



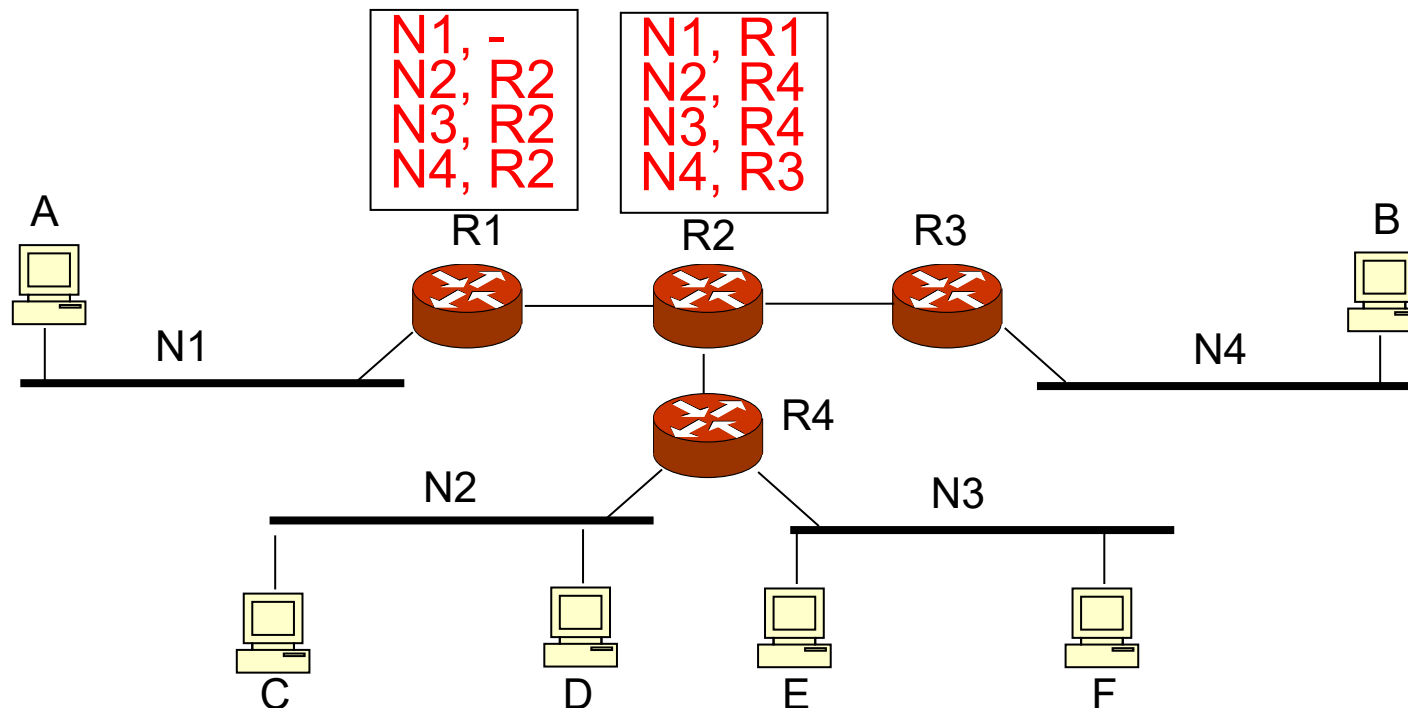
IK1203

Network Layer

Routing

Basic Routing

- Basic "manual" approach:
 - Next-hop routing
 - Logical (IP) addresses
 - Static Tables
- An approach with static tables works only for small IP networks
- We need to support dynamic large networks

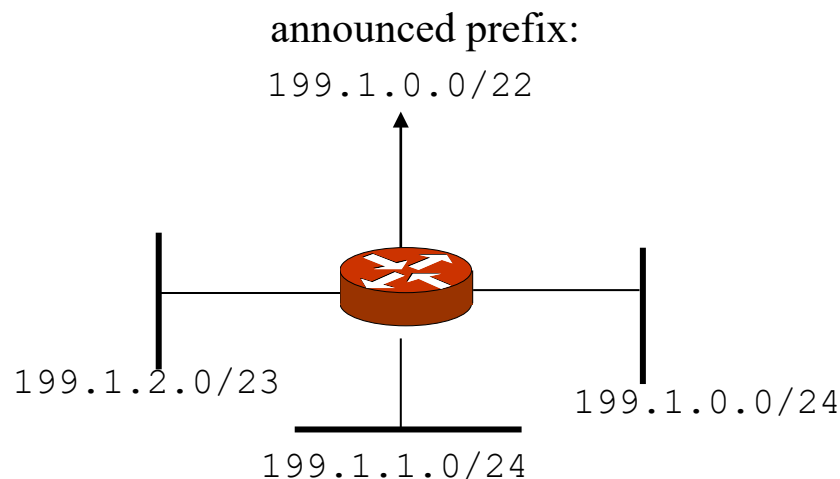


Reachability and Metrics

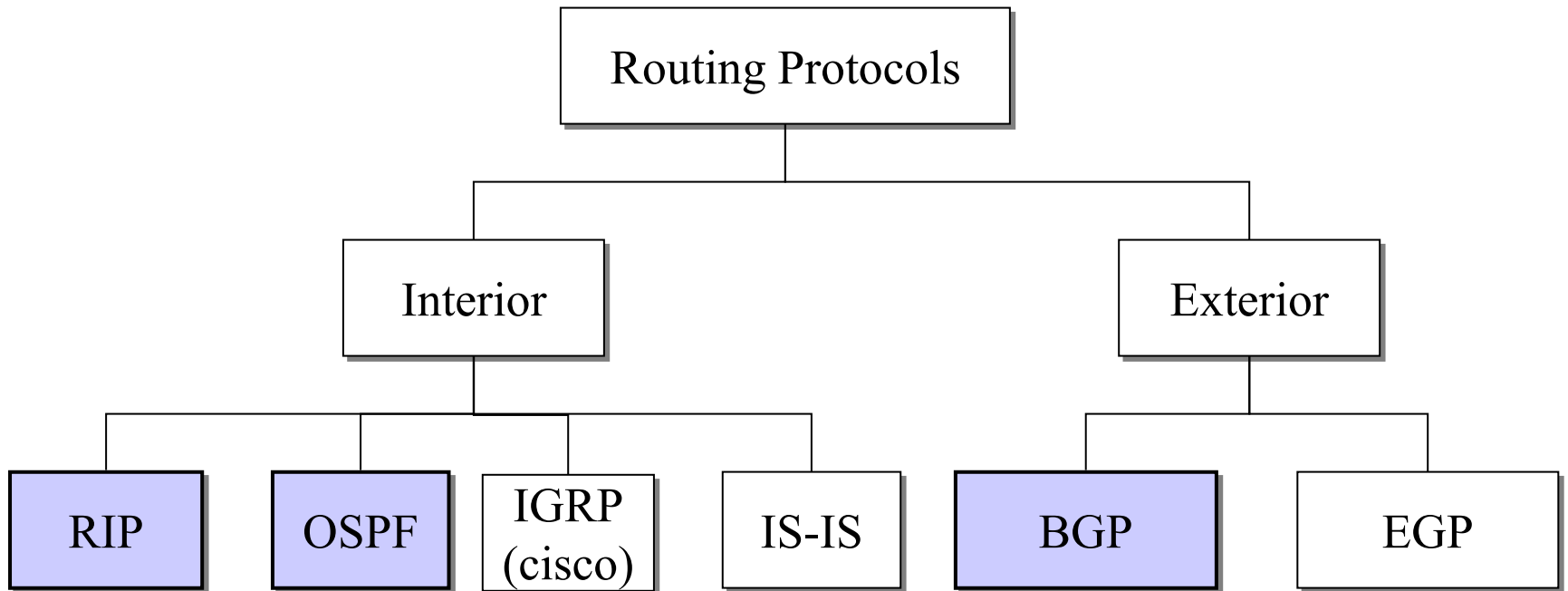
- The most fundamental functionality in a dynamic routing protocol:
 - Find the "best path" to a destination
- Two algorithms in use to find best path
 - Distance-Vector (Bellman-Ford)
 - Link-state (Dijkstra)
- But what is best path?
 - Interior routing: typically number of hops, or bandwidth
 - Exterior routing: business relations—peering
- Metrics
 - Number of hops (most common)
 - Bandwidth, Delay, Cost, Load, "Policies"

Aggregation and announcing prefixes

- Also called *summarization*
- The netid part of IPv4 addresses can be aggregated (summarized) into shorter prefixes.
 - Currently: over 600000 global prefixes
- Aggregation is often done manually
- Leads to smaller routing tables (fewer prefixes)
- Threats: multi-homing and load-balancing



Popular Routing Protocols

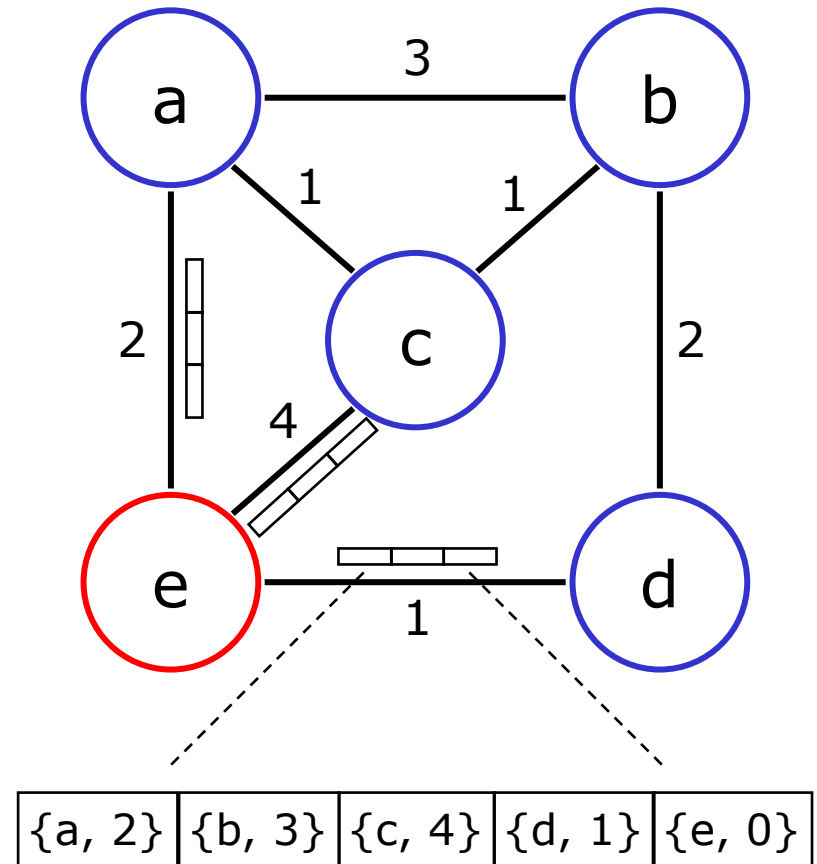


Routing Information Protocol - RIP

- RIP-1 (RFC 1058), RIP-2 (RFC 2453)
- Metric is Hop Counts
 - 1: directly connected
 - 16: infinity
 - RIP cannot support networks with diameter > 15 .
- RIP uses distance vector
 - RIP messages contain a vector of hop counts.
 - Every node sends its routes to its neighbours
 - Route information gradually spreads through the network
 - Every node selects the route with smallest metric.
- RIP messages are carried via UDP datagrams.
 - IP Multicast (RIP-2) or Broadcast (RIP-1)

Distance Vector

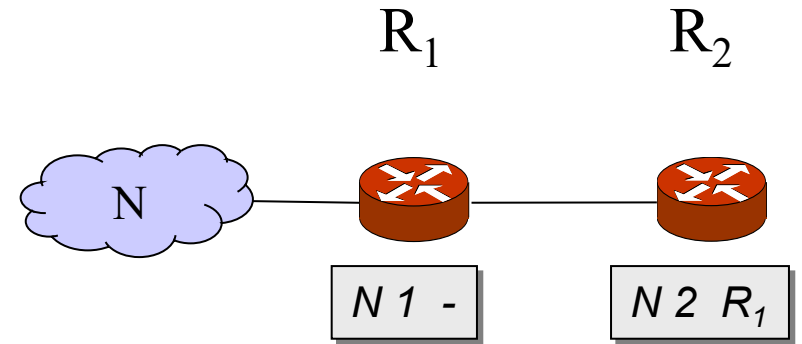
- A node advertizes its “distance-vector”
 - A list (vector) of all nodes that the node knows about
 - The distance to each of them
- Advertizements are sent to neighbours only
- Each neighbour updates its routing table and sends the new distance-vectors to its neighbours
 - Bellman-Ford algorithm



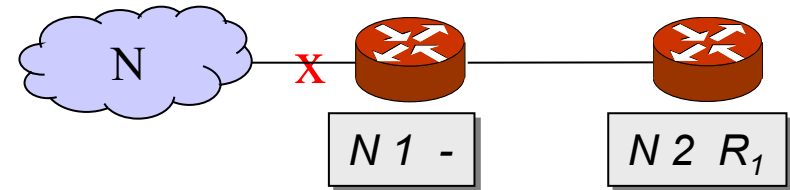
Distance-vector from "e"

RIP Problem: Count to Infinity

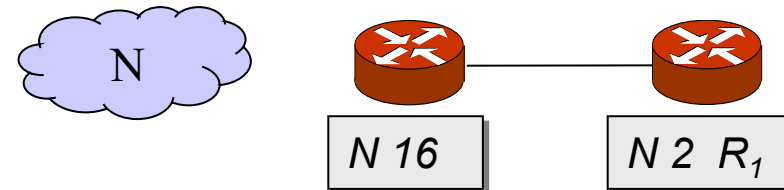
1. Initially, R_1 and R_2 both have a route to N with metric 1 and 2, respectively.



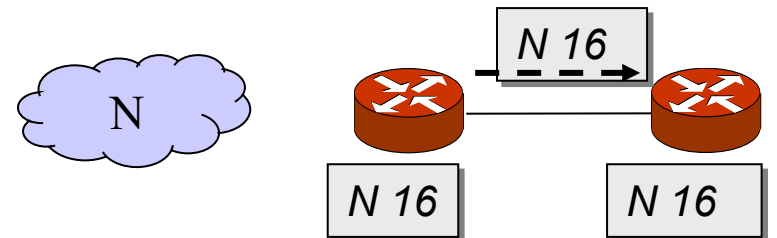
2. The link between R_1 and N fails.



3. Now R_1 removes its route to N , by setting its metric to 16 (infinity).

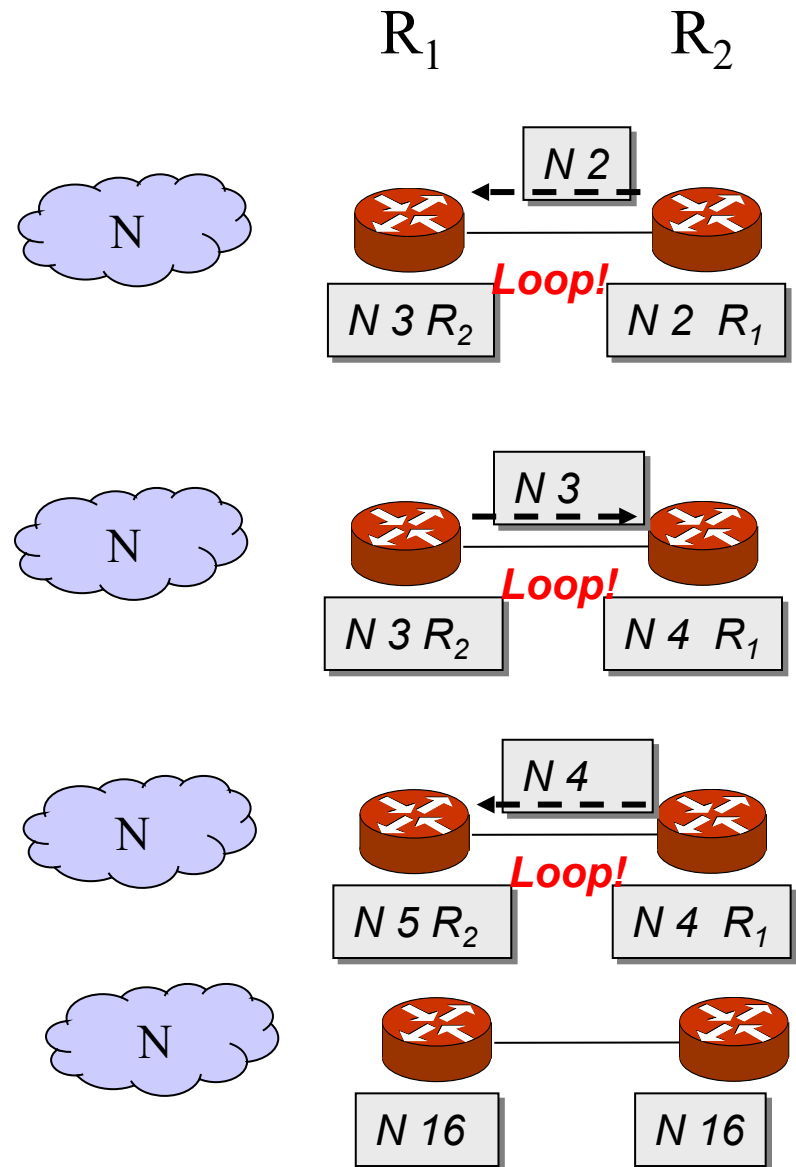


4. Now two things can happen: Either R_1 reports its route to R_2 . Everything is fine.



RIP Problem: Count to Infinity

5. The other alternative is that R_2 , which still has a route to N , advertises it to R_1 . Now things start to go wrong: packets to N are looped until their TTL expires!
6. Eventually (~ 10 - 20 s), R_1 sends an update to R_2 . The cost to N increases, but the loop remains.
7. Yet some time later, R_2 sends an update to R_1 .
- ...
13. Finally, the cost reaches infinity at 16, and N is unreachable. The loop is broken!

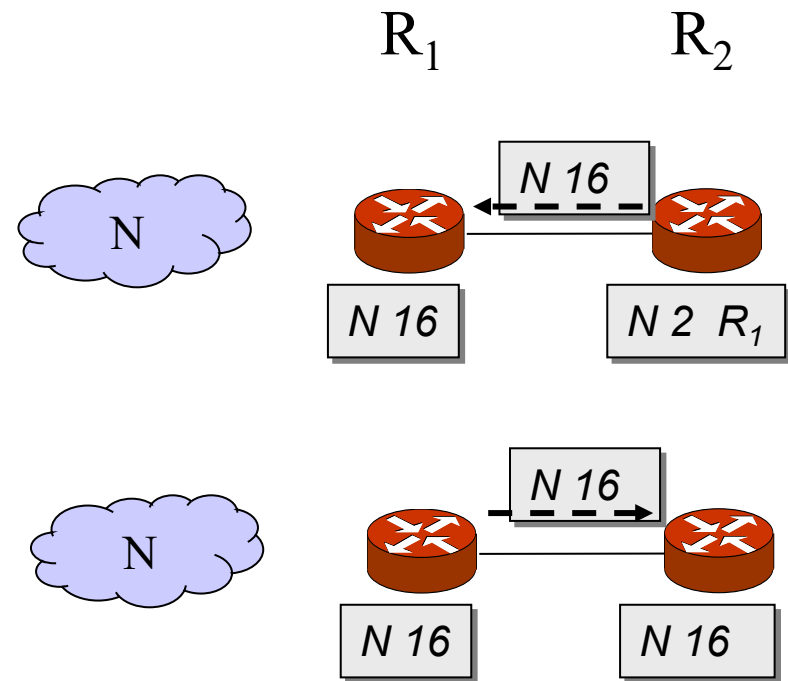


One Solution: Poison Reverse

- Advertise reverse routes with a metric of 16 (i.e., unreachable).

R_2 always announces an unreachable route to N to R_1 .

Eventually, R_1 reports its route to R_2 and everything is fine.



Disadvantages with RIP

- Slow convergence
 - Changes propagate slowly
 - Each neighbor only speaks ~every 30 seconds; information propagation time over several hops is long
- Instability
 - After a router or link failure RIP takes *minutes* to stabilize.
- Hops count may not be the best indication for which is the best route.
- The maximum useful metric value is 15
 - Network diameter must be less than or equal to 15.
- RIP uses lots of bandwidth
 - It sends the whole routing table in updates.

Why Use RIP?

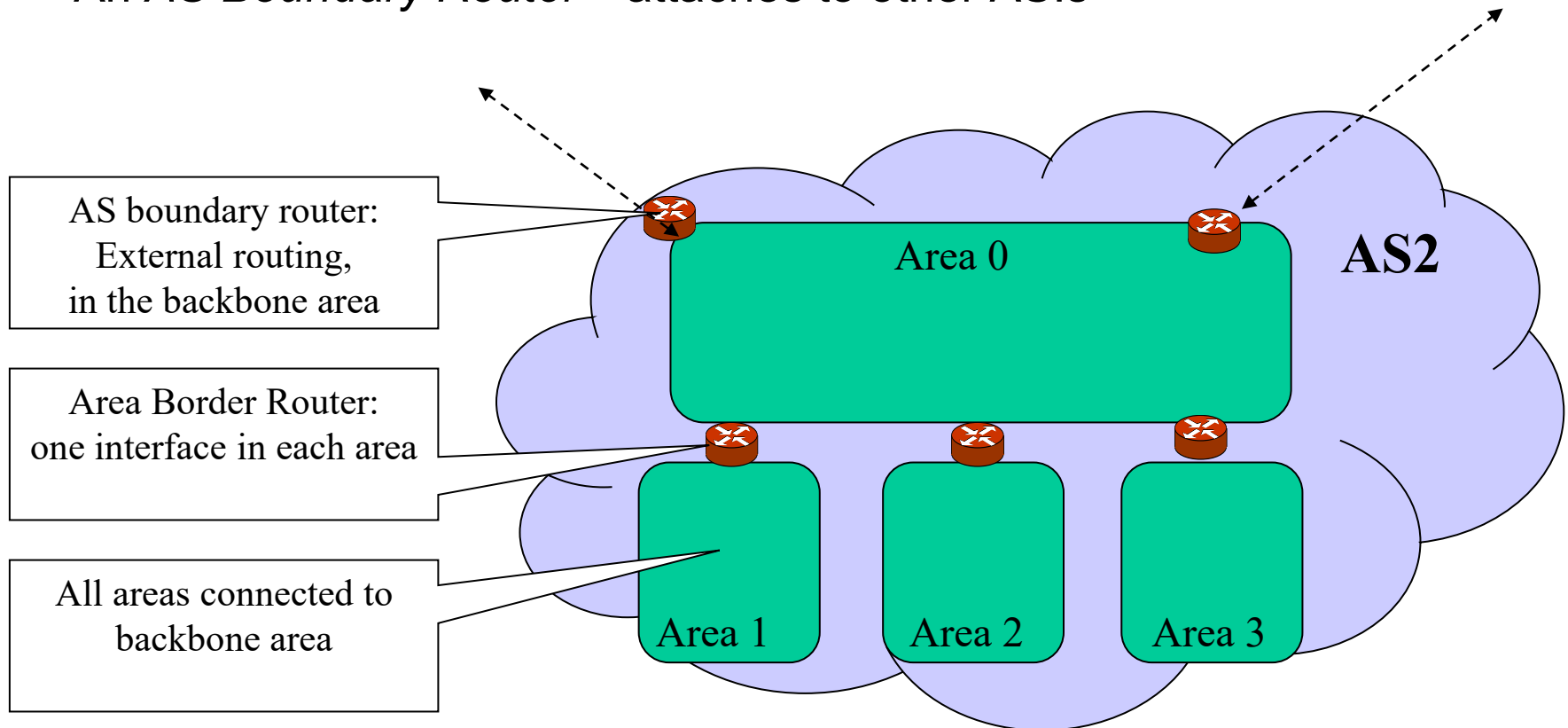
- After all these problems you might ask this question
- Answer
 - Because RIP is generally available
 - It is simple to configure.

Open Shortest Path First—OSPF

- OSPF version 2
 - RFC 2328
- OSPF is a link-state protocol.
 - Builds *Link State Advertisements* (LSAs)
 - Distributes LSAs to all other routers
 - Computes delivery tree using the *Dijkstra* algorithm
- OSPF uses IP *directly* (protocol field = 89)
 - Not UDP or TCP.
- OSPF networks are partitioned into *areas* to minimize cross-area communication.

OSPF Network Topology

- Area 0 is the *backbone* area. All traffic goes via the backbone.
- All other areas are connected to the backbone (1-level hierarchy)
- A *Border area router* has one interface in each area.
- An *AS Boundary Router*—attaches to other AS:s



Link-State Protocols (SPF)

- In SPF, every router does the following:
 1. Actively test the status of all neighbours/links
 2. Build a Link State Advertisement (LSA) from this information and propagate it to *all other* routers within an area.
 3. Using LSAs from all other routers, compute a shortest path delivery tree, typically using *Dijkstra shortest path algorithm*.
- Advantages (over distance-vector):
 - More functionality due to computation on original data and no dependence on intermediate routers
 - Full topology knowledge
 - Easier to Troubleshooting
 - Fast Convergence
- Disadvantage
 - uses more memory

OSPF Contains Three Protocols

1. The *Hello* protocol

- Check for neighbours, authentication, designated routers

2. The *Exchange* Protocol

- Exchange Link State Database between neighbours
- First get LSA headers
- Then transfer actual LSAs on request.

3. The *Flooding* protocol

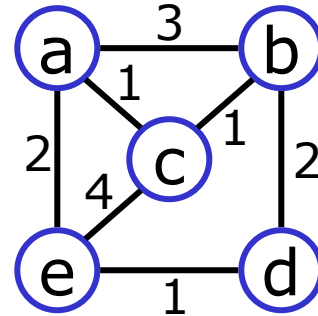
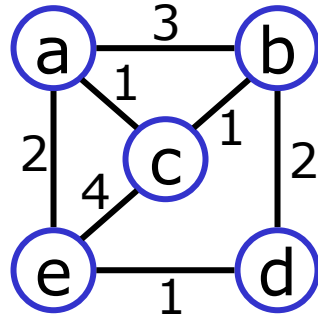
- When links change/age
- Send Link State updates to neighbours and flood *recursively*.
- If not seen before, propagate updates to all *adjacent* routers, except incoming

Distribution of Link State Advertisements

- Most complex and critical part of OSPF
- Initial topology transfer done with the exchange protocol.
- OSPF *floods* LSAs within an *area*
 - Recursively forward a new LSA to all neighbours (except the recipient)
 - An LSA will travel on all links exactly once
 - Uses sequence numbers and aging to avoid loops
- OSPF aggregates routes
 - Border Area Routers aggregates routes from an area into other areas.
 - AS Border Routers aggregates routes from other ASs.

Dijkstra Algorithm (Shortest Path First)

Find shortest paths from "a" to all other nodes!



M	D_b (path)	D_c (path)	D_d (path)	D_e (path)
{a}	3 (a-b)	1 (a-c)	∞ (--)	2 (a-e)
{a, c}	2 (a-c-b)	1 (a-c)	∞ (--)	2 (a-e)
{a, c, b}	2 (a-c-b)	1 (a-c)	4 (a-c-b-d)	2 (a-e)
{a, c, b, e}	2 (a-c-b)	1 (a-c)	3 (a-e-d)	2 (a-e)
{a, c, b, e, d}	2 (a-c-b)	1 (a-c)	3 (a-e-d)	2 (a-e)

Alternative to OSPF: IS-IS

- Link-State Routing
- Originally designed for Decnet and then CLNP (OSI)
- Has been stable for a longer time than OSPF
 - Large deployed base
 - Example: SUNET runs IS-IS
- More general hierarchies
 - Multiple levels in tree topology
 - Not strict two-levels as OSPF

Border Gateway Protocol—BGP

- Inter-domain routing
- Simple cases: *use static routing*
- Main purpose: Network reachability between autonomous systems
- BGP version 4 is *the* exterior routing protocol used in the Internet today.
- BGP uses TCP
 - TCP is reliable: reduces the protocol complexity
- BGP uses *path-vector* - enhancement of distance-vector.
- BGP implements *policies* – chosen by the local administrator.

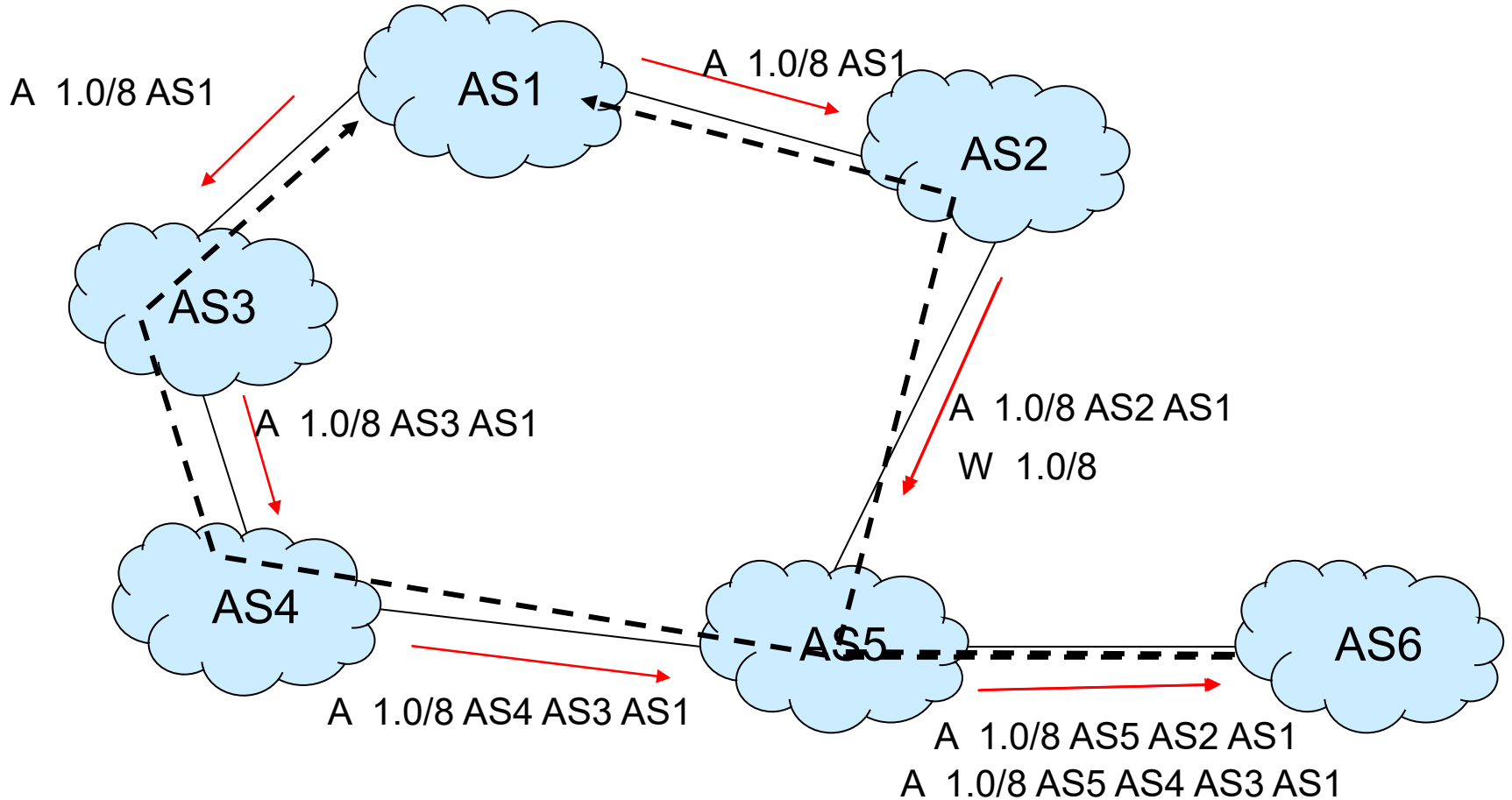
Autonomous Systems—RFC1930

- An *Autonomous system* is generally administered by a single entity.
 - Operators, ISPs (Internet Service Providers)
- An AS contains an arbitrary complex sub-structure.
- Each autonomous system selects the routing protocol to be used *within* the AS.
- Policies or updates within an AS are not propagated to other AS:s.
- An AS-number is (currently) a 16-bit unique identifier
- Interconnection between AS:s
 - Service Level Agreements (SLA:s)
 - Internet Exchange Points (IX:s)
 - Network Access Points (NAPs)

AS Number	Network
3	MIT
32	STANFORD
2839	KTH
1653	SUNET

BGP Simple example

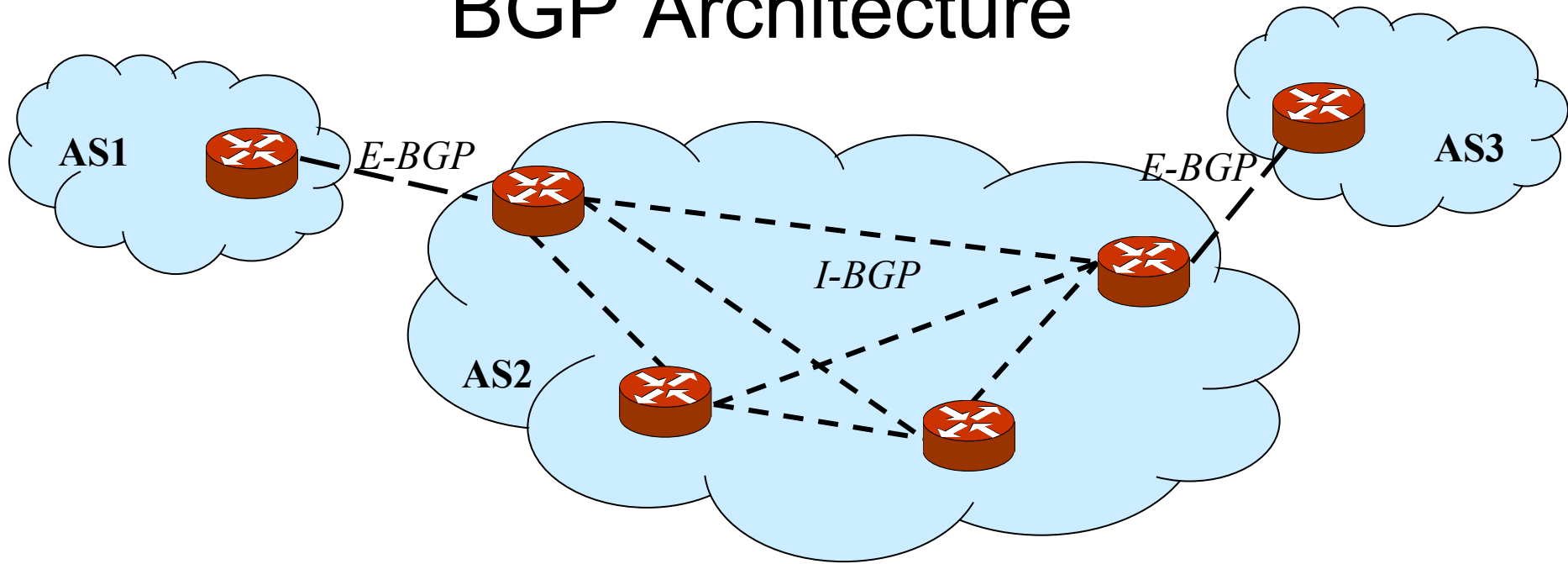
- AS1 has a network 1.0.0.0/8 that it announces



Motivation for Path-Vector

- Distance-vector
 - Hop-count too limited
 - Unstable
- Link-State
 - Link state database would be enormous
- Path-vector extends distance-vector
 - Instead of a simple cost, assign *an AS-Path* to every route
 - There may be many paths to the same destination (network *prefix*)
 - AS-Path used to implement *policies* and *loop prevention*

BGP Architecture



- BGP interacts with the internal routing (OSPF/IS-IS/RIP/...)
 - Redistributes routes between the two domains
- BGP really consists of two protocols:
 - E-BGP: coordinates between border routers *between* AS:s
 - I-BGP : coordinates between BGP peers *within* an AS

BGP Router Operation

- A BGP router receives routes
 - BGP peers (E-BGP)
 - Redistribution: IGP/static routes
- It aggregates routes
- It filters and modifies routes
 - According to some *policy*
- It advertizes routes to its EBGP neighbours in other AS:s