Routing (part 2)

Lecture 24

http://www.cs.rutgers.edu/~sn624/352-F24

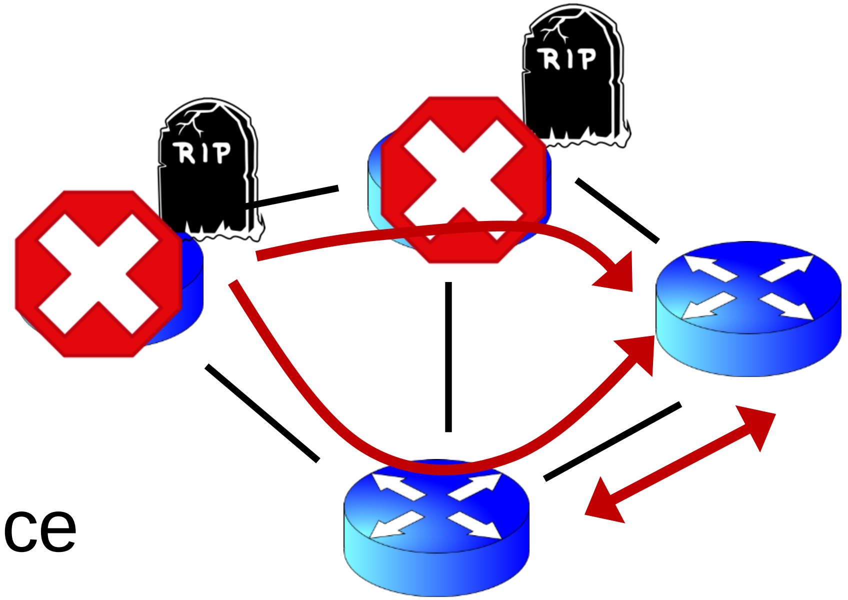
Srinivas Narayana

RUTGERS
UNIVERSITY | NEW BRUNSWICK

Goals of routing:

#1 Good paths

#2 Failure resilience



Routing: Google Maps for the Internet?

Routing protocols

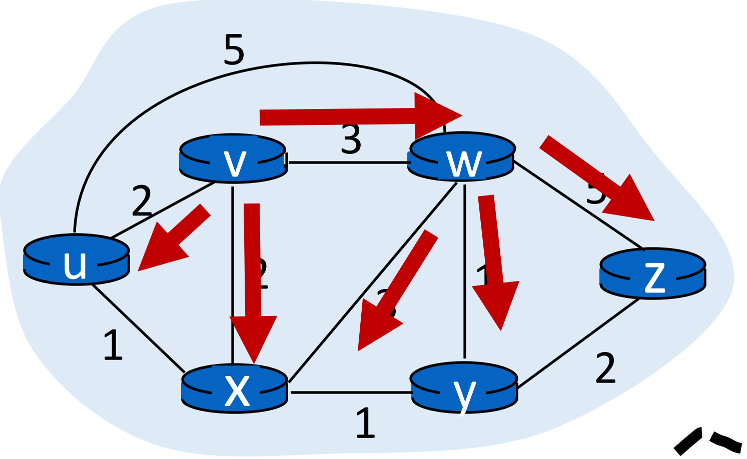
Link state protocols

Distance vector protocols

Q1. What info exchanged?

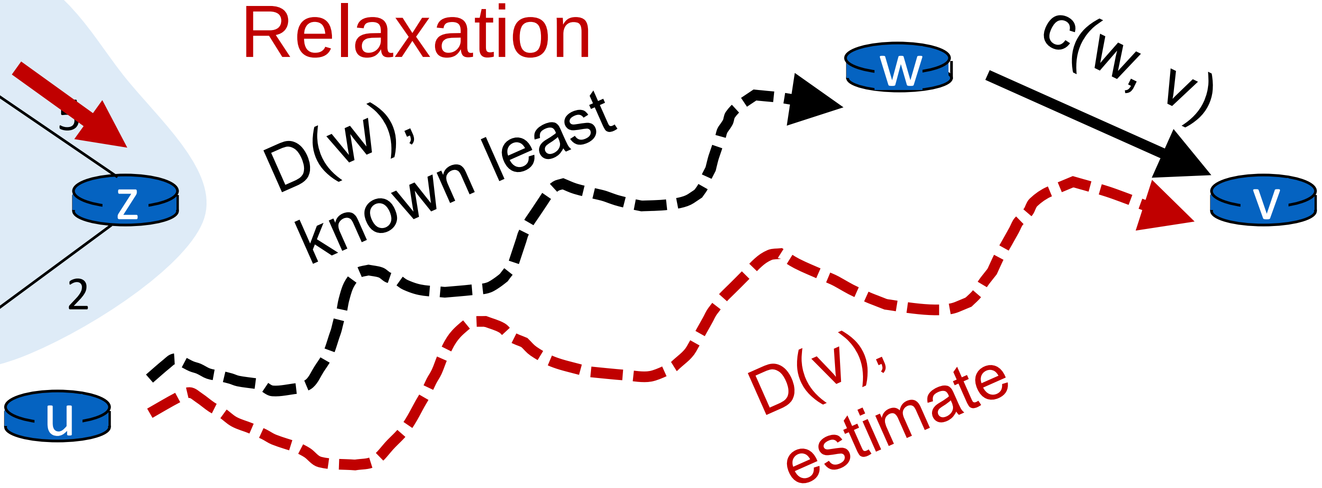
Q2. What computation?

Flooding



Dijkstra’s algorithm

Relaxation



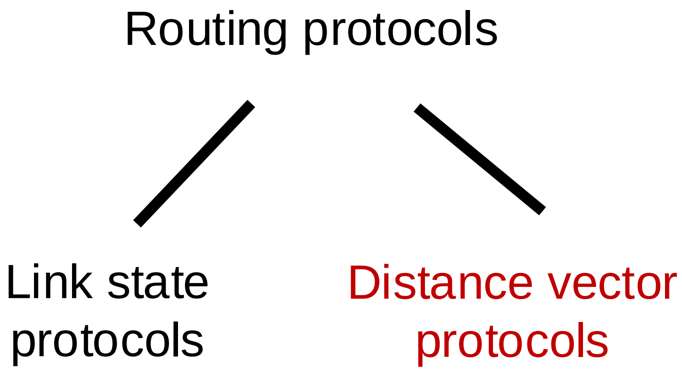
Distance Vector Protocols

Distance Vector Protocol

- Each router only exchanges a distance vector with its neighbors
  - Distance: how far the destination is
  - Vector: a value for each destination
- DVs are only exchanged between neighbors; not flooded
- Use incomplete view of graph derived from neighbors' distance vectors to compute the shortest paths

Q1. What info exchanged?

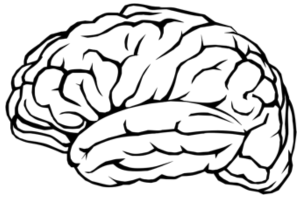
Q2. What computation?



Q1: Distance Vectors

- $D_x(y)$ = estimate of least cost from x to y
- Distance vector: $\mathbf{D}_x = [D_x(y): y \epsilon N]$
- Node x knows cost of edge to each neighbor v: $c(x,v)$
- Node x maintains $\mathbf{D}_x$
- Node x also maintains its neighbors' distance vectors
  - For each neighbor v, x maintains $\mathbf{D}_v = [D_v(y): y \epsilon N]$
- Nodes exchange distance vector periodically and whenever the local distance vector changes (e.g., link failure, cost changes)

Q2: Algorithm



Bellman-Ford algorithm

- Each node initializes its own distance vector (DV) to edge costs
- Each node sends its DVs to its neighbors
- When a node x receives new DV from a neighbor v, it updates its own DV using the Bellman-Ford equation:
- Given $d_x(y)$:= estimated cost of the least-cost path from x to y
- Update $d_x(y) = \min_v \{c(x,v) + d_v(y)\}$

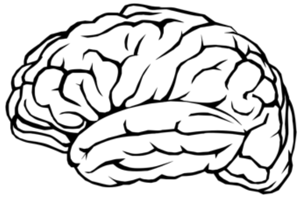
cost of path from neighbor v to destination y

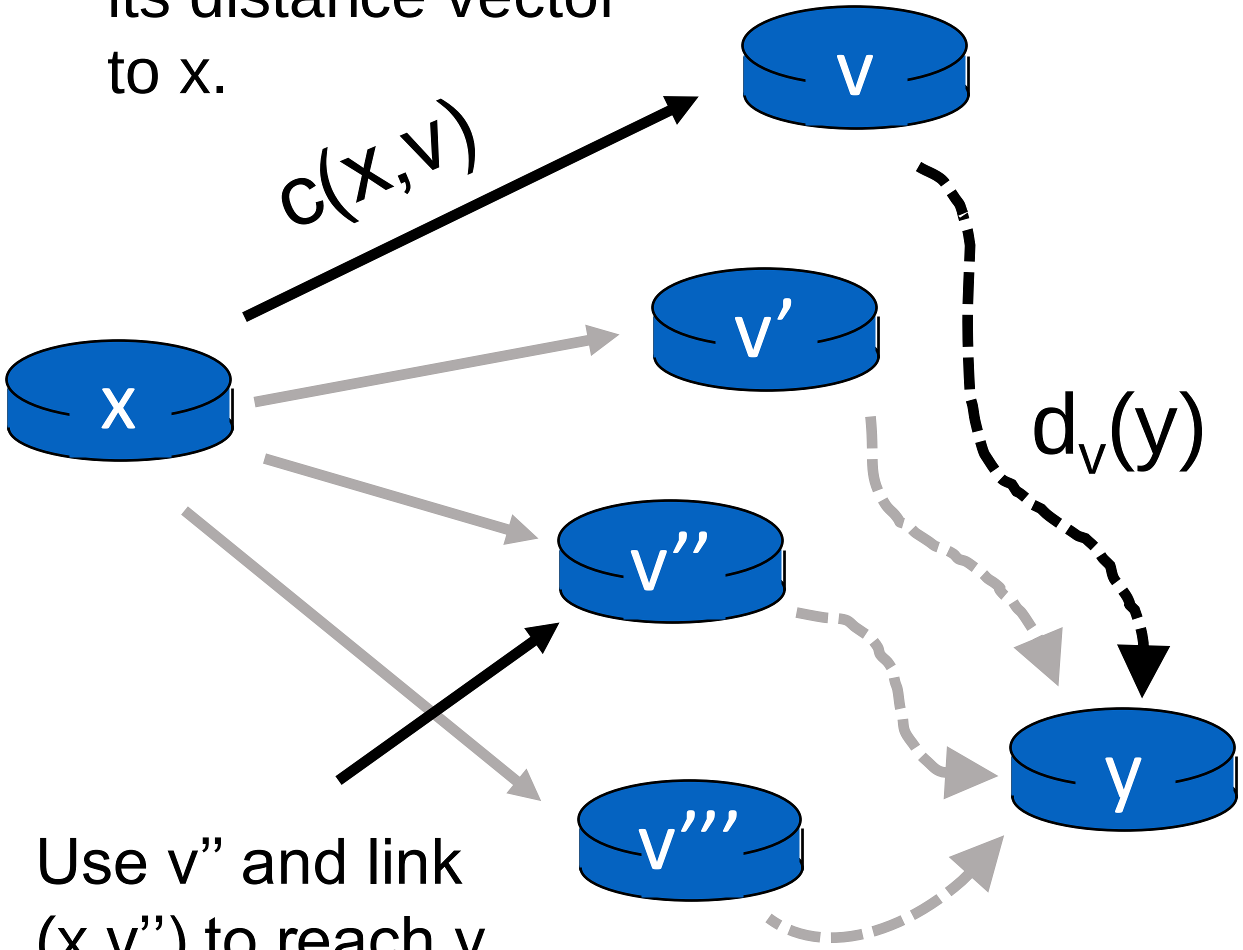
minimum taken over
all neighbors v of x

cost to reach neighbor v directly from x

Visualization

- Which neighbor v offers the current best path from x to y?
- Path through neighbor v has cost c(x,v) + $d_v(y)$
- Choose min-cost neighbor
- Remember min-cost neighbor as the one used to reach node y
- This neighbor determines the output port!





Neighbor v sends its distance vector to x.

Use v'' and link (x,v'') to reach y.

$$D_x(y) = \min\{c(x,y) + D_y(y),\ c(x,z) + D_z(y)\}$$
$$= \min\{2+0\ ,\ 7+1\} = 2$$

$$D_x(z) = \min\{c(x,y) + D_y(z),\ c(x,z) + D_z(z)\}$$
$$= \min\{2+1\ ,\ 7+0\} = 3$$

node x table

cost to

from	x	y	z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

cost to

from	x	y	z
x	0	2	3
y	2	0	1
z	7	1	0

cost to

from	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

node y table

cost to

from	x	y	z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

cost to

from	x	y	z
x	0	2	7
y	2	0	1
z	7	1	0

cost to

from	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

node z table

cost to

from	x	y	z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0

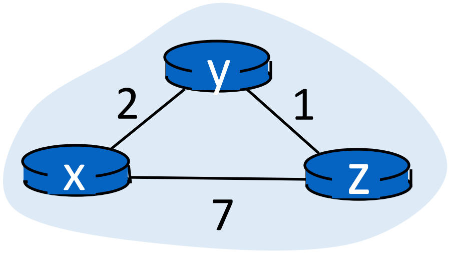
cost to

from	x	y	z
x	0	2	7
y	2	0	1
z	3	1	0

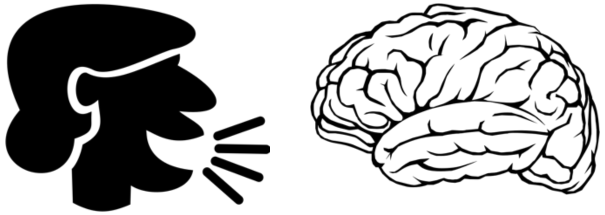
cost to

from	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

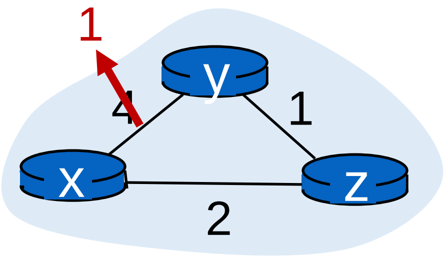
time



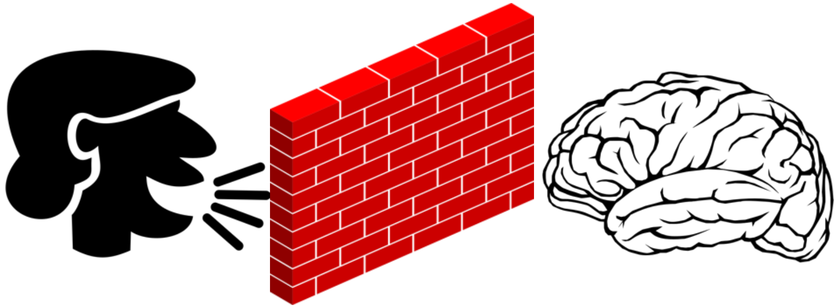
Good news travels fast



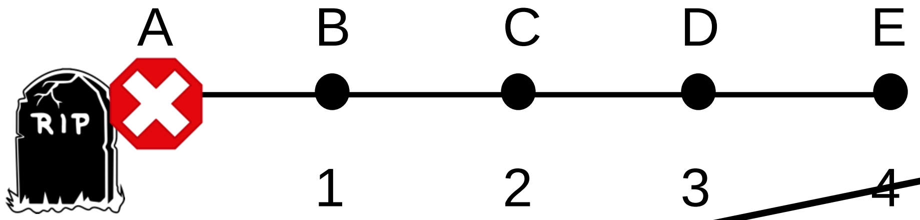
- Suppose the link cost reduces or a new better path becomes available in a network.
- The immediate neighbors of the change detect the better path immediately
- Since their DV changed, these nodes notify their neighbors immediately.
  - And those neighbors notify still more neighbors
  - … until the entire network knows to use the better path
- Good news travels fast through the network
- This is despite messages only being exchanged among neighbors



Bad news travels slowly



- If router goes down, could be a while before network realizes it.



1	2	3	4	Initially
3	2	3	4	After 1 exchange
3	4	3	4	After 2 exchanges
5	4	5	4	After 3 exchanges
5	6	5	6	After 4 exchanges
7	6	7	6	After 5 exchanges

B still thinks it can reach A through C… bad!

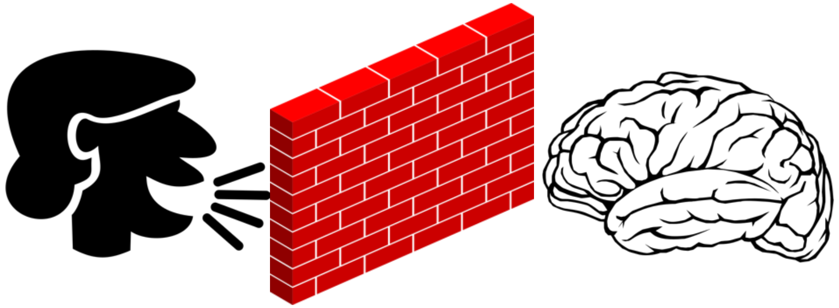
C thinks it can reach A through B… worse!

B, D think they can reach A through C… ugly!

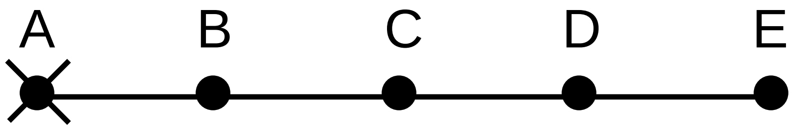
Count to infinity problem

etc… to infinity

Bad news travels slowly



- Reacting appropriately to bad news requires information that only other routers have. DV does not exchange sufficient info.



- B needs to know that C has no other path to A other than via B.
- DV does not exchange paths; just distances!
- Poisoned reverse: if X gets its route to Y via Z, then X will announce $d_X(Y) = \infty$ in its message to Z
  - Effect: Z won't use X to route to Y
  - However, this won't solve the problem in general (think why.)

Summary: Comparison of LS and DV

Link State Algorithms

- Nodes have full visibility into the network's graph
- Copious message exchange: each LSA is flooded over the whole network
- Robust to network changes and failures

OSPF

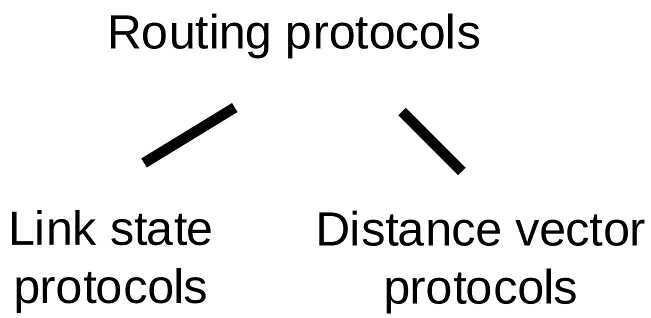
Open Shortest Path First
(v2 RFC 2328)

Distance Vector Algorithms

- Only distances and neighbors are visible
- Sparse message exchange: DVs are exchanged among neighbors only
- Brittle to router failures. Incorrect info may propagate all over net

EIGRP

Enhanced Interior Gateway Routing Protocol
(RFC 7868)



Routing protocols

Link state protocols

Distance vector protocols

Every router is aware of the existence of every other router.

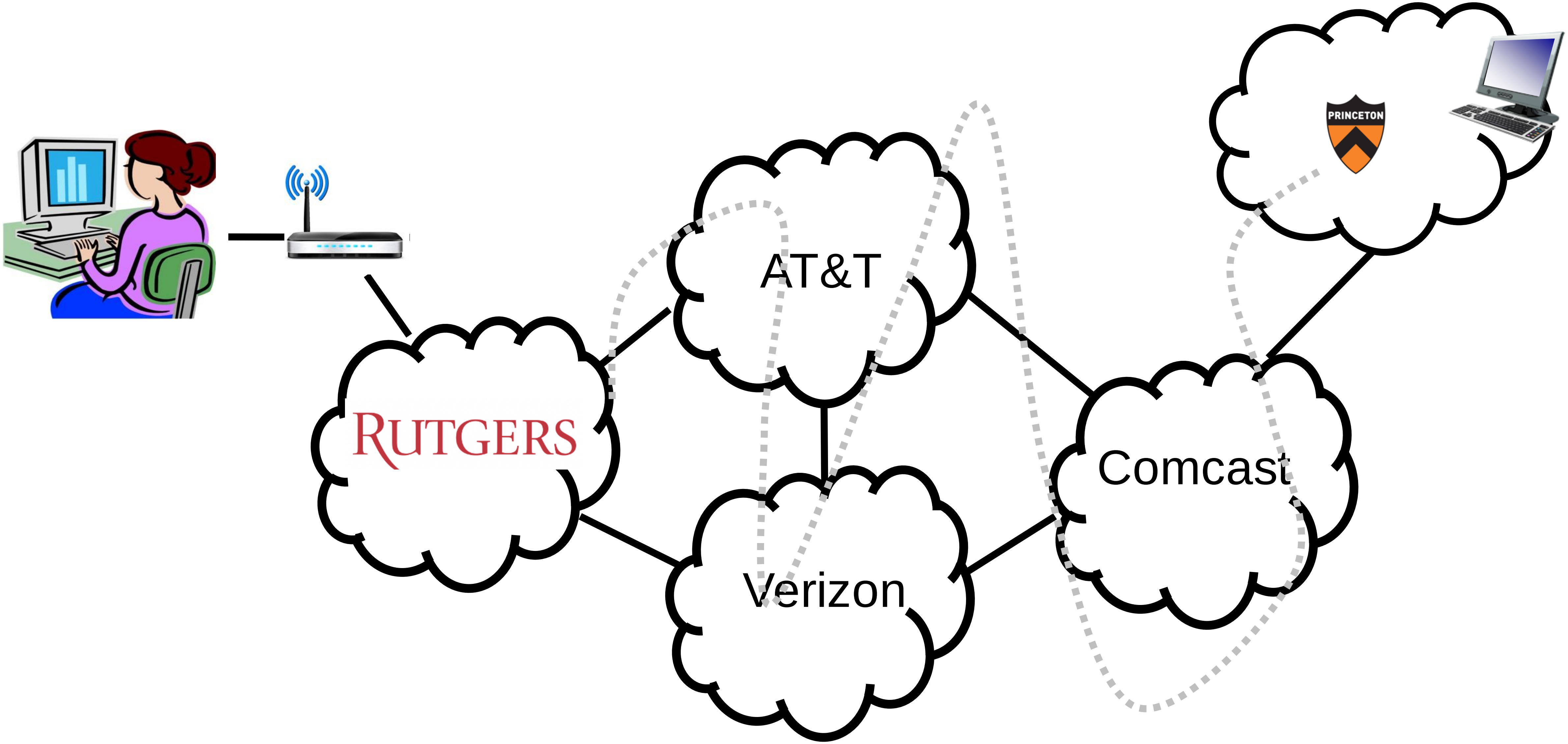
Messages reveal information on the full network (graph) structure.

Message exchange and forwarding tables scale with network size.

These assumptions/settings cannot work on the Internet.

Internet Routing

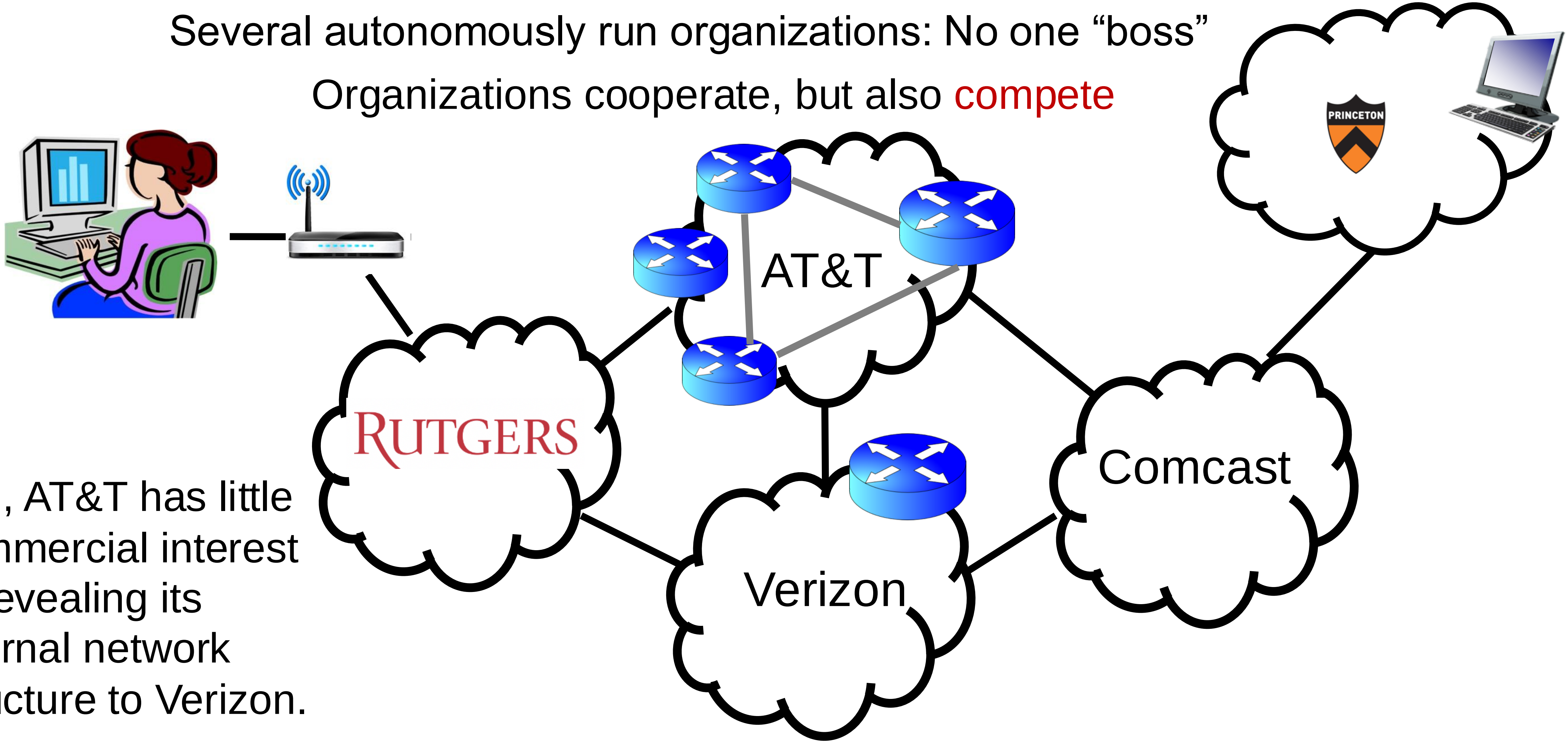
The Internet is a large federated network



The Internet is a large federated network

Several autonomously run organizations: No one “boss”

Organizations cooperate, but also compete

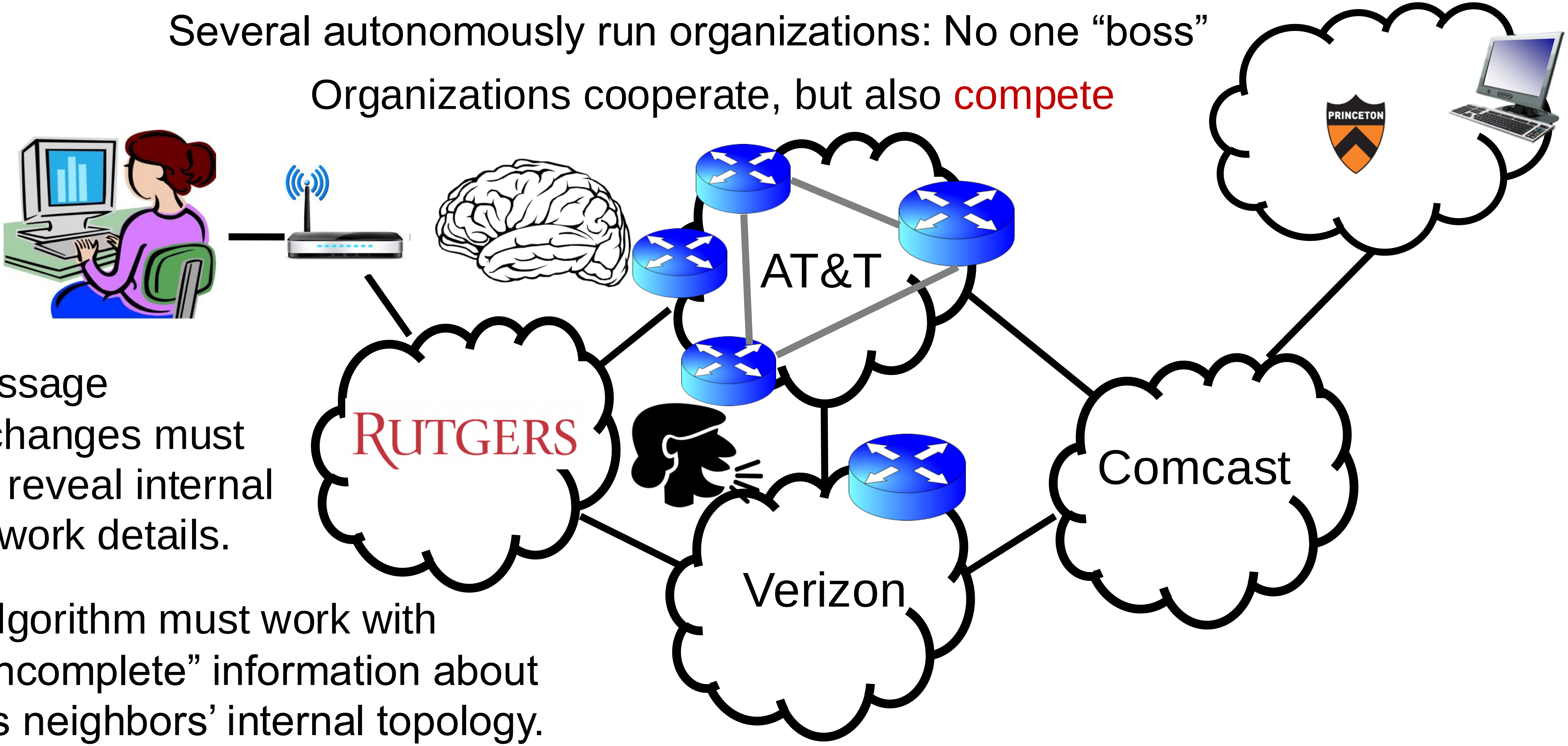


e.g., AT&T has little commercial interest in revealing its internal network structure to Verizon.

The Internet is a large federated network

Several autonomously run organizations: No one “boss”

Organizations cooperate, but also compete



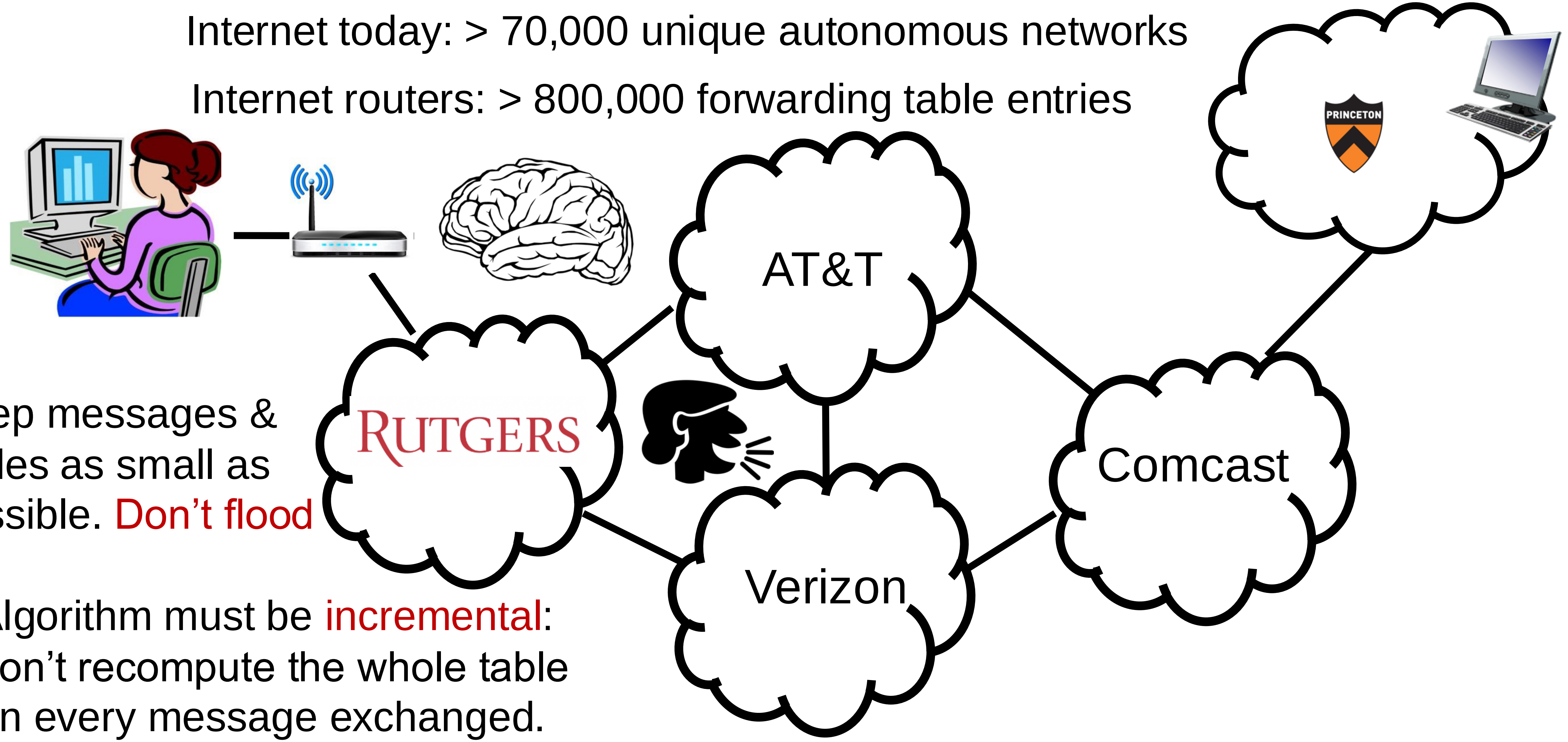
Message exchanges must not reveal internal network details.

Algorithm must work with “incomplete” information about its neighbors’ internal topology.

The Internet is a large federated network

Internet today: > 70,000 unique autonomous networks

Internet routers: > 800,000 forwarding table entries

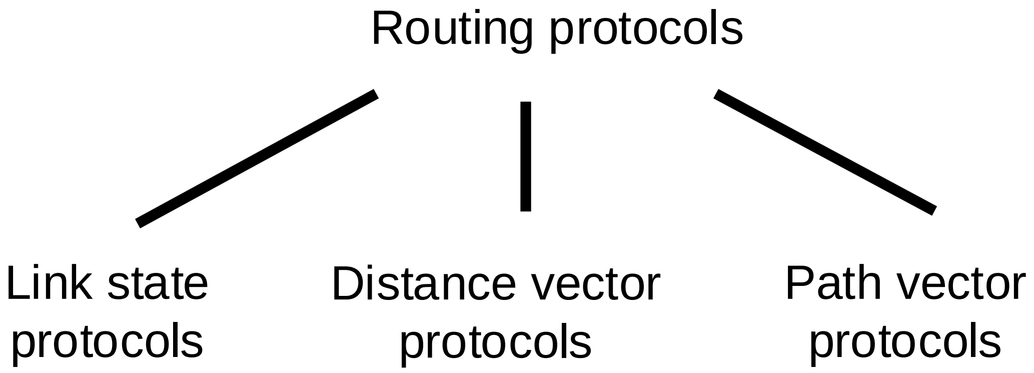


Keep messages & tables as small as possible. Don’t flood

Algorithm must be incremental: don’t recompute the whole table on every message exchanged.

Inter-domain Routing

- Routing approaches so far (LS + DV) are applicable within one autonomous system (AS), e.g., Rutgers
  - Called intra-domain routing protocols
- The Internet uses Border Gateway Protocol (BGP)
- All AS'es speak BGP. It is the glue that holds the Internet together
- BGP is a path vector protocol

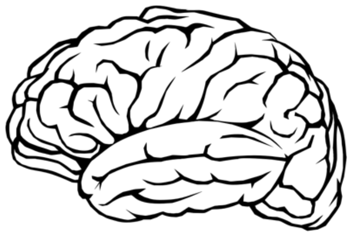

Routing protocols

Link state protocols

Distance vector protocols

Path vector protocols

Messages?


Algorithm?

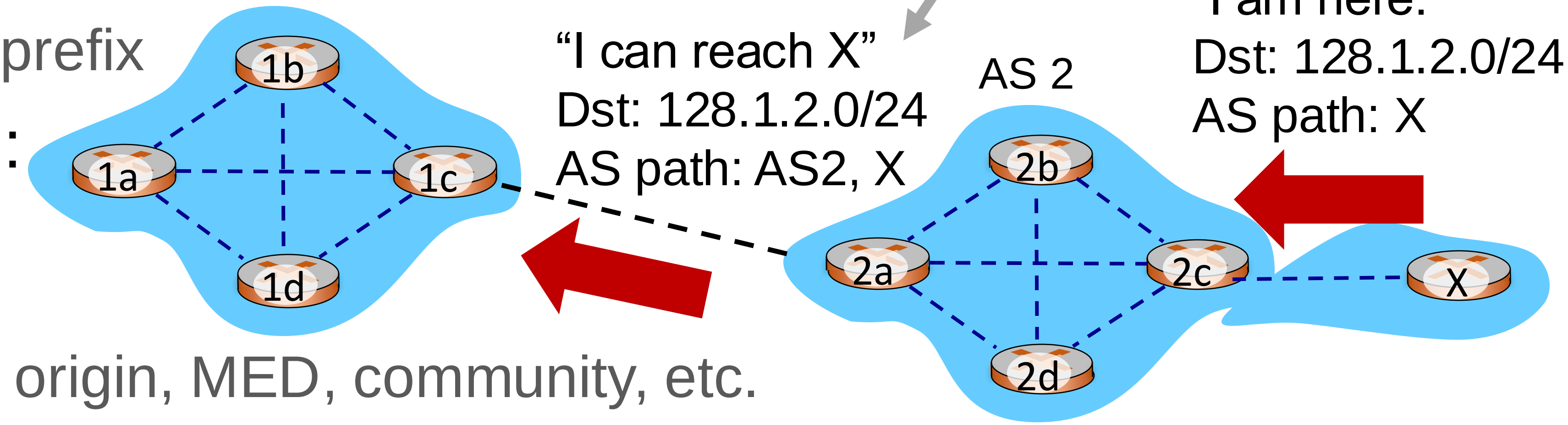
Q1. BGP Messages

Loop detection is easy
(no “count to infinity”)

Exchange paths: path vector

No link metrics, distances!

- Routing Announcements or Advertisements
  - “I am here” or “I can reach here”
  - Occur over a TCP connection (BGP session) between routers
- Route announcement = destination + attributes
  - Destination: IP prefix
- Route Attributes:
  - AS-level path
  - Next hop
  - Several others: origin, MED, community, etc.
- An AS promises to use advertised path to reach destination
- Only route changes are advertised after BGP session established

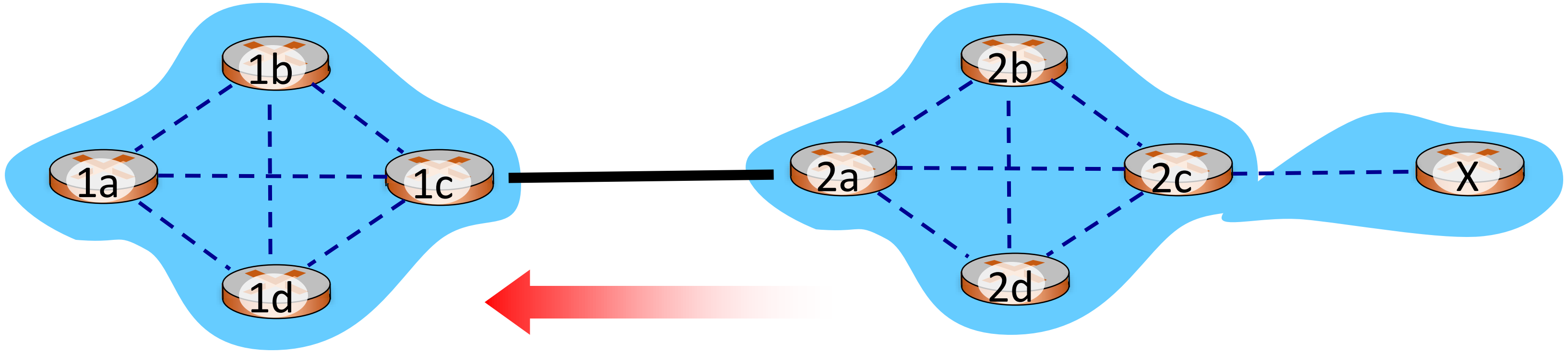


“I can reach X”
Dst: 128.1.2.0/24
AS path: AS2, X

“I am here.”
Dst: 128.1.2.0/24
AS path: X

Q1. Next Hop

- Next hop conceptually denotes the first router interface that begins the AS-level path
  - The meaning of this attribute is context-dependent
- In an announcement arriving from a different AS (eBGP), next hop is the router in the next AS which sent the announcement
  - Example: Next Hop of the eBGP announcement reaching 1c is 2a



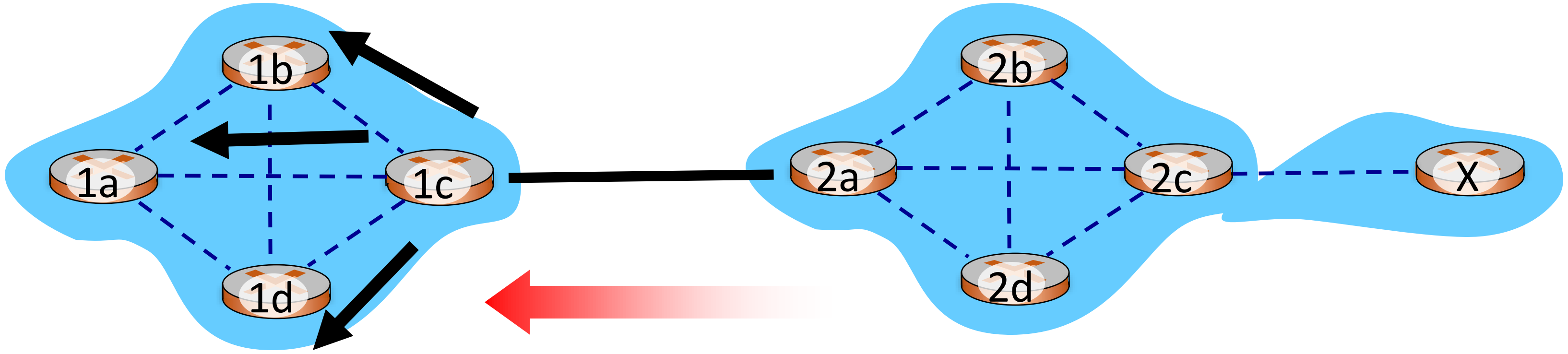
AS 1

eBGP announcement:
AS2, X

AS 2

Q1. Next Hop

- Suppose router 1c imports the path (more on this soon)
- Router 1c will propagate the announcement inside the AS using iBGP
- The next hop of this (iBGP) announcement is set to 1c
    - In particular, the next hop is an AS1 internal address

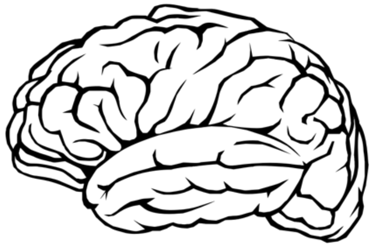


AS 1

eBGP announcement:
AS2, X

AS 2

Q2. The algorithm

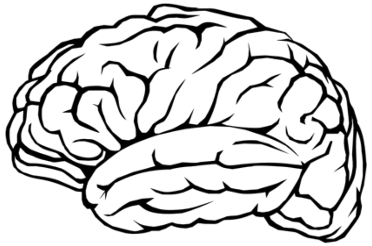


- A BGP router does *not* consider every routing advertisement it receives by default to make routing decisions!
  - An import policy determines whether a route is even considered a candidate
- Once imported, the router performs route selection
- A BGP router does *not* propagate its chosen path to a destination to all other AS'es by default!
  - An export policy determines whether a (chosen) path can be advertised to other AS'es and routers

Programmed by network operator

Policy considerations make BGP very different from intra-domain (LS / DV) protocols

Policies in BGP



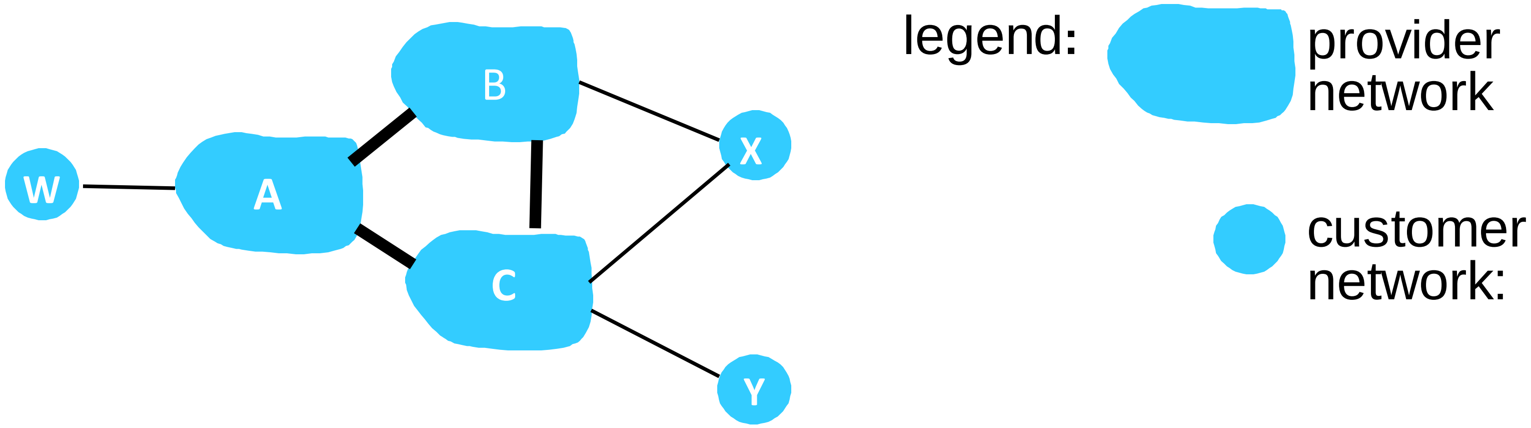
Policy arises from business relationships

- Customer-provider relationships:
  - E.g., Rutgers is a customer of AT&T

- Peer-peer relationships:
  - E.g., Verizon is a peer of AT&T

- Business relationships depend on where connectivity occurs
  - “Where”, also called a “point of presence” (PoP)
  - e.g., customers at one PoP but peers at another
  - Internet-eXchange Points (IXPs) are large PoPs where ISPs come together to connect with each other (often for free)

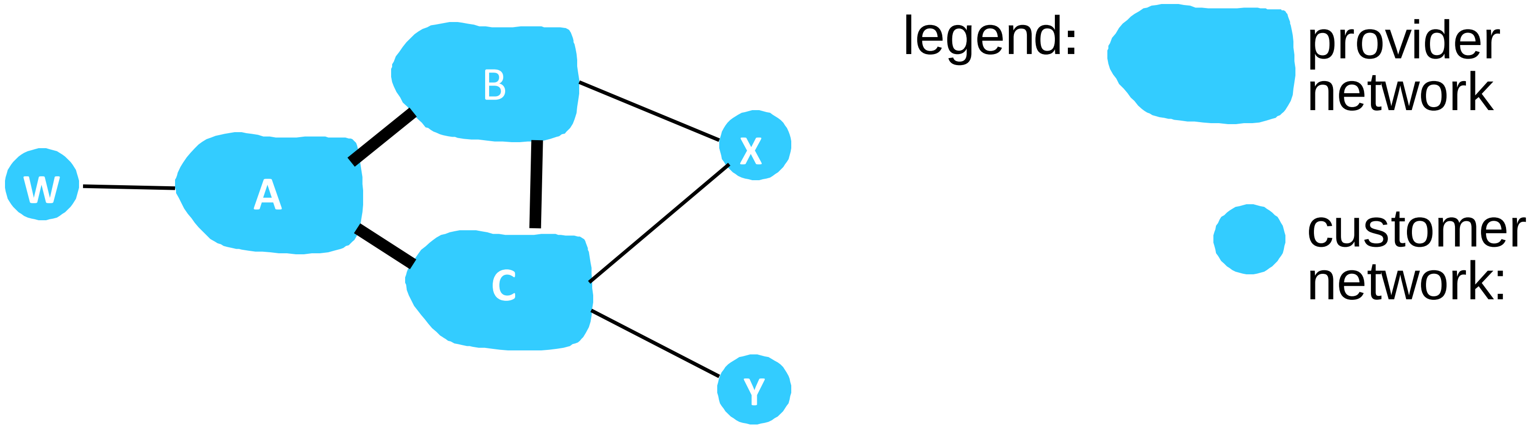
BGP Export Policy



Suppose an ISP only wants to route traffic to/from its customer networks (does not want to carry transit traffic between other ISPs)

- A,B,C are provider networks
- X,W,Y are customers (of provider networks)
- X is dual-homed: attached to two networks
- policy to enforce: X does not want to route from B to C via X
  - So, X will not announce to B a route to C

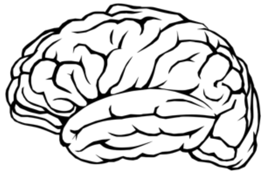
BGP Export Policy

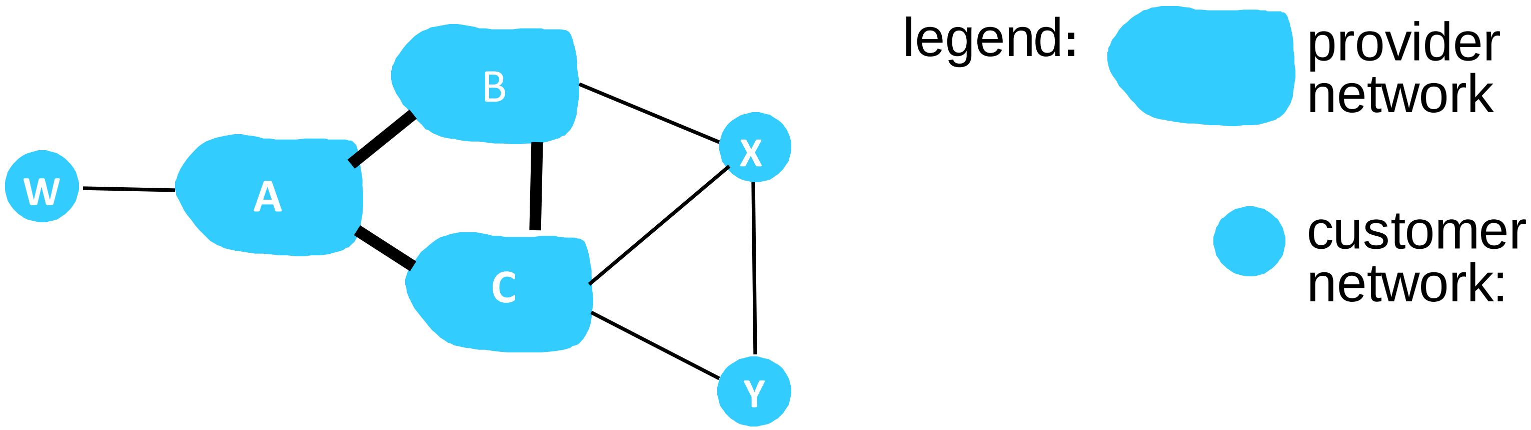


Suppose an ISP only wants to route traffic to/from its customer networks (does not want to carry transit traffic between other ISPs)

- A announces path Aw to B and to C
- B will not announce BAw to C:
  - B gets no “revenue” for routing CBAw, since none of C, A, w are B’s customers
- C will route CAw (not using B) to get to w

BGP Import Policy

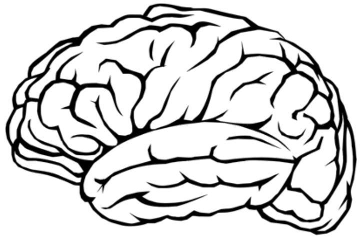




Suppose an ISP wants to minimize costs by avoiding routing through its providers when possible.

- Suppose C announces path Cy to x
- Further, y announces a direct path (“y”) to x
- Then x may choose not to import the path Cy to y since it has a peer path (“y”) towards y

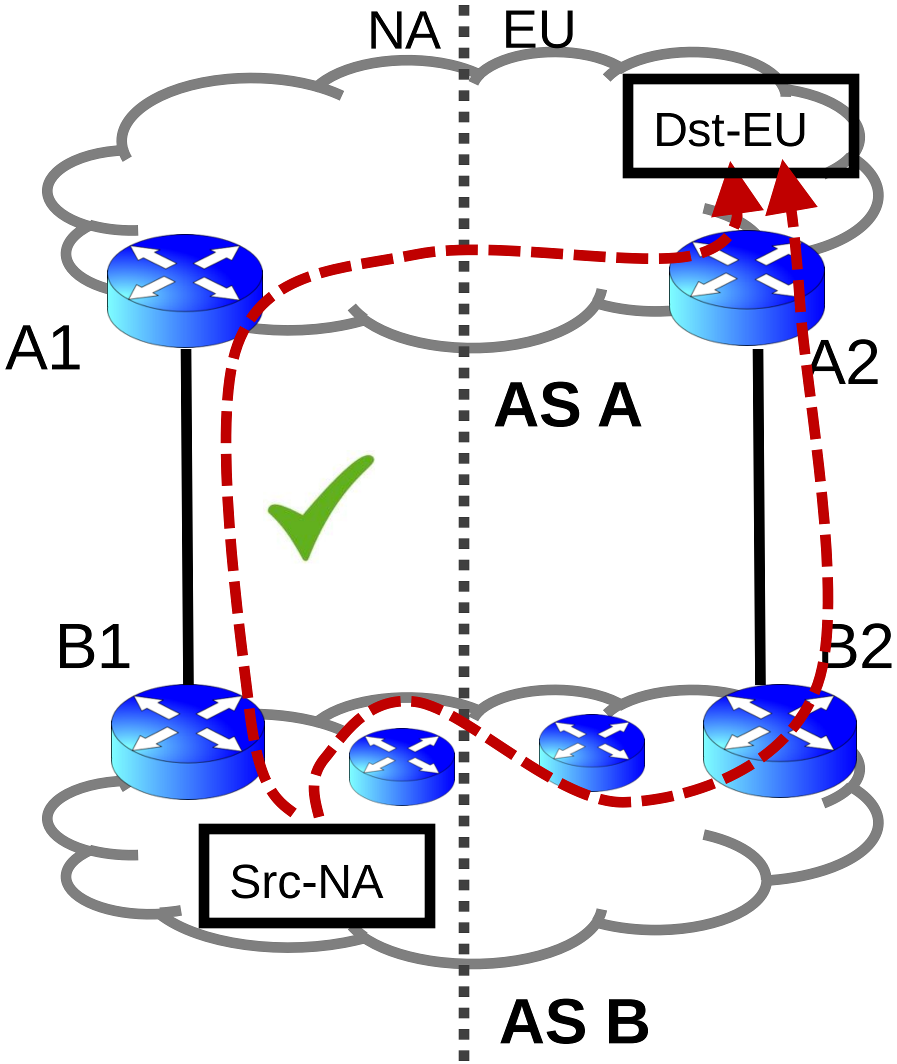
Q2. BGP Route Selection



- When a router imports more than one route to a destination IP prefix, it selects route based on:
  1. local preference value attribute (import policy decision -- set by network admin)
  2. shortest AS-PATH
  3. closest NEXT-HOP router
  4. Several additional criteria: You can read up on the full, complex, list of criteria, e.g., at https://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/13753-25.html

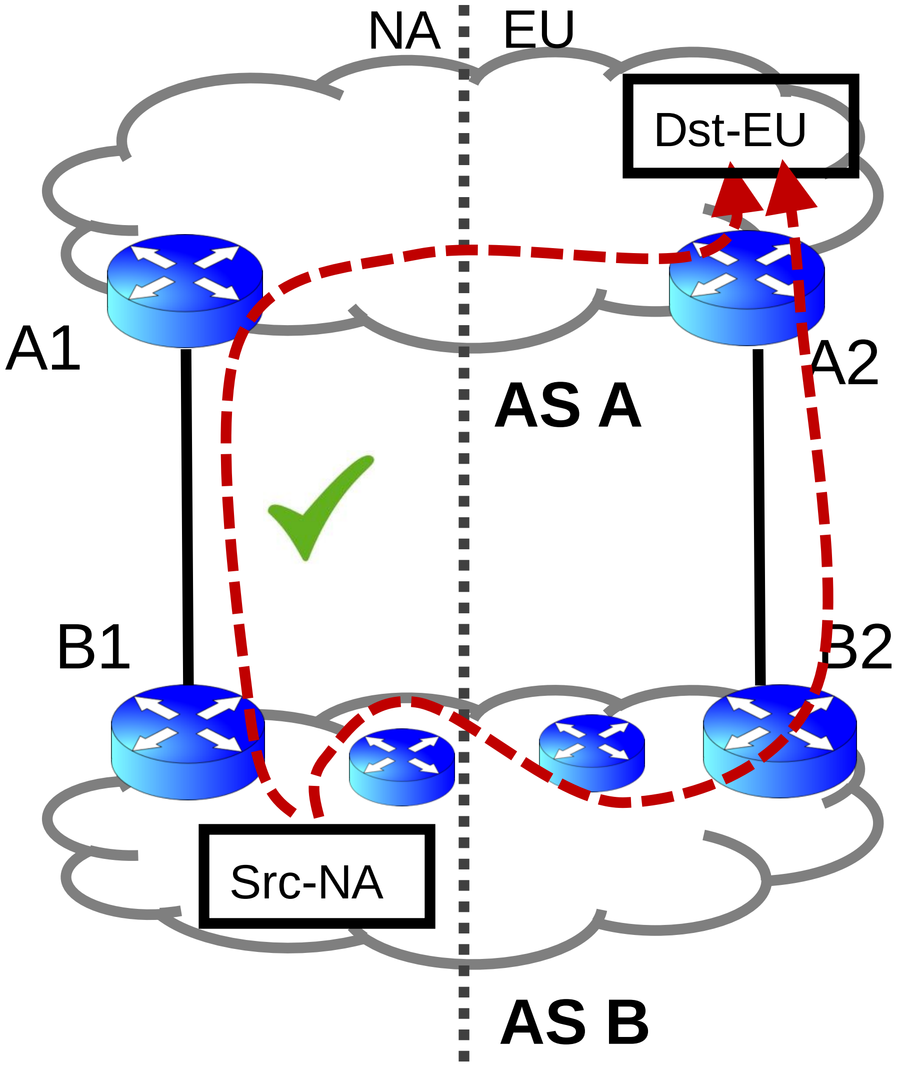
Example of route selection

- Suppose AS A and B are connected to each other both in North America (NA) and in Europe (EU)
- A source in NA wants to reach a destination in EU
- There are two paths available
  - *Assume* same local preference
  - Same AS path length
- Closest next hop-router: choose path via B1 rather than B2

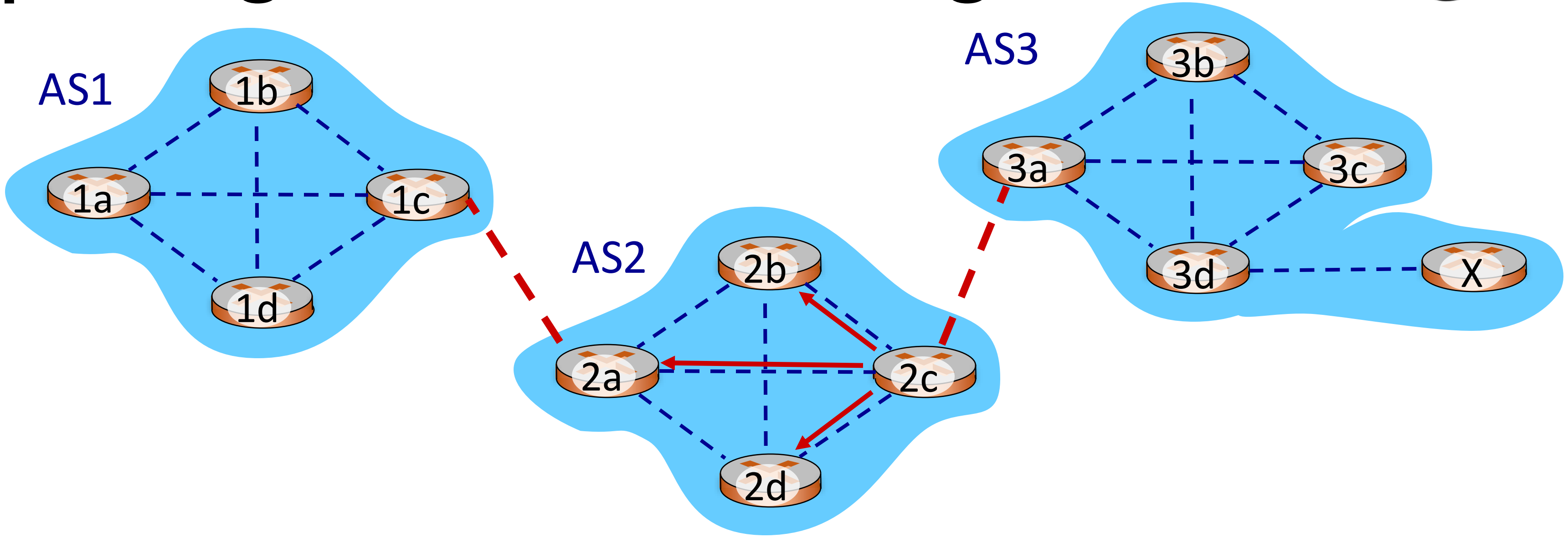


Example of route selection

- Choosing closest next-hop results in early exit routing
  - Try to exit the local AS as early as possible
  - Also called hot potato routing
- Reduce resource use within local AS
  - potentially at the expense of another AS

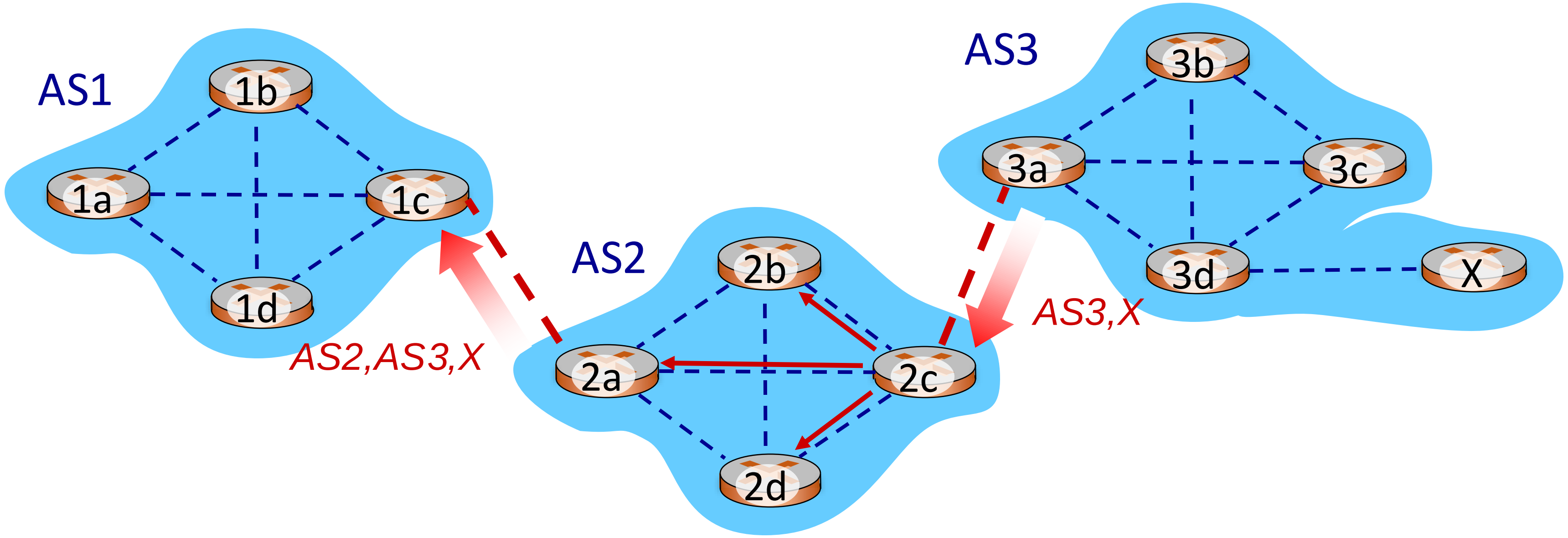


Computing the forwarding table



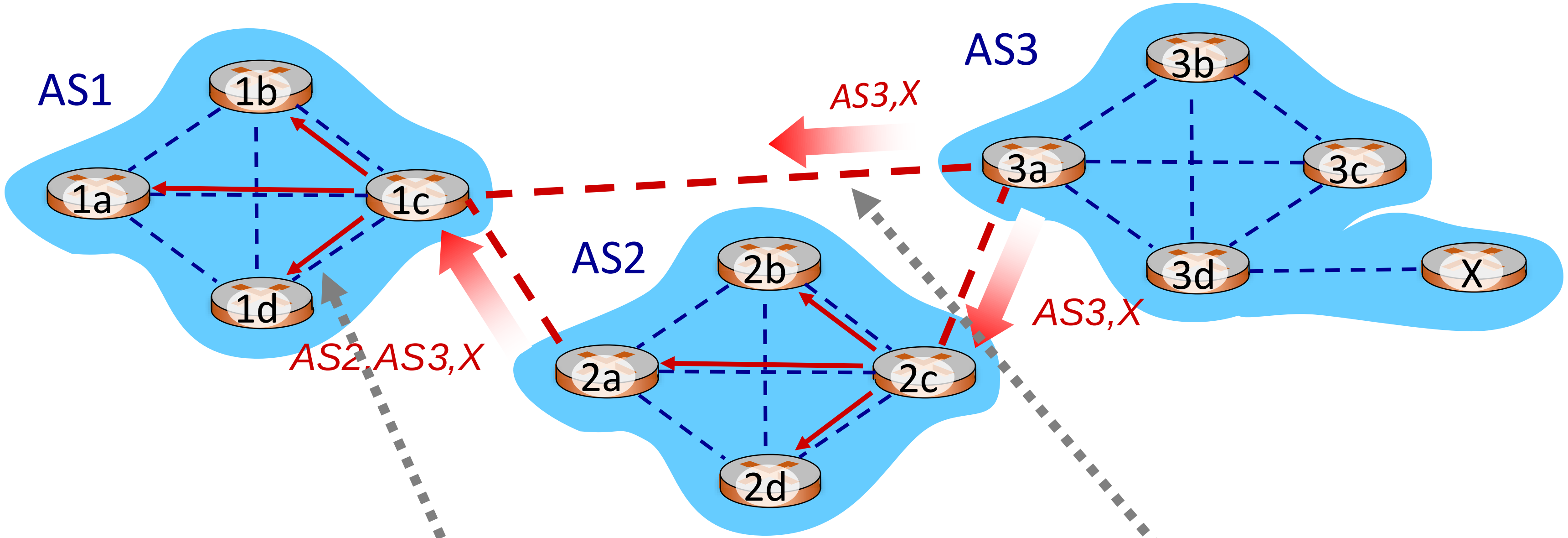
- Suppose a router in AS1 wants to forward a packet destined to external prefix X.
- How is the forwarding table entry for X at 1d computed?
- How is the forwarding table entry for X at 1c computed?

eBGP and iBGP announcements



- AS2 router 2c receives path announcement AS3,X (via eBGP) from AS3 router 3a
- Based on AS2 import policy, AS2 router 2c imports and selects path AS3,X, propagates (via iBGP) to all AS2 routers
- Based on AS2 export policy, AS2 router 2a announces (via eBGP) path AS2, AS3, X to AS1 router 1c

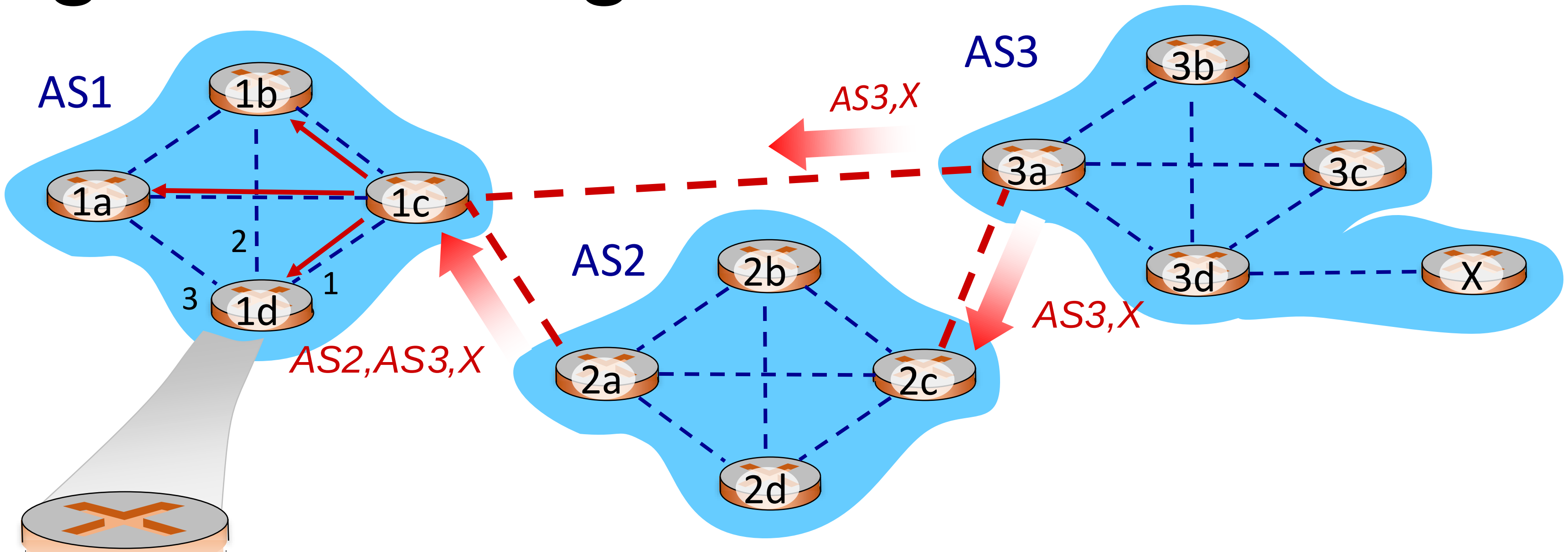
eBGP and iBGP announcements



A given router may learn about multiple paths to destination:

- AS1 gateway router 1c learns path AS2,AS3,X from 2a (next hop 2a)
- AS1 gateway router 1c learns path AS3,X from 3a (next hop 3a)
- Through BGP route selection process, AS1 gateway router 1c chooses path AS3,X, and announces path within AS1 via iBGP (next hop 1c)

Setting forwarding table entries



dest	Next-Hop
…	…
X	1c
…	…

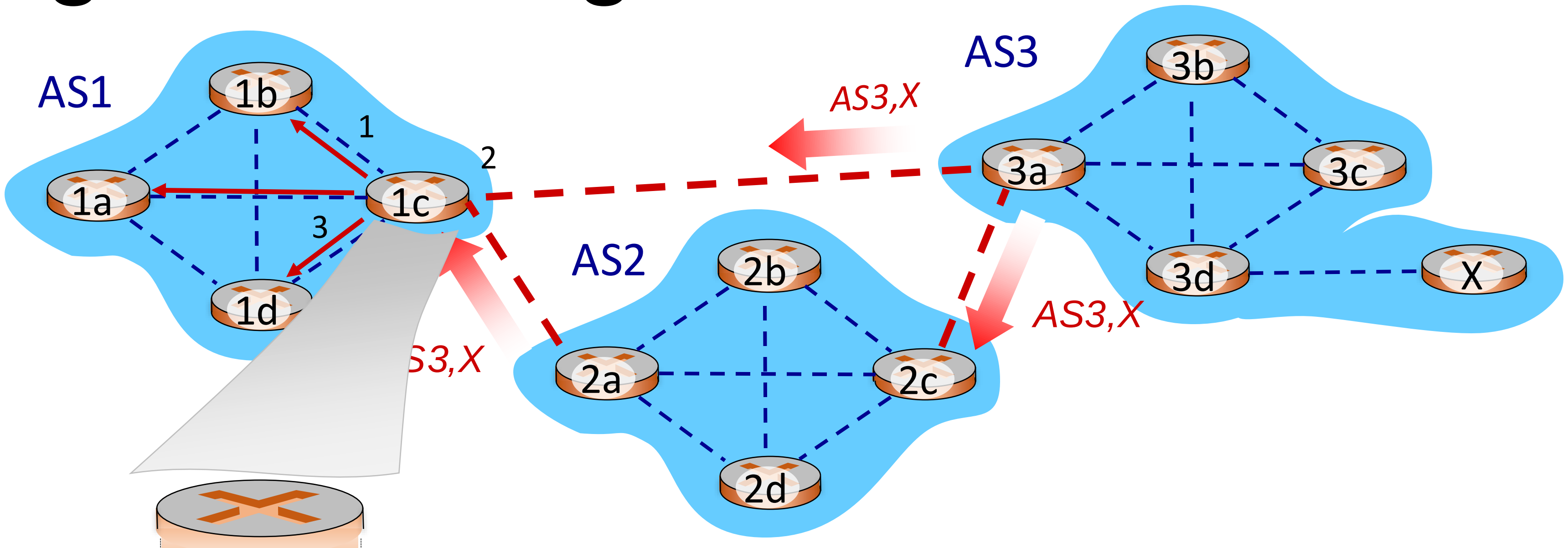
NH	interface
…	…
1c	1
…	…

IP Dst	Out port
X	1

Forwarding table

- recall: 1a, 1b, 1d learn about dest X via iBGP from next-hop 1c: “path to X goes through 1c”
- 1d: intra-domain routing: to get to 1c, forward over outgoing local interface 1

Setting forwarding table entries



dest	Next-Hop
…	…
X	3a
…	…

NH	interface
…	…
3a	2
…	…

IP Dst	Out port
X	2

Forwarding table

- recall: 1c learns about dest X via eBGP from next-hop 3a: “path to X goes through 3a”
- 1c: to get to link-local neighbor 3a, forward out interface 2

Summary: Inter-domain routing

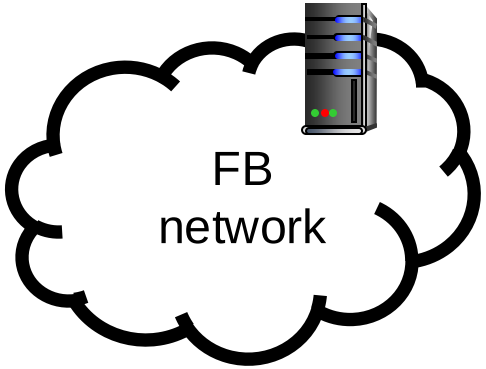
- Federation and scale introduce new requirements for routing on the Internet

- BGP is *the* protocol that handles Internet routing

- Path vector: exchange paths to a destination with attributes

- Policy-based import of routes, route selection, and export

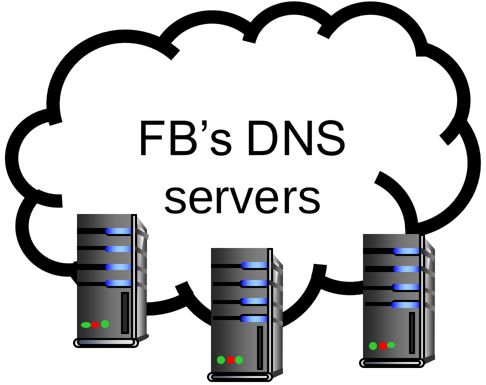
BGP’s impact: October ’21 FB++ outage

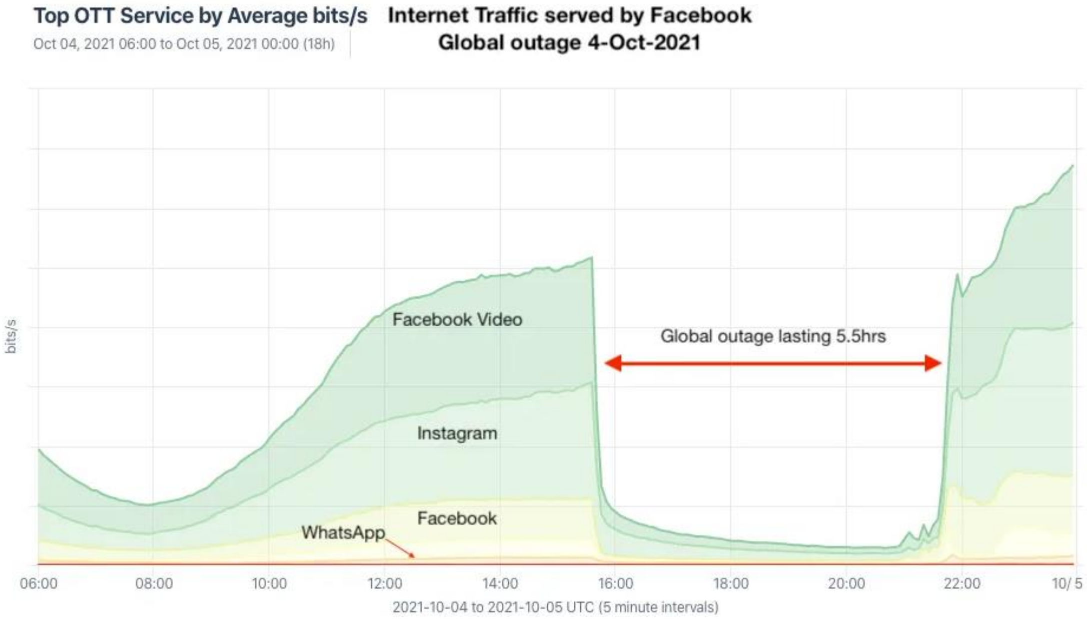


BGP route
withdrawal:
“I can’t reach FB
anymore”

Rest of the
Internet

BGP route
withdrawal: don’t
use me to get to FB




Top OTT Service by Average bits/s
Oct 04, 2021 06:00 to Oct 05, 2021 00:00 (18h)
Internet Traffic served by Facebook
Global outage 4-Oct-2021
bits/s
Facebook Video
Instagram
Facebook
WhatsApp
Global outage lasting 5.5hrs
06:00
08:00
10:00
12:00
14:00
16:00
18:00
20:00
22:00
10/5
2021-10-04 to 2021-10-05 UTC (5 minute intervals)

No remote access (no more reachability due to BGP withdrawal of DC and DNS servers)

Restricted physical access (prox can’t verify, can’t access prox server)

https://engineering.fb.com/2021/10/05/networking-traffic/outage-details/

By Doug Madory - https://www.kentik.com/blog/facebooks-historic-outage-explained/, CC BY 4.0, https://commons.wikimedia.org/w/index.php?curid=110816752