CSCI-1680
Network Layer: Inter-domain Routing

Nick DeMarinis

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?

From S:

| Dest.	Cost
A	2
B	3
C	5
E	2

Administrivia

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

- IP: Due next Thursday, October 17
  - See Implementation Start Guide, other resources
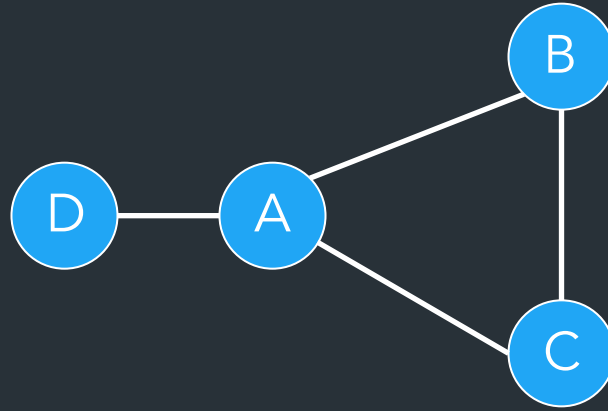 - Do not leave this until the last minute

Topics for today

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

- Inter-domain routing: BGP

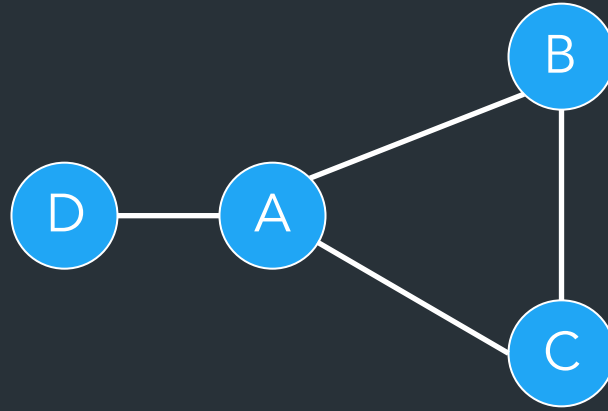# What happens when the D-A link fails?
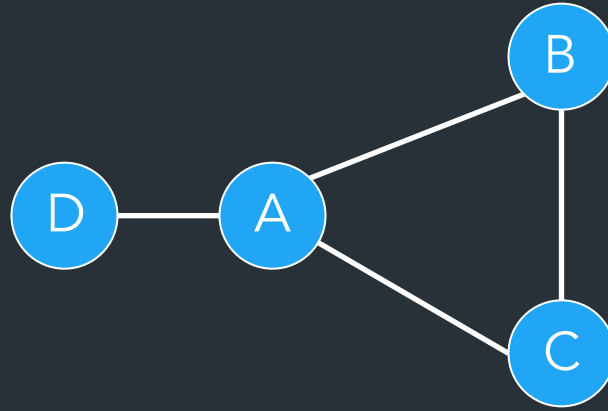
What happens when the D-A link fails?

=> "Count to Infinity" problem

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

Can we avoid loops?

- Does IP TTL help? Nope.
- Simple approach: consider a small cost n (e.g., 16) to be infinity

Can we avoid loops?

- Does IP TTL help? Nope.
- Simple approach: consider a small cost n (e.g., 16) to be infinity

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…

RFC1058 (1988): The original RIP standard*

RFC 1058   Routing Information Protocol   June 1988

supply the information that is needed to do routing.

1.1. **Limitations of the protocol**

This protocol does not solve every possible routing problem. As
mentioned above, it is primary intended for use as an IGP, in
reasonably homogeneous networks of moderate size. In addition, the
following specific limitations should be mentioned:

*: Obsoleted by RFC2453 (don't use RFC 1058 for the project, Use RFC 2453 instead)

One strategy: Split Horizon

- When sending updates to node A, don't include routes you learned from A

- Prevents B and C from sending cost 2 to A

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

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

=> Does it help?

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

$\Rightarrow$ Does it help? Not completely.

=> A common convention, might reduce time to converge, but overall hard to see effect vs. split horizon
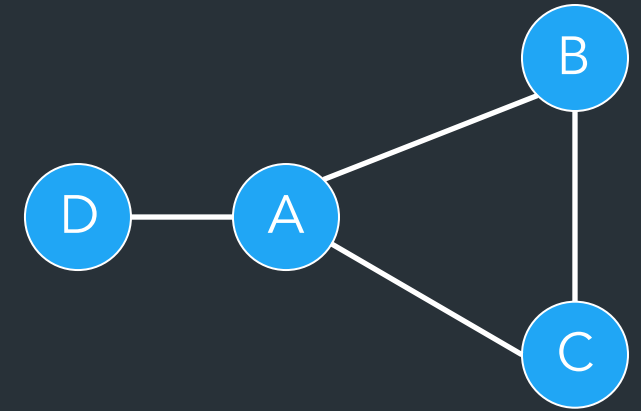
Even with split horizon + poison reverse,
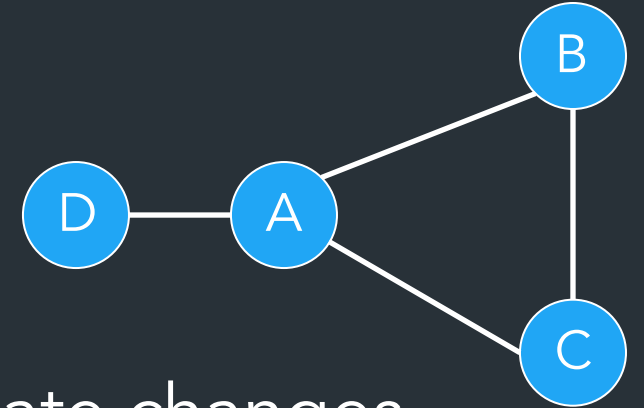can still create loops with >2 nodes!
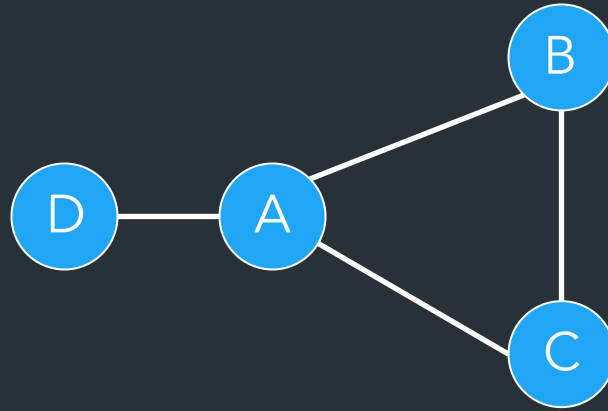
Even with split horizon + poison reverse,
can still create loops with >2 nodes!
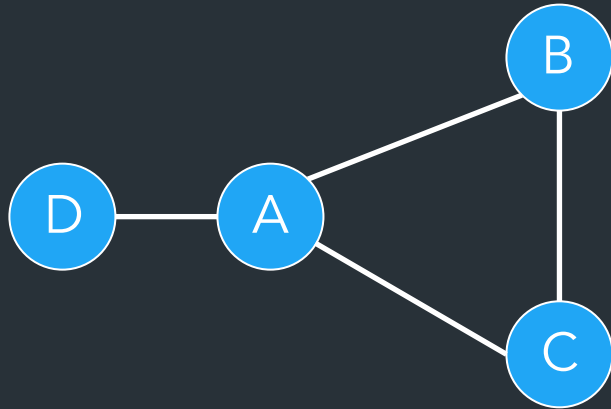
What else can we do?

Even with split horizon + poison reverse,
can still create loops with >2 nodes!

What else can we do?

- <u>Triggered updates</u>: send update as soon as link state changes
- <u>Hold down</u>: delay using new routes for certain time, affects convergence time

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?

From [RFC2453](#), RIP v2 (1998):

3.2 **Limitations of the Protocol**

This protocol does not solve every possible routing problem. As
mentioned above, it is primary intended for use as an IGP in networks
of moderate size. In addition, the following specific limitations
are be mentioned:

- The protocol is limited to networks whose longest path (the
  network's diameter) is 15 hops. The designers believe that the
  basic protocol design is inappropriate for larger networks. Note
  that this statement of the limit assumes that a cost of 1 is used
  for each network. This is the way RIP is normally configured. If
  the system administrator chooses to use larger costs, the upper
  bound of 15 can easily become a problem.

- The protocol depends upon "counting to infinity" to resolve certain
  unusual situations. (This will be explained in the next section.)
  If the system of networks has several hundred networks, and a
  routing loop was formed involving all of them, the resolution of
  the loop would require either much time (if the frequency of
  routing updates were limited) or bandwidth (if updates were sent
  whenever changes were detected). Such a loop would consume a large

Link State Routing

Link State Routing: The Alternative

Example: OSPF

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

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

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
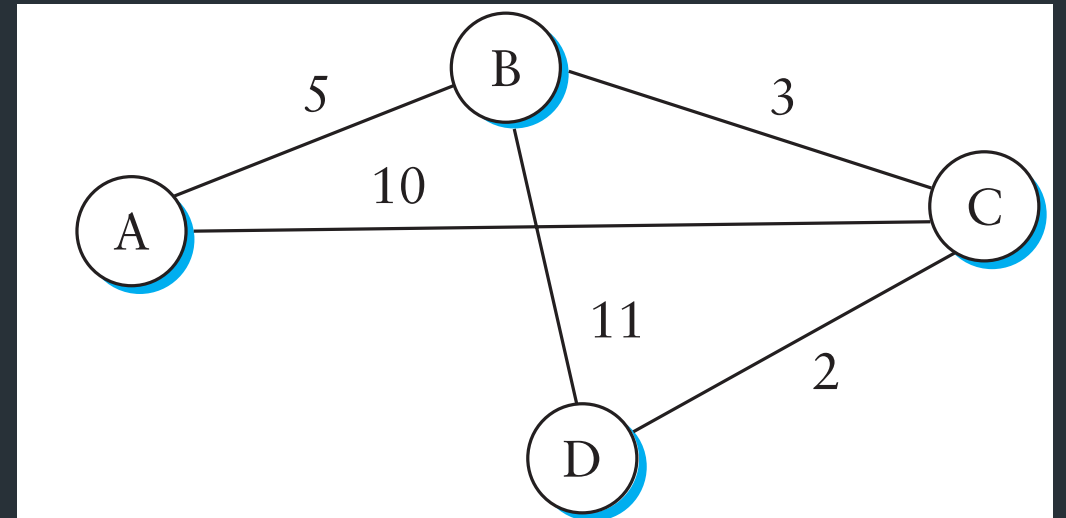
Tradeoffs?

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

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

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
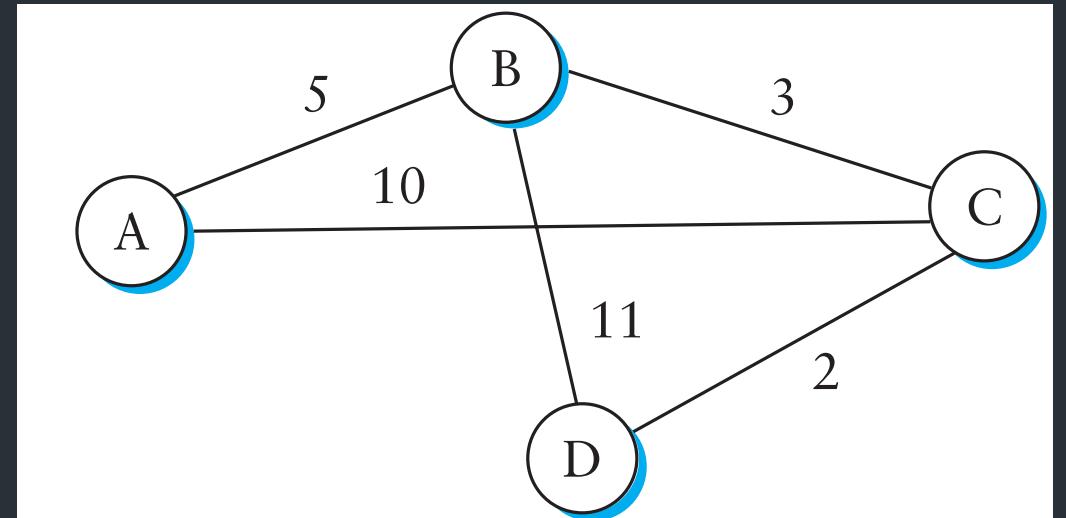
Tradeoffs?

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

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

Examples (and complexity)

Algorithm	Method	Pages in RFC
RIP v2 (RFC 2453)	Distance Vector	38

Examples (and complexity)

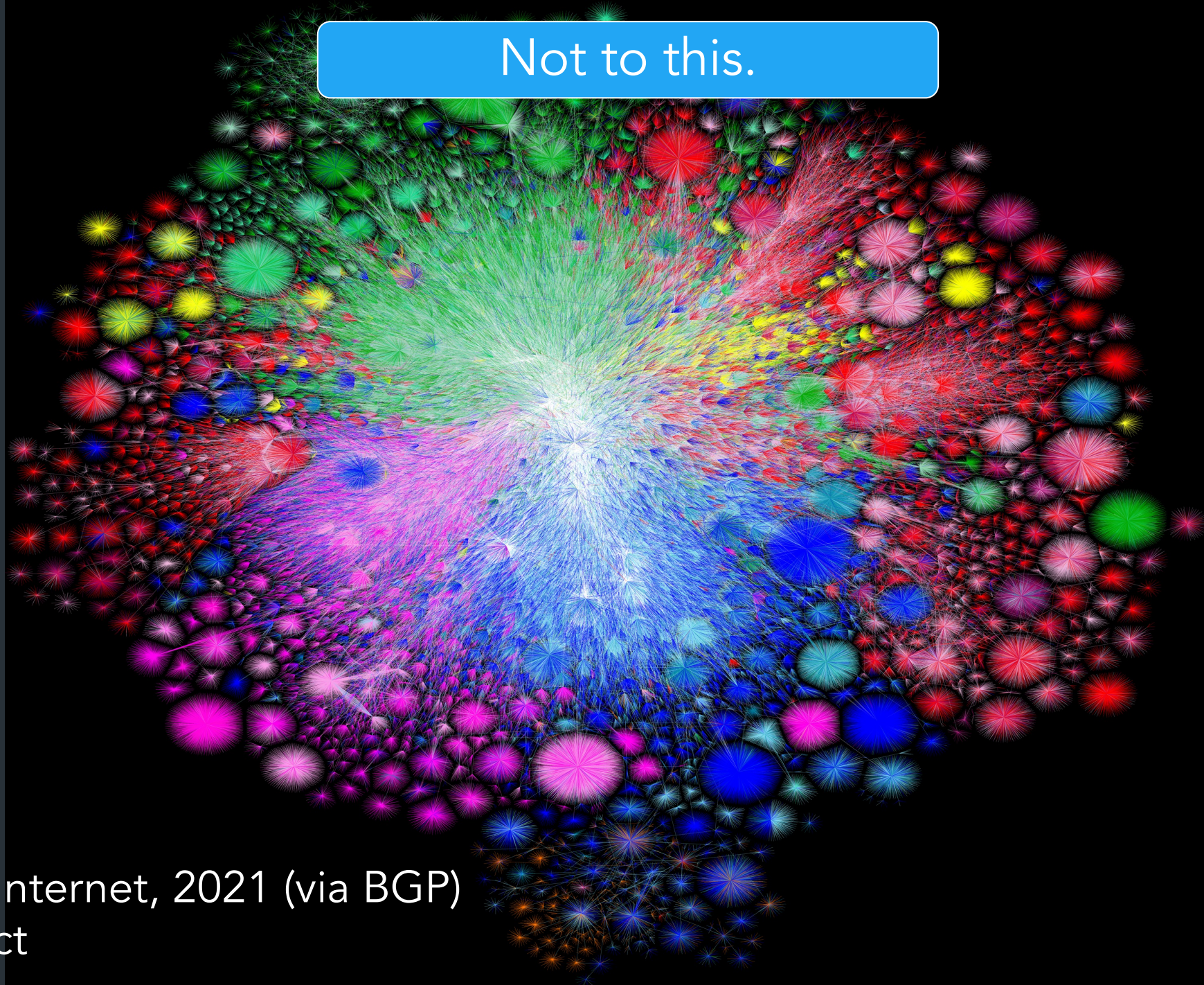| Algorithm	Method	Pages in RFC
RIP v2 (RFC 2453)	Distance Vector	38
OSPF (RFC 2328)	Link-State	244
IS-IS (OSI)	Link-State	210

OSPF: Open Shortest Path First
IS-IS: OSI standard similar to OSPF, doesn't depend on IP

So why not just use OSPF everywhere?

Does it scale?

Not to this.

Map of the Internet, 2021 (via BGP)
OPTE project

Why not?

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

⇒ More a policy problem than a technical problem

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 advertise some set of IP prefixes to the Internet

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

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

= > Generally, large networks 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	AS Info	Graph v4	Graph v6	Prefixes v4	Prefixes v6	Peers v4	Peers v6
olkit Home	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	🇺🇸

Sidebar links (partially visible):
- refix Report
- eer Report
- Traceroute
- nge Report
- Routes
- Report
- rigin Routes
- eport
- st Report
- Statistics
- g Glass
- k Tools App
- Pv6 Tunnel
- ertification
- rogress
- Native
- t Us

BGP: A Path Vector Protocol

Distance vector + extra information

eg. "I can reach prefix 128.148.0.0/16 through

ASes 44444 3356 14325 11078"

BGP: A Path Vector Protocol

Distance vector + 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 path to *advertise* to neighbors

Fun fact: loops are easy to avoid…

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

What would a loop look like?

BGP: A Path Vector Protocol

Distance vector + 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 path to *advertise* to neighbors

⇒ BGP routers look at path to decide how to "propagate" route, based on policy
⇒ Can easily avoid loops!

BGP Implications

- No loops!
- Not all ASs know all paths
- Reachability not guaranteed
  - Decentralized combination of policies

- Scaling
  - 74K ASs
  - 959K+ prefixes
  - ASs with one prefix: 25K
  - Most prefixes by one AS: 10008 (Uninet S.A. de C.V., MX)

Source: cidr-report 18Oct2022

Why study BGP?

BGP is what makes the Internet run.

Lots of problems...

Explainer

Facebook outage: what went wrong and why did it take so long to fix after social platform went down?

RYAN SINGEL   SECURITY   FEB 25, 2008 10:37 AM

Pakistan's Accidental YouTube Re-Routing Exposes Trust Flaw in Net

TECHNOLOGY
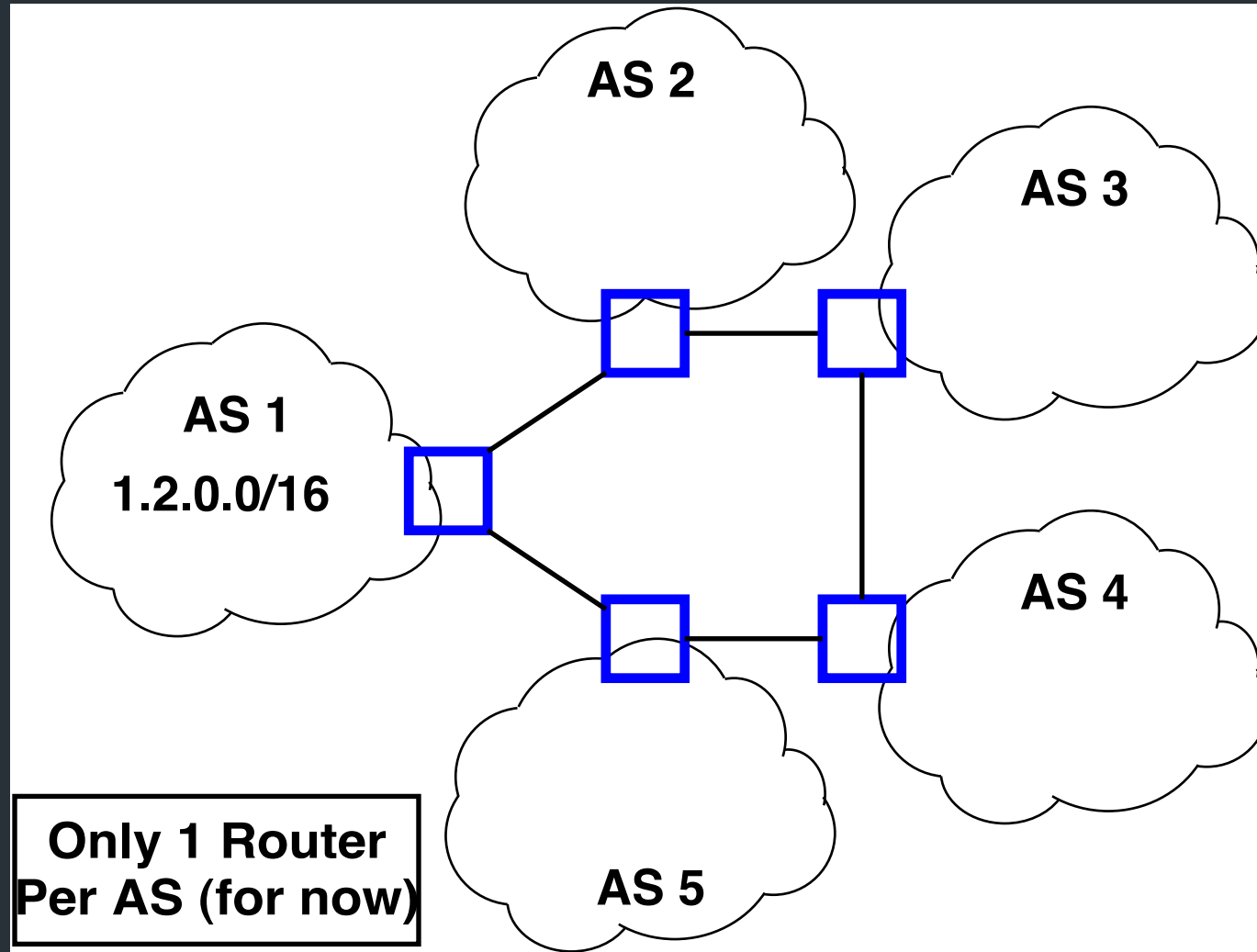
How Was Egypt's Internet Access Shut Off?

How Russia Took Over Ukraine's Internet in Occupied Territories

By Adam Satariano and
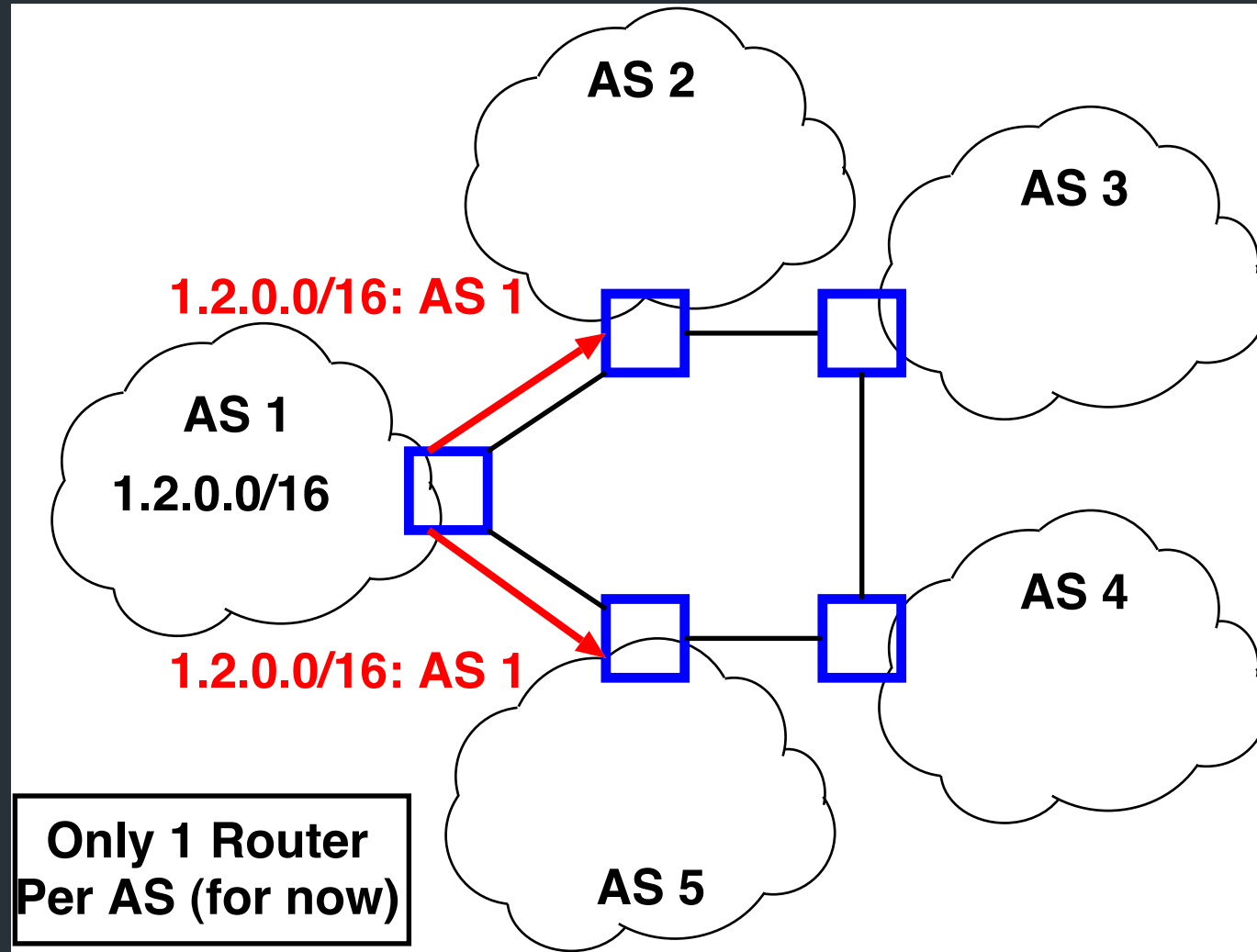Graphics by Scott Reinhard
Aug. 9, 2022

A Network Operations Center (NOC)

BGP Example

BGP Example
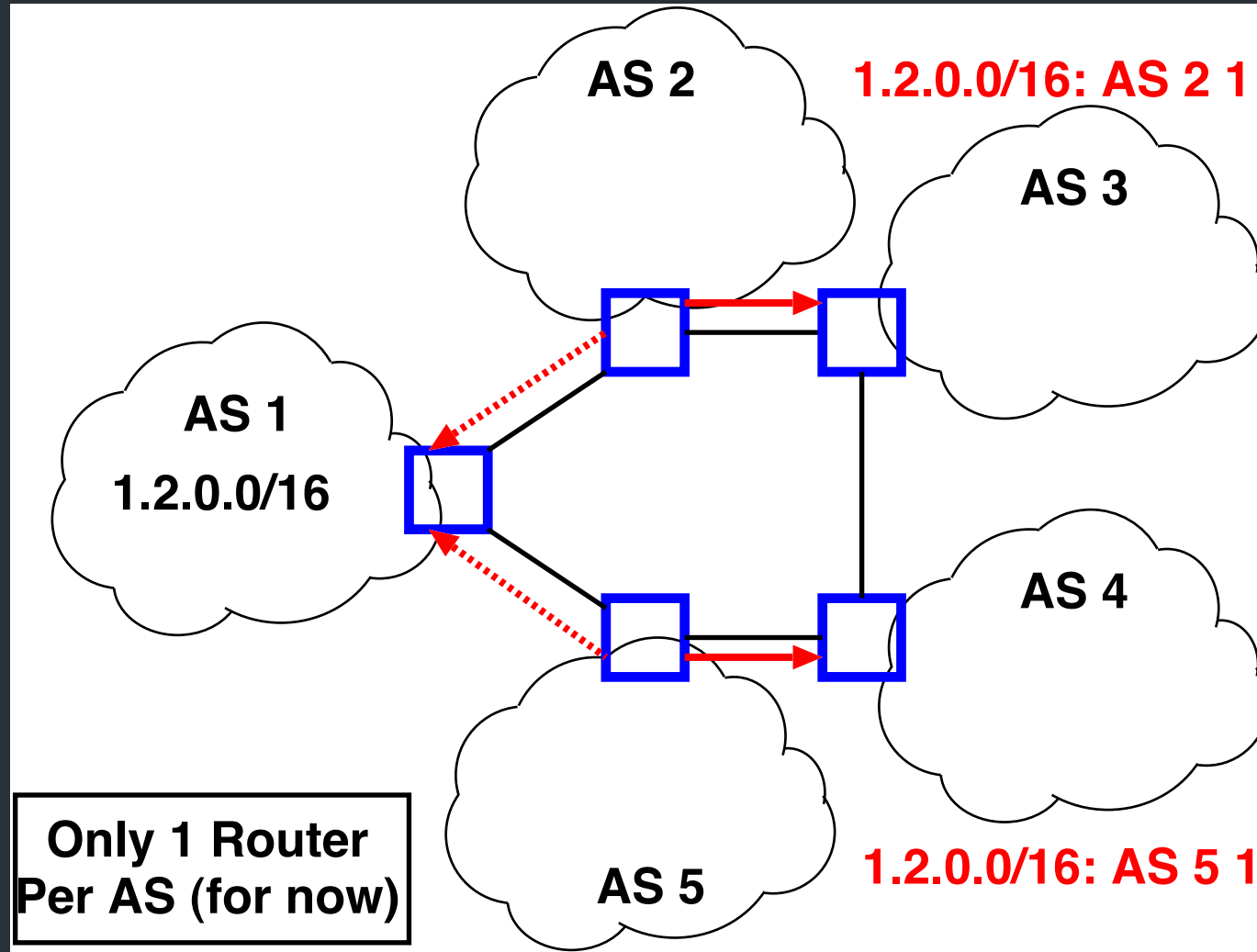
BGP Example
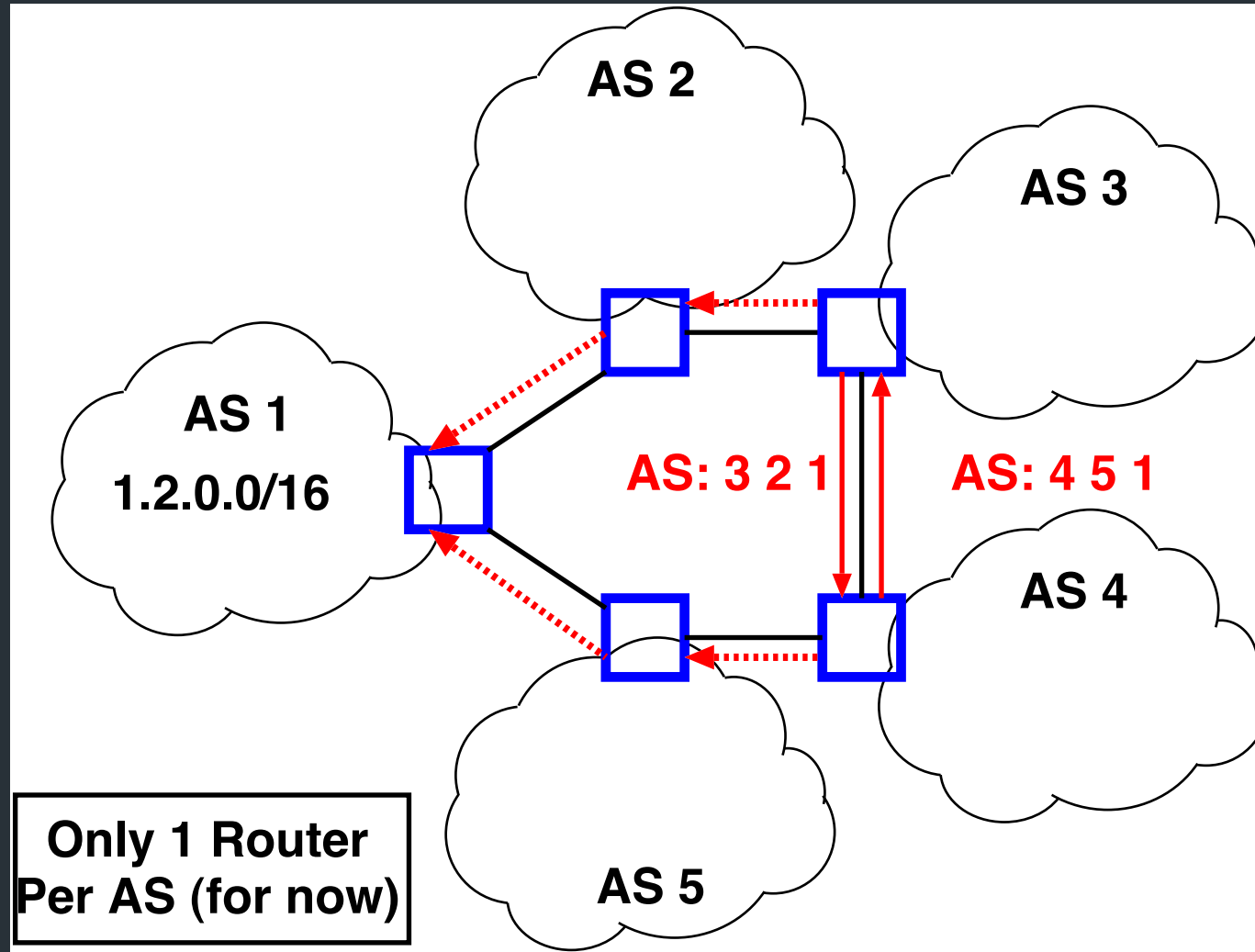
BGP Example

BGP Example
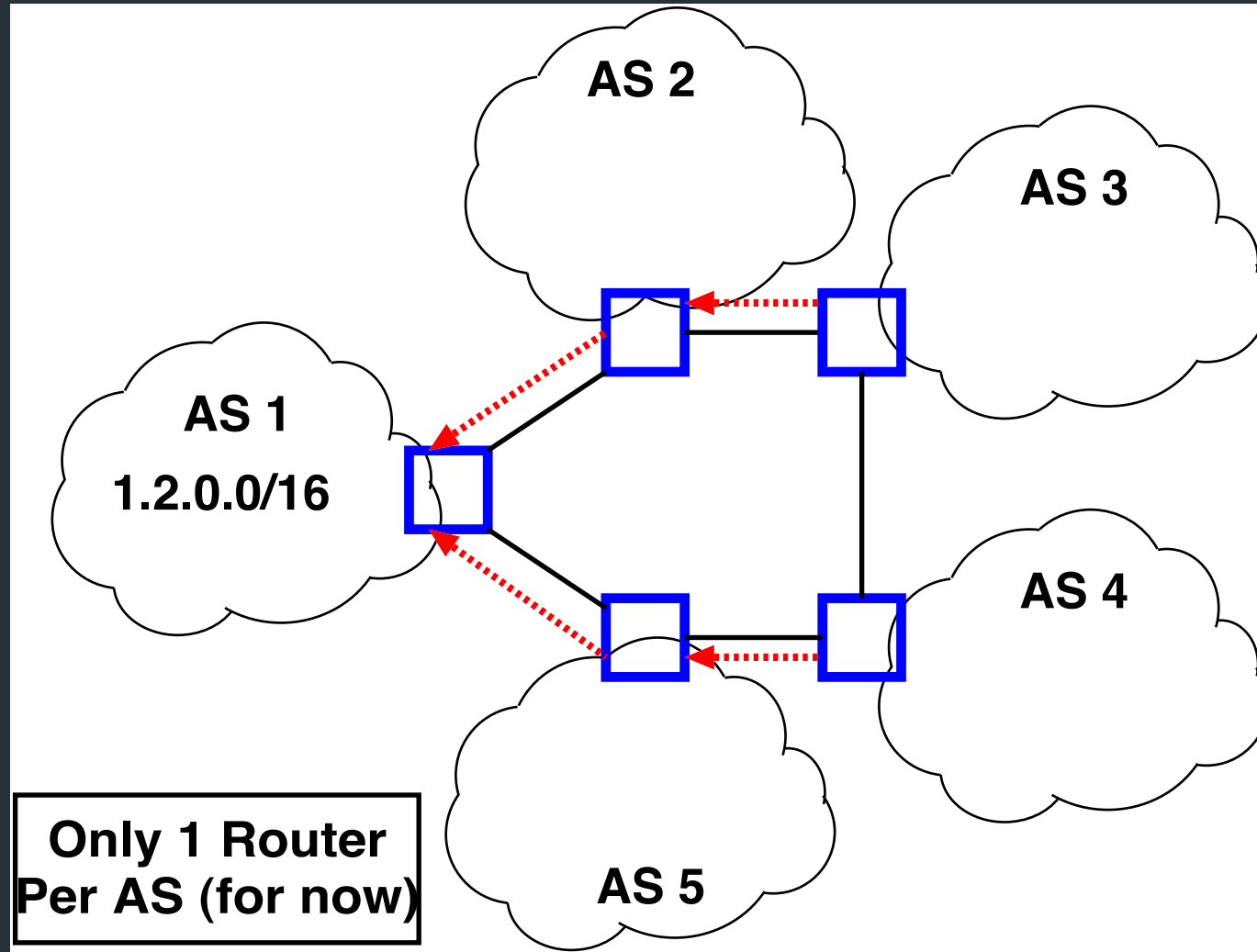
Demo: AS11078

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

AS Relationships
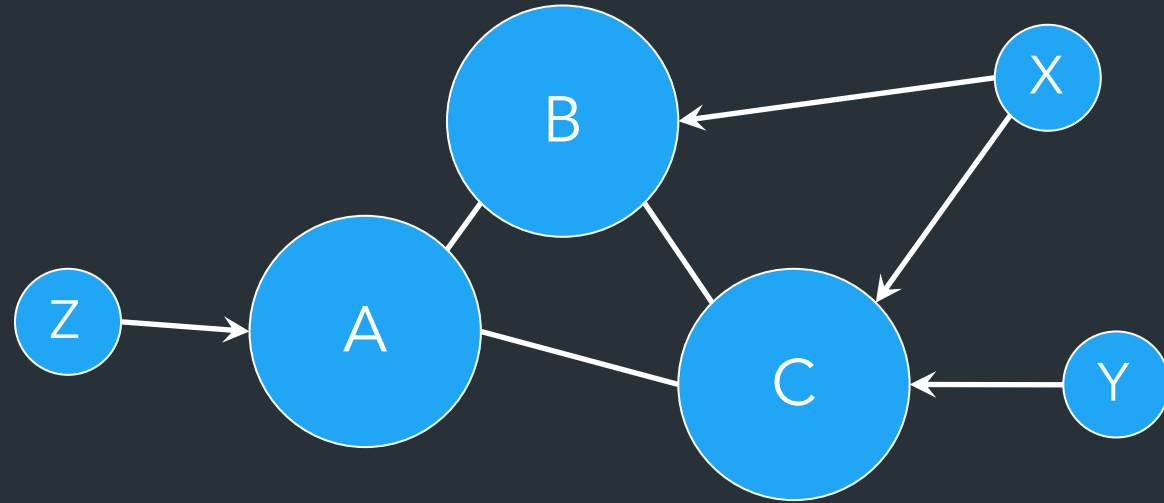
Policies are defined by relationships between ASes

- Provider

- Customer

- Peers

AS Relationships
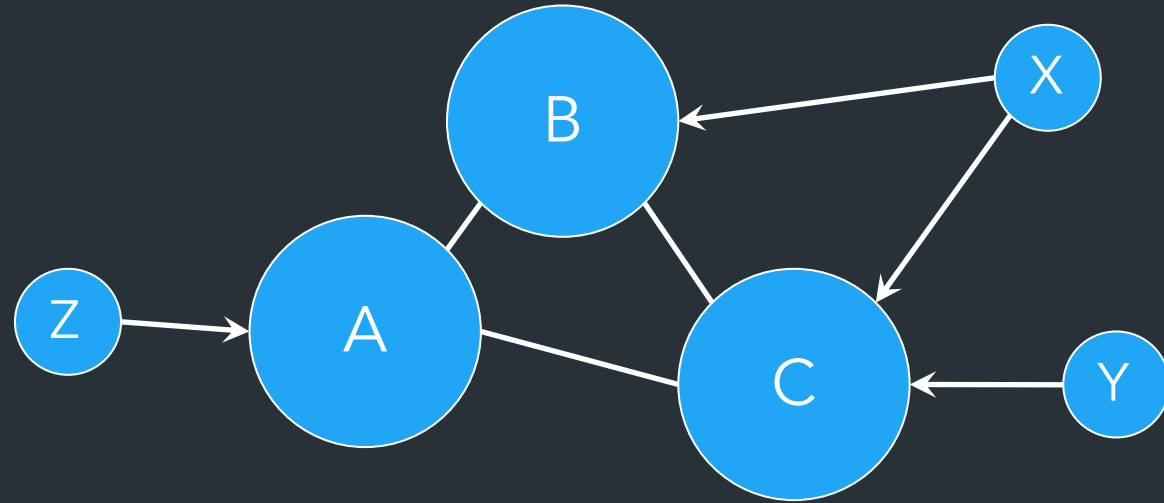
Policies are defined by relationships between Ases

- Provider
- Customer
- Peers

Where do we use policies?

Policies are imposed in how routes are selected and exported

- <u>Selection</u>: which path to use in your network

  - Controls if/how traffic *leaves* the network

- <u>Export</u>: which path to advertise

  - Controls how/if traffic *enters* the network

Update processing
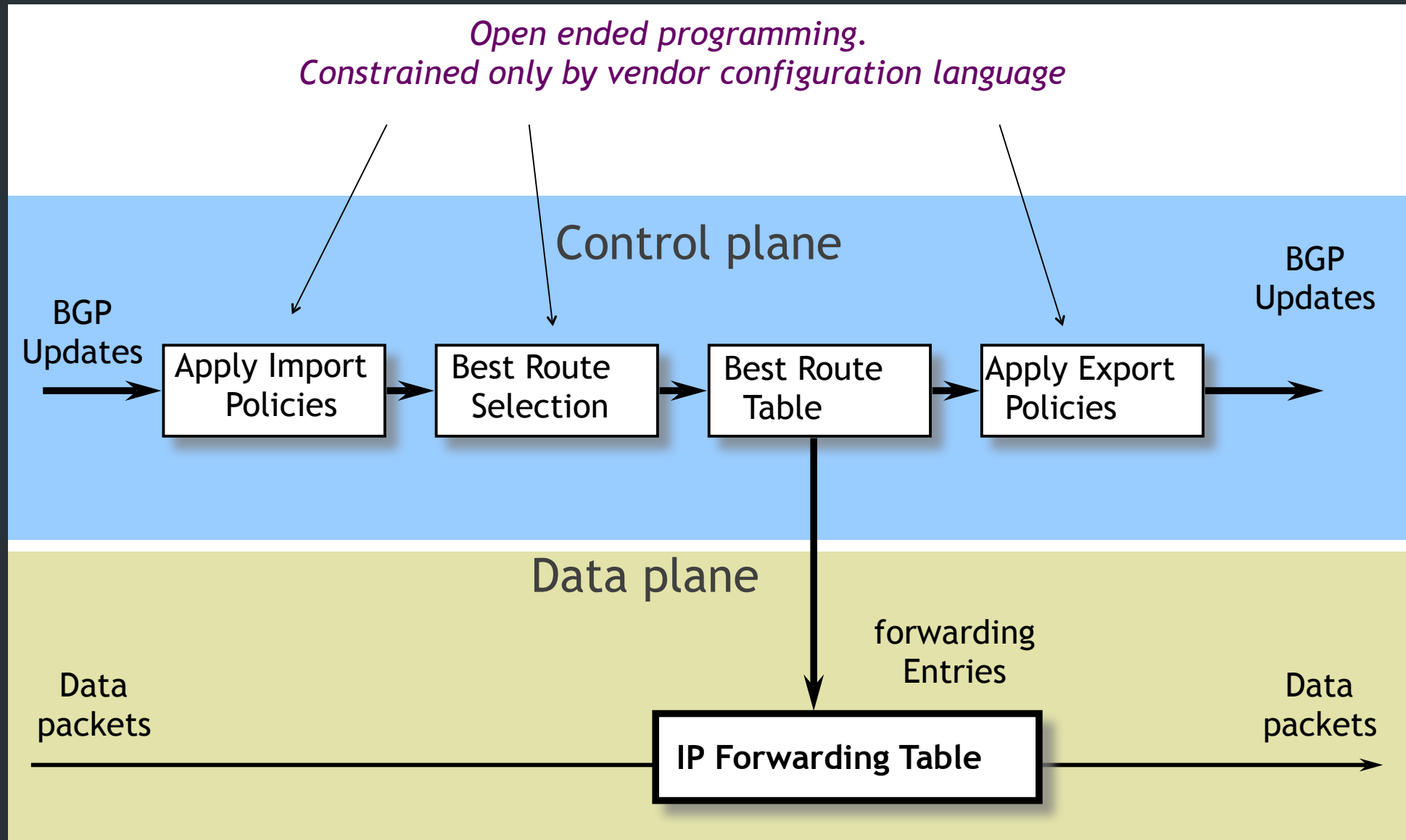
AS relationships

- Customer pays provider for connectivity
  - E.g. Brown contracts with OSHEAN
  - Customer is stub, provider is a transit
- Many customers are multi-homed
  - E.g., OSHEAN connects to Level3, Cogent
- Typical policies:
  - Provider tells all neighbors how to reach customer
  - Provider wants to send traffic to customers ($$$)
  - Customer does not provide transit service

Peer Relationships

- Peer ASs agree to exchange traffic for free
  - Penalties/Renegotiate if imbalance
- Tier 1 ISPs have no default route: all peer with each other
- You are Tier $i + 1$ if you have a default route to a Tier i
- Typical policies
  - AS only exports customer routes to peer
  - AS exports a peer's routes only to its customers
  - Goal: avoid being transit when no gain

Typical route selection policy

In decreasing priority order:

1. Make or save money (send to customer > peer > provider)
2. Try to maximize performance (smallest AS path length)
3. Minimize use of my network bandwidth ("hot potato routing"
4. …

Gao-Rexford Model

- (simplified) Two types of relationships: peers and customer/provider
- Export rules:
  - Customer route may be exported to all neighbors
  - Peer or provider route is only exported to customers
- Preference rules:
  - Prefer routes through customer ($$)
- If all ASes follow this, shown to lead to stable network

Typical Export Policy

Destination prefix advertised by…	Export route to…
Customer	Everyone (providers, peers, other customers…)
Peer	Customers only
Provider	Customers only

Known as Gao-Rexford principles: define common practices for AS relationships

AS Relationships
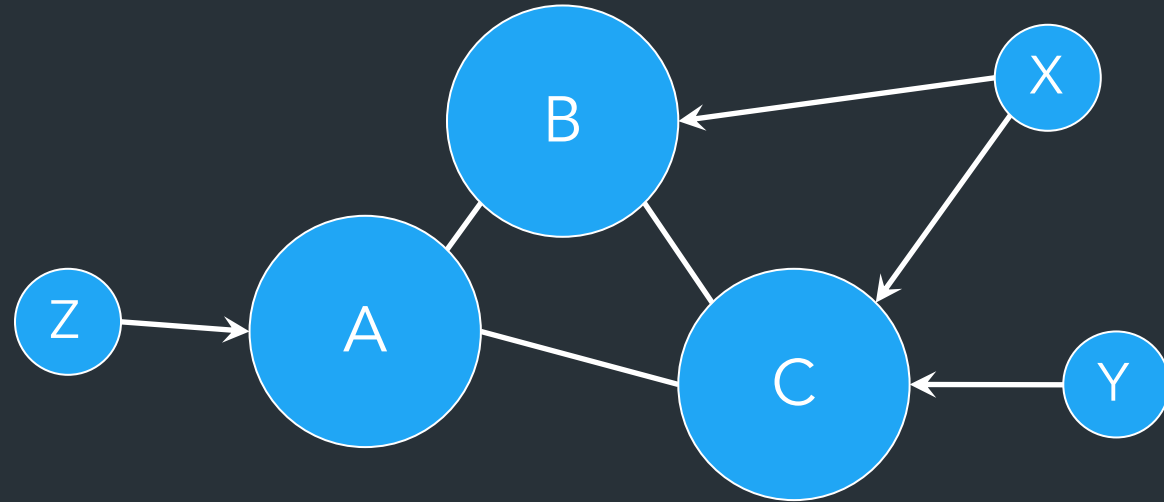
- How to prevent X from forwarding transit between B and C?
- How to avoid transit between CBA ?
  - B: BAZ -> X
  - B: BAZ -> C ? (=> Y: CBAZ and Y:CAZ)

Peering Drama

- Cogent vs. Level3 were peers
- In 2003, Level3 decided to start charging Cogent
- Cogent said no
- Internet partition: Cogent's customers couldn't get to Level3's customers and vice-versa
  - Other ISPs were affected as well
- Took 3 weeks to reach an undisclosed agreement

BGP can be fragile

- Individual router configurations and policy can affect whole network

- Consequences sometimes disastrous…

Some BGP Challenges

- Convergence
- Traffic engineering
  - How to assure certain routes are selected
- Misconfiguration
- Security

BGP can be fragile! One router configuration can affect a large portion of the network

Recent Notable incidents

- October 4 2021: Facebook accidentally removed routes for its DNS servers
    - Outside world couldn't resolve facebook.com, and neither could Facebook!
- June 24, 2019: Misconfigured router accepted lots of transit traffic

Jérôme Fleury

[URGENT] Route-leak from your customer

To: CaryNMC-IP@one.verizon.com,   peering@verizon.com,   help4u@verizon.com,

Demo

- Route views project: http://www.routeviews.org
  - telnet route-views.linx.routeviews.org
  - show ip bgp 128.148.0.0/16 longer-prefixes
- All paths are learned internally (iBGP)
- Not a production device

```
$ telnet route-views.telxatl.routeviews.org
Trying 67.23.60.46...
Connected to route-views.telxatl.routeviews.org.
Escape character is '^]'.


Hello, this is Quagga (version 1.1.0).
Copyright 1996-2005 Kunihiro Ishiguro, et al.


route-views.telxatl.routeviews.org> show ip bgp 128.148.0.0/16 longer-prefixes
BGP table version is 0, local router ID is 198.32.132.3
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
              i internal, r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete


   Network          Next Hop            Metric LocPrf Weight Path
*  128.148.0.0      198.32.132.152                        0 6082 2914 3257 14325 11078 i
*                   198.32.132.160                        0 27446 27446 6939 14325 11078 i
*                   198.32.132.12            0             0 19151 6939 14325 11078 i
*                   198.32.132.75                         0 15008 6939 14325 11078 i
*                   198.32.132.28                         0 4181 6939 14325 11078 i
*                   198.32.132.75                         0 3491 6939 14325 11078 i
*                   198.32.132.75                         0 53828 6939 14325 11078 i
*>                  198.32.132.75                         0 6939 14325 11078 i
```

11078 is Brown's ASN

14325 is Brown's Provider, OSHEAN

Anatomy of an UPDATE

- Withdrawn routes: list of withdrawn IP prefixes
- Network Layer Reachability Information (NLRI)
    – List of prefixes to which path attributes apply
- Path attributes
    – ORIGIN, AS_PATH, NEXT_HOP, MULTI-EXIT-DISC, LOCAL_PREF, ATOMIC_AGGREGATE, AGGREGATOR, …
    – Extensible: can add new types of attributes

Example

- NLRI: 128.148.0.0/16
- AS-Path: ASN 44444 3356 14325 11078
- Next Hop IP
- Various knobs for traffic engineering:
  - Metric, weight, LocalPath, MED, Communities
  - Lots of tuning by admins

Prefix aggregation

Warmup for discussion

Given this routing table, to which
prefix would a router map each IP?

- 1.2.3.4
- 138.16.100.5
- 138.16.10.200
- 12.34.5.120
- 12.34.18.5

Prefix	Next Hop
1.0.0.0/8	...
12.34.0.0/16	...
12.34.16.0/20	...
138.16.0.0/16	...
138.16.100.0/24	...

Longest Prefix Match

When performing a forwarding table lookup, select the most specific prefix that matches an address

- Eg. 12.34.18.5

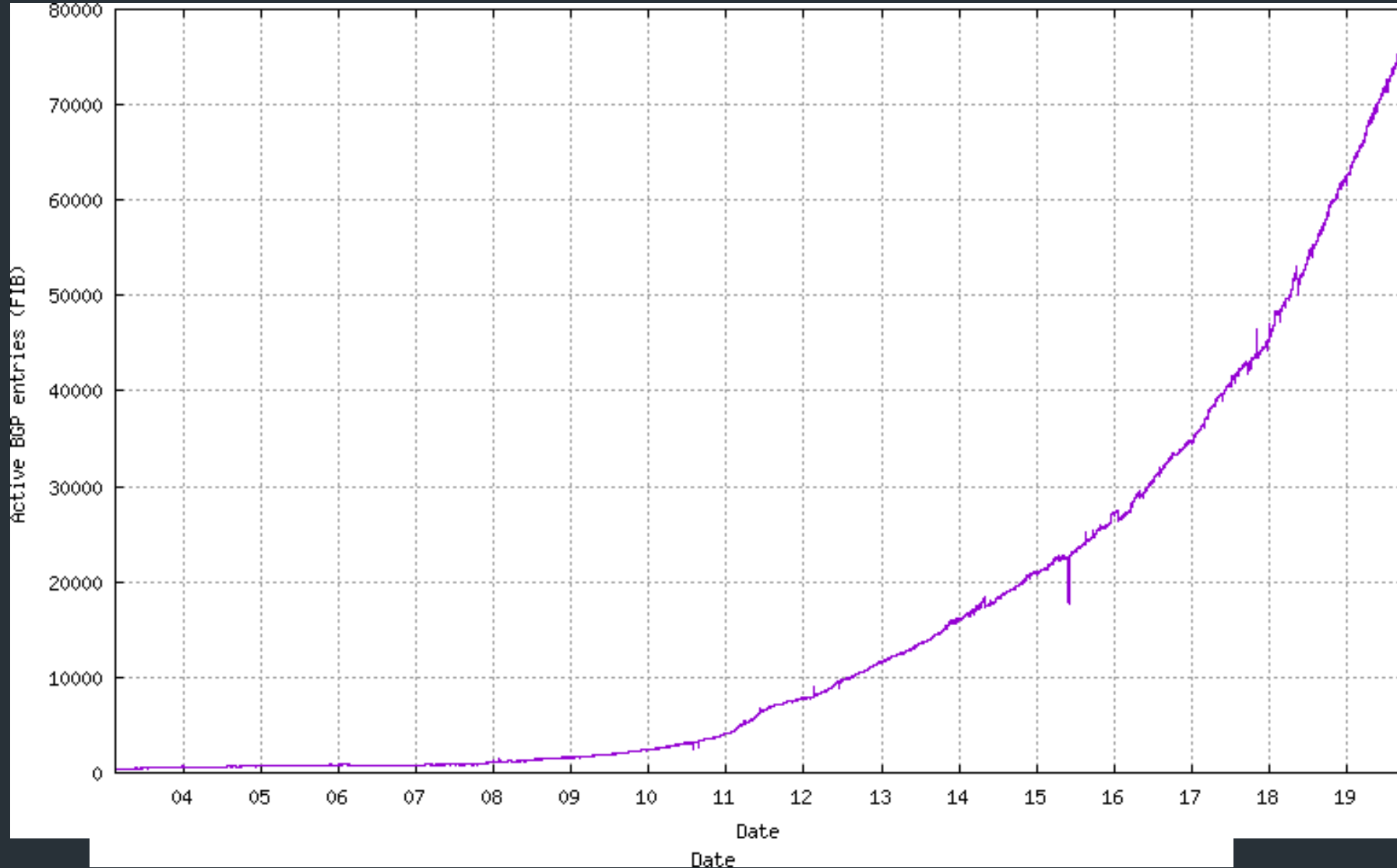| Prefix	Next Hop
1.0.0.0/8	…
12.34.0.0/16	…
12.34.16.0/20	…
138.16.0.0/16	…
138.16.100.0/24	…

Internet routers have specialized memory called TCAM (Ternary Content Addressable Memory) to do longest prefix match *fast* (one clock cycle!)
Goal: forward at *line rate* (as fast as link allows)

BGP Table Growth
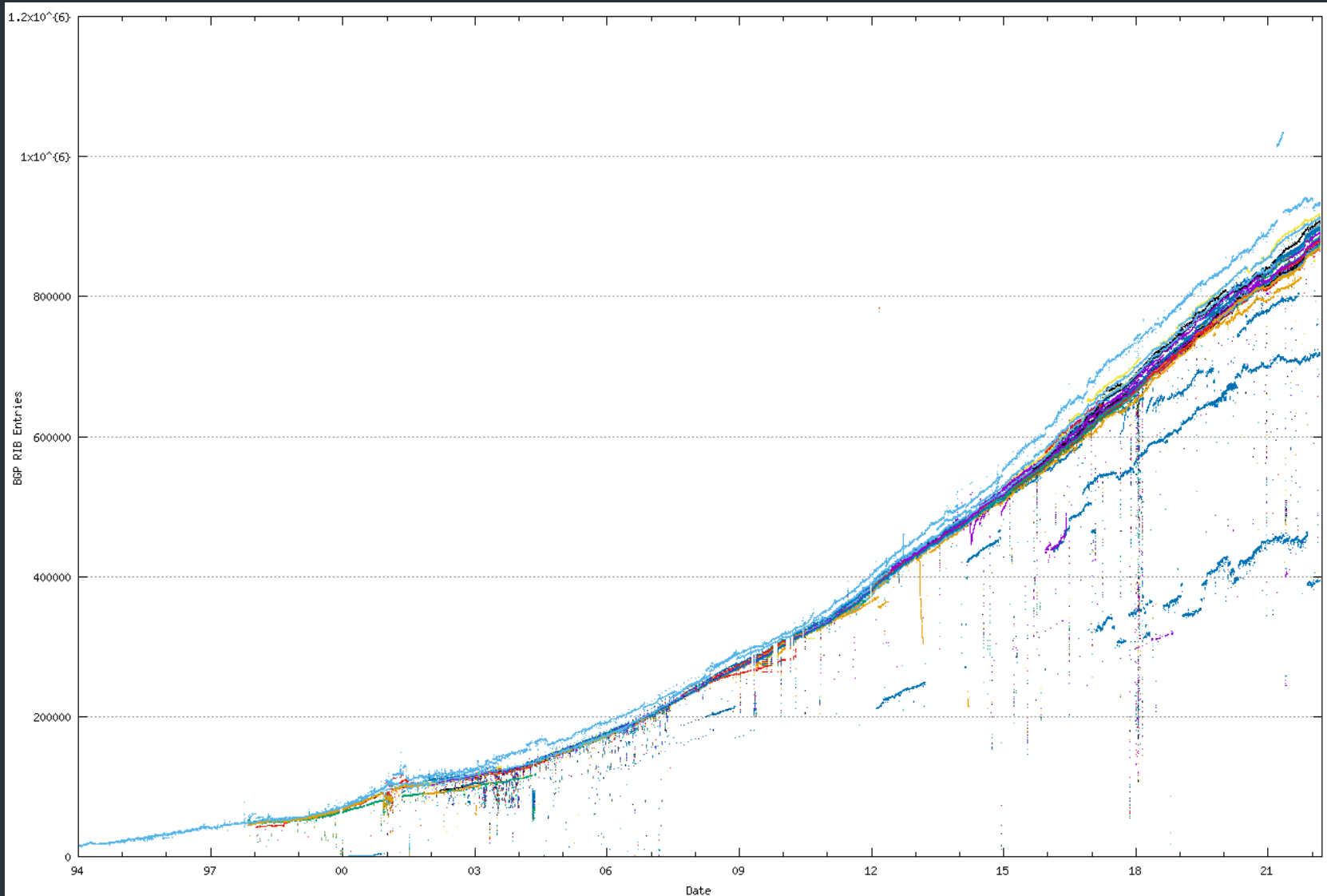
Source: bgp.potaroo.net

BGP Table Growth for v6

Source: bgp.potaroo.net

512k day

- On August 12, 2014, the full IPv4 BGP table reached 512k prefixes
- Many older routers had only 512k of TCAM, had to fall back to slower routing methods
- Caused outages in Microsoft Azure, ebay, others…

What can lead to table growth?

- More addresses being allocated
- Fragmentation
  - Multihoming
  - Change of ISPs
  - Address re-selling

Recall: BGP mechanics

- Path-vector protocol
- Exchange prefix reachability with neighbors (ASes)
  - E.g., "I can reach prefix 128.148.0.0/16 through ASes 44444 3356 14325 11078"
- Select routes to propagate to neighbors based on routing *policy, not shortest-path costs*

- Today: Policies and implications

Next class

- BGP Policy Routing and Security

Reliable Flooding

- Store most recent LSP from each node
  - Ignore earlier versions of the same LSP
- Forward LSP to all nodes but the one that sent it
- Generate new LSP periodically (increment SEQNO)
- Start at SEQNO=0 when reboot
  - If you hear your own packet with SEQNO=n, set your next SEQNO to n+1
- Decrement TTL of each stored LSP
  - Discard when TTL=0

Calculating best path

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

- Computation: more expensive than
- Example: D: (D,0,-) (C,2,C) (B,5,C) (A,10,C)
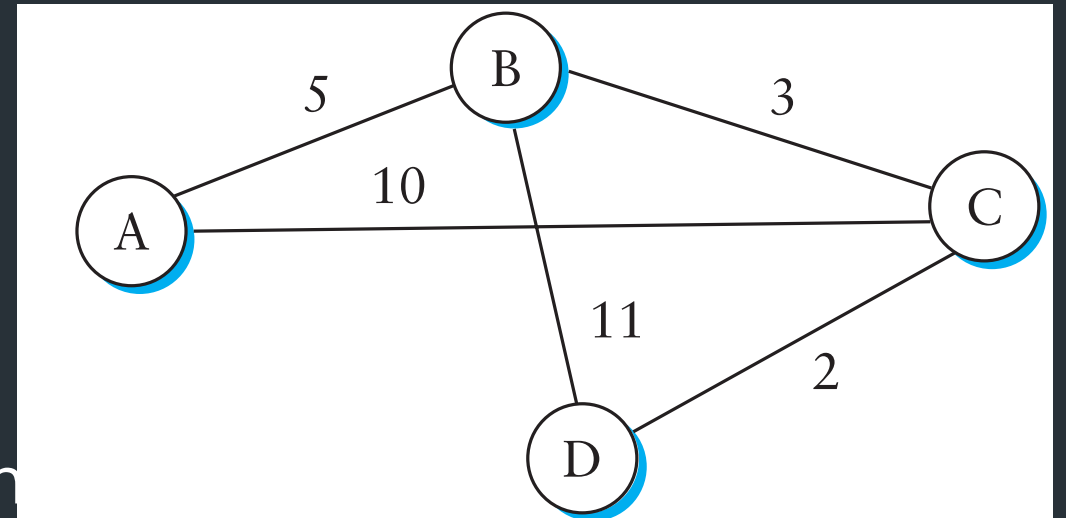
Distance Vector vs. Link State

- # of messages (per node)
  - DV: O(d), where d is degree of node
  - LS: O(nd) for n nodes in system
- Computation
  - DV: convergence time varies (e.g., count-to-infinity)
  - LS: $O(n^2)$ with O(nd) messages
- Robustness: what happens with malfunctioning router?
  - DV: Nodes can advertise incorrect *path* cost, which propagates through network
  - LS: Nodes can advertise incorrect *link* cost

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