Lab Course
“RouterLab”

OSPF - Open Shortest Path First
(RFC 2328)

Some of the slides come from: http://www.ietf.org/proceedings/07dec/slides/IDRTut-0.pdf

Miscellaneous

- Anything that needs discussion?

Internet Routing

- Distance Vector
  - I tell you all my “best” routes for all destinations that I know and you tell me yours.
  - Build simplified topology from local perspective
  - E.g. RIP
- Link State
  - I announce to everyone about my links and the addresses I originate on each link and listen to everyone’s announcement.
  - Build full topology
  - E.g. OSPF

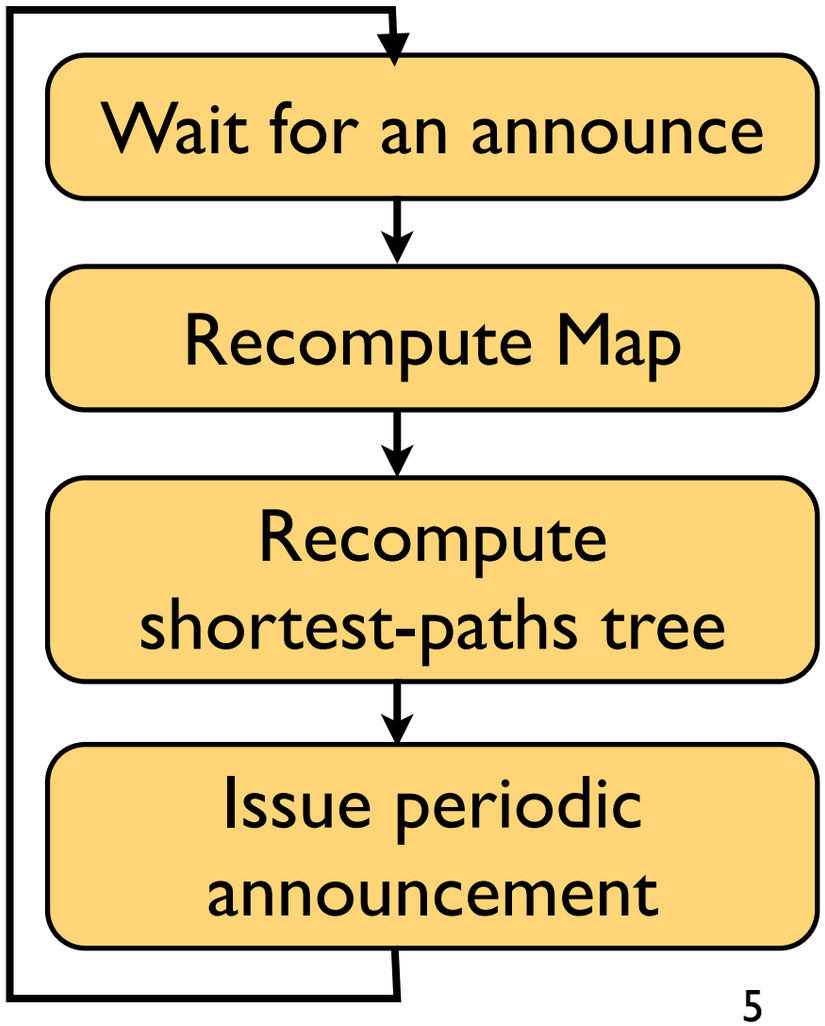
OSPF (Link State)

- I tell everyone about all my connections(links), with link up/down announcements
- I tell everyone about the addresses I originate on each link

- I listen to everyone else's link announcements
- I build a topology of every link (map)
- Then I compute the shortest path to every address
- I assume (trust) that everyone else has assembled the same map and performed the same path selection

OSPF (Link State)

- Is an instantiation of the Dijkstra Algorithm
  1. Set: $i = 0, S_0 = \{u_0 = s\}, L(u_0) = 0,$ and $L(v) = \infty,$ for $v \neq u_0$
  2. Compute: $\forall v \in (V \setminus S_i)\, L(v) = \min\{L(v), L(u_i) + d_v^{u_i}\}$
  3. Select: $u_{i+1} = v' : L(v') = \min_{\forall v \in (V \setminus S_i)}\{L(v)\}$
  4. Update: $S_{i+1} = S_i \cup u_{i+1}, i = i + 1$
  5. If $i =\mid V \mid -1$ stop, otherwise go to 2



OSPF

- Is more complex
  - (RFC 2328 is 244 pages, RIP is 39!)
- Converges extremely quickly
- Should be loop-free at all times
- Does not guarantee consistency of outcomes
- Relies on a “full disclosure” model across the whole domain
- Can be organized in several “areas”
- Still can’t scale

OSPF Components

- Hello Protocol:
  - Builds and maintains adjacencies
- Link State Announcements
- Database Synchronization:
  - Ensure consistency of the Database among neighbors
  - Reliable Flooding
- Shortest-Path Tree Computation
  - Based on the Routing Database

OSPF Packets

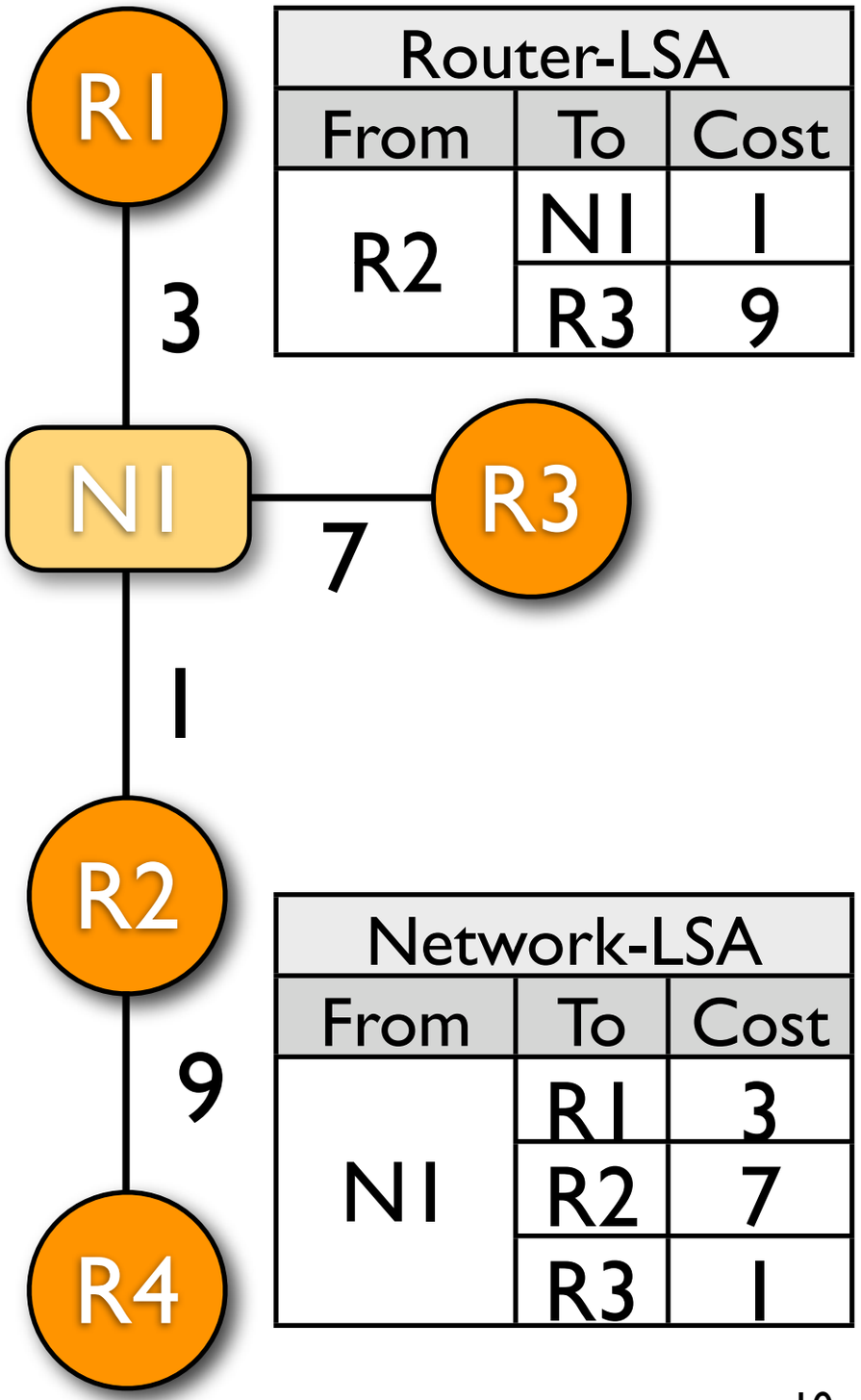
- OSPF runs directly over IP
  - Protocol Number: 89
- Packets' Type
  1. Hello - Discover/Maintain neighbors
  2. Database Description - Summarize database contents
  3. Link State Request - Database download
  4. Link State Update - Database update
  5. Link State Ack - Flooding Acknowledgement

Hello Protocol

- Task:
  - Discover/Maintain neighbor relationship
  - Discover bi-directionality
  - Negotiate capabilities (HelloInterval, Netmask, RouterDeadInterval)
- Method:
  - Periodical multicasting of Hello Packets containing:
    - List of routers whose Hello Packets have been seen recently
- Target:
  - Establish Adjacencies
  - Adjacent routers sync the link-state database

LSA - LS Announcements

- Router-LSA
  - Flooded
  - States of the links
- Network-LSA
  - Flooded by Designated Router
    - Routers use an ID (usually an IP address on the loopback)
  - Describes all routers attached to it
- The set of LSA form the Database

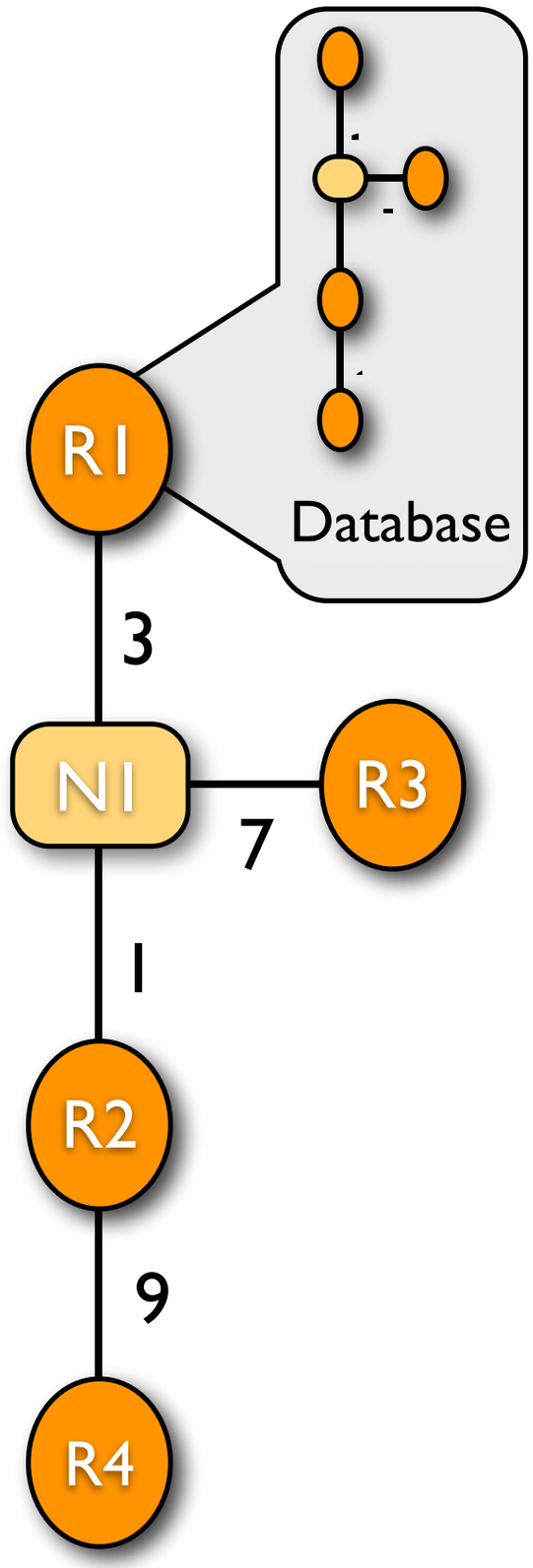


Router-LSA		
From	To	Cost
R2	N1	1
	R3	9

Network-LSA		
From	To	Cost
N1	R1	3
	R2	7
	R3	1

Database Sync

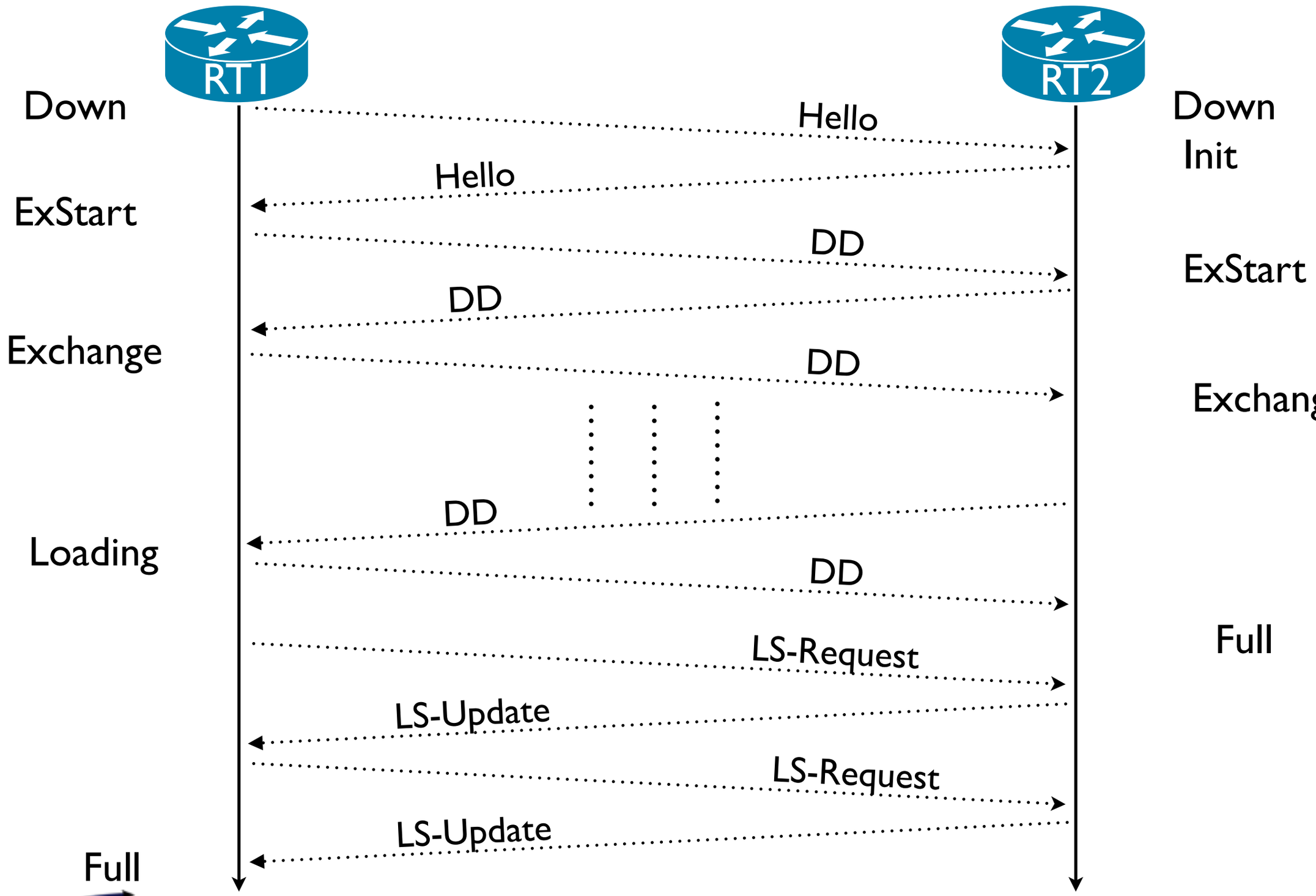
- Ideally all routers have to sync their Database
- In practice in OSPF only adjacent routers Sync their Database
- Types of Sync:
  - Initial Sync
    - when establishing adjacencies
  - Continuous Sync
    - when adjacencies are already established



Initial Database Sync

- Routers perform “Database Exchange Process”
  - State Machine
- Database Description Packets
  - Contains summaries of LS data
- DD are explicitly acknowledged
- More recent LS Data can be explicitly requested (Link State Request)
- Process ends with adjacency establishment

Sync Example

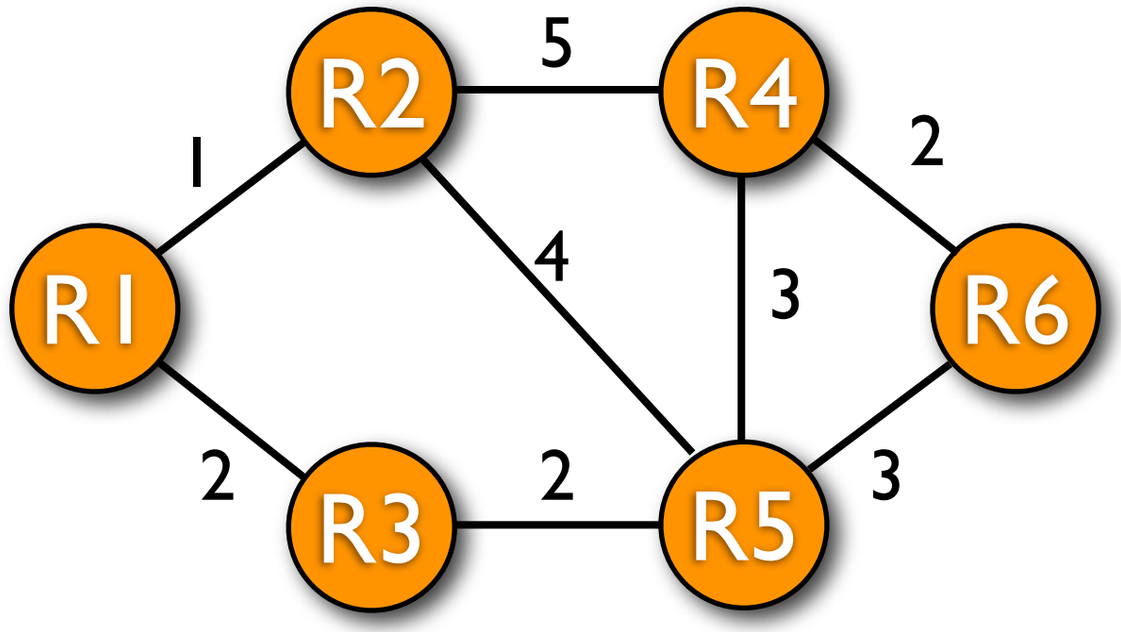


Continuous Database Sync

- Reliable Flooding
- LSA are generated
  - periodically if nothing changes (30 minutes)
  - upon specific events if they change the content of the LSA
- If a newer LSA is received by a router it is put in the database and a route computation is triggered
- The LSA is sent to all adjacencies which have to explicitly acknowledge it

Shortest-Path Tree

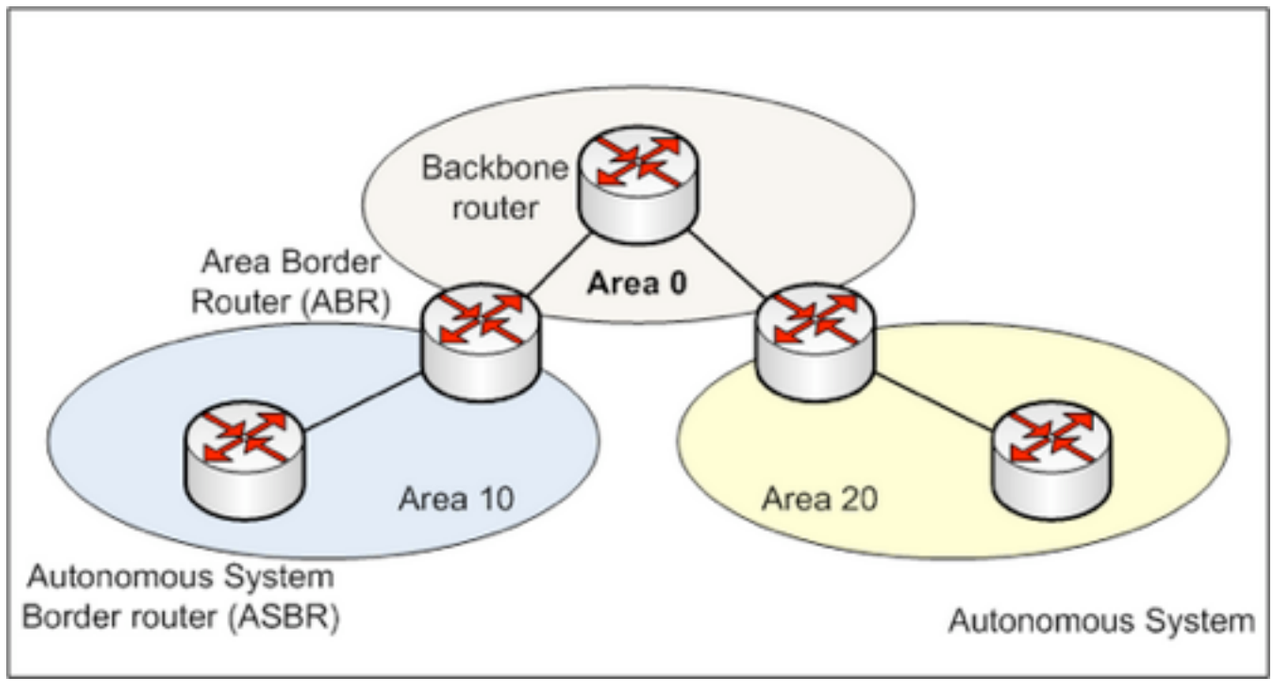
Destination	Cost	Next-Hop
R1	0	*
R2	1	link
R3	2	link
R5	4	R3
R4	6	R2
R6	7	R3



1. Added Destination = <R1,0>; Candidate Destination List = <R2,1> <R3,2>
2. Added Destination = <R2,1>; Candidate Destination List = <R3,2> **<R5,5> <R4,6>**
3. Added Destination = <R3,2>; Candidate Destination List = <R5,4> <R4,6>
4. Added Destination = <R5,4>; Candidate Destination List = <R4,6> **<R6,7>**
5. Added Destination = <R4,6>; Candidate Destination List = <R6,7>
6. Added Destination = <R6,7>; Candidate Destination List =
7. Done!

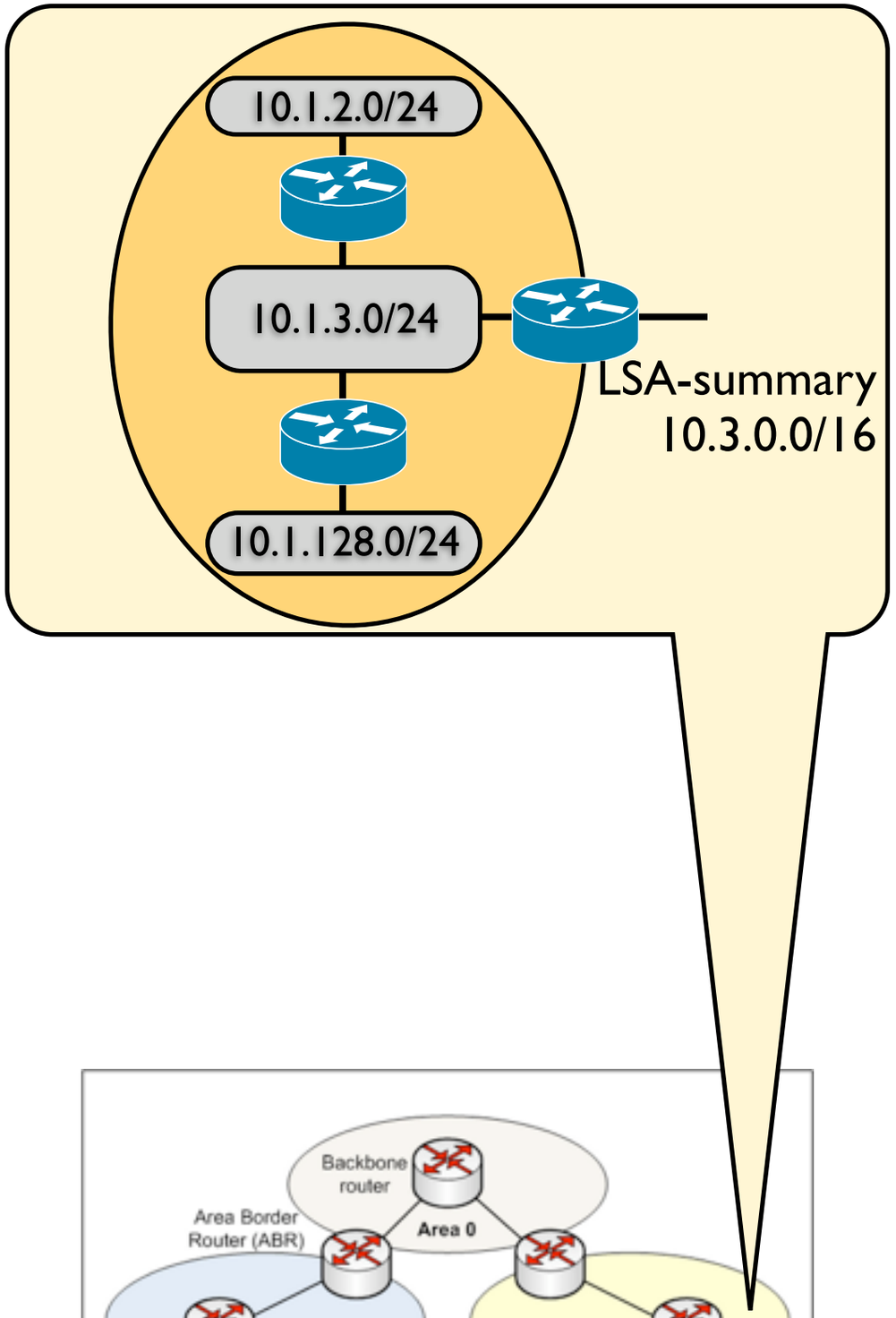
Hierarchical OSPF

- Why?
  - Reduce routing overhead
  - Increase scalability
  - Speed up convergence
  - Confine routing instabilities in contained areas



Hierarchical OSPF

- How
  - Inside Areas full OSPF
  - Flooding is limited to Areas
  - Area Border Routers summarize information
    - LSA-Summary
  - Area 0 (backbone mandatory)
  - Inter-area communication only through Area 0
  - No loops allowed among areas
  - Only 2 levels hierarchy allowed



Worksheet 4

- Use same VLANs topology like in Question 1 Work Sheet 2
- Target: logical networks communicate using OSPF
- Readings:
  - Cisco RIP
  - Juniper RIP
  - RFC 2328

New Schedule!

- Deadline Worksheet 4: 29th May 2009
- No debriefings/Tutorials in the week from 19th to 22nd May
- Debriefing Worksheet 3 (RIP) will be held on 27th May
- Tutorial on BGP will be held on Thursday 28th May

Any other Question?