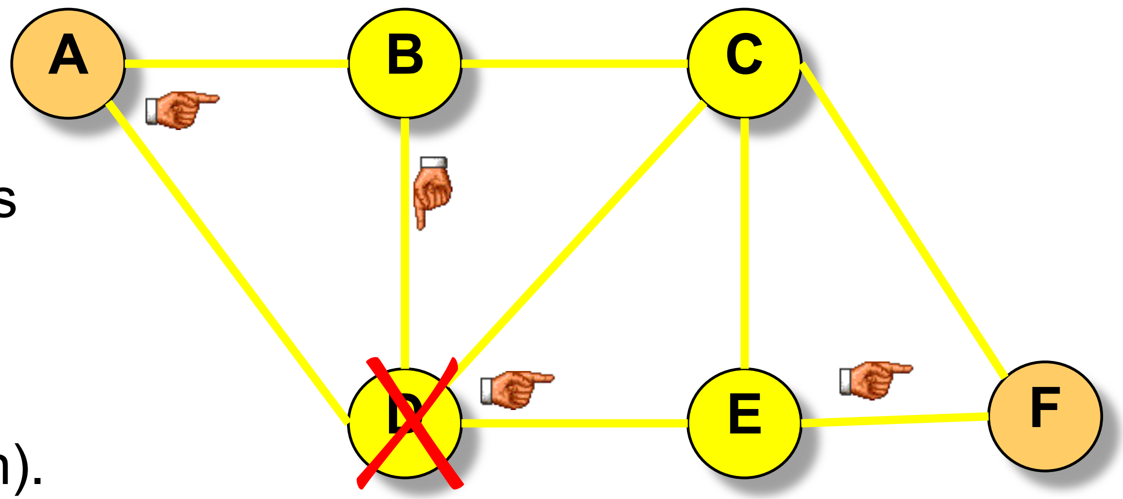
Dynamic Routing Protocols II
OSPF

This module covers link state routing and the Open Shortest Path First (OSPF) routing protocol.

Distance Vector vs. Link State Routing

- With distance vector routing, each node has information only about the next hop:
    - Node A: to reach F go to B
    - Node B: to reach F go to D
    - Node D: to reach F go to E
    - Node E: go directly to F



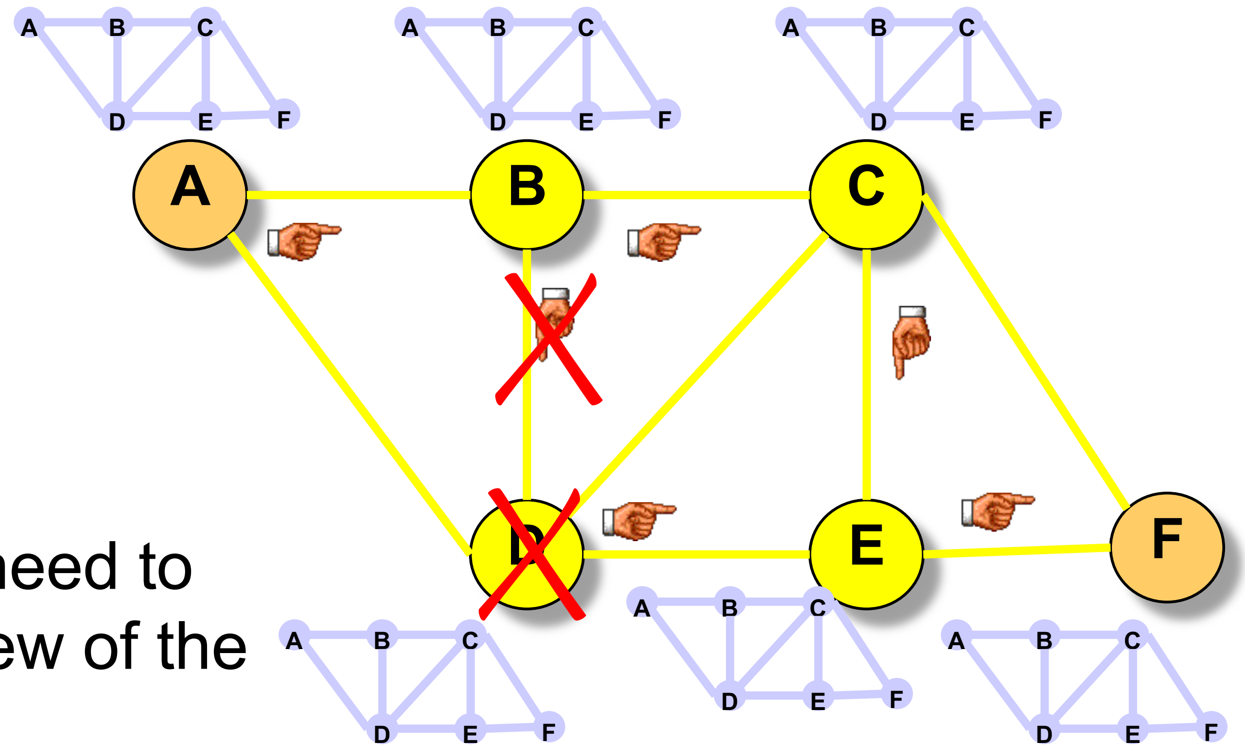
- Distance vector routing makes poor routing decisions if directions are not completely correct
(e.g., because a node is down).

- If parts of the directions incorrect, the routing may be incorrect until the routing algorithms has re-converged.

Distance Vector vs. Link State Routing

- In link state routing, each node has a complete map of the topology

- If a node fails, each node can calculate the new route



- Difficulty: All nodes need to have a consistent view of the network

Link State Routing: Properties

- Each node requires complete topology information
- Link state information must be flooded to all nodes
- Guaranteed to converge

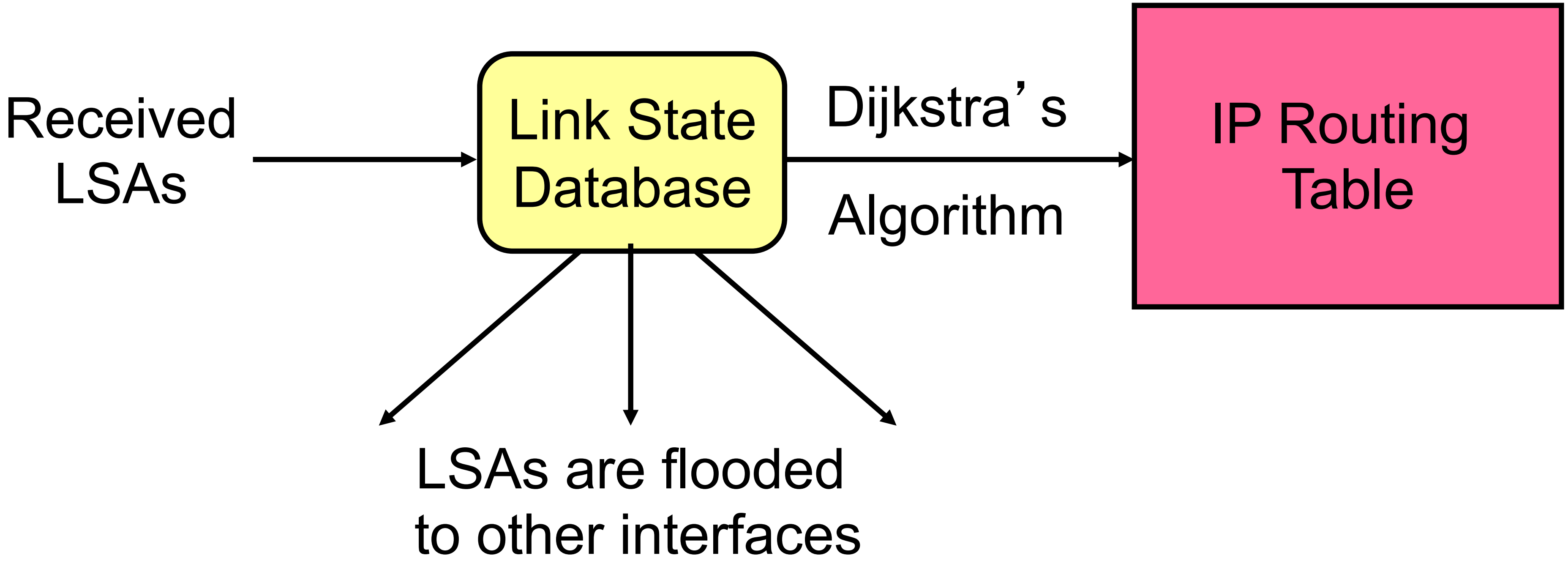
Link State Routing: Basic princples

1. Each router establishes a relationship *("adjacency")* with its neighbors
2. Each router generates *link state advertisements (LSAs)* which are distributed to all routers

   LSA = (link id, state of the link, cost, neighbors of the link)

3. Each router maintains a database of all received LSAs (*topological database* or *link state database*), which describes the network as a graph with weighted edges
4. Each router uses its link state database to run a shortest path algorithm (Dijikstra's algorithm) to produce the shortest path to each network

Operation of a Link State Routing protocol



Dijkstra's Shortest Path Algorithm for a Graph

Input: Graph (N, E) with

N the set of nodes and E the set of edges

d_{vw} link cost $(d_{vw}$ = infinity if $(v,w) \notin E$, $d_{vv} = 0)$

s source node.

Output: D_n cost of the least-cost path from node s to node n

```
M = {s};
for each  n ∉ M
        D_n  = d_sn;
while (M ≠ all nodes) do
        Find w ∉ M for which D_w = min{D_j ; j ∉ M};
        Add w to M;
        for each  n ∉ M
                D_n = min_w [ D_n, D_w + d_wn ];
                Update route;
enddo
```

OSPF

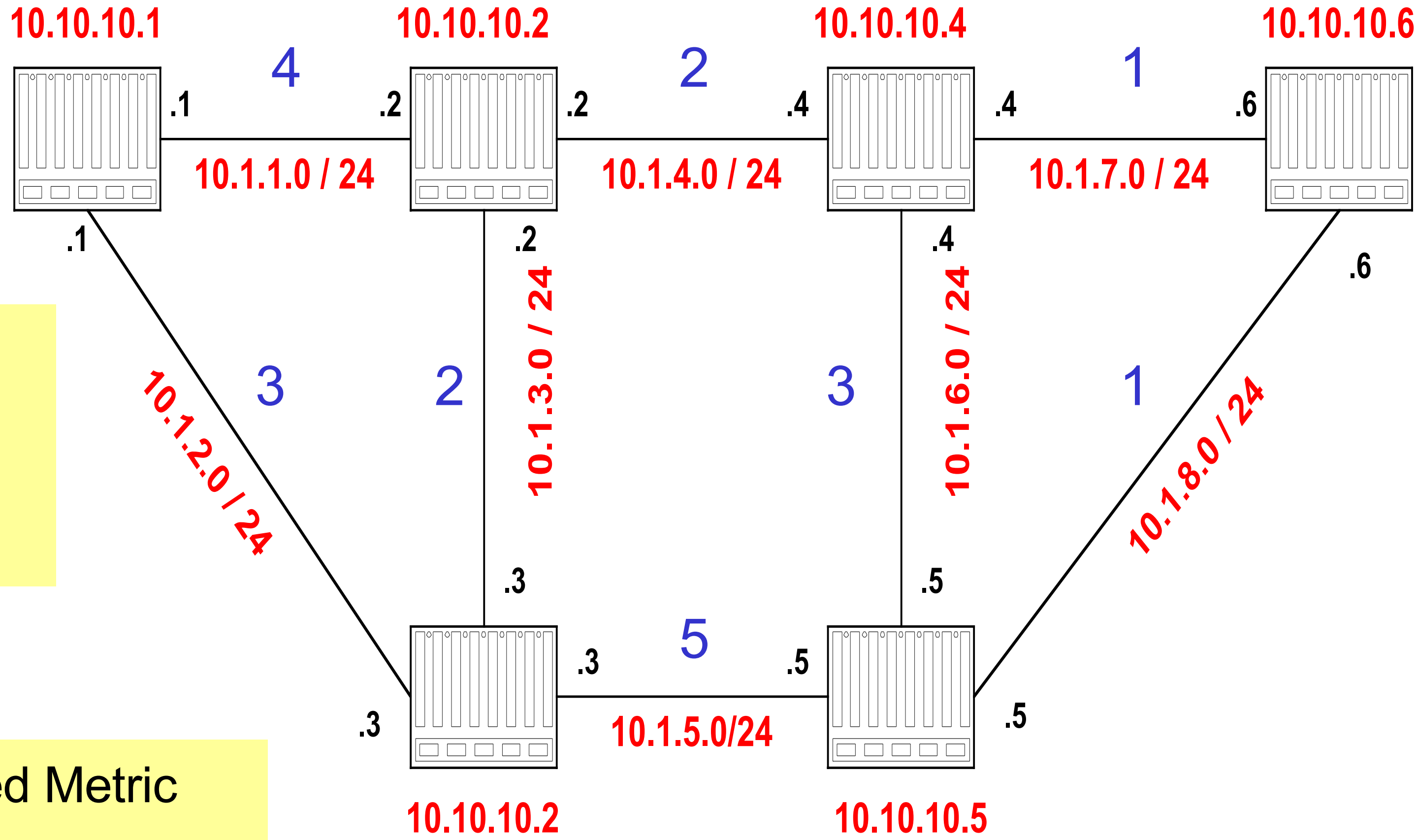
- OSPF = Open Shortest Path First
- The OSPF routing protocol is the most important link state routing protocol on the Internet
- The complexity of OSPF is significant

- History:
  - 1989: RFC 1131 OSPF Version 1
  - 1991: RFC1247 OSPF Version 2
  - 1994: RFC 1583 OSPF Version 2 (revised)
  - 1997: RFC 2178 OSPF Version 2 (revised)
  - 1998: RFC 2328 OSPF Version 2 (current version)

Features of OSPF

- Provides authentication of routing messages
- Enables load balancing by allowing traffic to be split evenly across routes with equal cost
- Type-of-Service routing allows to setup different routes dependent on the TOS field
- Supports subnetting
- Supports multicasting
- Allows hierarchical routing

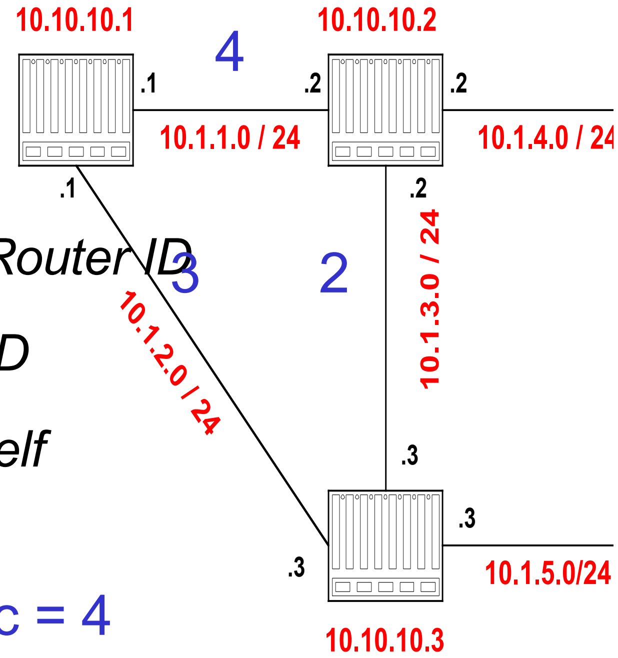
Example Network



Router IDs are selected independent of interface addresses

- Link costs are called Metric
- Metric is in the range [0 , 2^{16}]
- Metric can be asymmetric

Link State Advertisement (LSA)

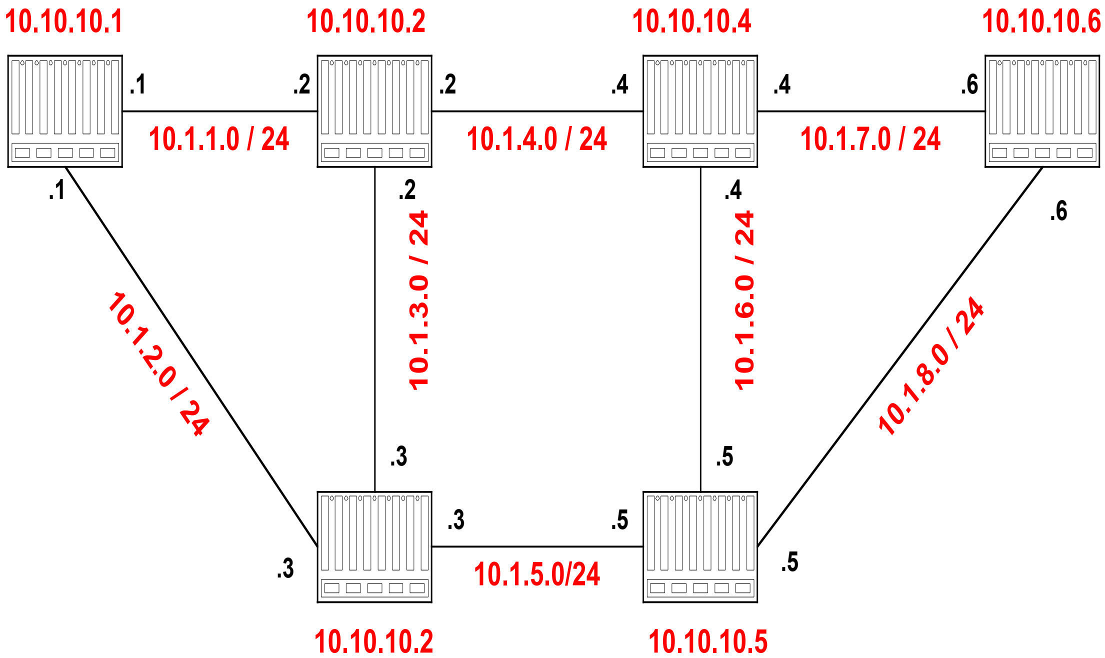


- **The LSA of router 10.10.10.1 is as follows:**
- **Link State ID**: 10.10.10.1 = *can be Router ID*
- **Advertising Router:** 10.10.10.1 = *Router ID*
- **Number of links:** 3 = *2 links plus router itself*
- **Description of Link 1:** Link ID = 10.1.1.1, Metric = 4
- **Description of Link 2:** Link ID = 10.1.2.1, Metric = 3
- **Description of Link 3:** Link ID = 10.10.10.1, Metric = 0

Each router sends its LSA to all routers in the network (using a method called reliable flooding)

Network and Link State Database

Each router has a database which contains the LSAs from all other routers

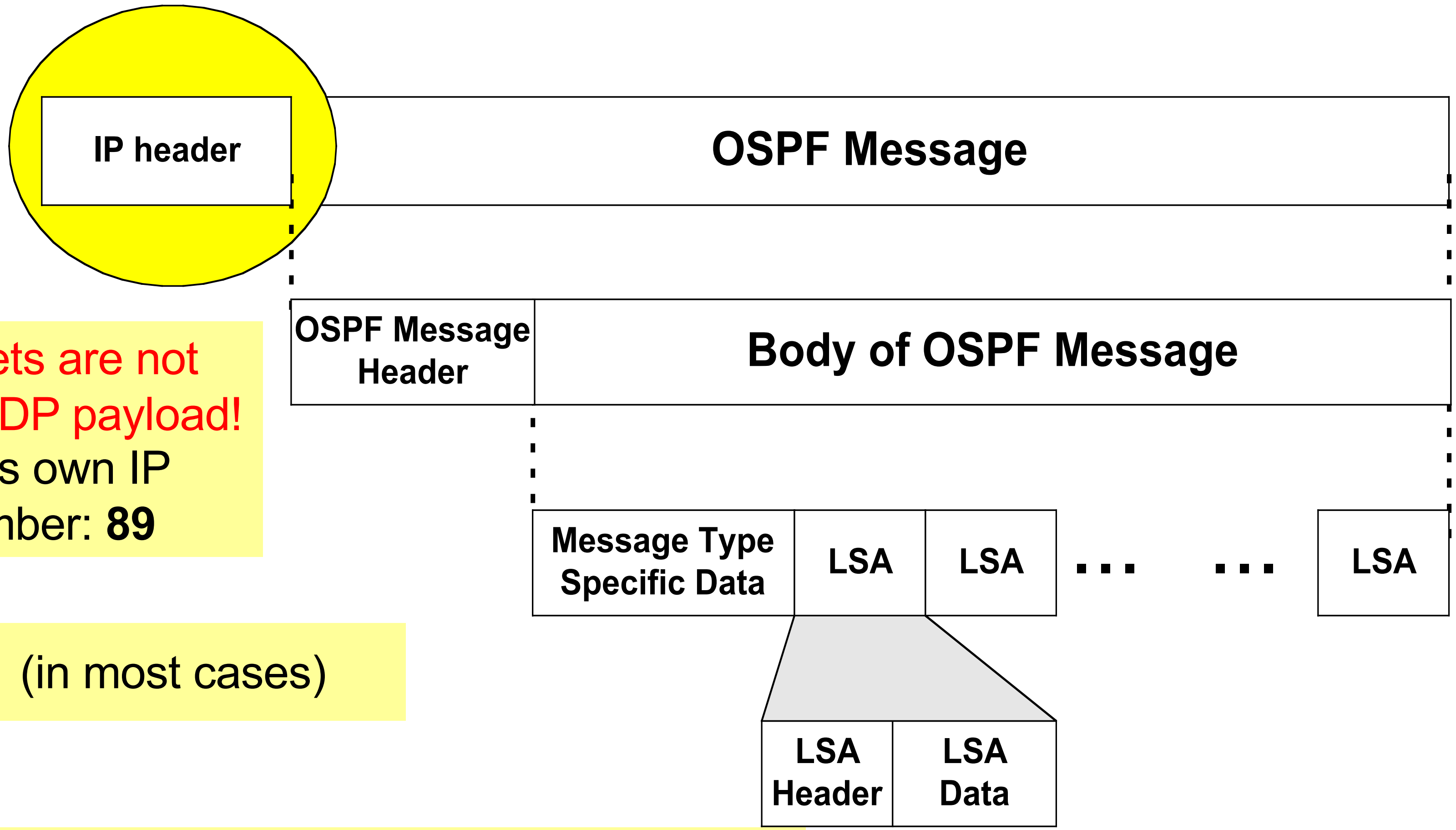


LS Type	Link StateID	Adv. Router	Checksum	LS SeqNo	LS Age
Router-LSA	10.1.10.1	10.1.10.1	0x9b47	0x80000006	0
Router-LSA	10.1.10.2	10.1.10.2	0x219e	0x80000007	1618
Router-LSA	10.1.10.3	10.1.10.3	0x6b53	0x80000003	1712
Router-LSA	10.1.10.4	10.1.10.4	0xe39a	0x8000003a	20
Router-LSA	10.1.10.5	10.1.10.5	0xd2a6	0x80000038	18
Router-LSA	10.1.10.6	10.1.10.6	0x05c3	0x80000005	1680

Link State Database

- The collection of all LSAs is called the link-state database
- Each router has and identical link-state database
  - Useful for debugging: Each router has a complete description of the network
- If neighboring routers discover each other for the first time, they will exchange their link-state databases
- The link-state databases are synchronized using reliable flooding

OSPF Packet Format

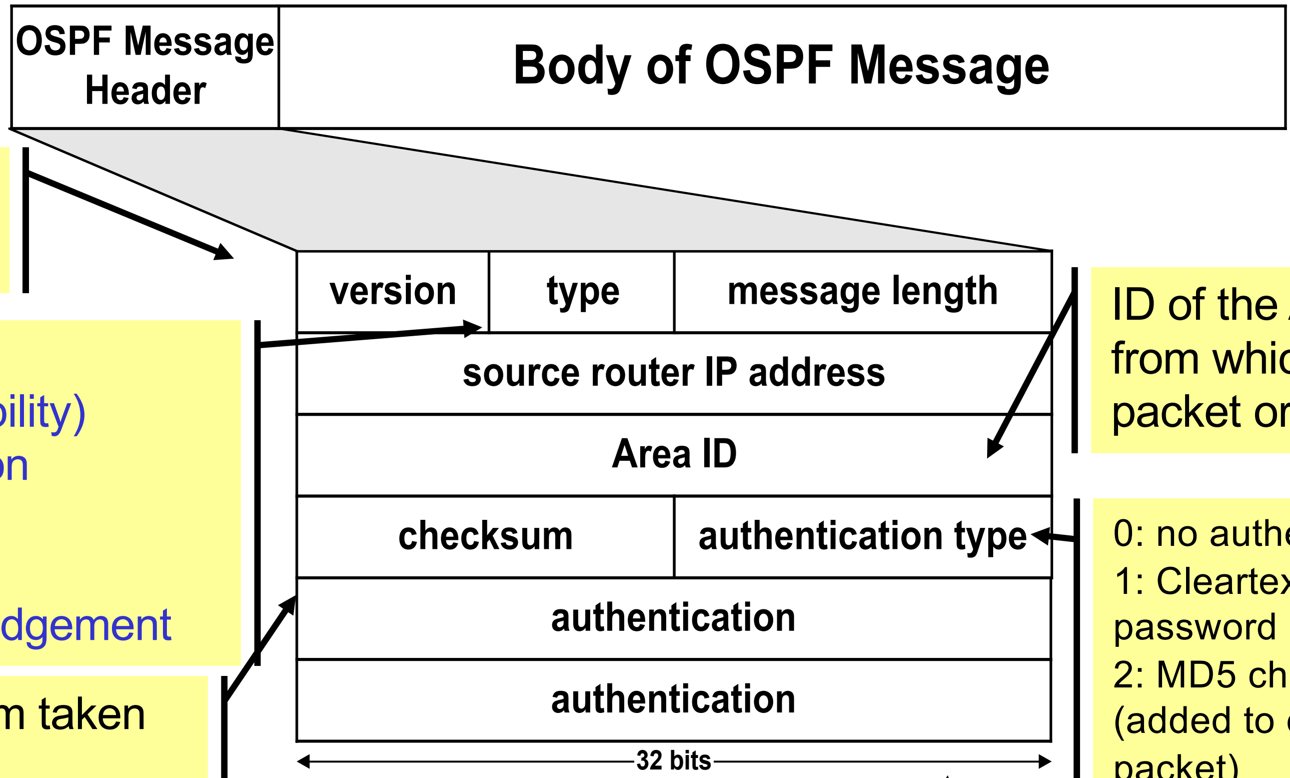


OSPF packets are not carried as UDP payload! OSPF has its own IP protocol number: **89**

TTL: set to 1 (in most cases)

Destination IP: neighbor's IP address or 224.0.0.5 (ALLSPFRouters) or 224.0.0.6 (AllDRouters)

OSPF Packet Format



2: current version is OSPF V2

Message types:
1: Hello (tests reachability)
2: Database description
3: Link Status request
4: Link state update
5: Link state acknowledgement

Standard IP checksum taken over entire packet

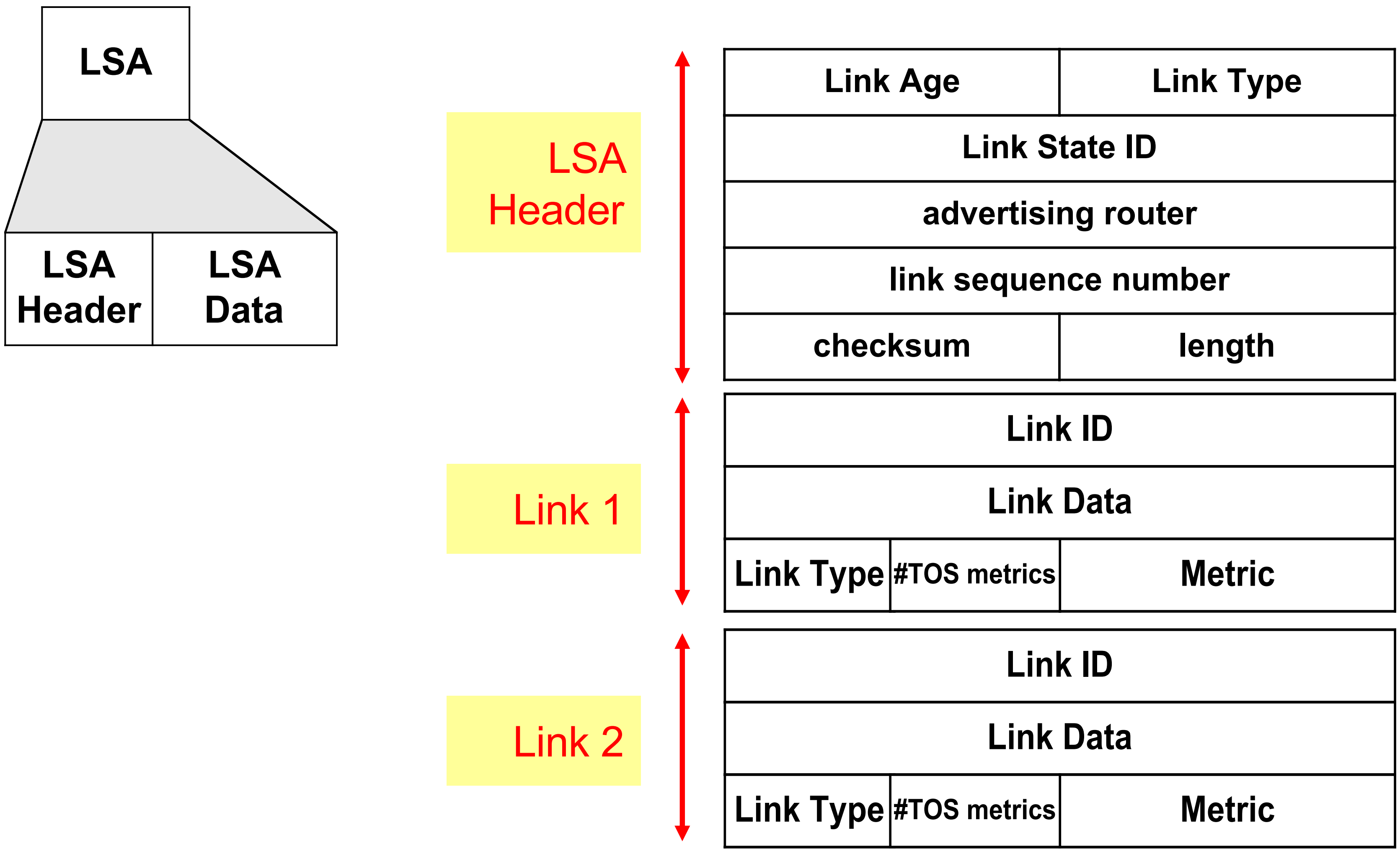
ID of the Area from which the packet originated

0: no authentication
1: Cleartext password
2: MD5 checksum (added to end packet)

Authentication passwd = 1: 64 cleartext password
Authentication passwd = 2: 0x0000 (16 bits)
KeyID (8 bits)
Length of MD5 checksum (8 bits)
Nondecreasing sequence number (32 bits)

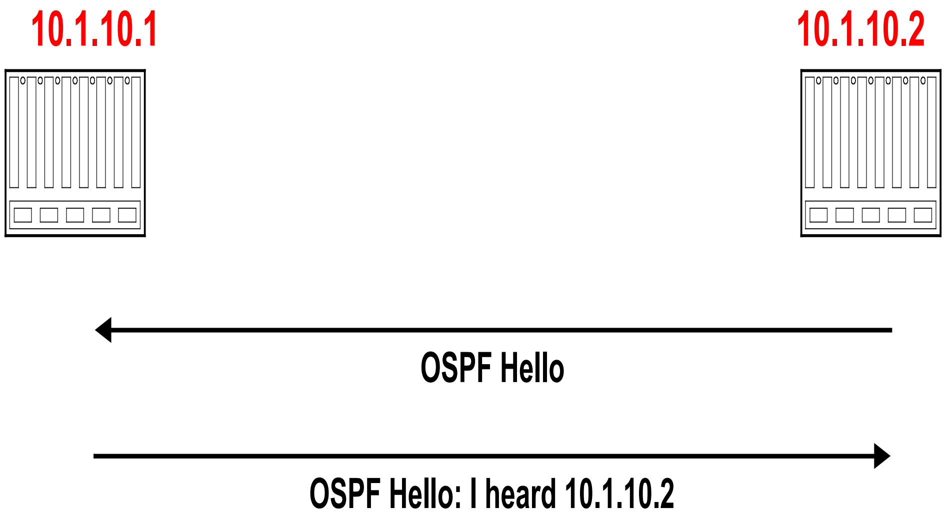
Prevents replay attacks

OSPF LSA Format



Discovery of Neighbors

- Routers multicasts OSPF Hello packets on all OSPF-enabled interfaces.
- If two routers share a link, they can become neighbors, and establish an adjacency



Scenario:
Router 10.1.10.2 restarts

- After becoming a neighbor, routers exchange their link state databases

Neighbor discovery and database synchronization

Scenario:
Router 10.1.10.2 restarts

10.1.10.1 | 10.1.10.2

Discovery of adjacency

- ← OSPF Hello
- → OSPF Hello: I heard 10.1.10.2

After neighbors are discovered the nodes exchange their databases

← Database Description: Sequence = X

Sends empty database description

→ Database Description: Sequence = X, 5 LSA headers =
Router-LSA, 10.1.10.1, 0x80000006
Router-LSA, 10.1.10.2, 0x80000007
Router-LSA, 10.1.10.3, 0x80000003
Router-LSA, 10.1.10.4, 0x8000003a
Router-LSA, 10.1.10.5, 0x80000038
Router-LSA, 10.1.10.6, 0x80000005

Sends database description. (description only contains LSA headers)

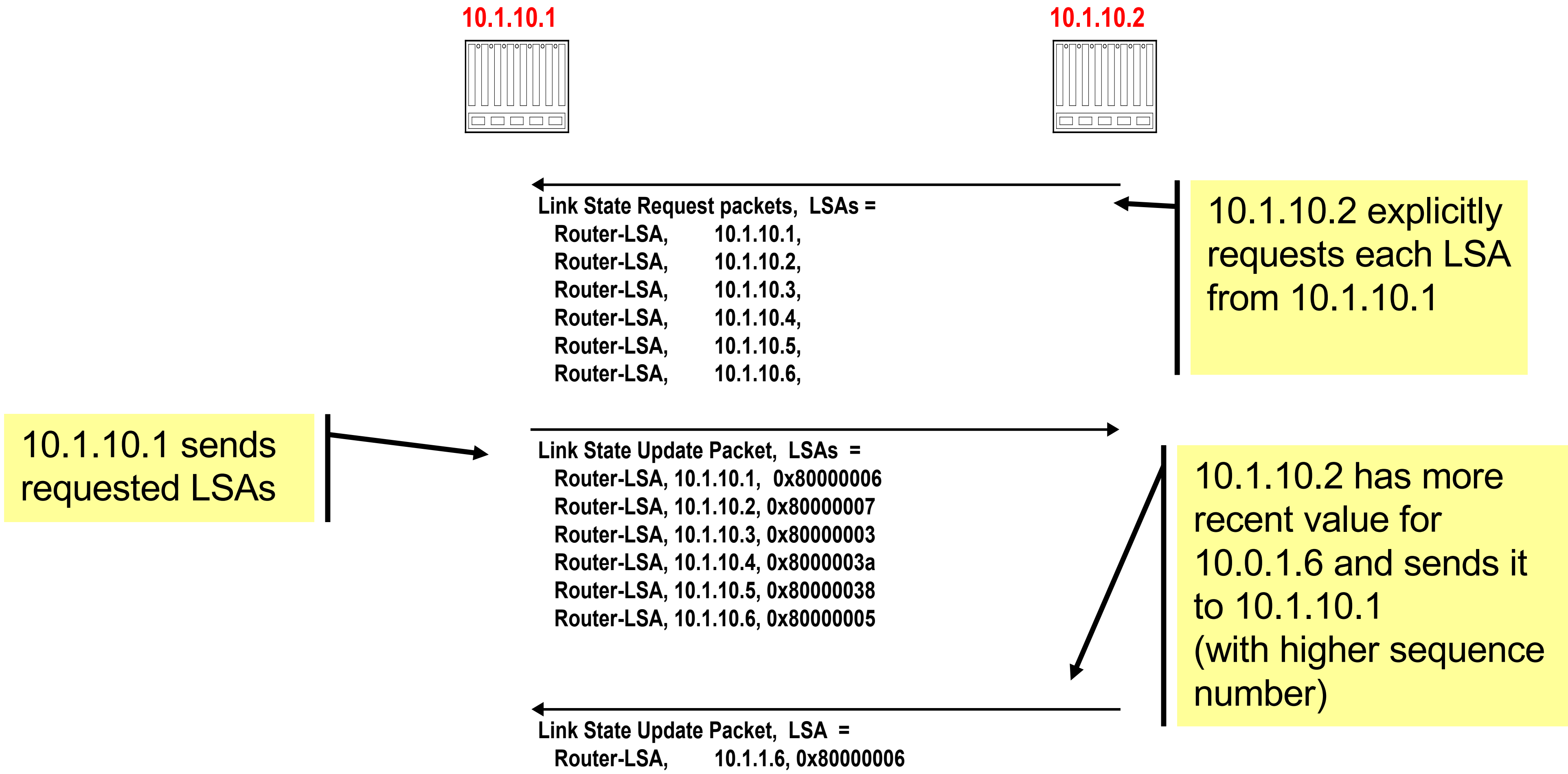
← Database Description: Sequence = X+1, 1 LSA header=
Router-LSA, 10.1.10.2, 0x80000005

Database description of 10.1.10.2

→ Database Description: Sequence = X+1

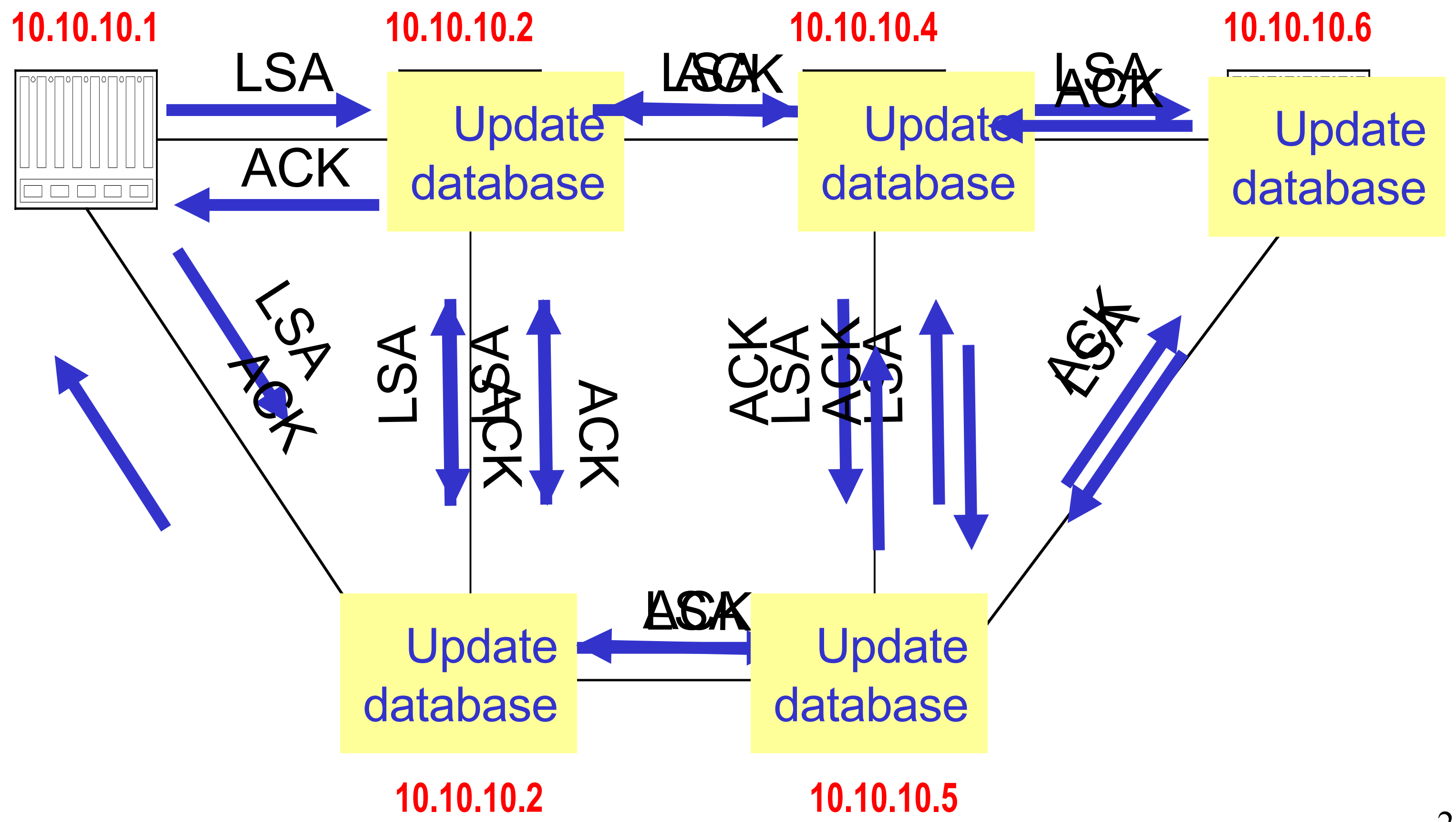
Acknowledges receipt of description

Regular LSA exchanges



Routing Data Distribution

- LSA-Updates are distributed to all other routers via **Reliable Flooding**
- **Example:** Flooding of LSA from 10.10.10.1



Dissemination of LSA-Update

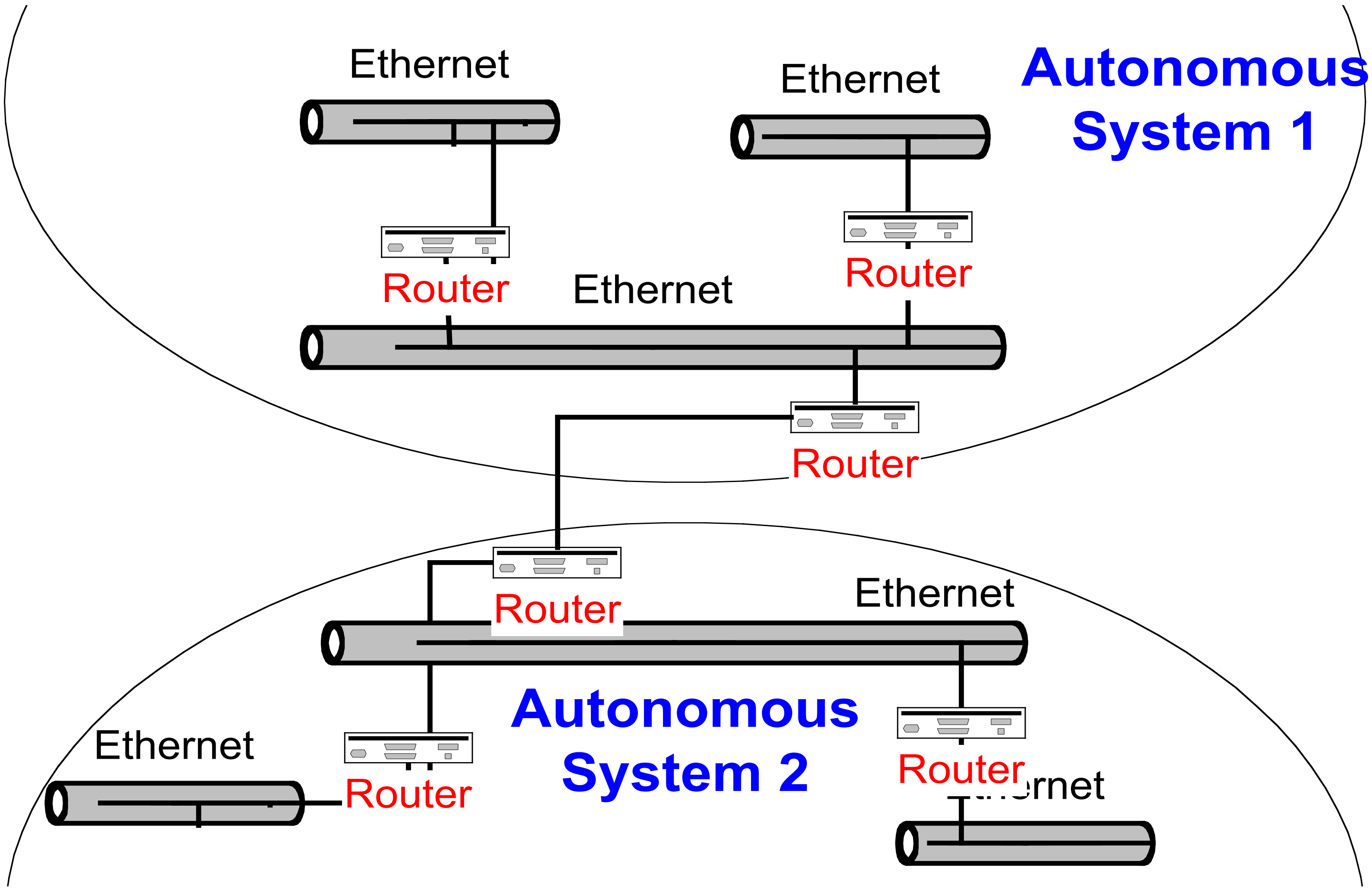
- A router sends and refloods LSA-Updates, whenever the topology or link cost changes. (If a received LSA does not contain new information, the router will not flood the packet)
- Exception: Infrequently (every 30 minutes), a router will flood LSAs even if there are no new changes.
- Acknowledgements of LSA-updates:
    - explicit ACK, or
    - implicit via reception of an LSA-Update

Autonomous Systems

- An **autonomous system** is a region of the Internet that is administered by a single entity.
- Examples of autonomous regions are:
  - UVA’s campus network
  - MCI’s backbone network
  - Regional Internet Service Provider

- Routing is done differently within an autonomous system (**intradomain routing**) and between autonomous system **(interdomain routing**).

Autonomous Systems (AS)



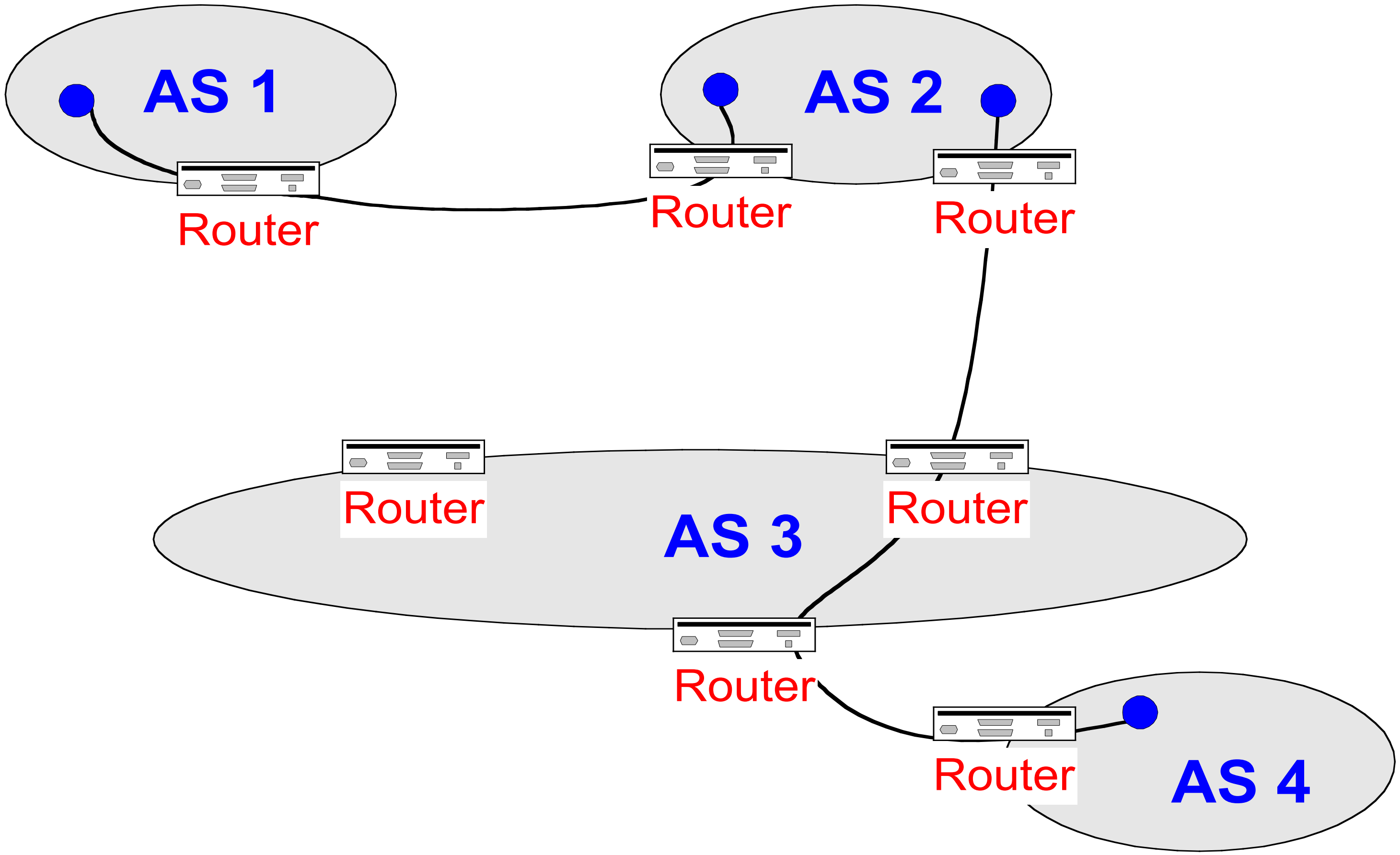
BGP

- BGP = Border Gateway Protocol
- Currently in version 4
- Note: In the context of BGP, a gateway is nothing else but an IP router that connects autonomous systems.
- Interdomain routing protocol for routing between autonomous systems
- Uses TCP to send routing messages
- BGP is neither a link state, nor a distance vector protocol. Routing messages in BGP contain complete routes.
- Network administrators can specify routing policies

BGP

- BGP's goal is to find any path (not an optimal one). Since the internals of the AS are never revealed, finding an optimal path is not feasible.
- For each autonomous system (AS), BGP distinguishes:
  - **local traffic** = traffic with source or destination in AS
  - **transit traffic** = traffic that passes through the AS
  - **Stub AS** = has connection to only one AS, only carry local traffic
  - **Multihomed AS** = has connection to >1 AS, but does not carry transit traffic
  - **Transit AS** = has connection to >1 AS and carries transit traffic

BGP


AS 1
Router
AS 2
Router
Router
Router
AS 3
Router
Router
Router
AS 4

BGP - Example

AS v4 BGP propagation graph example

1. Find Latvian ASNs Report https://bgp.he.net/report/world
2. Find AS Citadele Banka AS16279 https://bgp.he.net/AS16279
3. Look BGP Peers on Graph v4 (“Path to Internet”):
   - AS16279->AS12578->AS6939 (Hurricane Electric)
   - AS16279->AS13194->AS174 (Cogent Communication)
4. Read AS Info about all Ass
   - Company Name & Origin Country
   - Company Website & Network Map
   - Internet Exchanges Nrs
   - Prefixes Originated Nrs
   - Prefixes Announced Nrs
   - AS Paths Observed Nrs
5. Find CAIDA AS Rank on site
   https://asrank.caida.org/