ECE/COMPSCI 356 Computer Network Architecture

Lecture 13: OSPF

Neil Gong

neil.gong@duke.edu

Slides credit: Xiaowei Yang, PD

Today

- Link-state routing
  - Algorithm
  - Protocol: Open shortest path first (OSPF)

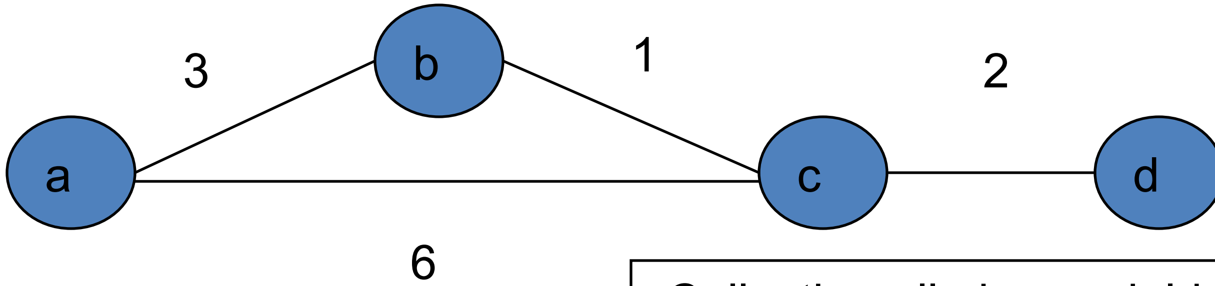
Link State Routing: Basic operations

1. Each router establishes *link adjacency*
    1. Neighbors and link costs to them
2. Each router generates a *link state advertisement (LSA), and* floods it to the network

3. Each router maintains a database of all received LSAs (*topological database* or *link state database*)

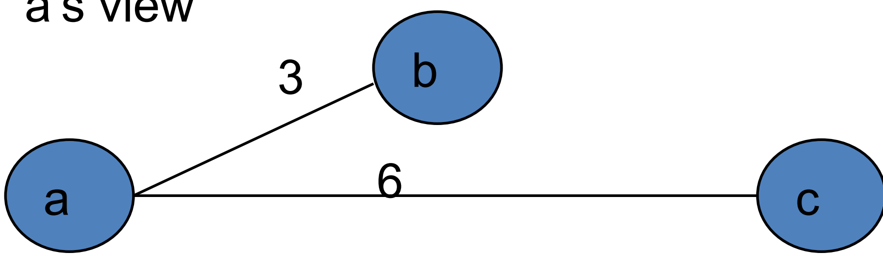
4. Each router runs the Dijkstra's algorithm

Link state routing: graphical illustration

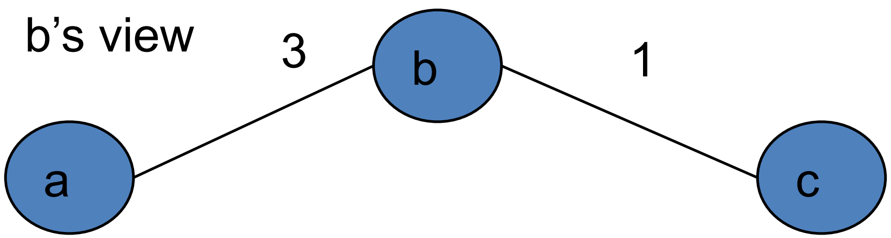


Collecting all pieces yield
a complete view of the network!

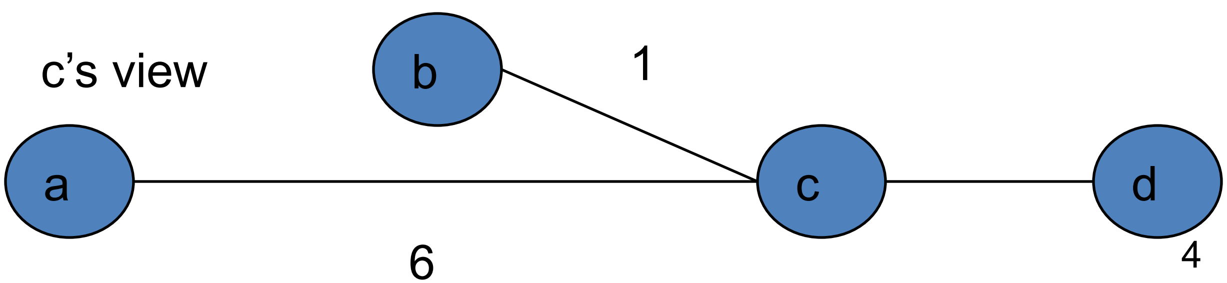
a’s view



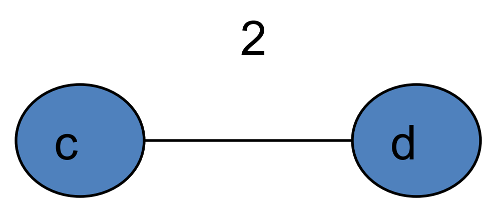
b’s view



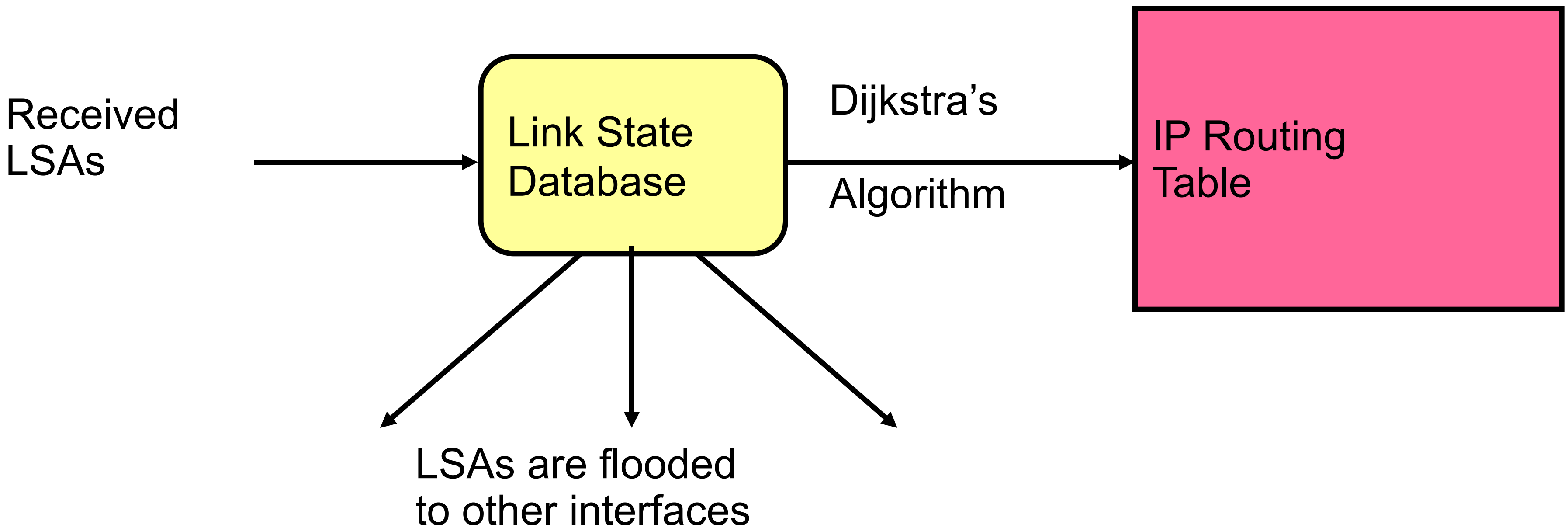
c’s view



d’s view



Operation of a Link State Routing protocol



Link State Advertisement (LSA)

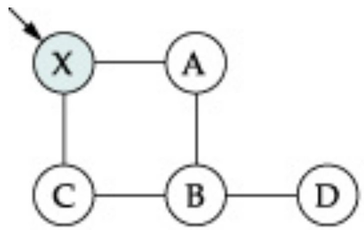
- Also known as Link State Packet (LSP)
  - id of the node that created the LSA/LSP
  - cost of link to each directly connected neighbor
  - sequence number (SeqNo)
  - time-to-live (TTL) for this packet

Reliable flooding

- LSPs are transmitted reliably between adjacent routers
  - ACK and retransmission

- For a node x, if it receives an LSA sent by y
  - Stores LSA if it does not have a copy
  - Otherwise, compares SeqNo. If newer, store; otherwise discard
  - If a new LSA, floods LSA to all neighbors except the incoming neighbor

An example of reliable flooding



(a)

When to flood an LSP

- Triggered if a link's state has changed
  - Detecting failure
    - Neighbors exchange hello messages
    - If not receiving hello, assume dead
- Periodically generating a new LSP

Path computation

Dijkstra's Shortest Path Algorithm for a Graph

Input: Graph **(N,E)** with
N the set of nodes and **E** the set of edges
c_{vw} link cost ($c_{vw} = \infty$ if $(v,w) \notin E$, $c_{vv} = 0$)
s source node.
Output: D_n cost of the least-cost path from node s to node n

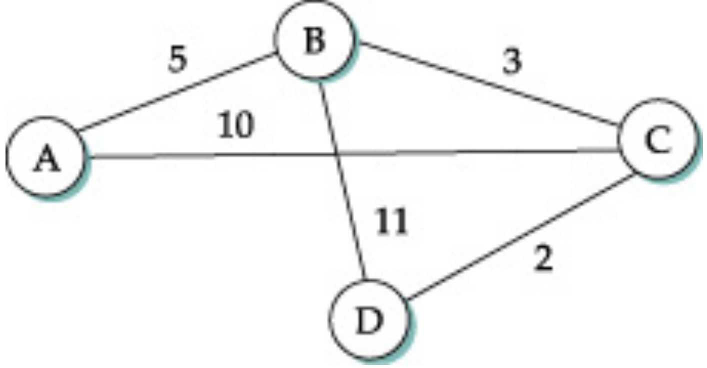
```
M = {s};
for each  n ∉ M
        D_n  = c_sn;
while (M ≠ all nodes) do
        Find w ∉ M for which D_w = min{D_j ; j ∉ M};
        Add w to M;
        for each neighbor n of w and n ∉ M
                D_n = min[ D_n, D_w + c_wn ];
                Update route;
enddo
```

Practical Implementation: forward search algorithm

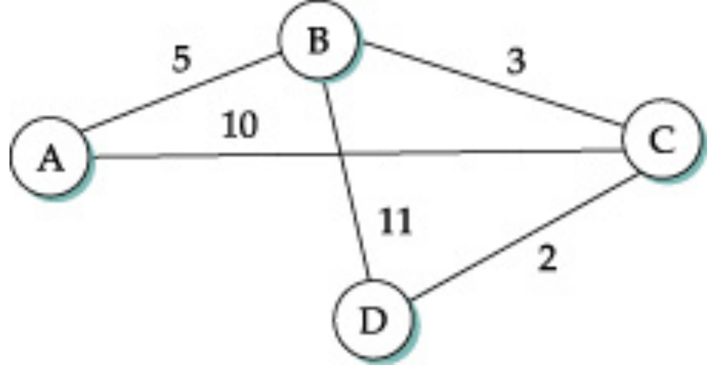
- More efficient: extracting min from a smaller set rather than the entire graph

- Two lists: Tentative and Confirmed
- Each entry: (destination, cost, NextHop)

1. Confirmed = {(s,0,s)}
2. Let Next = Confirmed.last
3. For each Nbr of Next
   - Cost = Next.cost + Next → Nbr
     - If Nbr not in Confirmed or Tentative
       - Add (Nbr, Cost, Nbr) to Tentative if Next.Nexthop is s
       - Add (Nbr, Cost, Next.Nexthop) to Tentative if Next.Nexthop is not s
     - If Nbr is in Tentative and Cost is less than Nbr.Cost
       - Update Nbr.Cost to Cost and Nbr.Nexthop to Next.Nexthop
4. If Tentative not empty, pick the entry with smallest cost in Tentative and move it to Confirmed, and return to Step 2
   - Pick the smallest cost from a smaller list Tentative, rather than the rest of the graph



Step	Confirmed	Tentative
1	(D,0,D)	
2		
3		
4		
5		
6		
7		

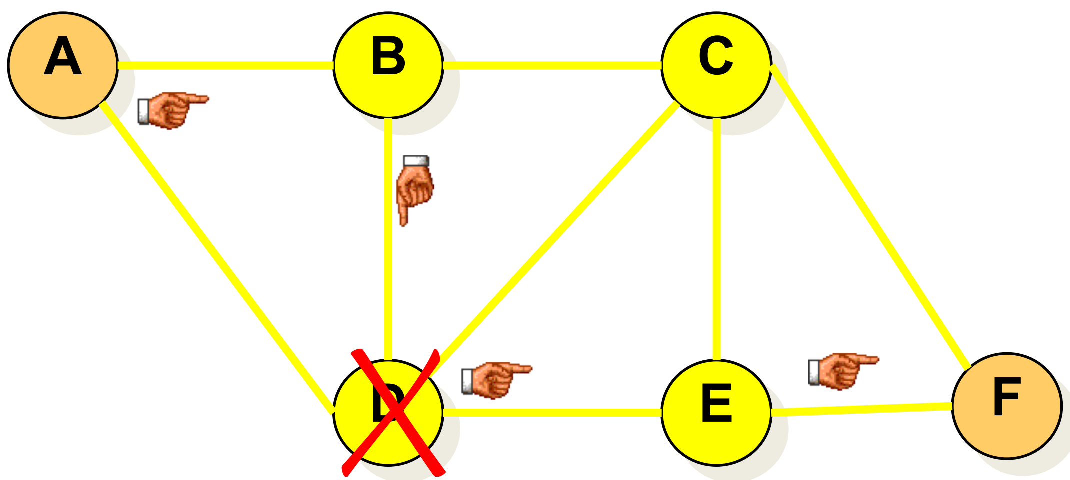


Step	Confirmed	Tentative
1	(D,0,D)	
2	(D,0,D)	(B,11,B), (C,2,C)
3	(D,0,D), (C,2,C)	(B,11,B)
4	(D,0,D), (C,2,C)	(B,5,C) (A,12,C)
5	(D,0,D), (C,2,C), (B,5,C)	(A,12,C)
6	(D,0,D),(C,2,C),(B,5,C)	(A,10,C)
7	(D,0,D),(C,2,C),(B,5,C), (A,10,C)	

Distance Vector vs. Link State Routing

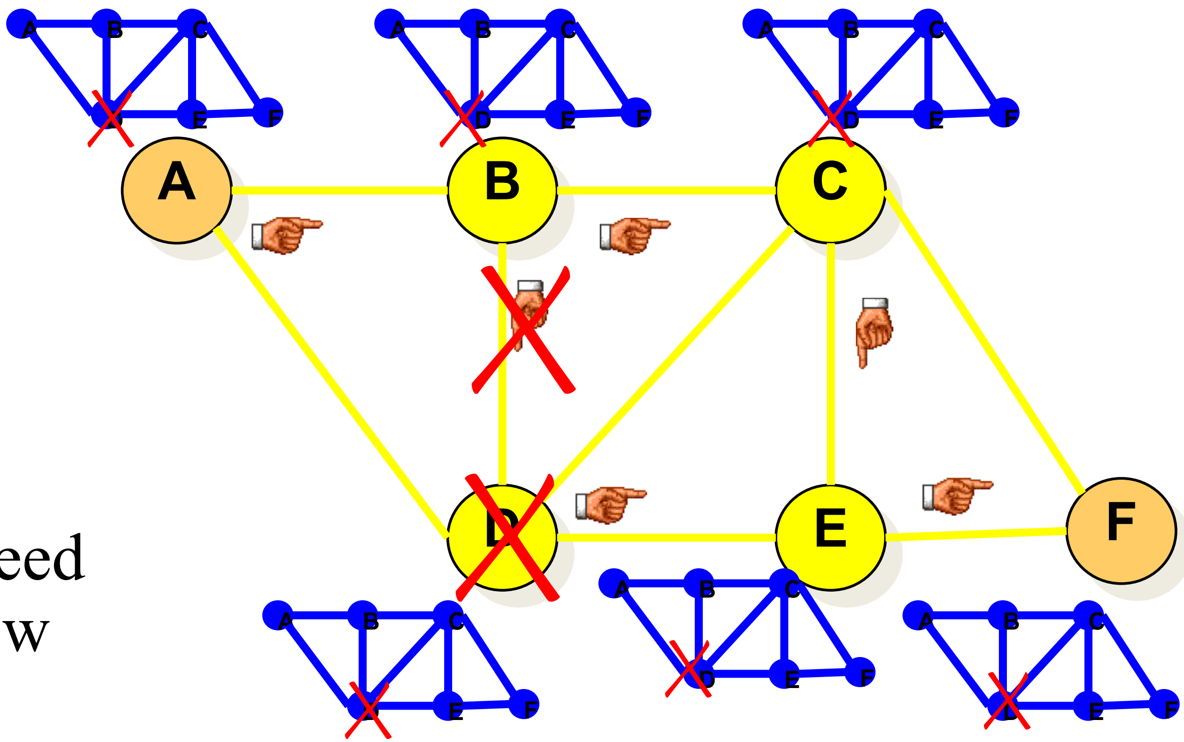
- DV only sees next hop “direction”
  - Node A: to reach F go to B
  - Node B: to reach F go to D
  - Node D: to reach F go to E
  - Node E: go directly to F

- Count to infinity



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



OSPF

- OSPF = Open Shortest Path First
  - Open stands for open, non-proprietary

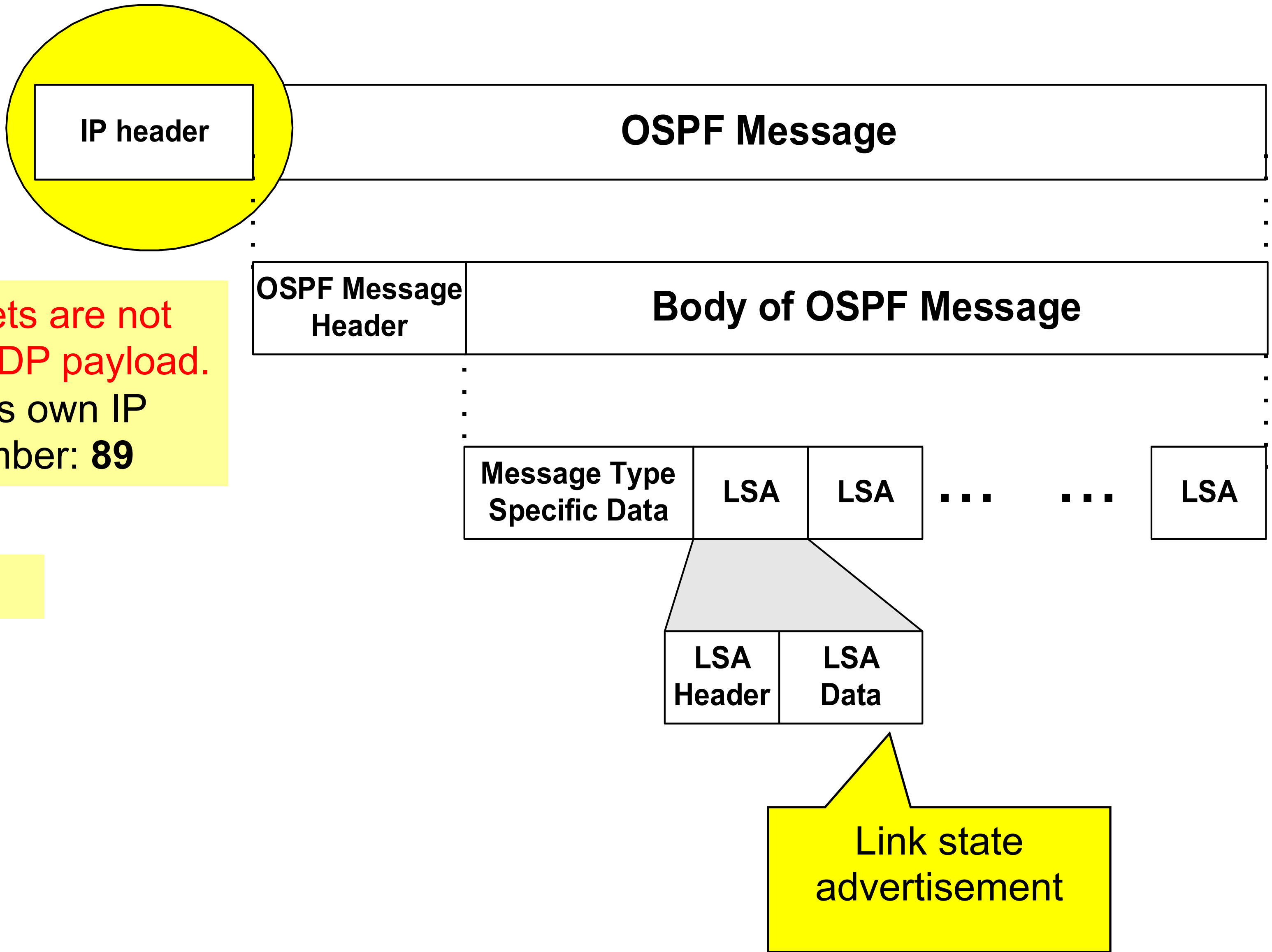
- A link state routing protocol
- OSPF has significant complexity
  - RIP (RFC 2453 ~ 40 pages)
  - OSPF (RFC 2328 ~ 250 pages)
- History:
  - 1989: RFC 1131 OSPF Version 1
  - 1991: RFC1247 OSPF Version 2
  - 1994: RFC 1583 OSPF Version 2 (revised)
  - 1997: RFC 2178 OSPF Version 2 (revised)
  - 1998: RFC 2328 OSPF Version 2 (current version)

Features of OSPF

- Provides authentication of routing messages

- Allows hierarchical routing
  - Divide a domain into sub-areas

OSPF Packet Format



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

TTL: set to 1

OSPF Common header

OSPF Message Header	Body of OSPF Message

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

version	type	message length
source router IP address		
Area ID		
checksum	authentication type	
authentication		
authentication		

32 bits

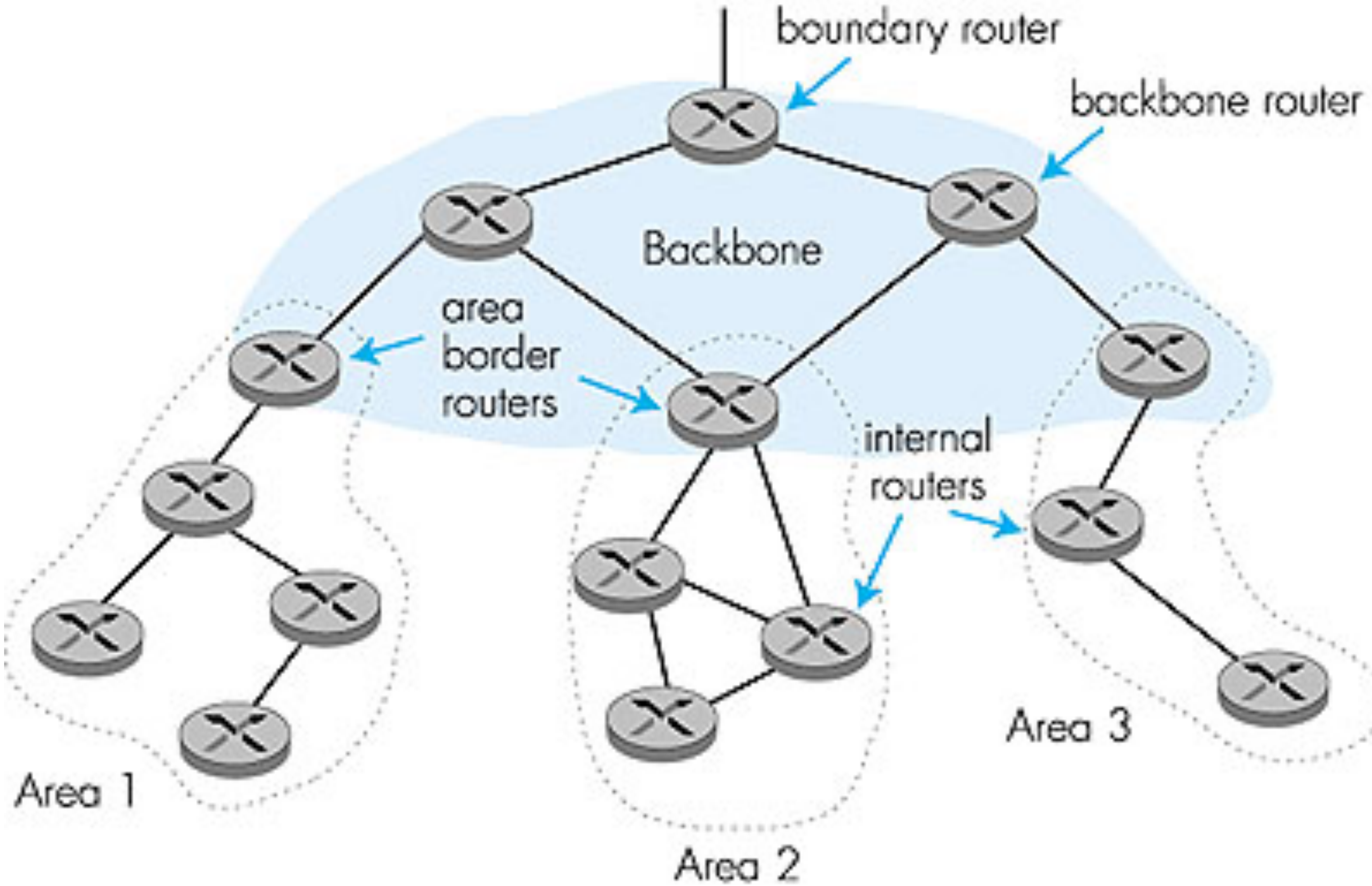
ID of the Area from which the packet originated

0: no authentication
1: Cleartext password
2: MD5 checksum

How to set link cost?

- Still an open question
- Design choice 1: all to 1
- Design choice 2: based on load of a link
  - Dynamically change
  - Often not used

Hierarchical OSPF



Hierarchical OSPF

- Two-level hierarchy: local area, backbone.
  - Link-state advertisements only in area
  - Each node has detailed area topology; only know direction (shortest path) to nets in other areas.
- **Area border routers**: "summarize" distances to nets in own area, advertise to other Area Border routers.

OSPF summary

- A link-state routing protocol
- Each node has a map of the network and uses Dijkstra to compute shortest paths
- Nodes use reliable flooding to keep an identical copy of the network map