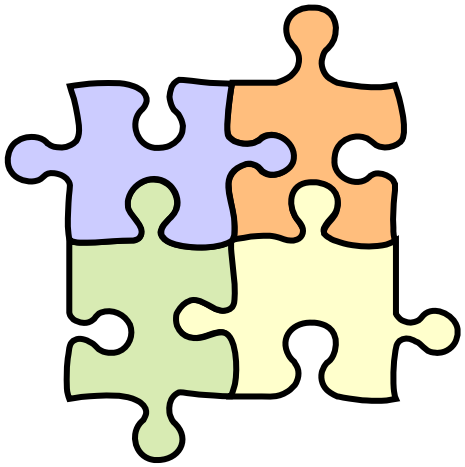


OSPF – Introduction

The IETF Routing Master

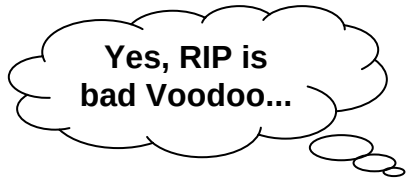
Part 1

“Dijkstra probably hates me”

Linus Torvalds in kernel/sched.c

"Open Shortest Path First"

- Official (IETF) successor of RIP
  - RIP is slow
  - RIP is unreliable
  - RIP produces too much routing traffic
  - RIP only allows 15 hop routes
- OSPF is a link-state routing protocol
  - Inherently fast convergence
  - Designed for large networks
  - Designed to be reliable



OSPF Background

- OSPF is the IGP recommended by the IETF
- "Open" means "not proprietary"
- Dijkstra's Shortest Path First algorithm is used to find the best path
- OSPF's father: John Moy
  - Version 1: RFC 1131
  - Version 2: RFC 2328 (244 pages !!!)
  - And a lot of additional OSPF related RFCs available...

Dijkstra's SPF Algorithm

- Used in graph theory
- Very efficient
- Calculates all paths to all destinations at once
- Creates a (loop-free) tree with local router as source
- See SPF section for more details

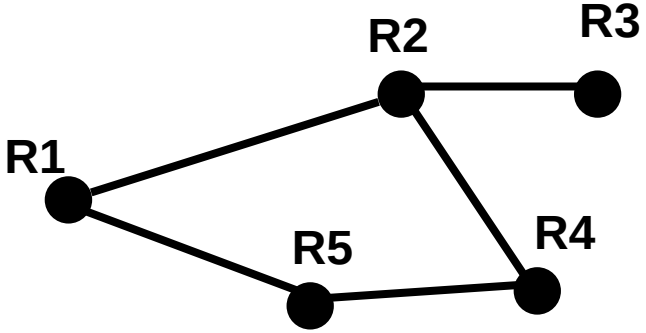
Edsger W. Dijkstra
(1930-2002)

OSPF Ideas

- Metric: "Cost" = 10^8/BW (in bit/s)
  - Therefore easily configurable per interface
- OSPF Routers exchange real topology information
  - Stored in dedicated topology databases
- Now Routers have a "roadmap"
  - Instead of signposts (RIP)
- Incremental updates
  - NO updates when there is NO topology change

What is Topology Information?

- The smallest topological unit is simply the information element ROUTER-LINK-ROUTER
- So the question is: Which router is linked to which other routers?



=

Link Database:

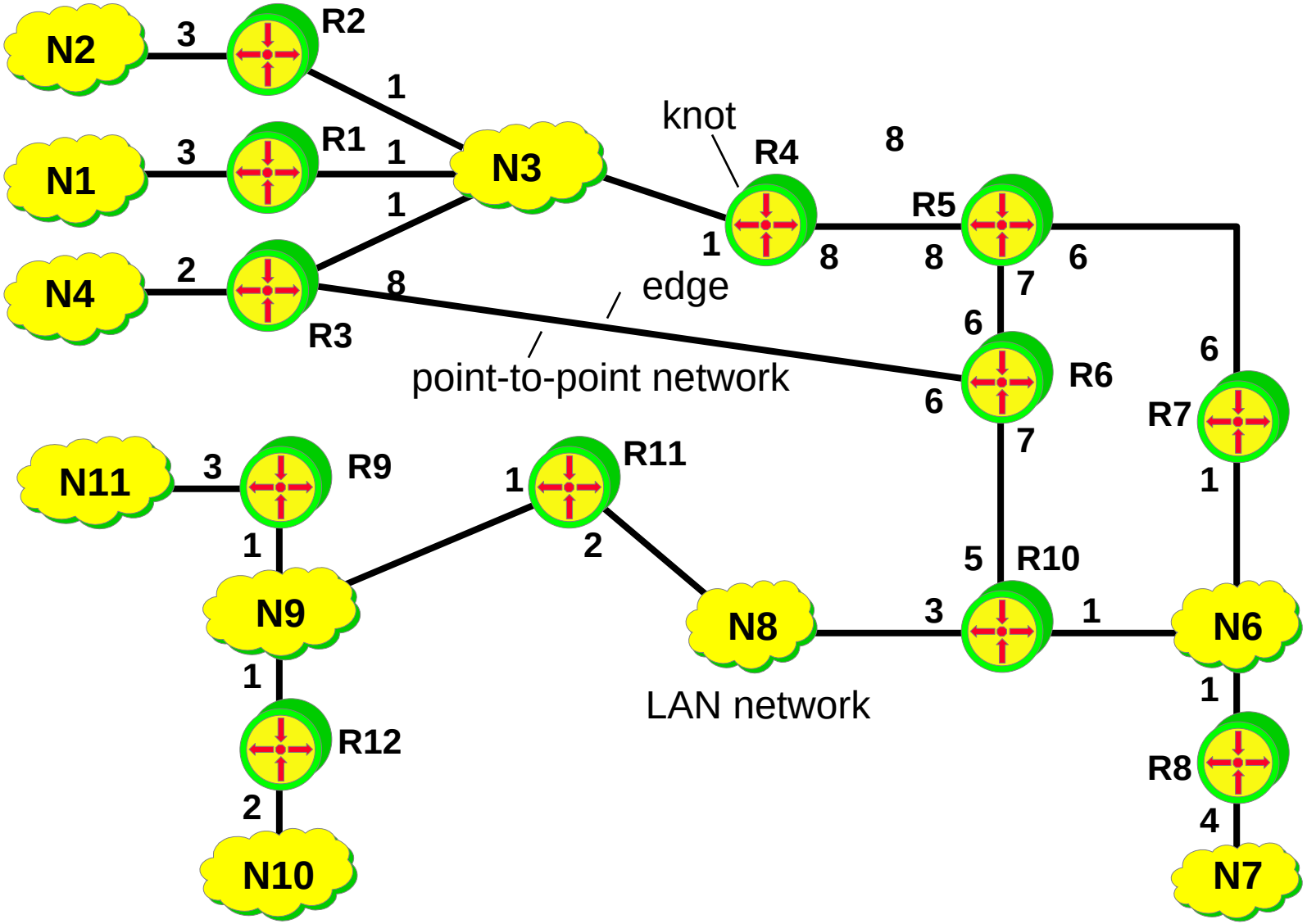
R1– R2
R1– R5
R2– R3
R2– R4
R4– R5

The Link Database exactly describes the roadmap

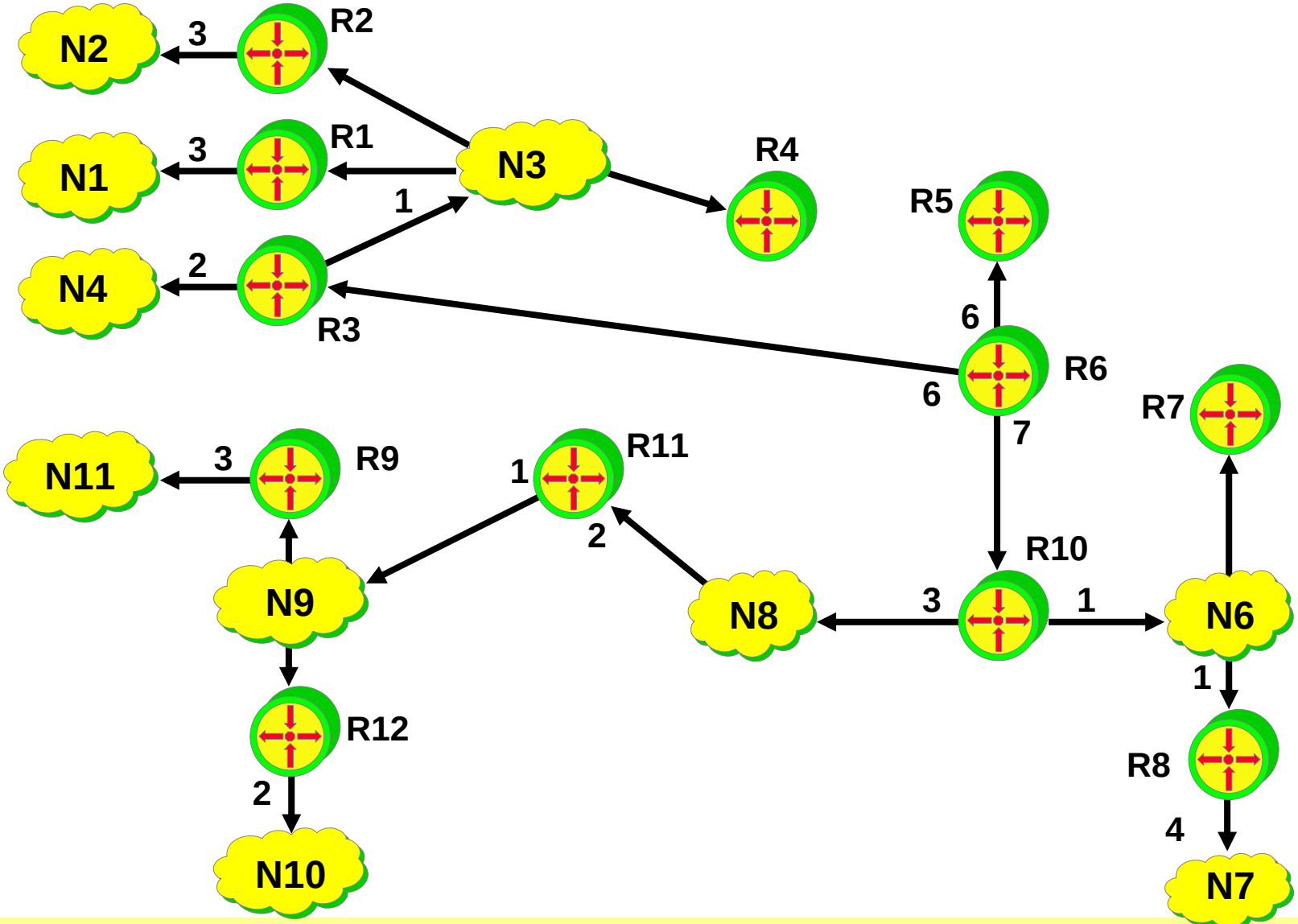
OSPF Topology Database

- **every router maintains a topology database**
  - is like a "network roadmap"
  - describes the whole network !!
  - note: RIP provides only "signposts"
- **database is based on a graph**
  - where each knot stands for a router
  - where each edge stands for a subnet
    - connecting the routers
    - path-costs are assigned to the edges
  - where the actual router uses the graph as root
    - to calculate shortest paths to all subnets

OSPF Domain


N2
3
R2
1
N1
3
R1
1
N3
knot
R4
8
1
8
R5
8
6
7
1
N4
2
8
edge
R3
6
R6
6
7
point-to-point network
6
R7
1
N11
3
R9
1
R11
1
2
5
R10
N9
N8
3
1
N6
1
LAN network
1
R12
R8
2
4
N10
N7

Shortest Paths regarding Router R6



Creating the Database

- **until now an a priori existing consistent database in every router has been assumed**
- **in fact, the basic means for creating and maintaining the database are the so-called**

link states

- **a link state stands for a local neighbourhood between two routers**
  - the link state is created by these two routers
  - other routers are notified about this link state via a broadcast-mechanism ("traffic-news")
  - link states are verified continuously

How are Link States used?

- **adjacent routers declare themselves as neighbours by setting the link state up (or down otherwise)**
  - the link-state can be checked with hello messages
- **every link state change is published to all routers of the OSPF domain using Link State Advertisements (LSAs)**
  - is a broadcast mechanism
  - LSAs are much shorter than routing tables
    - because LSAs contain only the actual changes
    - that's why distance vector protocols are much slower
  - whole topology map relies on LSAs

OSPF Routing Updates

- The routing updates are actually link state updates
  - Parts of link state database are exchanged
  - Instead of parts of routing table (RIP)
- Applying the SPF algorithm on the link state database, each router can create routing table entries by its own

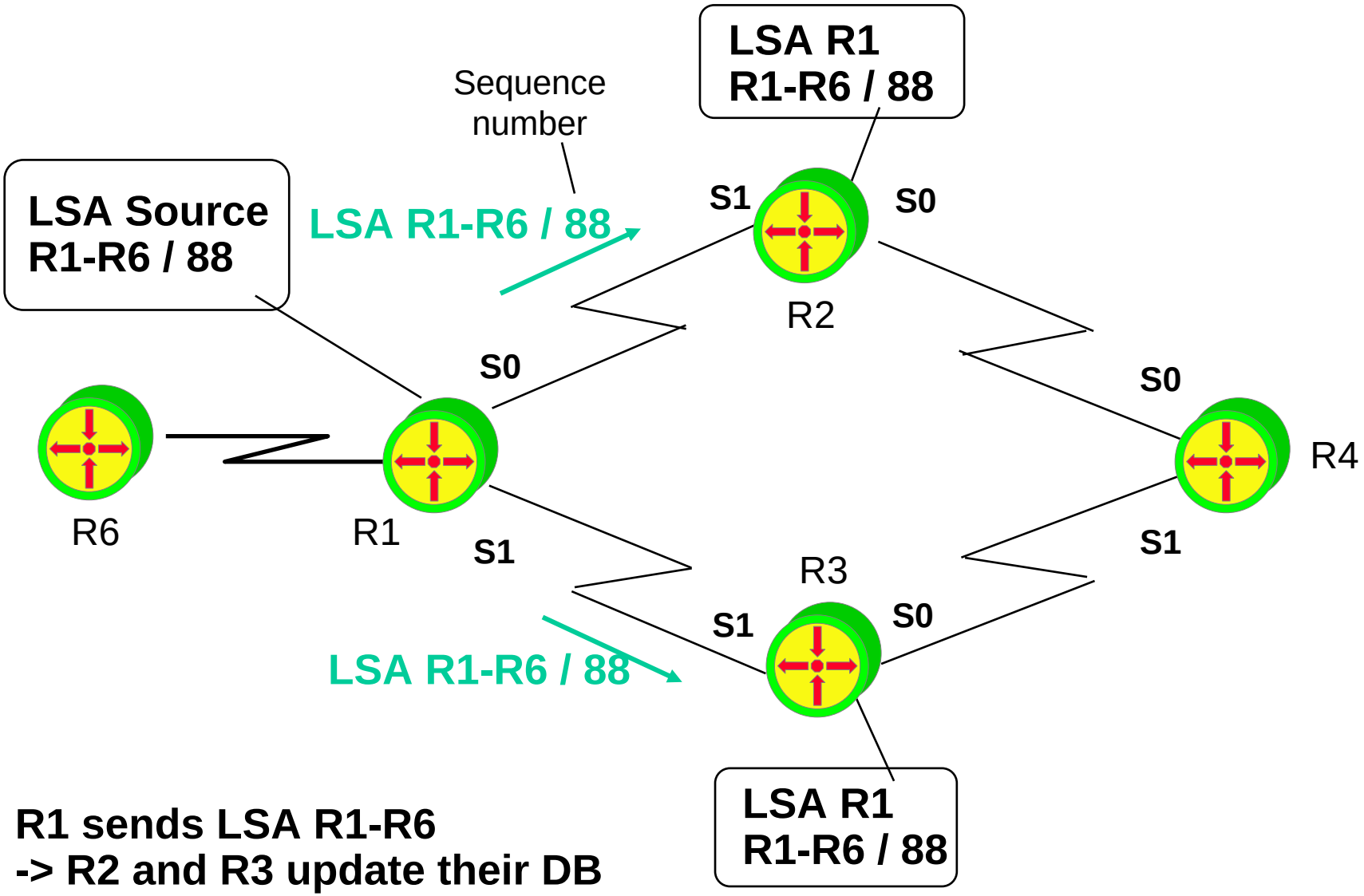
OSPF Protocol

- All OSPF messages are carried within the IP payload ("raw IP")
  - Protocol number 89
- Error recovery and session management is covered by OSPF itself
- Multicast address 224.0.0.5
  - "All OSPF routers"

LSA Broadcast Mechanism

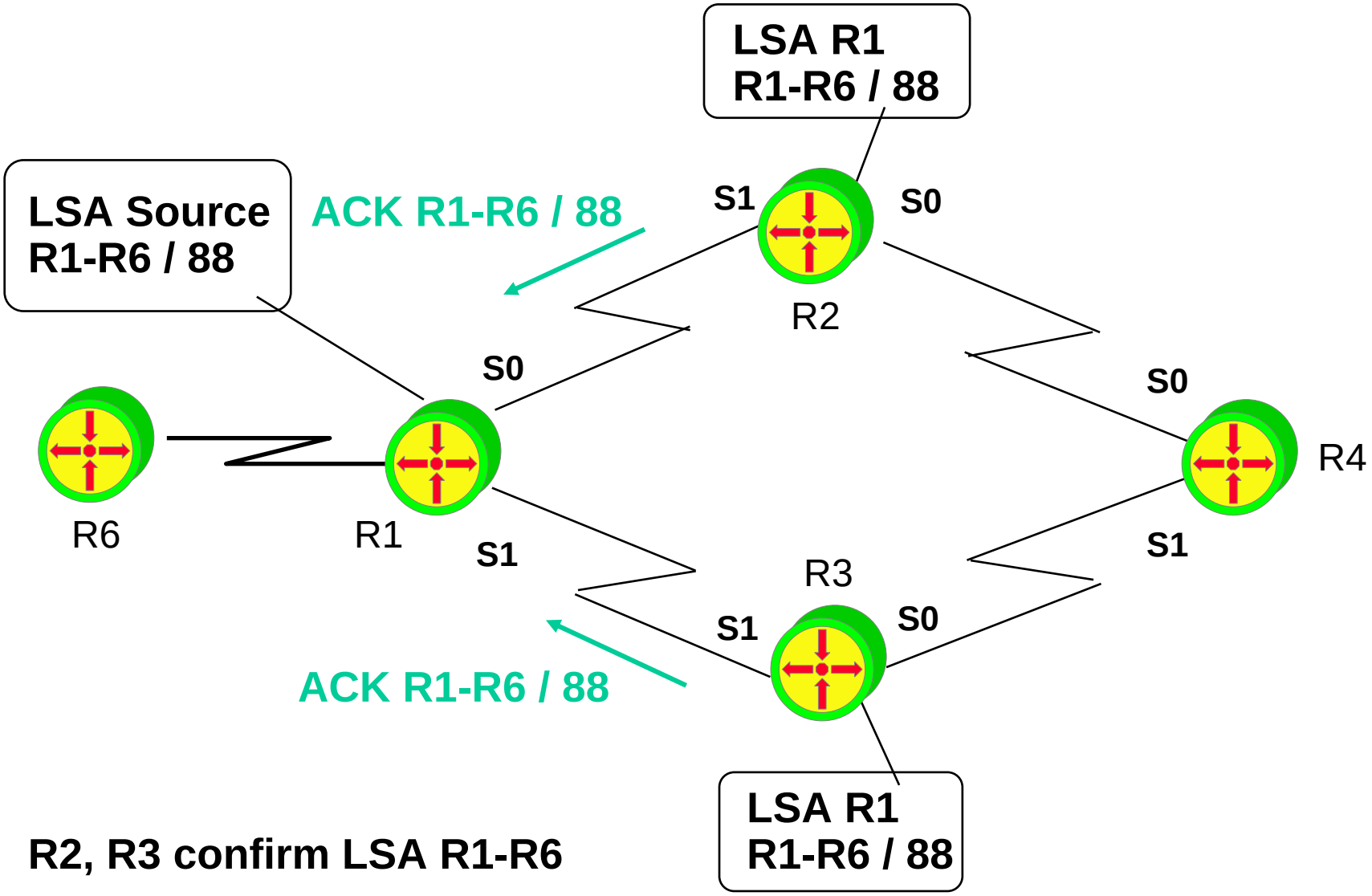
- **LSA must be distributed safely to all routers within an area (domain)**
  - consistency of the topology-database depends on it
  - every LS-update is acknowledged explicitly (using LS-ack) by the neighbour-router
  - if a LS-ack stays out, the LS-update is repeated (timeout)
  - if the LS-ack fails after several trials, the adjacency-relation is cleared
  - this method assures a reliable distribution of LSAs

LSA Broad 1



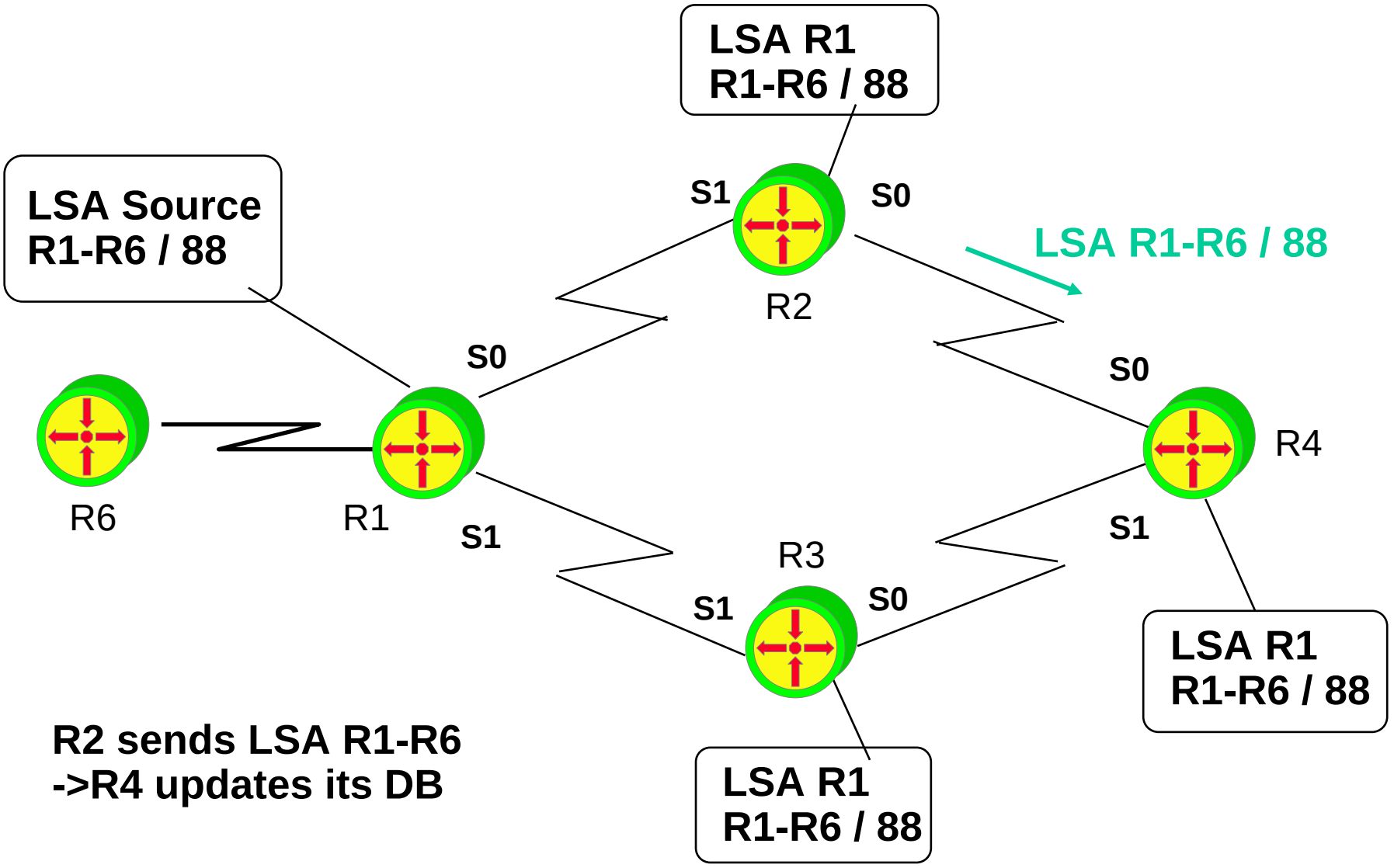
R1 sends LSA R1-R6
-> R2 and R3 update their DB

LSA Broad 2

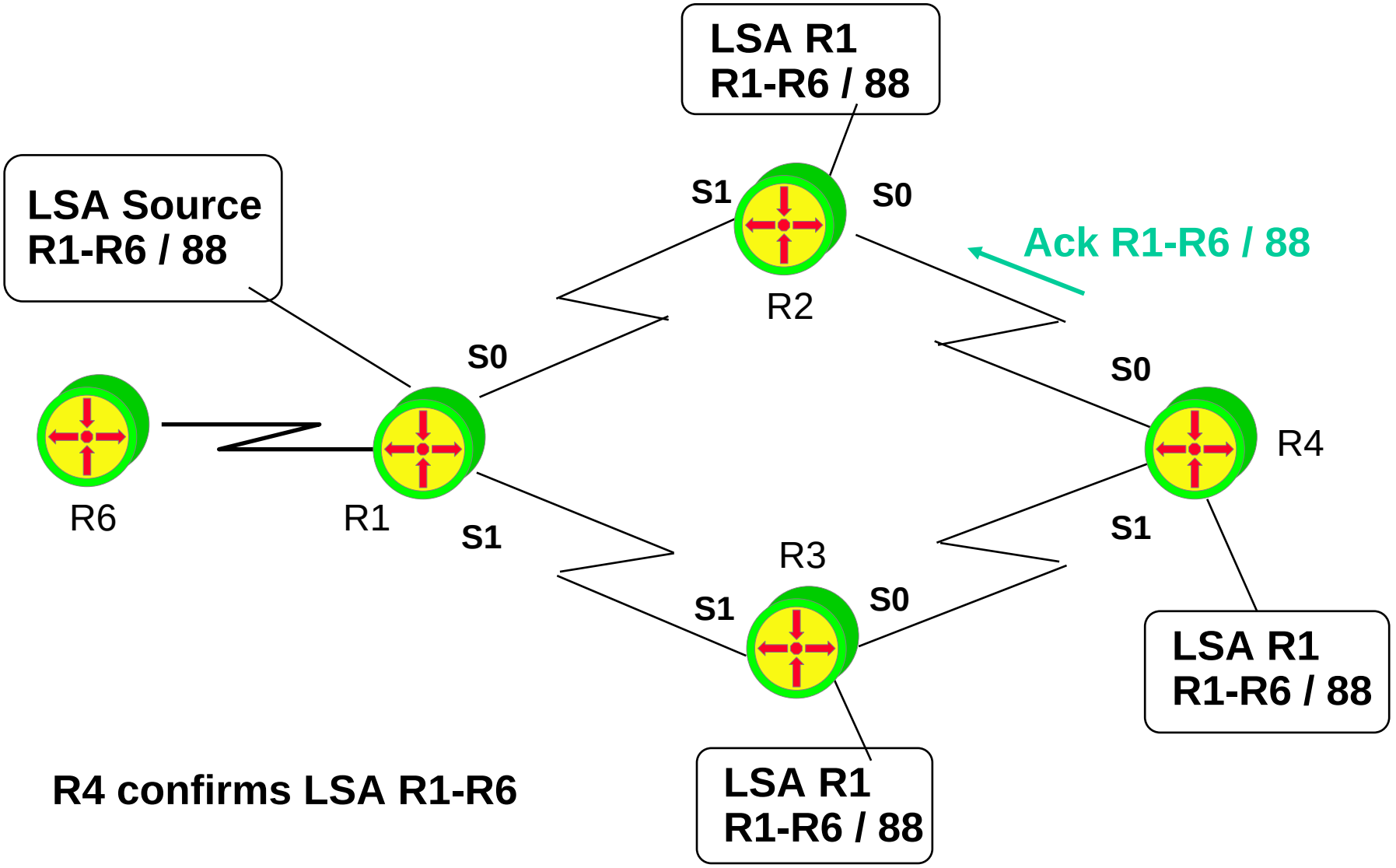


R2, R3 confirm LSA R1-R6

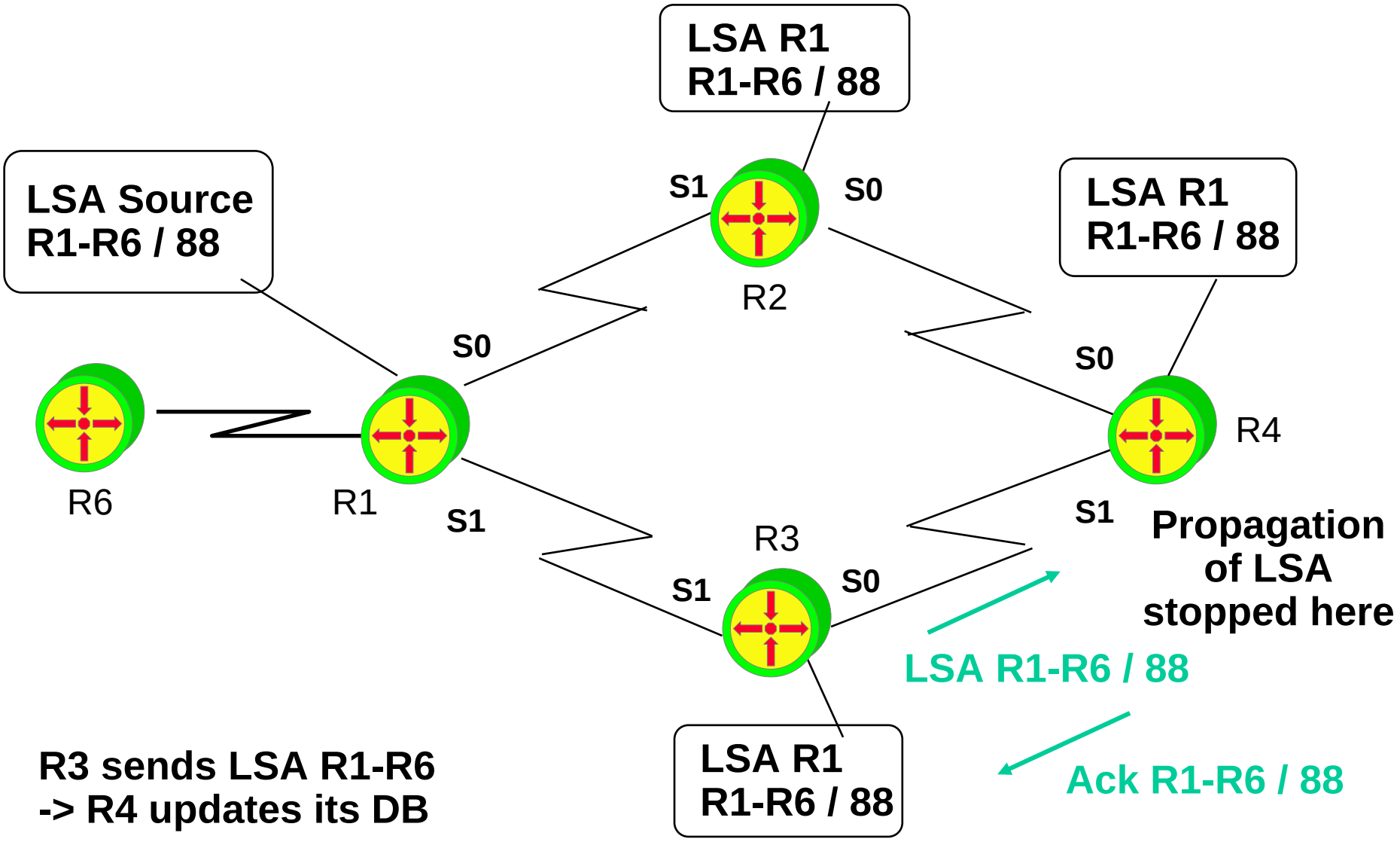
LSA Broad 3


LSA R1
R1-R6 / 88
LSA Source
R1-R6 / 88
S1
S0
LSA R1-R6 / 88
R2
S0
S0
R4
R6
R1
S1
S1
R3
S1
S0
LSA R1
R1-R6 / 88
LSA R1
R1-R6 / 88
R2 sends LSA R1-R6
->R4 updates its DB

LSA Broad 4


LSA R1
R1-R6 / 88
LSA Source
R1-R6 / 88
S1
S0
Ack R1-R6 / 88
R2
S0
S0
R4
R6
R1
S1
S1
R3
S1
S0
LSA R1
R1-R6 / 88
LSA R1
R1-R6 / 88
R4 confirms LSA R1-R6

LSA Broad 5


LSA R1
R1-R6 / 88
LSA Source
R1-R6 / 88
S1
S0
LSA R1
R1-R6 / 88
R2
S0
S0
R4
R6
R1
S1
S1
R3
S1
S0
Propagation
of LSA
stopped here
LSA R1-R6 / 88
Ack R1-R6 / 88
LSA R1
R1-R6 / 88
R3 sends LSA R1-R6
-> R4 updates its DB

LSA Flooding

- LSA's are small packets, forwarded by each router without much modifications through the whole OSPF area (!)
- Much faster than RIP updates
  - RIP must receive, examine, create, and send
- Convergence time
  - Detection time + LSA flooding + 5 seconds before computing the topology table = "a few seconds"

OSPF Overview

- **Large networks: "Divide and conquer" into areas**
  - **LSA-procedures inside each area**
  - **But *distance-vector updates between areas***
- **Additional complexity because of performance optimizations**
  - **Limit number of adjacencies in a multi-access network OSPF**
  - **Limit scope of flooding through "Areas"**
  - **Deal with stub areas efficiently**
  - **Learn external routes efficiently**
  - **Realized through different LSA types**
- **Fast convergence, almost no routing traffic in absence of topology changes**