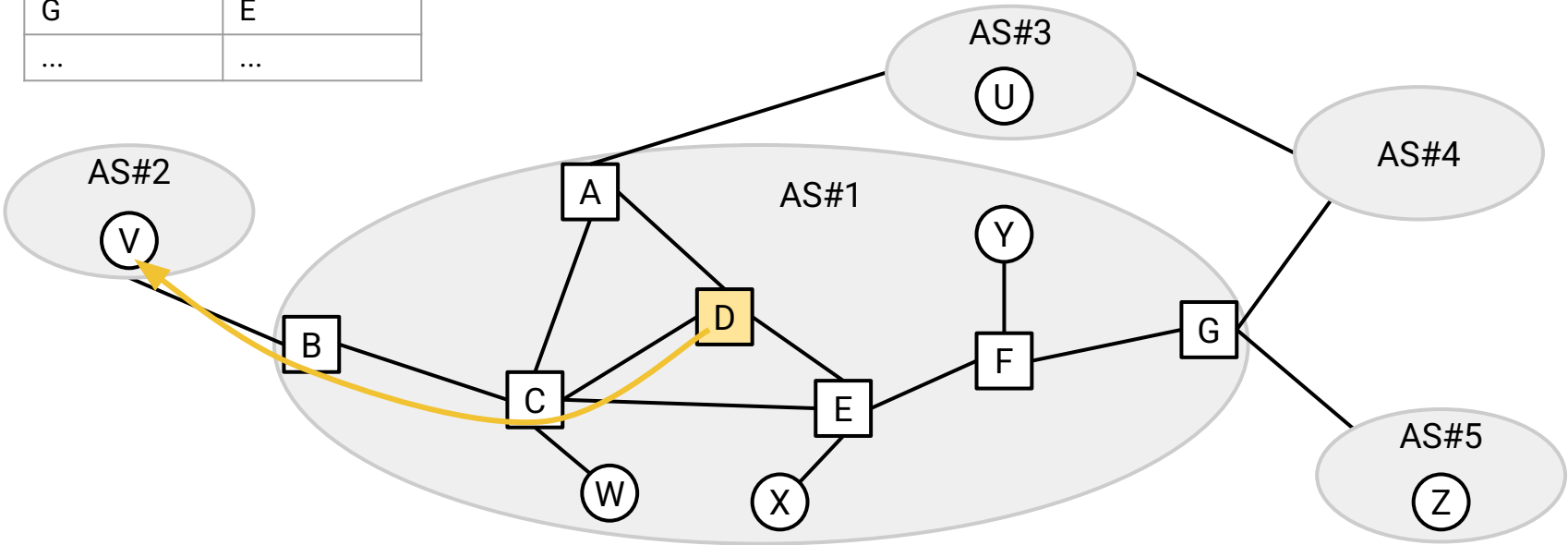
BGP and IGP Routing Tables

Internal Destinations	
Destination	Next Hop
B	C
W	C
X	E
Y	E
G	E
...	...

External Destinations	
Destination	Egress via
U	A
V	B
Z	G

To reach V, the egress router is B.

To reach B, the next hop is C.

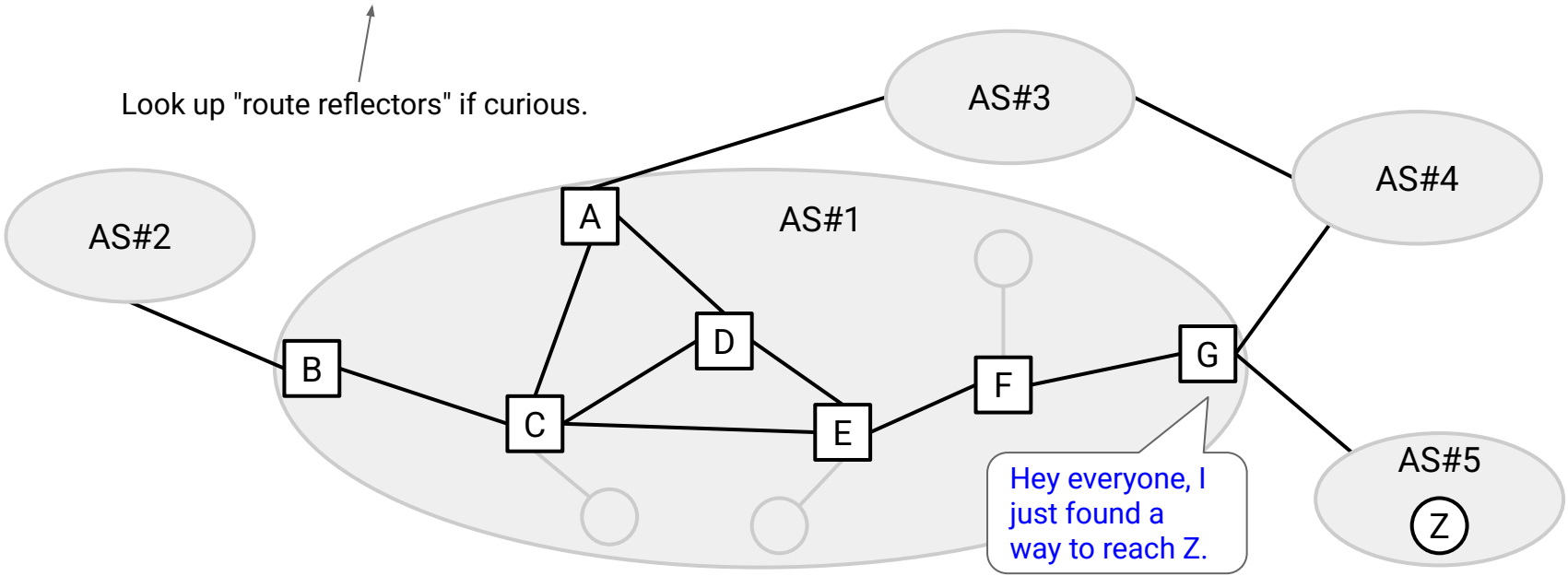


Implementing iBGP

Border routers use iBGP to distribute the routes it discovers to everyone else in the AS.

How is this actually implemented?

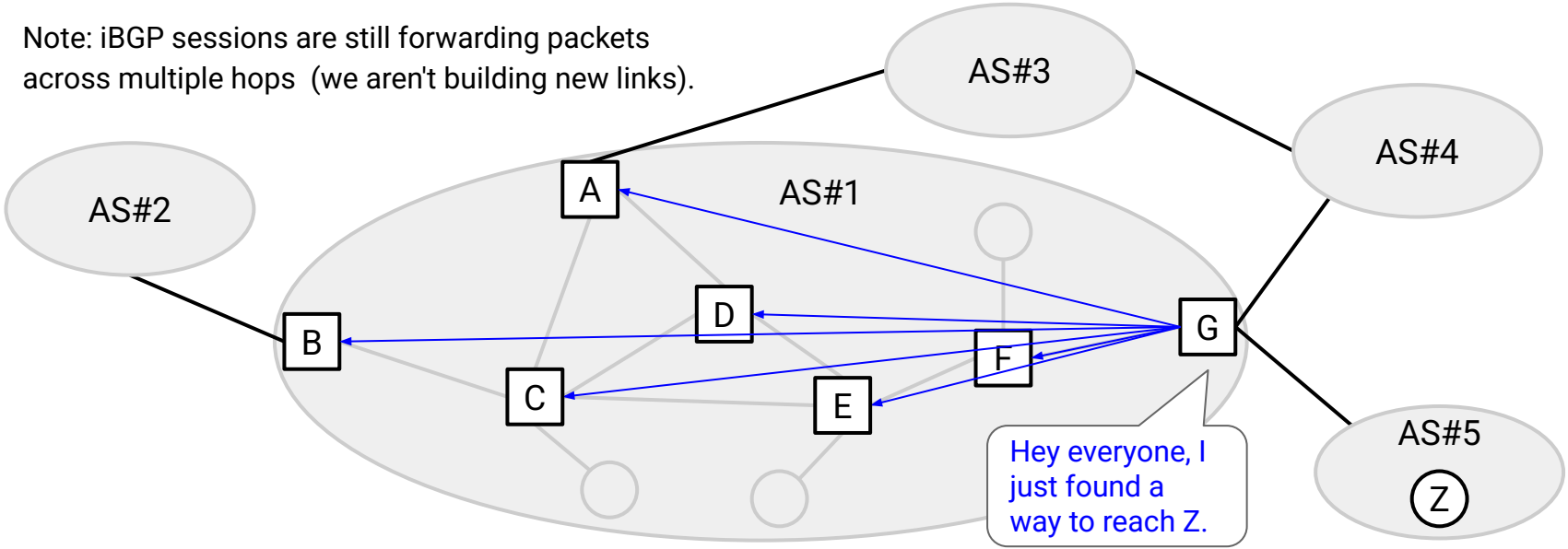
Many different approaches, though we'll assume a simple one.



Implementing iBGP

The simple approach: Run an iBGP session between every pair of routers.

- For N total routers and B border routers, requires ~ N × B sessions.
  - Example: N = 7, B = 3 inside AS#1.
- Could be a scaling issue on larger networks.



Multiple Links Between ASes

Lecture 10, CS 168, Spring 2025

BGP Implementation

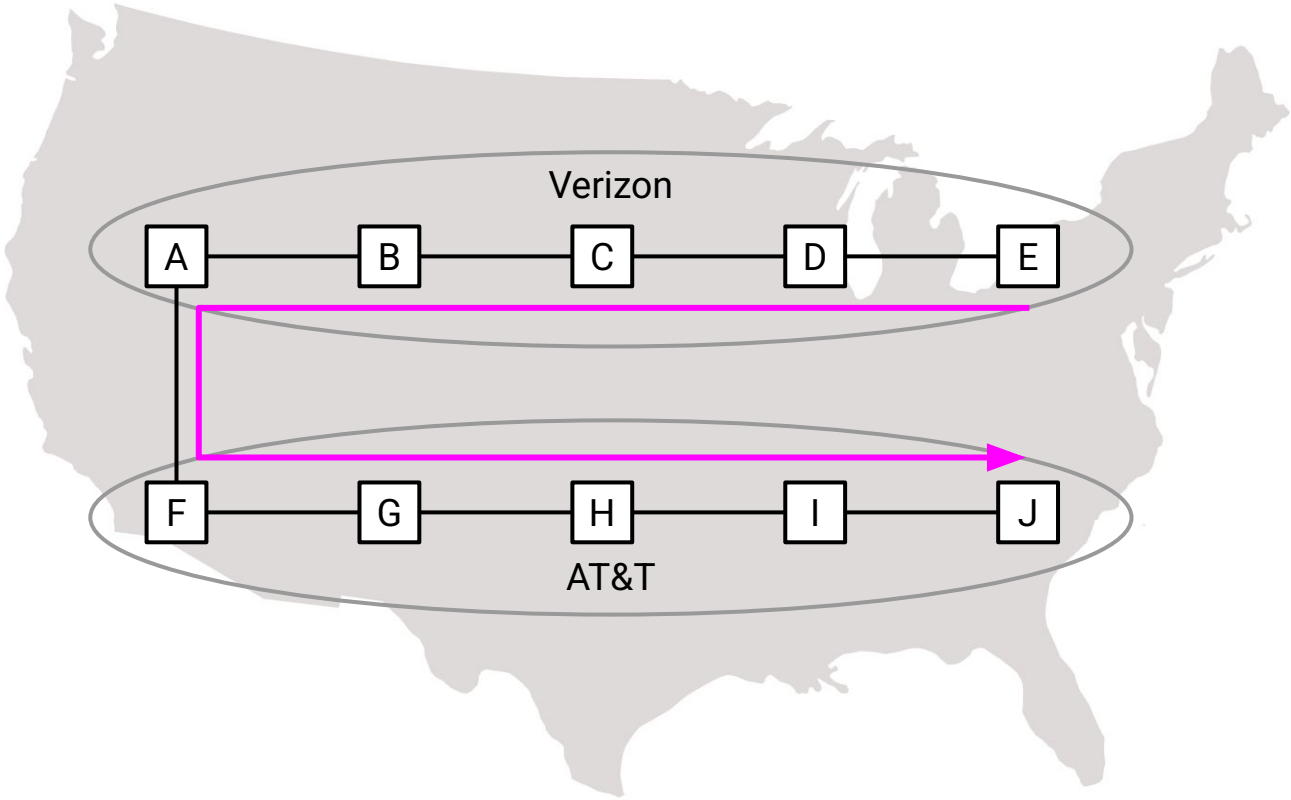
- External BGP and Internal BGP
- **Multiple Links Between ASes**
- Message Types and Route Attributes
- Issues with BGP

IP Header

- IPv4 Header Fields
- IPv6 Changes
- Security

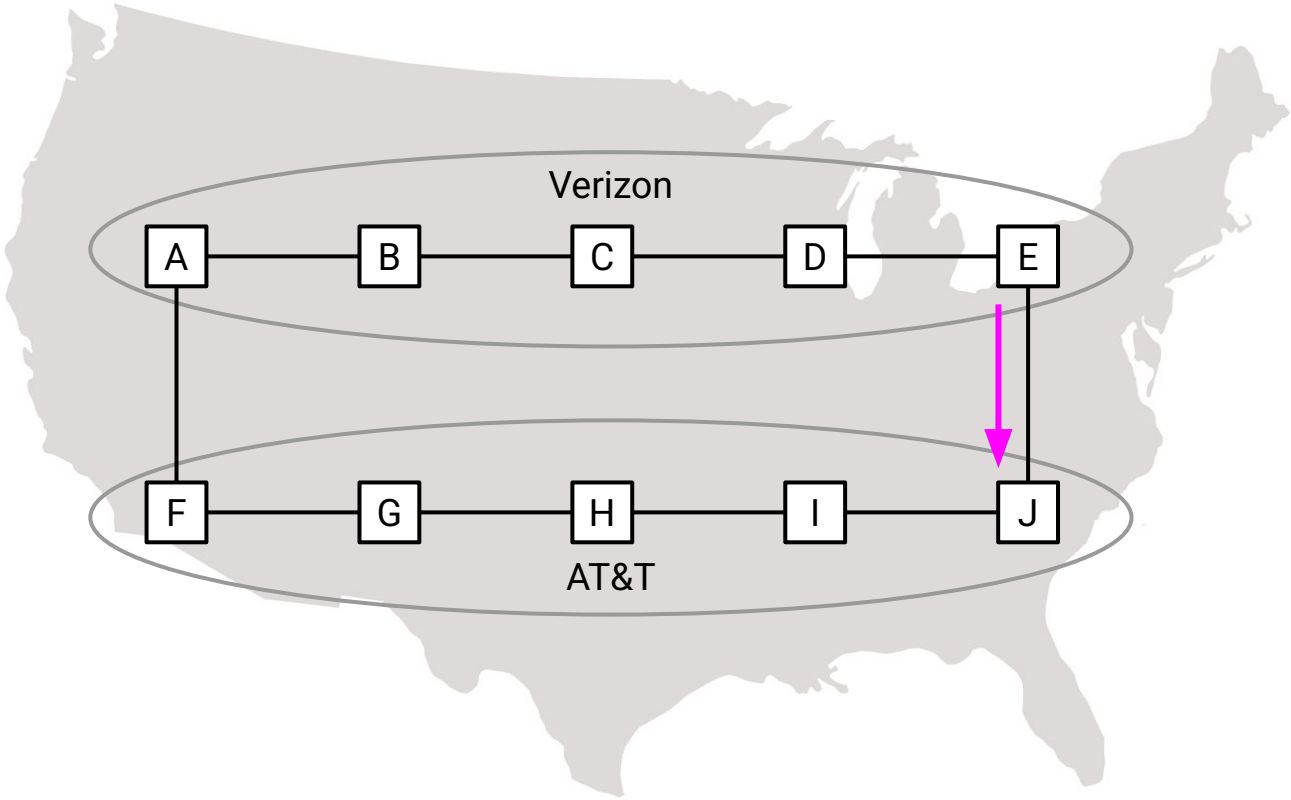
Multiple Links Between ASes

With only a single link between AT&T and Verizon, packets might travel long routes.



Multiple Links Between ASes

Adding a second link between AT&T and Verizon creates shorter routes.



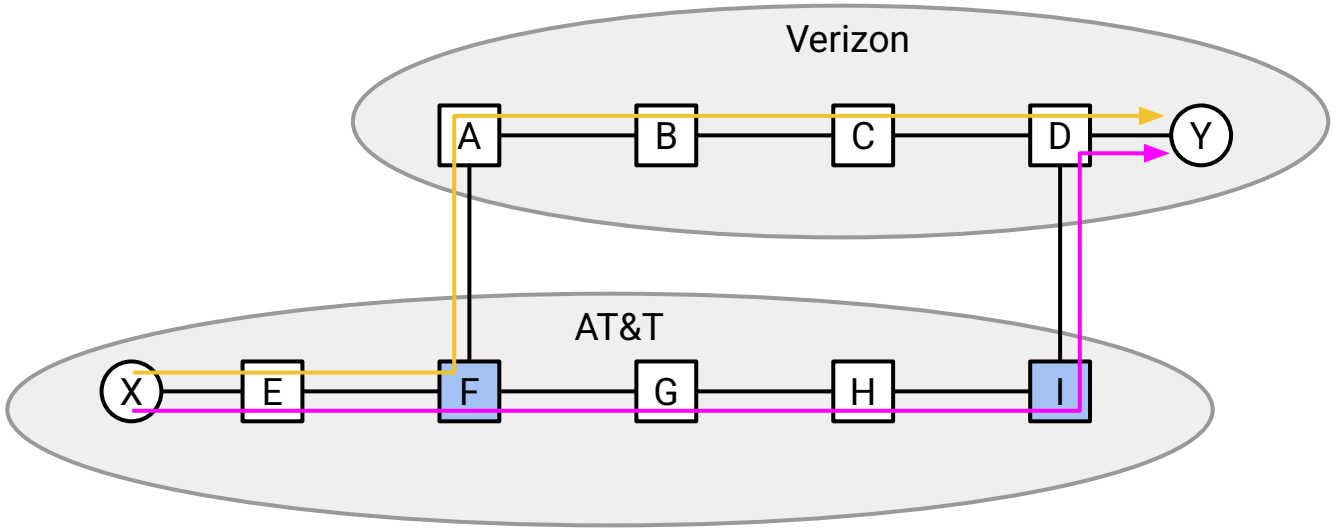
Multiple Links Between ASes

There can be multiple paths between the same two ASes.

- From the AS graph perspective, both routes are identical.

If you're AT&T, which route do you prefer?

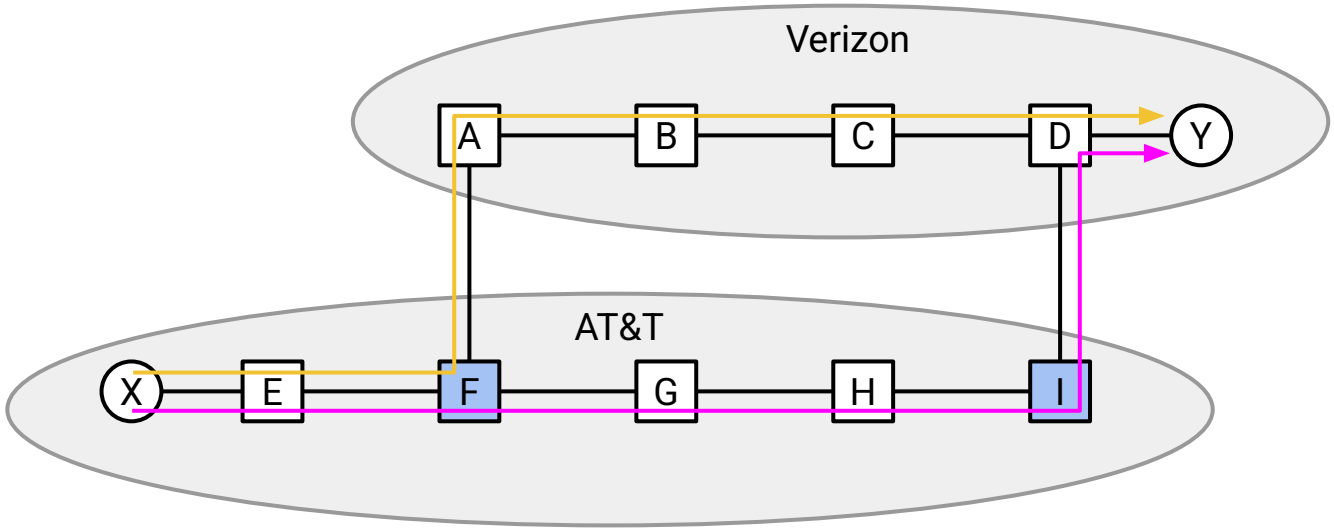
- Gao-Rexford rules won't help – the next AS is Verizon in both routes.



Hot Potato Routing

All else equal, an AS prefers the route that uses the least of its own resources.

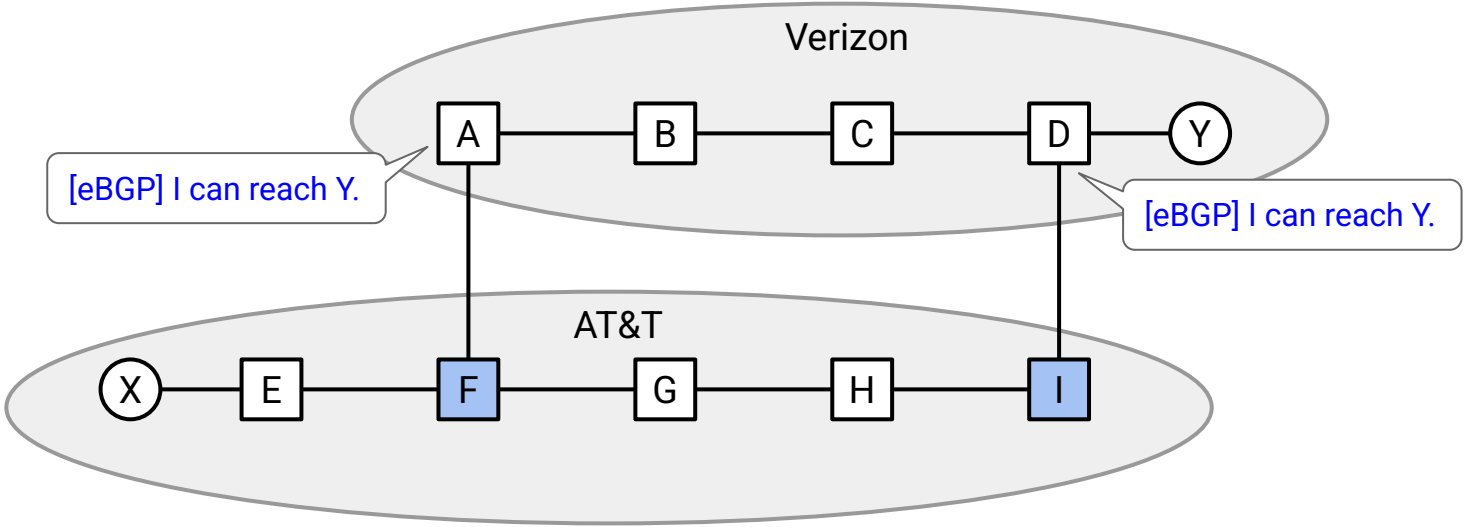
- AT&T prefers the gold path, where the packet mostly travels on Verizon's AS.
- **Hot potato routing**: Get rid of the packet ASAP, and make someone else carry it the rest of the way.
- As AS prefers the nearest egress router (if there are multiple).



Hot Potato Routing

All else equal, an AS prefers the route that uses the least of its own resources.

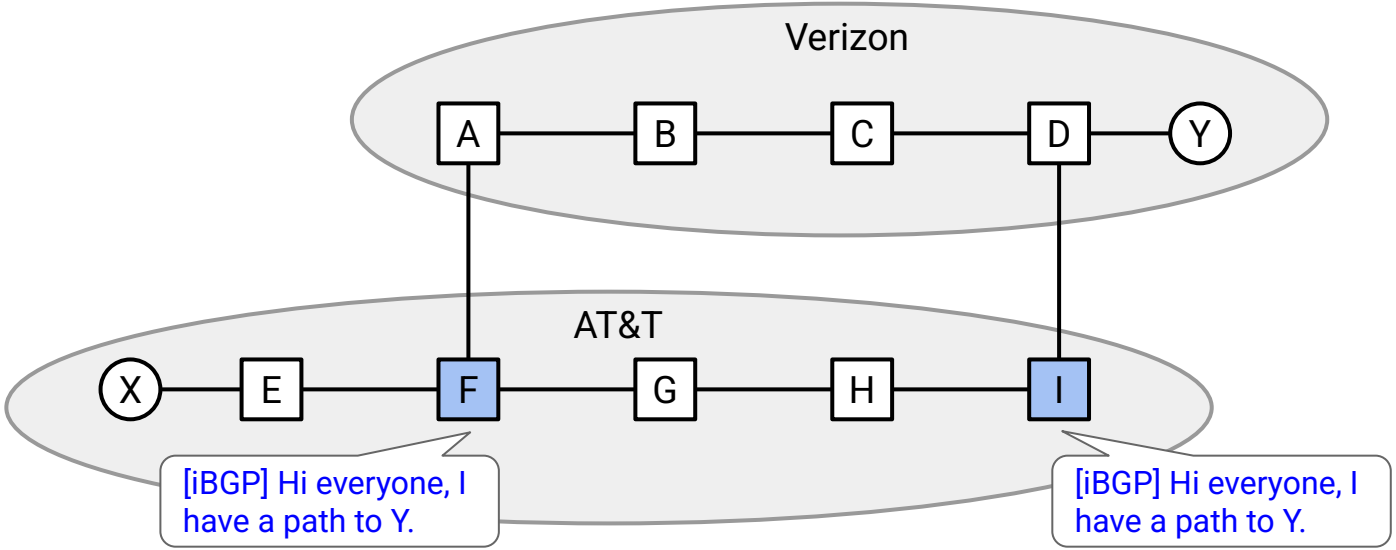
- AT&T prefers the gold path, where the packet mostly travels on Verizon's AS.
- **Hot potato routing**: Get rid of the packet ASAP, and make someone else carry it the rest of the way.
- As AS prefers the nearest egress router (if there are multiple).



Hot Potato Routing

All else equal, an AS prefers the route that uses the least of its own resources.

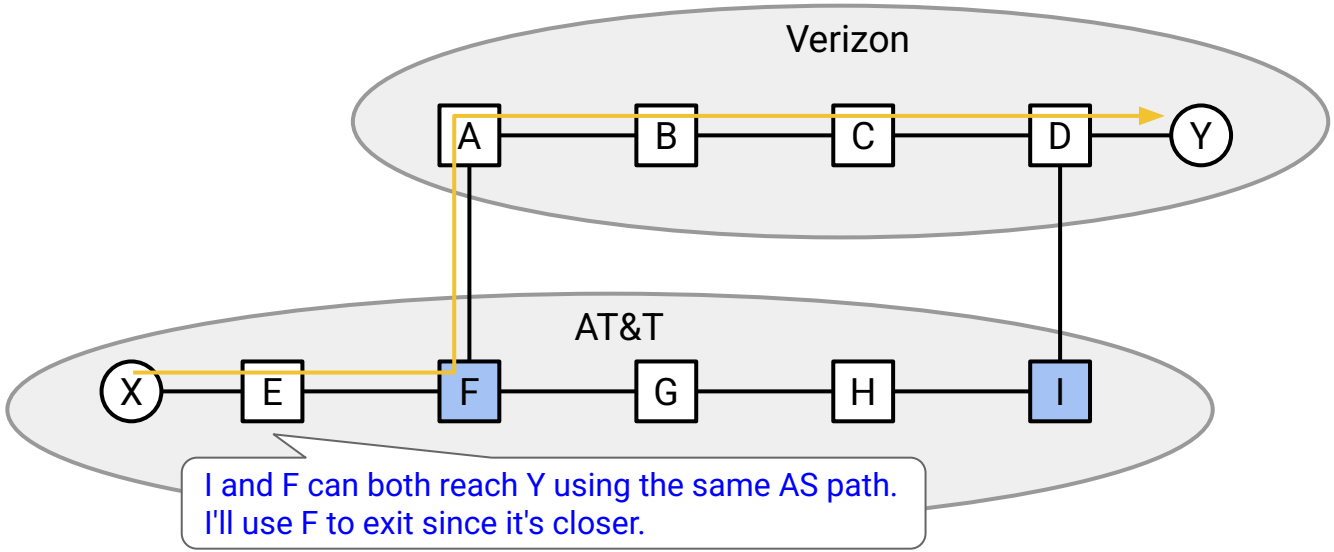
- AT&T prefers the gold path, where the packet mostly travels on Verizon's AS.
- **Hot potato routing**: Get rid of the packet ASAP, and make someone else carry it the rest of the way.
- As AS prefers the nearest egress router (if there are multiple).



Hot Potato Routing

All else equal, an AS prefers the route that uses the least of its own resources.

- AT&T prefers the gold path, where the packet mostly travels on Verizon's AS.
- **Hot potato routing**: Get rid of the packet ASAP, and make someone else carry it the rest of the way.
- As AS prefers the nearest egress router (if there are multiple).

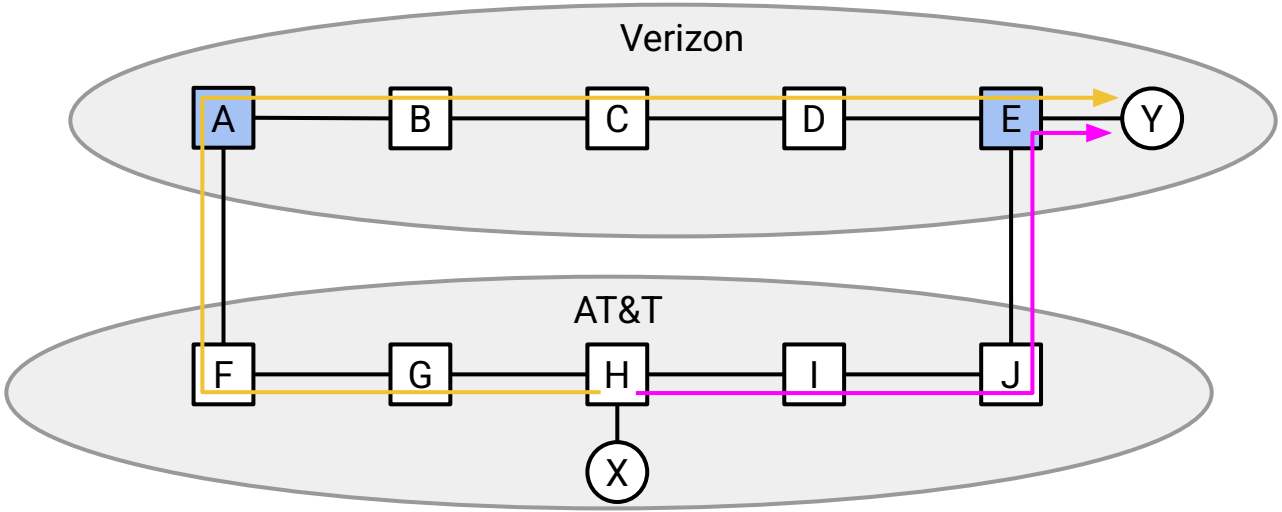


Multiple Links Between ASes

What if two egress routers are equally close?

AT&T is indifferent between the paths.

If you're Verizon, which route do you prefer?



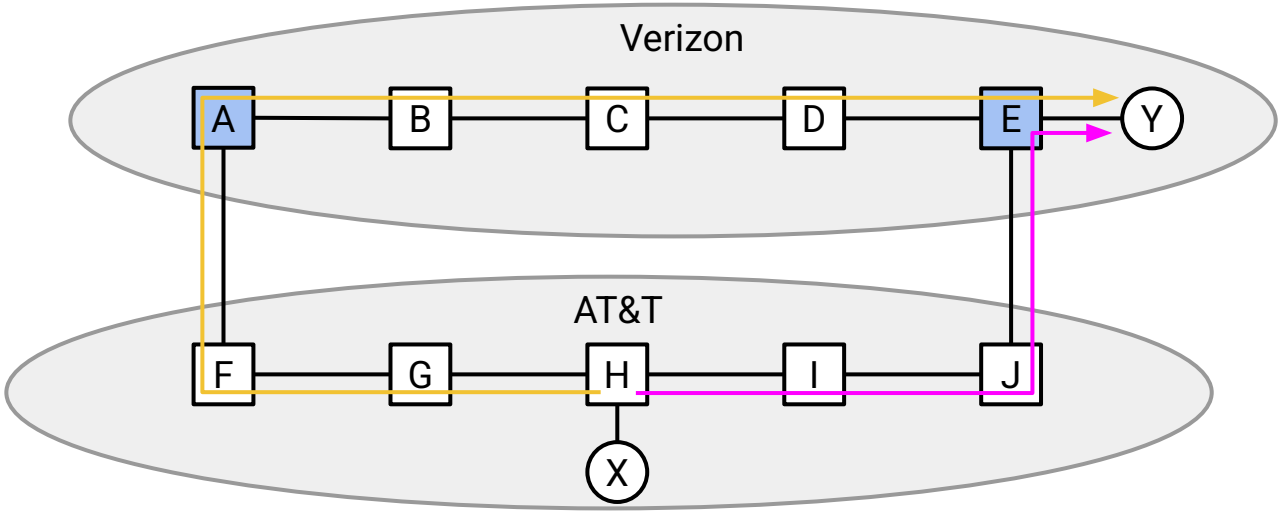
Multi-Exit Discriminator (MED)

All else equal, an AS prefers the route that uses the least of its own resources.

- Verizon prefers the pink path, because E is closer to the destination than A.
- Using the closer border router = Verizon uses less bandwidth.

When announcing routes, include a **Multi-Exit Discriminator (MED)**.

- MED represents distance from router to destination. Lower number is preferred.



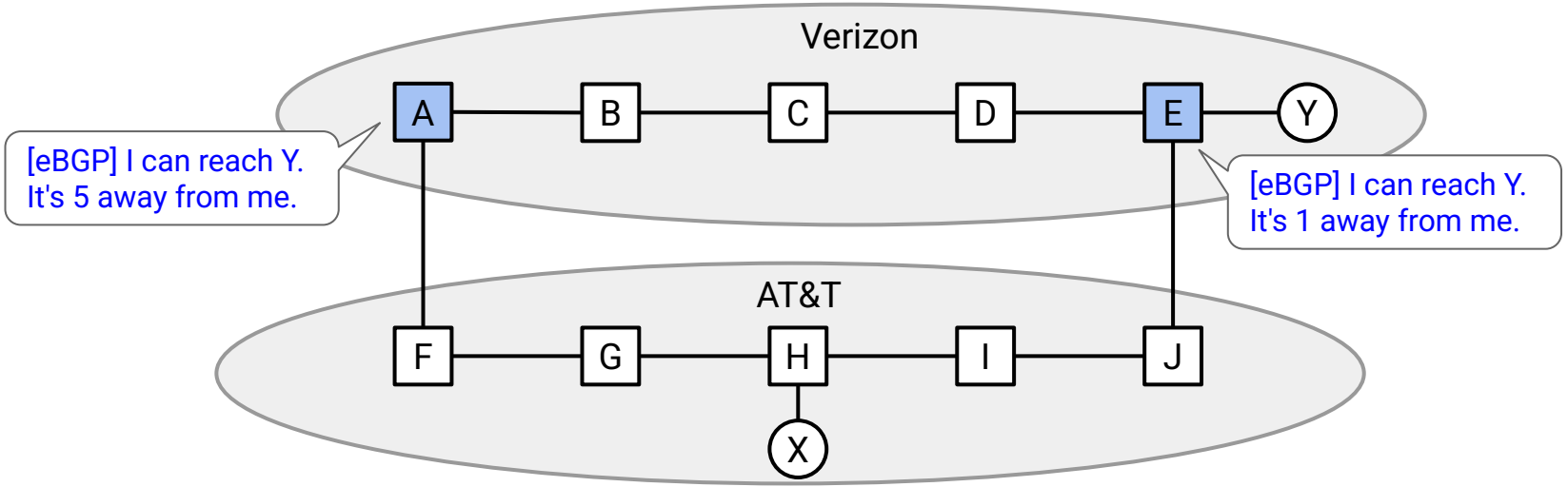
Multi-Exit Discriminator (MED)

All else equal, an AS prefers the route that uses the least of its own resources.

- Verizon prefers the pink path, because E is closer to the destination than A.
- Using the closer border router = Verizon uses less bandwidth.

When announcing routes, include a **Multi-Exit Discriminator (MED)**.

- MED represents distance from router to destination. Lower number is preferred.



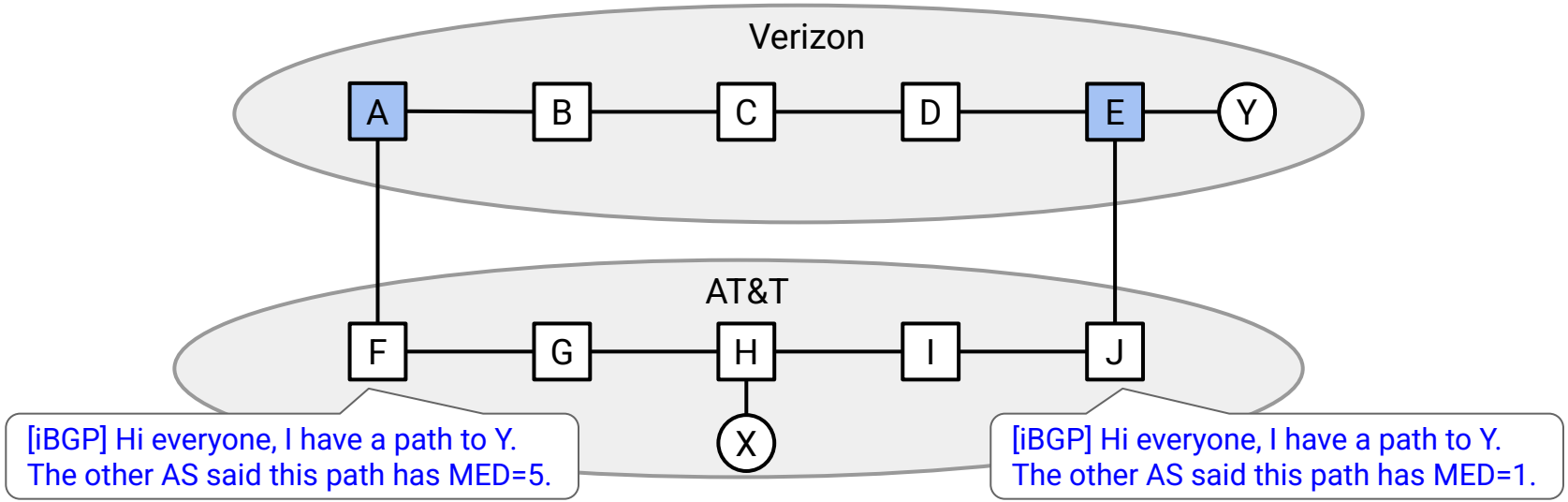
Multi-Exit Discriminator (MED)

All else equal, an AS prefers the route that uses the least of its own resources.

- Verizon prefers the pink path, because E is closer to the destination than A.
- Using the closer border router = Verizon uses less bandwidth.

When announcing routes, include a **Multi-Exit Discriminator (MED)**.

- MED represents distance from router to destination. Lower number is preferred.



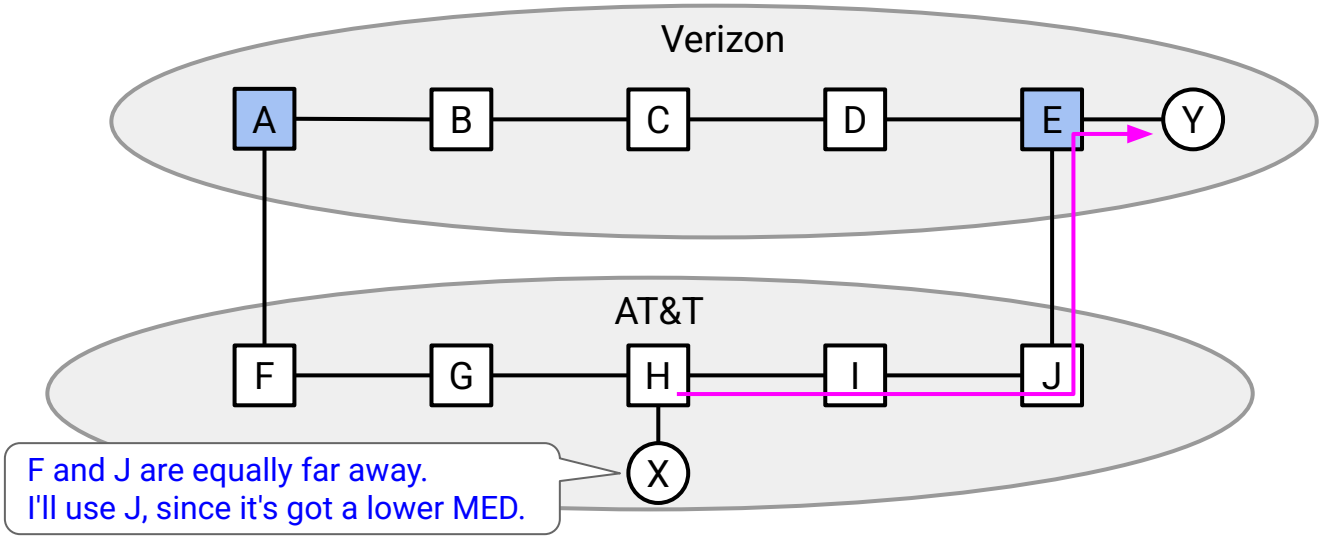
Multi-Exit Discriminator (MED)

All else equal, an AS prefers the route that uses the least of its own resources.

- Verizon prefers the pink path, because E is closer to the destination than A.
- Using the closer border router = Verizon uses less bandwidth.

When announcing routes, include a **Multi-Exit Discriminator (MED)**.

- MED represents distance from router to destination. Lower number is preferred.



Importing and Exporting with Tiebreaking Policies

Exporting rules:

- Use Gao-Rexford rules to decide which ASes hear about this route.
- In the announcement, include the MED (distance from router to destination). Lower MED = Prefer to use this router.

Importing rules, in order of priority:

1. Use Gao-Rexford rules (customer > peer > provider).
    - Goal: Save money.
2. If 1 tied: Pick the shorter path (the path passing through fewer ASes).
    - Goal: Maximize performance.
3. If 1-2 tied: Pick the path with the nearest egress router (hot potato routing).
    - Goal: Minimize use of my network bandwidth.
4. If 1-3 tied: Pick the path with the lower MED.
5. If 1-4 tied: Arbitrary tiebreaker (e.g. pick router with lower IP address).

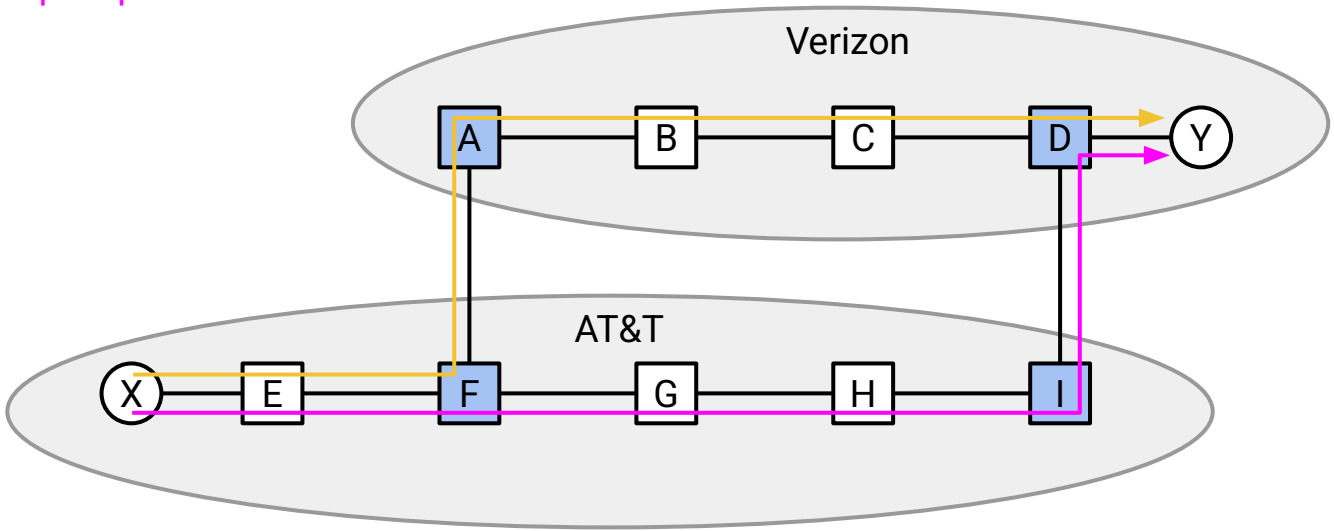
Hot Potato Routing and MED are Contradictory

The importing and exporting ASes *both* want to minimize their own link usage.

- Importing AS does this with hot potato routing: Pick the nearest egress.
- Exporting AS does this with MED: Send me the packet on the closest router.

AT&T prefers gold path.

Verizon prefers pink path.



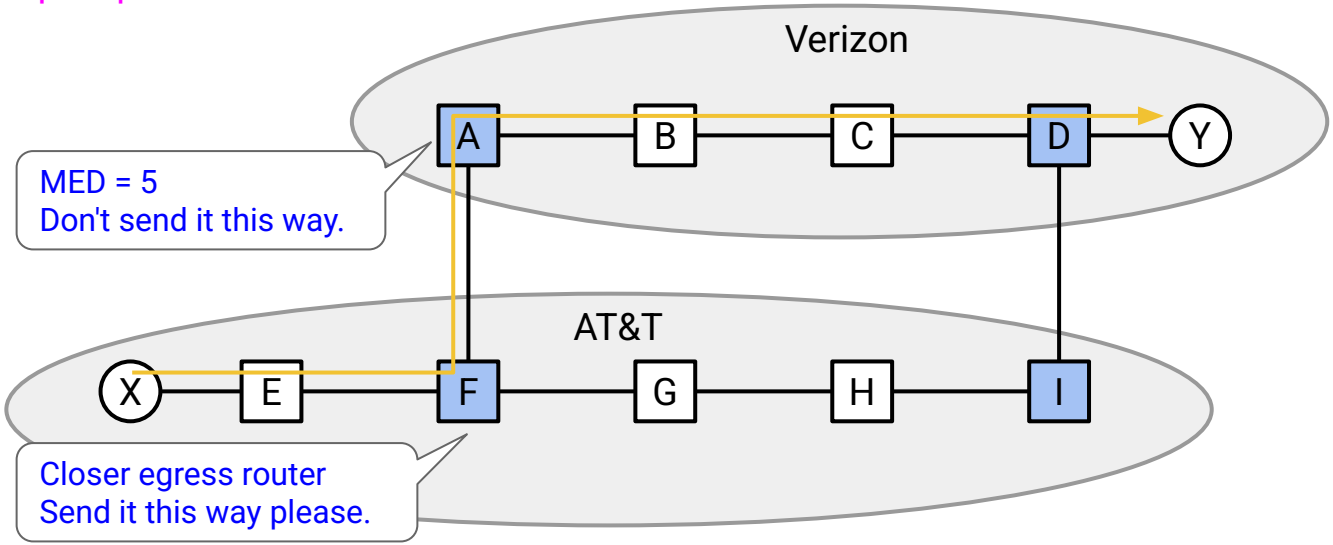
Hot Potato Routing and MED are Contradictory

The importing and exporting ASes *both* want to minimize their own link usage.

- Importing AS does this with hot potato routing: Pick the nearest egress.
- Exporting AS does this with MED: Send me the packet on the closest router.

AT&T prefers gold path.

Verizon prefers pink path.



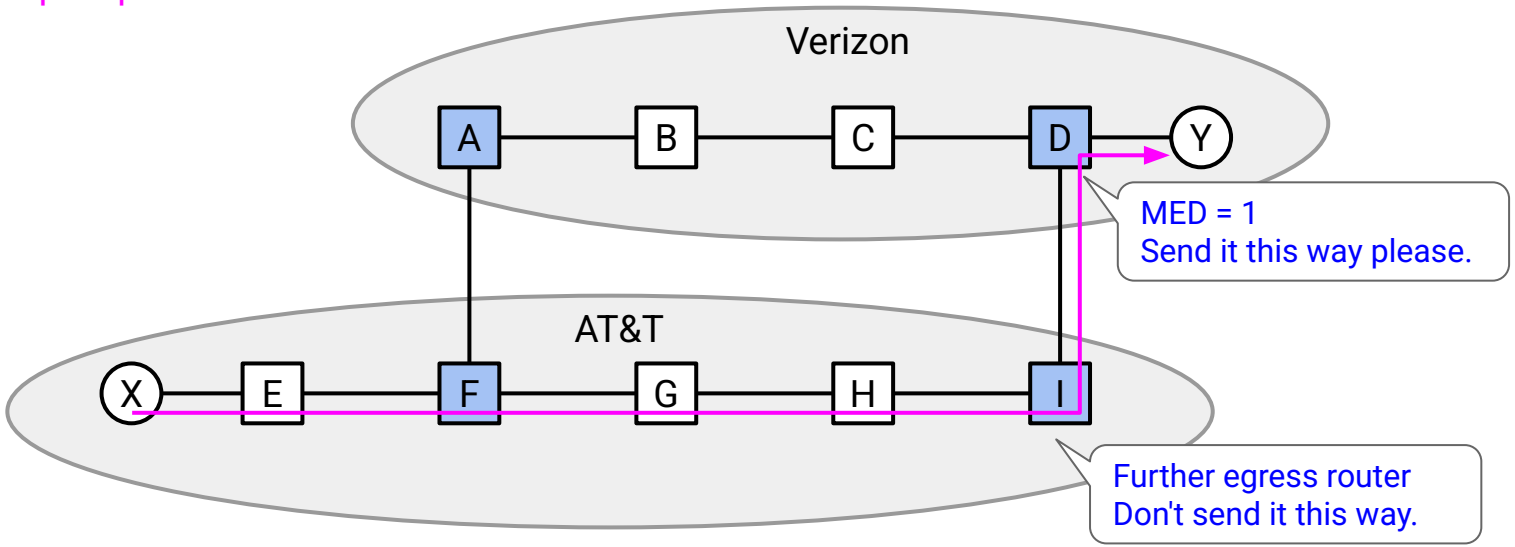
Hot Potato Routing and MED are Contradictory

The importing and exporting ASes *both* want to minimize their own link usage.

- Importing AS does this with hot potato routing: Pick the nearest egress.
- Exporting AS does this with MED: Send me the packet on the closest router.

AT&T prefers gold path.

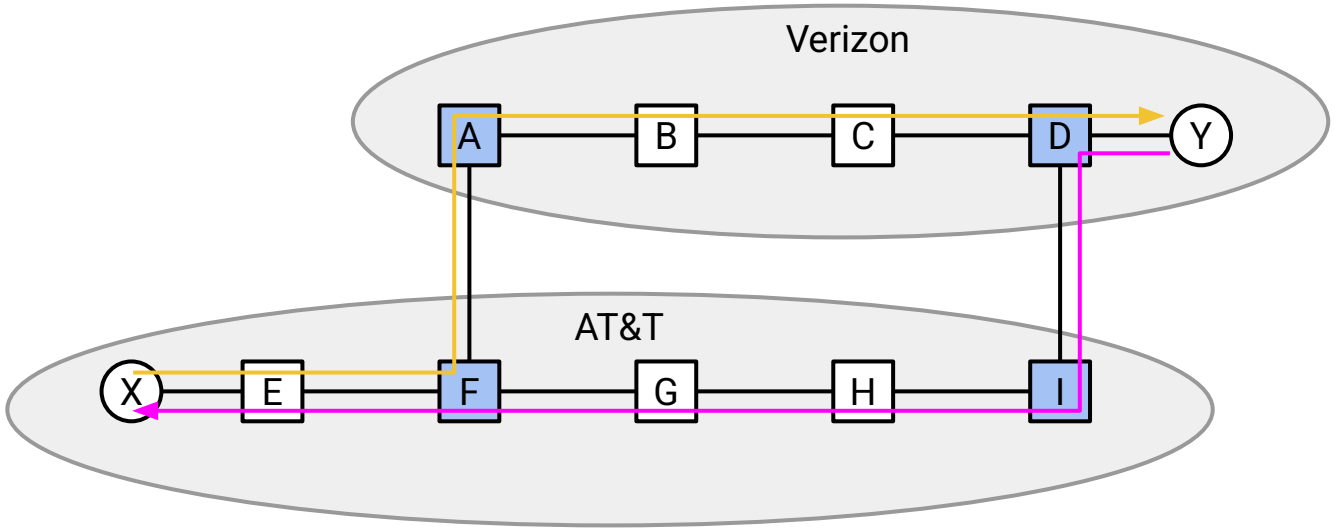
Verizon prefers pink path.



Hot Potato Routing and MED are Contradictory

Hot potato routing can lead to asymmetric paths.

- X → Y traffic: AT&T ditches packet quickly, Verizon carries it most of the way.
- Y → X traffic: Verizon ditches packet quickly, AT&T carries it most of the way.



Message Types, Route Attributes

Lecture 10, CS 168, Spring 2025

BGP Implementation

- External BGP and Internal BGP
- Multiple Links Between ASes
- **Message Types and Route Attributes**
- Issues with BGP

IP Header

- IPv4 Header Fields
- IPv6 Changes
- Security

Implementing BGP: Message Types

To actually implement BGP, we need to specify:

- Syntax: Format of BGP message.
- Semantics: How to process BGP messages.

There are 4 BGP message types:

- Open: Start a BGP session.
- KeepAlive: I'm still here, don't close the BGP session.
- Notification: An error occurred.
- Update: Announce a route.
  - Could be a new route, an update to an old route, or withdrawing a route.
  - We'll focus on this one.

BGP Update Message Format

Update message encodes an announcement using two values:

- Destination prefix.
- One or more **route attributes**.

Route attributes:

- Used to express properties about routes.
- Can be used in route import/export decisions.
- Written as name-value pairs.
- Some attributes are local: Only announced in iBGP.
- Some attributes are global: Announced in eBGP.
- There are many standardized attributes, but we'll look at 3.

Route Attribute #1: ASPATH

Name: **ASPATH**. Value: List of all ASes in this route, in reverse order.

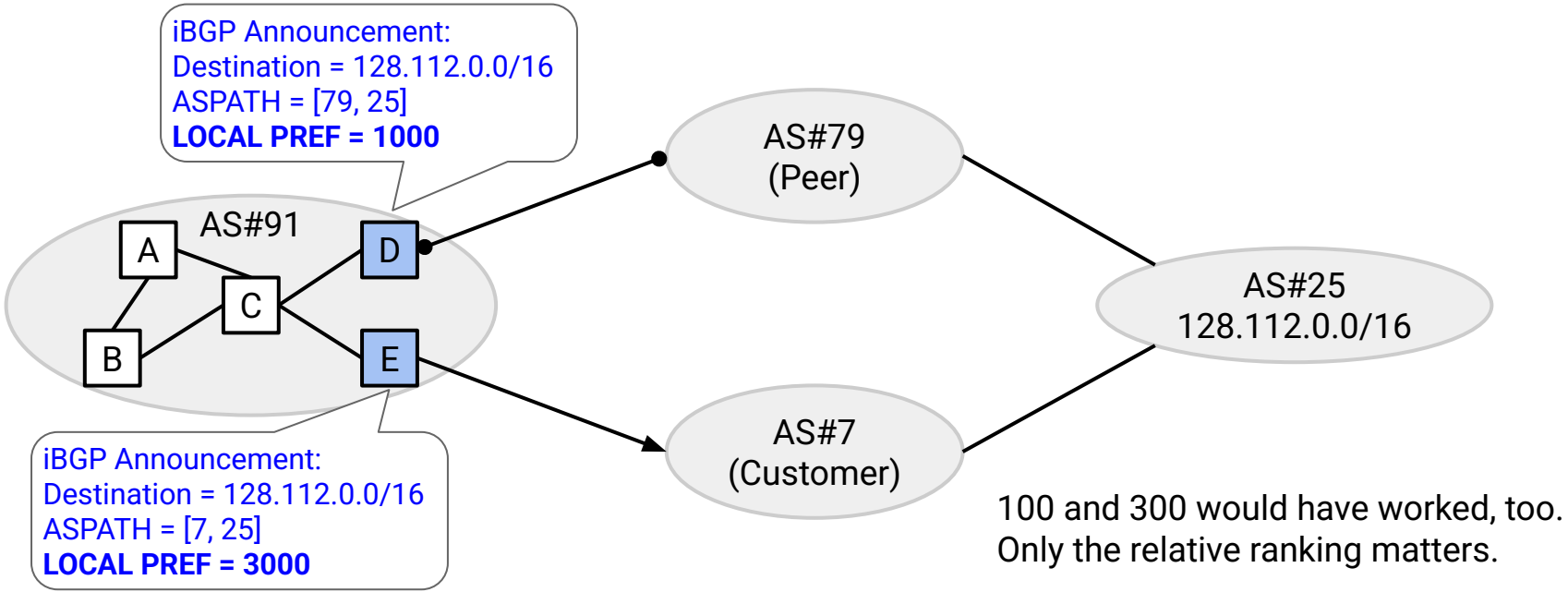
Scope: Global (included in both eBGP and iBGP announcements).

Route Attribute #2: LOCAL PREFERENCE

Name: **LOCAL PREFERENCE**. Value: A number, *higher* is better.

Scope: Local (only included in iBGP announcements).

Used to encode policy preference between AS paths (e.g. Gao-Rexford rules).



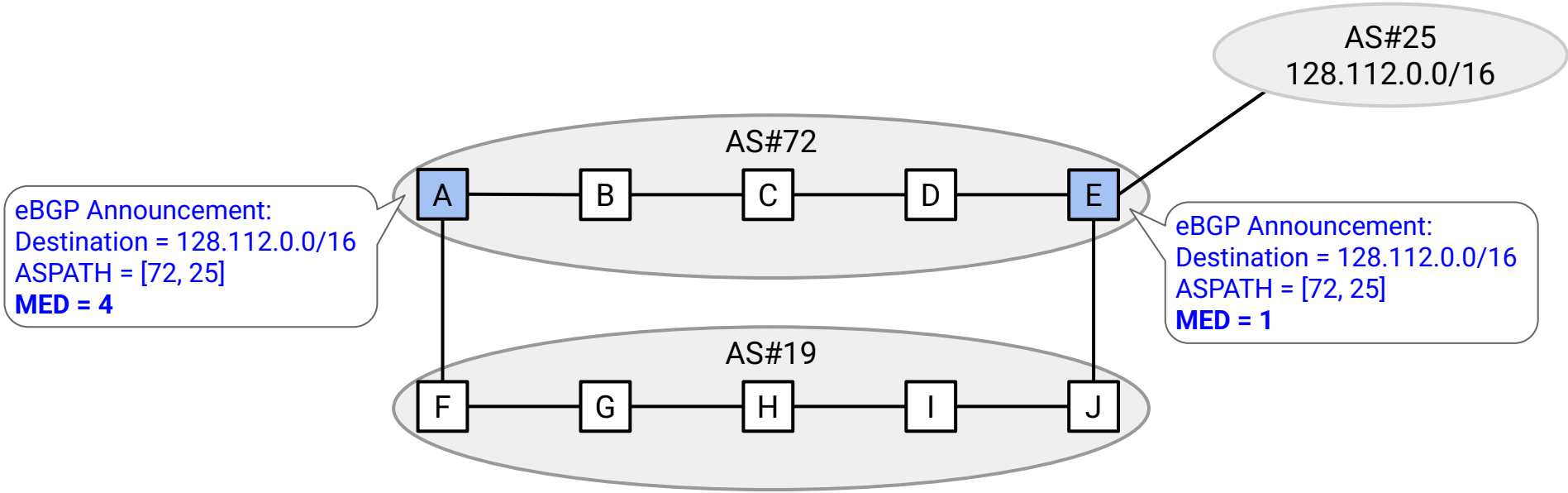
Route Attribute #3: MED

Name: **MED**. Value: A number, *lower* is better.

Scope: Global (included in both eBGP and iBGP announcements).

Used when there are multiple links between ASes.

Specifies distance from announcing router to destination.



Import Policy with Attributes

Priority	Rule	Implementation	Goal
1	Gao-Rexford rules	Highest LOCAL PREF	Make/save money
2	Shortest path	Shortest ASPATH length	Maximize performance
3	Hot potato routing	Nearest egress router (lowest IGP cost)	Minimize use of my bandwidth
4	Multiple links: Send to router closer to destination	Smallest MED	Minimize use of other AS's bandwidth
5	Arbitrary	Lowest router IP	N/A

Issues with BGP

Lecture 10, CS 168, Spring 2025

BGP Implementation

- External BGP and Internal BGP
- Multiple Links Between ASes
- Message Types and Route Attributes
- **Issues with BGP**

IP Header

- IPv4 Header Fields
- IPv6 Changes
- Security

Issues with BGP

Security:

- No guarantee that an AS owns the prefixes it advertises.
- No guarantee that an AS will follow the path it advertises.

Performance trade-offs:

- Policy-based paths are not always least-cost.
- AS path length can be misleading (path inside an AS could be 1 hop or 100 hops).

Complicated and prone to misconfiguration:

- Many BGP attributes. Configuration is often manual and ad-hoc.
- BGP misconfiguration is a major source of Internet outages!

Invalid routes:

- Reachability and convergence are not guaranteed without Gao-Rexford.

IPv4 Header Fields

Lecture 10, CS 168, Spring 2025

BGP Implementation

- External BGP and Internal BGP
- Multiple Links Between ASes
- Message Types and Route Attributes
- Issues with BGP

IP Header

- **IPv4 Header Fields**
- IPv6 Changes
- Security

Introducing the IP Header

Our goal:

- Understand all of the fields in this header.
- Understand the design choices in making this header.

<table>
<tr><td>Version (4)</td><td>Hdr len (4)</td><td>Type of Service (8)</td><td colspan="2">Total Length in Bytes (16)</td></tr>
<tr><td colspan="3">Identification (16)</td><td>Flags (3)</td><td>Fragment Offset (13)</td></tr>
<tr><td colspan="2">TTL (8)</td><td>Protocol (8)</td><td colspan="2">Header Checksum (16)</td></tr>
<tr><td colspan="5">Source IP Address (32)</td></tr>
<tr><td colspan="5">Destination IP Address (32)</td></tr>
<tr><td colspan="5">Options (if any)</td></tr>
<tr><td colspan="5">Payload</td></tr>
</table>

IP Header Design Goals

Think of the IP header as an interface.

- Allows the source and intermediate routers to exchange information.
- Allows the source and destination to exchange information.

Design goals:

- Small. A larger IP header makes every single packet larger.
- Simple. Routers have to process packets quickly.

What fields are in the IP header?

- It depends on what tasks IP needs to perform.

IP Tasks

IP functionality can be classified into 6 tasks.

Some tasks are done by only routers or the destination host. Others are done by both.

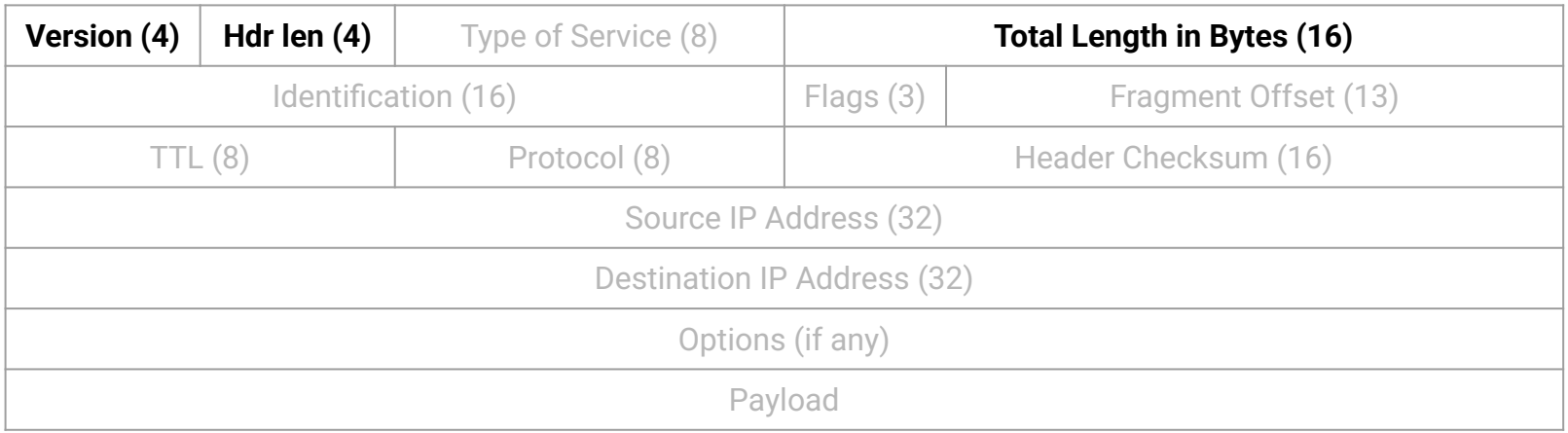
1. Parse the packet. (both router and destination)
2. Forward packet to the next hop. (router only)
3. Tell the destination what to do next. (destination only)
4. Send responses back to the source. (both router and destination)
5. Handle errors. (both router and destination)
6. Specify any special packet handling. (both router and destination)

Task #1: Parse the Packet

Version: What version of IP?

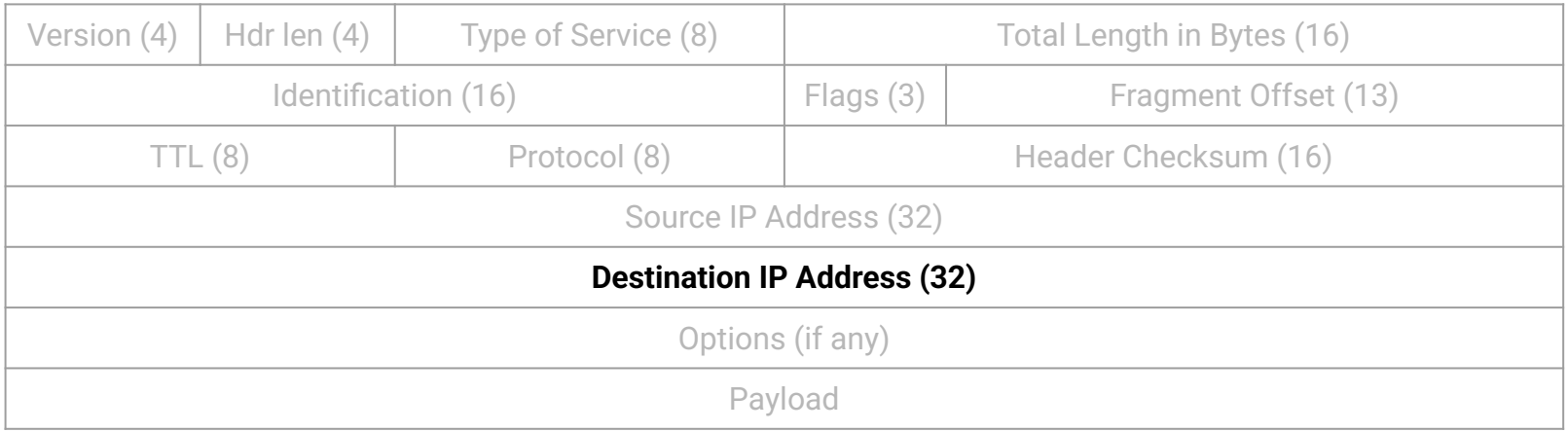
Header length (measured in 4-byte words): Where does the header end?

Total length (measured in bytes): Where does the packet end?



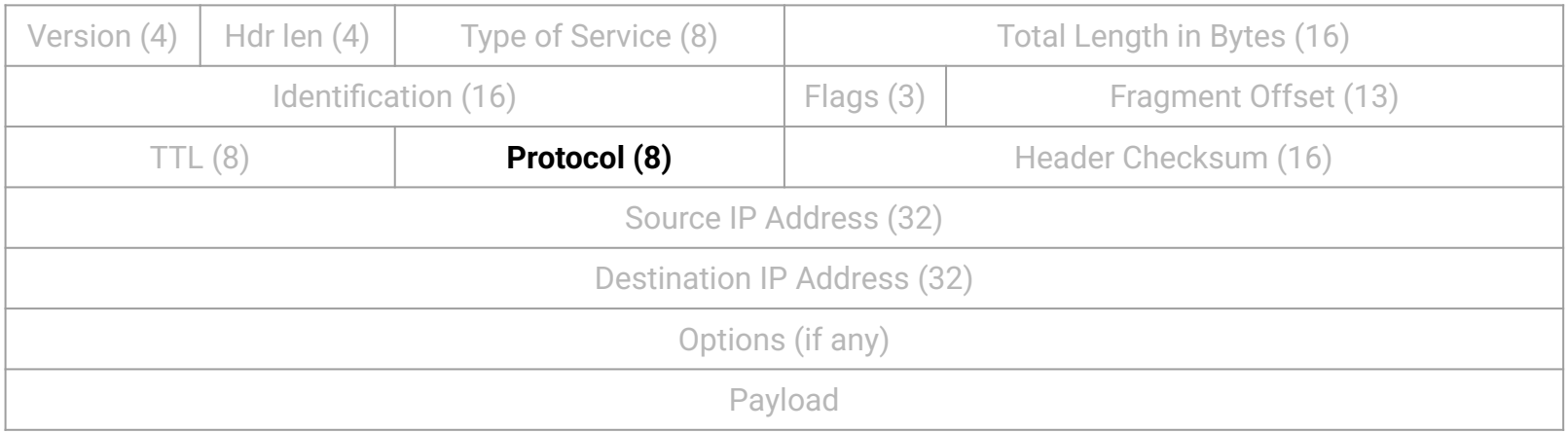
Task #2: Forward Packet to Next Hop

Destination IP address.



Task #3: Tell Destination What to Do Next

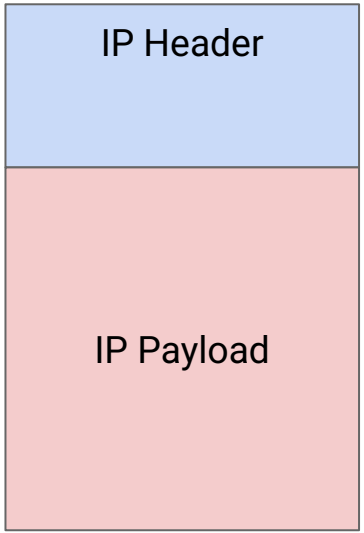
Protocol field: Identify the Layer 4 protocol to pass the payload to.



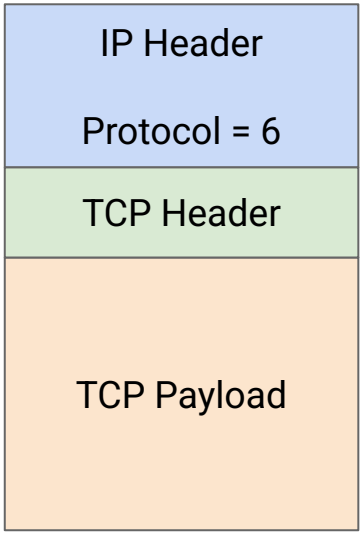
Task #3: Tell Destination What to Do Next

We need to indicate which protocol should handle the packet next.

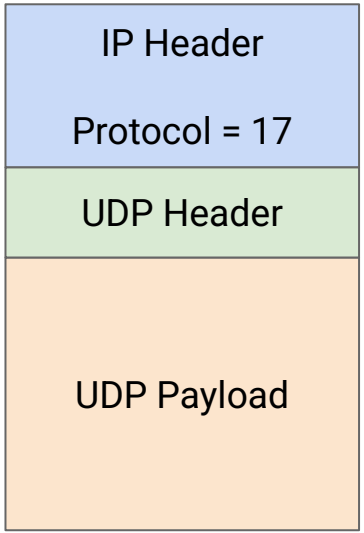
De-multiplexing: Picking one of several possible L4 protocols.



Without the protocol field, we have no idea what to do with the bits in red.



If we see protocol = 6:
Pass the red bits to TCP code.

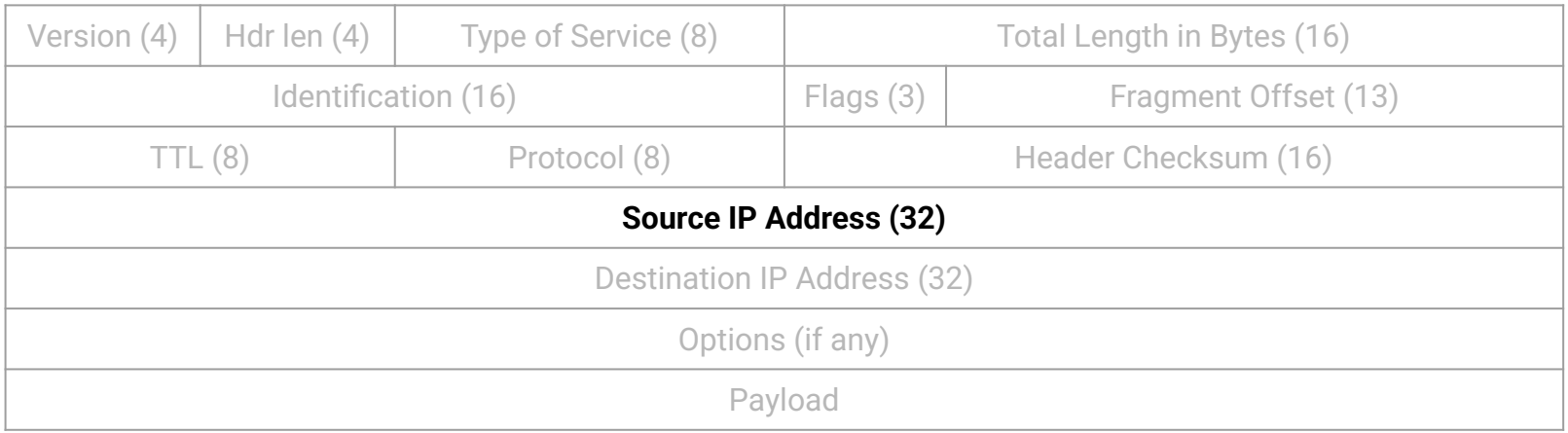


If we see protocol = 17:
Pass the red bits to UDP code.

Task #4: Send Responses Back to Source

Source IP address.

- Note: We said both the router and destination might do this.
- The router might want to send a response back to the source too (e.g. if an error occurs).

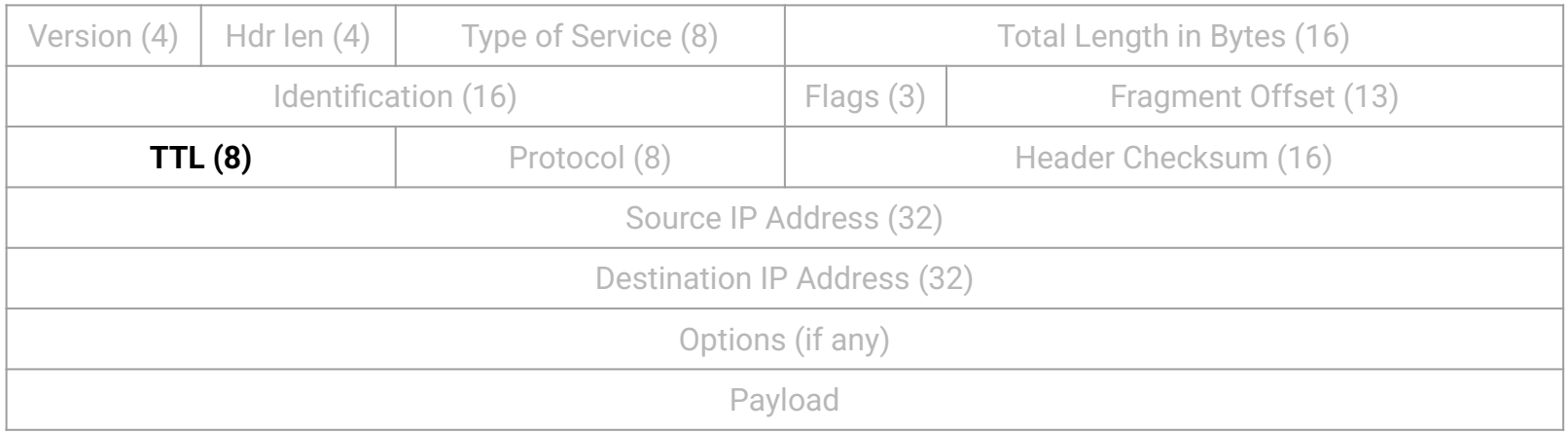


Task #5: Handle Errors: TTL

Problem: Forwarding loops cause packets to cycle indefinitely. Wastes bandwidth.

Time-to-live (TTL): Specifies maximum number of hops a packet can take.

- TTL is decremented at every hop by the router.
- If router receives packet with TTL 1:
  - Discard packet.
  - Send a "time exceeded" message back to the source.

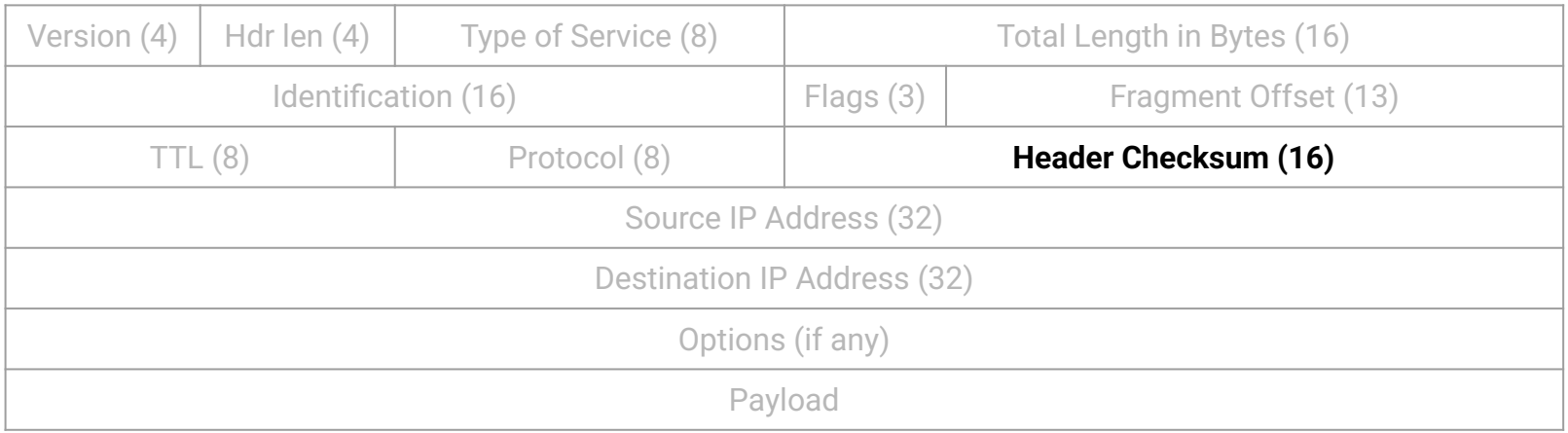


Task #5: Handle Errors: Checksum

Problem: The packet could get corrupted in transit.

Checksum: Small number of bits used to verify that data hasn't been corrupted.

If checksum is incorrect, router/destination discards packet.



Task #5: Handle Errors: Checksum

The checksum is only computed on the IP header, not the payload.

- IP header can't detect the payload being corrupted
- Why this design? End-to-end principle.
  The payload should be checked by the end hosts, not intermediate routers.

The checksum has to be updated at every router.

- Why? Because the TTL changes.
- Alternate design: Don't include the TTL in the checksum.
  Now, routers don't have to recompute checksum (less work for routers).

Task #5: Handle Errors: Fragmentation

Problem: The packet might be too large for a link.

- Every link has a **Maximum Transmission Unit (MTU)**:
  Largest number of bits the link can carry as one unit.
- If a packet size exceeds a link's MTU, the router must split the packet into multiple **fragments**.
- Then, the host has to reassemble the fragments to recover the original packet.

Task #5: Handle Errors: Fragmentation

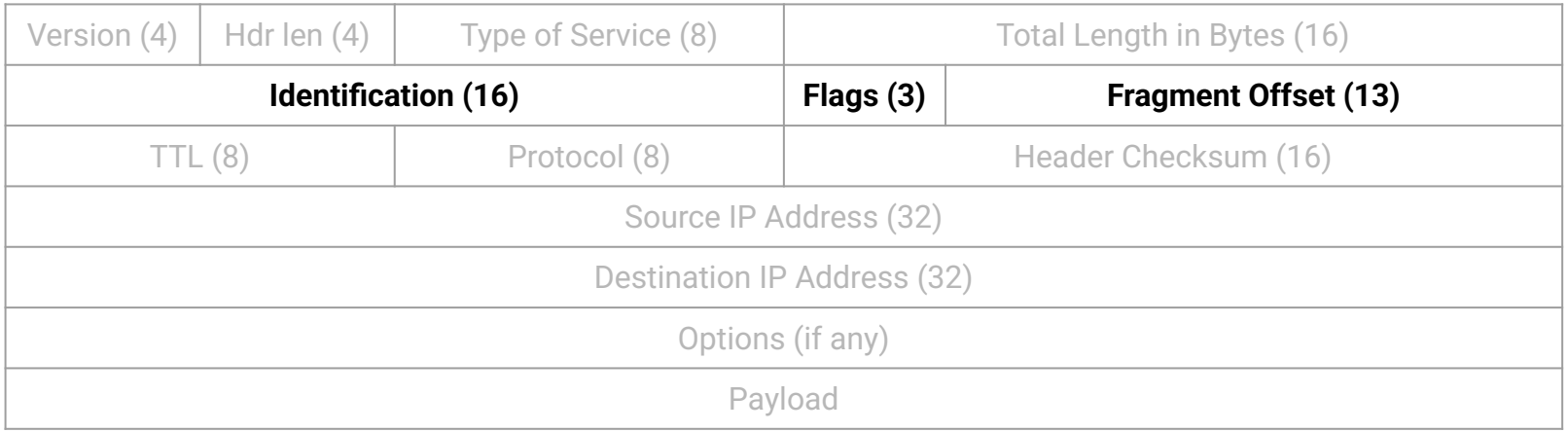
Problem: The packet might be too large for a link.

ID: Each fragment of the same packet has the same ID.

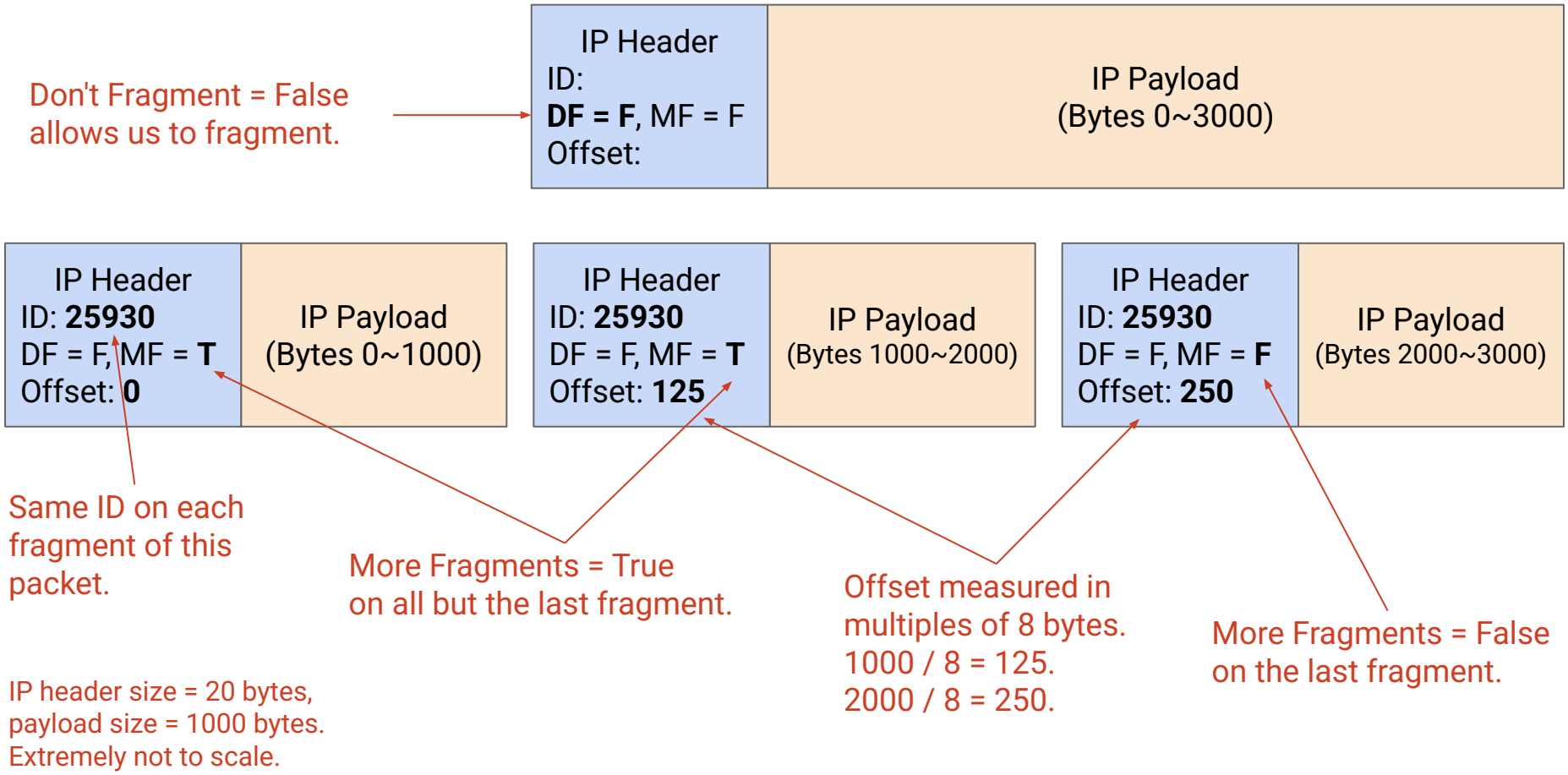
Flags (3 bits, 1 bit unused):

- DF: Don't Fragment. If this packet is too large, just drop it.
- MF: More Fragments. There's more data after this fragment.

Offset: Identifies which bytes of the original packet are in this fragment.



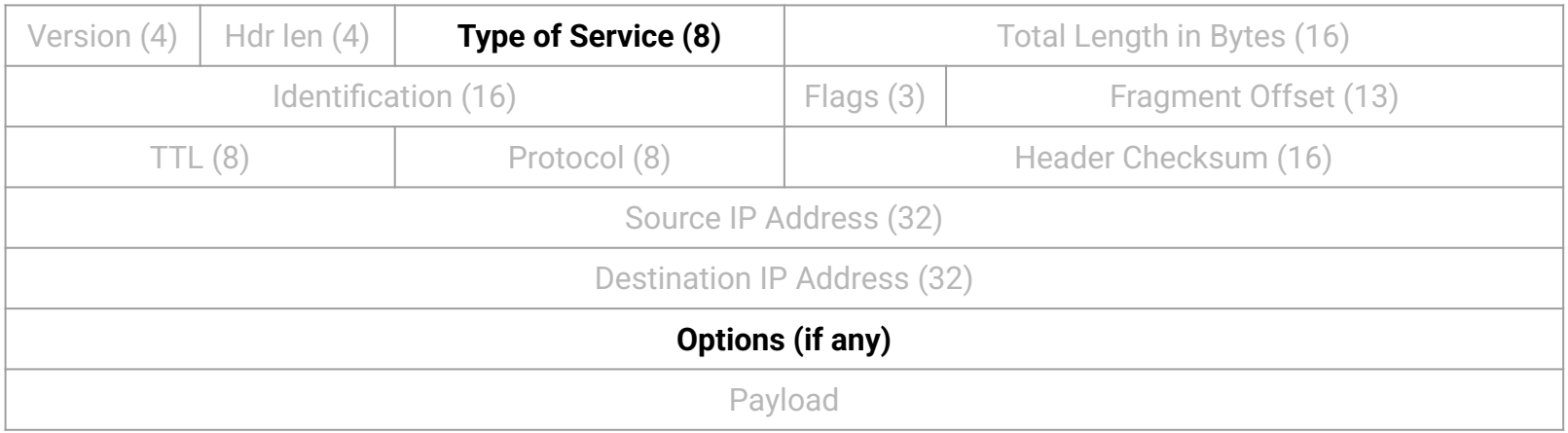
Task #5: Handle Errors: Fragmentation



Task #6: Specify Special Packet Handling

Type of Service: Treat packets differently depending on application/customer needs.

Options: Request advanced functionality for this packet.



Task #6: Specify Special Packet Handling

Type of Service: Treat packets differently depending on application/customer needs.

Original idea:

- Request different forms of delivery depending on these bits.
- Based on priority, delay, throughput, reliability, cost, etc.
- These bits were frequently re-defined and never fully deployed.
- Only the notion of priorities remained.

Modern usage:

- Differentiated Services Code Point (DSCP): Defines "classes" of traffic.
- Explicit Congestion Notification (ECN): Used in congestion control (coming soon).

Task #6: Specify Special Packet Handling

Options: Request advanced functionality for this packet.

Examples:

- Record Route: Tell me the path that this packet took (useful for diagnostics).
- Source Route: Send the packet along the route I provide.
- Timestamp: Record when each router processes this packet.

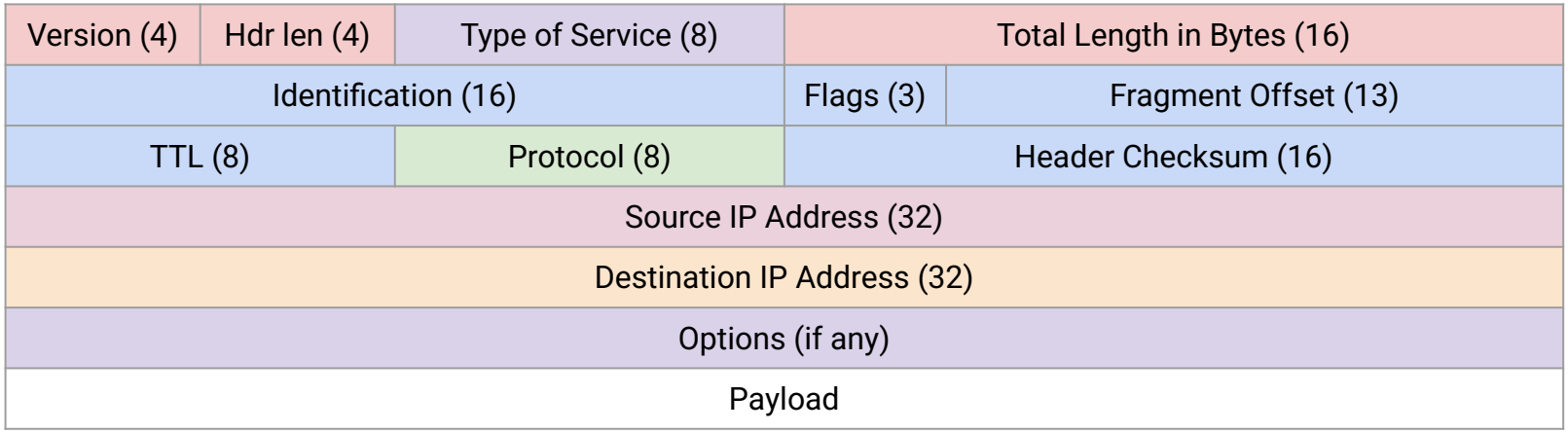
Options lead to more complex implementation.

- Options are variable-length, so routers have to check header length.
- Higher processing overhead for routers.

On the modern Internet, we avoid options when possible.

Introducing the IP Header

1. Parse the packet. (both router and destination)
2. Forward packet to the next hop. (router only)
3. Tell the destination what to do next. (destination only)
4. Send responses back to the source. (both router and destination)
5. Handle errors. (both router and destination)
6. Specify any special packet handling. (both router and destination)



IPv6 Changes

Lecture 10, CS 168, Spring 2025

BGP Implementation

- External BGP and Internal BGP
- Multiple Links Between ASes
- Message Types and Route Attributes
- Issues with BGP

IP Header

- IPv4 Header Fields
- **IPv6 Changes**
- Security

Why IPv6?

Main motivation: IP address exhaustion.

- IPv4 addresses are 32 bits long.
  There are $2^{32} \approx 4.2$ billion IPv4 addresses.
- We're running out of IPv4 addresses.
- IPv6 addresses are 128 bits long.
  2^{128} > number of atoms in the universe. We won't run out.

Took the opportunity to do some spring cleaning.

- Modernize (update or remove) outdated fields.
- IPv6 was originally supposed to be more ambitious, but additional features were never realized.
- The result is an elegant, if unambitious, protocol.

Cleanup #1: Eliminate Checksums

IPv6 eliminates checksums in the IP header.

- Benefit of checksum:
  - If a packet is corrupted but not detected, we keep sending it, which wastes bandwidth.
  - Checksum lets us drop the packet early.
- Why IPv6 made the change:
  - Today, bandwidth is less of a bottleneck.
  - It's okay if a few corrupt packets are sent through the network.
- Benefit of no checksum:
  - Less work for routers to do. Routers can process packets faster.

Cleanup #2: Eliminate Fragmentation

IPv6 eliminates fragmentation.

- If a packet is too large for a link:
  - Drop the packet.
  - Send an error back to the source with the maximum packet size (MTU).
  - It's the sender's job to split up data into smaller packets.
- Benefit of this change:
  - Less work for routers to do. Routers can process packets faster.

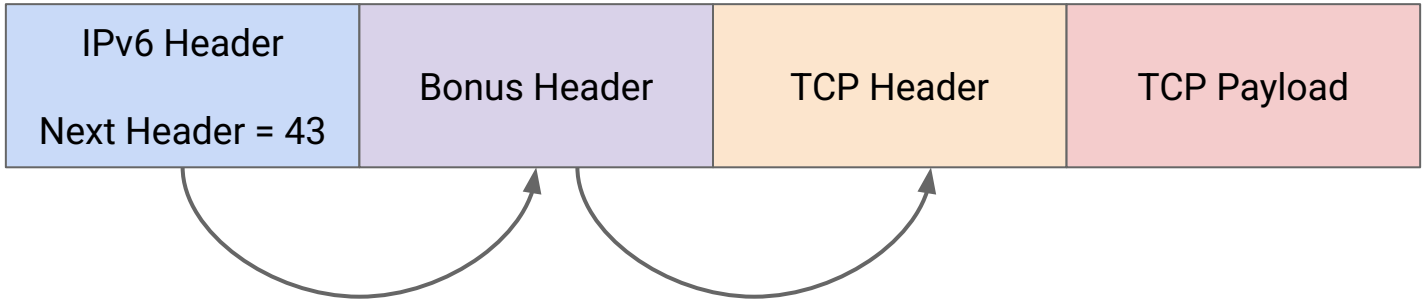
Cleanup #3: Eliminate Options

IPv6 eliminates options.

- IP header is now fixed size.
- Benefit: Less work for routers to do. Routers can process packets faster.

Replacement mechanism: Next Header.

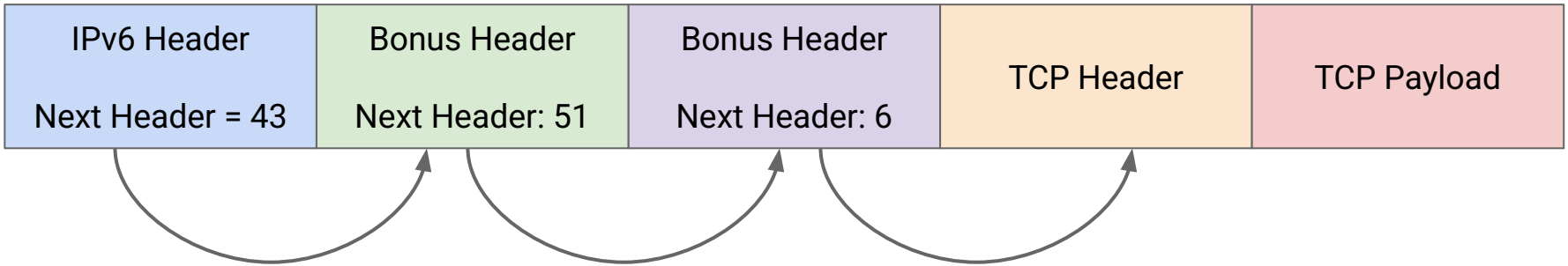
- Generalization of the IPv4 Protocol field.
- If you want extra processing, put the protocol ID in the Next Header field.
- IPv6 will pass the packet to the protocol for extra processing.
- When the extra processing is done, the protocol passes the packet to Layer 4.



Cleanup #3: Eliminate Options

Next Header can be generalized to multiple extra protocols in between Layer 3 and 4.

- IPv6 and the extra protocols all include a Next Header field to pass the packet to the appropriate next protocol.
- Eventually, someone's Next Header is a Layer 4 protocol (6 = TCP, 17 = UDP).
- Future-proofing: Elegant way to support new protocols as they're invented.



Cleanup #4: Flow Label

The **flow label** field lets us denote when packets are related to each other.

- Original Layer 3 design:
  - Every packet is sent independently.
  - Only higher layers need to think about grouping packets into flows.
- Why IPv6 made the change:
  - Modern routers often install **middlebox** systems to peek at IP packets, even if the router is not the destination.
  - Example: Firewall on the router peeks at packets to block malicious traffic.
  - Middleboxes might want to consider packets in a flow together.
  - Flow label gives us a way to tell middleboxes when packets are related.

IPv4 → IPv6 Header

Renamed and moved:

- Type of Service → Traffic Class
- Total Length → Payload Length
- TTL → Hop Limit

1. Eliminate checksums.
2. Eliminate fragmentation.
3. Eliminate options, add next header.
4. Add flow label.

Version	~~Hdr len~~	Type of Service	Total Length in Bytes
~~Identification~~		~~Flags~~	~~Fragment Offset~~
TTL	Protocol	~~Header Checksum~~	
Source IP Address (32 bits)			
Destination IP Address (32 bits)			
Options (if any)			
Payload			

Vers	Traffic Class	Flow Label	
Payload Length		Next Header	Hop Limit
Source IP Address (128 bits)			
Destination Address (128 bits)			

Philosophy of IPv6 Changes

Leave problems to end hosts.

- Eliminated fragmentation.
- Eliminated checksum.
- Could we eliminate TTL and leave that to end hosts?
  - No, avoiding loops is fundamentally a router problem.

Simplify: Less work for routers.

- Eliminated options.
- Fixed-size header length.

Future-proof: Allow for extensibility.

- Generalized next-header approach.
- Generalized flow label for packets.

IP Security

Lecture 10, CS 168, Spring 2025

BGP Implementation

- External BGP and Internal BGP
- Multiple Links Between ASes
- Message Types and Route Attributes
- Issues with BGP

IP Header

- IPv4 Header Fields
- IPv6 Changes
- **Security**

Exploiting Source IP: Spoofing

Spoofing: Lie about the source address.

- Attacker pretends to be somebody else and sends the packet.

Using spoofing to attack a destination:

- Denial-of-service (DoS) attack: Overwhelm a service by flooding it with packets.
- Without spoofing: The service can block your source address.
- Spoofing makes DoS more effective: Server doesn't know which packets to block.

Using spoofing to attack the impersonated user:

- Attacker pretends to be Bob and sends a virus.
- Now Bob is wrongly blamed for sending the virus.
- Return traffic (e.g. angry messages) goes to Bob instead.

Exploiting Type of Service

Recall type of service (ToS): Bits indicating priority of the packet.

Attack: Prioritize your own traffic.

- If anybody can set ToS bits, attackers can make their own packets high-priority.
- Network prefers attacker traffic.
- Today, ToS bits are mostly set/used by operators, not end hosts.

Attack: Use spoofing to make someone else pay for priority traffic.

- Suppose the network charges for high-priority traffic.
- Attacker sends a spoofed packet: "I am Bob and this packet is high-priority."
- Network makes Bob pay for that packet.

Exploiting Fragmentation and Options

Recall: Fragmentation and options are more work for the router.

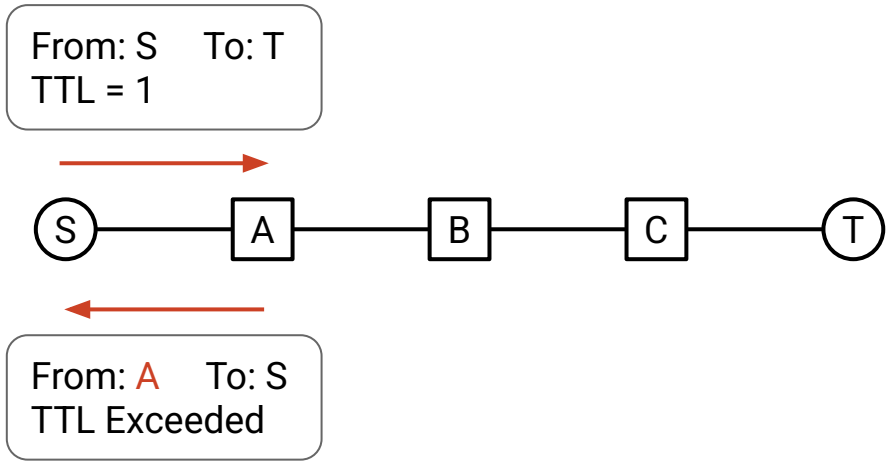
Denial-of-service (DoS) attack: Overwhelm routers by making them do more work.

- Fragmentation: Send large packets on purpose.
- Options: Send packets with complicated options on purpose.
  - Defense: Routers often ignore options, or drop packets with options.

Exploiting TTL: Traceroute

Traceroute: Exploiting the TTL to discover the path a packet travels along.

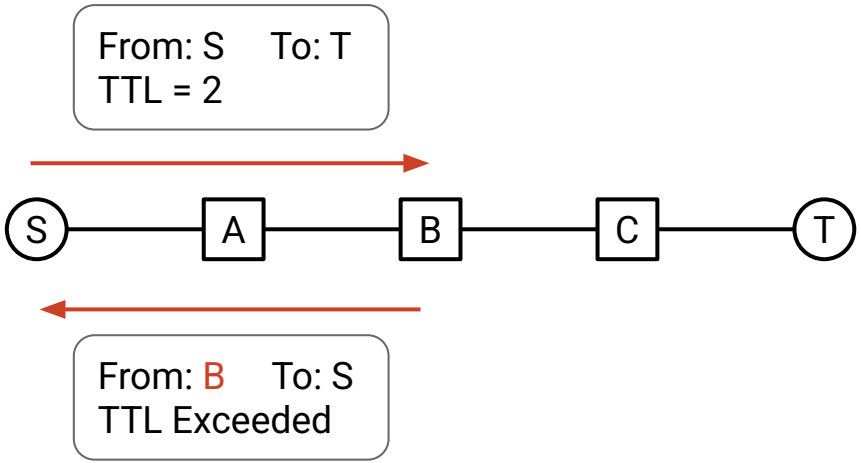
- You'll do this in Project 1.
- Key idea: When TTL expires, the router sends back a "TTL Exceeded" error. This allows you to learn the identity one of the routers on the path.
- Basic approach: Send packets with TTL=1, TTL=2, TTL=3, etc.
- Not all routers respond with the "TTL Exceeded" message.



Exploiting TTL: Traceroute

Traceroute: Exploiting the TTL to discover the path a packet travels along.

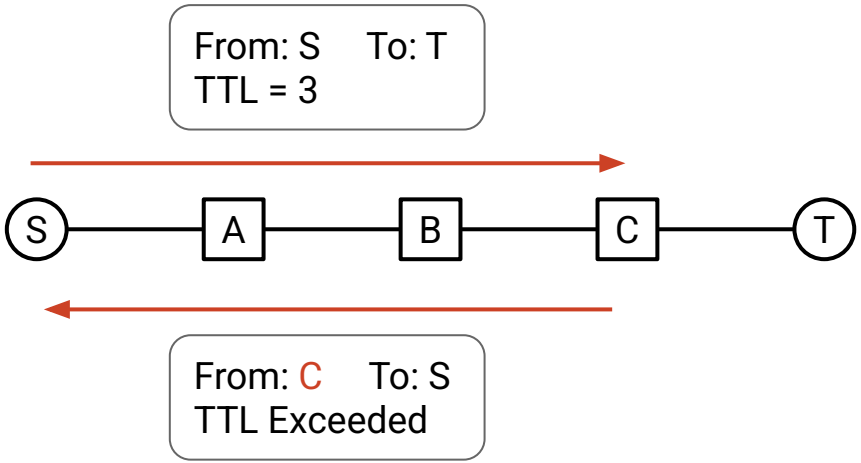
- You'll do this in Project 1.
- Key idea: When TTL expires, the router sends back a "TTL Exceeded" error. This allows you to learn the identity one of the routers on the path.
- Basic approach: Send packets with TTL=1, TTL=2, TTL=3, etc.
- Not all routers respond with the "TTL Exceeded" message.



Exploiting TTL: Traceroute

Traceroute: Exploiting the TTL to discover the path a packet travels along.

- You'll do this in Project 1.
- Key idea: When TTL expires, the router sends back a "TTL Exceeded" error. This allows you to learn the identity one of the routers on the path.
- Basic approach: Send packets with TTL=1, TTL=2, TTL=3, etc.
- Not all routers respond with the "TTL Exceeded" message.



Exploiting Other Fields?

Exploiting protocol field?

- If protocol is set incorrectly, the next layer will find the packet malformed.

Exploiting checksum?

- A bad checksum causes the network to drop the packet.

These are not effective attacks.

Summary of IP Attacks

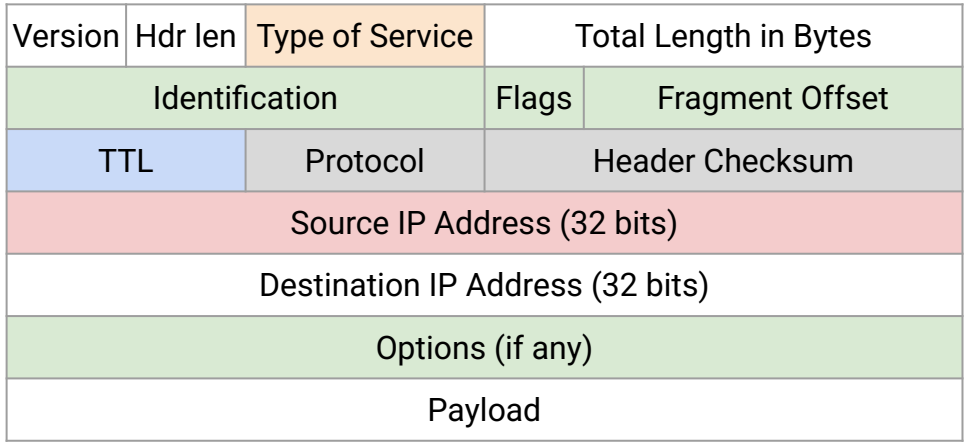
Source IP address: Spoofing.

Type of service: Prioritize attacker traffic.

Fragmentation, Options: Denial-of-service.

TTL: Traceroute.

Protocol, Checksum: No apparent problems.



Recap: IP Header Design

Header design is more nuanced than it first seems!

We have to juggle multiple goals:

- Efficient implementation
- Security
- Future needs