RIP

Signpost Routing, Version 1

Routing Information Protocol

- Interior Gateway Protocol (IGP)
- Distance-Vector Routing Protocol
  - Bellman Ford Algorithm
  - RFC 1058 released in 1988
- Classful
  - No subnet masks carried
- Distributed through BSD UNIX 4.2 in 1982 (routed)

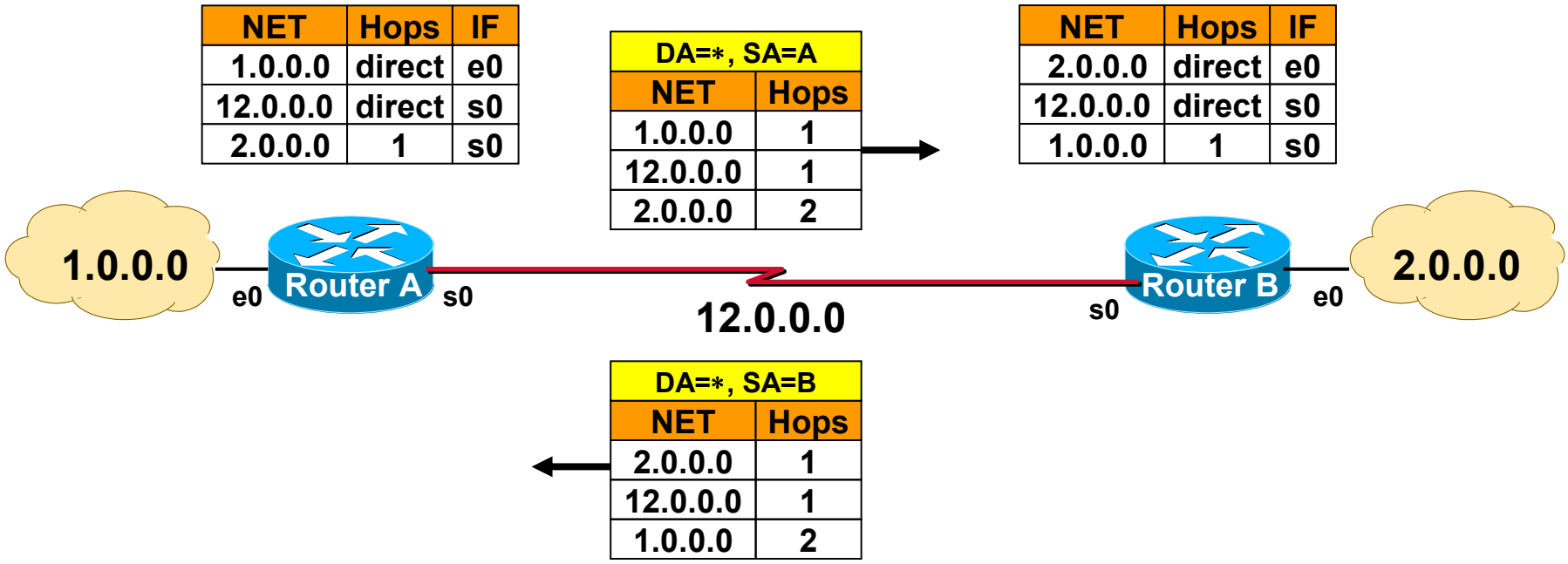
RIP Basics

- Signpost principle
  - Own routing table is sent periodically (every 30 seconds)
- Receiver of update extracts new information
  - Known routes with worse metric are ignored
- What is a signpost made of ?
  - Destination network
  - Hop Count (metric, "distance")
  - Next Hop ("vector", given implicitly by sender's address!)

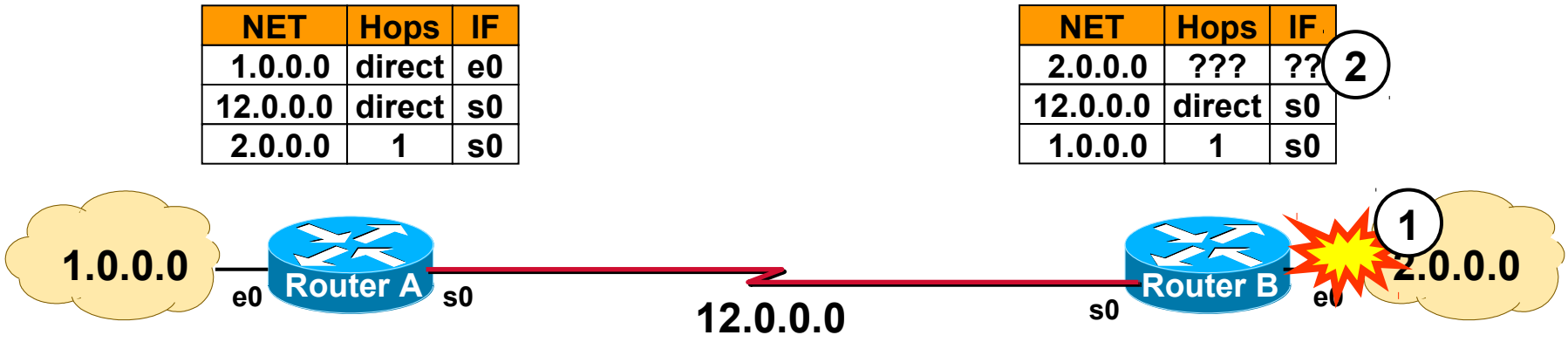
"Routing By Rumour"

- Good news propagate quickly
  - 30 seconds per network
- Bad news are ignored
  - Except when sent by routers from which these routes had been learned initially
  - But better news from ANY router will be preferred
- Unreachable messages propagate slowly
  - 180 seconds per network

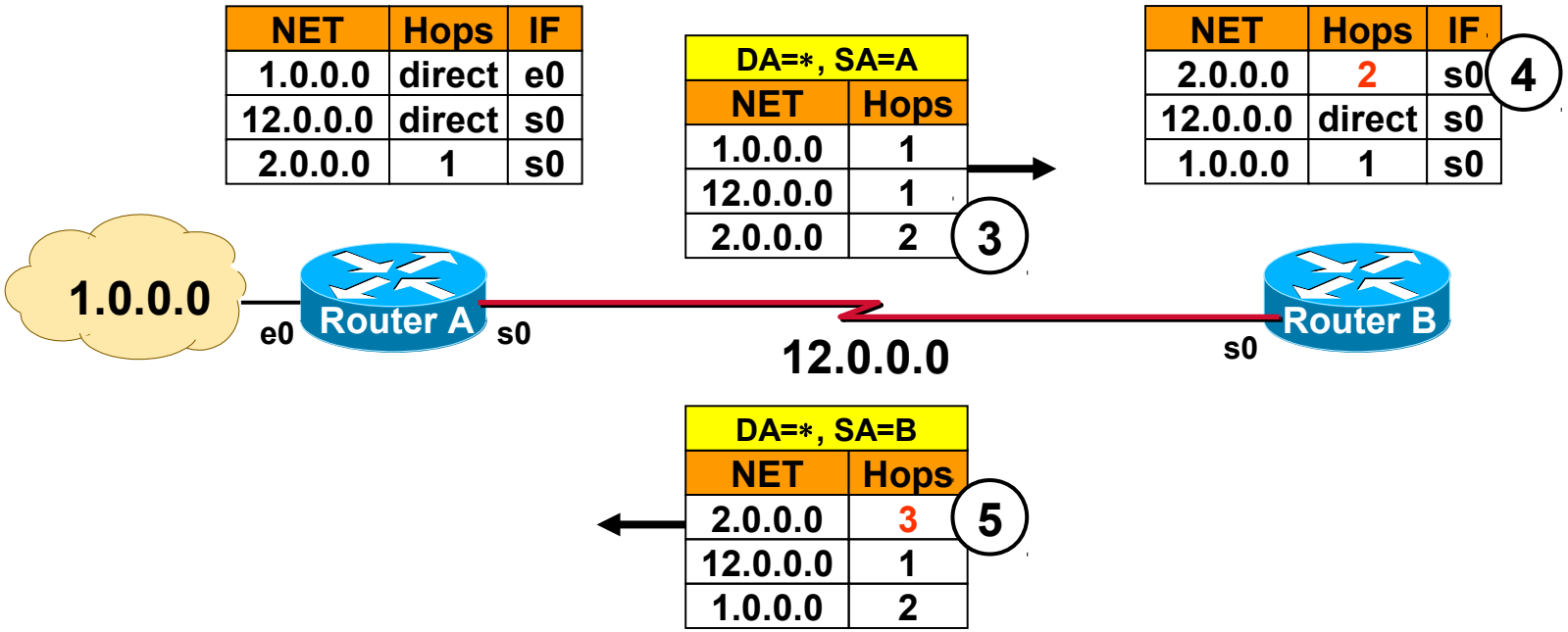
Without Split Horizon (1)



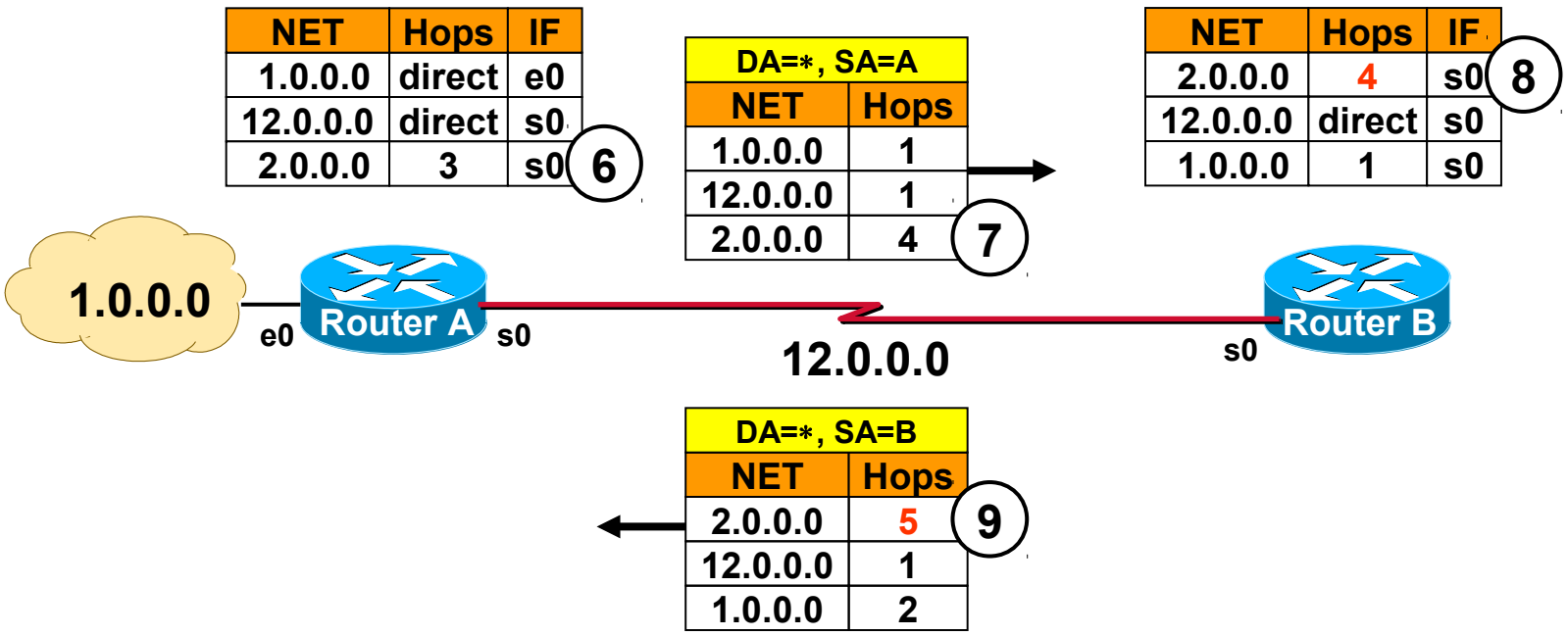
Without Split Horizon (2)



Without Split Horizon (3)



Without Split Horizon (4)



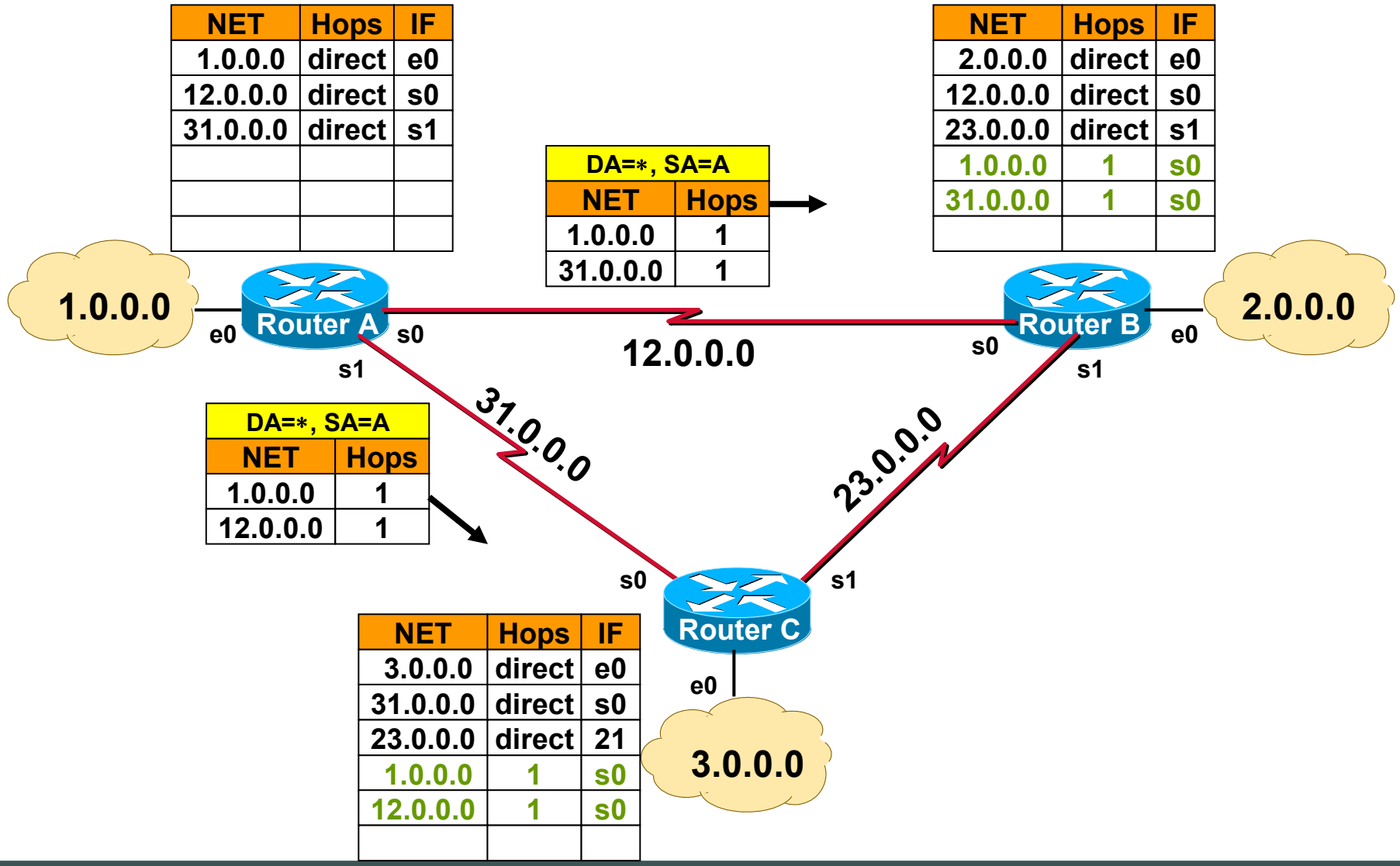
...Count to Infinity...

During count to infinity packets to network 2.0.0.0 are caught in a routing loop

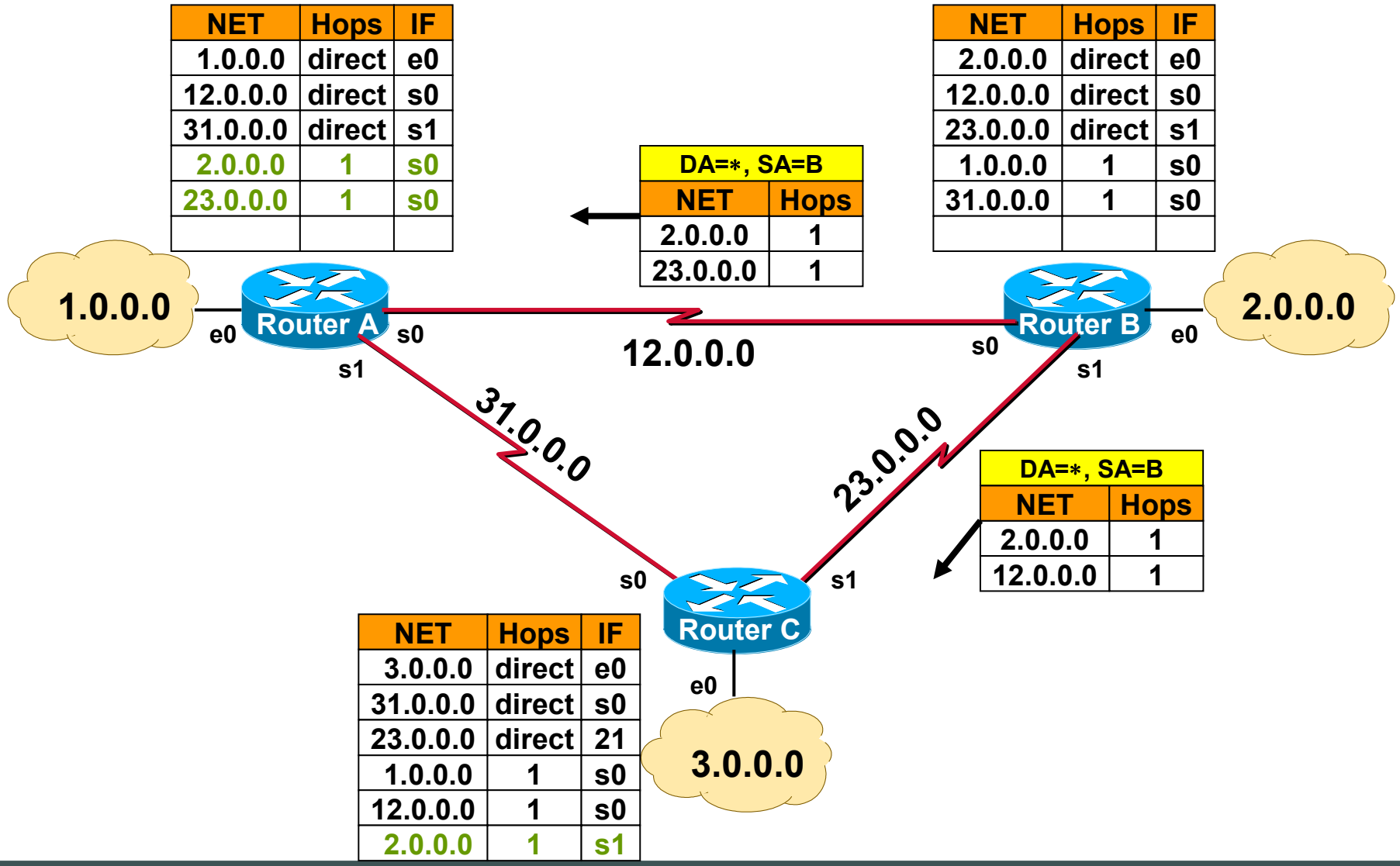
Split Horizon

- A router will *not* send information about routes through an *interface* over which the router has *learned* about those routes
  - Exactly THIS is split horizon
- Idea: "Don't tell neighbor of routes that you learned from this neighbor"
  - That's what humans (almost) always do: *Don't tell me what I've told you !*
- Cannot 100% avoid routing loops!

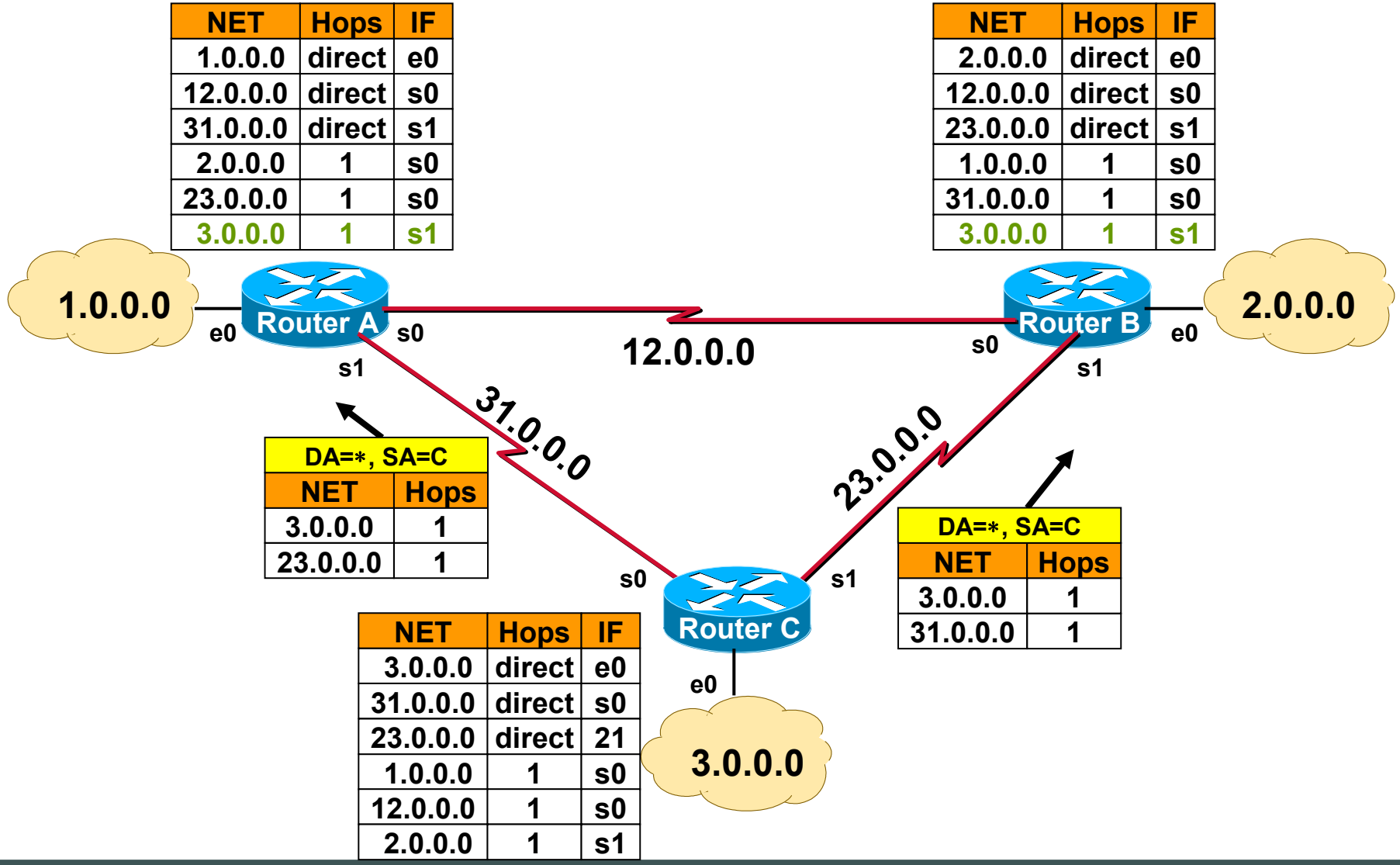
RIP At Work (A)


NET | Hops | IF
1.0.0.0 | direct | e0
12.0.0.0 | direct | s0
31.0.0.0 | direct | s1
DA=*, SA=A
NET | Hops
1.0.0.0 | 1
31.0.0.0 | 1
NET | Hops | IF
2.0.0.0 | direct | e0
12.0.0.0 | direct | s0
23.0.0.0 | direct | s1
1.0.0.0 | 1 | s0
31.0.0.0 | 1 | s0
1.0.0.0
e0
Router A
s0
s1
12.0.0.0
Router B
s0
e0
2.0.0.0
31.0.0.0
23.0.0.0
DA=*, SA=A
NET | Hops
1.0.0.0 | 1
12.0.0.0 | 1
s0
s1
Router C
e0
3.0.0.0
NET | Hops | IF
3.0.0.0 | direct | e0
31.0.0.0 | direct | s0
23.0.0.0 | direct | 21
1.0.0.0 | 1 | s0
12.0.0.0 | 1 | s0

RIP At Work (B)


NET | Hops | IF
1.0.0.0 | direct | e0
12.0.0.0 | direct | s0
31.0.0.0 | direct | s1
2.0.0.0 | 1 | s0
23.0.0.0 | 1 | s0
DA=*, SA=B
NET | Hops
2.0.0.0 | 1
23.0.0.0 | 1
NET | Hops | IF
2.0.0.0 | direct | e0
12.0.0.0 | direct | s0
23.0.0.0 | direct | s1
1.0.0.0 | 1 | s0
31.0.0.0 | 1 | s0
1.0.0.0
e0
Router A
s0
s1
12.0.0.0
31.0.0.0
23.0.0.0
Router B
s0
s1
e0
2.0.0.0
DA=*, SA=B
NET | Hops
2.0.0.0 | 1
12.0.0.0 | 1
s0
s1
Router C
e0
3.0.0.0
NET | Hops | IF
3.0.0.0 | direct | e0
31.0.0.0 | direct | s0
23.0.0.0 | direct | 21
1.0.0.0 | 1 | s0
12.0.0.0 | 1 | s0
2.0.0.0 | 1 | s1

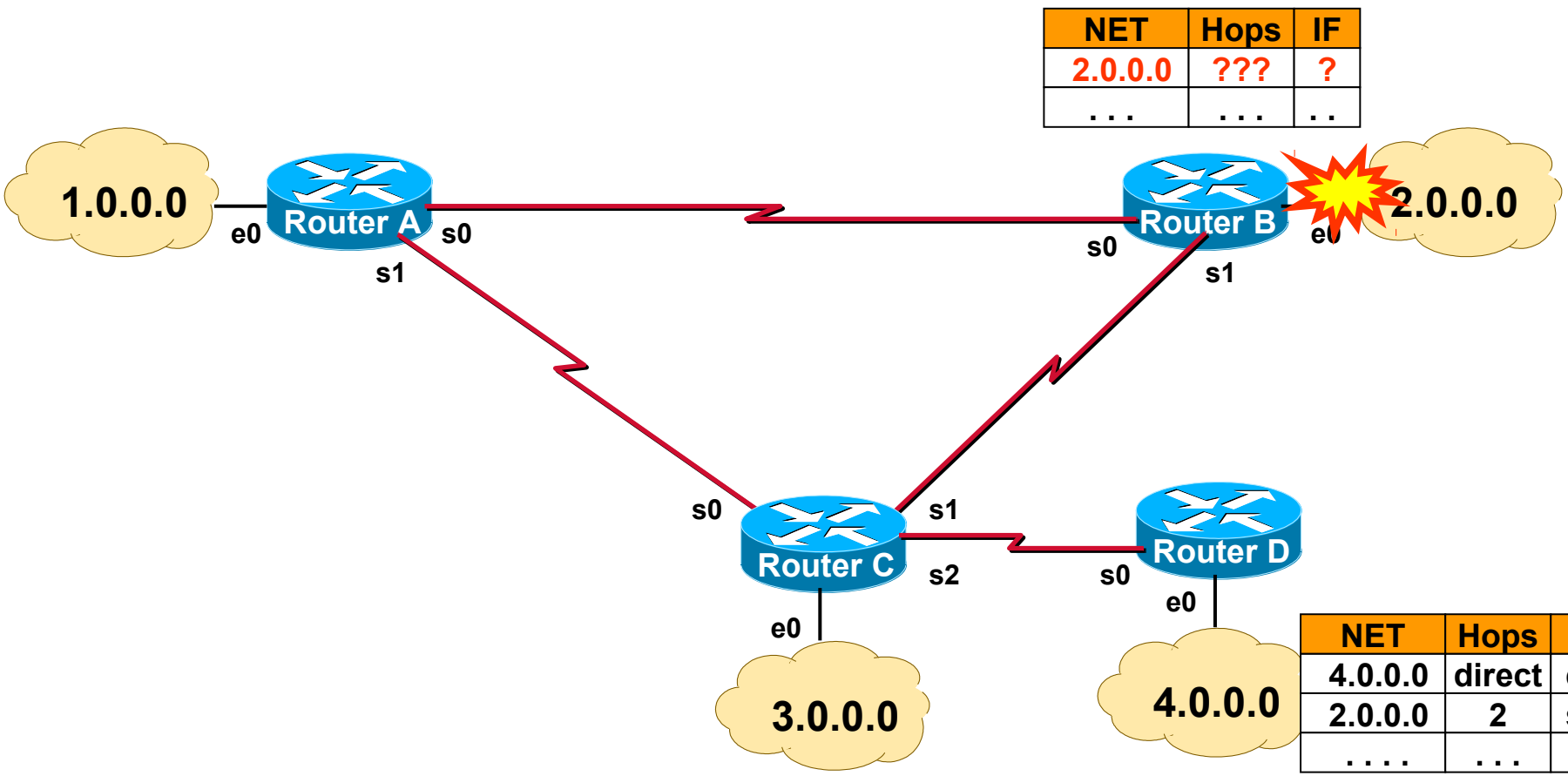
RIP At Work (C)


NET | Hops | IF
1.0.0.0 | direct | e0
12.0.0.0 | direct | s0
31.0.0.0 | direct | s1
2.0.0.0 | 1 | s0
23.0.0.0 | 1 | s0
3.0.0.0 | 1 | s1
NET | Hops | IF
2.0.0.0 | direct | e0
12.0.0.0 | direct | s0
23.0.0.0 | direct | s1
1.0.0.0 | 1 | s0
31.0.0.0 | 1 | s0
3.0.0.0 | 1 | s1
1.0.0.0
Router A
e0
s0
s1
12.0.0.0
Router B
s0
e0
s1
2.0.0.0
31.0.0.0
23.0.0.0
DA=*, SA=C
NET | Hops
3.0.0.0 | 1
23.0.0.0 | 1
DA=*, SA=C
NET | Hops
3.0.0.0 | 1
31.0.0.0 | 1
s0
s1
Router C
e0
3.0.0.0
NET | Hops | IF
3.0.0.0 | direct | e0
31.0.0.0 | direct | s0
23.0.0.0 | direct | 21
1.0.0.0 | 1 | s0
12.0.0.0 | 1 | s0
2.0.0.0 | 1 | s1

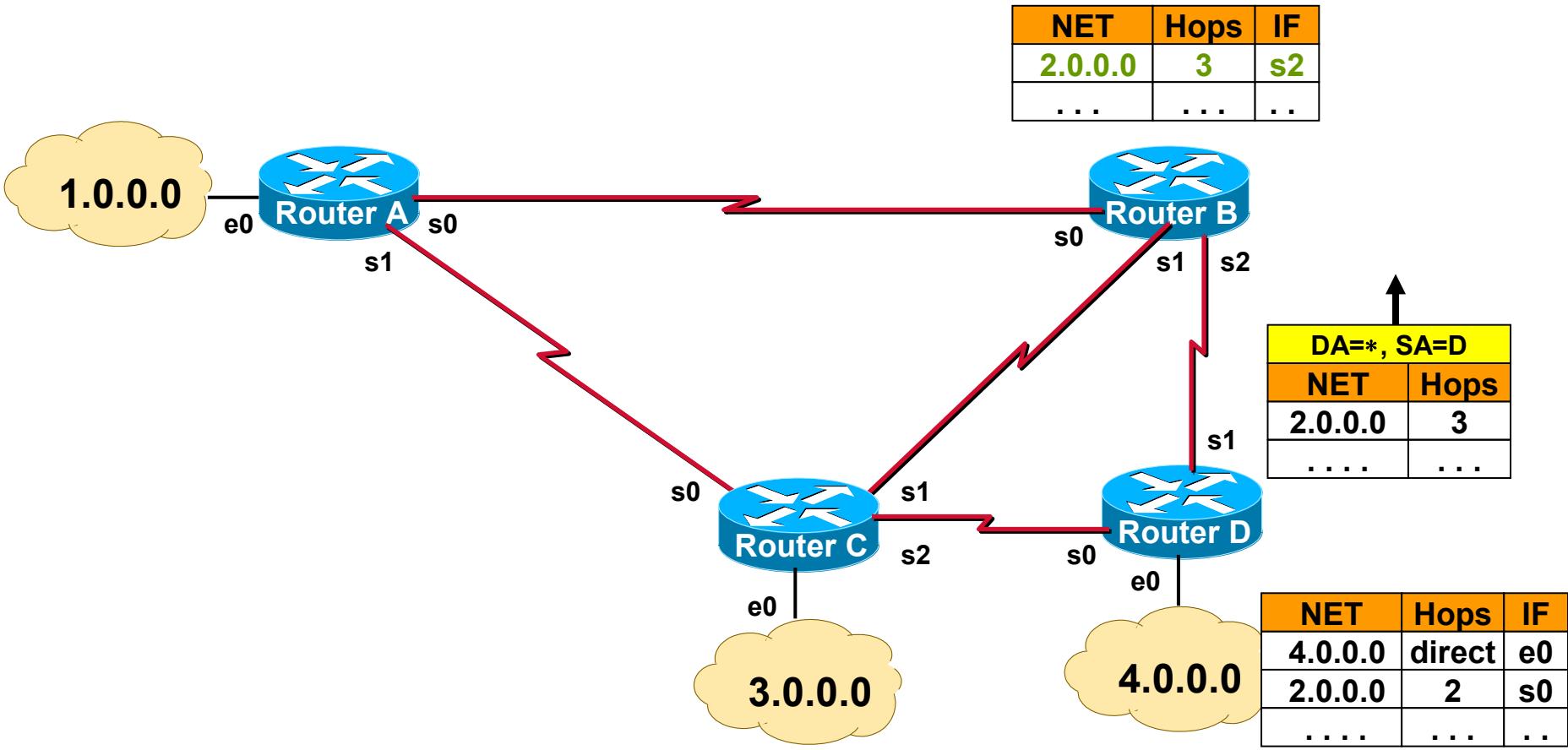
Count To Infinity

- Main problem with distance vector protocols
- Unforeseeable situations can lead to count to infinity
  - Access lists
  - Disconnection and connections
  - Router malfunctions
  - ....
- During that time, routing loops occur!

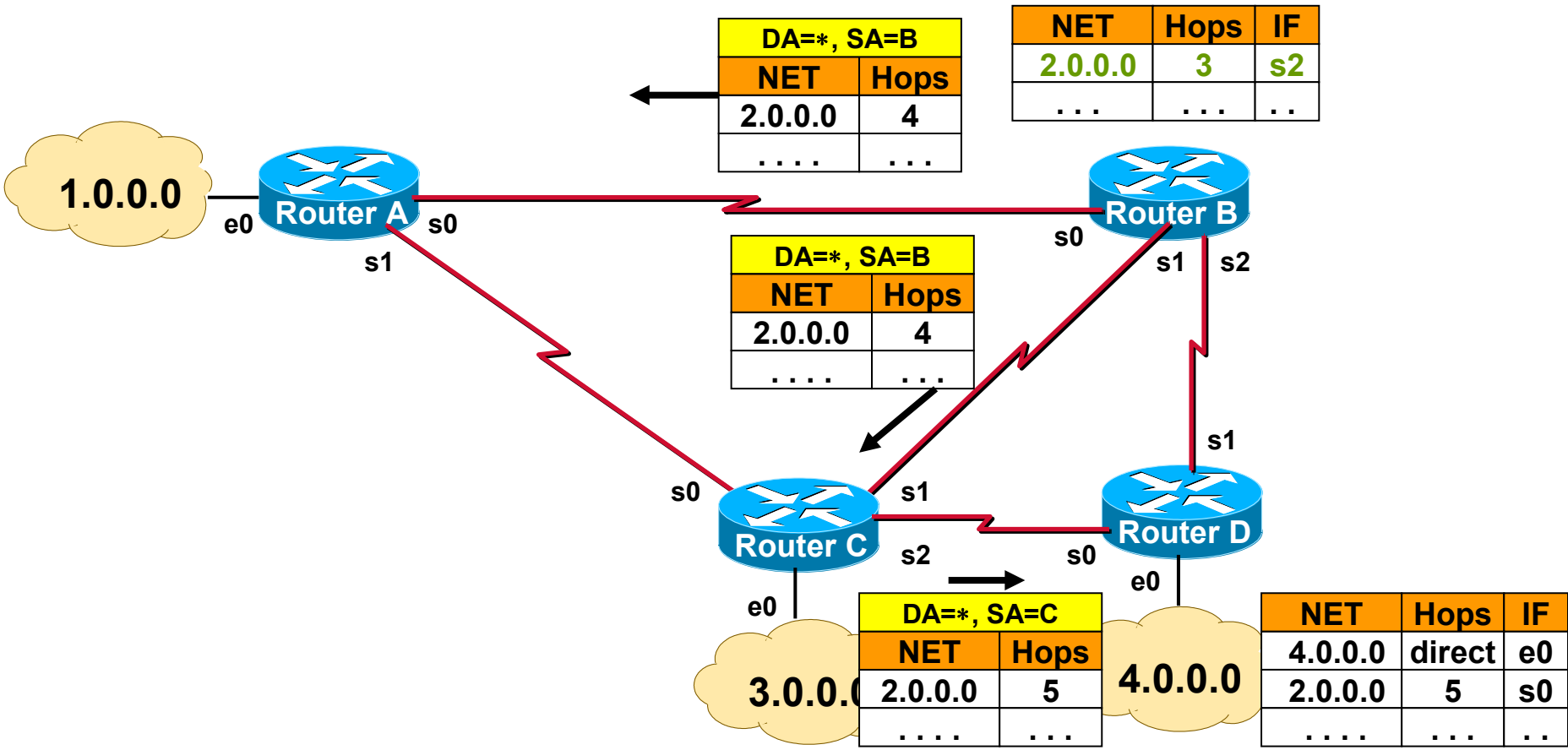
Count To Infinity (1)



Count To Infinity (2)



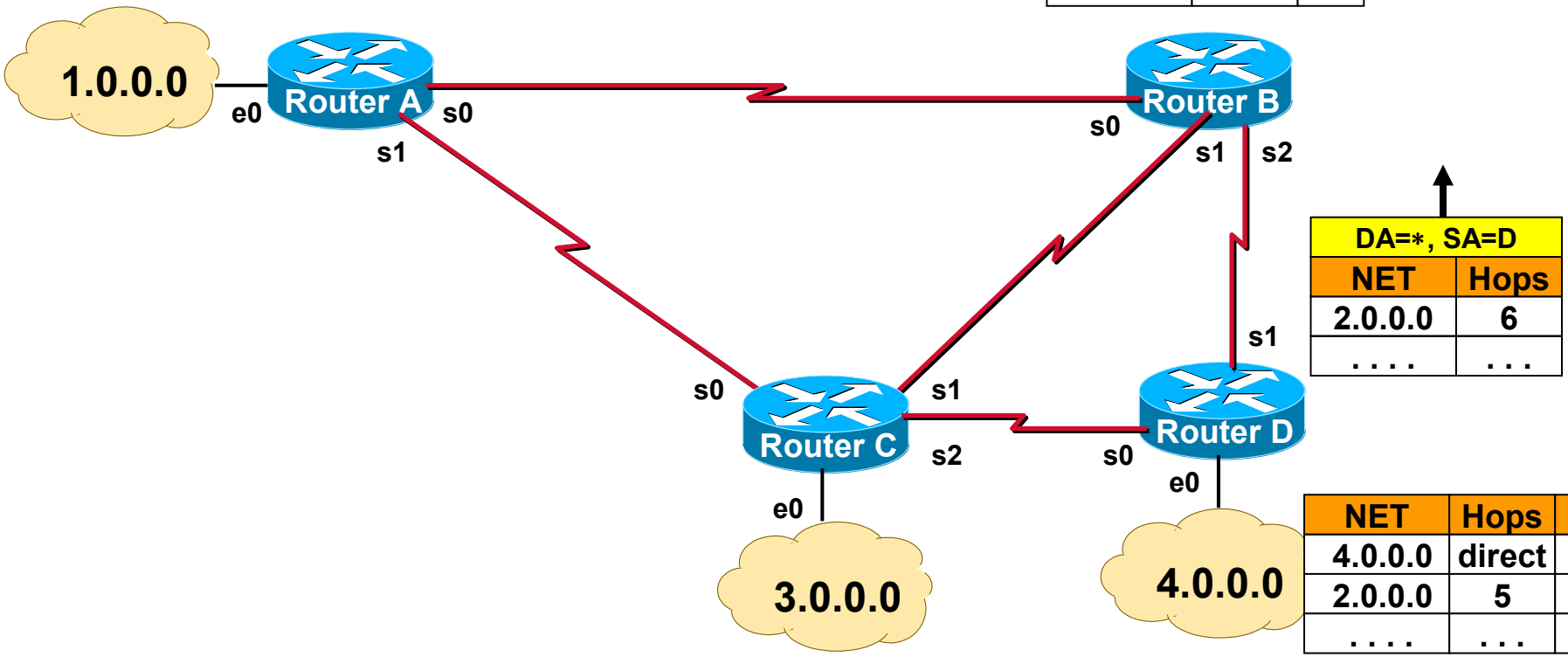
Count To Infinity (3)



Count To Infinity (4)

Count to Infinity situations cannot be avoided in any situation (drawback of signpost principle)
Basic solution: Maximum Hop Count = 16

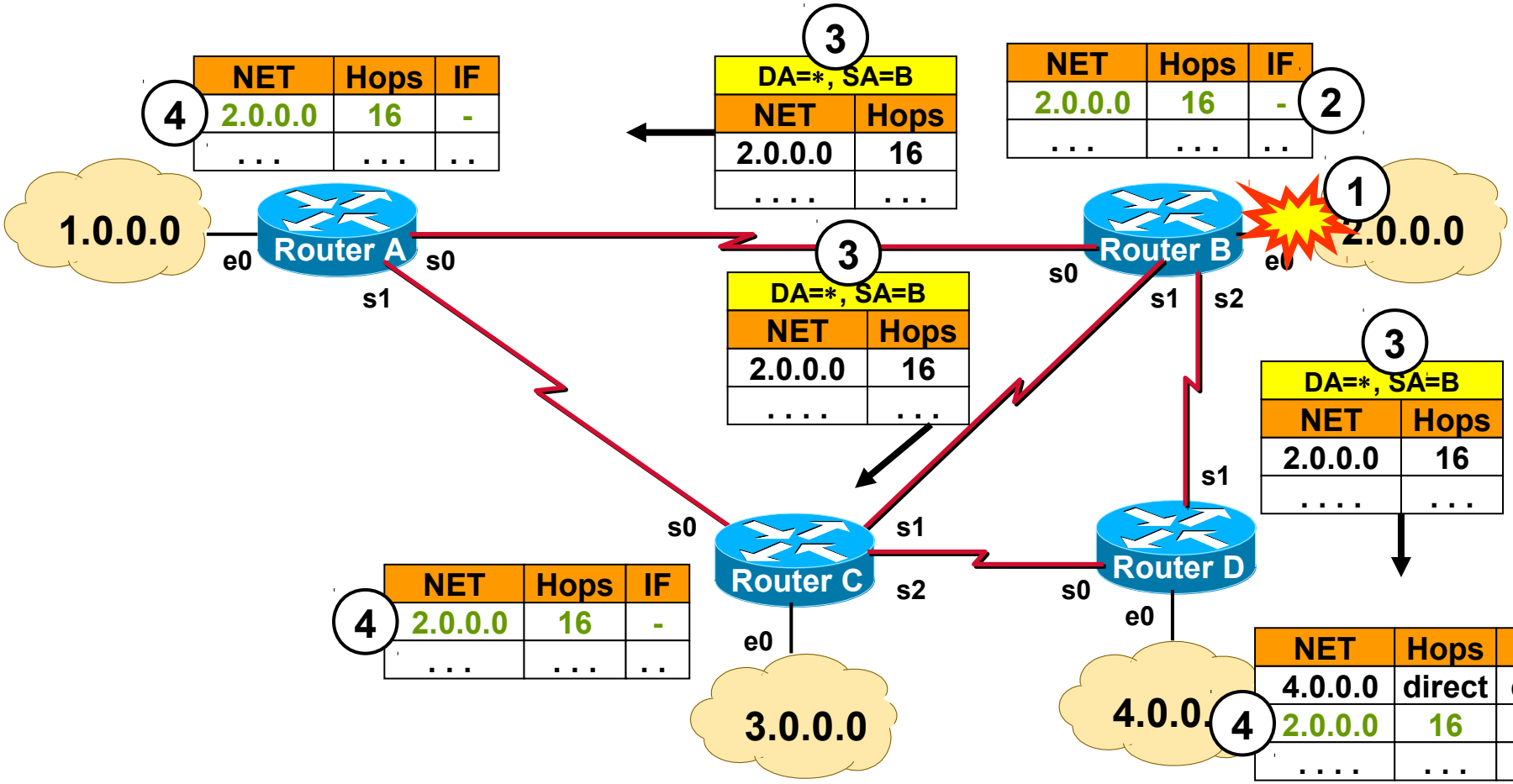
NET	Hops	IF
2.0.0.0	6	s2
. . .	. . .	. .



NET	Hops	IF
4.0.0.0	direct	e0
2.0.0.0	5	s0
. . . .	. . .	. .

Maximum Hop Count = 16

Upon network failure, the route is marked as INVALID (hop count 16) and propagated.



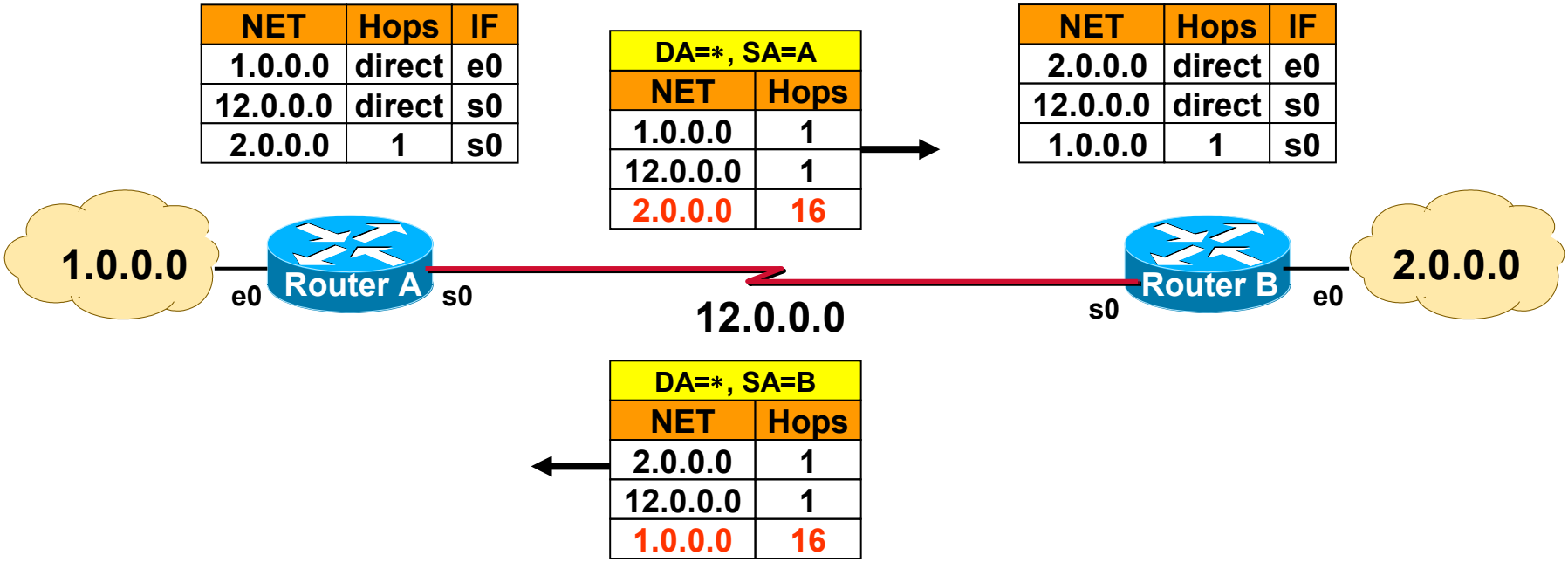
Maximum Hop Count

- Defining a maximum hop count of 16 provides a basic safety factor
- But restricts the maximum network diameter
- Routing loops might still exist during 480 seconds (16×30s)
- Therefore several other measures necessary

Additional Measures

- Split Horizon
  - Suppressing information that the other side should know better
  - Used during normal operation but cannot prevent routing loops !!!
- Split Horizon with Poison Reverse
  - Declare learned routes as unreachable
  - "Bad news is better than no news at all"
  - Stops potential loops due to corrupted routing updates

Split Horizon With Poison Reverse



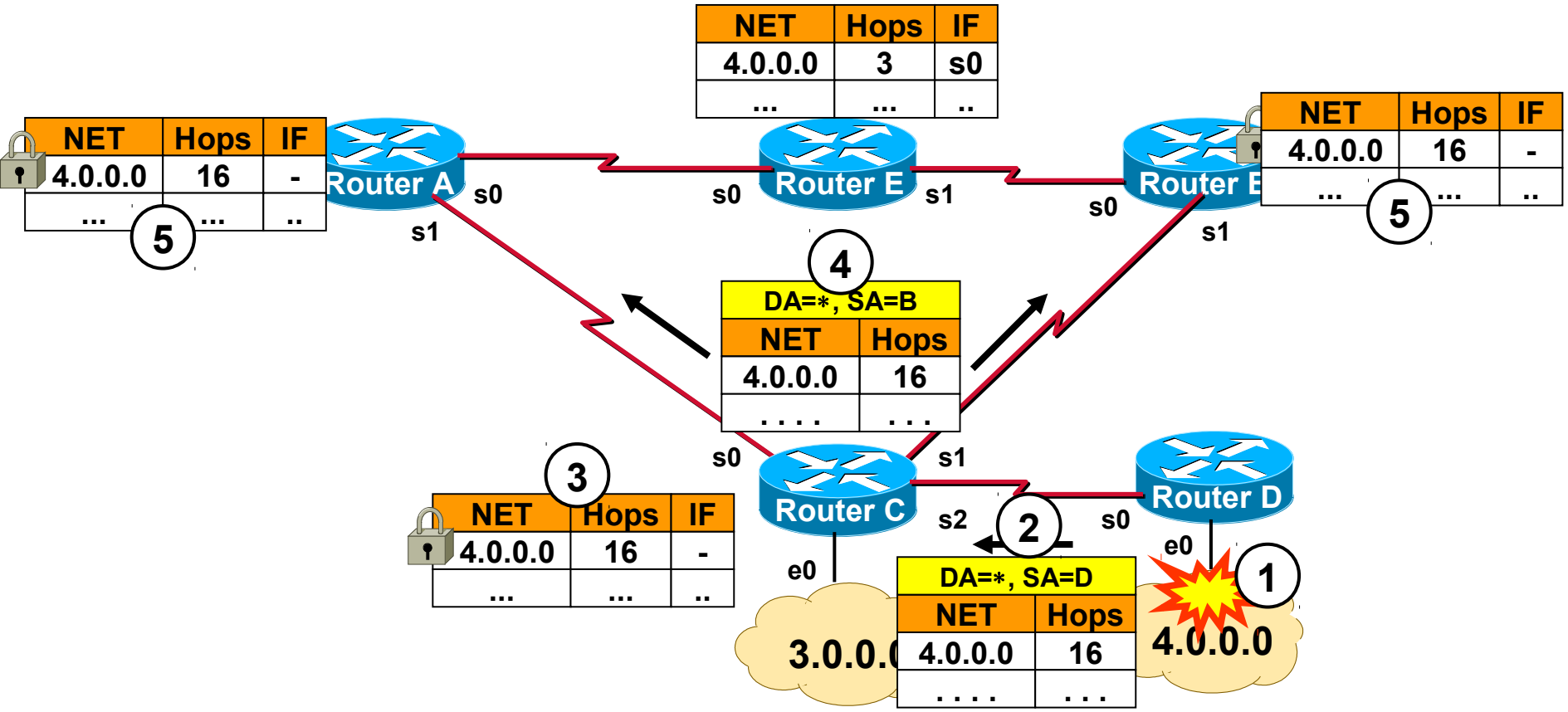
Note: poison reverse overrides split horizon when a network is lost

Additional Measures

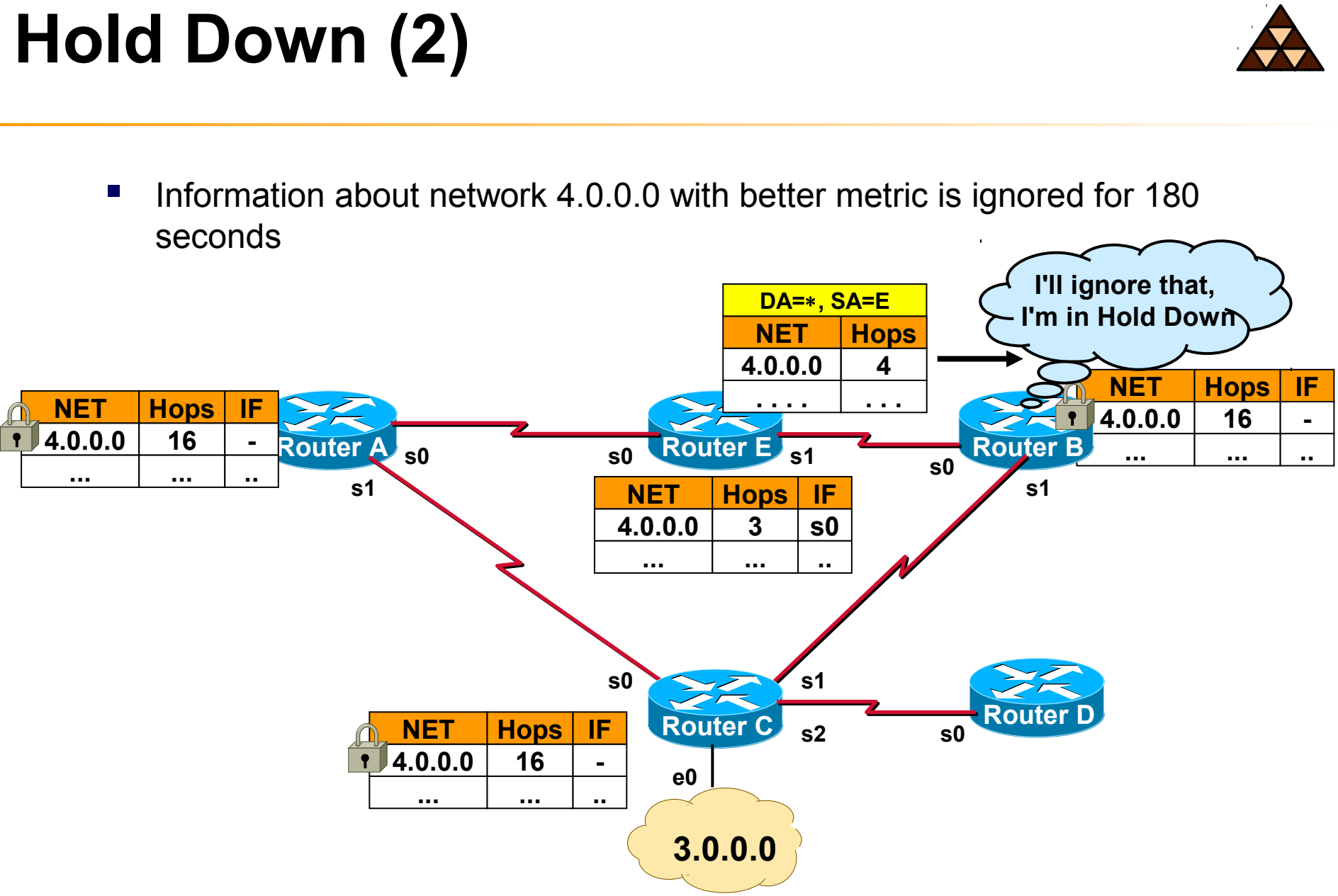
- Remember: good news overwrite bad news
  - Unreachable information could be overwritten by uninformed routers (which are beyond scope of split horizon)
- Hold Down
  - Guarantees propagation of bad news throughout the network
  - Routers in hold down state ignore good news for 180 seconds

Hold Down (1)

- Router C receives unreachable message (4.0.0.0, 16) from router D
- Router C declares 4.0.0.0 as invalid (16) and enters hold-down state

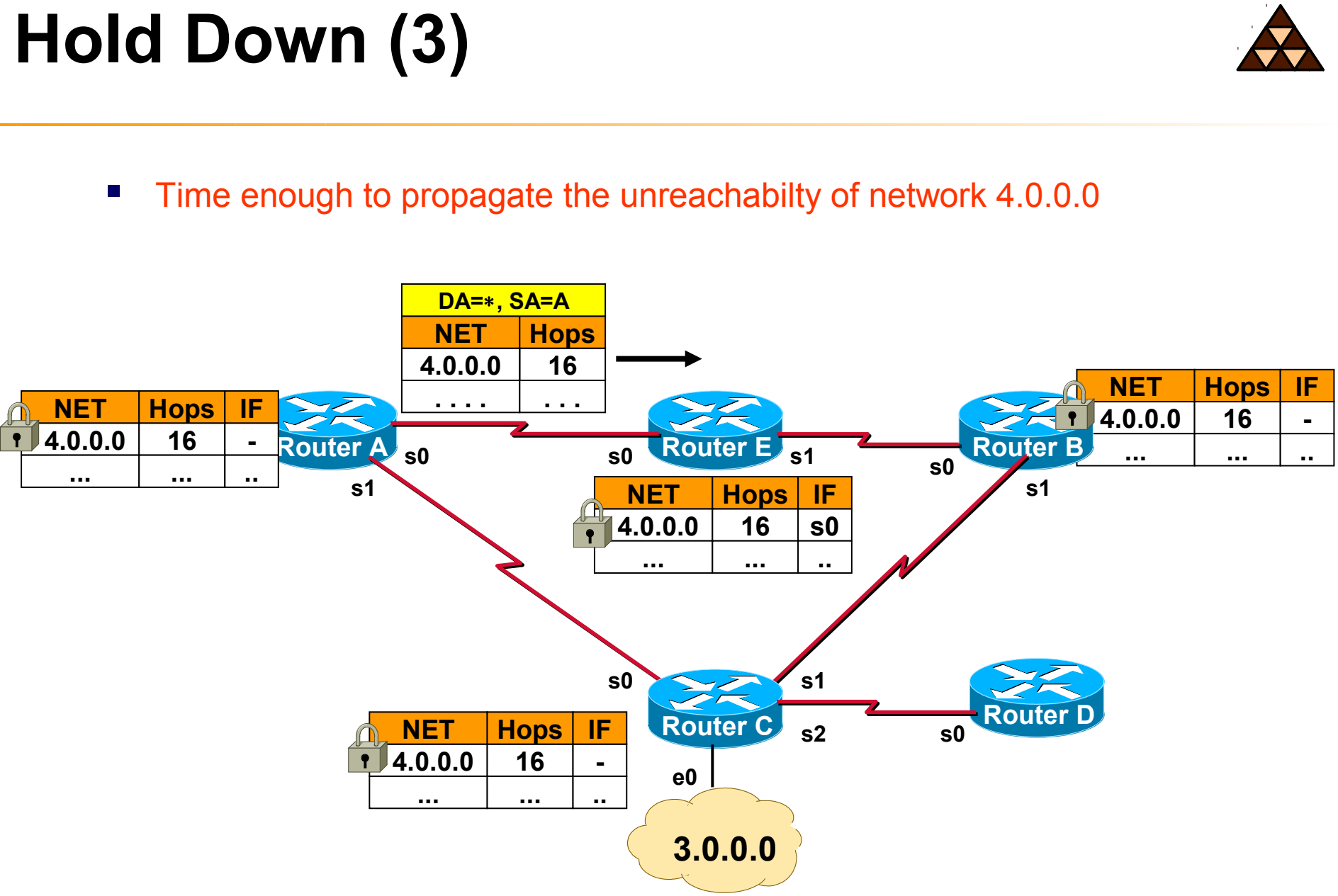


Hold Down (2)



- Information about network 4.0.0.0 with better metric is ignored for 180 seconds

Hold Down (3)



- Time enough to propagate the unreachabilty of network 4.0.0.0

Triggered Update

- To reduce convergence time, routing updates are sent immediately upon events (changes)
- On receiving a different routing update a router should also send immediately an update
  - Called triggered update

RIP Timers Summary

- UPDATE (30 seconds)
  - Period to send routing update
- INVALID (180 seconds)
  - Aging time before declaring a route invalid ("16") in the routing table
- HOLDDOWN (180 seconds)
  - After a route has been invalided, how long a router will wait before accepting an update with better metric
- FLUSH (240 seconds)
  - Time before a non-refreshed routing table entry is removed

Message Format

Command	Version	Must be zero
Address Family Identifier		Must be zero
IP Address		
Must be zero		
Must be zero		
Metric		
Address Family Identifier		Must be zero
IP Address		
Must be zero		
Must be zero		
Metric		
. . . . .		

Up to 25 route entries

RIP Messages

- Request (command = 1)
  - Ask neighbor to send response containing all or part of the routing table
  - Typically used at startup only
- Response (command = 2)
  - THE Routing Update
  - Typically sent every 30 seconds without explicit request

Details

- RIP message is sent within UDP payload
  - UDP Port 520, both source and destination port
  - Maximum message size is 512 bytes
- L2 Broadcast + IP Broadcast
  - Because we do not know neighbor router addresses
  - On shared media one update is sufficient
- Version = 1
- Address family for IP is 2

Timer Synchronization

- In case of many routers on a single network
  - Processing load might affect update timer
  - Routers might get synchronized
  - Collisions occur more often
- Therefore either use
  - External timer
  - Or add a small random time to the update timer
    (30 seconds + RIP_JITTER = 25...35 seconds)

RIP Disadvantages

- Big routing traffic overhead
  - Contains nearly entire routing table
  - WAN links (!)
- Slow convergence
- Small network diameter
- No discontiguos subnetting
- Only equal-cost load balancing supported
  - (if you are lucky)

Summary

- First important distance vector implementation (not only for IP)
- Main problem: Count to infinity
  - Maximum Hop Count
  - Split Horizon
  - Poison Reverse
  - Hold Down
- Classless, Slow, Simple

Quiz

- How could slower gateways/links be considered for route calculation
- Wouldn't TCP be more reliable than UDP?
- Does maximum hop-count mean that I can only have 15 net-IDs ?