CSE 461: Distance Vector Routing

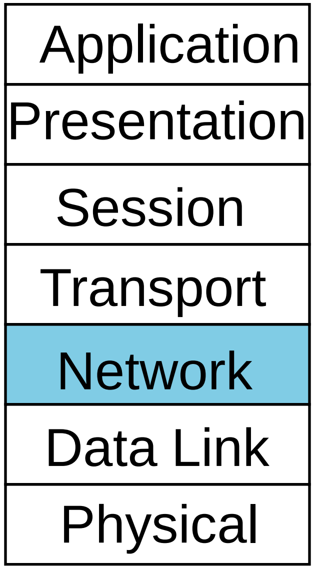
Jeremy Elson (jelson@gmail.com)
Microsoft Research

Ben Greenstein (ben@cs.washington.edu)
Intel Research

October 22, 2008

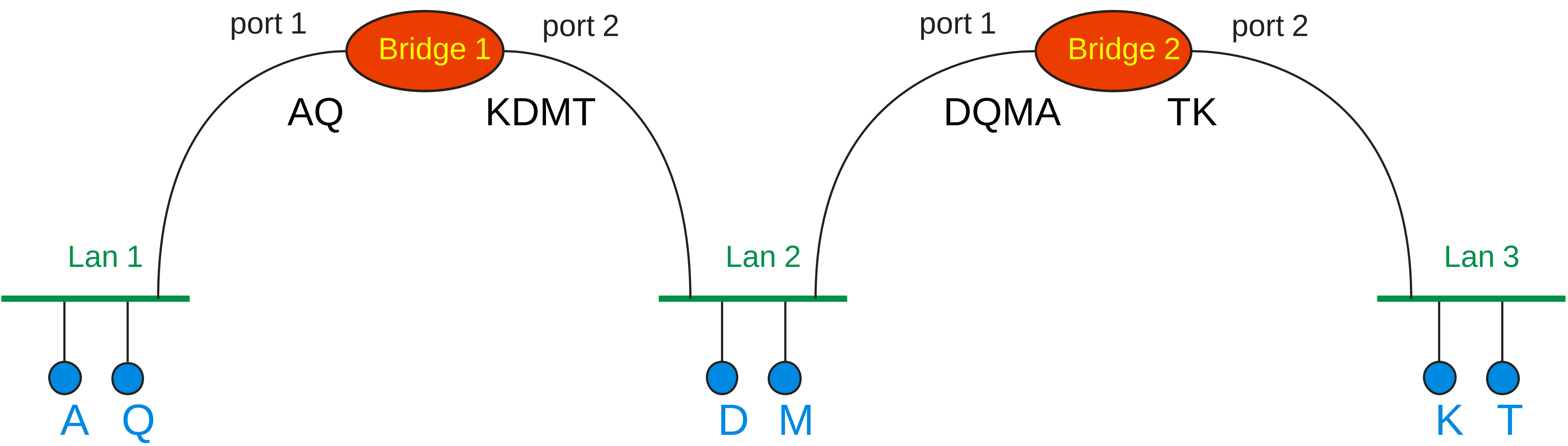
Next Topic

- Focus
  - Why is routing necessary?
  - How do we calculate routes?
  - How are routes used?
- Routing Algorithms
  - Distance Vector routing
  - A real-world implementation: RIP



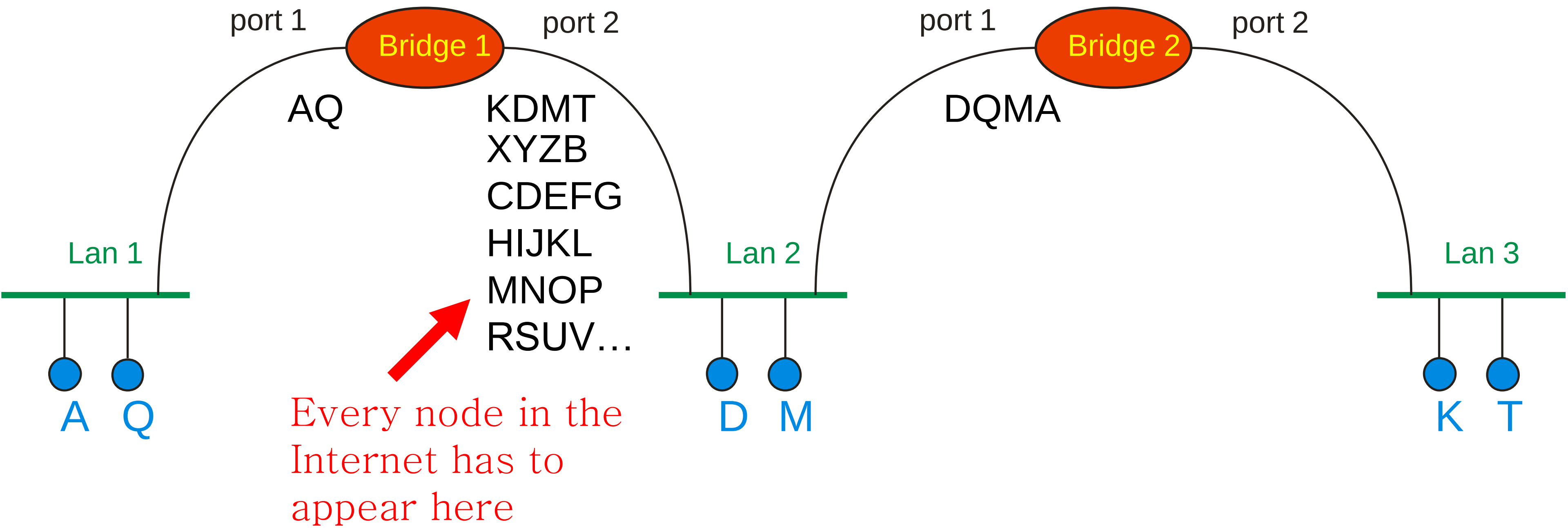
Bridges worked pretty well, right?

Why can't we make bridged networks bigger and bigger?
Why not make them Internet-sized?



Scaling Limitations of Bridged Networks

Table size: works fine for a few thousand nodes.
Can it scale to a billion?
Can we do lookups at “wire speed”?

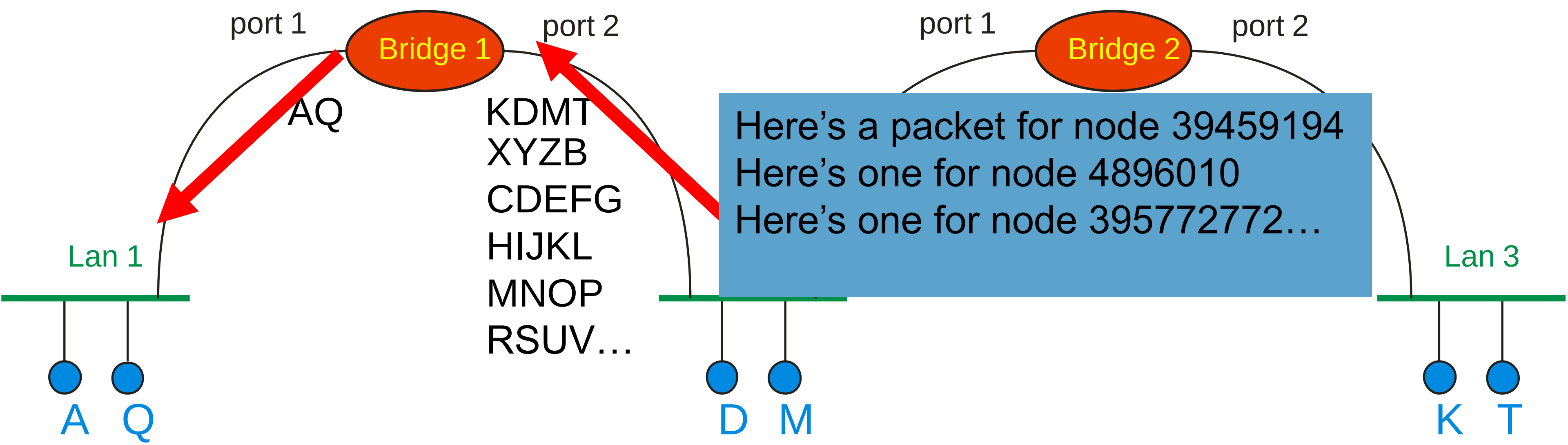


Scaling Limitations of Bridged Networks

Table maintenance: to find unknown nodes, we broadcast.

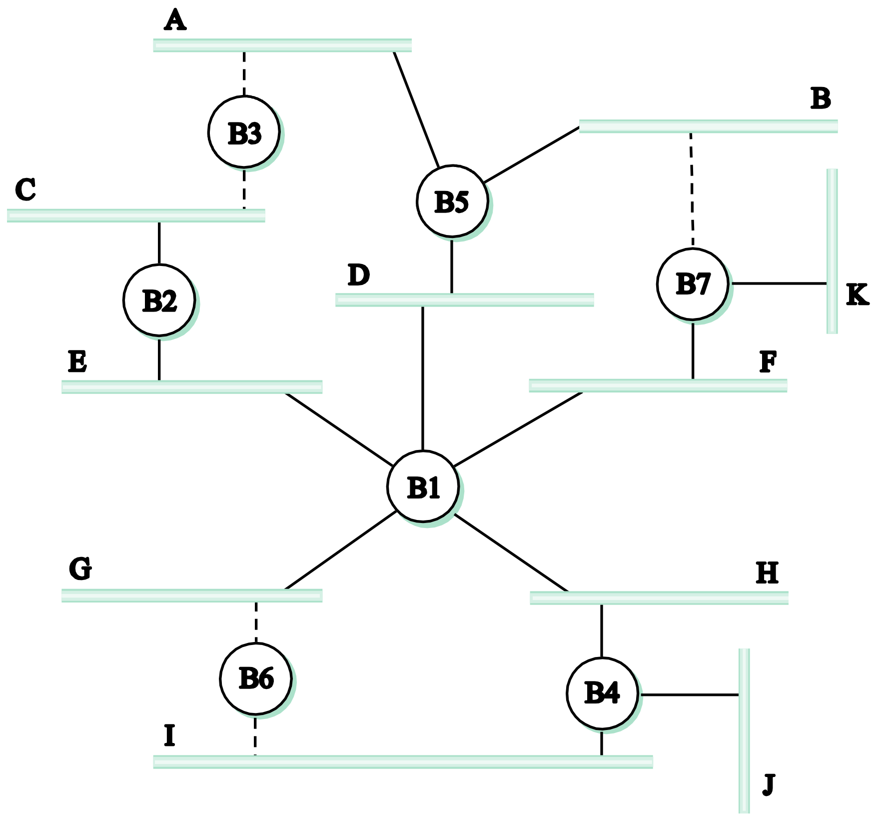
n nodes looking for *m* destinations:

nm stray packets on your link!

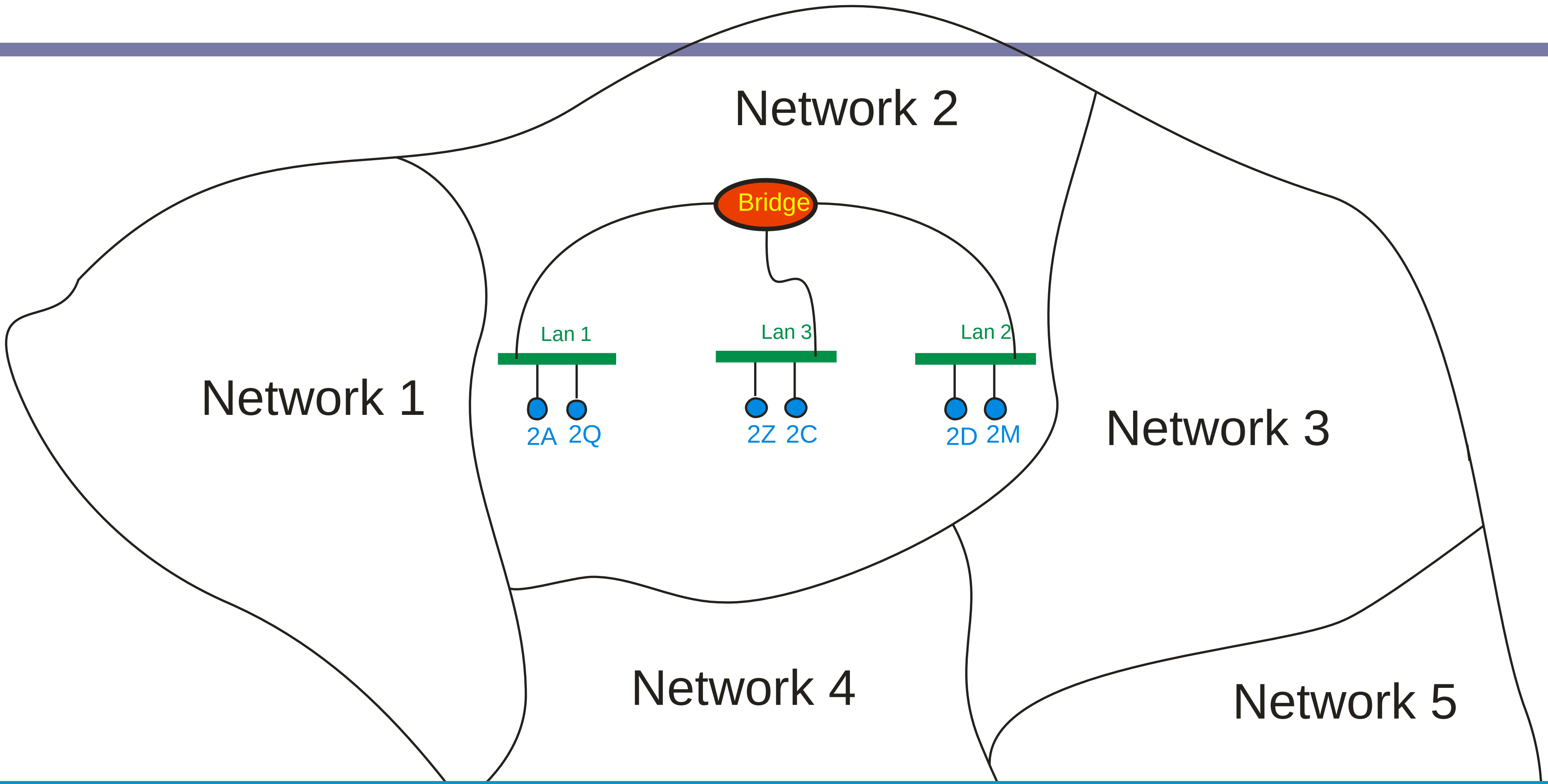


Scaling Limitations of Bridged Networks

Spanning tree algorithm: Picking one root doesn't work!
Who gets to be the root?
We need something more "egalitarian".

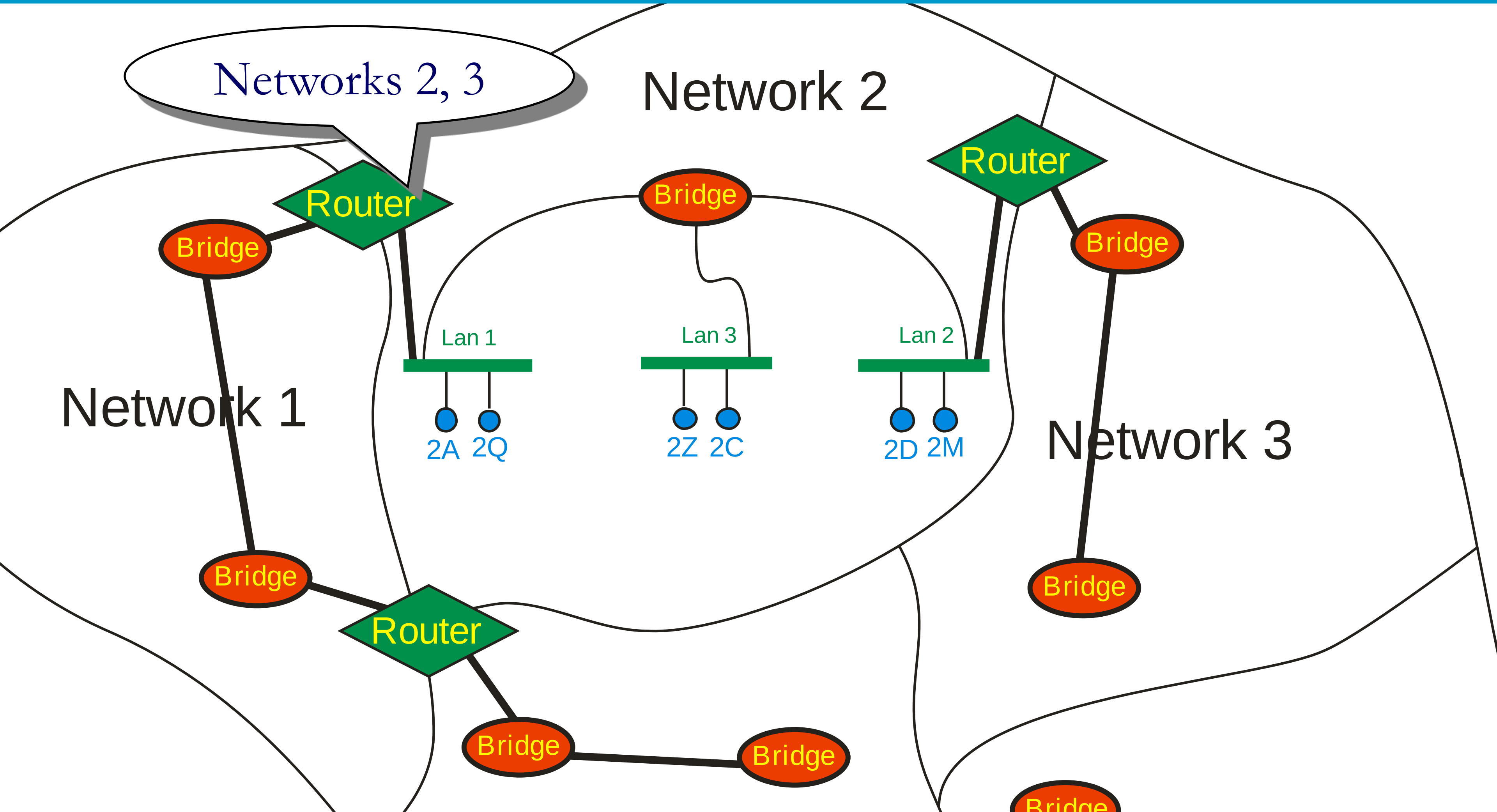


Hierarchy to the rescue!



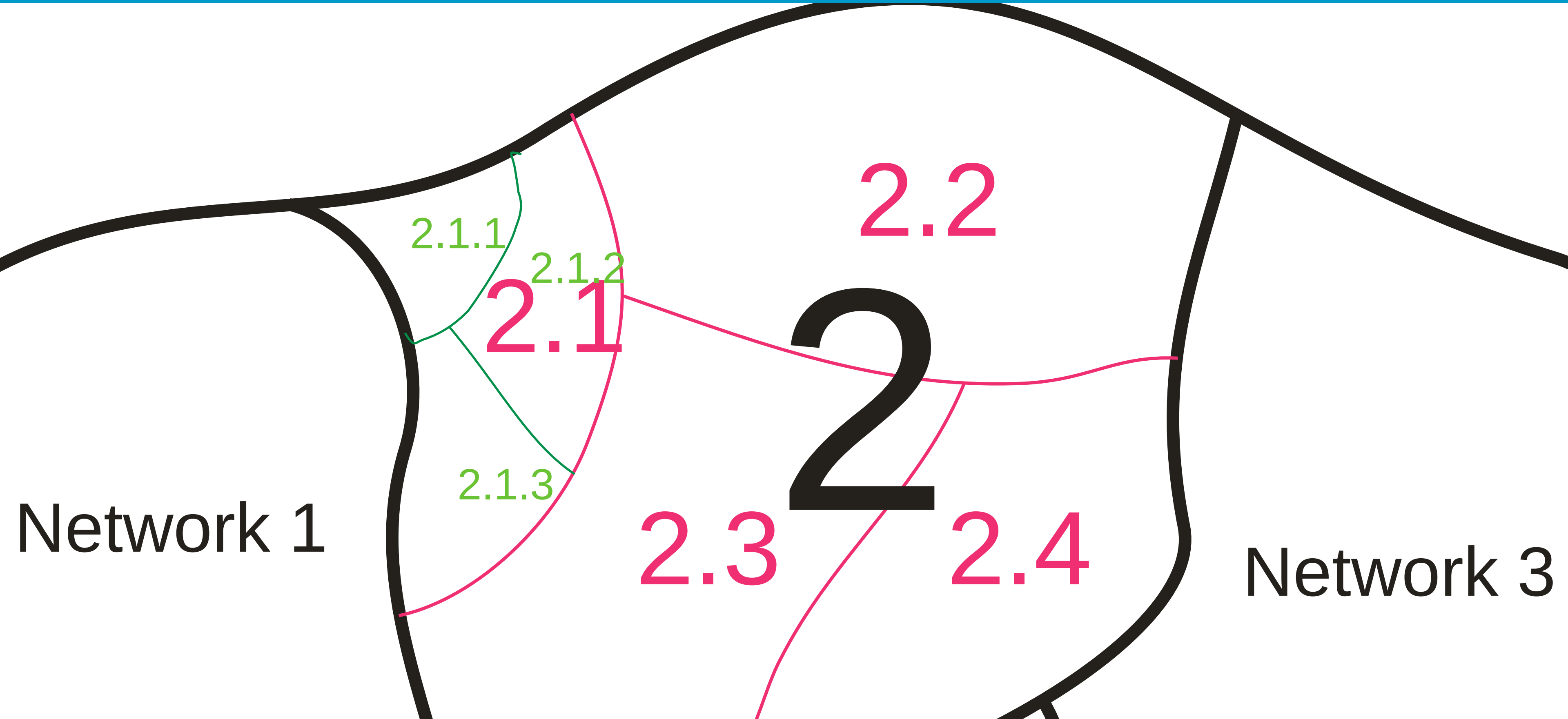
Disadvantage: The network is no longer "plug and play". We need to assign addresses – not just unique identifiers.

Advantages: We can now *aggregate* routing information.
*1/n*th as many networks as hosts → fewer updates, smaller tables.
Local changes don't cause global updates.



The 0, 1, Infinity Principle At Work

The original Internet had exactly 1 level of hierarchy:
Network Address and Host Address (Class A, B, C…)
From the mid-90’s: CIDR allows arbitrary sub-networking.
Further improves route aggregation in the Internet core.



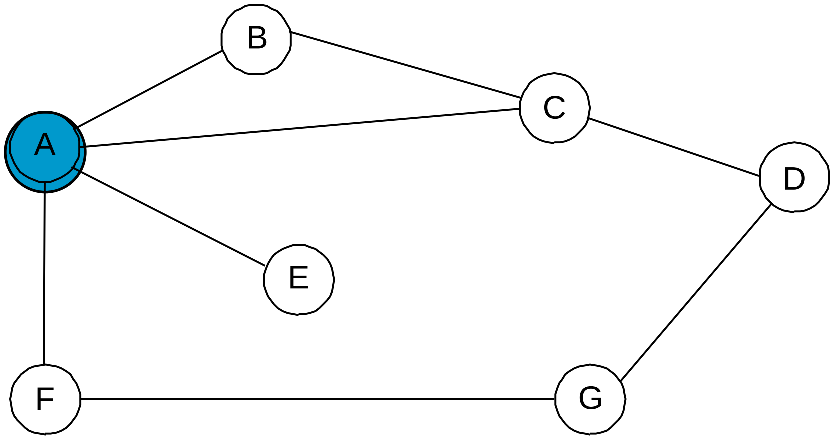
Forwarding and Routing

- Each node has a “routing table”: tells the router which outgoing link should be used for each known destination network
- Routing is the process that all routers go through to calculate the routing tables
  - Involves *global* decisions
- Forwarding is the process that each router goes through for every packet to send it on its way
  - Involves *local* decisions
  - In the Internet, more specific routes are encountered as you approach your destination

What's in a Routing Table?

- The routing table at A, for example, lists at a minimum the next hops for the different destinations

Dest	Next Hop
B	B
C	C
D	C
E	E
F	F
G	F



Kinds of Routing Schemes

- Many routing schemes have been proposed/explored!

- Distributed or centralized
- Hop-by-hop or source-based
- Deterministic or stochastic
- Single or multi-path
- Static or dynamic route selection

- Internet is to the left ☺

Routing Questions/Challenges

- How do we choose best path? (What does "best" mean?)
- How do we scale to billions of nodes?
- How do we adapt to failures or changes?
  - Node and link failures, plus message loss
  - We will use distributed algorithms
- "Real world" concerns of the Internet (ignore for now):
  - Parties don't trust each other
  - Policy has to come into play

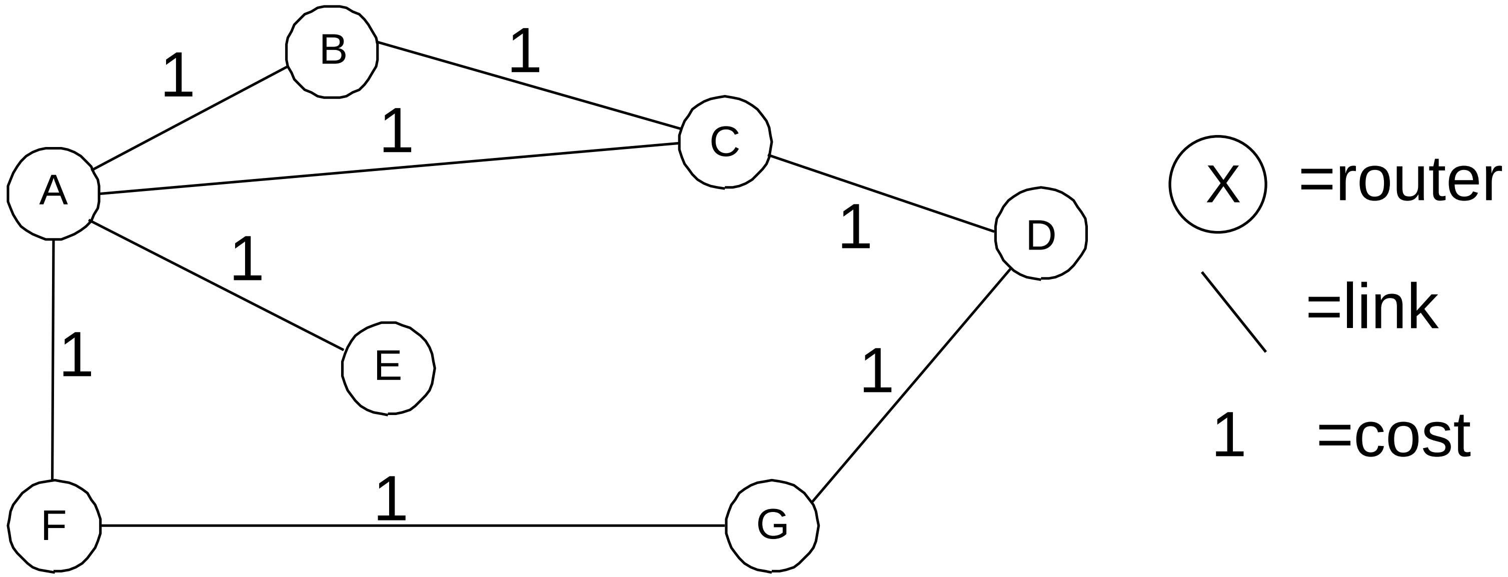
Some Pitfalls

- Using global knowledge is challenging
  - Hard to collect
  - Can be out-of-date
  - Needs to summarize in a locally-relevant way

- Inconsistencies in local /global knowledge can cause:
  - Loops (black holes)
  - Oscillations, esp. when adapting to load

Network as a Graph

- Routing is essentially a problem in graph theory. Remember Bellman-Ford Single-Source Shortest Path?



Distance Vector Routing

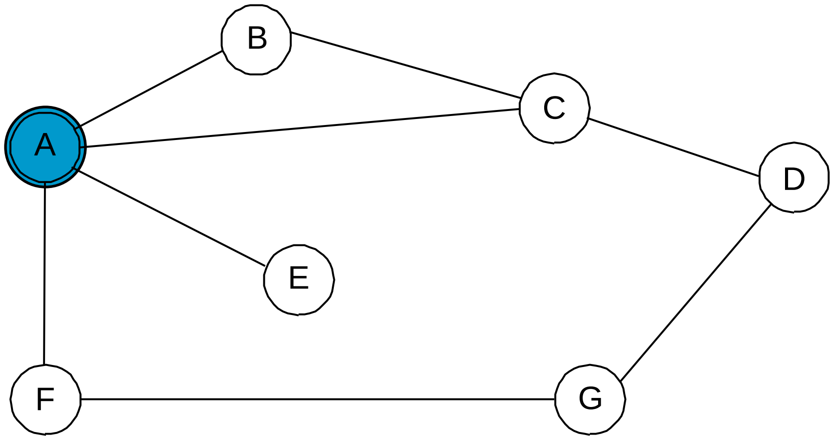
- Assume:
  - Each router knows only address/cost of neighbors
- Goal:
  - Calculate routing table of next hop information for each destination at each router
- Idea:
  - Tell neighbors about learned distances to all destinations

- This is (vaguely) like running Bellman-Ford once for each source everywhere in parallel

DV Algorithm

- Each router maintains a vector of costs to all destinations as well as routing table
  - Initialize neighbors with known cost, others with infinity
- Periodically send copy of distance vector to neighbors
- On reception of a vector, if your neighbor's path to a destination plus cost to that neighbor cost is better
  - Update the cost and next-hop in your outgoing vectors
- Assuming no changes, will converge to shortest paths
  - But what happens if there are changes?

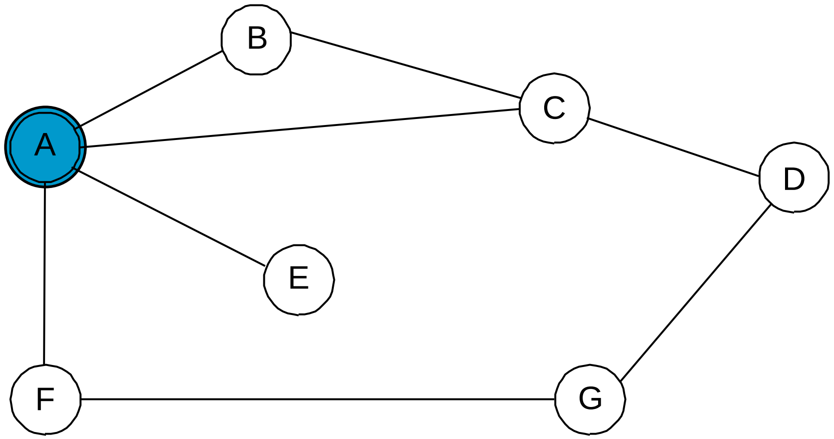
DV Example – Initial Table at A



Dest	Cost	Next
B	1	B
C	1	C
D	∞	-
E	1	E
F	1	F
G	∞	-

DV Example – Final Table at A

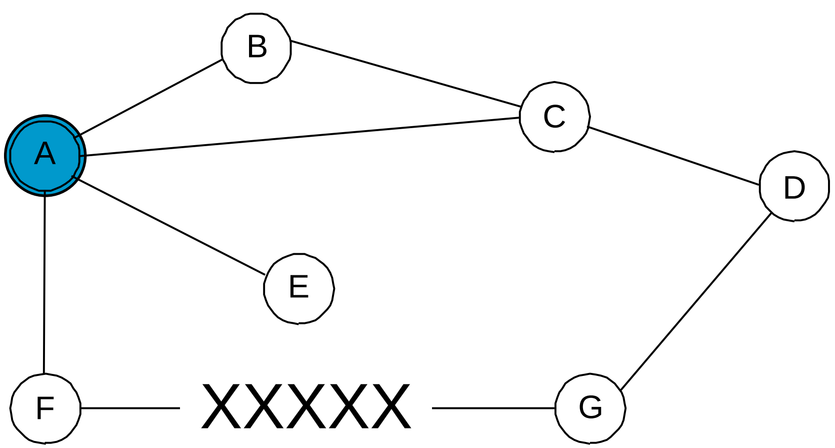
This simple example converges after one iteration



Dest	Cost	Next
B	1	B
C	1	C
D	2	C
E	1	E
F	1	F
G	2	F

What if there are changes?

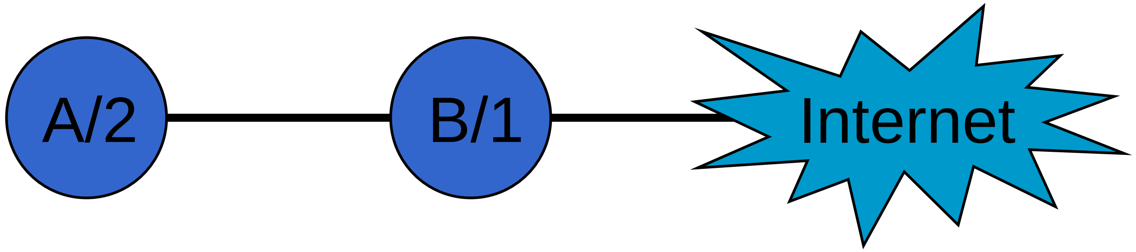
- One scenario: Suppose link between F and G fails
  1. F notices failure, sets its cost to G to infinity and tells A
  2. A sets its cost to G to infinity too, since it learned it from F
  3. A learns route from C with cost 2 and adopts it



Dest	Cost	Next
B	1	B
C	1	C
D	2	C
E	1	E
F	1	F
G	3	C

Count To Infinity Problem

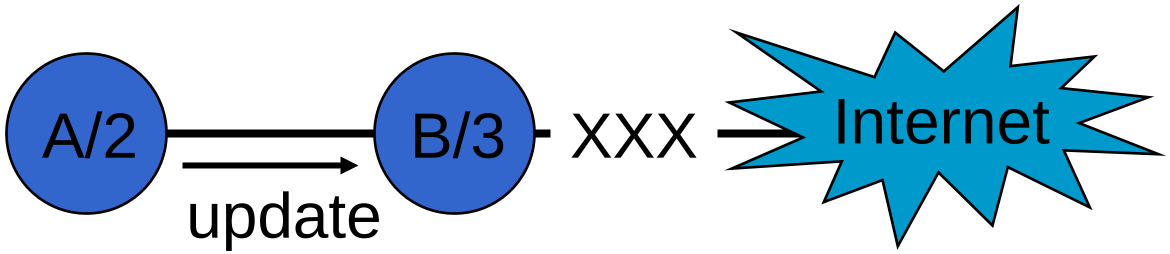
- Imagine two nodes want to maintain routes to the Internet.



What happens when the link between B and Internet fails?

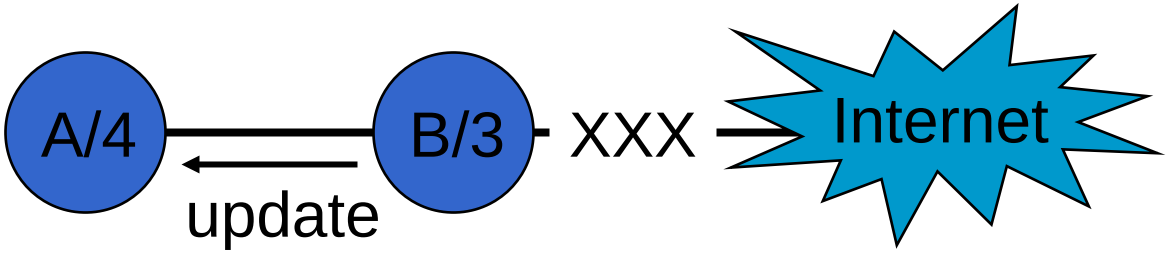
Count To Infinity Problem

- B hears of a route to the Internet via A with cost 2
- So B switches to the "better" (but wrong!) route



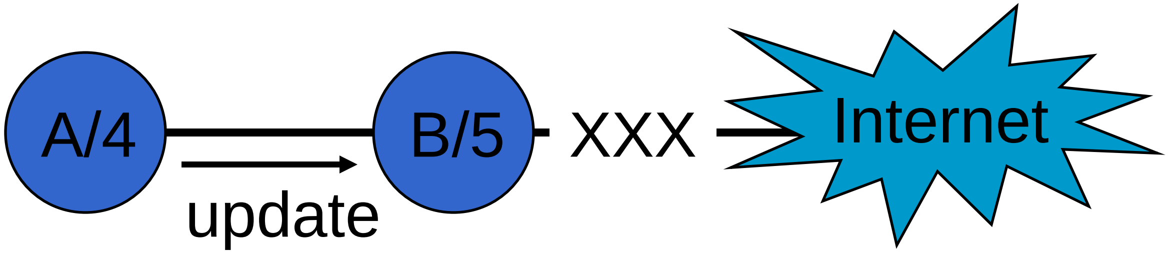
Count To Infinity Problem

- A hears from B and increases its cost



Count To Infinity Problem

- B hears from A and (surprise) increases its cost
- Cycle continues and we "count to infinity"



- Packets caught in the crossfire loop between A and B

Split Horizon

- Solves trivial count-to-infinity problem
- Router never advertises the cost of a destination back to its next hop – that's where it learned it from!
- Poison reverse: go even further – advertise back infinity
- However, DV protocols still subject to the same problem with more complicated topologies – e.g., 3 node loops
  - Many enhancements suggested

Routing Information Protocol (RIP)

- DV protocol with hop count as metric
  - Infinity value is 16 hops; limits network size
  - Includes split horizon with poison reverse
- Routers send vectors every 30 seconds
  - With triggered updates for link failures
  - Time-out in 180 seconds to detect failures

- RIPv1 specified in RFC1058
  - www.ietf.org/rfc/rfc1058.txt
- RIPv2 (adds authentication etc.) in RFC1388
  - www.ietf.org/rfc/rfc1388.txt

RIP is an "Interior Gateway Protocol"

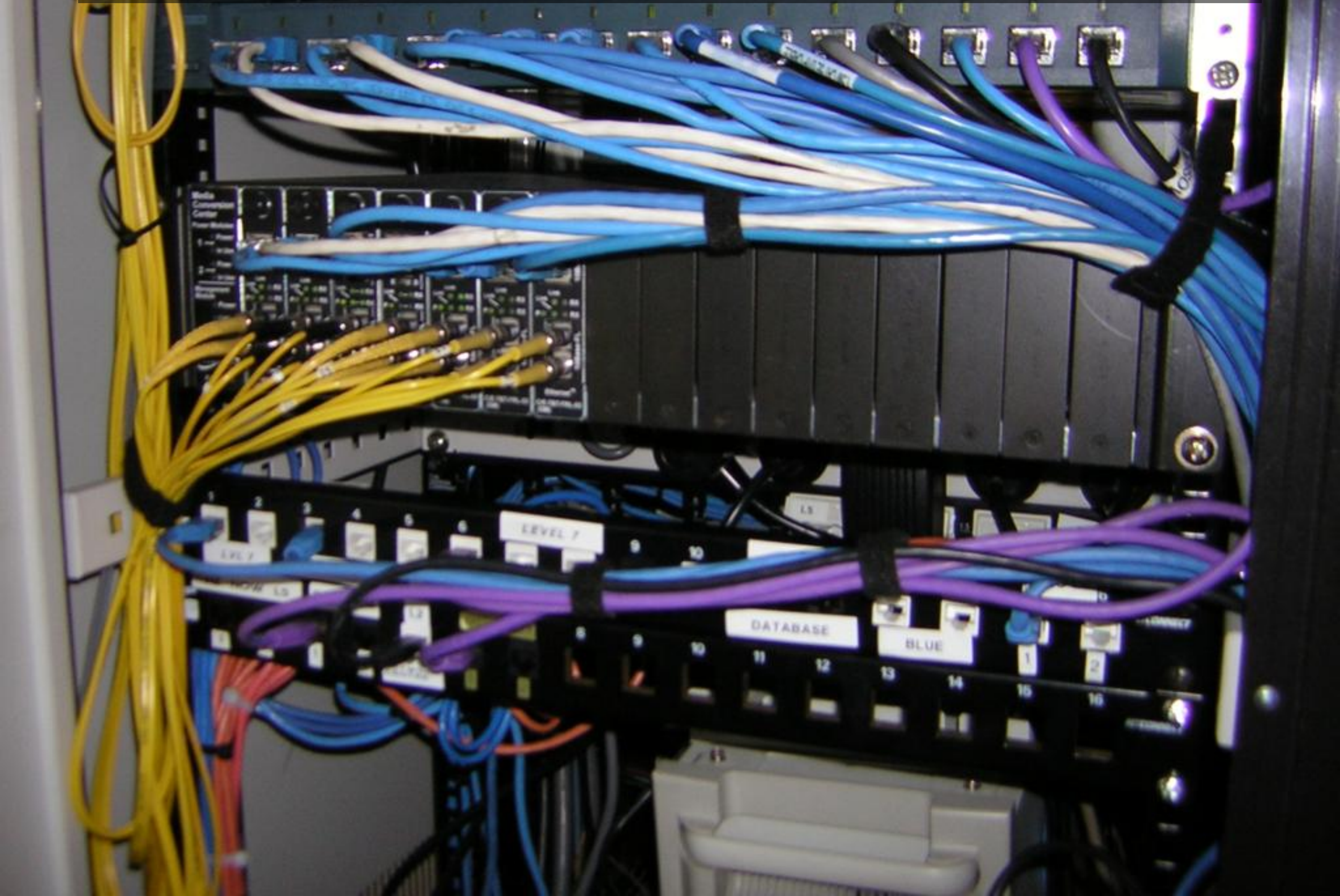
- Suitable for small- to medium-sized networks
  - such as within a campus, business, or ISP
- Unsuitable for Internet-scale routing
  - hop count metric poor for heterogeneous links
  - 16-hop limit places max diameter on network
- Later, we'll talk about "Exterior Gateway Protocols"
  - used between organizations to route across Internet

IP Datagram Forwarding

- Routing algorithms run on routers, typically not on hosts
- Hosts have few (or one) simple rules:
  - If the destination is on my network, send it directly (ARP for the host)
  - If the destination is on another network, send it to the *default router* (ARP for the router)
  - Ethernet header addressed to router; IP header addressed to end-host
- Routers (sometimes) have a more global view
  - If you're the "last router", ARP for the host
  - Otherwise, send to the next router (may not be Ethernet, so may not literally ARP)
- Border routers often have small routing tables and default gateways
- Internet core routers are behemoths that know all top-level networks

Internet Core Routers

Internet Peering Points



Key Concepts

- Hierarchy and route aggregation are necessary for scaling
- Things we must care about: scale, dynamics
- Routing is a global process, forwarding is local one
- The Distance Vector algorithm and RIP
  - Simple and distributed exchange of shortest paths.
  - Weak at adapting to changes (loops, count to infinity)