CSCI-1680
Network Layer:
Inter-domain Routing

Nick DeMarinis

Based partly on lecture notes by Rodrigo Fonseca, Rob Sherwood, David Mazières, Phil Levis, John Jannotti

Warmup

Suppose router R has the following table:

What happens when it gets
this update from router S?

Warmup

Suppose router R has the following table:

Dest.	Cost	Next Hop
A	3	S
B	4	T
C	5	S
D	6	U

What happens when it gets this update from router S?

Warmup

Suppose router R has the following table:

Dest.	Cost	Next Hop
A	3	S
B	4	T
C	5	S
D	6	U

(NO CHANGE, SAME COST)
(TIE)
(COST INCREASED)
(BETTER → UPDATE)

What happens when it gets this update from router S?

Dest.	Cost
A	2
B	3
C	5
D	4
E	2

ADD

Administrivia

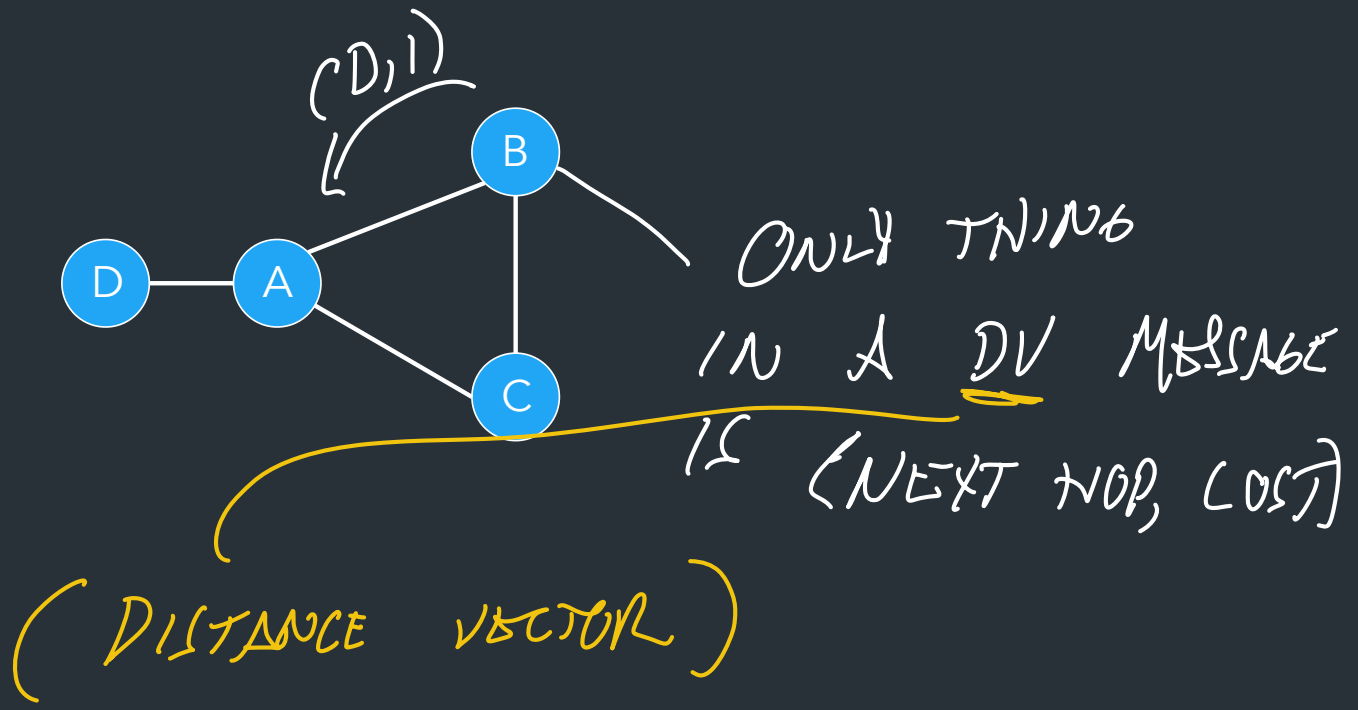
- You should have completed your IP milestone meeting
  - If not, contact us ASAP
- HW2: Out today, probably

- IP: Due next Thursday, October 19
  - New Wireshark testing guide, other resources
  - Do not leave this until the last minute

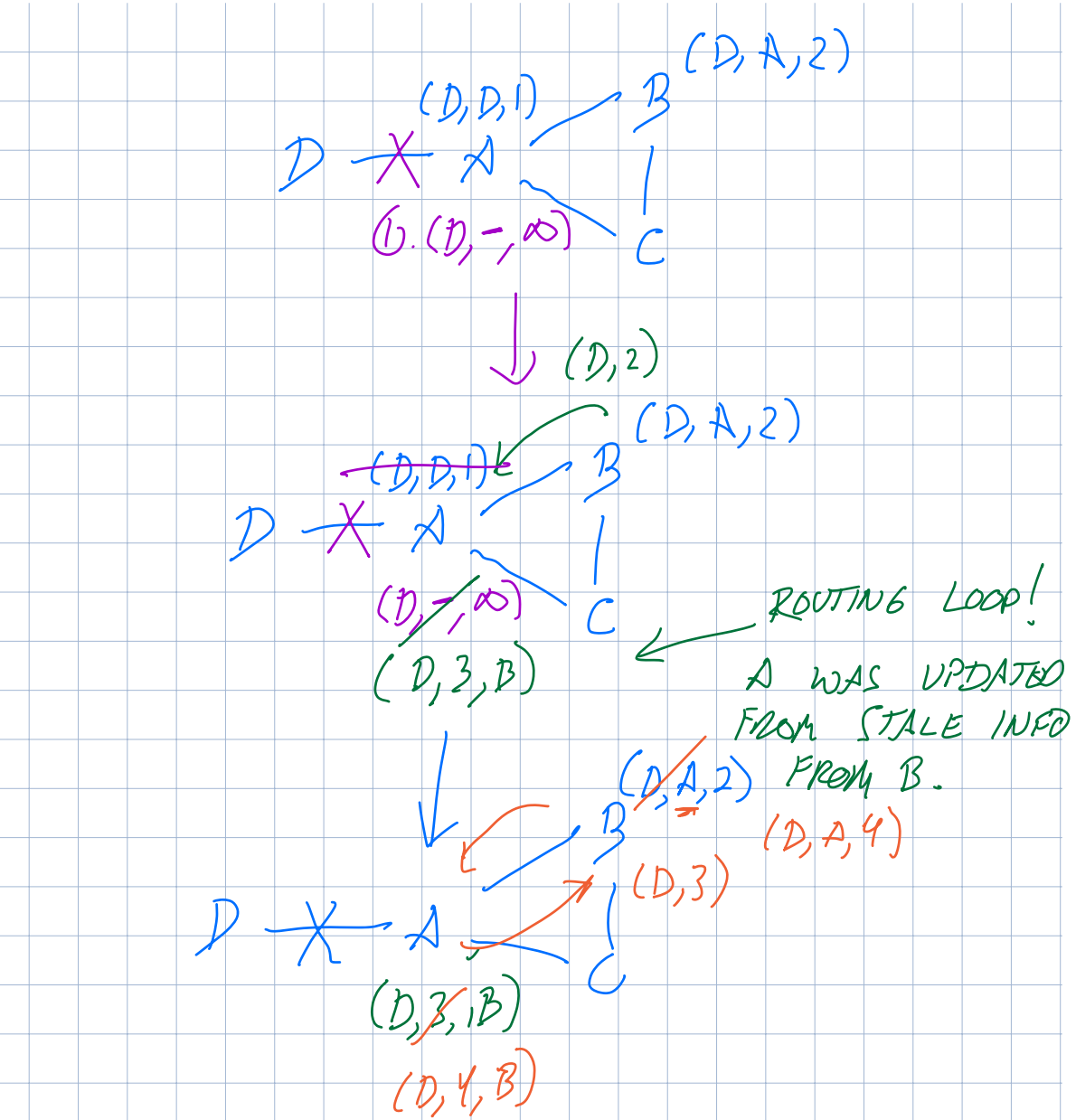
Topics for today

- More on intra-domain (interior) routing
  - Challenges in RIP
  - Link-state routing
- Inter-domain routing: BGP

What happens when the D-A link fails?



Updates occur in a loop with increasing cost until cost reaches infinity (16)!
=> Count to infinity => long time to converge when links fail



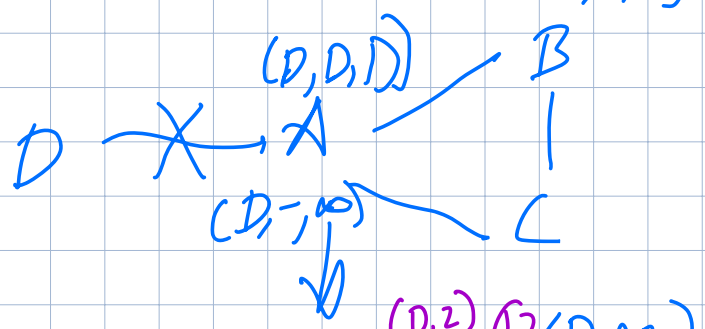
$\Rightarrow$ COUNT TO INFINITY $\Rightarrow$ COST KEEPS INCREASING UNTIL REACH ∞

$\Rightarrow$ "BAD NEWS TRAVELS SLOWLY"

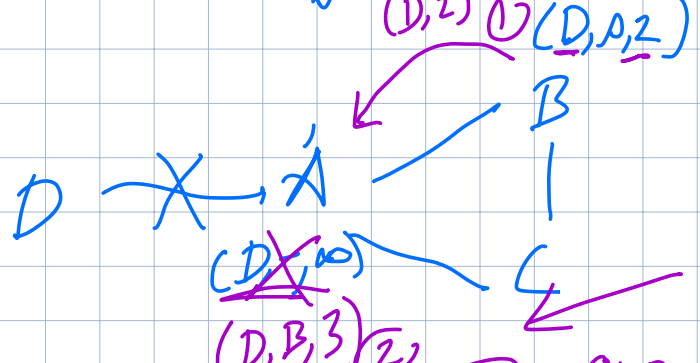
$\Rightarrow$ RIP = "INFINITY" = 16

RIP: WHAT HAPPENS WHEN D-A LINK FAILS?

(D, A, 2) IN RIP
∞ = 16

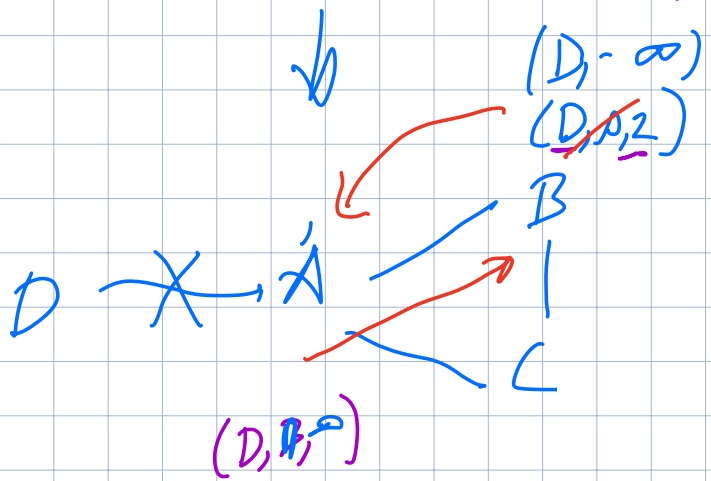


SLIGHTLY CLEANER VERSION FROM LAST YEAR



ROUTING LOOP.

OLD INFO OVERWRITES STATE!



⇐ - UPDATES OCCUR IN A LOOP W/ INCREASING COST UNTIL COST REACHES ∞

⇒ COUNT TO INFINITY - LONG CONVERGE TIME.

Can we avoid loops?

- Does IP TTL help? Nope.
- Simple approach: consider a small cost n (e.g., 16) to be infinity
  - After n rounds decide node is unavailable
  - But rounds can be long, this takes time

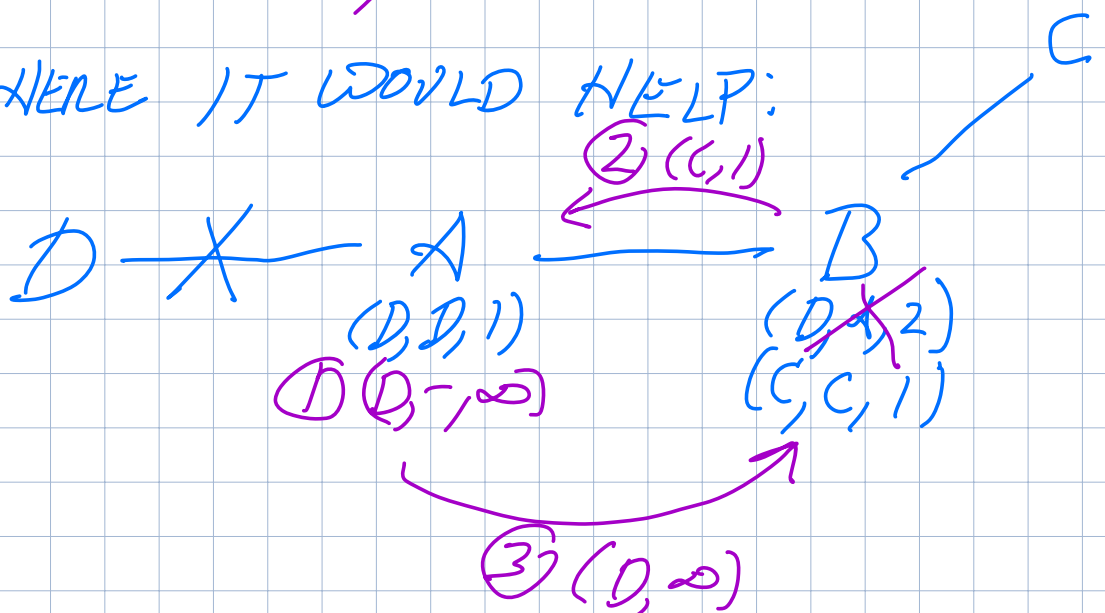
Fundamental problem: distance vector only based on local information!
=> Not enough info to resolve loops, race conditions, count-to-infinity, but there are some tricks we can do…

SPLIT HORIZON

- IF A USES N AS NEXT HOP FOR D, DO NOT REPORT TO N ABOUT D

⇒ PREVENTS "LINEAR" ROUTING LOOPS, BUT NOT OTHERS

WHERE IT WOULD HELP:



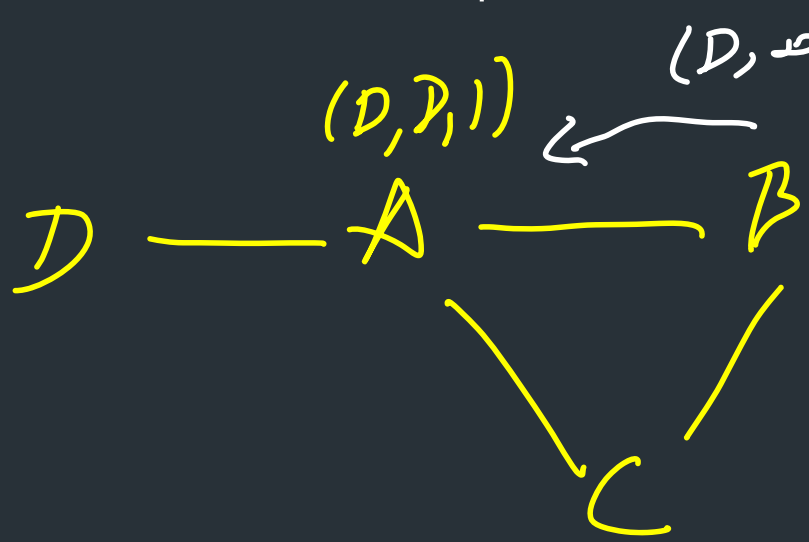
① D-A LINK GOES DOWN

② WHEN B SENDS UPDATE TO A, IT WOULD NOT TELL INCLUDE A

③ A UPDATES B W/ (D, ∞)

Split Horizon + Poison Reverse

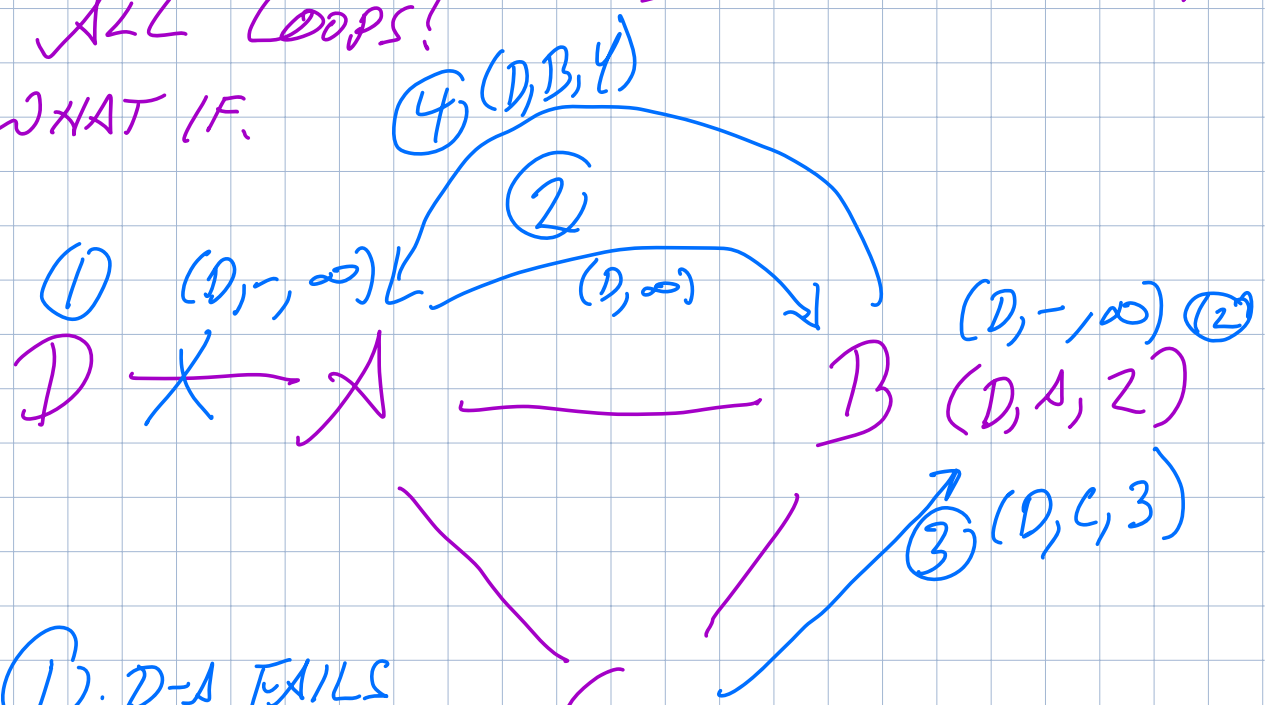
- Rather than not advertising routes learned from A, explicitly include cost of ∞.
- Faster to break out of loops, but increases advertisement sizes



POISON REVERSE IS A COMMON CONVENTION, MIGHT HELP TIME TO CONVERGE IN SOME CASES, BUT HARD TO SEE EFFECT VS. SPLIT HORIZON.

BUT EVEN W/ SPLIT HORIZON + POISON REVERSE, CAN'T PREVENT ALL LOOPS!

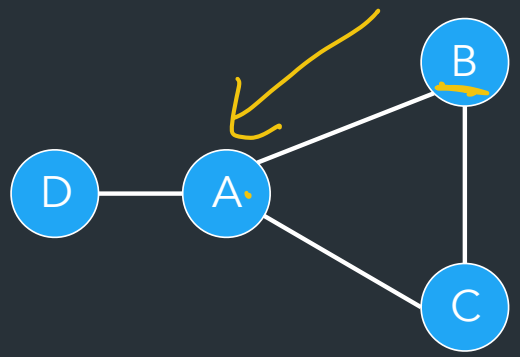
WHAT IF.



① . D-A FAILS
② A UPDATES B (D, ∞)
③ . C SENDS (D, 2) TO B!
↳ RACE CONDITION! C MIGHT SEND OLD UPDATE TO B BEFORE C GETS UPDATE FROM A!
④ B UPDATES A, OVERWRITES A'S ENTRY
⑤ ... COUNT TO INFINITY ...

WHAT CAN WE DO?

Practice



B's routing table

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

Routers A,B,C,D use RIP. When B sends a periodic update to A, what does it send…

- When using standard RIP?
- When using split horizon + poison reverse?

STANDARD
(A, 1)
(C, 1)
(D, 2)

SH + PR
(A, ∞)
(C, 1)
(D, ∞)

Link State Routing

(INTRA-ON ROUTING: WITHIN AN ORGANIZATION)

Link State Routing: The Alternative

Example: OSPF

Strategy: each router sends information about its neighbors to all nodes

- Nodes build the full graph, not just neighbor info

⇒ MORE UPDATES, TO MORE NODES.

- Updates have more state info

⇒ VERSIONING, TTL, . . .

Link State Routing: The Alternative

Example: OSPF

Strategy: each router sends information about its neighbors to all nodes

- Nodes build the full graph, not just neighbor info

- Updates have more state info

Tradeoffs?

Link State Routing: The Alternative

Strategy: each router sends information about its neighbors to all nodes

- Nodes build the full graph, not just neighbor info

  => Can define "areas" to scale this in large networks

- Updates have more state info
  - Node IDs, version info (sequence number, TTL), …

    => Can be used to detect loops, stale info

↳ HARD PROBLEM TO GET INFO TO ALL NODES.

Link State Routing: The Alternative

Strategy: each router sends information about its neighbors to all nodes

- Nodes build the full graph, not just neighbor info

  => Can define "areas" to scale this in large networks

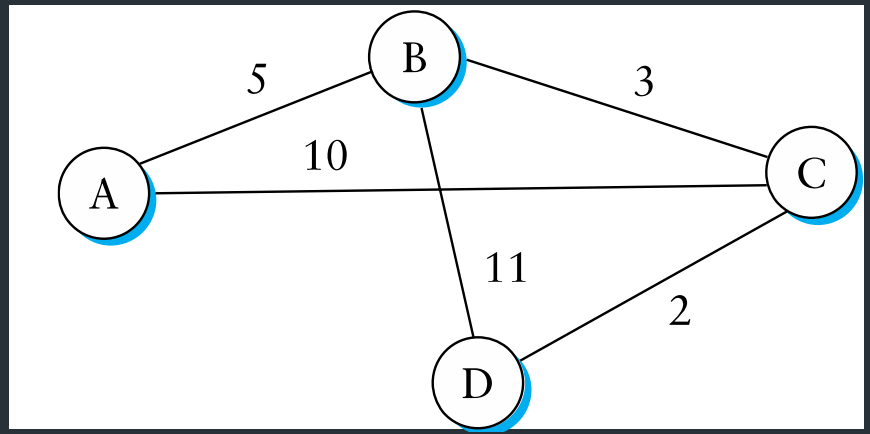
- Updates have more state info
  - Node IDs, version info (sequence number, TTL), …

  => Can be used to detect loops, stale info

⇒ Focuses on building a consistent view of network state

Link State Routing: how it works

- Each node computes shortest paths from itself
- How? Dijkstra's algorithm
  - Given: full graph of nodes
  - Find best next hop to each other node



Tradeoffs?

Tradeoffs: Link State (LS) vs. Distance Vector (DV)

- LS sends more messages vs. DV
- LS requires more computation vs. DV
- Convergence time
  - DV: Varies (count-to-infinity)
  - LS: Reacts to updates better
- Robustness
  - DV: Bad updates can affect whole network
  - LS: Bad updates affect a single node's update

=> MORE INFO IN LS

=> MORE COMPUTATION AT EACH NODE.

LS: HARDER TO HAVE BAD INFO PROPAGATE.

Tradeoffs: Link State (LS) vs. Distance Vector (DV)

- LS sends more messages vs. DV
- LS requires more computation vs. DV
- Convergence time
  - DV: Varies (count-to-infinity)
  - LS: Reacts to updates better
- Robustness
  - DV: Bad updates can affect whole network
  - LS: Bad updates affect a single node's update

=> RIP isn't used in production environments anymore…

Examples

- RIPv2
  - Fairly simple implementation of DV
  - RFC 2453 (38 pages)
- OSPF (Open Shortest Path First)
  - More complex link-state protocol
  - Adds notion of areas for scalability
  - RFC 2328 (244 pages)
- ISIS (Intermediate System to Intermediate System)
  - OSI standard (210 pages)
  - Link-state protocol (similar to OSPF)
  - Does not depend on IP

So why not just use OSPF everywhere?

Does it scale?

Why not?

⇒ Can't build a full routing graph with the whole Internet

⇒ More a policy problem than a technical problem

- No unified way to represent cost
- No single administrator
- Networks (ASes) have different policies on what "best" routes to choose

DIFFERENT FOR EVERY AS, ADMIN.

Why not?

⇒ Can't build a full routing graph with the whole Internet

⇒ More a policy problem than a technical problem

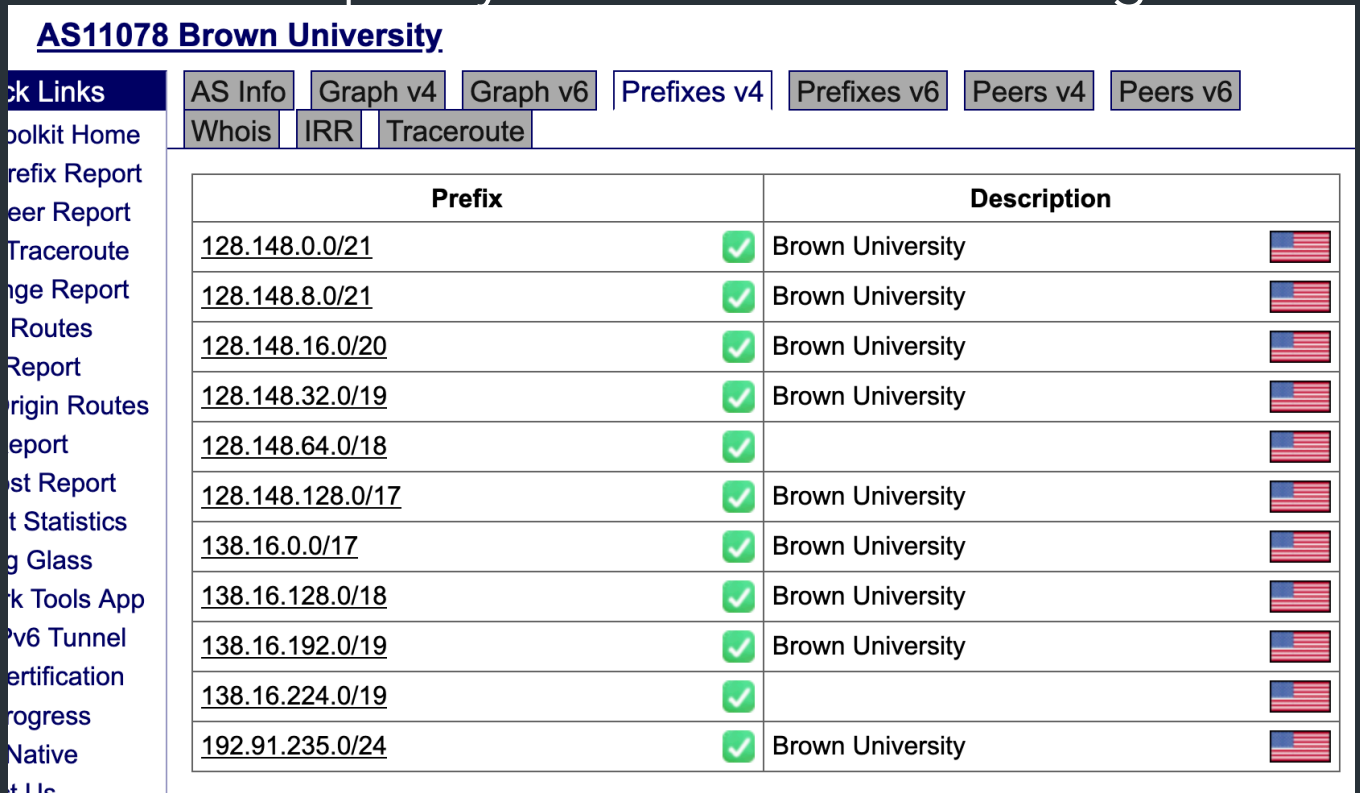
- No unified way to represent cost
- No single administrator
- Networks (ASes) have different policies on what "best" routes to choose

Need a different routing mechanism for exterior routing => BGP

With BGP: we talk about routing to Autonomous Systems (ASes)

= > Generally, large networks that advertise some set of IP prefixes to the Internet

=> Each AS has its own policy for how it does routing



AS11078 Brown University

ck Links
oolkit Home
refix Report
eer Report
Traceroute
nge Report
Routes
Report
rigin Routes
eport
st Report
t Statistics
g Glass
k Tools App
v6 Tunnel
ertification
rogress
Native

AS Info | Graph v4 | Graph v6 | Prefixes v4 | Prefixes v6 | Peers v4 | Peers v6 | Whois | IRR | Traceroute

Prefix	Description
128.148.0.0/21	Brown University
128.148.8.0/21	Brown University
128.148.16.0/20	Brown University
128.148.32.0/19	Brown University
128.148.64.0/18	
128.148.128.0/17	Brown University
138.16.0.0/17	Brown University
138.16.128.0/18	Brown University
138.16.192.0/19	Brown University
138.16.224.0/19	
192.91.235.0/24	Brown University

With BGP: we talk about routing to Autonomous Systems (ASes)

= > Generally, large networks that advertise some set of IP prefixes to the Internet

=> Each AS has its own policy for how it does routing

↳ SEPARATE GOALS
INTERNAL
LAWYERS
POLITICAL AGENDAS
FINANCIAL INCENTIVES.

ORGANIZATION-LEVEL ENTITIES.

BGP: A Path Vector Protocol

Distance vector algorithm with extra information

eg. "I can reach prefix 128.148.0.0/16 through ASes 44444 3356 14325 11078"

BGP UPDATE

AS-PATH

BGP: A Path Vector Protocol

Distance vector algorithm with extra information

eg. "I can reach prefix 128.148.0.0/16 through ASes 44444 3356 14325 11078"

- For each route, router store the complete path (ASs)
- No extra computation, just extra storage (and traffic)

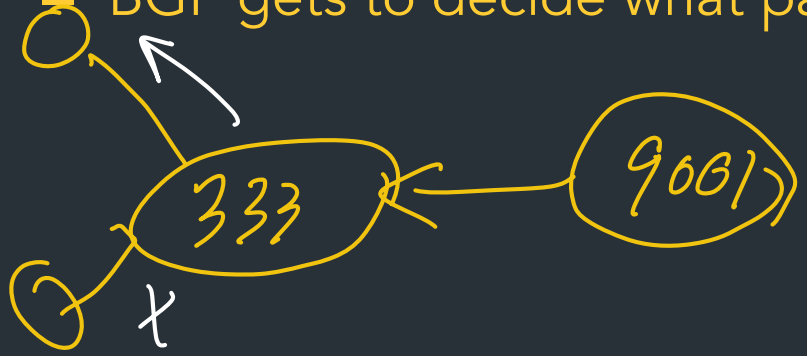
⇒ Can look at path to decide what to do with route
⇒ Can easily avoid loops!

BGP: A Path Vector Protocol

Distance vector algorithm with extra information

eg. "I can reach prefix 128.148.0.0/16 through ASes 44444 3356 14325 11078"

- For each route, router store the complete path (ASs)
- No extra computation, just extra storage (and traffic)
- BGP gets to decide what paths to propagate (send to neighbors)



↳ CUSTOM LOGIC TO DECIDE WHICH NEIGHBORS GET UPDATES, AND WHICH DON'T.

BGP: A Path Vector Protocol

Distance vector algorithm with extra information

eg. "I can reach prefix 128.148.0.0/16 through ASes 44444 3356 14325 11078"

- For each route, router store the complete path (ASs)
- No extra computation, just extra storage (and traffic)
- BGP gets to decide what paths to propagate (send to neighbors)

⇒ Allows enforcing custom policy on how to do routing

BGP Implications

- Explicit AS Path == Loop free (most of the time)
- Not all ASs know all paths
- Reachability not guaranteed
  - Decentralized combination of policies
- AS abstraction -> loss of efficiency
- Scaling
  - 74K ASs
  - 959K+ prefixes
  - ASs with one prefix: 25K
  - Most prefixes by one AS: 10008 (Uninet S.A. de C.V., MX)

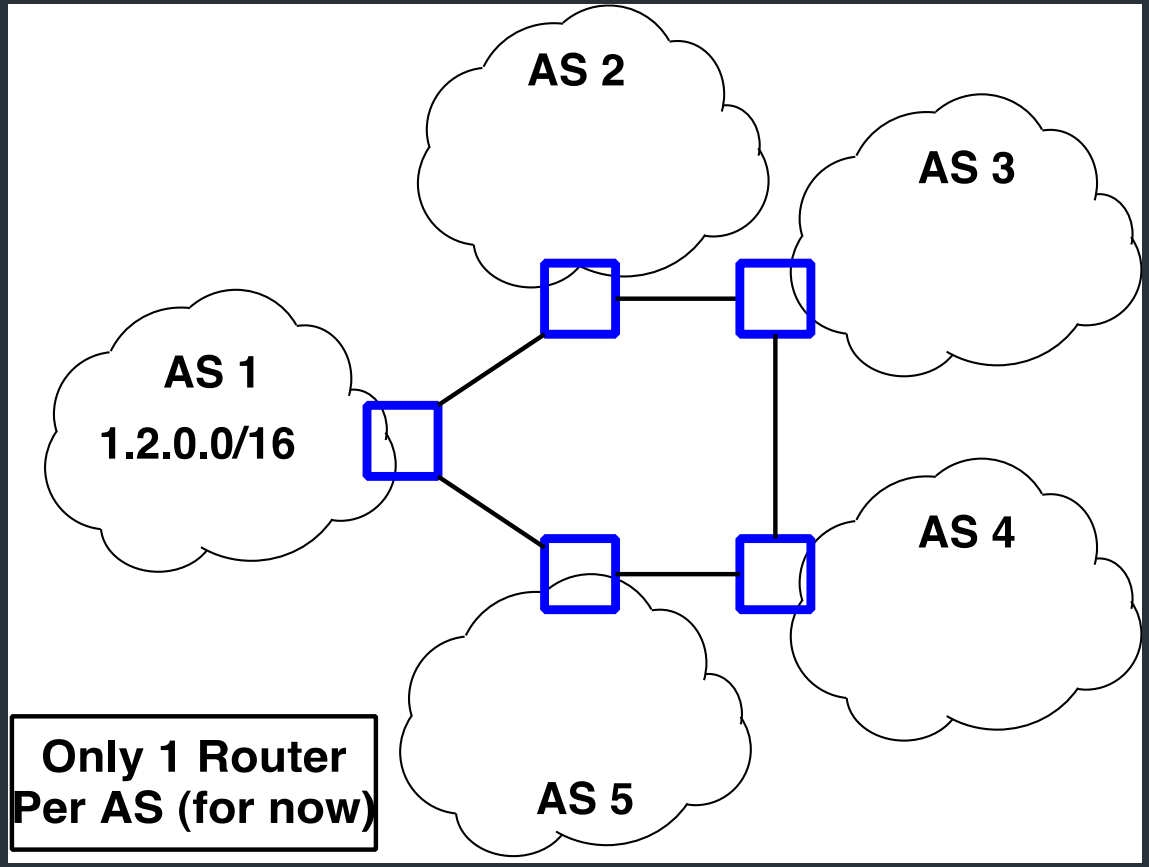
NOT NECESSARILY OPTIMAL

Source: cidr-report 18Oct2022

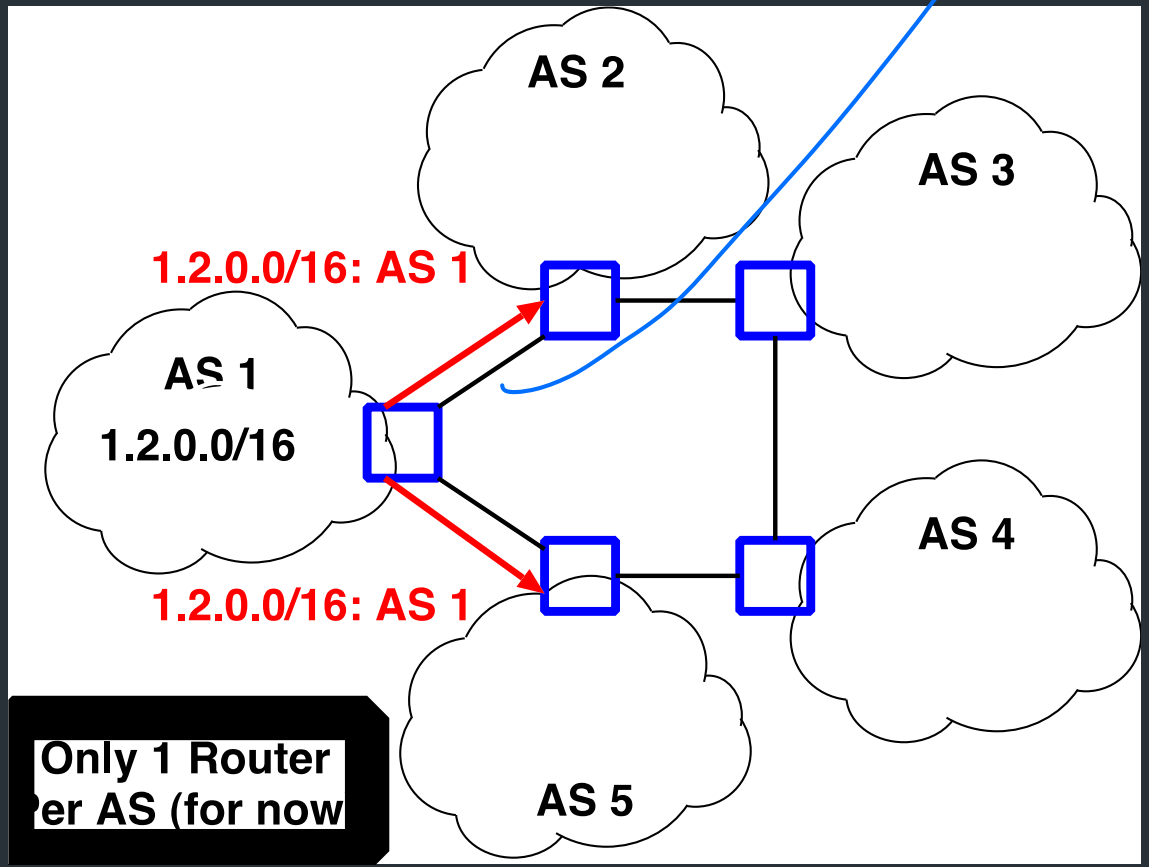
Why study BGP?

- Critical protocol: makes the Internet run
  - Only widely deployed EGP
- Active area of problems!
  - Efficiency
  - Cogent vs. Level3: Internet Partition
  - Spammers use prefix hijacking
  - Pakistan accidentally took down YouTube
  - Egypt disconnected for 5 days
  - NOW: Russia taking over Ukraine's traffic

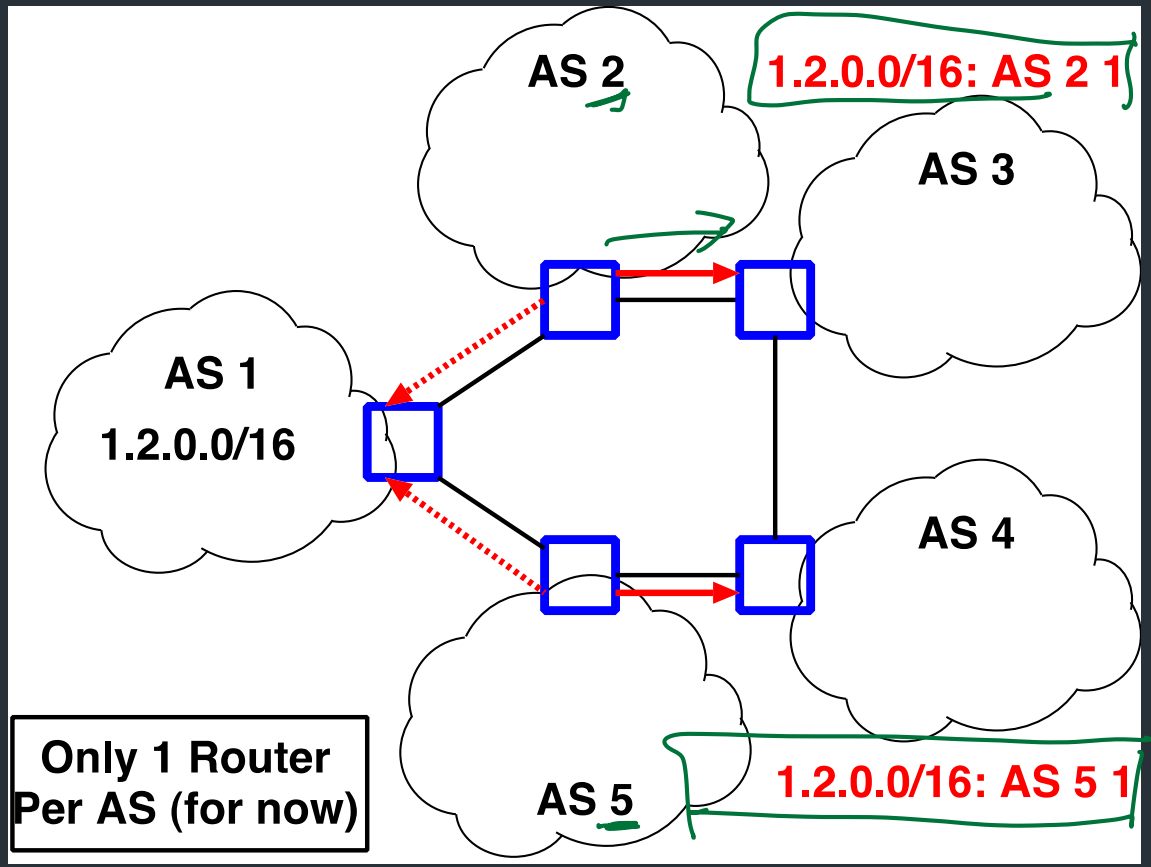
BGP Example



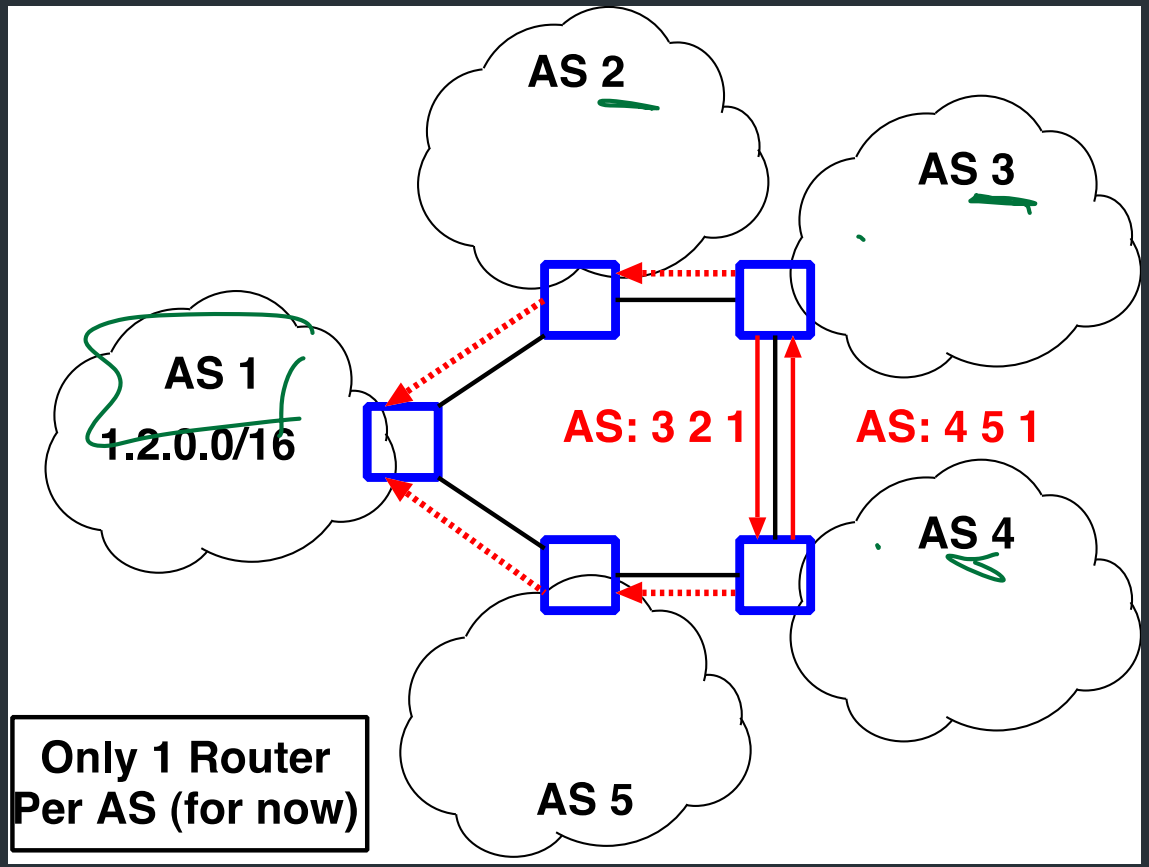
BGP Example


PEERING RELATIONSHIP
AS 2
AS 3
1.2.0.0/16: AS 1
AS 1
1.2.0.0/16
AS 4
1.2.0.0/16: AS 1
AS 5
Only 1 Router
er AS (for now

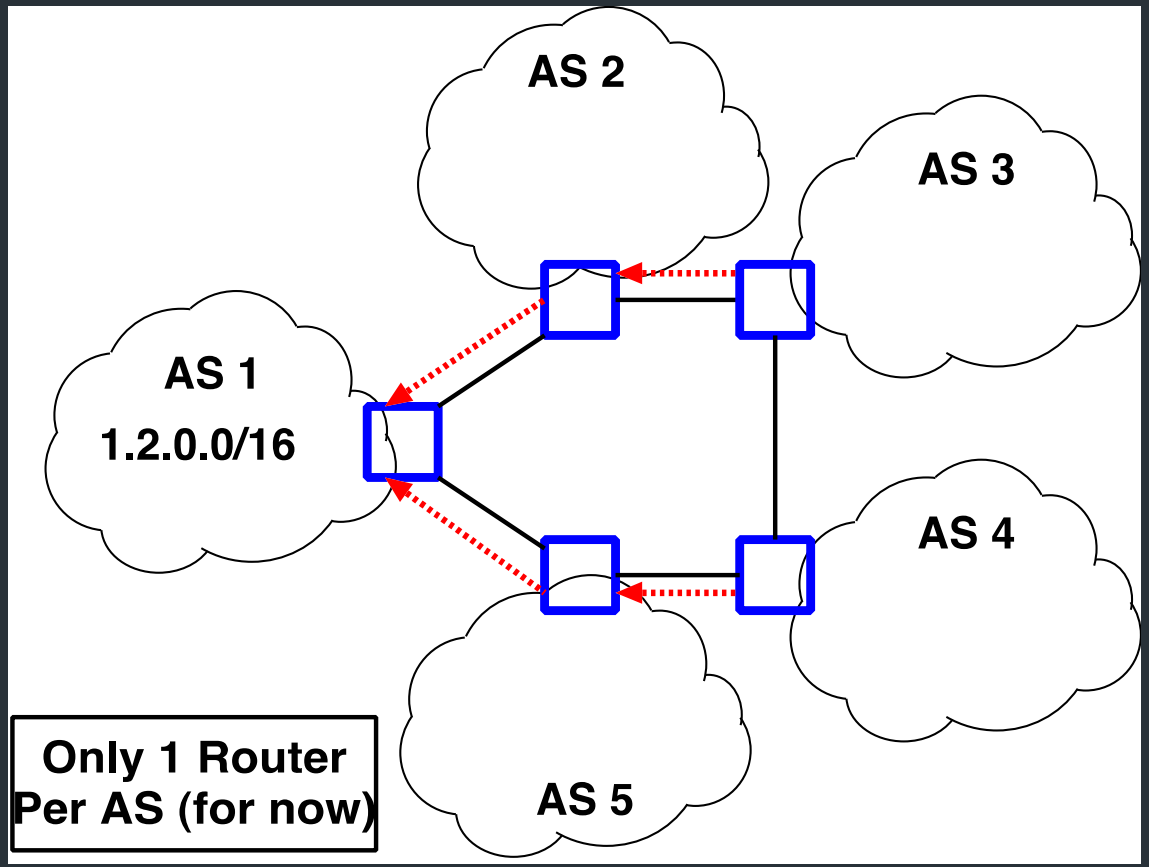
BGP Example


AS 2
1.2.0.0/16: AS 2 1
AS 3
AS 1
1.2.0.0/16
AS 4
Only 1 Router
Per AS (for now)
AS 5
1.2.0.0/16: AS 5 1

BGP Example


AS 2
AS 3
AS 1
1.2.0.0/16
AS: 3 2 1
AS: 4 5 1
AS 4
AS 5
Only 1 Router
Per AS (for now)

BGP Example



Demo: AS11078

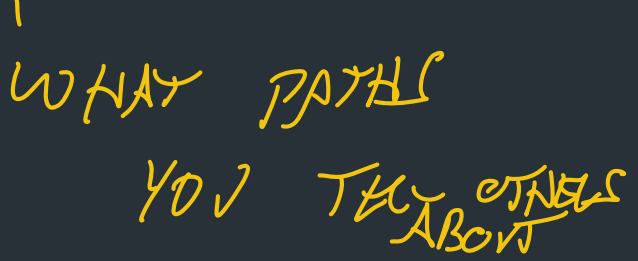
BGP Protocol Details

- BGP speakers: nodes that communicates with other ASes over BGP
- Speakers connect over TCP on port 179
- Exact protocol details are out of scope for this class; most important messages have type UPDATE

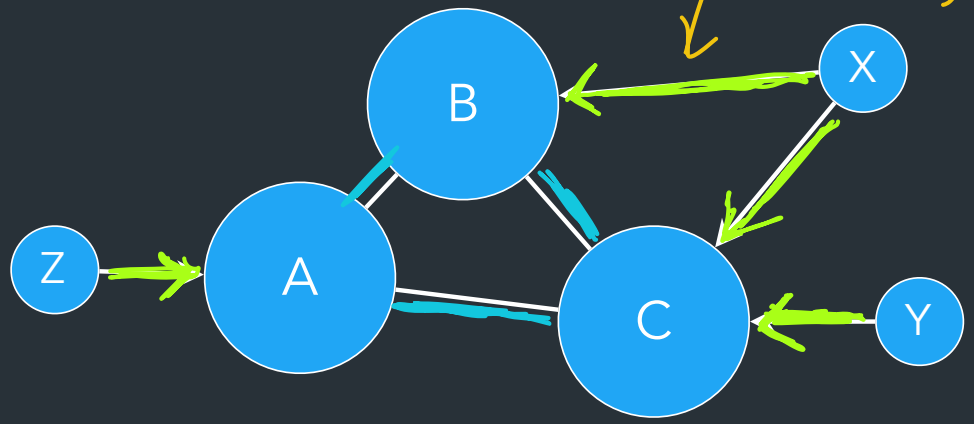
Where do we use policies?

Policies are imposed in how routes are selected and exported

- Selection: which path to use in your network
  - Controls if/how traffic leaves the network
- Export: which path to advertise
  - Controls how/if traffic enters the network



AS Relationships



BIGGER → SMALLER NODES (MORE HIGHLY CONNECTED NODES)

Policies are defined by relationships between Ases

PROVIDER: HIGHLY-CONNECTED, CUSTOMERS PAY THEM FOR CONNECTIVITY (A, B, C IN EXAMPLE)

CUSTOMER: PAYS PROVIDER TO CONNECT, SMALLER (X, Y, Z)

PEER: MUTUALLY-BENEFICIAL, COST-FREE PAIRING BETWEEN PROVIDERS

Example from Kurose and Ross, 5th Ed