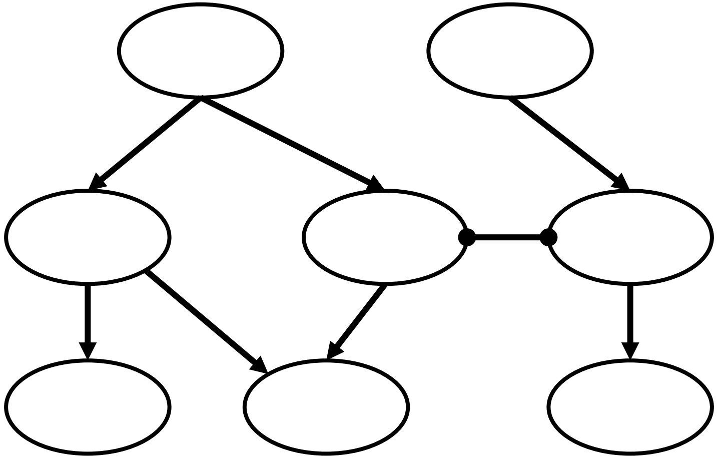


Lecture 8 (Routing 5)

Inter-Domain Routing

Slides credit: CS 168 @ UC Berkeley

Autonomous Systems

Lecture 8, Spring 2026

Autonomous Systems

- **What are ASes?**
- Business Relationships

Goals of Inter-Domain Routing

- Policy-Based Routing
- Gao-Rexford Rules

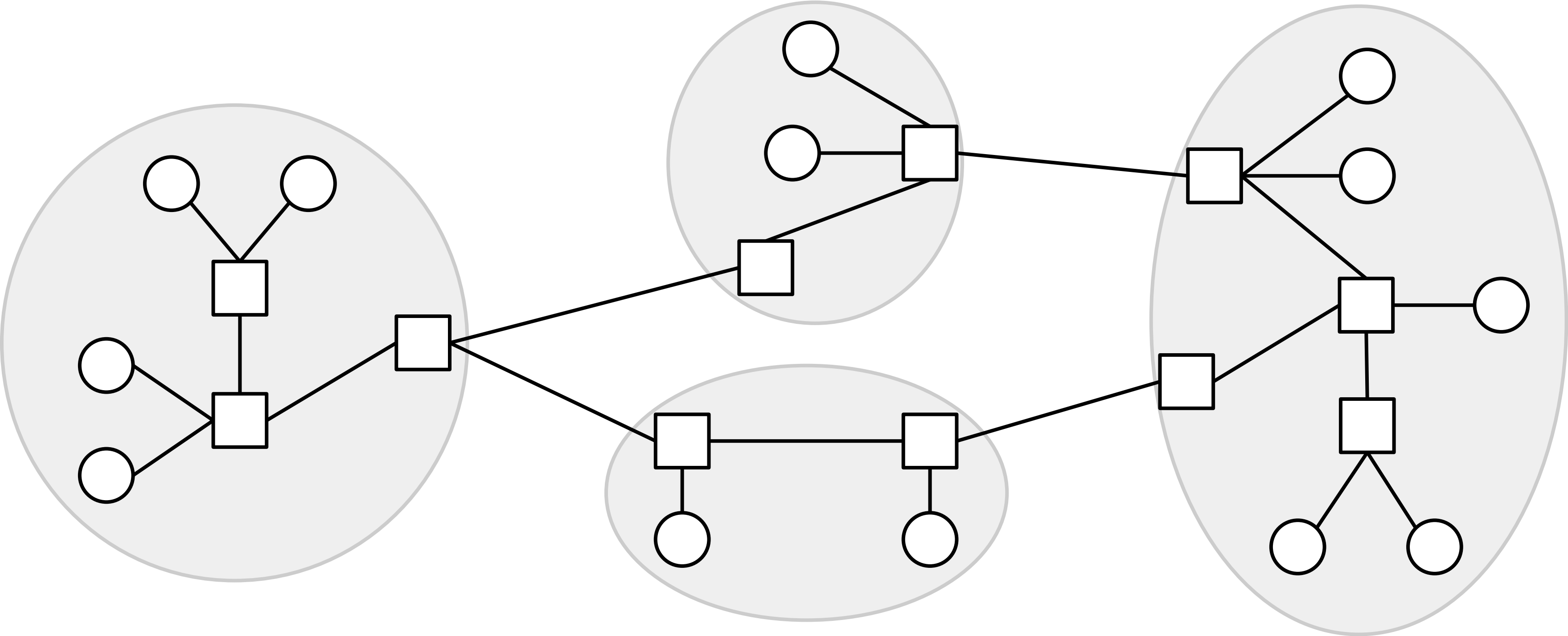
BGP (Border Gateway Protocol)

- Importing and Exporting
- Aggregation and Path-Vector

Intra-Domain vs. Inter-Domain Routing

Recall: The Internet is a network of networks.

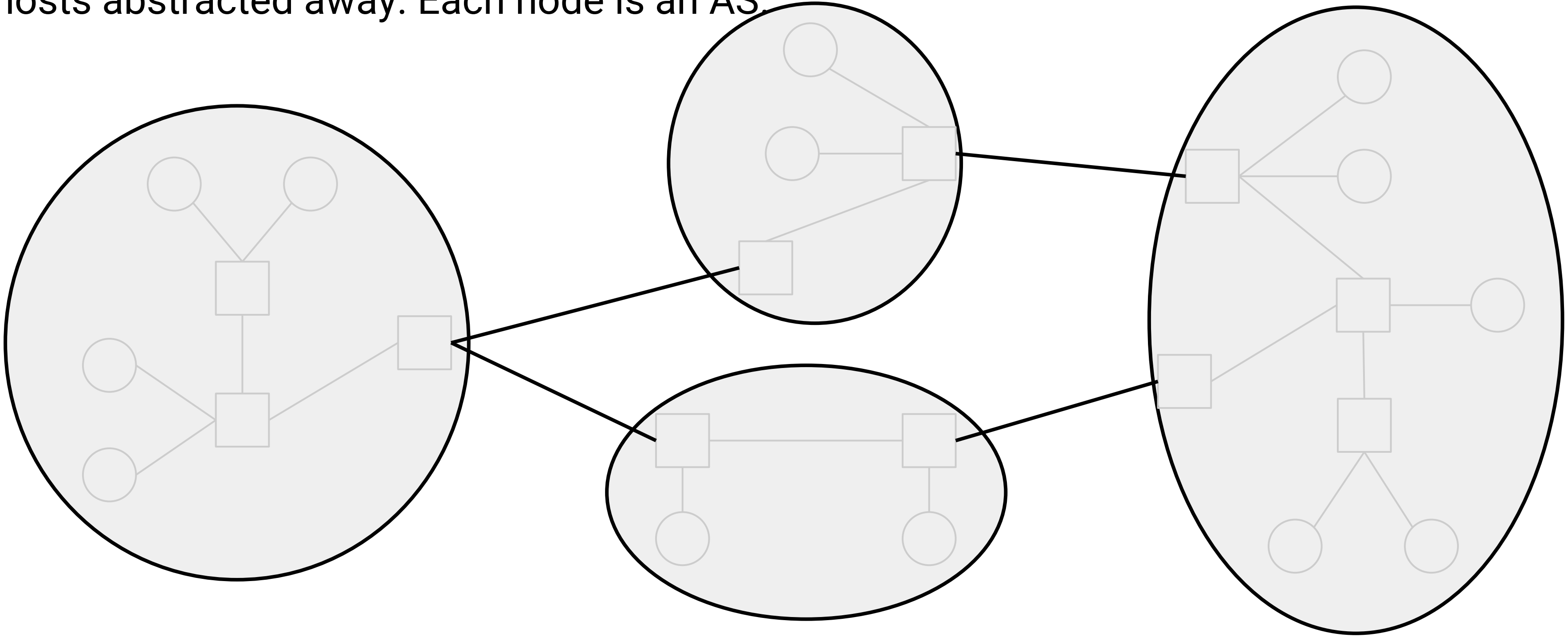
We used intra-domain routing to compute routes inside each network.



Intra-Domain vs. Inter-Domain Routing

Autonomous System (AS): One or more local networks, all under a single administrative control. Sometimes informally called "domains. Each AS is assigned a unique AS number (ASN).

Inter-domain topology (or **AS graph**): A graph of ASes, with the individual routers and hosts abstracted away. Each node is an AS.

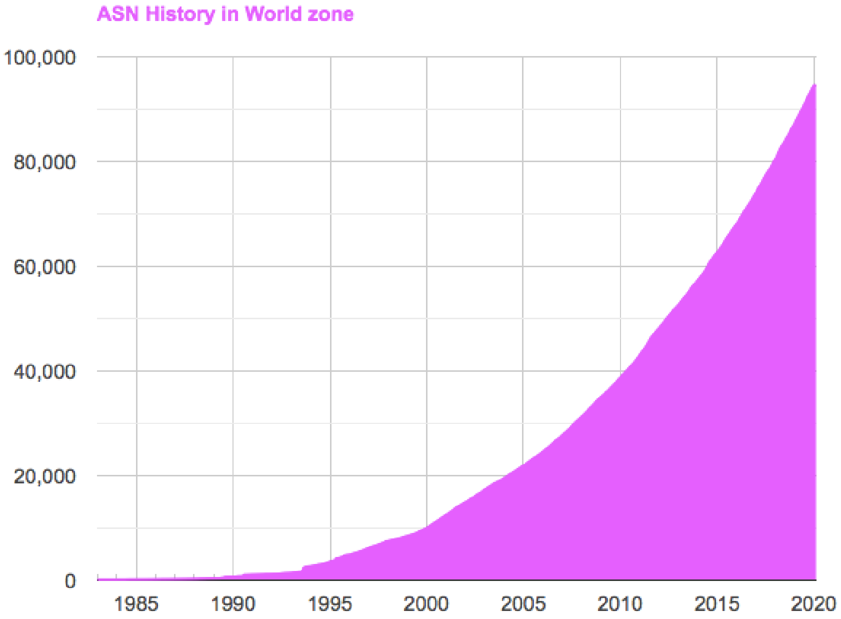


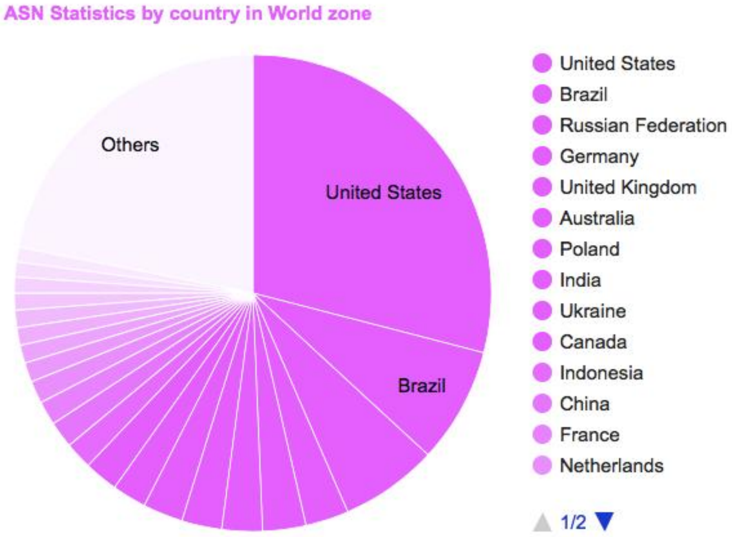
Brief History of Autonomous Systems

New ASNs are assigned by Internet Assigned Numbers Authority (IANA).

Number of ASes increases over time: Over 90,000 today!

ASes by country: USA has the most, followed by Brazil.





Types of Autonomous Systems

Stub AS: Only sends/receives packets on behalf of hosts inside the AS.

- Similar to end hosts in intra-domain model.
- Does not forward packets between other ASes.
- Examples: Hofstra Univ, local bank.
- Most ASes are this type.

Transit AS: Forwards packets on behalf of other ASes.

- Similar to routers in intra-domain model.
- Can still send/receive packets for hosts inside the AS.
- Examples: AT&T, Verizon.
- Can vary in scale (global, regional).

Note: Some modern ASes (e.g. Google, Amazon) blur the line between stub and transit.

Business Relationships

Lecture 8, Spring 2026

Autonomous Systems

- What are ASes?
- **Business Relationships**

Goals of Inter-Domain Routing

- Policy-Based Routing
- Gao-Rexford Rules

BGP (Border Gateway Protocol)

- Importing and Exporting
- Aggregation and Path-Vector

Business Relationships

Inter-domain topology is shaped between business relationships between ASes.

Two different ways for a pair of ASes to be related:

- **Customer-provider** relationship:
  - The customer pays the provider.
  - In exchange, the provider offers to forward traffic to/from the customer.
- **Peering** relationship:
  - Peers don't pay each other.
  - Peers exchange roughly equal traffic.

AS Graph with Business Relationships

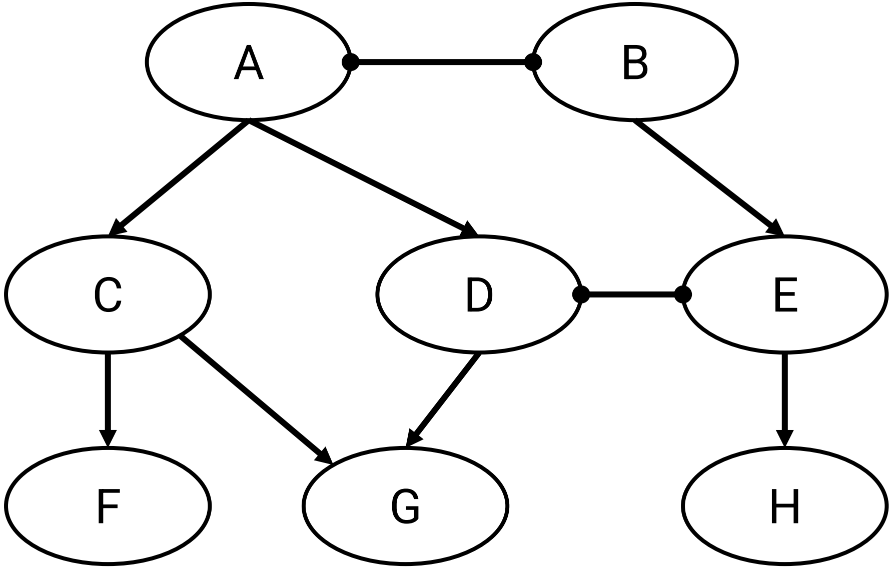
Representing business relationships in the AS graph: ⟶

- Provider ●—● Customer
- Peer Peer

The arrows do not represent direction of packets (e.g. F can send packets to C).

The AS graph forms a hierarchy (all arrows point down).

- Service flows down: Higher nodes provide service to lower nodes.
- Money flows up: Lower nodes pay money to higher nodes.

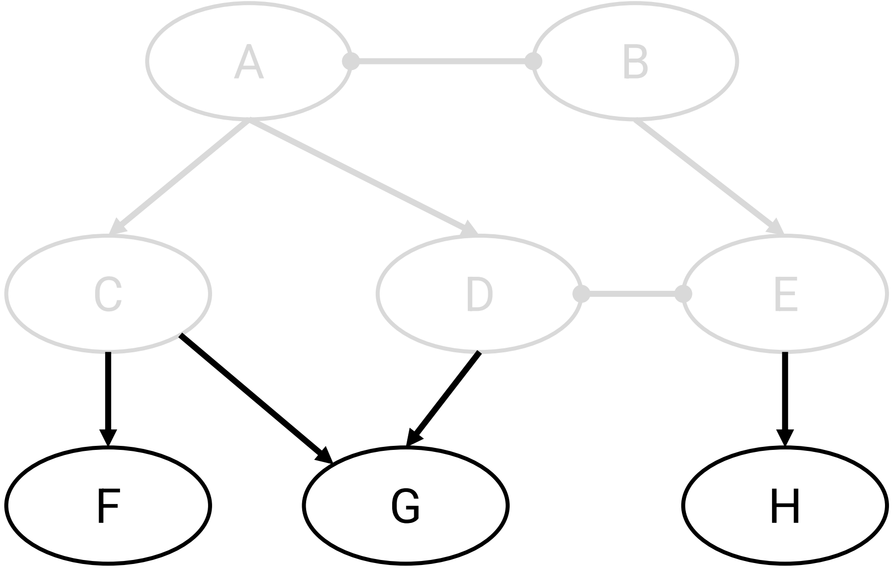


AS Graph with Business Relationships

Stub ASes in this graph: F, G, H.

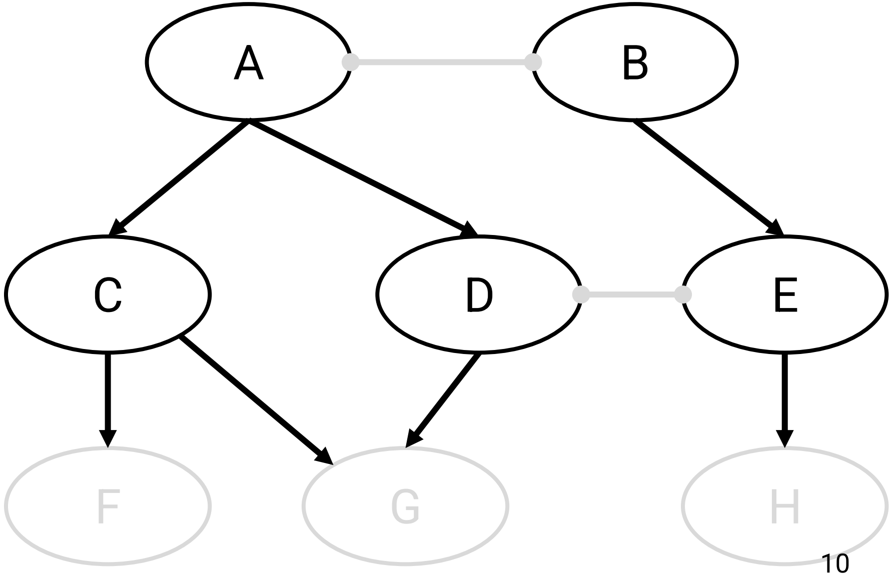
No outgoing edges: Not providing service to anybody.

Incoming edge(s) shows who they're buying service from.



Transit ASes in this graph: A, B, C, D, E.

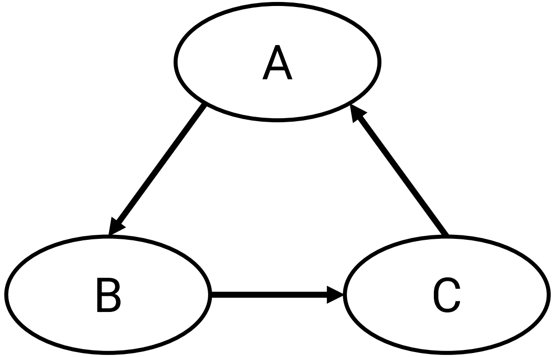
Outgoing edges indicate they're providing service to somebody.



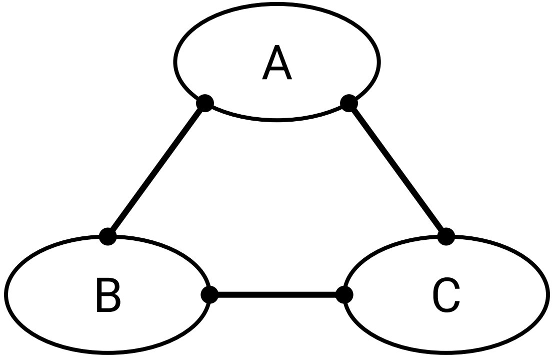
AS Graph is Acyclic

The AS graph has no cycles.

- A cycle means you're paying yourself, which doesn't make sense.
- A cycle of peering relationships is okay.



Invalid



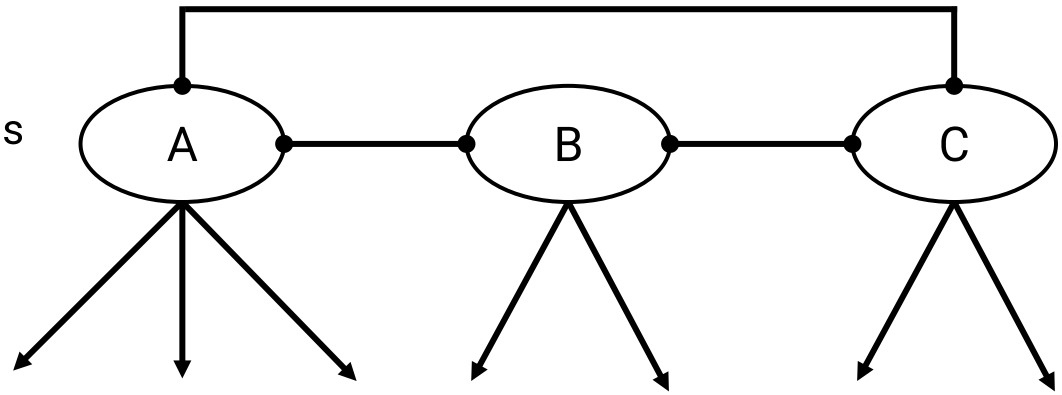
Valid

Tier 1 ASes

Tier 1 ASes: ASes at the top of the hierarchy.

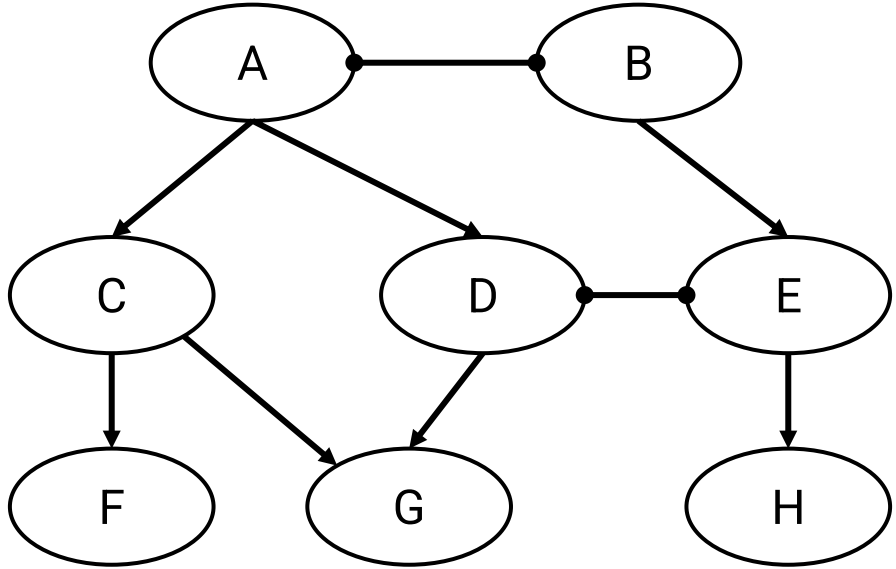
A Tier 1 AS:

- Has no providers (no incoming edges).
- Has a peering relationship with every other Tier 1 AS.
  - This helps ensure the AS graph is connected (no disconnected components).
- ~20 Tier 1 ASes in real life.
  - USA: AT&T, Verizon
  - Europe: Telecom Italia, France Telecom
  - Asia: NTT Communications (Japan)
- These companies usually own infrastructure spanning the whole world.



Properties of AS Graphs

- Every non-Tier 1 AS has at least one provider (incoming edge).
- Starting at any AS and walking up the graph will eventually lead to a Tier 1 AS.



Goals of Inter-Domain Routing: Policy-Based Routing

Lecture 8, Spring 2026

Autonomous Systems

- What are ASes?
- Business Relationships

Goals of Inter-Domain Routing

- **Policy-Based Routing**
- Gao-Rexford Rules

BGP (Border Gateway Protocol)

- Importing and Exporting
- Aggregation and Path-Vector

Goals of Inter-Domain Routing

Scalability: Routing must scale to the entire Internet.

- Solution: Hierarchical IP addressing.

Privacy: ASes don't want to explicitly announce sensitive information.

- I shouldn't have to tell everybody who my provider is.
- Companies might not want to reveal information to rivals.

Autonomy: ASes want the freedom to choose their own policies.

- Policy is usually based on business goals.

Policy-Based Routing

Autonomy: ASes want the freedom to choose their own **policies**.

Recall our routing goals:

- Valid paths:
  - Intra-domain definition: No loops, no dead ends.
  - Inter-domain definition: Same.
- Good paths:
  - Intra-domain definition: Least-cost paths.
  - Inter-domain definitions: Paths that respect every AS's policies.

Policy-Based Routing

Autonomy: ASes want the freedom to choose their own **policies**.

Examples of policies:

- Define how I will handle traffic from others:
    - I don't want to carry AS#2046's traffic through my network.
- Defines how others should handle my traffic:
    - I prefer if my traffic was carried by AS#10 instead of AS#4.
    - Don't send my traffic through AS#54 unless absolutely necessary.

We have to find paths that respect every AS's policies.

Gao-Rexford Rules

Lecture 8, Spring 2026

Autonomous Systems

- What are ASes?
- Business Relationships

Goals of Inter-Domain Routing

- Policy-Based Routing
- **Gao-Rexford Rules**

BGP (Border Gateway Protocol)

- Importing and Exporting
- Aggregation and Path-Vector

Gao-Rexford Rules: Choosing Routes

Most ASes follow standard conventions: **Gao-Rexford Rules**.

These conventions reflect real-world business practices:

- Making money is good.
- Don't do work for free.

Distance-vector protocol: Prefer the *shortest* path.

Gao-Rexford rules: Prefer the *most profitable* path.

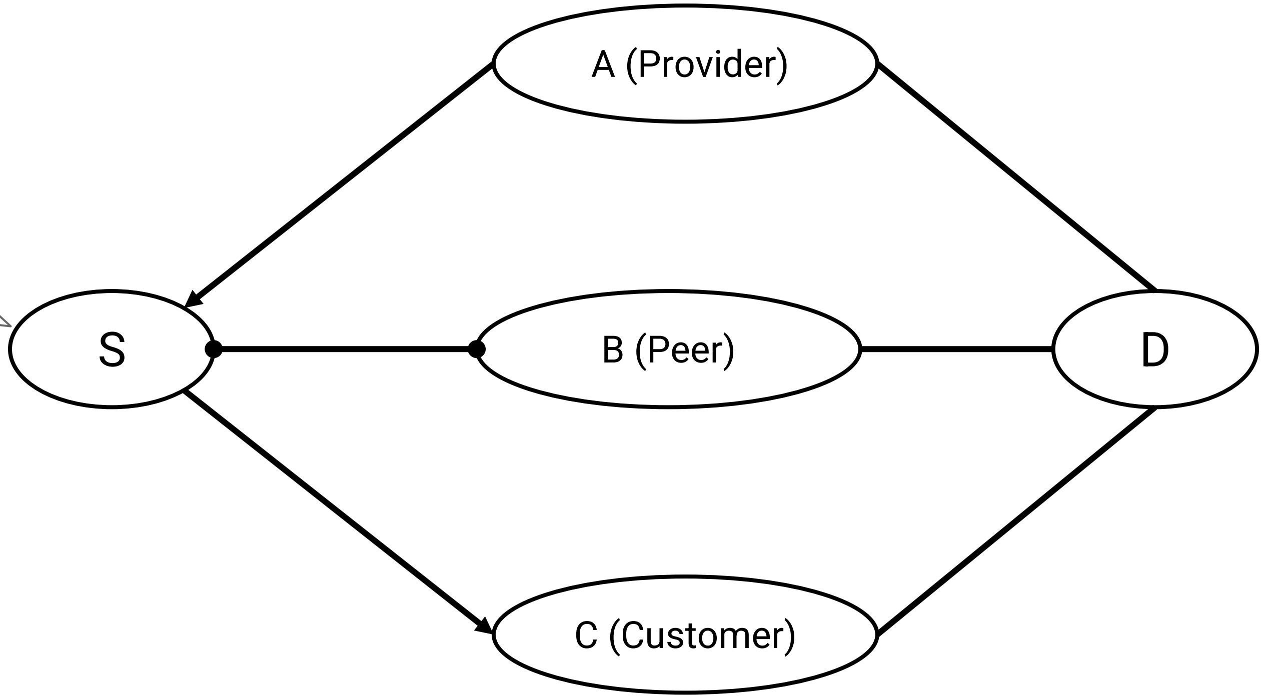
- Best: Path where the next hop is a customer. (They pay me.)
- Less good: Path where the next hop is a peer. (I don't make money.)
- Worst: Path where the next hop is a provider. (I have to pay.)

Reflects real-world business practice: Making money is good.

Gao-Rexford Rules: Choosing Routes

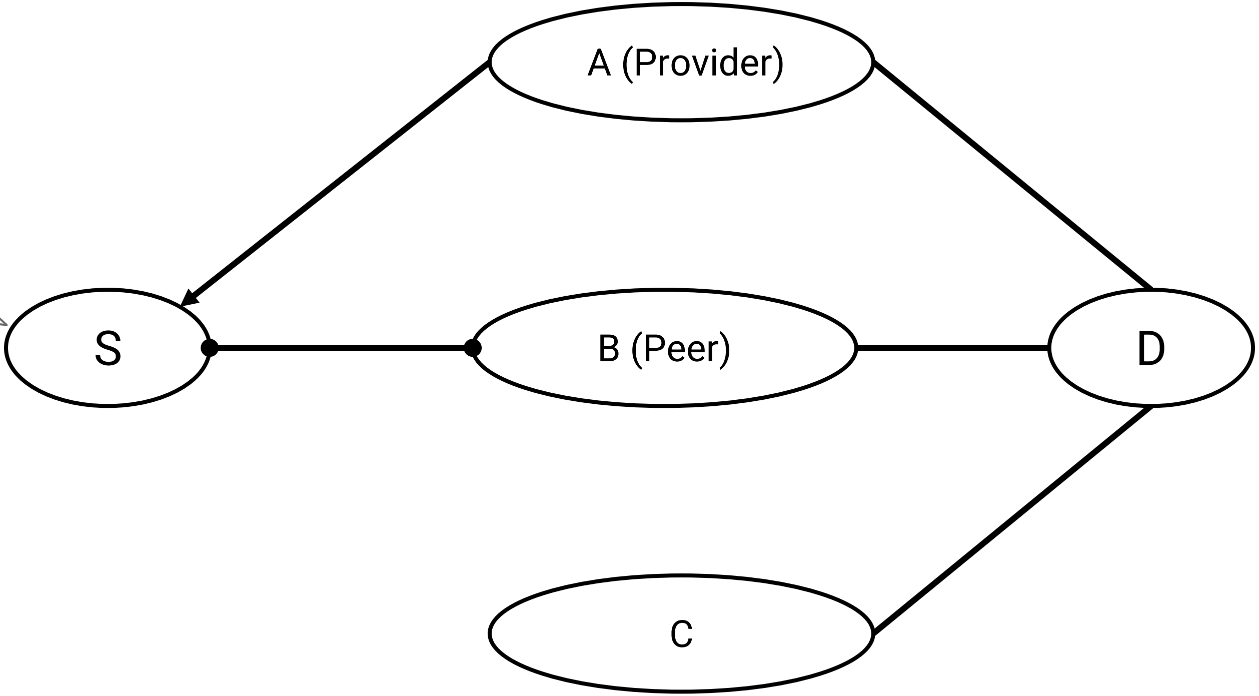
Gao-Rexford rules: If I'm offered multiple paths, pick the *most profitable* one.

I prefer the path via C, because C will pay me to forward packets.



Gao-Rexford Rules: Choosing Routes

Gao-Rexford rules: If I'm offered multiple paths, pick the *most profitable* one.



I prefer the path via B. I don't make money, but at least I don't have to pay.

Gao-Rexford Rules: Participating in Routes

Distance-vector protocol: I am okay with participating in any route.

- "Participating": Being one of the routers forwarding packets along a path.

Gao-Rexford rules: I will only participate in routes where I get paid.

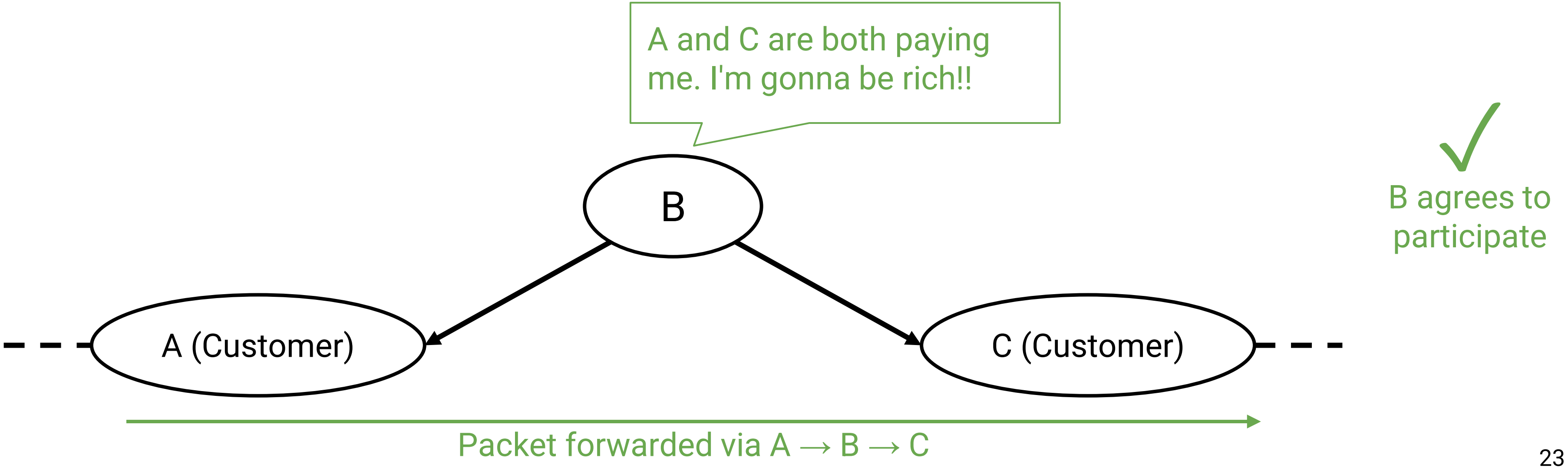
- Reflects real-world business relationship: Don't do work for free.

How to check if I get paid:

- Look at my two neighbors along the path.
- I get paid if and only if one of my neighbors is a customer.

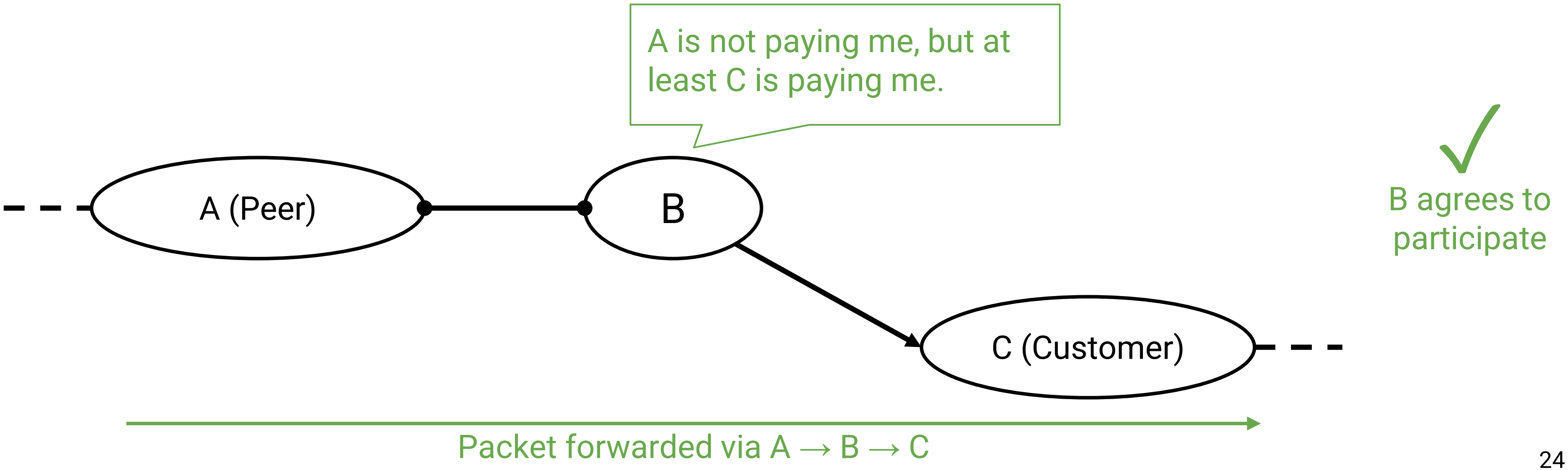
Gao-Rexford Rules: Participating in Routes

I get paid and participate if and only if at least one of my neighbors is a customer.


A and C are both paying me. I'm gonna be rich!!
B
A (Customer)
C (Customer)
✓
B agrees to participate
Packet forwarded via A → B → C

Gao-Rexford Rules: Participating in Routes

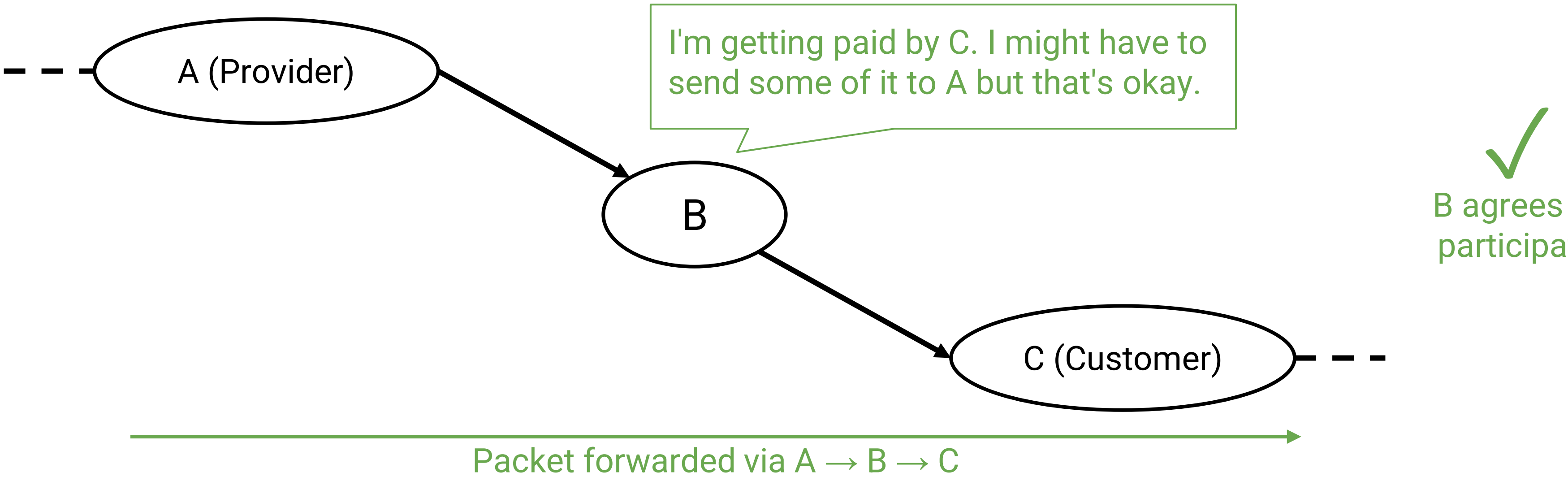
I get paid and participate if and only if at least one of my neighbors is a customer.


A is not paying me, but at least C is paying me.
A (Peer)
B
C (Customer)
✓
B agrees to participate
Packet forwarded via A → B → C

Gao-Rexford Rules: Participating in Routes

I get paid and participate if and only if at least one of my neighbors is a customer.

- In this scenario: I get paid by C, but I have to pay A.
- I should still participate because this is how I offer service to my customers.



A (Provider)

B

C (Customer)

I'm getting paid by C. I might have to send some of it to A but that's okay.

✓

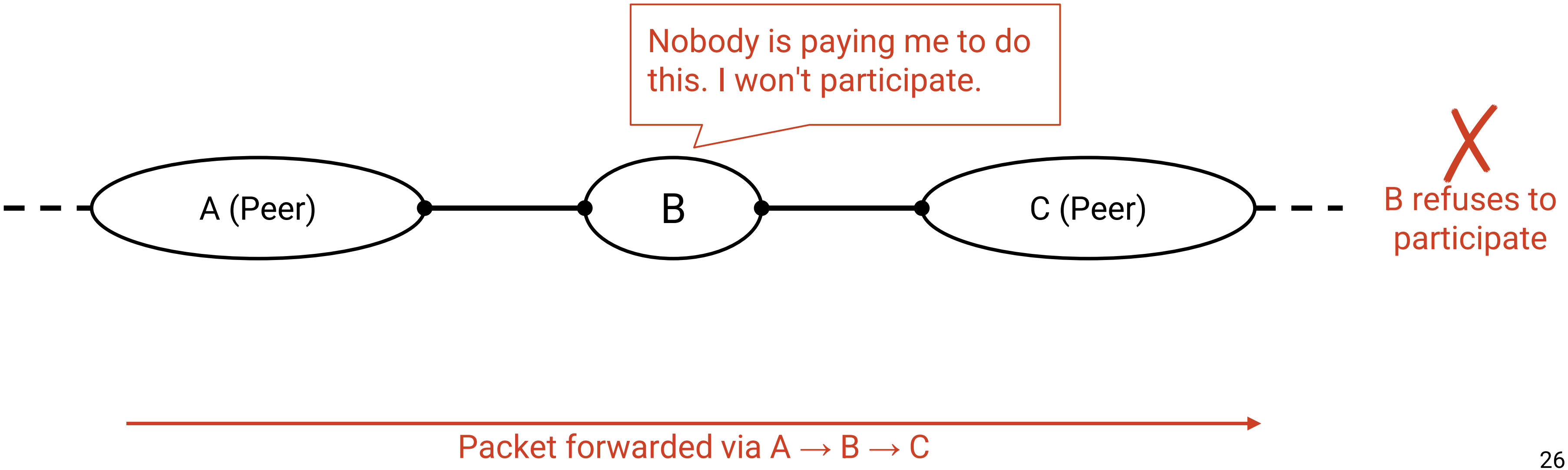
B agrees to participate

Packet forwarded via A → B → C

Gao-Rexford Rules: Participating in Routes

I get paid and participate if and only if at least one of my neighbors is a customer.

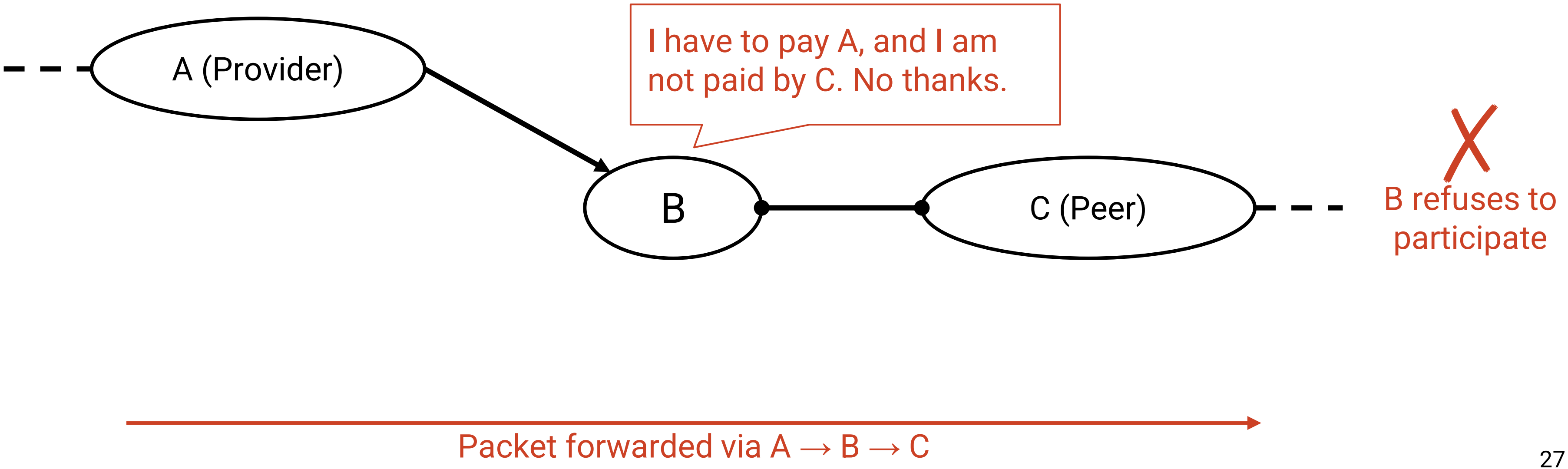
- Peers do not provide transit between other peers.


Nobody is paying me to do this. I won't participate.
A (Peer)
B
C (Peer)
✗
B refuses to participate
Packet forwarded via A → B → C

Gao-Rexford Rules: Participating in Routes

I get paid and participate if and only if at least one of my neighbors is a customer.

- If one of my neighbors is a peer, the other neighbor must be a customer.
- An AS only carries traffic to/from its own customers over a peering link.



I have to pay A, and I am not paid by C. No thanks.

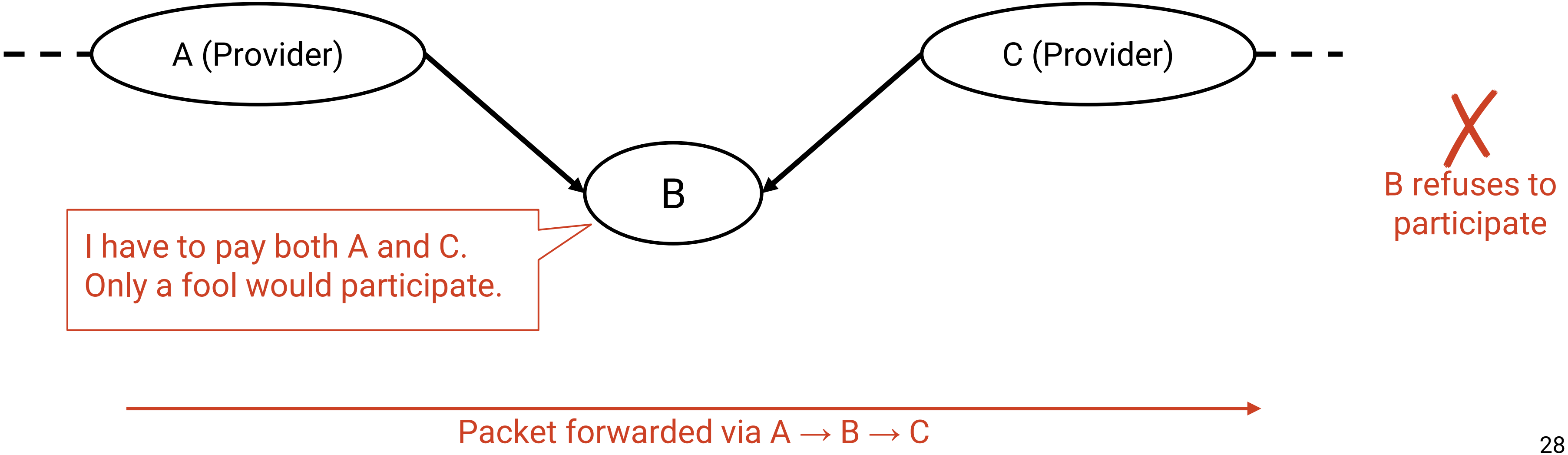
B refuses to participate

Packet forwarded via A → B → C

Gao-Rexford Rules: Participating in Routes

I get paid and participate if and only if at least one of my neighbors is a customer.

- Routes are **valley-free**.
- Travelling downhill, and turning around and going back uphill is not allowed.



B refuses to participate

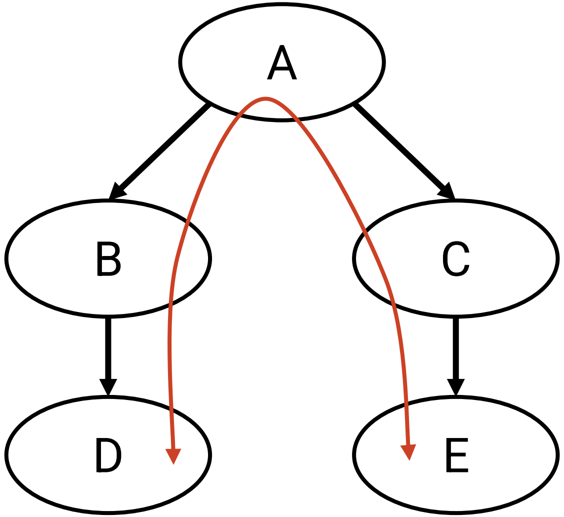
I have to pay both A and C.
Only a fool would participate.

Packet forwarded via A → B → C

Policy-Based Routing Example

D and E want to exchange traffic along this path.

Will the transit ASes (A, B, C) agree to participate?

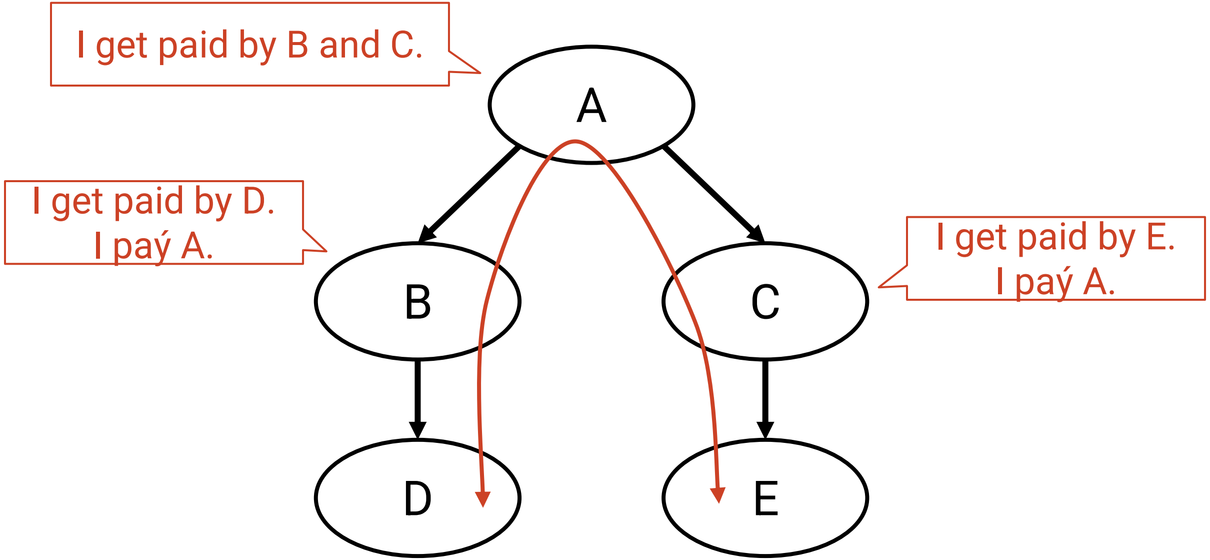


Policy-Based Routing Example

D and E want to exchange traffic along this path.

Will the transit ASes (A, B, C) agree to participate?

Yes, each transit AS has an adjacent customer.



I get paid by B and C.

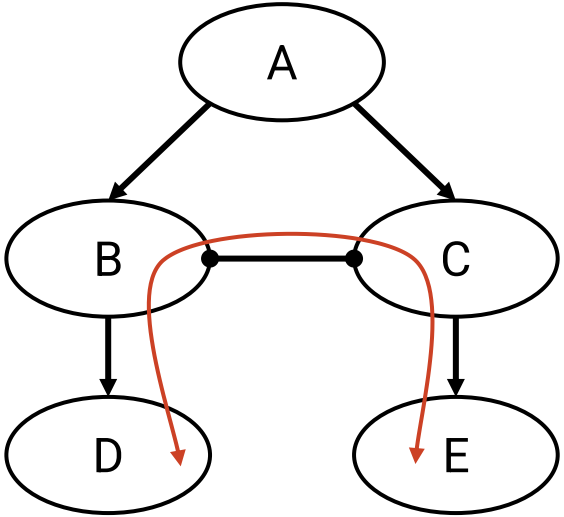
I get paid by D.
I paý A.

I get paid by E.
I paý A.

Policy-Based Routing Example

Suppose we add a new link (B and C establish a peering relationship).

Will the transit ASes (B, C) agree to participate in this new route?

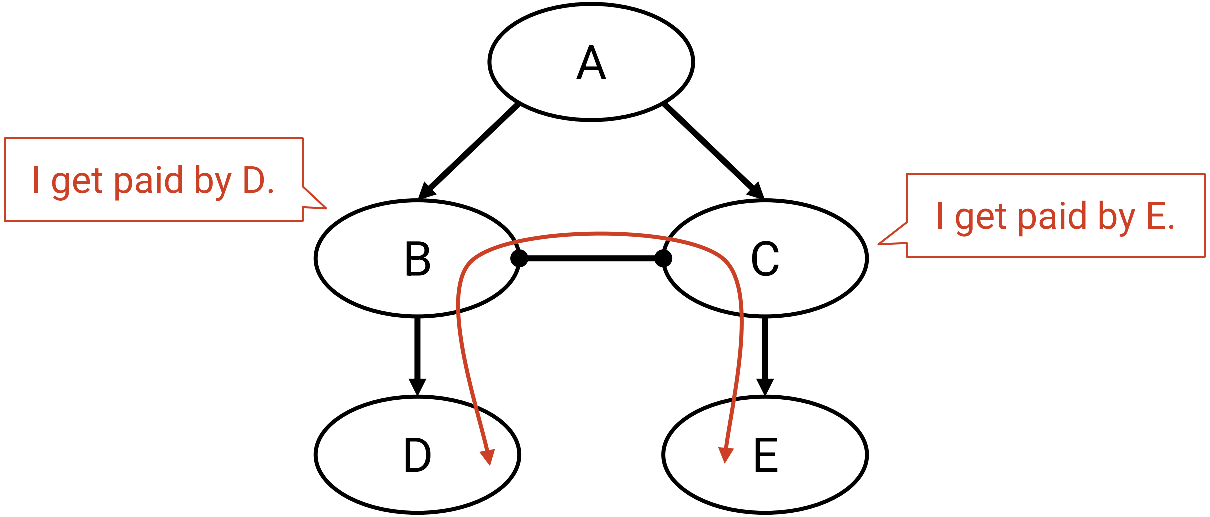


Policy-Based Routing Example

Suppose we add a new link (B and C establish a peering relationship).

Will the transit ASes (B, C) agree to participate in this new route?

Yes, each transit AS has an adjacent customer.



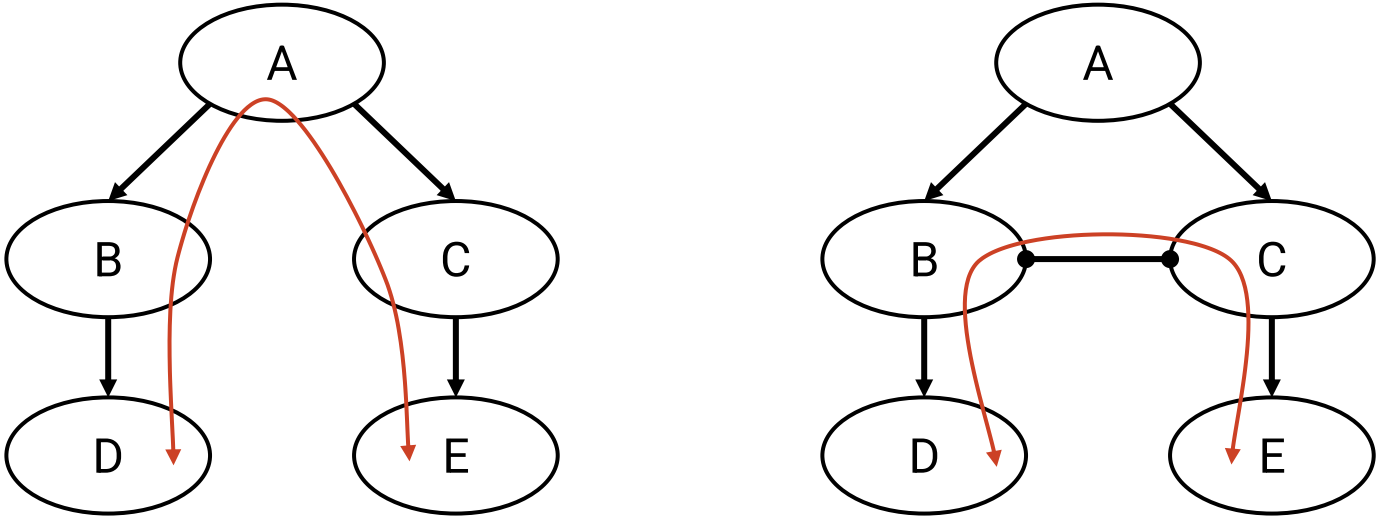
I get paid by D.

I get paid by E.

Policy-Based Routing Example

With the new link, B and C save money (they don't have to pay A anymore).

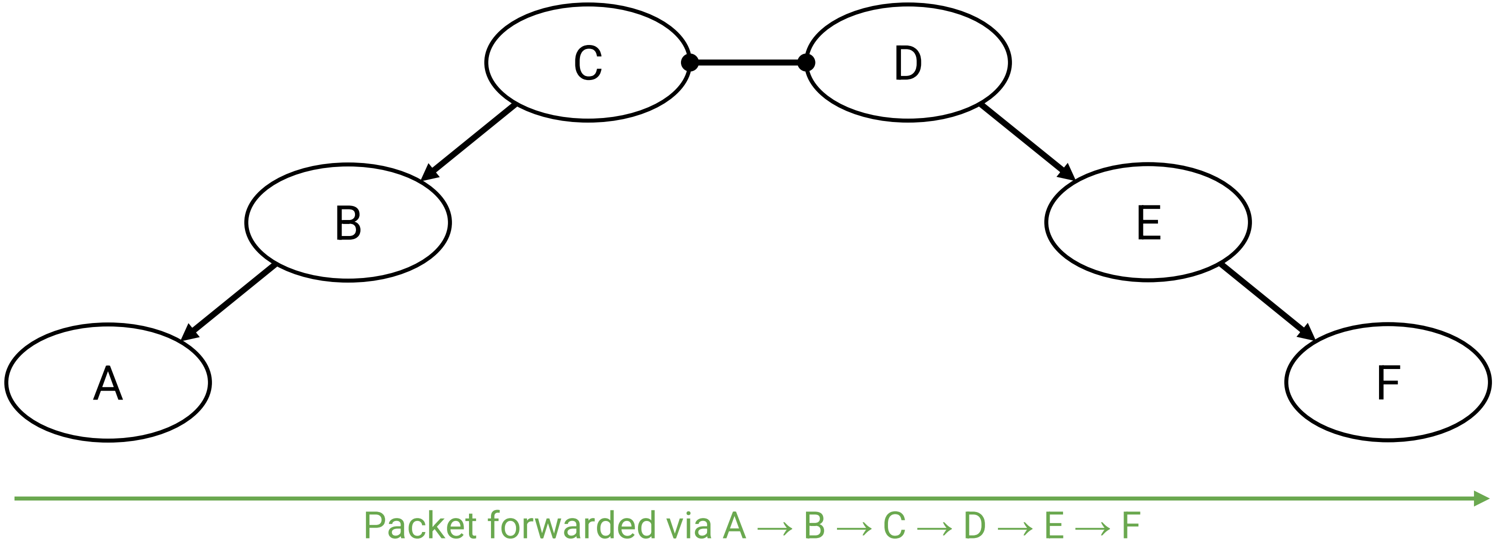
Trade-off: B and C have to pay to build that link.



Properties of Routes with Gao-Rexford Rules

Routes are **single-peaked**.

- We first climb 0 or more uphill links.
- Then, we reach a peak, and traverse 0 or 1 peering links.
- Then, we go strictly downhill all the way to the destination.



Packet forwarded via A → B → C → D → E → F

Proving Reachability and Convergence with Gao-Rexford

If all of these are true:

- Starting from any AS and moving up the hierarchy will lead to a Tier 1 AS.
- The AS graph has no cycles.
- All ASes follow the Gao-Rexford rules.

Then we can guarantee these two properties:

- **Reachability**: Any two ASes in the graph can communicate.
- **Convergence**: All ASes agree on paths.

These properties hold in steady-state.

- Steady-state: Network topology and policies remain unchanged.
- If something changes, it might take some time for ASes to reconverge.

If ASes pick arbitrary policies (not following Gao-Rexford), we can't guarantee these properties.

Summary: Autonomous Systems, Policy-Based Routing

- AS topology reflects business relationships between ASes.
- Business relationships impact what routes are chosen, and what routes are acceptable.
- Inter-domain routing design must support:
  - Scalability
  - Autonomy (support policy choices)
  - Privacy

BGP: Importing and Exporting

Lecture 8, Spring 2026

Autonomous Systems

- What are ASes?
- Business Relationships

Goals of Inter-Domain Routing

- Policy-Based Routing
- Gao-Rexford Rules

BGP (Border Gateway Protocol)

- **Importing and Exporting**
- Aggregation and Path-Vector

A Brief History of BGP

- Least-cost routing algorithms are based on old ideas.
    - Shortest-path algorithms predate the Internet: Dijkstra's (1956), Bellman-Ford (1958).
- Autonomous systems grew out of necessity.
    - Internet control transferred from the US government to various companies.
    - The Internet grew too big for least-cost routing algorithms.
- BGP is the one and only inter-domain routing protocol.
    - Remember: Everyone must agree on the same one.

Starting Point: Distance-Vector or Link-State?

Distance-vector or link-state: Which would be a better starting point?

Link-state is a bad choice.

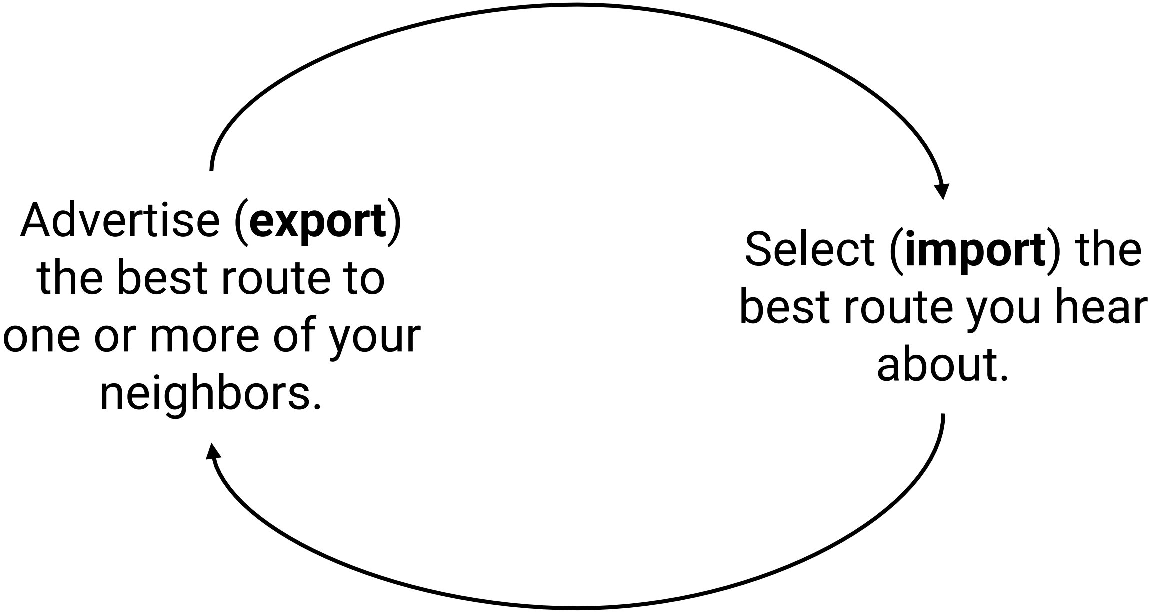
- No privacy: I have to reveal my policies to everybody else.
- No autonomy: Everyone has to agree on a metric (e.g. least-cost) for consistency.

Distance-vector is a better choice.

- BGP extends distance-vector to accommodate policy.
- BGP and distance-vector are similar:
    - Advertisements are specific to one destination (i.e. one prefix).
    - No global sharing of network topology.
    - Iterative and distributed convergence on paths.

Importing and Exporting Paths

New terminology for old ideas:



Advertise (**export**) the best route to one or more of your neighbors.

Select (**import**) the best route you hear about.

Use policy to decide which route to import, and who to export routes to.

Importing Paths

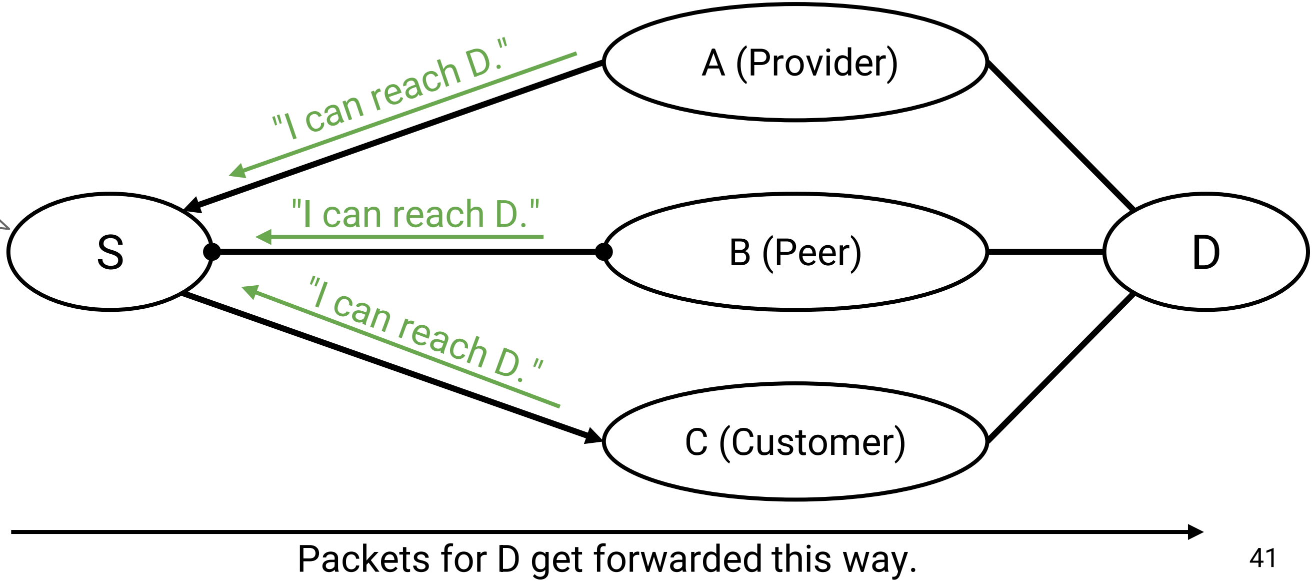
Gao-Rexford import policy: Pick the route advertised by customer > peer > provider.

- Contrast with distance-vector: Pick the shortest route.

Import decision determines where an AS sends its outbound traffic.

- Why? Because this involves choosing a route.

I prefer the path via C, because C will pay me to forward packets.



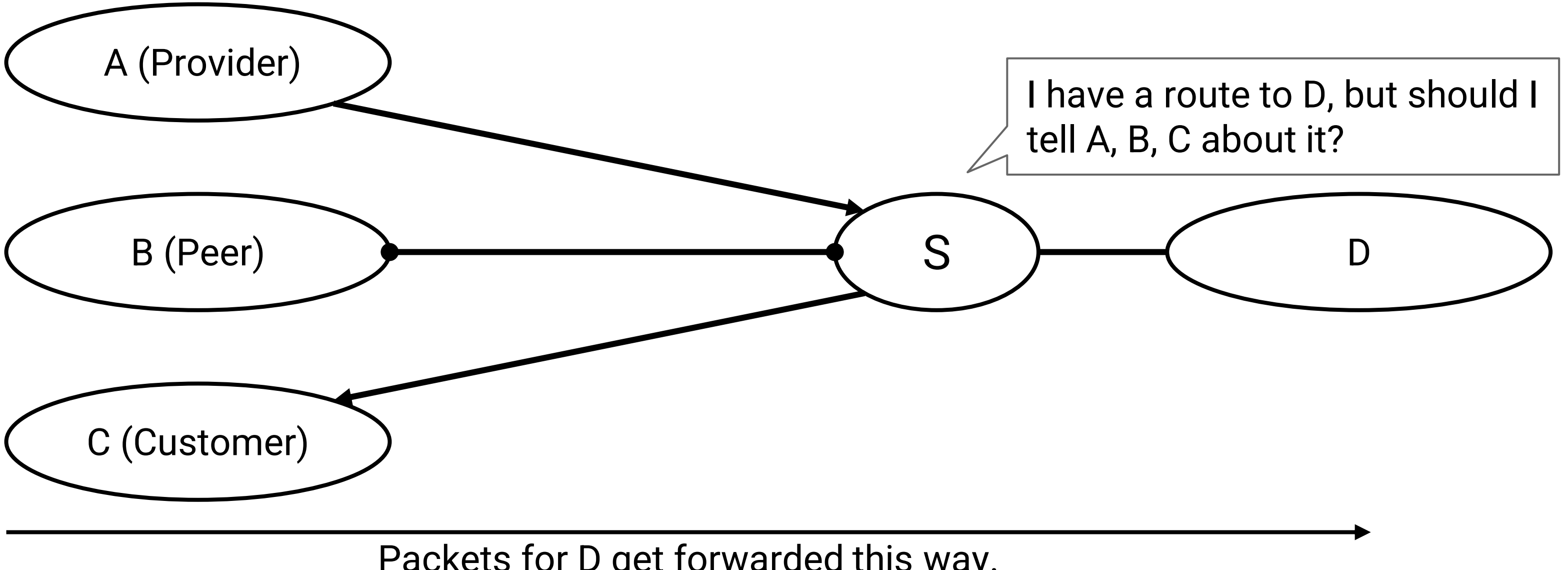
Exporting Paths

Gao-Rexford export policy: I only export a route if participating in it makes me money.

- Contrast with distance-vector: Advertise a route to all neighbors.

Export decision determines what traffic an AS will carry.

- Why? Advertising a route means I'm letting others forward traffic through me.



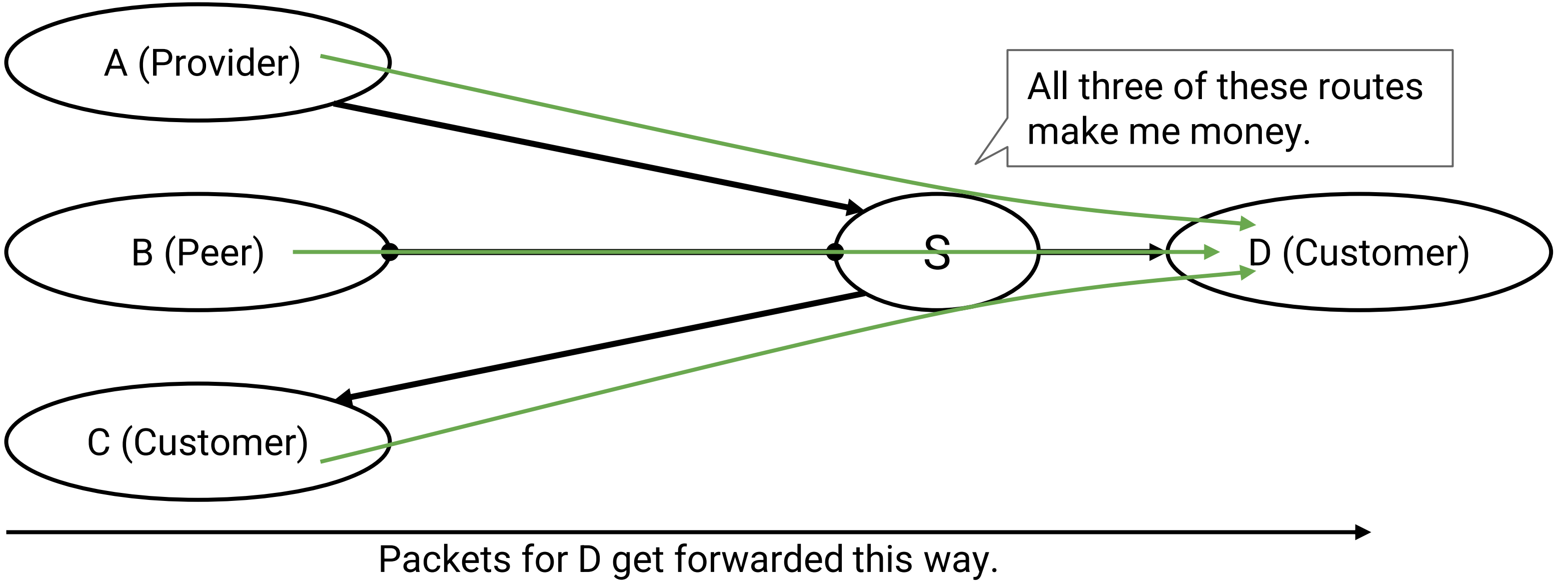
I have a route to D, but should I tell A, B, C about it?

Packets for D get forwarded this way.

Exporting Paths From Customers

Gao-Rexford export policy: I only export a route if participating in it makes me money.

- If I receive a route from a customer, export it to everybody.
- The resulting route will be profitable: There's a customer on one end.



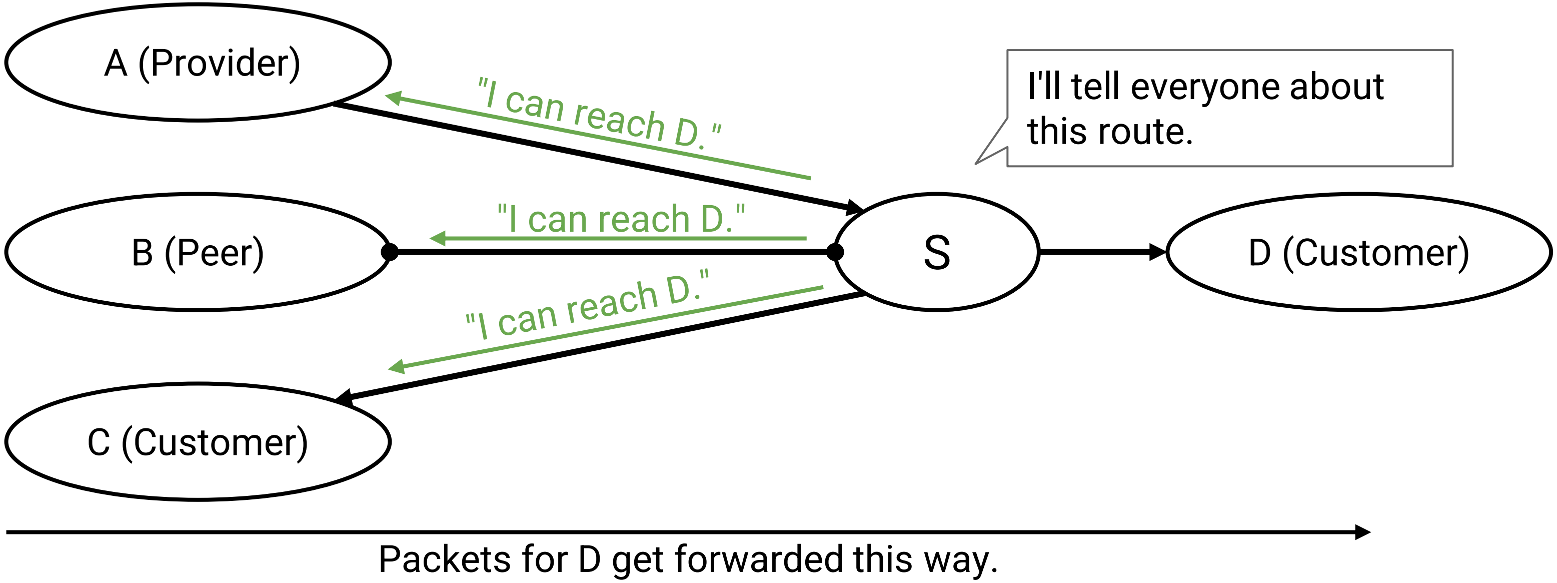
All three of these routes make me money.

Packets for D get forwarded this way.

Exporting Paths From Customers

Gao-Rexford export policy: I only export a route if participating in it makes me money.

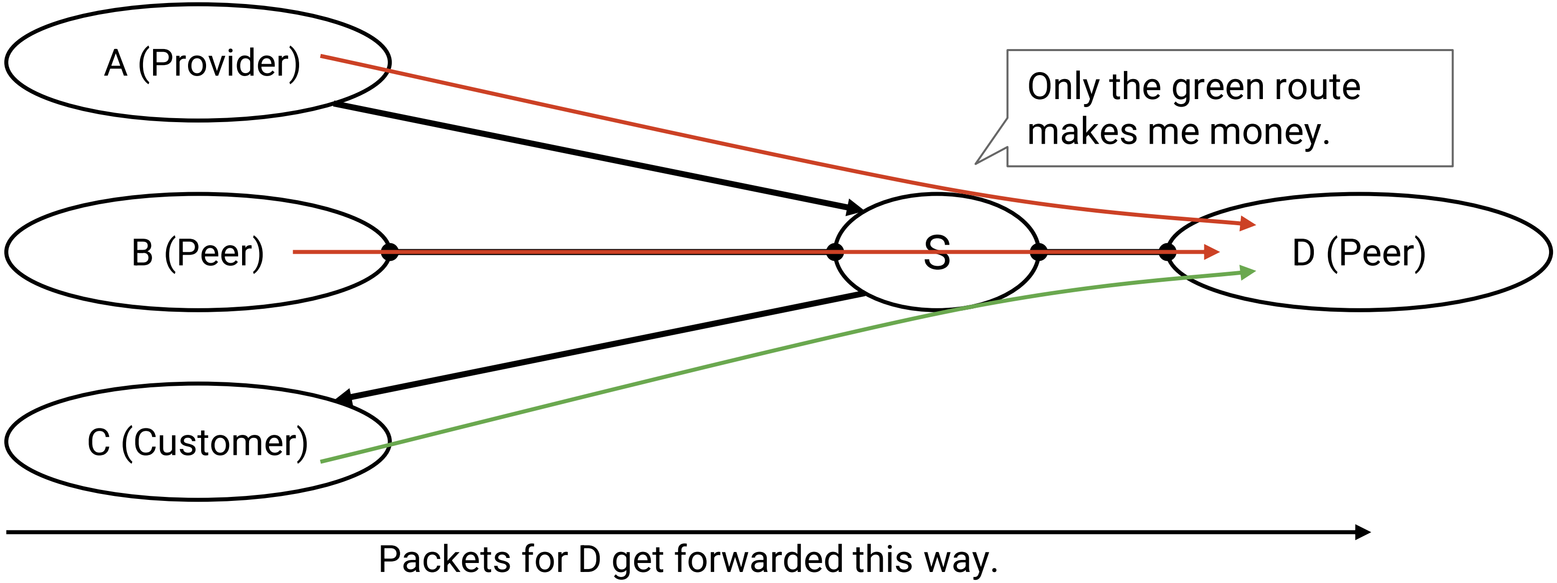
- If I receive a route from a customer, export it to everybody.
- The resulting route will be profitable: There's a customer on one end.



Exporting Paths From Peers

Gao-Rexford export policy: I only export a route if participating in it makes me money.

- If I receive a route from a peer, only export it to a customer.
- The peer isn't paying, so I only profit if the person on the other end is a customer.



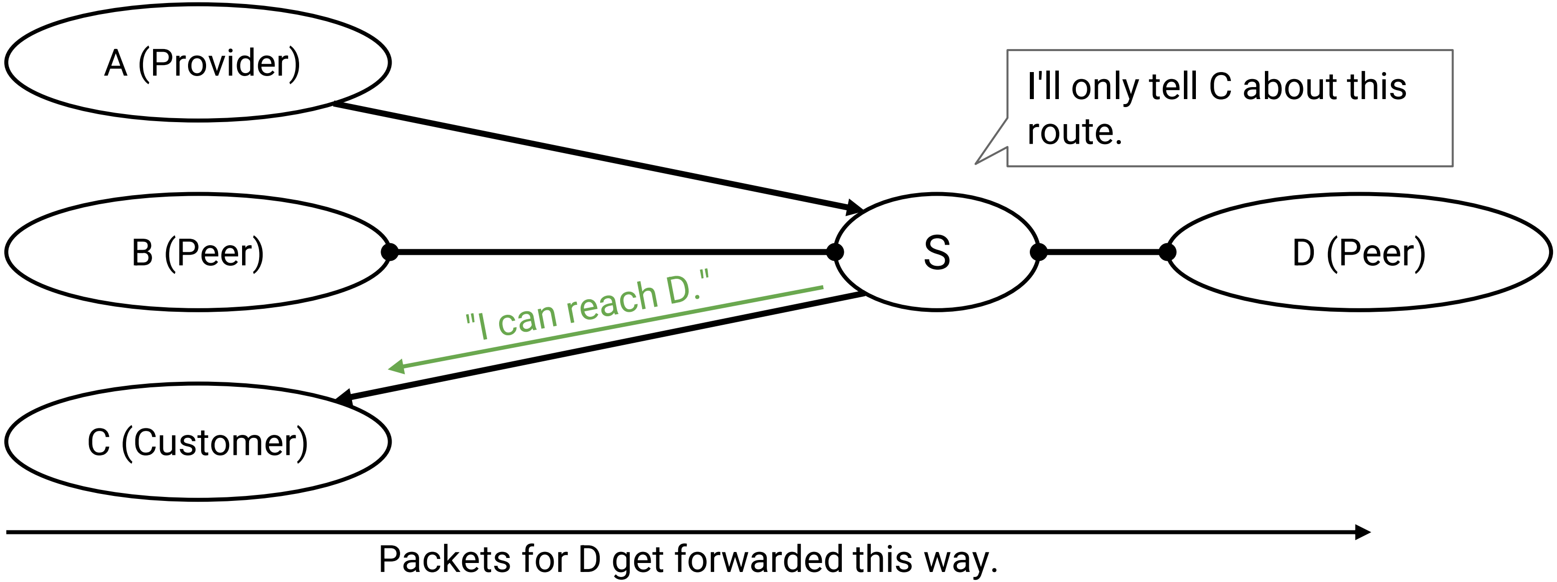
Only the green route makes me money.

Packets for D get forwarded this way.

Exporting Paths From Peers

Gao-Rexford export policy: I only export a route if participating in it makes me money.

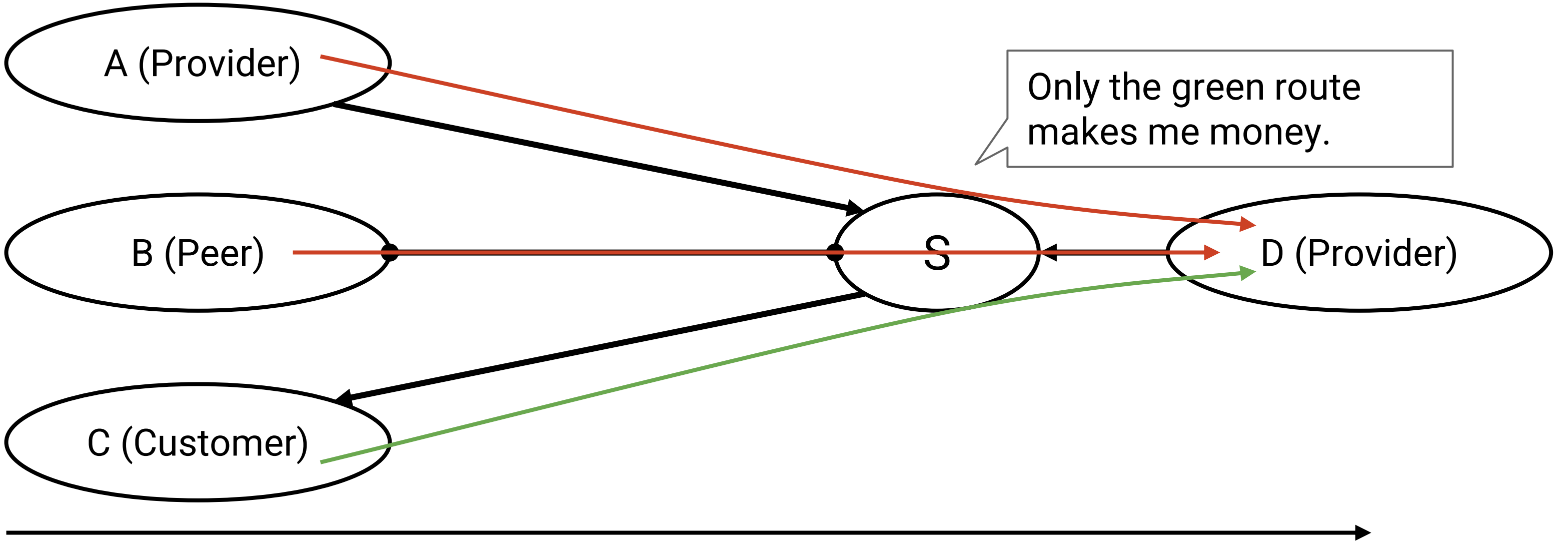
- If I receive a route from a peer, only export it to a customer.
- The peer isn't paying, so I only profit if the person on the other end is a customer.


A (Provider)
B (Peer)
C (Customer)
S
D (Peer)
"I can reach D."
I'll only tell C about this route.
Packets for D get forwarded this way.

Exporting Paths From Providers

Gao-Rexford export policy: I only export a route if participating in it makes me money.

- If I receive a route from a provider, only export it to a customer.
- I pay the provider, so I only profit if the person on the other end is a customer.



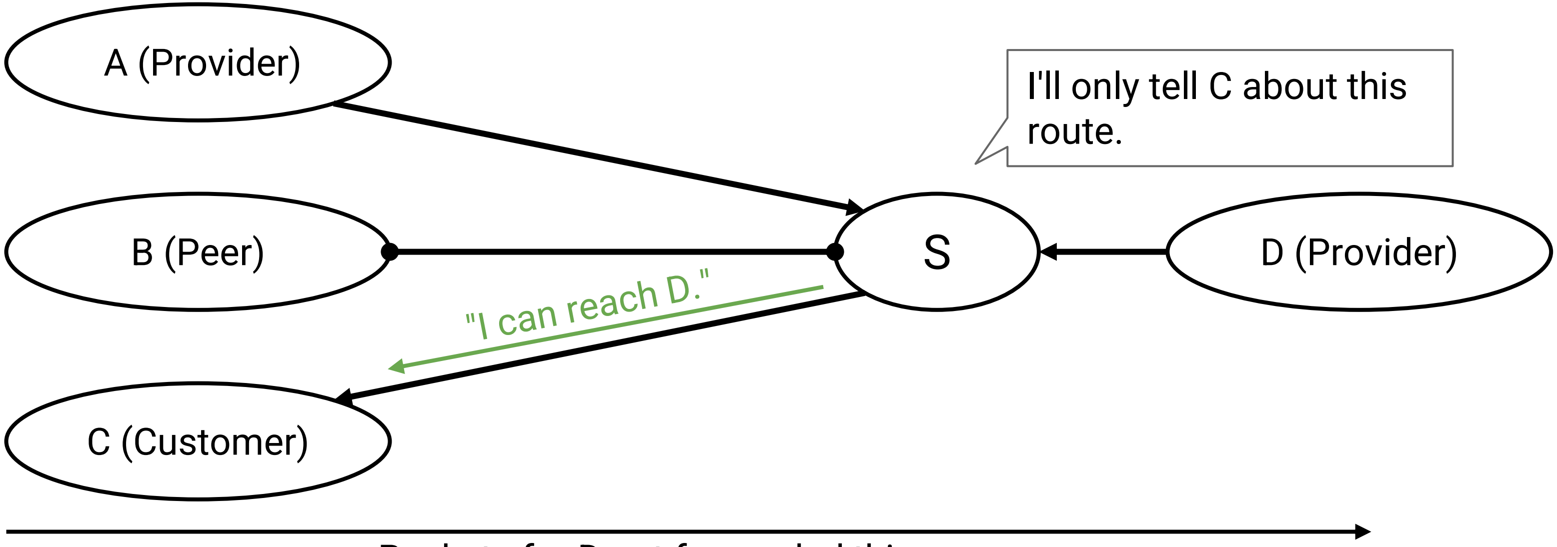
Only the green route makes me money.

Packets for D get forwarded this way.

Exporting Paths From Providers

Gao-Rexford export policy: I only export a route if participating in it makes me money.

- If I receive a route from a provider, only export it to a customer.
- I pay the provider, so I only profit if the person on the other end is a customer.



I'll only tell C about this route.

Packets for D get forwarded this way.

Exporting Paths From Providers

Gao-Rexford export policy: I only export a route if participating in it makes me money.

Key idea: The route needs a customer on at least one side.

- If a customer advertised the route, we already have a customer on one side.
  The other side can be anybody.
- If a peer/provider advertised the route, we're still missing a customer.
  The other side must be a customer.

Route advertised by...	Export route to...
Customer	Everyone (providers, peers, customers)
Peer	Customers only
Provider	Customers only

BGP: Aggregation and Path-Vector

Lecture 8, Spring 2026

Autonomous Systems

- What are ASes?
- Business Relationships

Goals of Inter-Domain Routing

- Policy-Based Routing
- Gao-Rexford Rules

BGP (Border Gateway Protocol)

- Importing and Exporting
- **Aggregation and Path-Vector**

Modifications from Distance-Vector to BGP

Changes we've made so far:

- Import paths based on policy (e.g. prefer customer > peer > provider), instead of based on least-cost.
- Export paths based on policy (e.g. only advertise profitable routes), instead of advertising all routes.

We need to make two more changes:

- Aggregating prefixes for scalability.
- Changing from distance-vector to path-vector.

Modification #1: Aggregating Prefixes

Recall: For scalability, each forwarding table entry maps a range of IPs to a next hop.

Each AS is addressed by a prefix (all machines inside the AS share the same prefix).

BGP can **aggregate** multiple entries into one, combining ranges into one larger range.

Destination Prefix	Next hop
12.1.0.0/16	Physical port #1
12.2.0.0/16	Physical port #1
12.3.0.0/16	Physical port #1

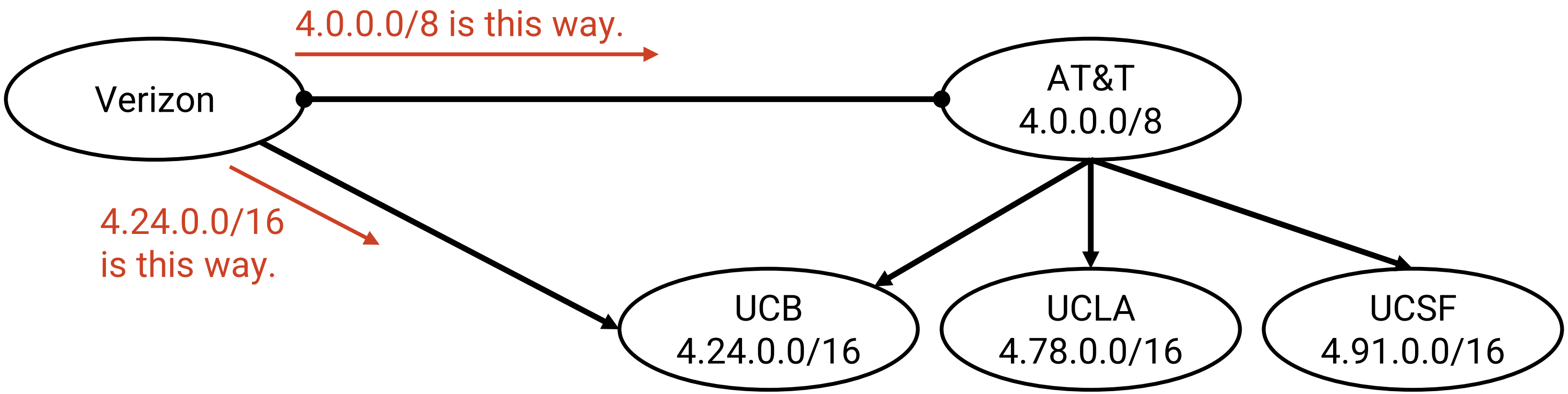
Destination Prefix	Next hop
12.0.0.0/8	Physical port #1

Modification #1: Aggregating Prefixes

Recall: Aggregation scales because addresses are hierarchical.

Multi-homing: An AS having multiple providers.

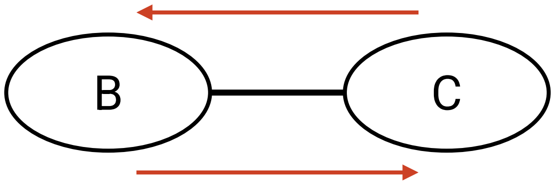
- Multi-homing limits aggregation.



Modification #2: Distance-Vector → Path-Vector

Problems with distance-vector-based BGP:

- There might be loops.
  - Least-cost routing guaranteed no loops.
  - Switching to policy-based routing caused us to lose that guarantee.



B's policy: "I like sending packets via C."
C's policy: "I like sending packets via B."

- We can support Gao-Rexford rules, but not arbitrary policies.
  - Suppose my policy is: "My traffic should never pass through AS#2019.
  - I get an announcement: "I am AS#20 and I can reach D with cost 10.
  - How do I know if this route satisfies my policy?

Modification #2: Distance-Vector → Path-Vector

Solution: Change from distance-vector to **path-vector**.

- Instead of advertising distance to destination, advertise the whole AS path.
- Example: "I am A, and I have a path to D via A → B → C → D."

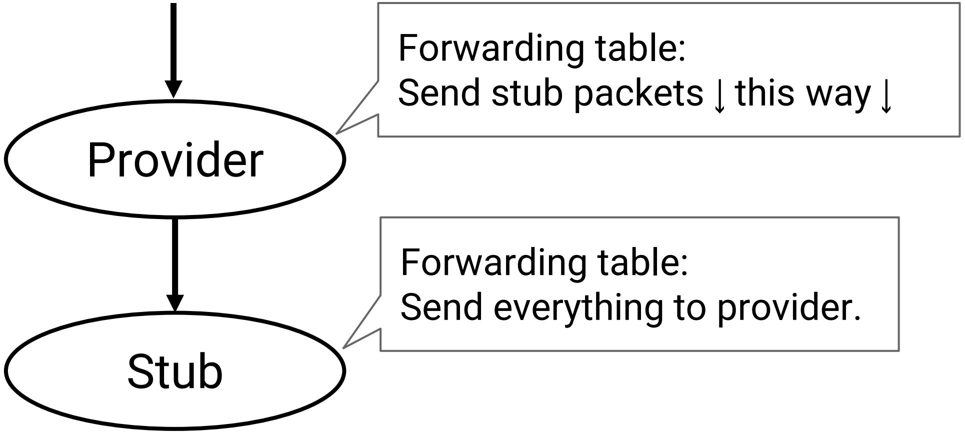
This solves both of our problems.

- Loop detection: Check if adding myself to the path creates a loop.
  - If I'm C, adding myself to the path creates a cycle: C → A → B → C → D.
  - If the path has a cycle (after adding myself), reject the advertisement.
- Arbitrary policies: The entire path is available for checking against policy.
  - If my policy is "don't send my packets through B," I can check the path and reject the advertisement.

Stub ASes Use Default Routes

If a stub AS is connected to a single provider, it doesn't need to run BGP.

- The stub AS default-routes everything to the provider.
- The provider advertises that they can reach the stub AS's prefix.
  - The provider is advertising on behalf of the stub.



Forwarding table:
Send stub packets ↓ this way ↓

Forwarding table:
Send everything to provider.

Summary: Inter-Domain Routing

- In the AS graph, edges reflect physical connections and business relationships.
    - Customers pay providers.
    - Peres don't pay each other.
- Paths are selected based on policy.
- Policy (e.g. Gao-Rexford rules) reflects business goals.
    - Making money is good.
    - Don't do work for free.
    - Good stuff (reachability, converge) happens if you follow Gao-Rexford rules.
- BGP extends distance-vector to implement inter-domain routing.
    - Destinations are IP prefixes that can be aggregated.
    - Each AS advertises its path to a prefix.
    - Policy dictates which paths an AS selects (import policy),
      and which paths it advertises (export policy).