CompSci 356: Computer Network Architectures

Lecture 13: Dynamic routing protocols: Link State
Chapter 3.3.3, 3.2.9

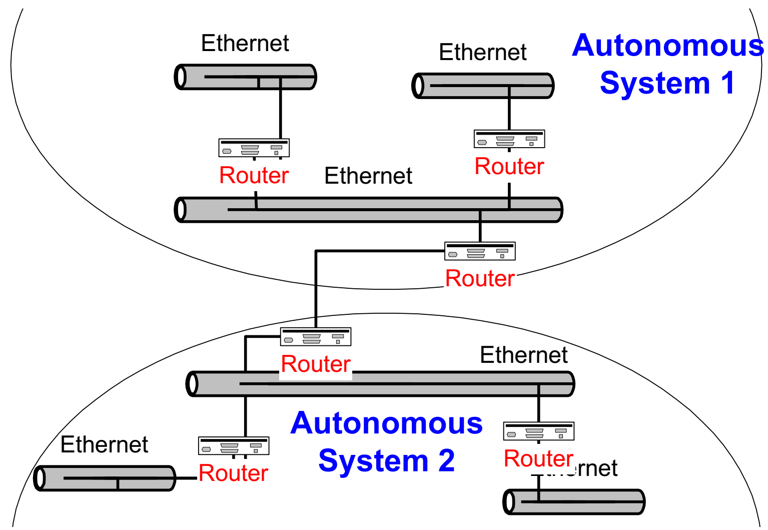
Xiaowei Yang
xwy@cs.duke.edu

Today

- Clarification on RIP
- Link-state routing
  - Algorithm
  - Protocol: Open shortest path first (OSPF)
- IP tunnels

Will a router’s routing table keep growing if we run RIP over time?

Autonomous systems

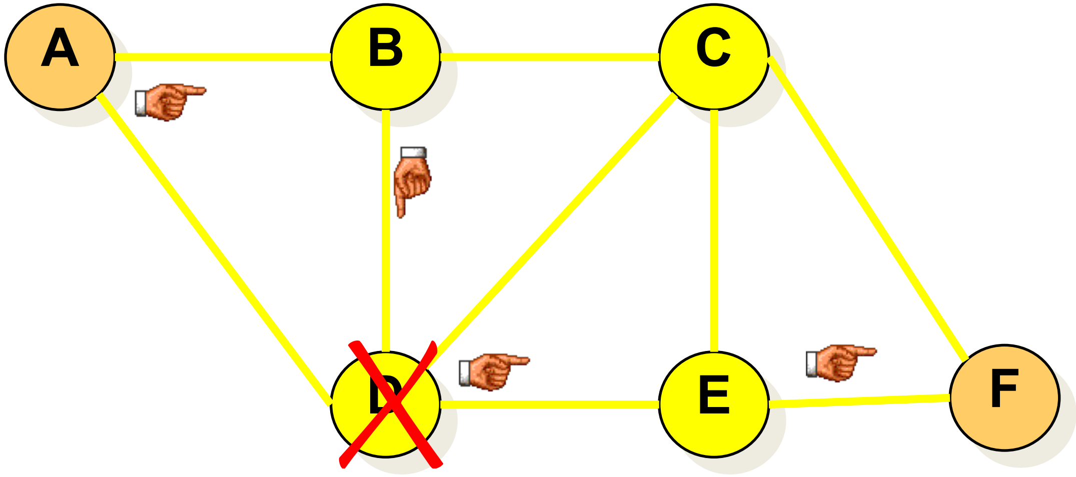


- Aggregate routers into regions, “autonomous systems” (AS) or domain
- Routers in the same AS run the same routing protocol
  - “intra-AS” or intra-domain routing protocol
  - routers in different AS can run different intra-AS routing protocol

Distance Vector vs. Link State Routing

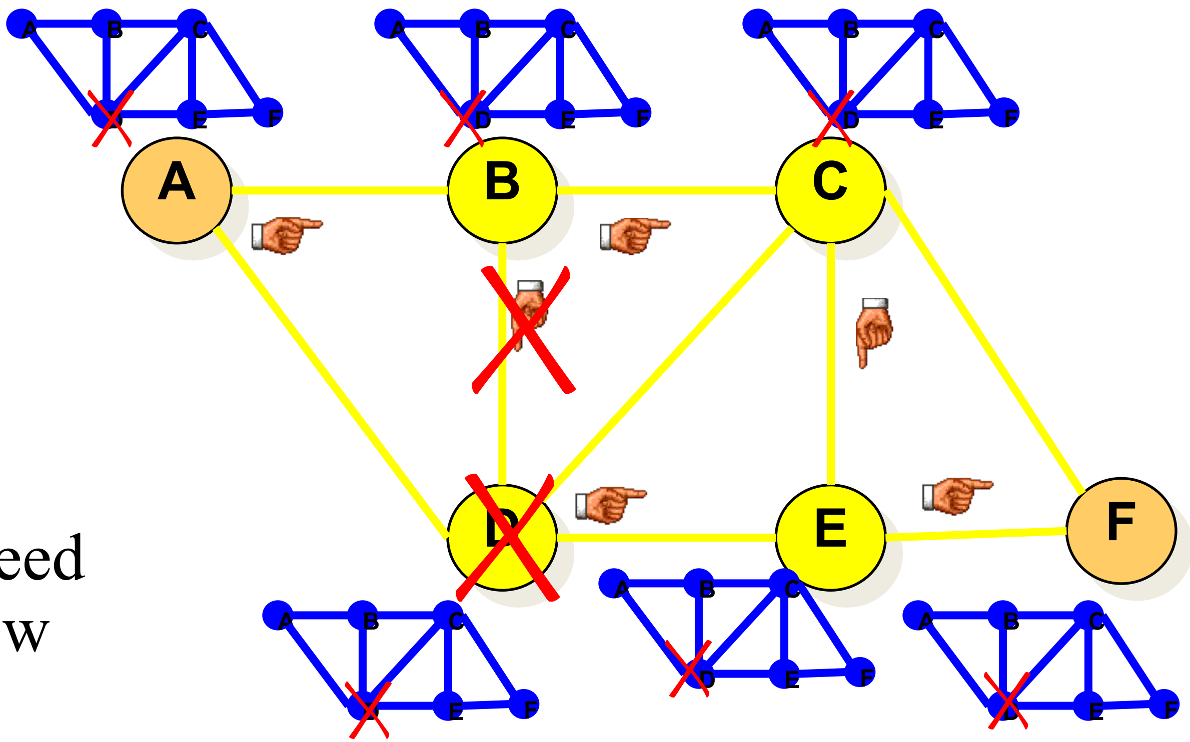
- DV only sees next hop “direction”
  - Node A: to reach F go to B
  - Node B: to reach F go to D
  - Node D: to reach F go to E
  - Node E: go directly to F

- Wrong directions lead to wrong routes
  – Count to infinity



Distance Vector vs. Link State Routing

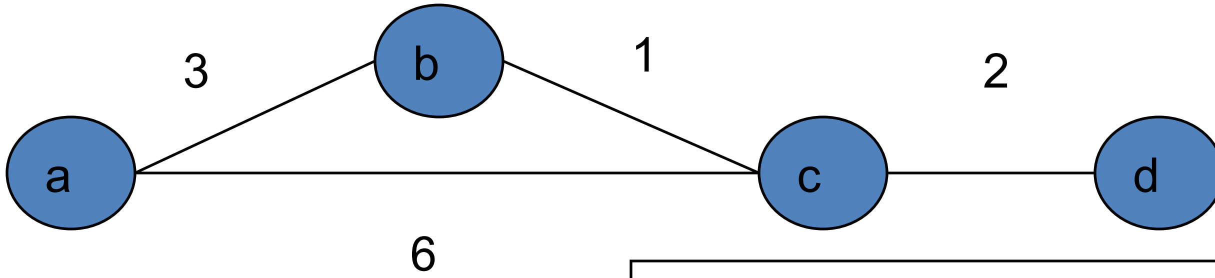
- In link state routing, each node has a complete map of the topology
- If a node fails, each node can calculate the new route
- Challenge: All nodes need to have a consistent view of the network



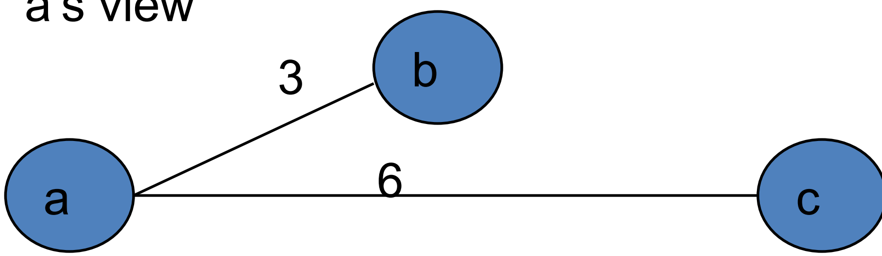
Link State Routing: Basic operations

1. Each router establishes *link adjacency*
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

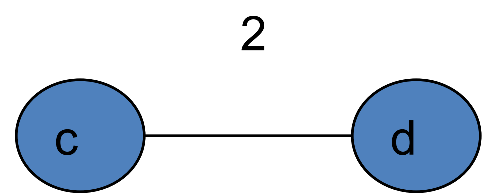


a’s view

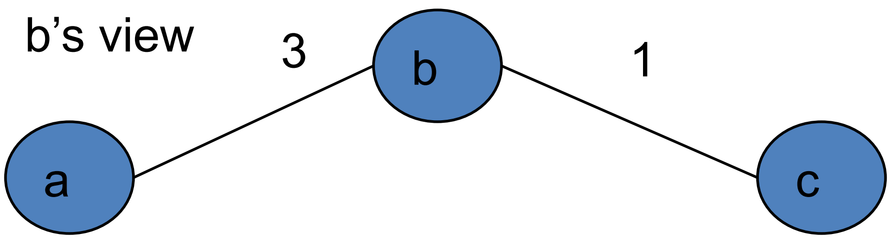


Collecting all pieces yield
a complete view of the network!

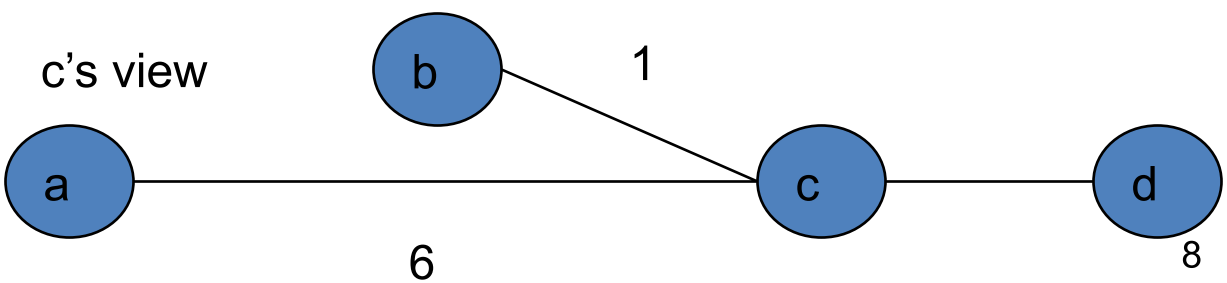
d’s view



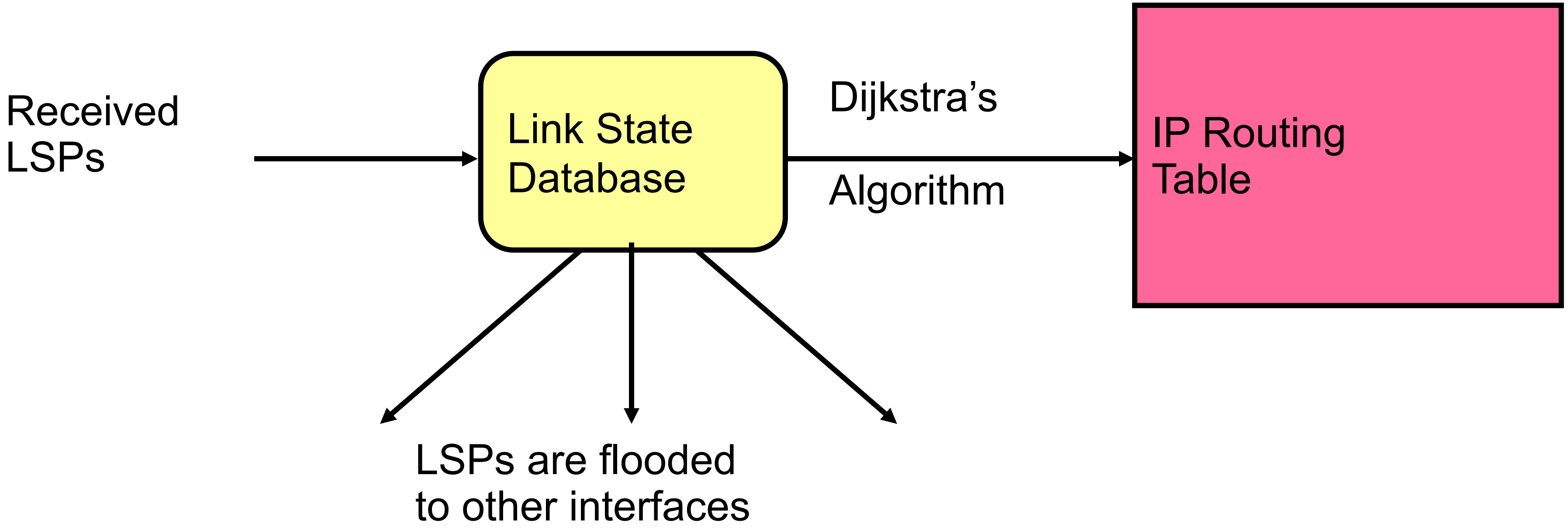
b’s view



c’s view



Operation of a Link State Routing protocol



Reliable flooding

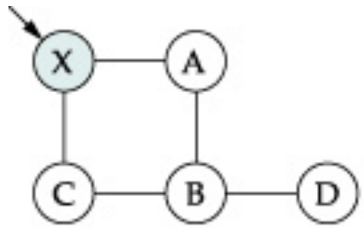
- We've learned a flooding algorithm used by Ethernet switches
- Question: why is it insufficient for link-state routing?
  - Lost LSAs may result in inconsistent topologies at different routers
  - Inconsistent topologies may lead to routing loops

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(y) if it does not have a copy
  – Otherwise, compares SeqNo. If newer, store; otherwise discard
  – If a new LSA(y), floods LSA(y) 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
- Periodic generating a new LSA
  - Fault tolerance (what if LSA in memory is corrupted?

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

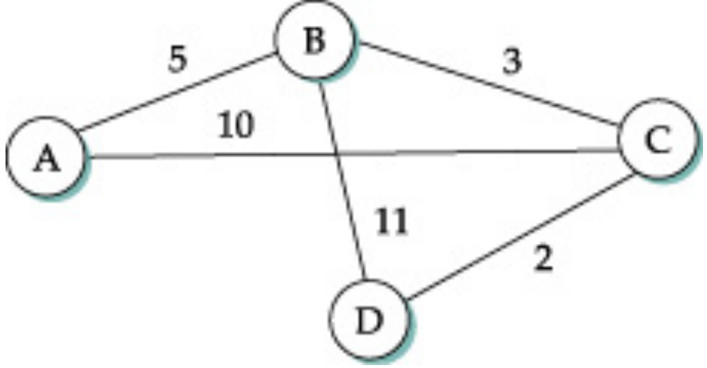
```
M = {s};
for each  n ∉ M
        D_n  = c_sn;
while (M ≠ all nodes) do
        Find w ∉ M for which D_w = min{D_j ; j ∉ M};
        Add w to M;
        for each neighbor n of w and n ∉ M
                D_n = min[ D_n, D_w + c_wn ];
                Update route;
enddo
```

Practical Implementation: forward search algorithm

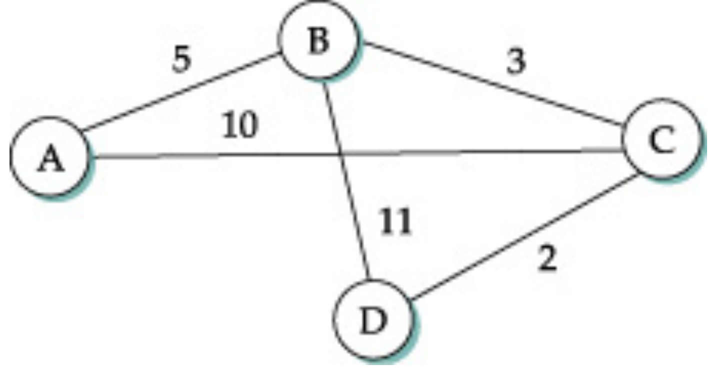
- More efficient: extracting min from a smaller set rather than the entire graph

- 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 = my→Next + Next → Nbr
     - If Neighbor not in Confirmed or Tentative
       - Add (Nbr, Cost, my.Nexthop(Next)) to Tentative
   - If Nbr is in Tentative, and Cost is less than Nbr.Cost, update Nbr.Cost to Cost
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,-)	
2		
3		
4		
5		
6		
7		



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

OSPF

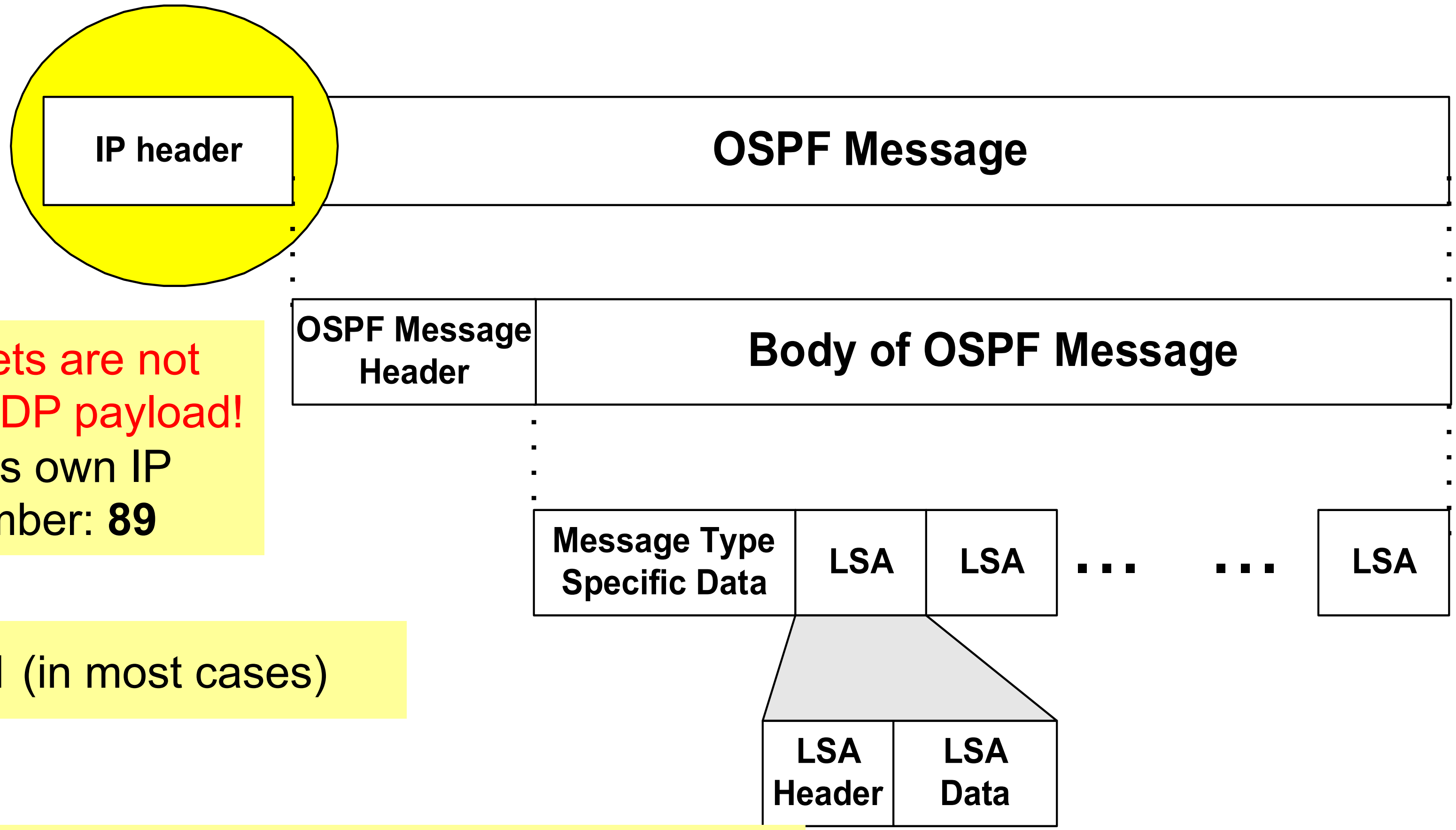
- OSPF = Open Shortest Path First
  - Open stands for open, non-proprietary

- A link state routing protocol
- The complexity of OSPF is significant
  - 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
  - Similar to RIP 2
- Allows hierarchical routing
  - Divide a domain into sub-areas
- Enables load balancing by allowing traffic to be split evenly across routes with equal cost

OSPF Packet Format



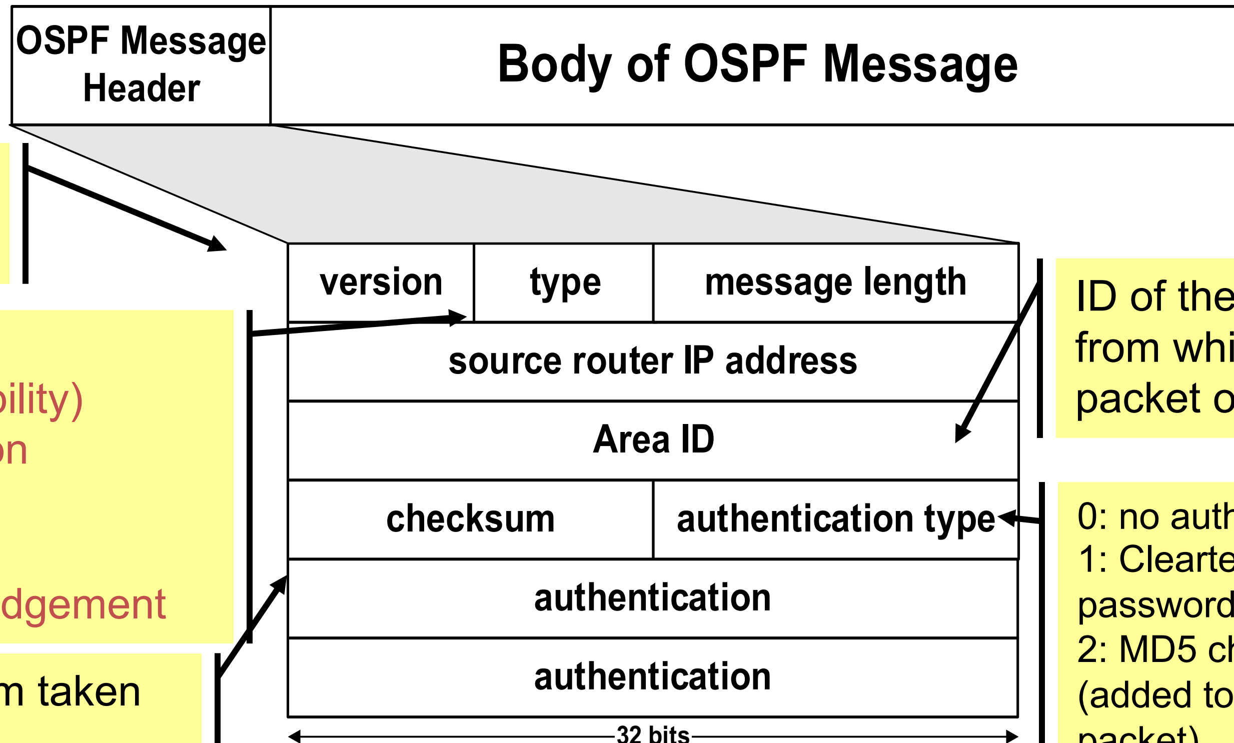
OSPF packets are not carried as UDP payload!
OSPF has its own IP protocol number: **89**

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)

Link state advertisement

OSPF Common header



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

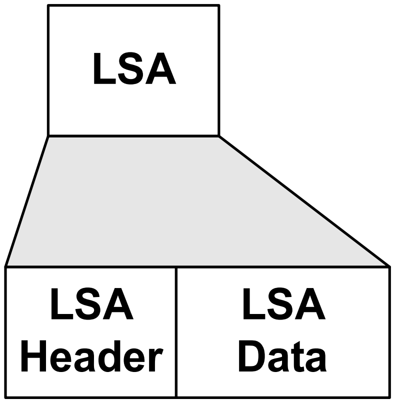
ID of the Area from which the packet originated

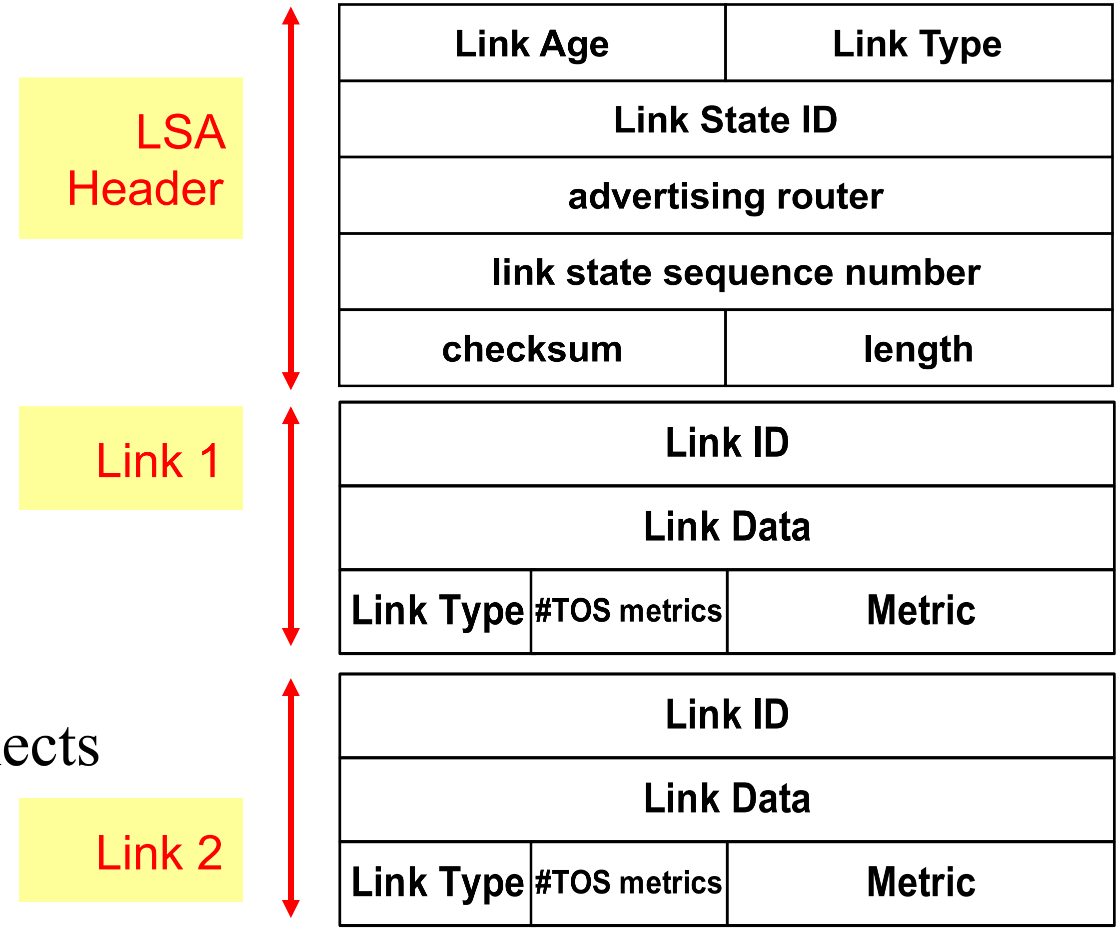
0: no authentication
1: Cleartext password
2: MD5 checksum (added to end packet)

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





LSAs

- Type 1: cost of links between routers
- Type 2: networks to which the router connects
- Others: hierarchical routing

Type 1 LSA

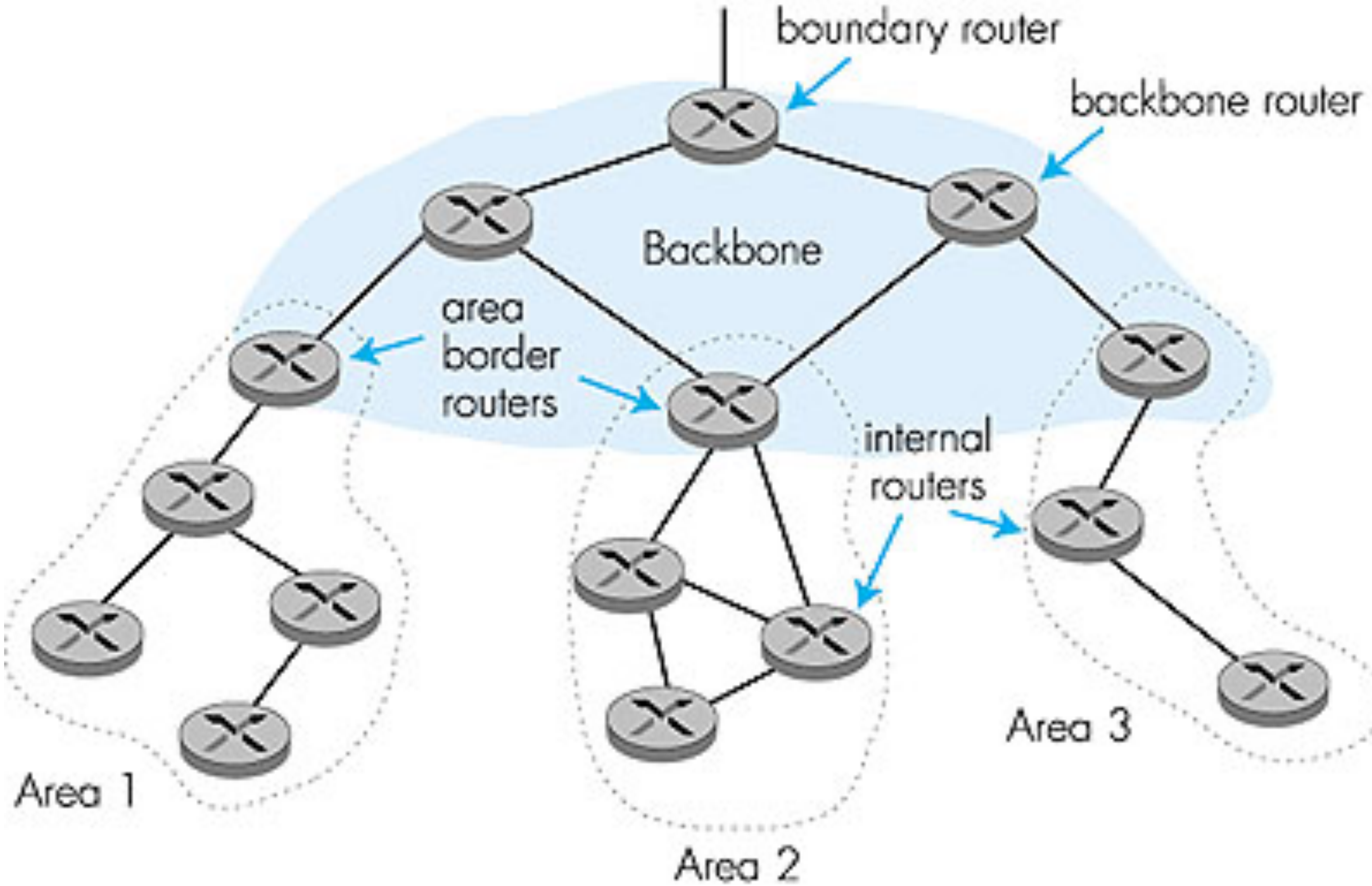
- Link state ID and Advertising router are the same, 32-bit router ID

- Link ID: router ID at the other end of the link
- Link Data: identify parallel links
- Metric: cost of the link
- Type: types of the link e.g., point-to-point

Open question

- How to set link metrics?
- Design choice 1: all to 1
- Design choice 2: based on load
  - Problems?
- In practice: static

Hierarchical OSPF

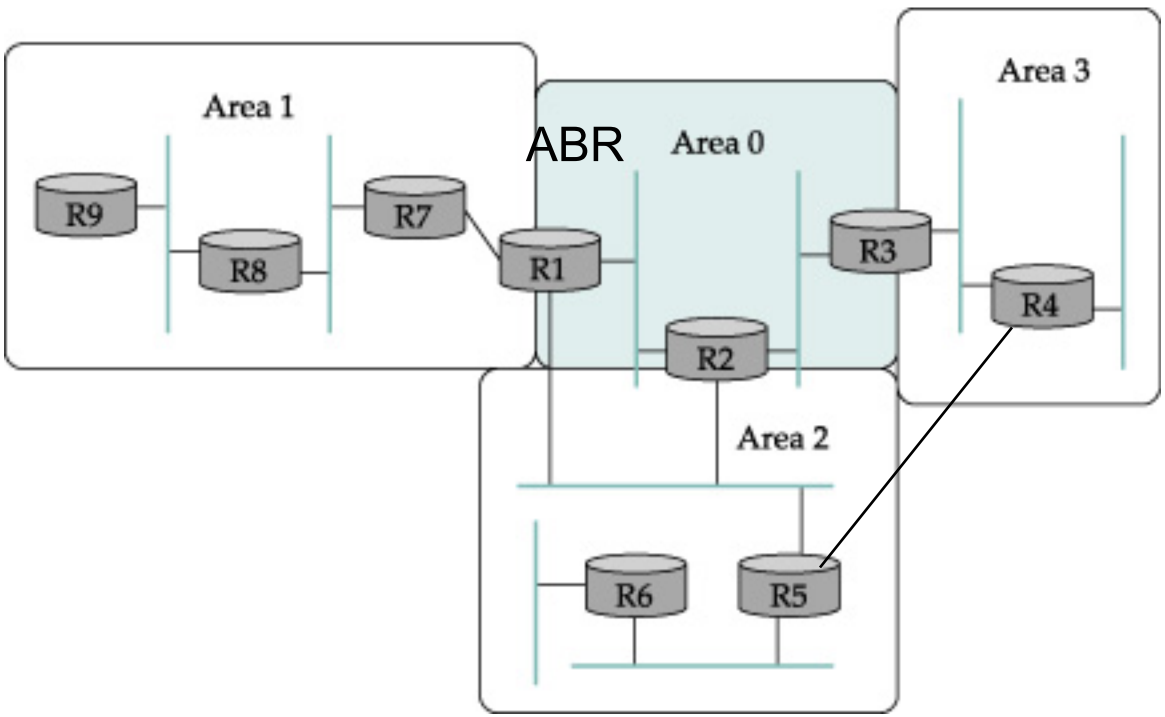


Hierarchical OSPF

- Two-level hierarchy: local area, backbone.
  - Link-state advertisements only in area
  - Each nodes 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.
- **Backbone routers:** run OSPF routing limited to backbone.

Scalability and Optimal Routing

- A frequent tradeoff in network design
- Hierarchy introduces information hiding



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

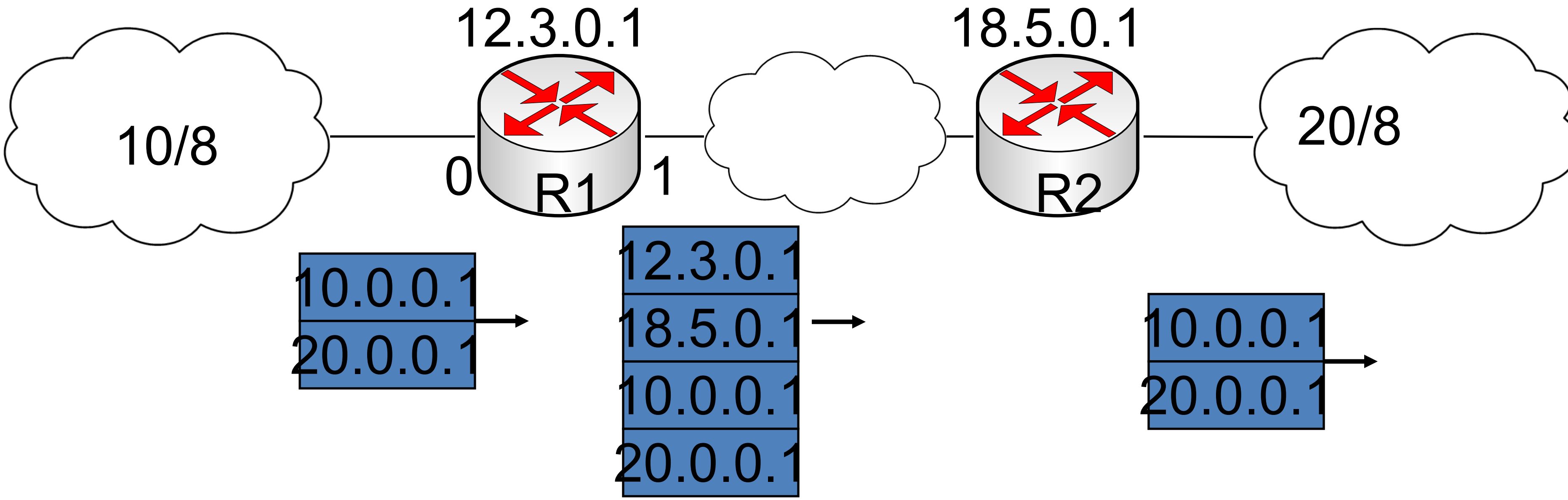
Today

- Clarification on RIP
- Link-state routing
  - Algorithm
  - Protocol: Open shortest path first (OSPF)
- IP tunnels

IP tunnels

- Tunnels
  - A technique used in many scenarios
    - VPN, IPv4-v6 transition, Mobile IP, Multicast, Non-IP forwarding, IPsec

What is a tunnel



- A "pseudowire", or a virtual point-to-point link
- The head router encapsulates a packet in an outer header destined to the tail router

Virtual interface

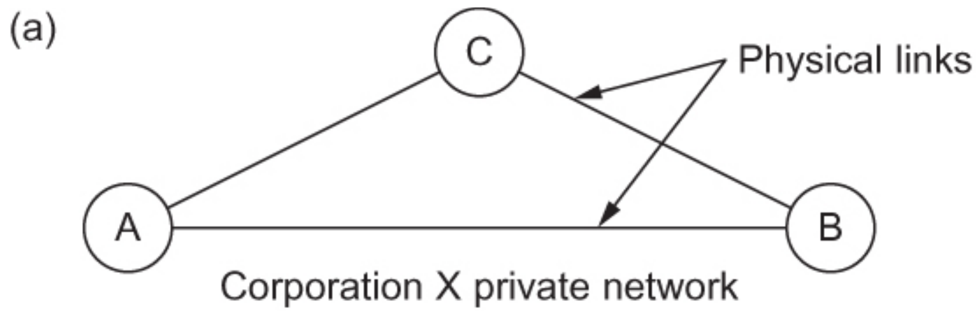
- A router adds a tunnel header for packets sent to a virtual interface

NetworkNum	nextHop
10/8	ether0
20/8	tun0
0/0	ether1

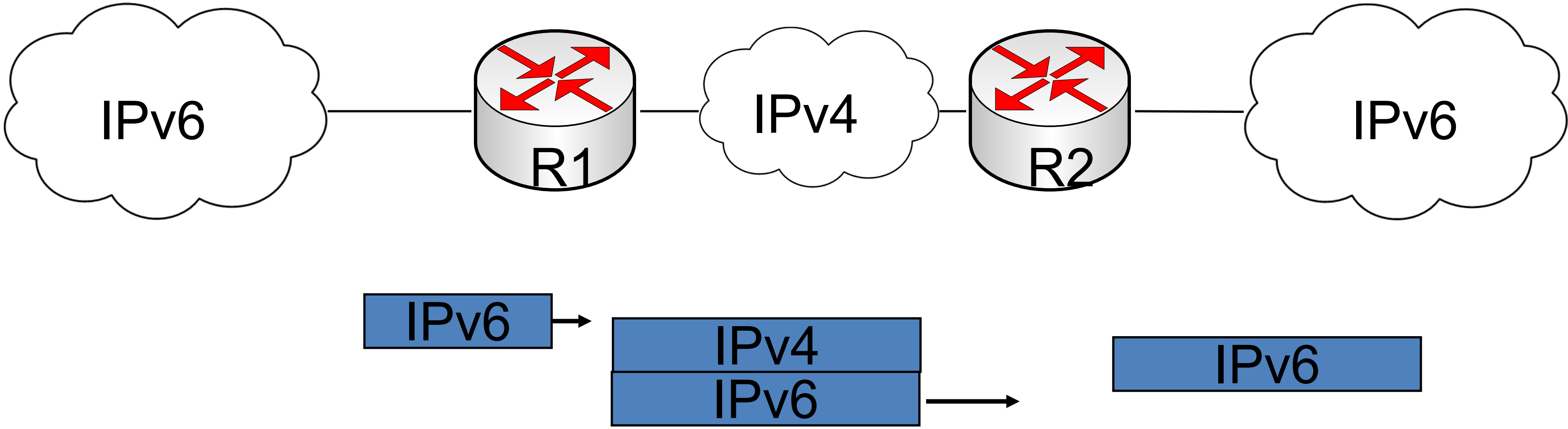
Tunnel applications

- Traversing a region of network with a different addressing format or with insufficient routing knowledge
- Building virtual private networks

FIGURE 3.26 An example of virtual private networks: (a) two separate private networks; (b) two virtual private networks sharing common switches.

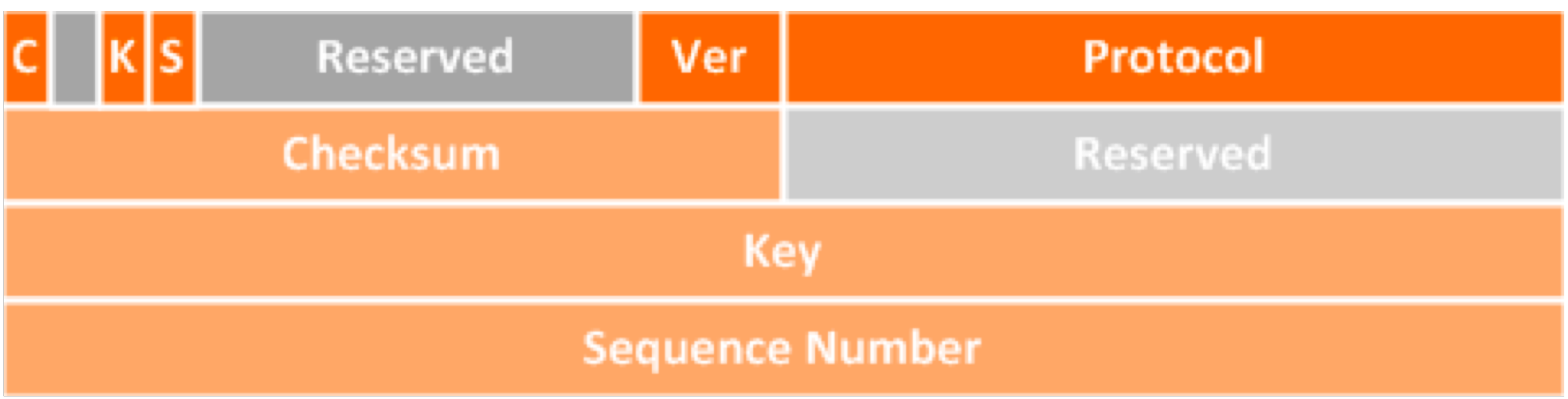


IPv4-v6 transition



Generic Routing Encapsulation (GRE)

- Defined in RFC 2784 and updated by RFC 2890



Today

- Clarification on RIP
- Link-state routing
  - Algorithm
  - Protocol: Open shortest path first (OSPF)
- IP tunnels
- Next: BGP