UNIT-3

Routing and Next Generation IP

Routing in the Internet

Forwarding versus Routing

- Forwarding:
  - to select an **output port** based on **destination address and routing table**
- Routing:
  - process by which **routing table is built**

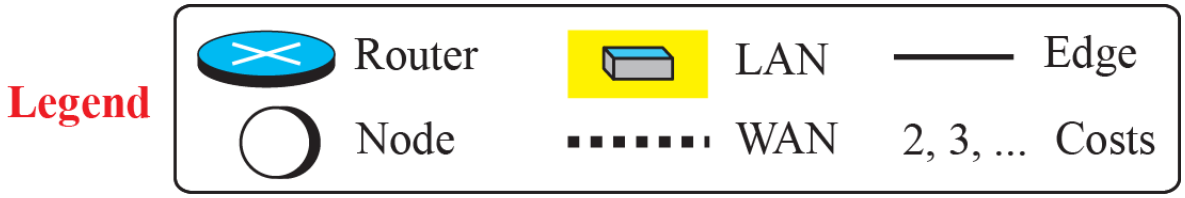
UNICAST ROUTING

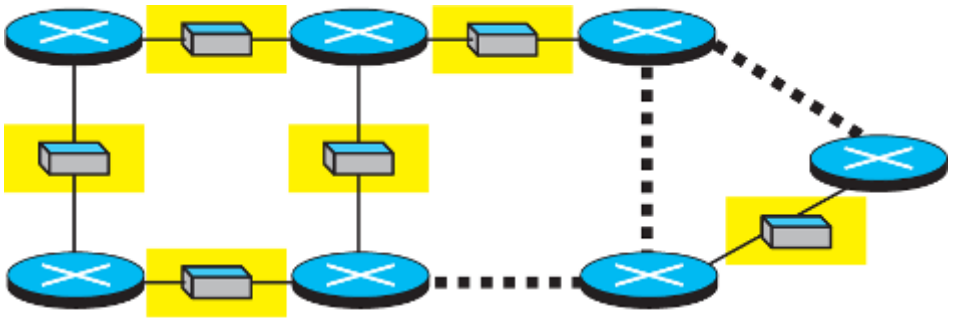
In an internet, **the goal of the network layer is to deliver a datagram from its source to its destination** or destinations. If a datagram is destined for **only one destination (one-to-one delivery), we have unicast routing.** In this section and the next, we discuss only unicast routing; multicast and broadcast routing will be discussed later in the chapter.

4.3.1 General Idea

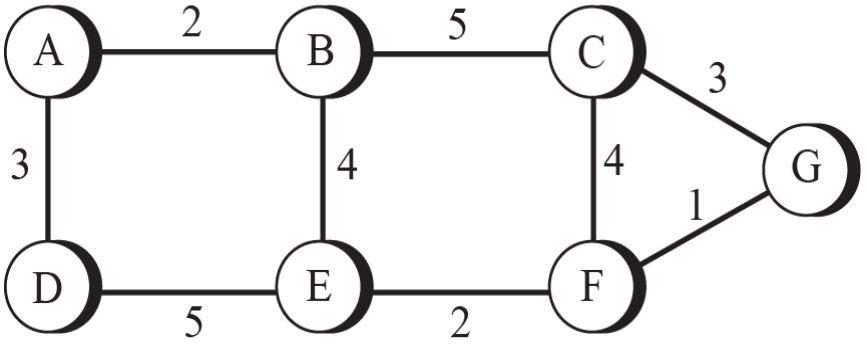
In unicast routing, a packet is **routed, hop by hop, from its source to its destination by the help of forwarding tables.** The source host needs no forwarding table because it delivers its packet to the default router in its local network. The destination host needs no forwarding table either because it receives the packet from its default router in its local network. This means that only the routers that glue together the networks in the internet need forwarding tables.

Graph abstraction





a. An internet



b. The weighted graph

key question: what is the least-cost path between ant two nodes ?

routing algorithm: algorithm that finds that least cost path

Routing Algorithm

Several routing algorithms have been designed in the past. **The differences between these methods are in the way they interpret the least cost and the way they create the least-cost tree** for each node. In this section, we discuss the common algorithm; later we show how a routing protocol in the Internet implements one of these algorithms.

- The basic problem of routing is to find the **lowest-cost path** between any two nodes

  Where the cost of a path equals the sum of the costs of all the edges that make up the path

Routing algorithm classification

Q: global or decentralized information?

global:

- all routers have complete topology, link cost info
- "link state" algorithms

decentralized:

- router knows physically-connected neighbors, link costs to neighbors
- iterative process of computation, exchange of info with neighbors
- "distance vector" algorithms

Q: static or dynamic?

static:

- routes change slowly over time

dynamic:

- routes change more quickly
  - periodic update
  - in response to link cost changes

Two main classes of protocols

- **Distance Vector –uses Bellman Ford's algorithm (RIP)**
- **Link State – uses Dijkstra's algorithm (OSPF)**

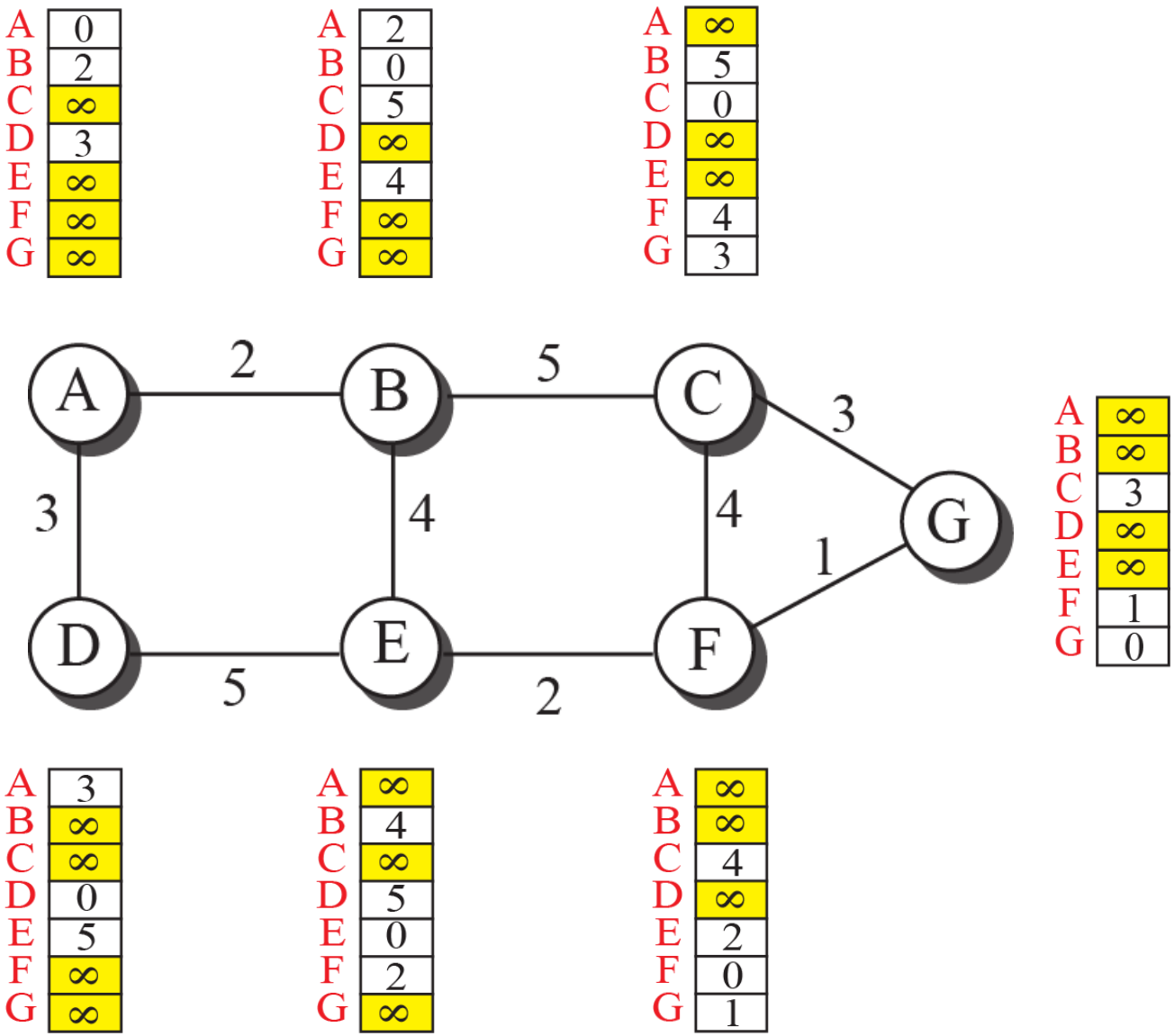
Distance Vector Routing

- Each node constructs a **one dimensional array (a vector)** containing the **“distances” (costs)** to all other nodes and distributes that vector to its immediate neighbors
- Starting assumption is that each node knows the cost of the link to each of its directly connected neighbors
- A link that is down is assigned an infinite cost.

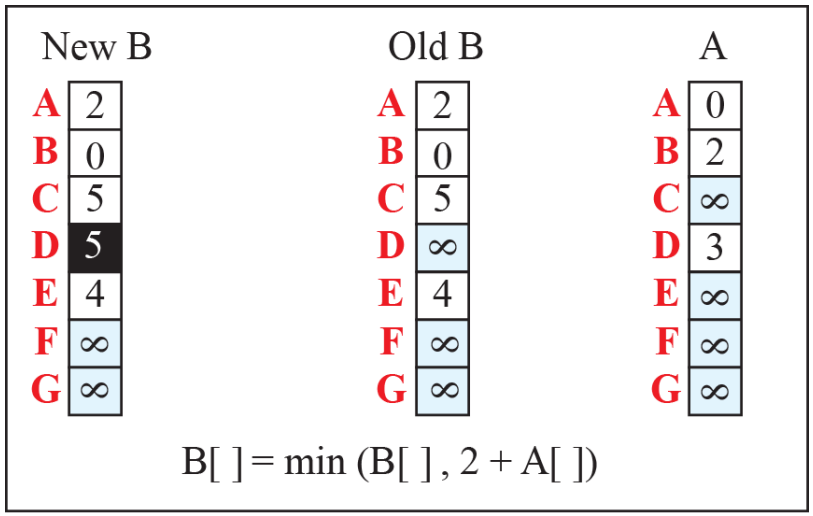
Distance Vector Routing

- **Next step** – every node sends a message to its directly connected neighbors containing its personal list of distances.
- In the absence of any topology changes, after few exchanges every router knows the distance to every other router. This state is called **Convergence state.**
- Then **data will be forwarded as per the entries in the routing table.**

The first distance vector for an internet

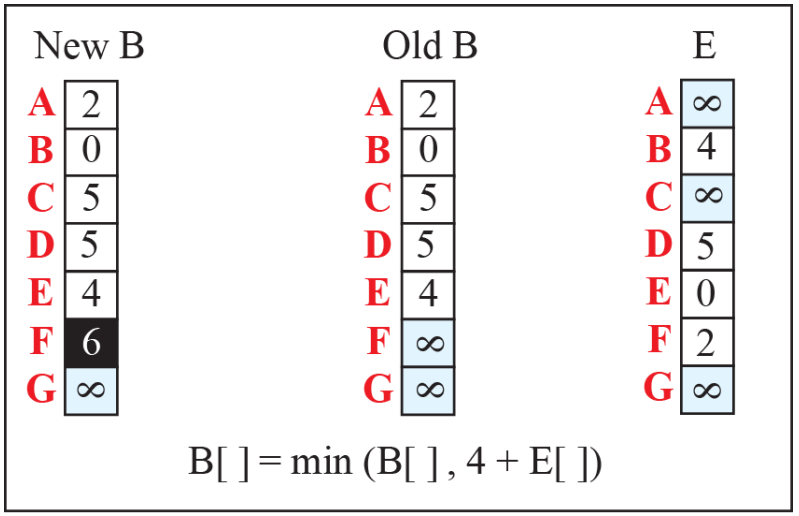


Updating distance vectors



a. First event: B receives a copy of A's vector.

Note:
X[ ]: the whole vector



b. Second event: B receives a copy of E's vector.

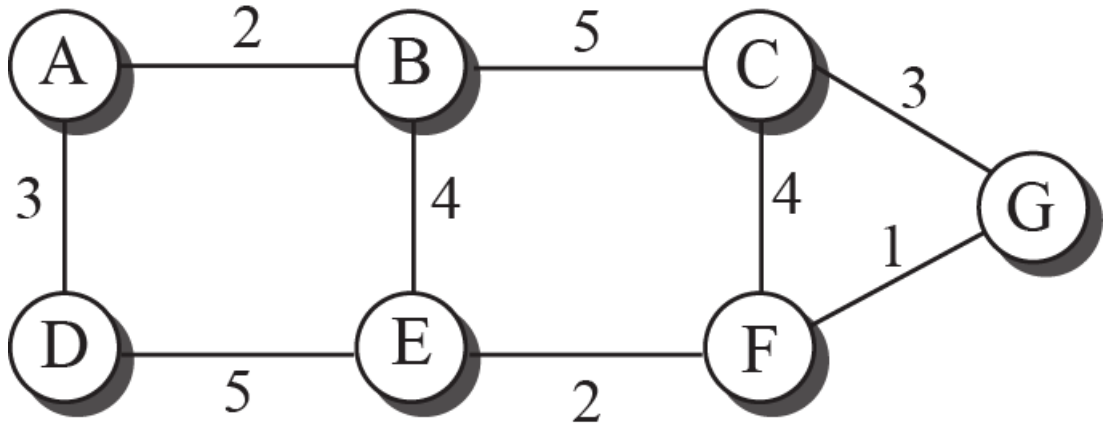
The final distance vector for an internet

A	0
B	2
C	7
D	3
E	6
F	8
G	9

A	2
B	0
C	5
D	5
E	4
F	6
G	7

A	7
B	5
C	0
D	10
E	6
F	4
G	3

A	9
B	7
C	3
D	8
E	3
F	1
G	0



A	3
B	5
C	10
D	0
E	5
F	7
G	8

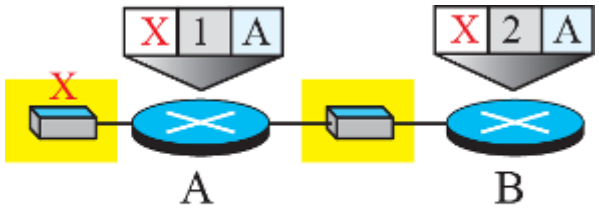
A	6
B	4
C	6
D	5
E	0
F	2
G	3

A	8
B	6
C	4
D	7
E	2
F	0
G	1

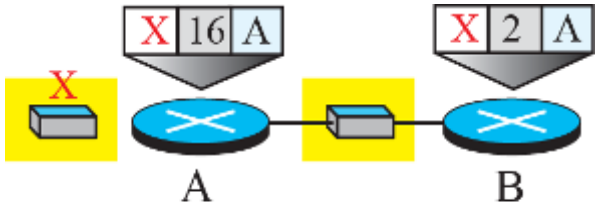
Updates

- Every **T seconds** each router sends its table to its neighbor. Each router then updates its table based on the new information **(periodic updates)**
- Whenever there is a topology change then the router will send the **triggered updates.**

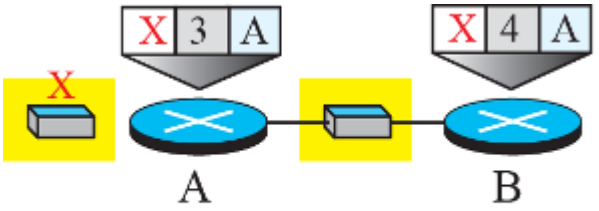
Two-node instability (count to infinity)



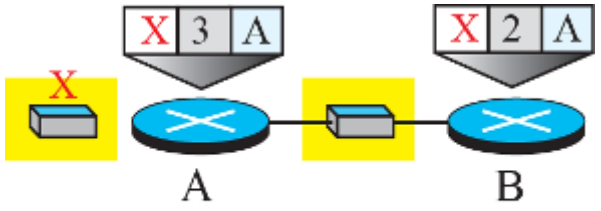
a. Before failure



b. After link failure

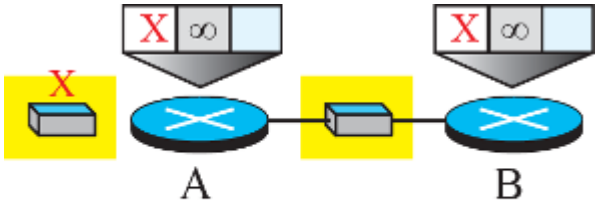


d. After B is updated by A



c. After A is updated by B

• • •



e. Finally

Link State Routing

Strategy: Send to all nodes (not just neighbors) information about directly connected links (not entire routing table).

Link state Routing uses **two mechanisms:**

1) Reliable dissemination of link state information

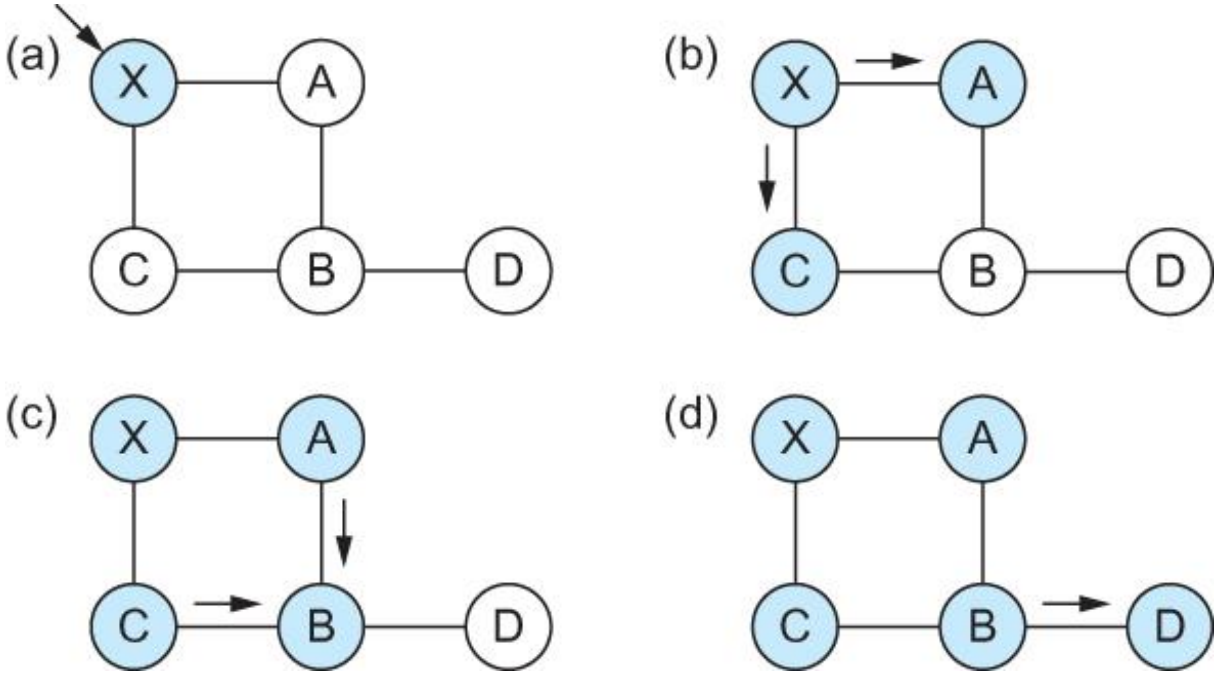
2) Route Calculation (formation of least cost trees)

Reliable flooding: give link state information from every node to everybody else in the network reliably.

- **Link State Packet (LSP**)
  - id of the node that created the LSP
  - cost of link to each directly connected neighbor
  - sequence number (SEQNO)
  - time-to-live (TTL) for this packet
  - 1 and 2 used for route calculation and
  - 3 and 4 used for reliable flooding.

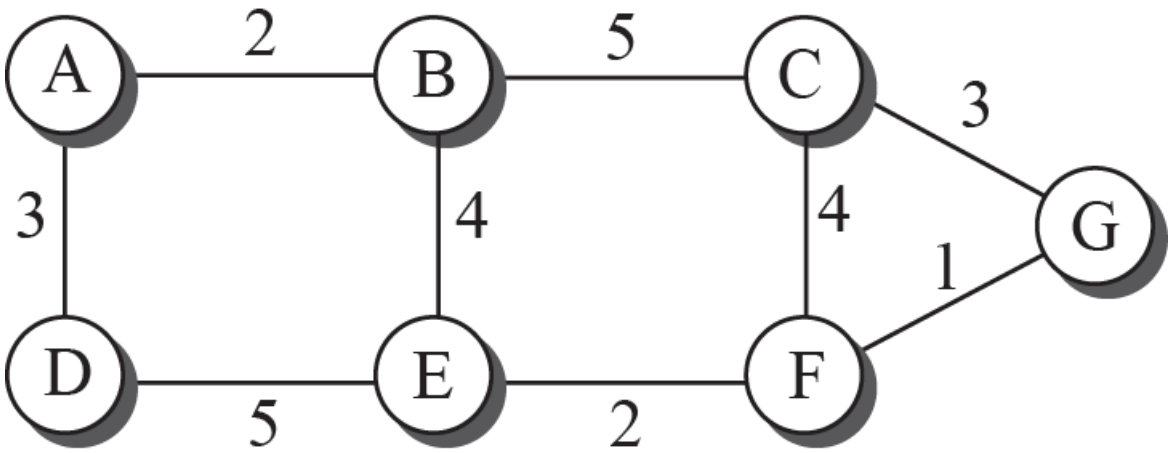
Link State Routing

Reliable Flooding



Flooding of link-state packets. (a) LSP arrives at node X; (b) X floods LSP to A and C; (c) A and C flood LSP to B (but not X); (d) flooding is complete

Example of a link-state database

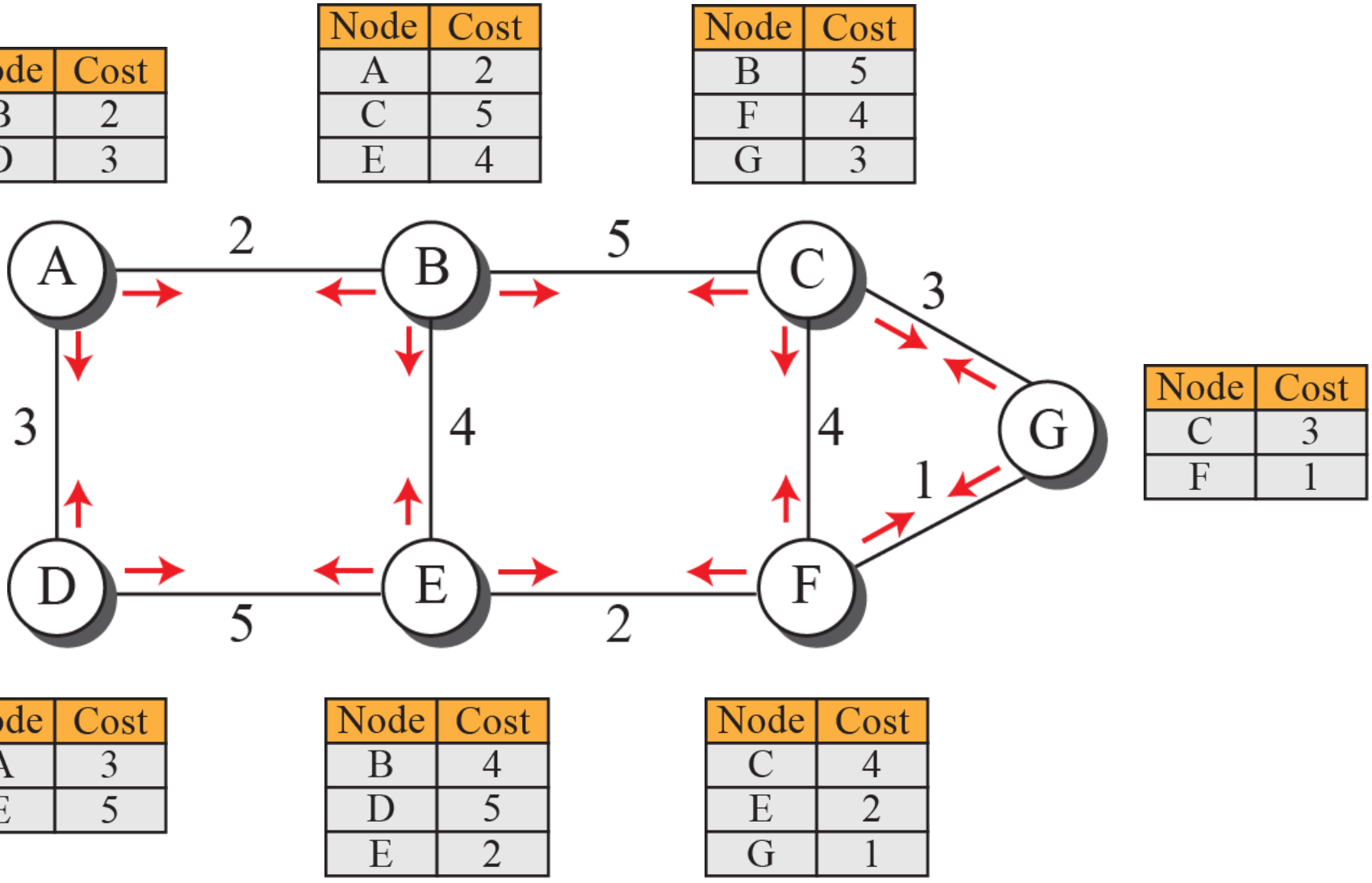


a. The weighted graph

	A	B	C	D	E	F	G
A	0	2	∞	3	∞	∞	∞
B	2	0	5	∞	4	∞	∞
C	∞	5	0	∞	∞	4	3
D	3	∞	∞	0	5	∞	∞
E	∞	4	∞	5	0	2	∞
F	∞	∞	4	∞	2	0	1
G	∞	∞	3	∞	∞	1	0

b. Link state database

LSPs created and sent out by each node to build LSDB



Route calculation

- Every node has the LSP from everybody else.
- So each node has the full topology of the network **(LSDB**)
- It uses the **Dijkstra's shortest path algorithm** to calculate the **best route** to every destination.
- Actually it creates a **shortest path tree** by keeping the source as the root. (by referring to LSDB)
- Also called as **Single Source Shortest Path** problem

Shortest Path Routing

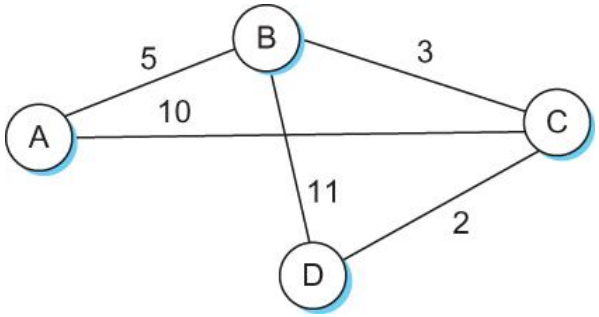
- Dijkstra's Algorithm - Assume non-negative link weights
  - N: set of nodes in the graph
  - l((i, j): the non-negative cost associated with the edge between nodes i, j $\in$N and l(i, j) = $\propto$ if no edge connects i and j
  - Let s $\in$N be the starting node which executes the algorithm to find shortest paths to all other nodes in N
  - Two variables used by the algorithm
    - M: set of nodes incorporated so far by the algorithm (processed nodes)
    - C(n) : the cost of the path from s to each node n
    - The algorithm

```
M = {s}
For each n in N – {s}
    C(n) = l(s, n)
while ( N ≠ M)
  M = M ∪ {w} such that C(w) is the minimum
                              for all w in (N-M)
  For each n in (N-M)
          C(n) = MIN (C(n), C(w) + l(w, n))
```

Route calculation

- Specifically each switch maintains two lists, known as **Tentative** and **Confirmed**
- Each of these lists contains a set of entries of the form **(Destination, Cost, NextHop)**
- **Algorithm:**
  - Initialize the **Confirmed** list with an entry for **source**; this entry has a cost of 0
  - For the node just added to the **Confirmed** list in the previous step, call it node **Next**, select its **LSP**
  - For each neighbor (Neighbor) of **Next**, calculate the cost (Cost) to reach this Neighbor as the sum of the cost from myself to Next and from Next to Neighbor
    - If Neighbor is currently on neither the **Confirmed** nor the **Tentative** list, then add (Neighbor, Cost, Nexthop) to the **Tentative** list, where Nexthop is the direction I go to reach Next
    - If Neighbor is currently on the **Tentative** list, and the Cost is less than the currently listed cost for the Neighbor, then replace the current entry with (Neighbor, Cost, Nexthop) where Nexthop is the direction I go to reach Next
  - If the **Tentative** list is empty, stop. Otherwise, pick the entry from the **Tentative** list with the lowest cost, move it to the **Confirmed** list, and return to Step 2.

Shortest Path Routing



Source: D

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

Comparison of LS and DV algorithms

message complexity

- ***LS:*** with n nodes, E links, O(nE) msgs sent
- ***DV:*** exchange between neighbors only
  - convergence time varies

speed of convergence

- ***LS:*** $O(n^2)$ algorithm requires O(nE) msgs
  - may have oscillations
- ***DV:*** convergence time varies
  - may be routing loops
  - count-to-infinity problem

robustness: what happens if router malfunctions?

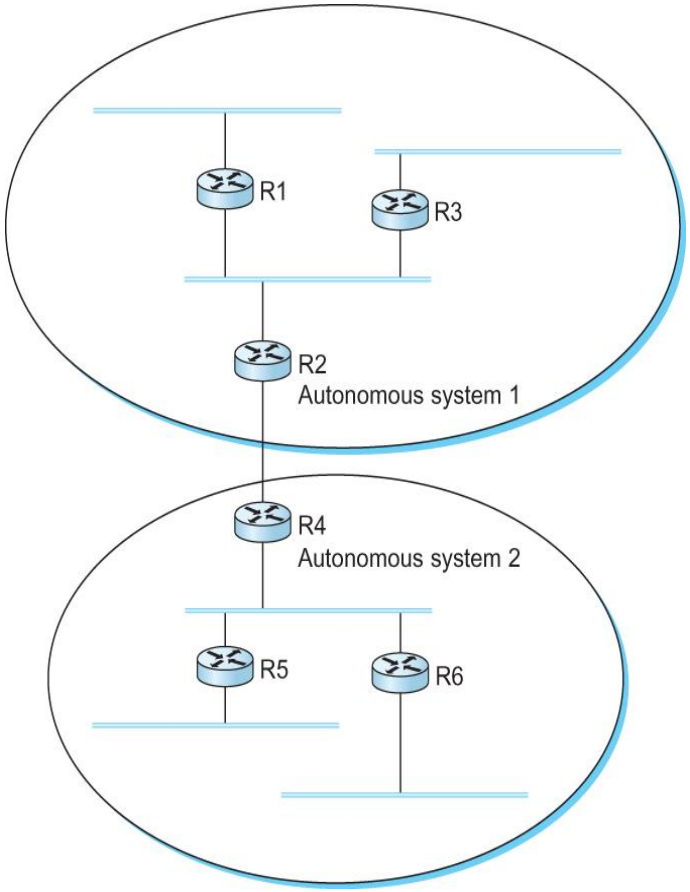
LS:

- node can advertise incorrect *link* cost
- each node computes only its *own* table

DV:

- DV node can advertise incorrect *path* cost
- each node's table used by others
  - error propagate thru network

Interdomain Routing



A network with two autonomous system

Divide the routing problem in two parts:

- Routing within a single autonomous system
- Routing between autonomous systems

Another name for autonomous systems in the Internet is **routing domains**

- Two-level route propagation hierarchy
  - Inter-domain routing protocol (Internet-wide standard)
  - Intra-domain routing protocol (each AS selects its own)

Hierarchical routing

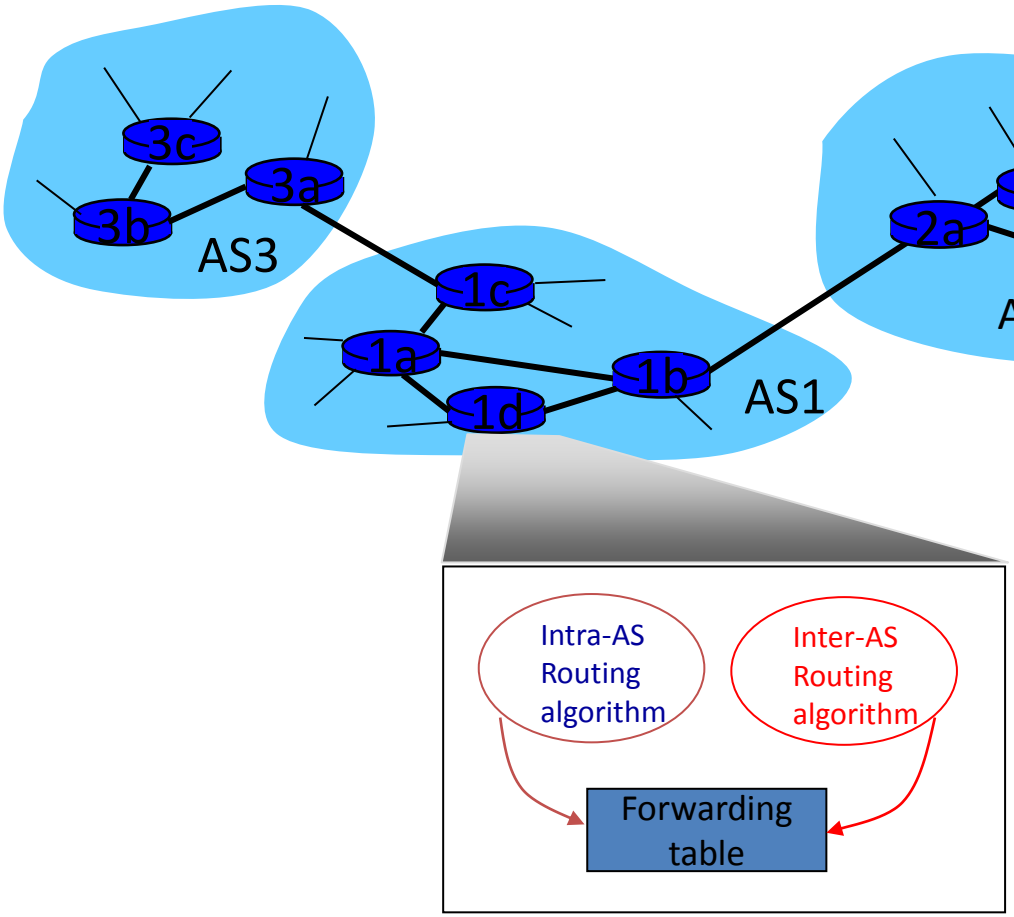
administrative autonomy

- Internet is organized as autonomous systems (AS) each of which is under the control of a single administrative entity
- collect routers into regions, "autonomous systems" (AS)
- Each AS within an ISP
  - ISP may consist of one or more Ass
- routers in same AS run same routing protocol
  - "intra-AS" routing protocol

gateway router:

- at "edge" of its own AS
- has link to router in another AS

Interconnected ASes



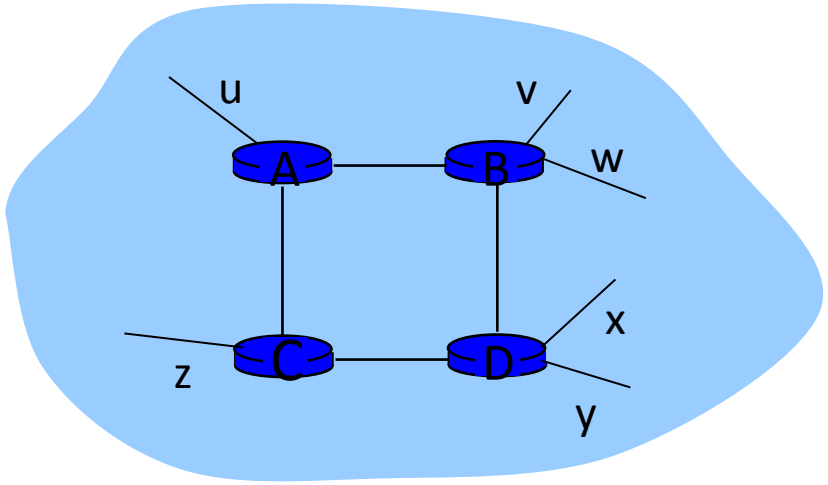
- forwarding table configured by both intra- and inter-AS routing algorithm
  - intra-AS sets entries for internal dests
  - inter-AS & intra-AS sets entries for external dests

Intra-AS Routing

- also known as *interior gateway protocols (IGP)*
- most common intra-AS routing protocols:
  - RIP: Routing Information Protocol
  - OSPF: Open Shortest Path First
  - IGRP: Interior Gateway Routing Protocol (Cisco proprietary)

RIP (Routing Information Protocol)

- included in BSD-UNIX distribution in 1982
- distance vector algorithm
  - distance metric: # hops (max = 15 hops), each link has cost 1
  - DVs exchanged with neighbors every 30 sec in response message (aka advertisement)
  - each advertisement: list of up to 25 destination *subnets (in IP addressing sense)*



from router A to destination *subnets:*

subnet	hops
u	1
v	2
w	2
x	3
y	3
z	2

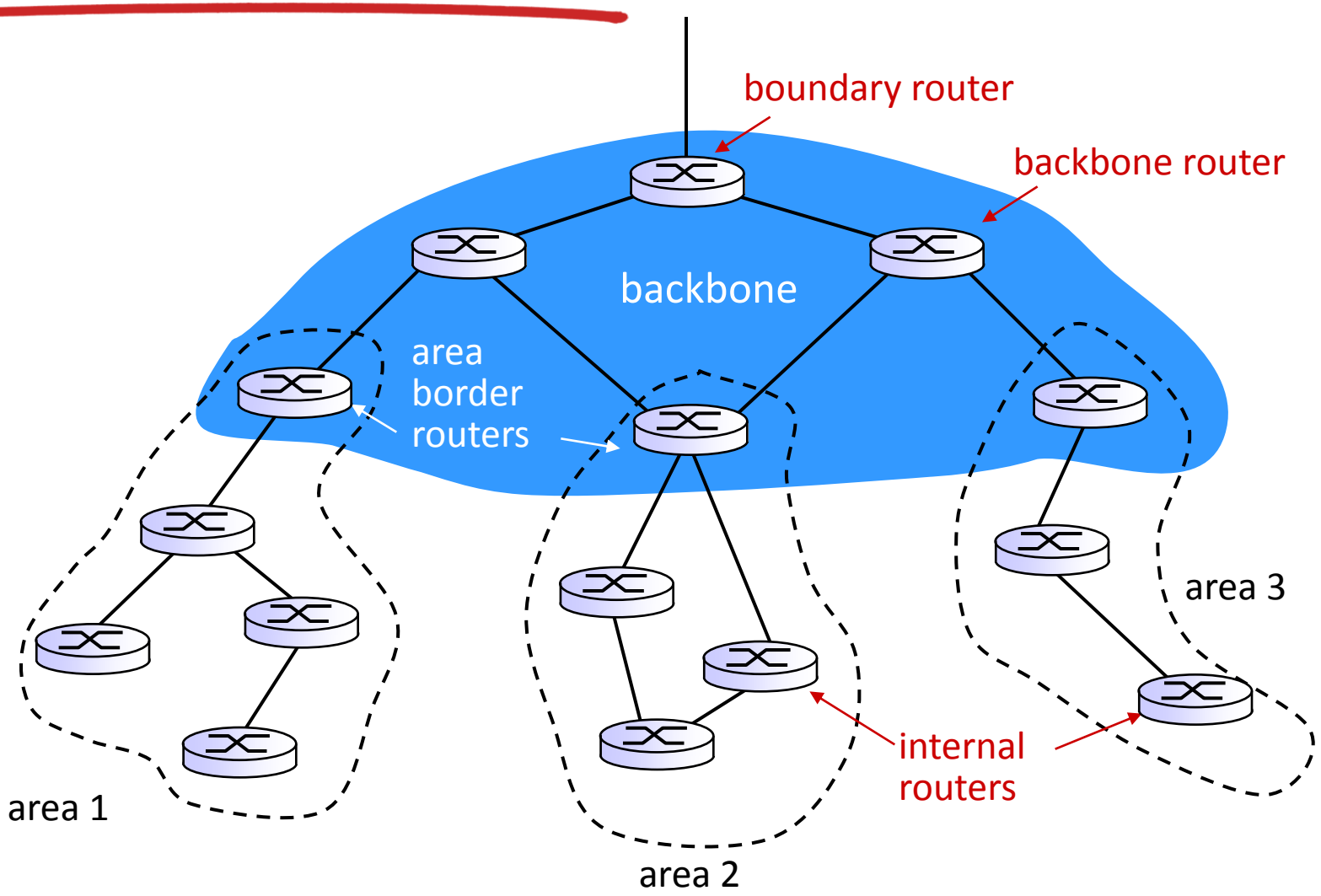
OSPF (Open Shortest Path First)

- "open": publicly available
- uses link state algorithm
  - LS packet dissemination
  - topology map at each node
  - route computation using Dijkstra's algorithm
- OSPF advertisement carries one entry per neighbor
- advertisements flooded to *entire* AS
  - carried in OSPF messages directly over IP (rather than TCP or UDP)
- *IS-IS routing* protocol: nearly identical to OSPF

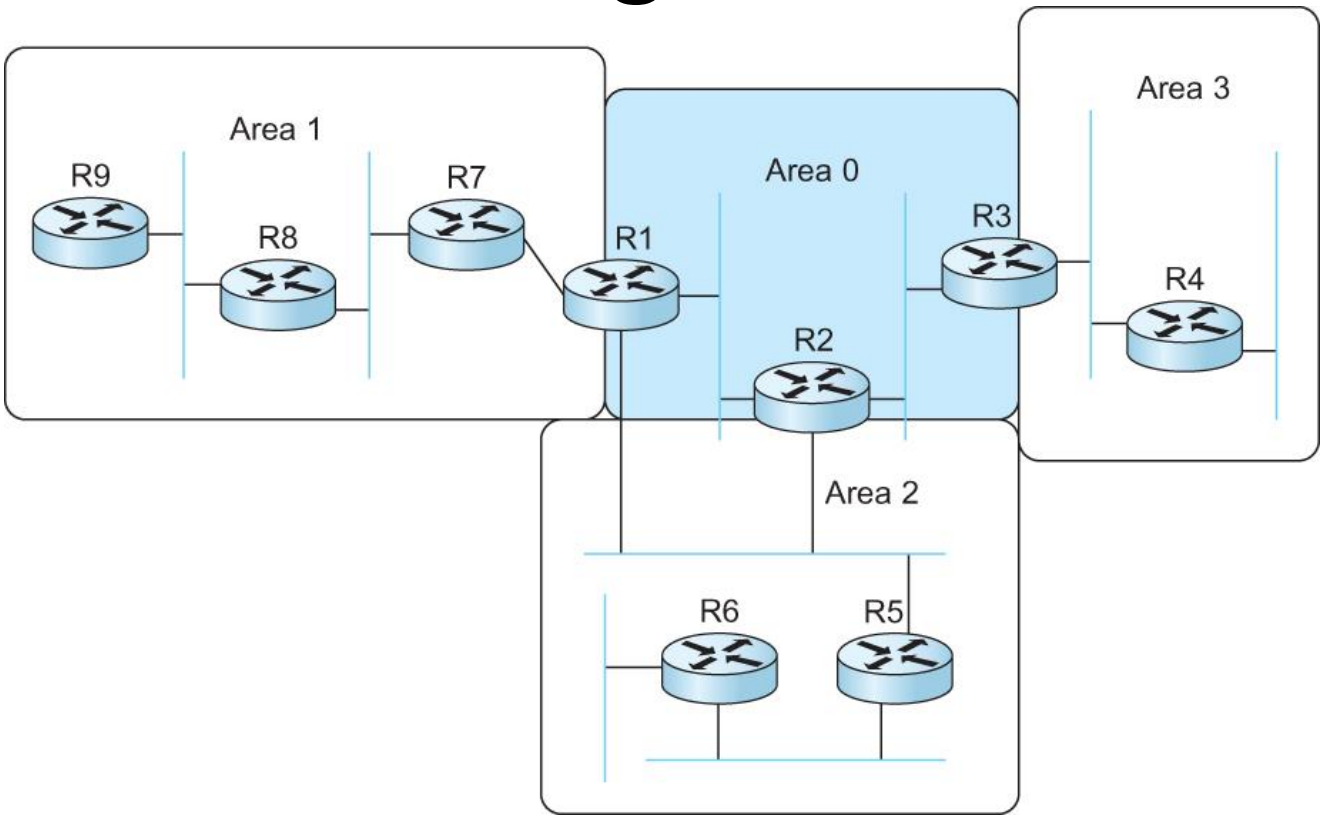
OSPF "advanced" features (not in RIP)

- *security:* all OSPF messages authenticated (to prevent malicious intrusion)
- multiple same-cost paths allowed (only one path in RIP)
- for each link, multiple cost metrics for different TOS (e.g., satellite link cost set "low" for best effort ToS; high for real time ToS)
- integrated uni- and multicast support:
  - Multicast OSPF (MOSPF) uses same topology data base as OSPF
- hierarchical OSPF in large domains.

Hierarchical OSPF



Routing Areas



A domain divided into area

Backbone area

Area border router (ABR)

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.
- *boundary routers:* connect to other AS's.

Internet inter-AS routing: BGP

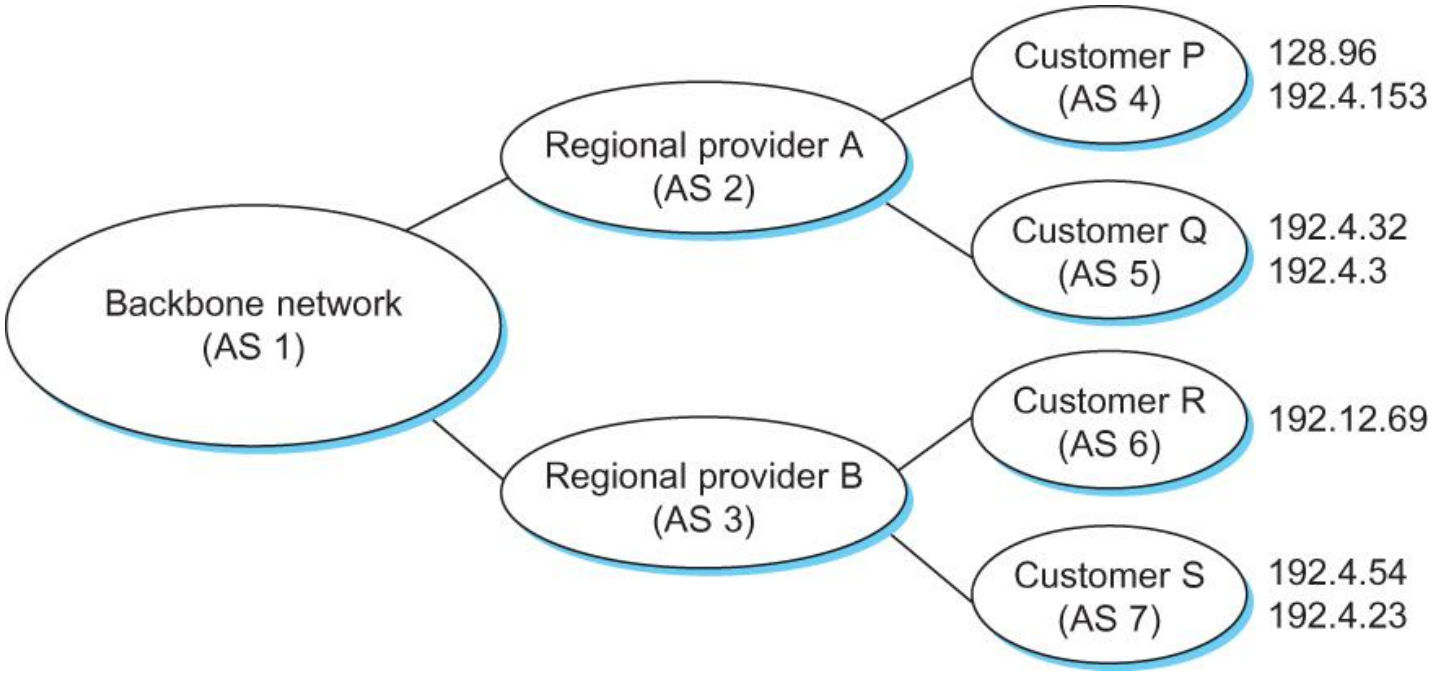
- BGP (Border Gateway Protocol): *the* de facto inter-domain routing protocol
- The goal of Inter-domain routing is to find **some loop free** path to the intended destination
- We are concerned with **reachability than optimality**
- BGP provides each AS a means to:
  - obtain subnet reachability information from neighboring AS's: eBGP
  - propagate reachability information to all AS-internal routers: iBGP
  - determine "good" routes to other networks based on reachability information and policy.

BGP

Each AS has:

- One BGP *speaker* that advertises:
  - local networks
  - other reachable networks
  - gives *path* information
- In addition to the BGP speakers, the AS has one or more **border "gateways"** which need not be the same as the speakers
- The border gateways are the routers through which packets enter and leave the AS
- BGP advertises *complete paths* as an enumerated lists of ASs to reach a particular network

BGP Example



Example of a network running BGP

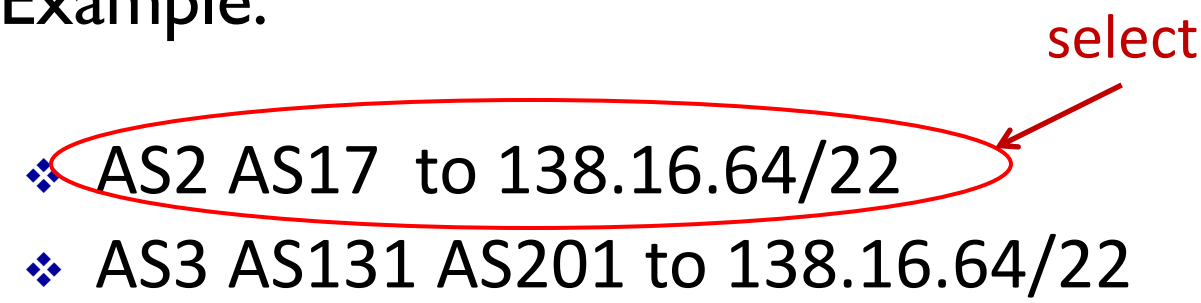
BGP Example

- Speaker for AS 2 advertises reachability to P and Q
  - Network 128.96, 192.4.153, 192.4.32, and 192.4.3, can be reached directly from AS 2.
- Speaker for backbone network then advertises
  - Networks 128.96, 192.4.153, 192.4.32, and 192.4.3 can be reached along the path <AS 1, AS 2>.
- Speaker can also cancel previously advertised paths

Select best BGP route to prefix

- Router may receive multiple routes for same prefix
- Has to select one route
- Router selects route based on shortest AS-PATH

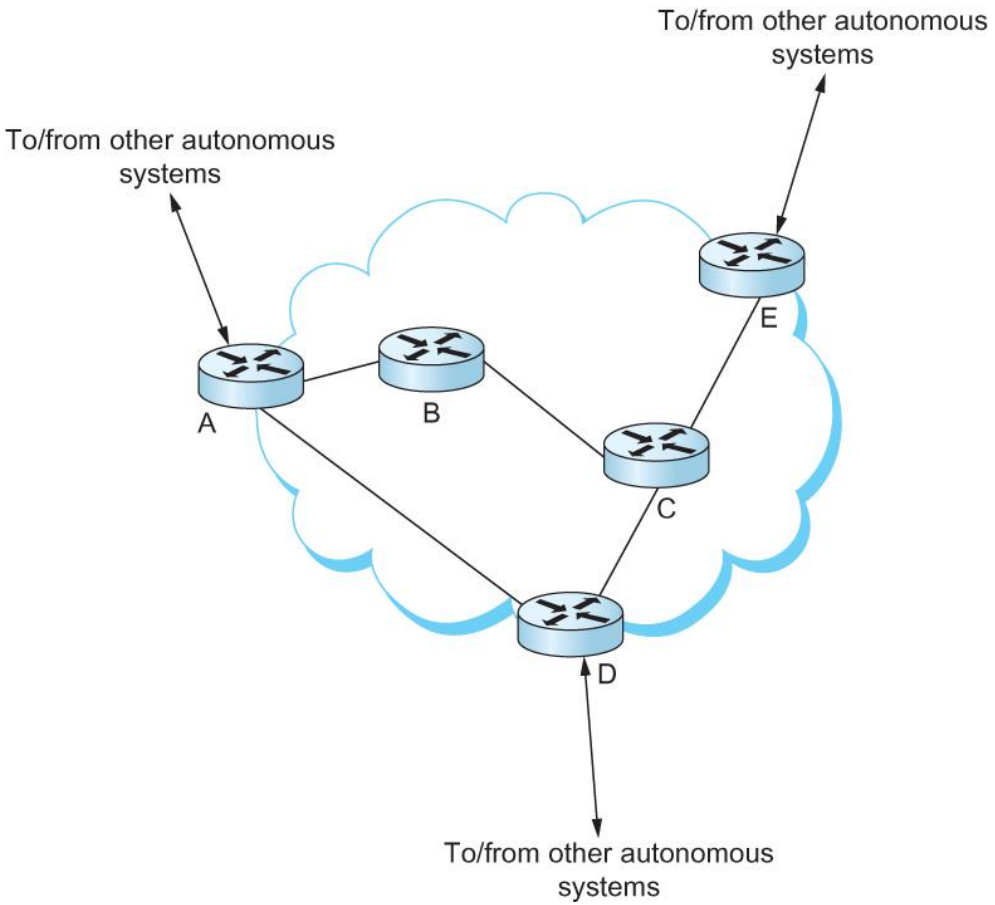
- Example:



BGP Issues

- It should be apparent that the AS numbers carried in BGP need to be unique
- For example, AS 2 can only recognize itself in the AS path in the example if no other AS identifies itself in the same way
- AS numbers are 16-bit numbers assigned by a central authority

Integrating Interdomain and Intradomain Routing



All routers run iBGP and an intradomain routing protocol. Border routers (A, D, E) also run eBGP to other ASs

Integrating Interdomain and Intradomain Routing

1. Router becomes aware of prefix
   - via BGP route advertisements from other routers
2. Determine router output port for prefix
   - Use BGP route selection to find best inter-AS route
   - Use OSPF to find best intra-AS route leading to best inter-AS route
   - Router identifies router port for that best route
3. Enter prefix-port entry in forwarding table

Integrating Interdomain and Intradomain Routing

Prefix	BGP Next Hop
18.0/16	E
12.5.5/24	A
128.34/16	D
128.69./16	A

BGP table for the AS

Router	IGP Path
A	A
C	C
D	C
E	C

IGP table for router B

Prefix	IGP Path
18.0/16	C
12.5.5/24	A
128.34/16	C
128.69./16	A

Combined table for router B

BGP routing table, IGP routing table, and combined table at router B

Why different Intra-, Inter-AS routing ?

policy:

- inter-AS: admin wants control over how its traffic routed, who routes through its net.
- intra-AS: single admin, so no policy decisions needed

scale:

- hierarchical routing saves table size, reduced update traffic

performance:

- intra-AS: can focus on performance
- inter-AS: policy may dominate over performance

Broadcast and Multicast routing

- Communication in the Internet today is not only unicasting;
- multicasting communication is growing fast. Here we first discuss the general ideas behind unicasting(already discussed) , multicasting, and broadcasting.

Figure 4.87: Unicasting

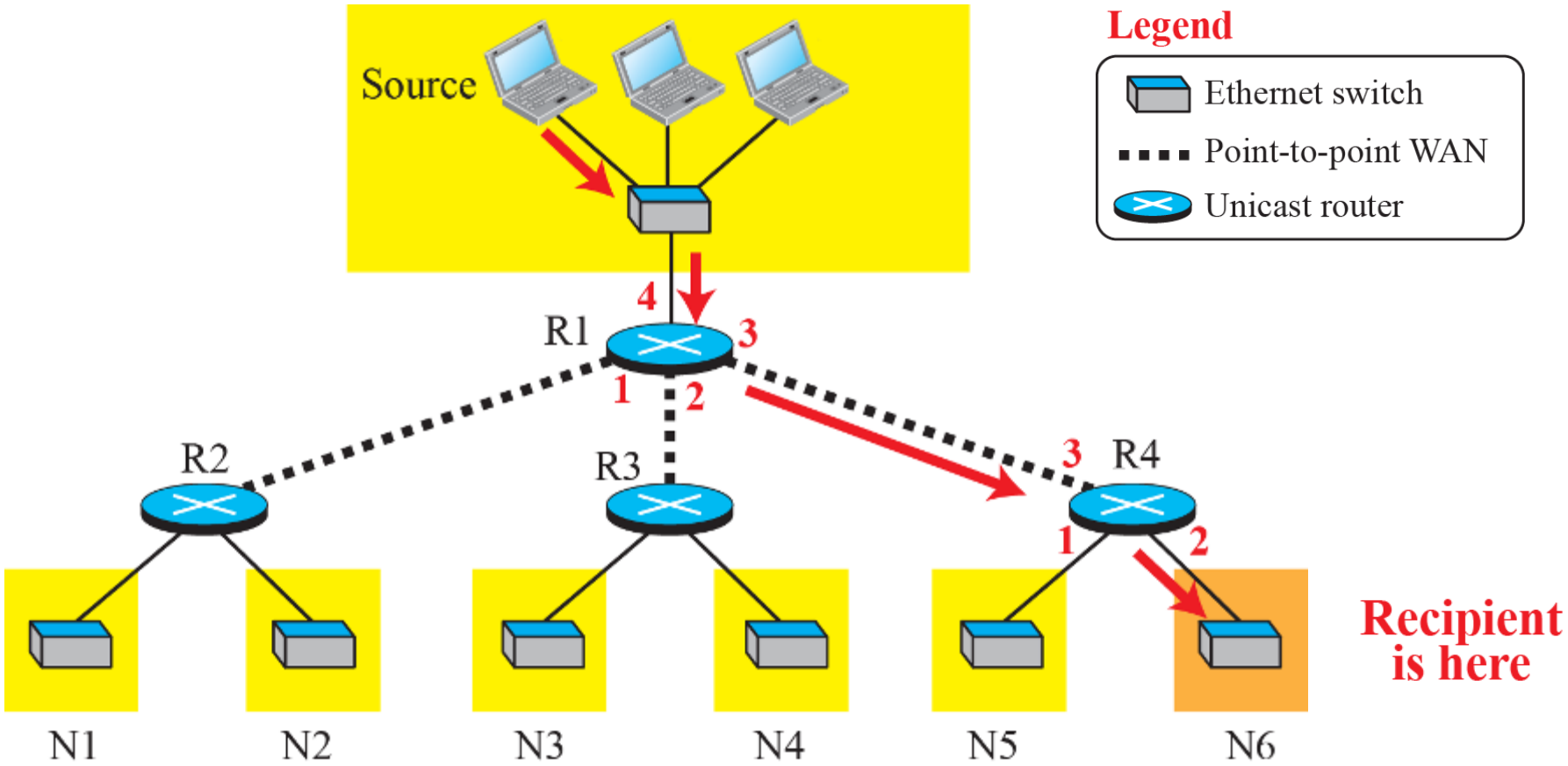
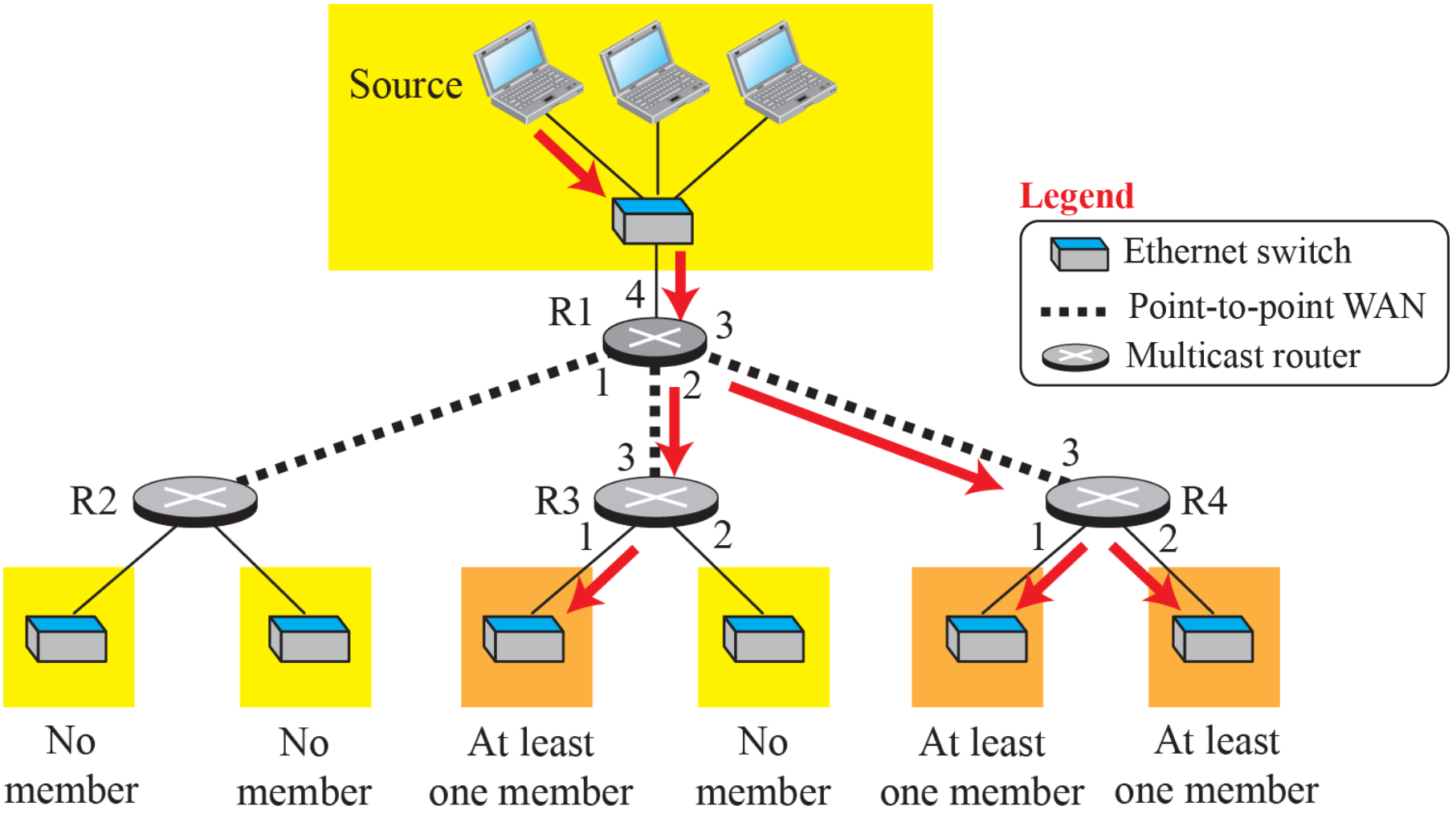


Figure 4.88: Multicasting



20.21.2 Multicasting

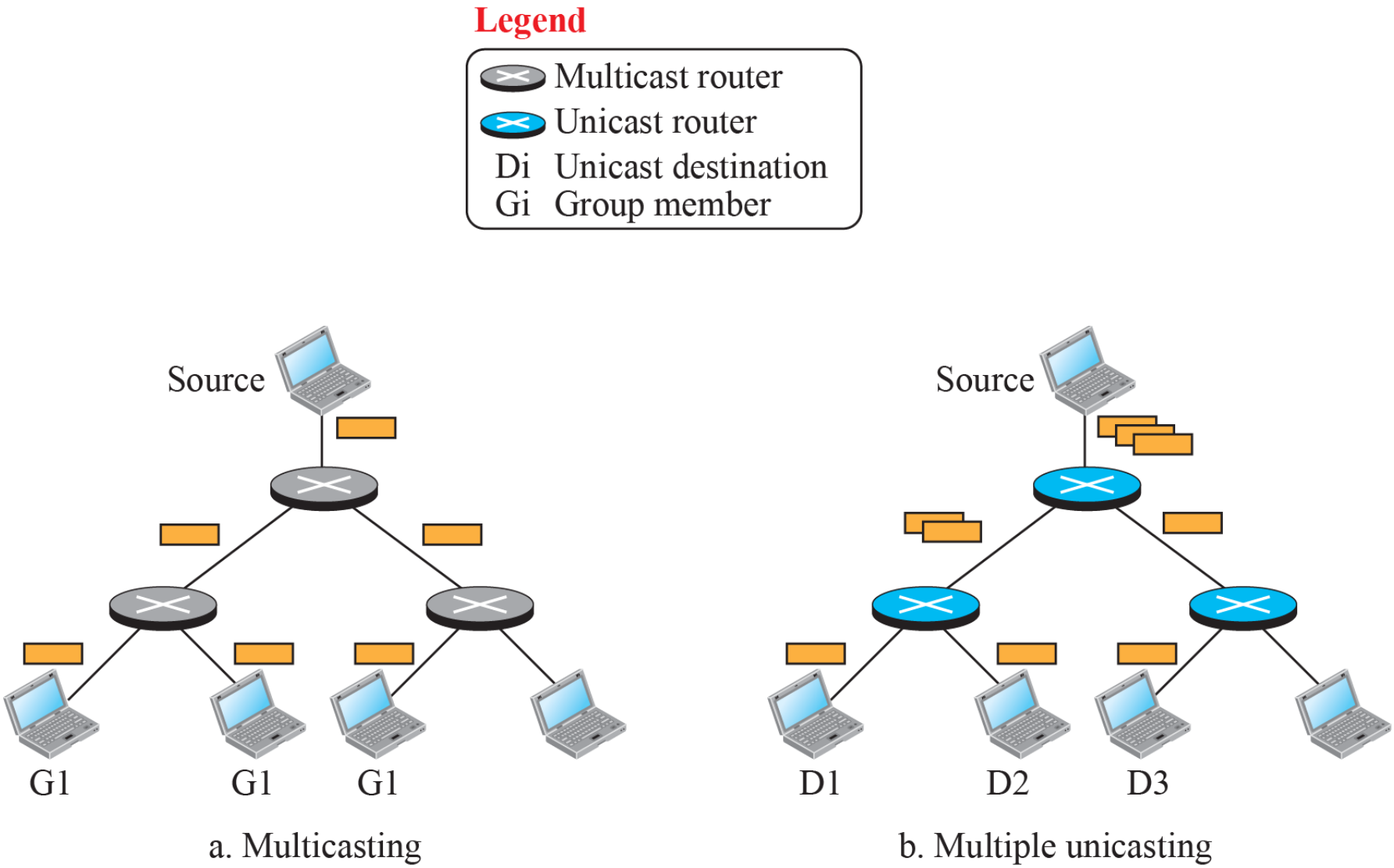
In multicasting, there is one source and a group of destinations.

The relationship is **one to many.**

In this type of communication, **the source address is a unicast address, but the destination address is a group address**, a group of one or more destination networks in which there is at least one member of the group that is interested in receiving the multicast datagram.

The group address defines the members of the group.

Figure 221.3: Multicasting versus multiple unicasting



Multicasting versus multiple unicasting

- **Multicasting**: starts with single packet and gets duplicated at routers whenever necessary.
- **Multiple Unicasting**: starts with multiple packets and gets duplicated at routers whenever necessary. – requires more bandwidth and also the delay between the first packet and the last packet is significant.

20.21.3 Broadcasting

Broadcasting means **one-to-all communication:** a host sends a packet to all hosts in an internet. Broadcasting in this sense is not provided at the Internet level for the obvious reason that it may create a huge volume of traffic and use a huge amount of bandwidth. Partial broadcasting, however, is done in the Internet. For example, some peer-to-peer applications may use broadcasting to access all peers. **Controlled broadcasting** may also be done in a domain (area or autonomous system) mostly as a step to achieve multicasting.

Multicast Applications

- Access to distributed Databases
- Information Dissemination (eg: software update dissemination to valid customers)
- Teleconferencing
- Distance learning

21-1 MULTICAST BASICS

Before discussing multicast routing protocols in the Internet, we need to discuss some **multicasting basics: multicast addressing, collecting information about multicast groups, and multicast optimal trees.**

221.2.1 Multicast Addresses

In multicast communication, the sender is only one, but the receiver is many, sometimes thousands or millions spread all over the world. It should be clear that we cannot include the addresses of all recipients in the packet. The destination address of a packet, as described in the Internet Protocol (IP) should be only one. For this reason, we need multicast addresses. **A multicast address defines a group of recipients,** not a single one. In other words, a multicast address is an identifier for a group.

Figure 221.4: Needs for multicast addresses

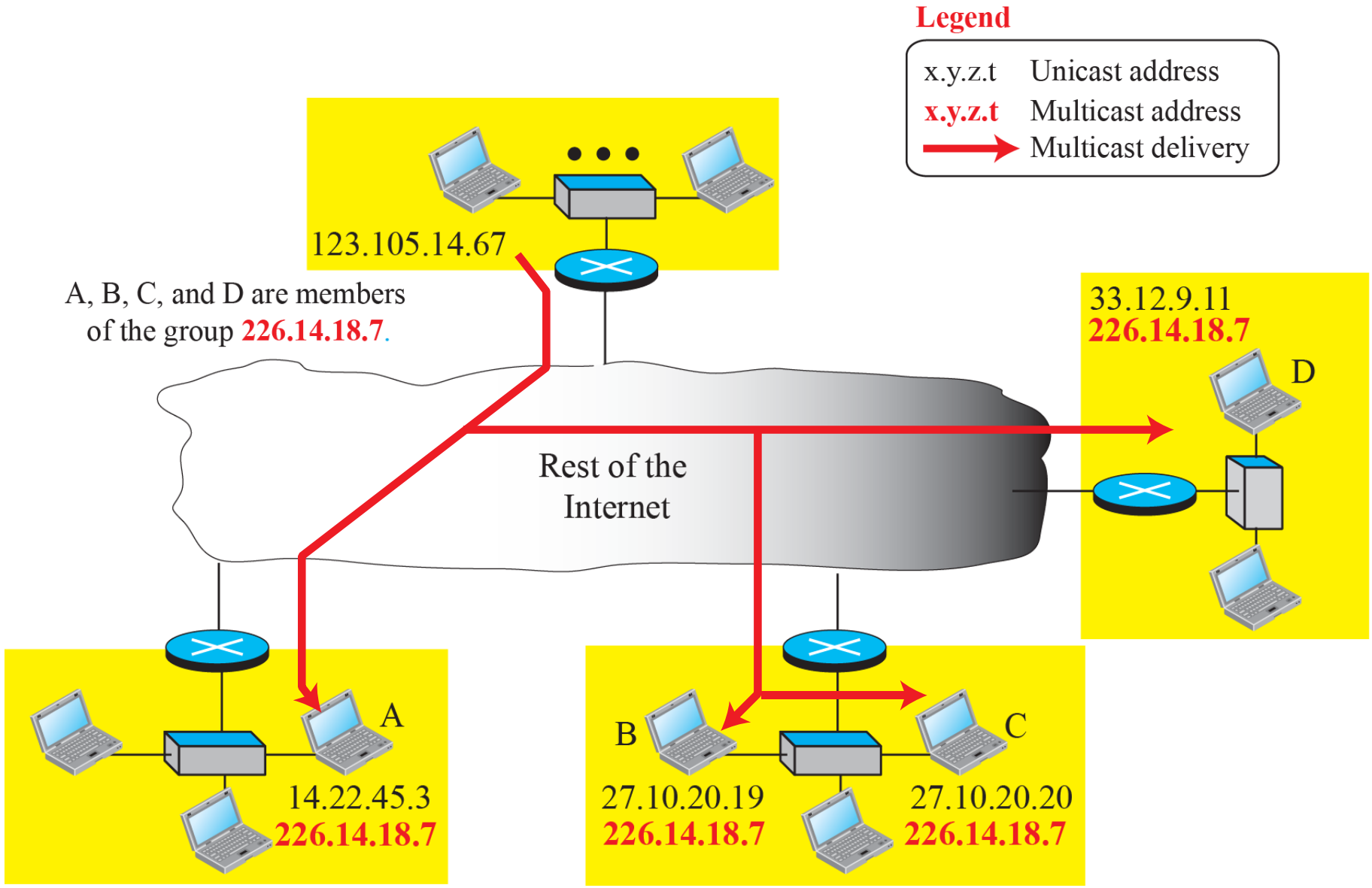
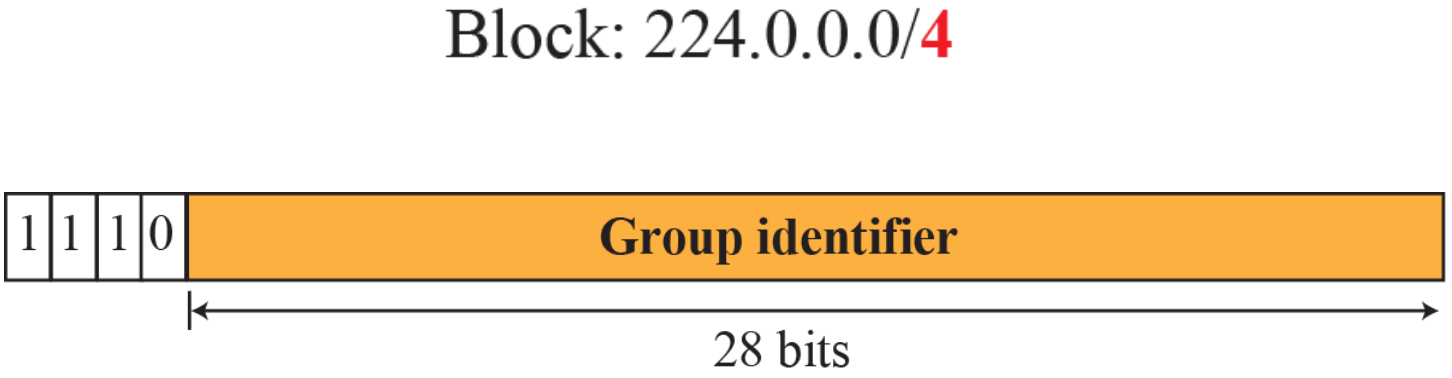


Figure 221.5: A multicast address in binary

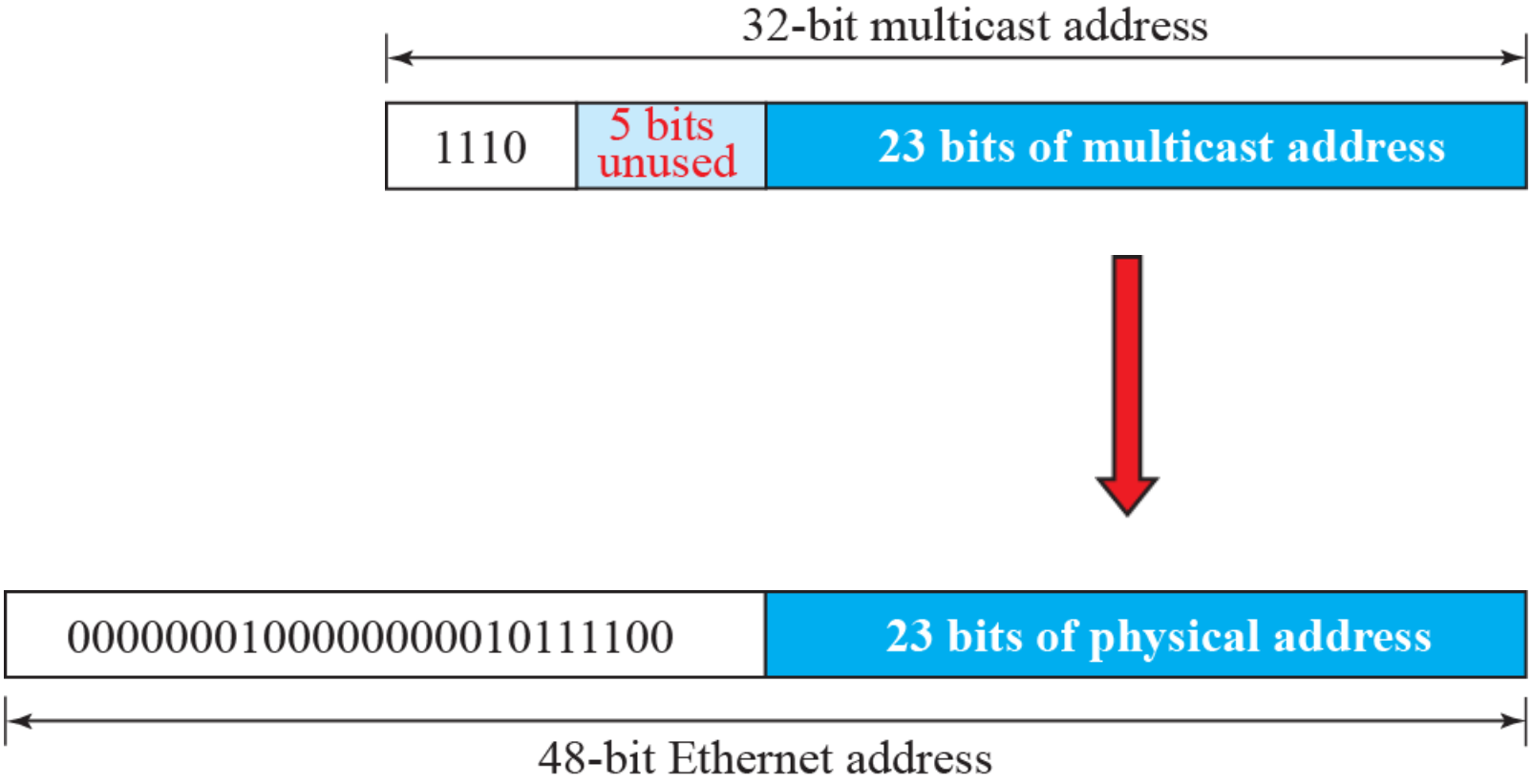


224.0.0.0 to 239.255.255.255 - Multicast Block (2^{28})

221.2.2 Delivery at Data-Link Layer

In multicasting, the delivery at the Internet level is done using network-layer multicast addresses. However, data-link layer multicast addresses are also needed to deliver a multicast packet encapsulated in a frame. In the case of unicasting, this task is done by the ARP protocol, but, because the IP packet has a multicast IP address, the **ARP protocol cannot find the corresponding MAC (physical) address to forward a multicast packet at the data-link layer.** It depends on whether or not the underlying data-link layer supports physical multicast addresses.

Figure 221.6: Mapping class D to Ethernet physical address



01 : 00:5E

Change the multicast IP address 232.43.14.7 to an Ethernet multicast physical address.

Solution

We can do this in two steps:

a. We write the rightmost 23 bits of the IP address in hexadecimal. In our example, the result is 2B:0E:07.

b. We add the result of part a to the starting Ethernet multicast address, which is 01:00:5E:00:00:00. The result is

01:00:5E:2B:0E:07

Example 221.2

Change the multicast IP address 238.212.24.9 to an Ethernet multicast address.

Solution

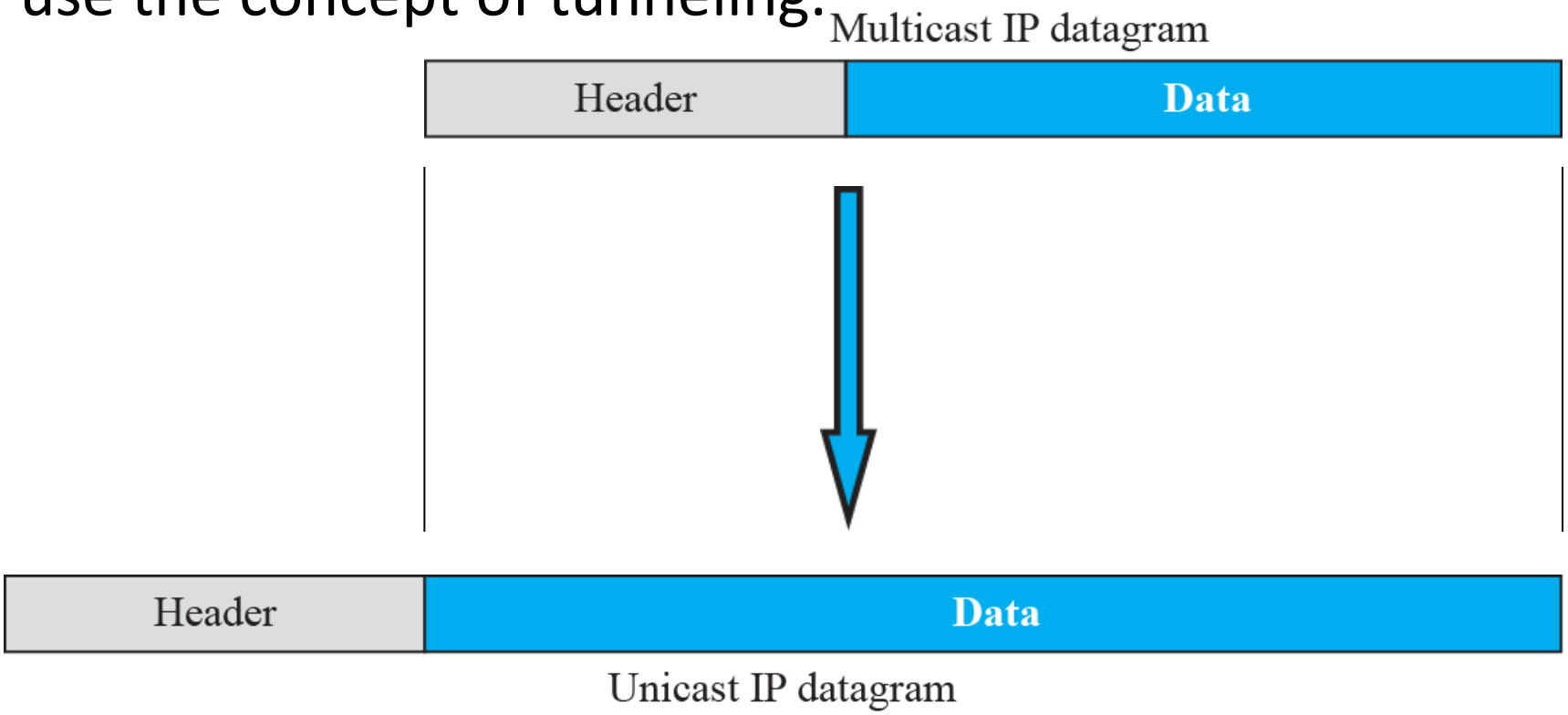
a. The rightmost 3 bytes in hexadecimal are D4:18:09. We need to subtract 8 from the leftmost digit, resulting in 54:18:09.

b. We add the result of part a to the Ethernet multicast starting address. The result is

01:00:5E:54:18:09

Figure 221.7: Tunneling

Most WANs does not support Multicasting. Then we use the concept of tunneling.



221.2.3 Collecting Information

Creation of forwarding tables in both unicast and multicast routing involves two steps:

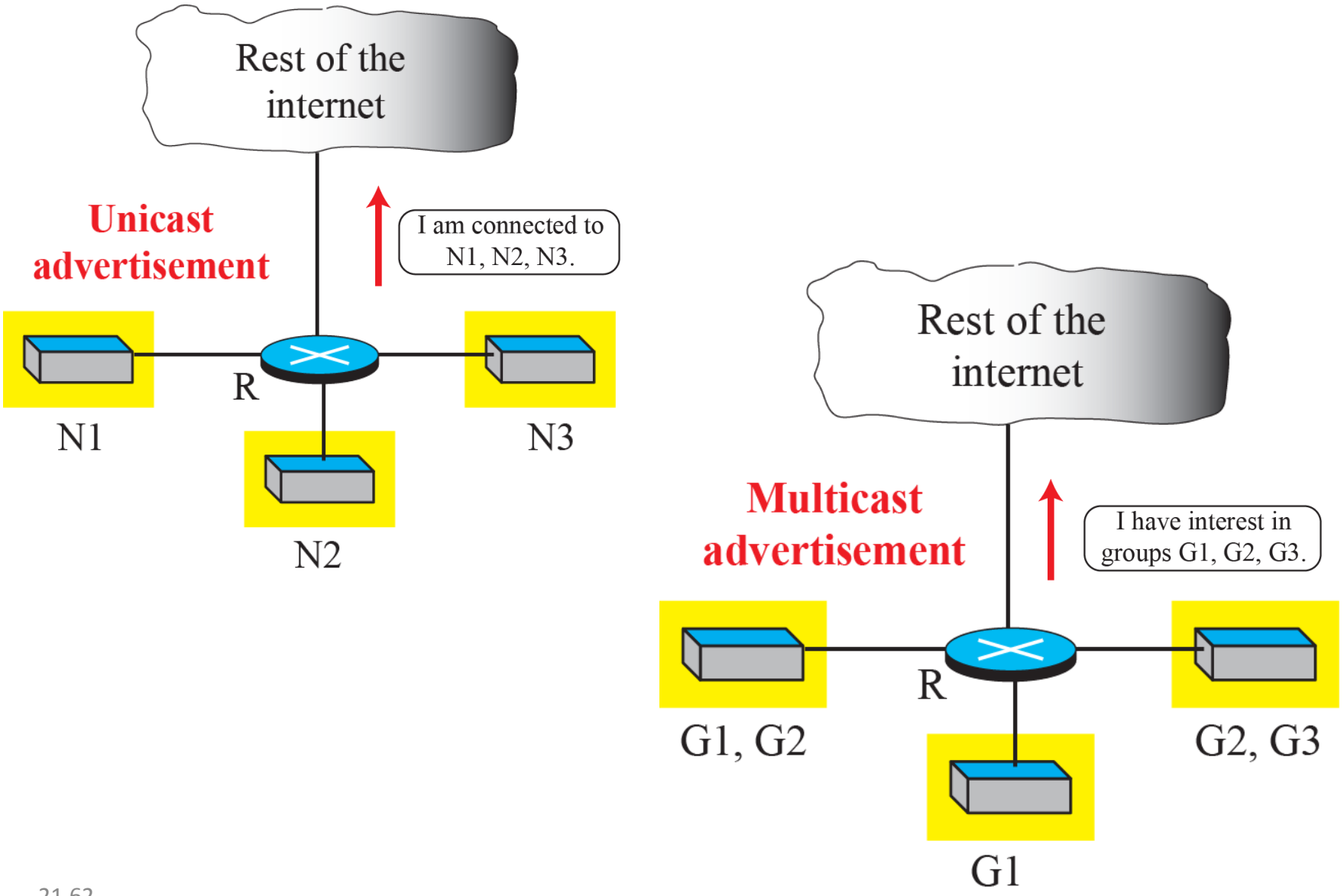
1. A router needs to know to which destinations it is connected. (network part)
2. Each router needs to propagate information obtained in the first step to all other routers so that each router knows to which destination each other router is connected. (routing protocol)

In **multicasting we can not do the same thing.**

What groups are active in a particular interface?

Group membership not permanent

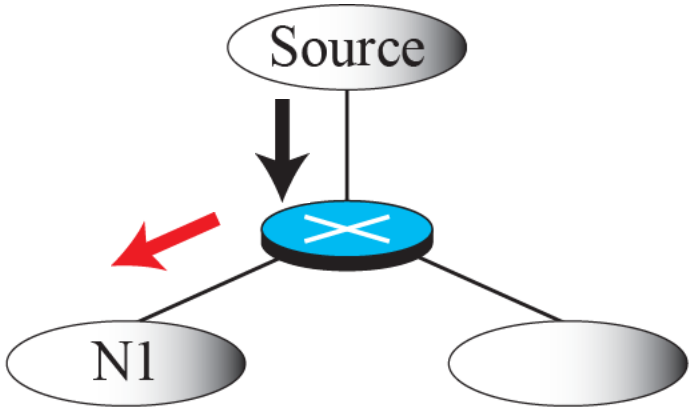
Figure 221.8: Unicast versus multicast advertisement



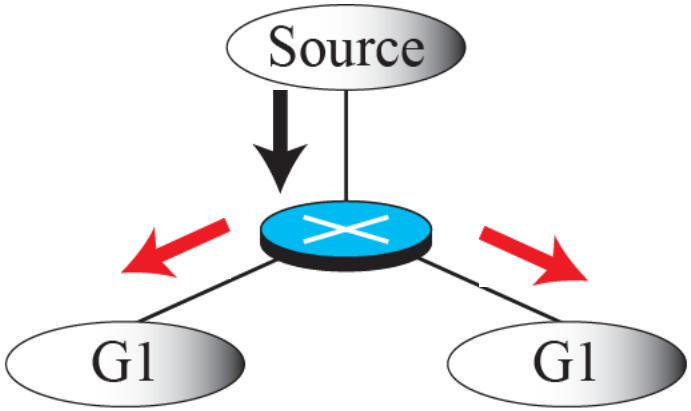
221.2.4 Multicast Forwarding

Another important issue in multicasting is the decision a router needs to make to forward a multicast packet. Forwarding in unicast and multicast communication is different.

Figure 221.9: Destination in unicasting and multicasting

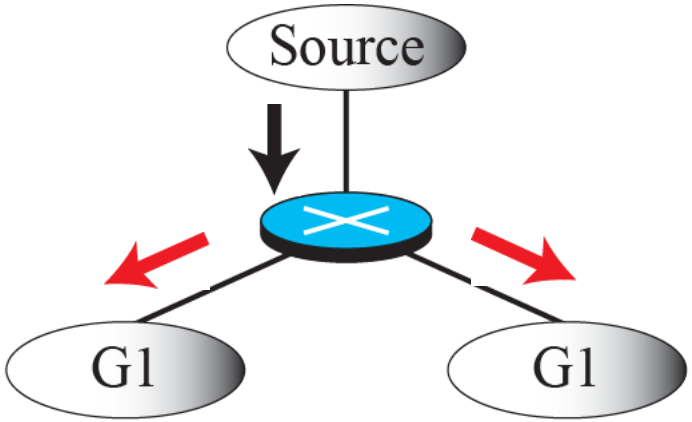


a. Destination in unicasting is one

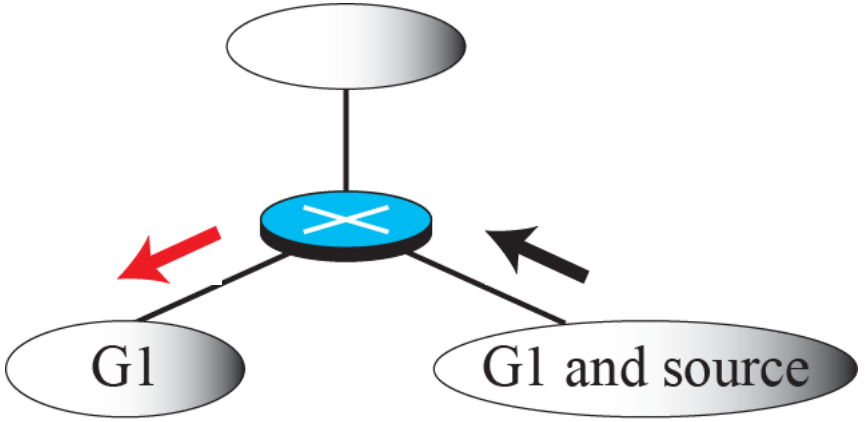


b. Destination in mulicasting is more than one

Figure 221.10: Forwarding depends on the destination and the source



a. Packet sent out of two interfaces



b. Packet sent out of one interface

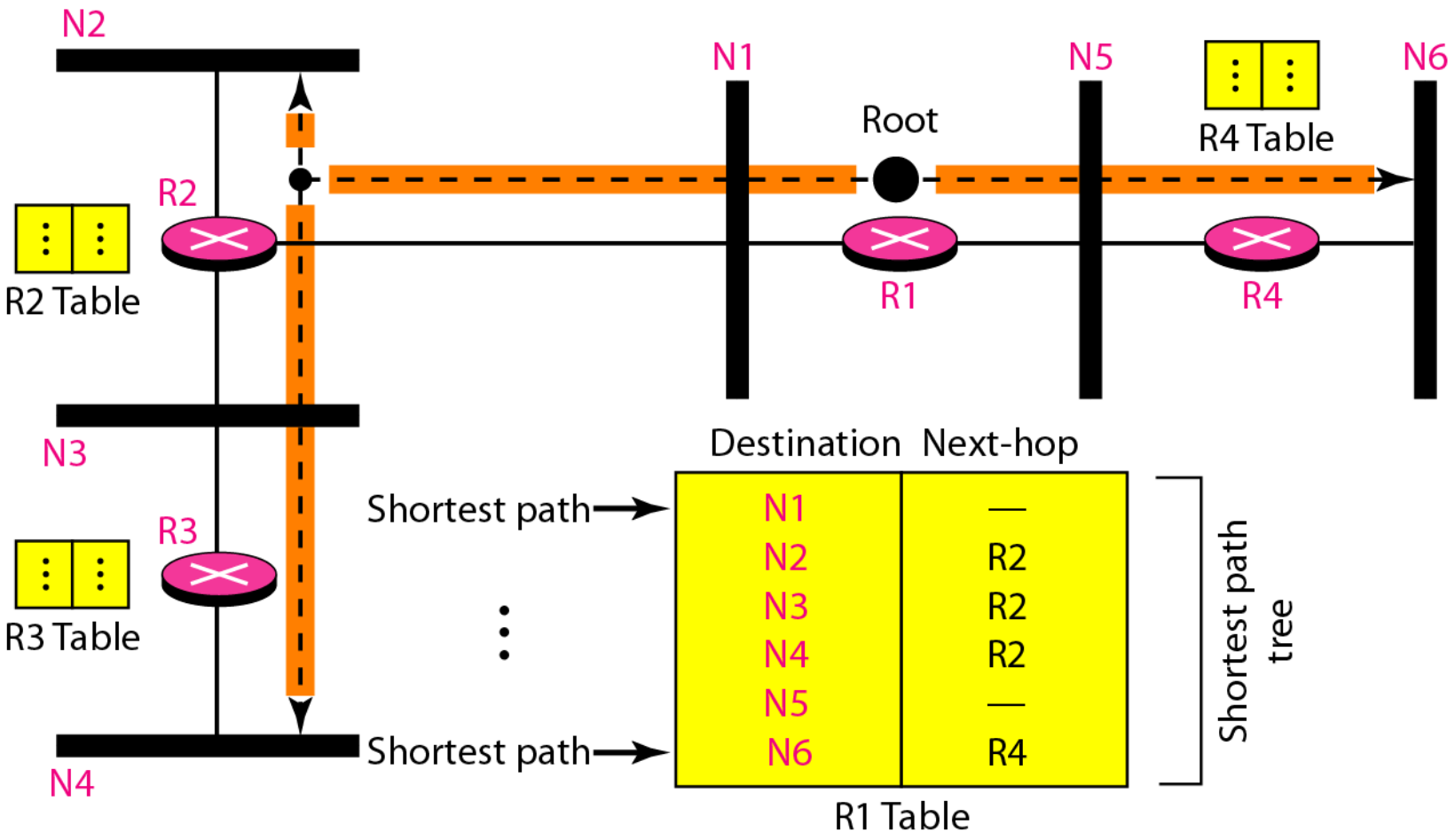
221.2.5 Two Approaches to Multicasting

In multicast routing, as in unicast routing, we need to create routing trees to optimally route the packets from their source to their destination. However, as we discussed before, the multicast routing decision at each router depends not only on the destination of the packet, but also on the source of the packet. The involvement of the source in the routing process makes **multicast routing much more difficult than unicast routing.** For this reason, **two different approaches** in multicast routing have been developed: routing using **source-based trees and routing using group-shared trees**.

Note

In unicast routing, each router in the domain has a table that defines a shortest path tree to possible destinations.

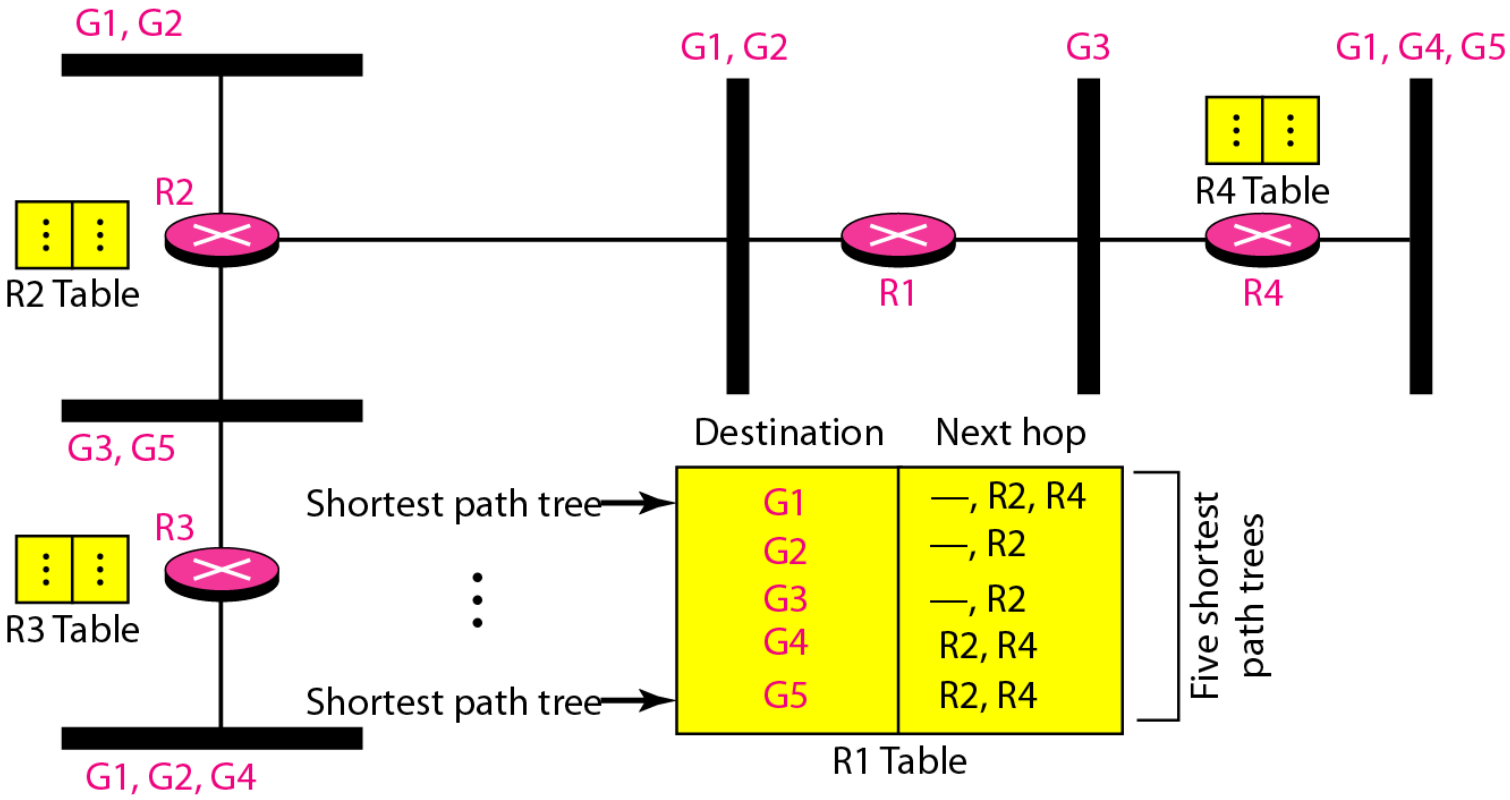
Figure 22.36 *Shortest path tree in unicast routing*



Note

In multicast routing, each involved router needs to construct a shortest path tree for each group.

Figure 22.37 *Source-based tree approach*



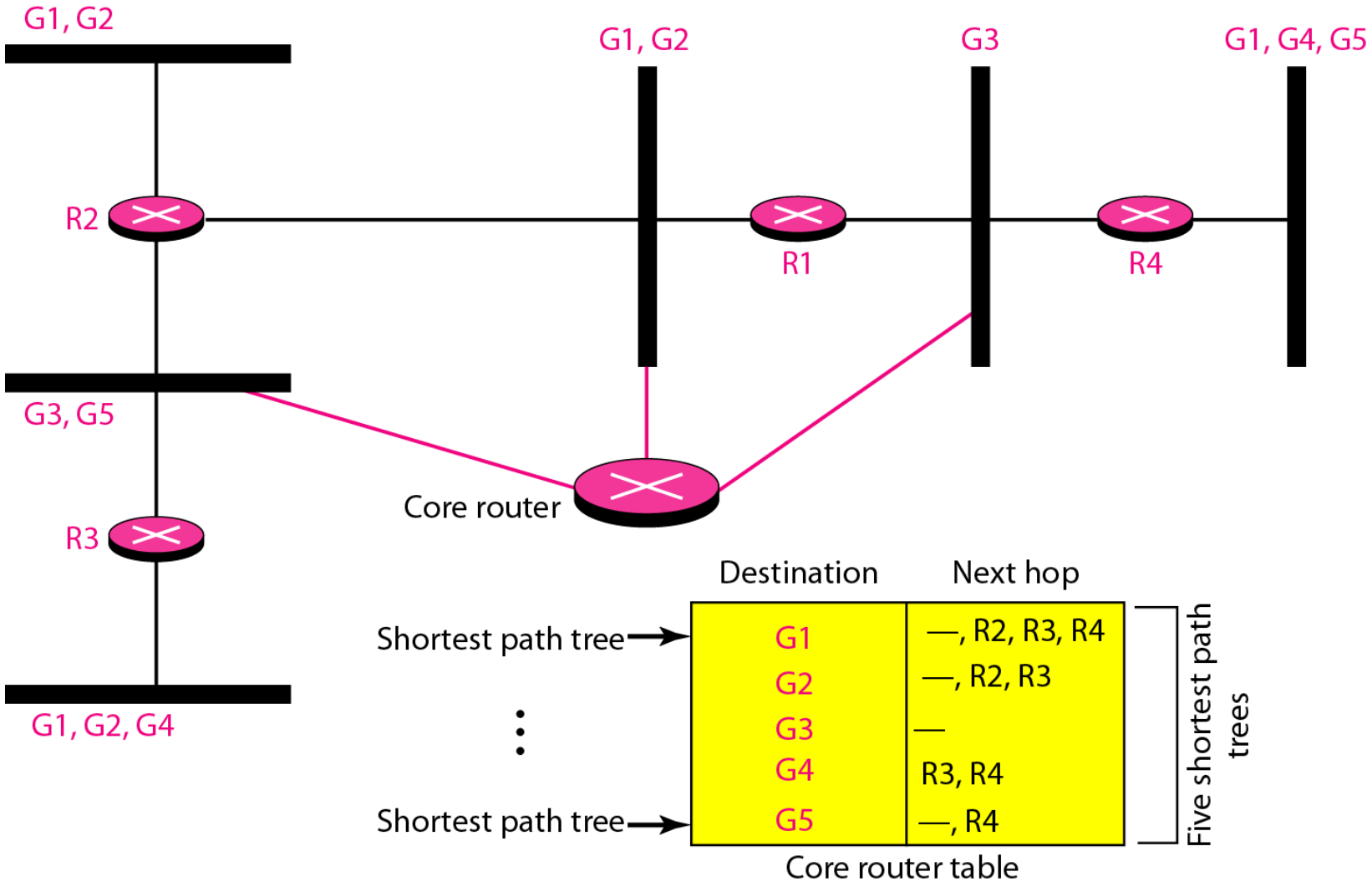
source-based trees

Note

In the source-based tree approach, each router needs to have one shortest path tree for each group.

If there are m groups and n sources in the internet, then the router need to **create m x n routing trees**. In each tree source is the root of the tree and the members are the leaves.

Figure 22.38 *Group-shared tree approach*



Group shared Trees

Note

In the group-shared tree approach, only the core router, which has a shortest path tree for each group, is involved in multicasting.

Here one router is designated as the **Core or Rendezvous router** and acts as the representative for the group. **Source unicast the packet to core (uses tunneling concept) and the core multicast it to the group members.** Core creates one tree for each group, so there are **m trees.**

21-3 INTRADOMAIN MULTICAST PROTOCOLS

During the last few decades, several intradomain multicast routing protocols have emerged.

In this section, we discuss **three of these protocols.** Two are extensions of unicast routing protocols (RIP and OSPF), using the source-based tree approach; the third is an independent protocol which is becoming more and more popular.

221.3.1 DVMRP

The Distance Vector Multicast Routing Protocol (DVMRP) is the extension of the Routing Information Protocol (RIP) which is used in unicast routing.

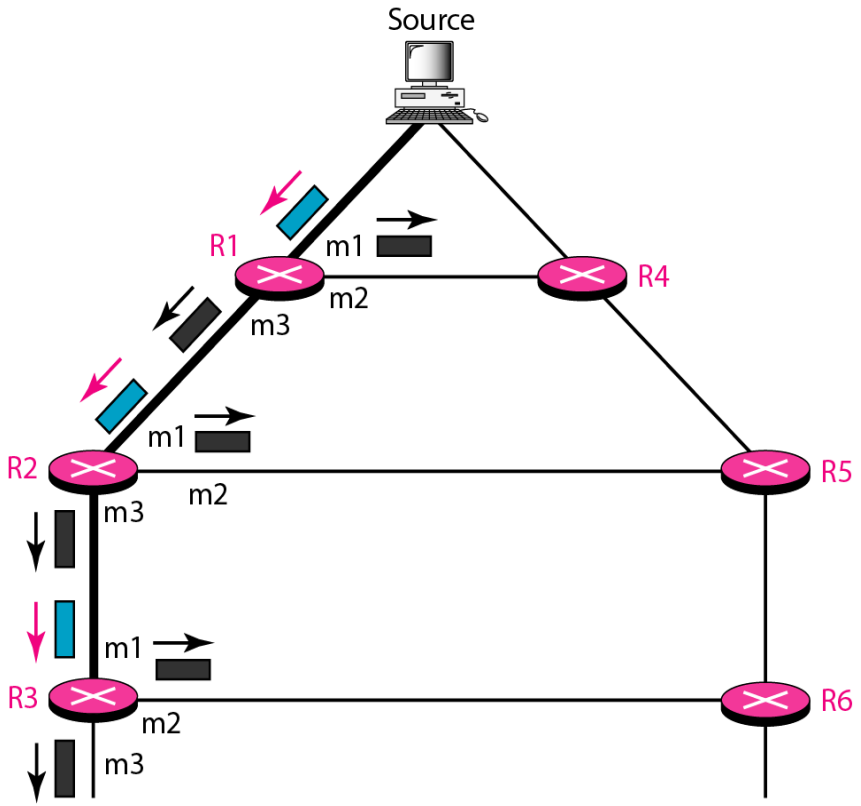
It uses the **source-based tree approach** to multicasting. It is worth mentioning that each router in this protocol that receives a multicast packet to be forwarded implicitly **creates a source-based multicast tree in three steps:**

DVMRP

1. The router uses an algorithm called **Reverse Path Forwarding (RPF)** to simulate creating part of the optimal source-based tree between the source and itself.
2. The router uses an algorithm called **Reverse Path Broadcasting (RPB)** to create a broadcast tree whose root is the router itself and whose leaves are all networks in the internet.
3. The router uses an algorithm called **Reverse Path Multicasting (RPM)** to create a multicast tree by cutting some branches of the tree that end in networks with no member in the group.

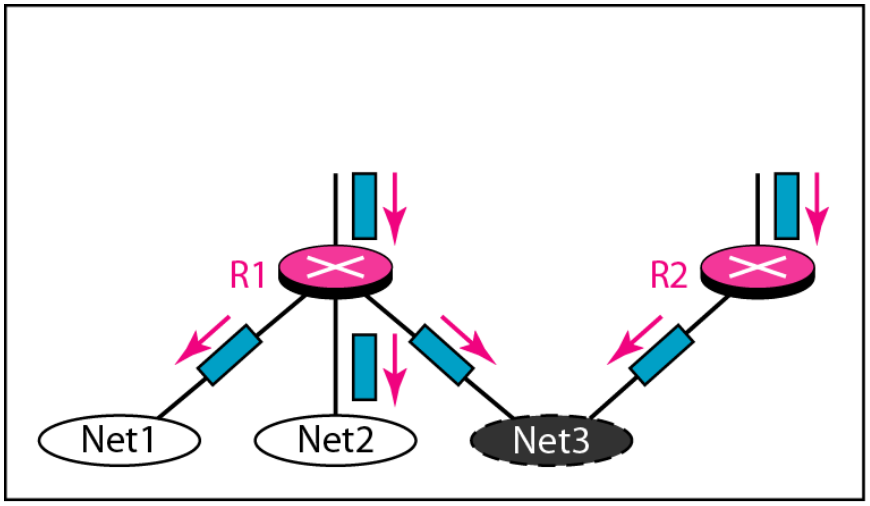
RPF

- Forces the router to forward a packet from one specific interface – the one which has come through the shortest path from the source to the router.

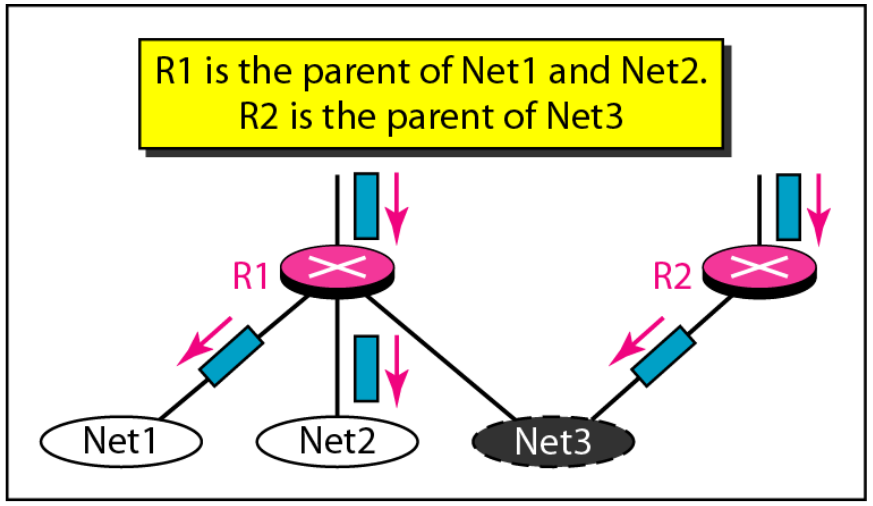


RPB

- If the network connected to two routers, it may receive two copies of same packet. To eliminate this **one router is designated as the parent for each network**.

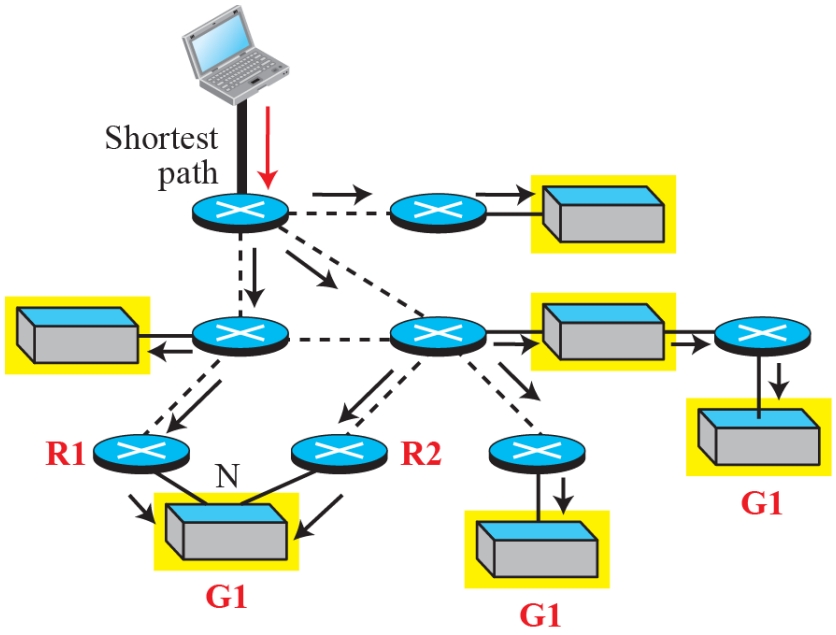


a. RPF

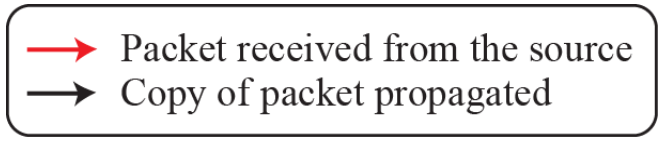


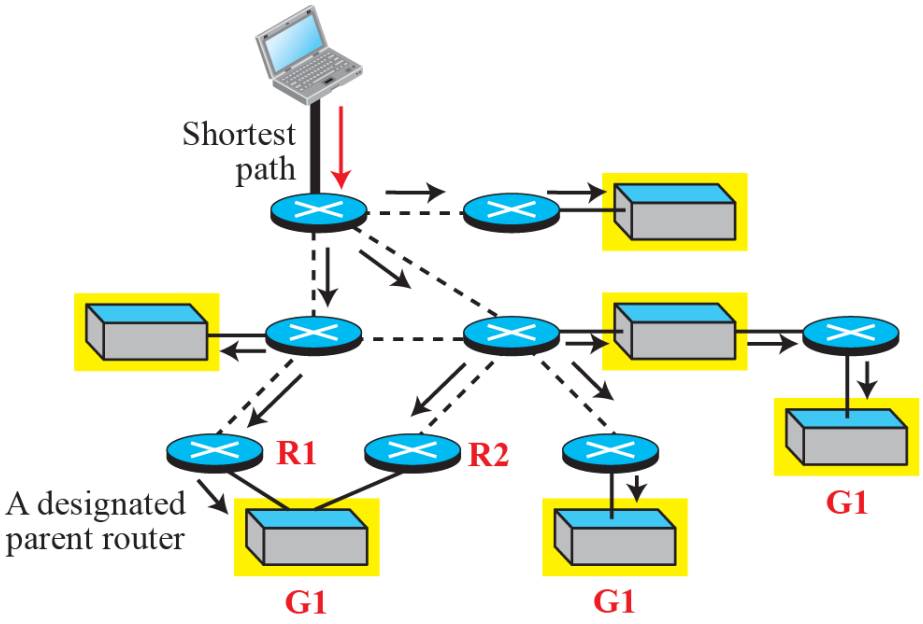
b. RPB

Figure 221.11: RPF versus RPB



a. Using RPF, N receives two copies.



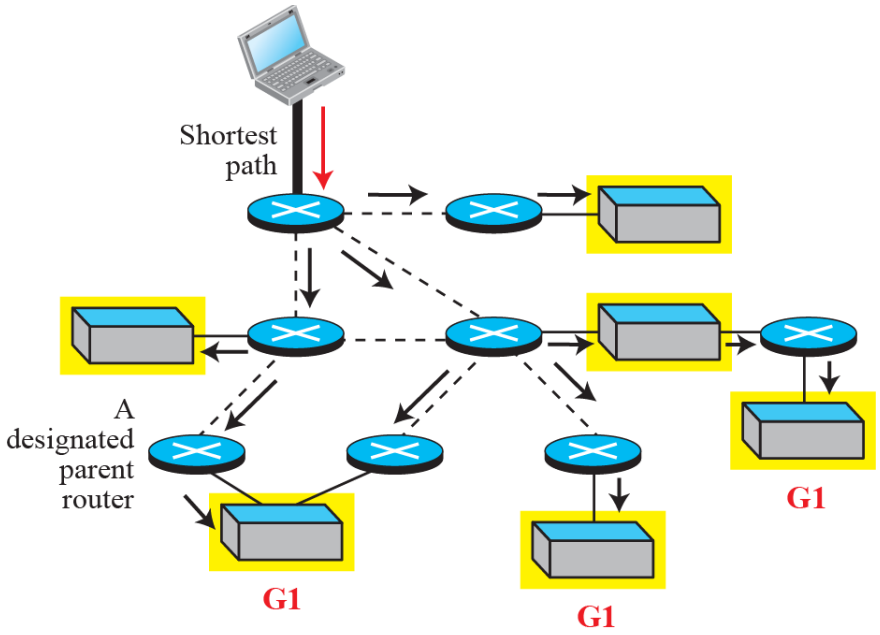


b. Using RPB, N receives only one copy.

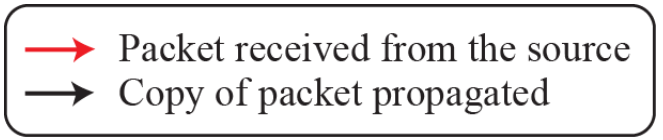
RPM

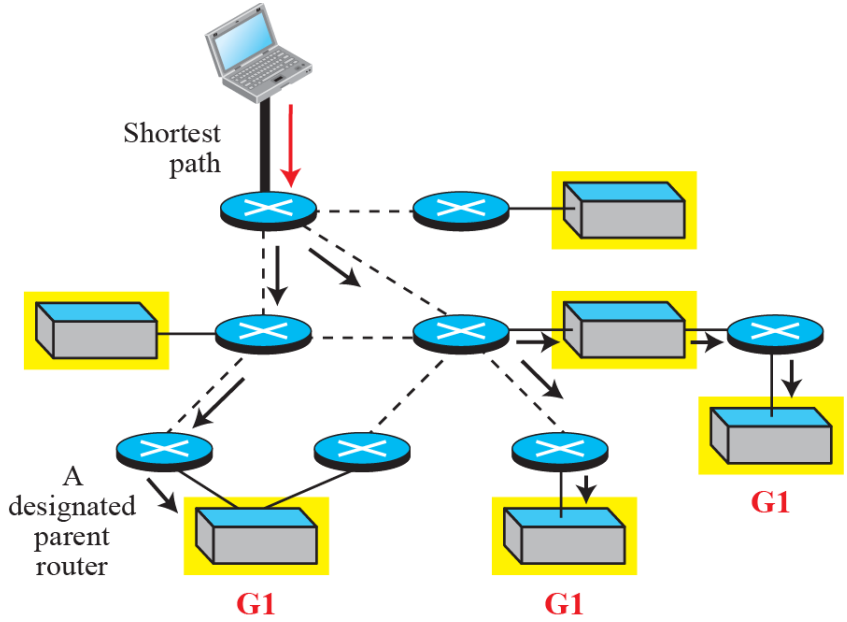
- RPB creates a shortest path broadcast tree from the source to each destination. **It guarantees that each destination receives one and only one copy of the packet.**
- RPM **adds pruning and grafting to RPB** to create a multicast shortest path tree that supports **dynamic membership changes**.
- Uses bottom up approach – join and leave the group

Figure 221.12: RPB versus RPM



a. Using RPB, all networks receive a copy.





b. Using RPM, only members receive a copy.

221.3.2 Multicast Link State (MOSPF)

Multicast Open Shortest Path First (MOSPF) is the extension of the Open Shortest Path First (OSPF) protocol, which is used in unicast routing. It also uses the **source-based tree approach** to multicasting.

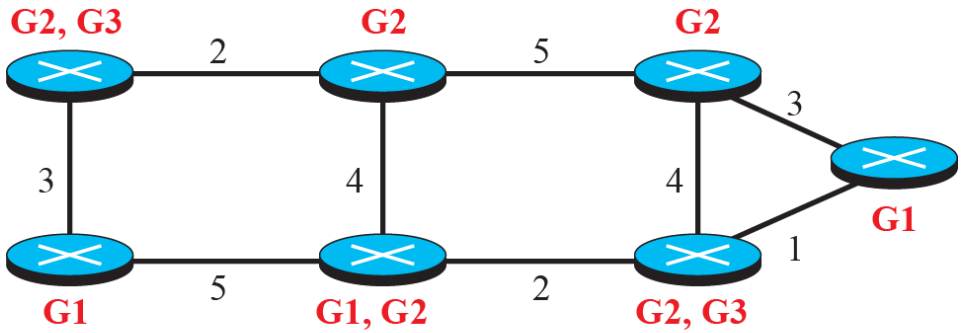
If the internet is running a unicast link-state routing algorithm, the idea can be extended to provide a multicast link-state routing algorithm.

Unicasting uses the LSDB database. To extend unicasting to multicasting, each router needs to **have another database**, as with the case of unicast distance-vector routing, **to show which interface has an active member in a particular group.**

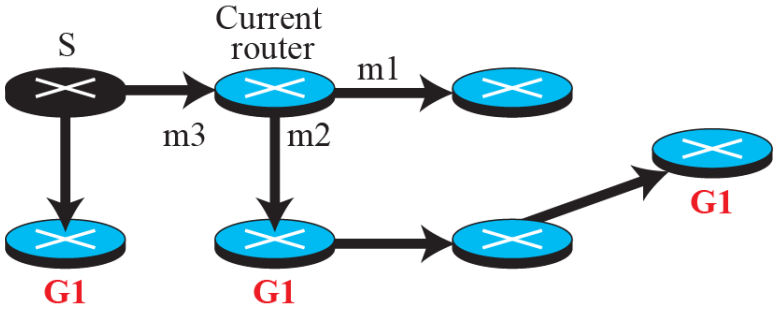
Multicast Link State (MOSPF)

- The router uses the Dijkstra's algorithm to create a shortest path tree **with S as the root** (not router as the root as in Unicast routing) and all destinations in the network as the leaves. (uses LSDB)
- The router creates a shortest path subtree with itself as the root of the subtree. Resultant tree is a broadcast tree.
- Create the **multicast tree** similar to DVMRP (using RPM). This multicast tree is used by every router to forward the packet to the group members.

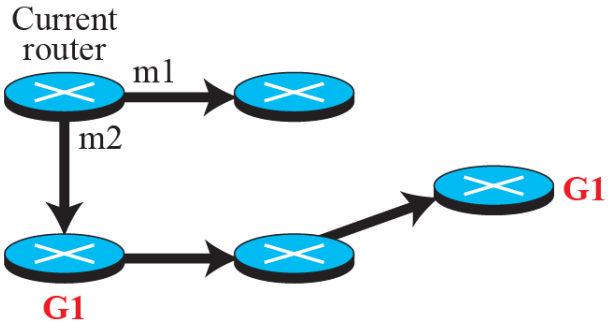
Figure 221.13: Example of tree formation in MOSPF



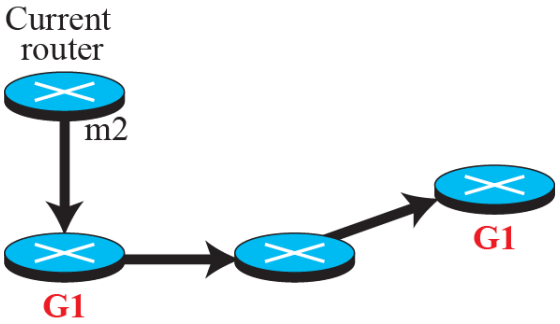
a. An internet with some active groups



b. S-G1 shortest-path tree



c. S-G1 subtree seen by current router



d. S-G1 pruned subtree

Forwarding table for current router

Group-Source	Interface
S, G1	m2
...	...

221.3.3 PIM

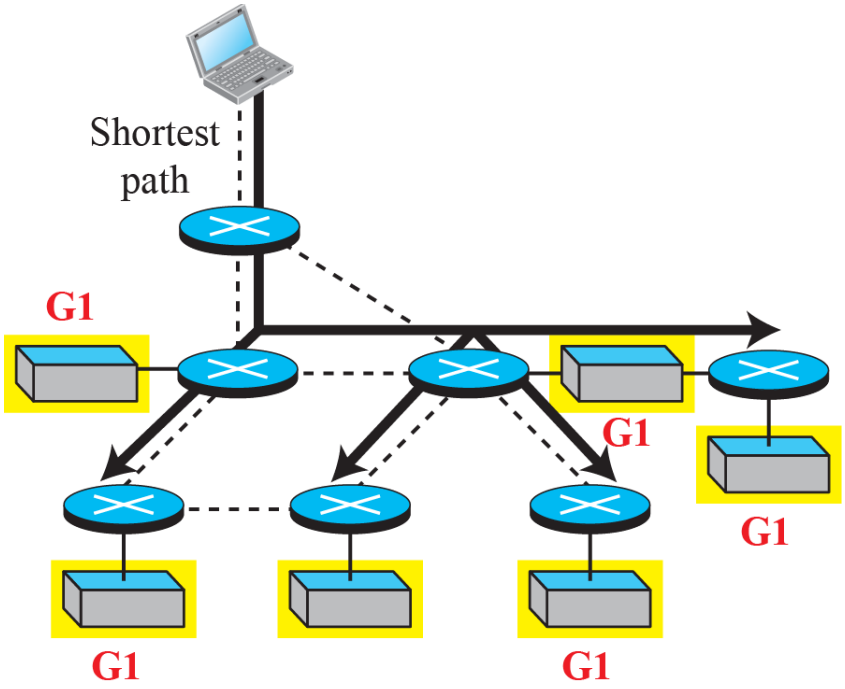
Protocol Independent Multicast (PIM) is the name given to a common protocol that needs a unicast routing protocol for its operation, but **the unicast protocol can be either a distance-vector protocol or a link-state protocol.**

In other words, **PIM needs to use the forwarding table of a unicast routing protocol to find the next router in a path to the destination,** but it does not matter how the forwarding table is created. PIM has another interesting feature: it can work in two different modes: **dense and sparse**.

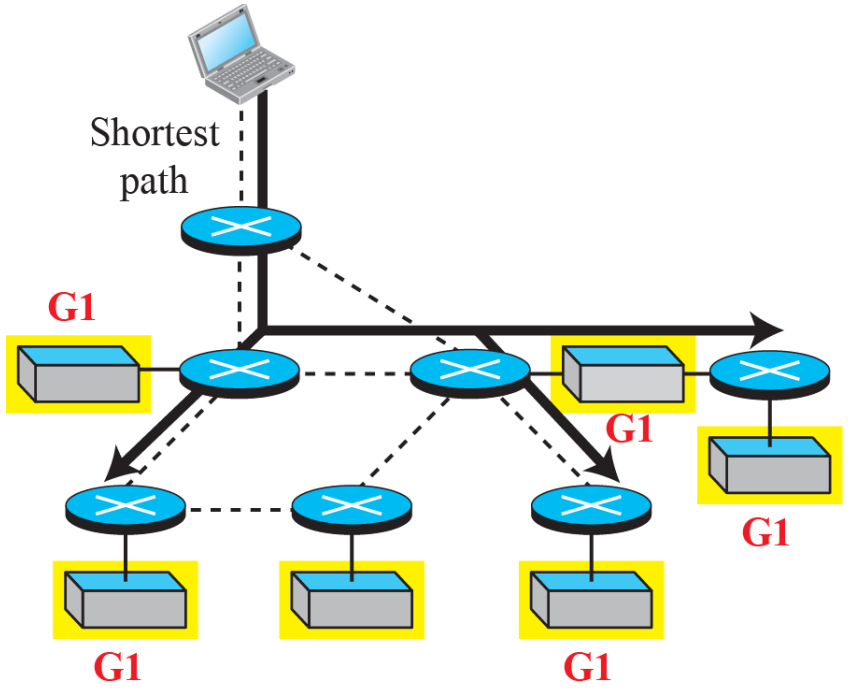
Protocol Independent Multicast – Dense Mode (PIM-DM))

- Number of routers with active members for the group is large.
- Uses source-based tree approach.
- It uses RPF to avoid receiving the duplicate packets.
- Then it floods the packet through all interfaces(broadcast)
- On receiving unwanted packet sends prune message to the upstream router (achieve multicast)

Figure 221.14: Idea behind PIM-DM



a. First packet is broadcast.

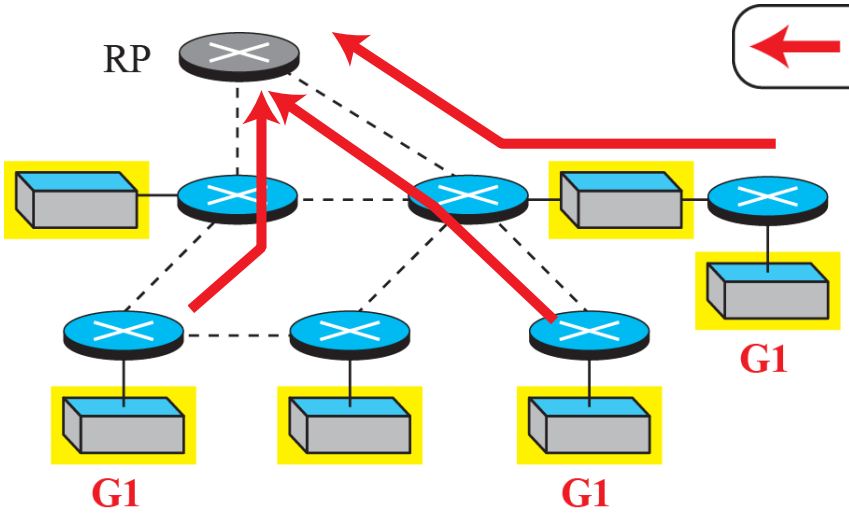


b. Second packet is multicast.

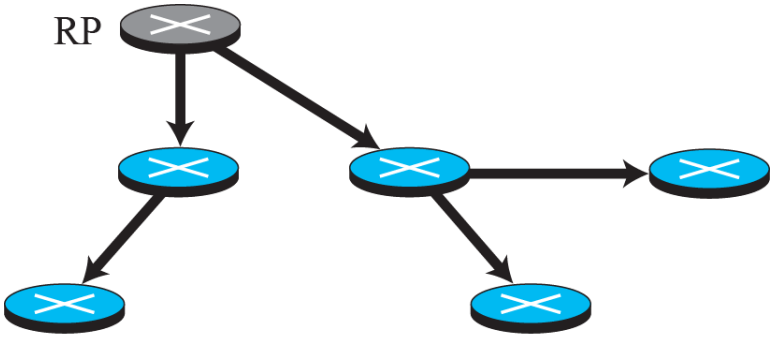
Protocol Independent Multicast – Sparse Mode (PIM-SM))

- It is used when number of routers with attached members is small.
- Uses group shared tree approach
- 2 steps – Every router forwards the multicast packet to core router using tunneling concept.
- The core de-capsulate and send the multicast packet to its destinations
- How to select core router?

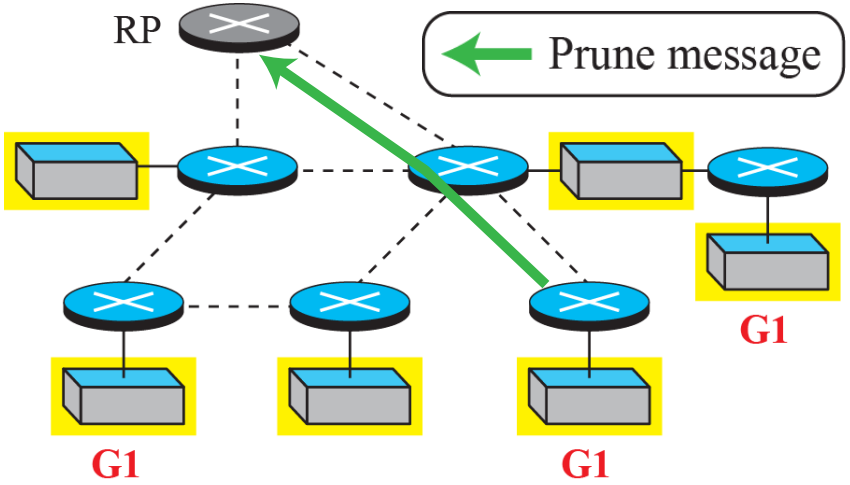
Figure 221.15: Idea behind PIM-SM



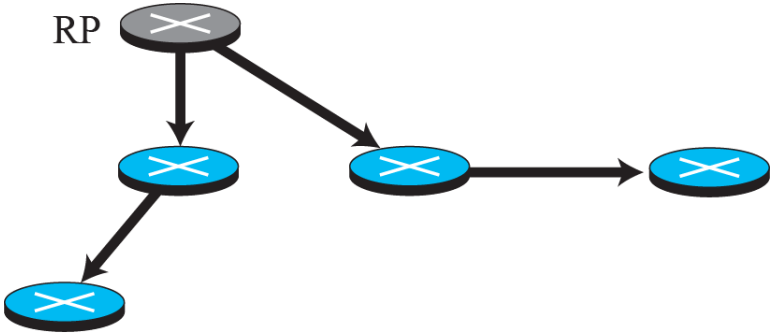
a. Three networks join group G1



b. Multicast tree after joins



c. One network leaves group G1



d. Multicast tree after pruning

221.4 INTERDOMAIN PROTOCOLS

The three protocols we discussed for multicast routing, DVMRP, MOSPF, and PIM, are designed to provide multicast communication inside an autonomous system. When the members of the groups are spread among different domains (ASs), we need an interdomain multicast routing protocol.

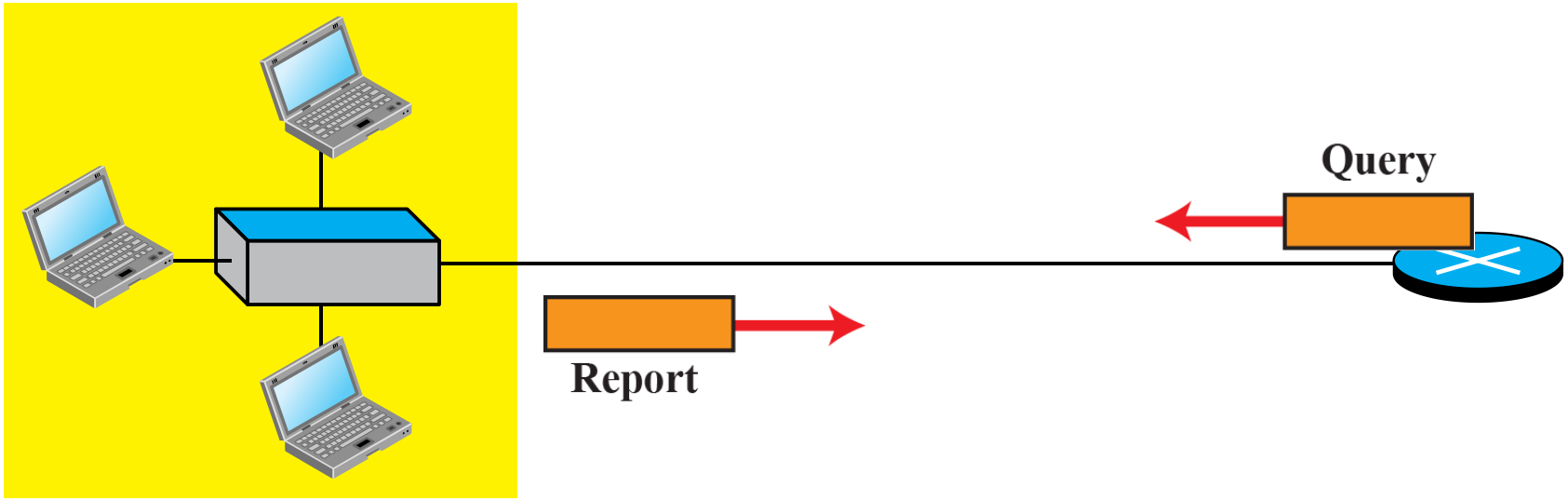
221.5 IGMP

The protocol that is used today for collecting information about group membership is the Internet Group Management Protocol (IGMP). IGMP is a protocol defined at the network layer; it is one of the auxiliary protocols, like ICMP, which is considered part of the IP. IGMP messages, like ICMP messages, are encapsulated in an IP datagram.

221.5.1 Messages

There are only two types of messages in IGMP version 3, query and report messages, as shown in Figure 221.16. A query message is periodically sent by a router to all hosts attached to it to ask them to report their interests about membership in groups. A report message is sent by a host as a response to a query message.

Figure 221.16: IGMP operation



221.5.2 Propagation of Information

After a router has collected membership information from the hosts and other routers at its own level in the tree, it can propagate it to the router located in a higher level of the tree. Finally, the router at the tree root can get the membership information to build the multicast tree. The process, however, is more complex than what we can explain in one paragraph. Interested readers can check the book website for the complete description of this protocol.

221.5.3 Encapsulation

The IGMP message is encapsulated in an IP datagram with the value of the protocol field set to 2 and the TTL field set to 21. The destination IP address of the datagram, however, depends on the type of message, as shown in Table 221.21.

Table 221.1: Destination IP Addresses

Message Type	*IP Address*
General Query	224.0.0.1
Other Queries	Group address
Report	224.0.0.22

4-5 NEXT GENERATION IP

The address depletion of IPv4 and other shortcomings of this protocol prompted a new version of IP protocol in the early 1990s. The new version, which is called Internet Protocol version 6 (IPv6) or IP new generation (IPng) was a proposal to augment the address space of IPv4 and at the same time redesign the format of the IP packet and revise some auxiliary protocols such as ICMP.

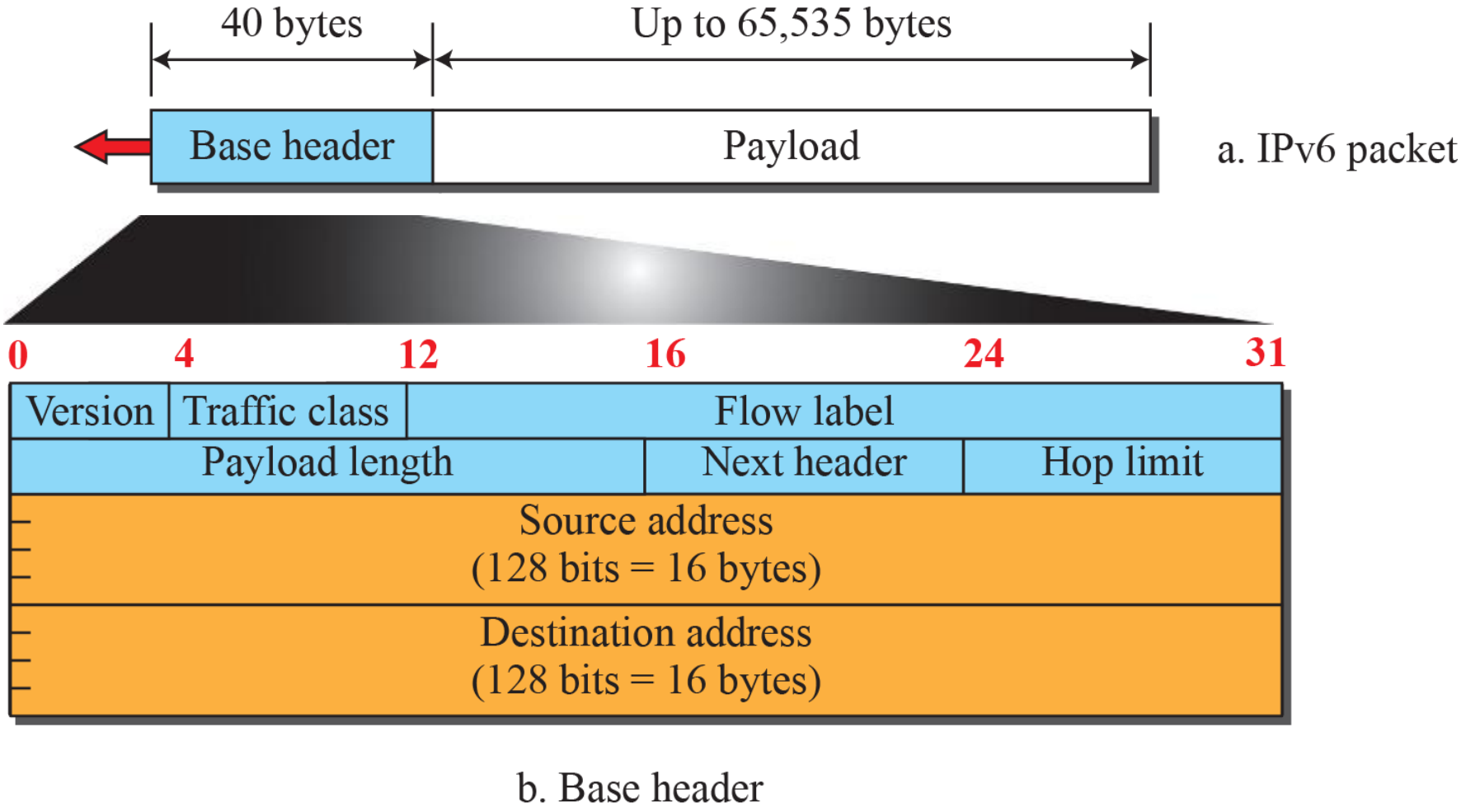
IPv6 Benifits

- More addresses
- Allow Auto-configuration (plug-n-play)
- More levels of Hierarchy
  - Route aggregation
- Ability to do end to end IPsec
  - No need of NAT
- Simplified Headers
- Support for new options
- Allowance for extensions
- Support for more security
  - Encryption and Authentication options
- Support for QOS

4.5.1 Packet Format

The IPv6 packet is shown in Figure 4.101. Each packet is composed of a base header followed by the payload. The base header occupies 40 bytes, whereas payload can be up to 65,535 bytes of information.

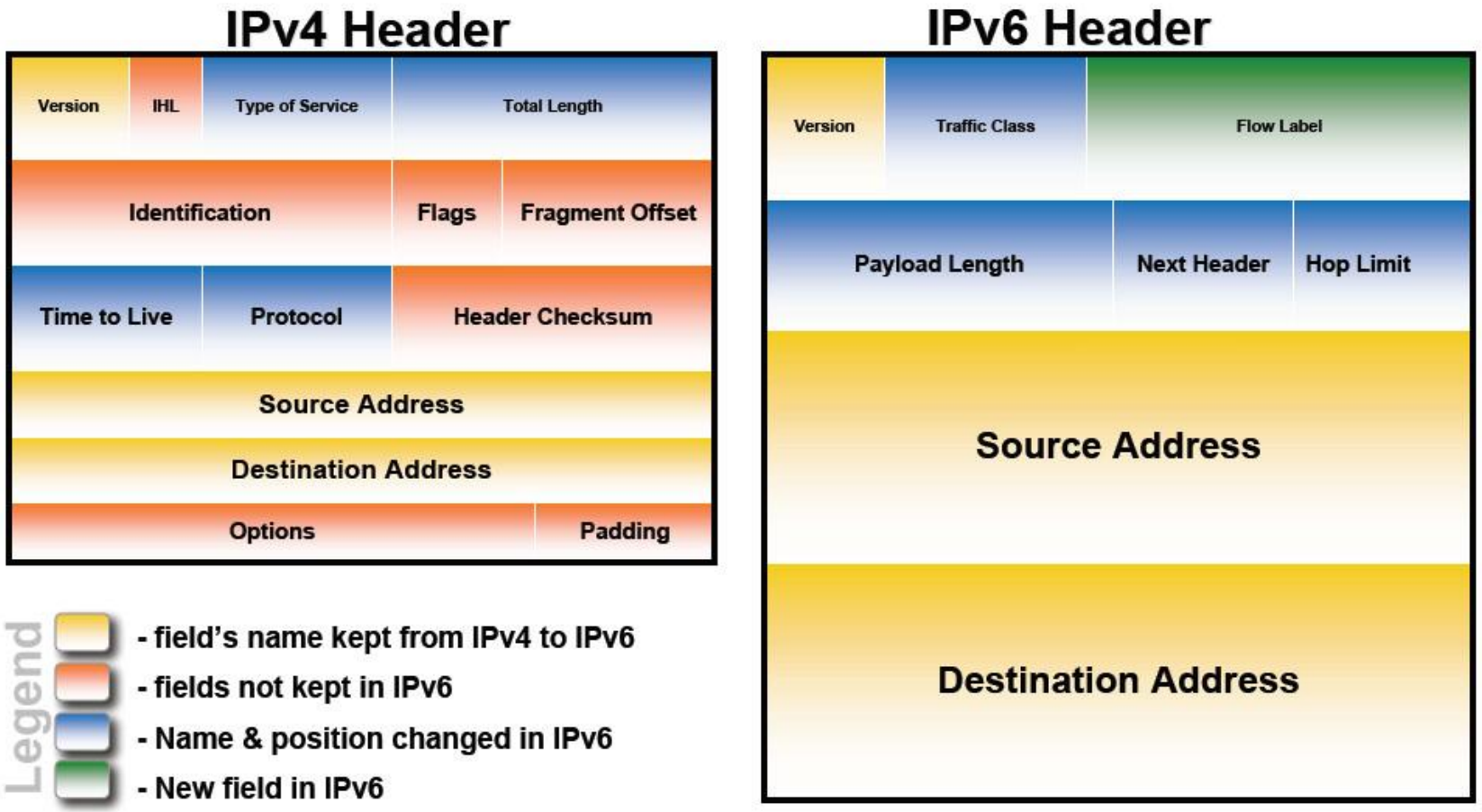
Figure 4.101: IPv6 datagram



Header comparisons

- 40 Bytes
- Address increased from 32 bits to 128 bits
- Header checksum and Hlen is removed
- New flow label field is added
- TOS – Traffic class
- Protocol – next header
- TTL – Hop Limit

IPv6 Header


IPv4 Header
Version
IHL
Type of Service
Total Length
Identification
Flags
Fragment Offset
Time to Live
Protocol
Header Checksum
Source Address
Destination Address
Options
Padding
Legend
- field's name kept from IPv4 to IPv6
- fields not kept in IPv6
- Name & position changed in IPv6
- New field in IPv6
IPv6 Header
Version
Traffic Class
Flow Label
Payload Length
Next Header
Hop Limit
Source Address
Destination Address

4.5.2 IPv6 Addressing

The main reason for migration from IPv4 to IPv6 is the small size of the address space in IPv4. In this section, we show how the huge address space of IPv6 prevents address depletion in the future. We also discuss how the new addressing responds to some problems in the IPv4 addressing mechanism. An IPv6 address is 128 bits or 16 bytes (octets) long, four times the address length in IPv4.

IPv6 Addressing

128 Bits = 16 Bytes(128/8) = 32 Hexadecimal digits(1byte = 2hex)

32 Hexadecimal digits is divided into eight groups of 4 hexadecimal digits separated by a colon(:)

2000:1111:2222:3333:0000:0000:0001:1111

1 2 3 4 5 6 7 8

IPv6 Addressing Contd.

- Two Addressing Rules:
- Leading zeroes can be omitted

eg:2000:1111:0000:0001:0010:0111:1100:0001

is equivalent to

2000:1111:0:1:10:111:1100:1

- Series of successive zeroes can be replaced by a :: **only once**

eg: 2000:0000:0000:0000:0000:1111:0000:0000

is equivalent to 2000::1111:0:0

IPv6 Addressing Contd.

- Write the abbreviated form of
  2000:0d02:0000:0000:0015:0000:0000:0073

a) 2000:d02::15:0:0:73

b) 2000:d02:0:0:15::73

IPv6 Addressing Contd.

- *Expand the address 0:15::1:12:1213 to its original*

- Answer
  - Align to the left side of double colon
  - Align to the right side of double colon
  - Fill the gaps with zeros

```
xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx
  0:   15:                   :    1:   12:1213
```

  - The original address is

```
0000:0015:0000:0000:0000:0001:0012:1213
```

IPv6 Addressing Contd.

- Address types
  - Unicast (one-to-one)
    - aggregatable with prefixes of arbitrary length
      - similar to CIDR
  - Anycast (one-to-nearest)
    - Packet delivered to one member of group
  - Multicast (one-to-many)
    - No traditional broadcast in IPv6
    - Broadcasting is considered as special case of multicast
    - Uses special link-local all nodes multicast group
    - FF02::1 - Analogous to 224.0.0.1
    - FF02::2 - link local scope all routers

IPv6 Addressing Contd.

- Unicast
  - Global Unicast
    - special purpose global unicast e.g. embedded with IPv4
  - Unique Local
  - Site-local unicast (deprecated)
  - Link-local unicast
- Global Unicast
  - For one to one communication between hosts
  - Currently, first 3 bits are `001`
  - sub divided into 3 parts
    - global routing prefix
    - subnet identifier
    - interface identifier
      - similar to hostid in IPv4

IPv6 Addressing Contd.

- `::/128` - unspecified address
- `::/0` - default route
- `::1/128` - loopback address
- FECx-FEFx - Site local (Deprecated)
- `FE80::/10` - link local addresses
  - unique on a single link
- `FC00:/7` - Unique local
  - routable within a specific site
  - equivalent to `10.0.0.0/8, 172.16/12, 192.168.0.0/16`
- `::FFFF:0:0/96` - IPv4 Mapped address
- `FF00::/8` - Multicast addresses

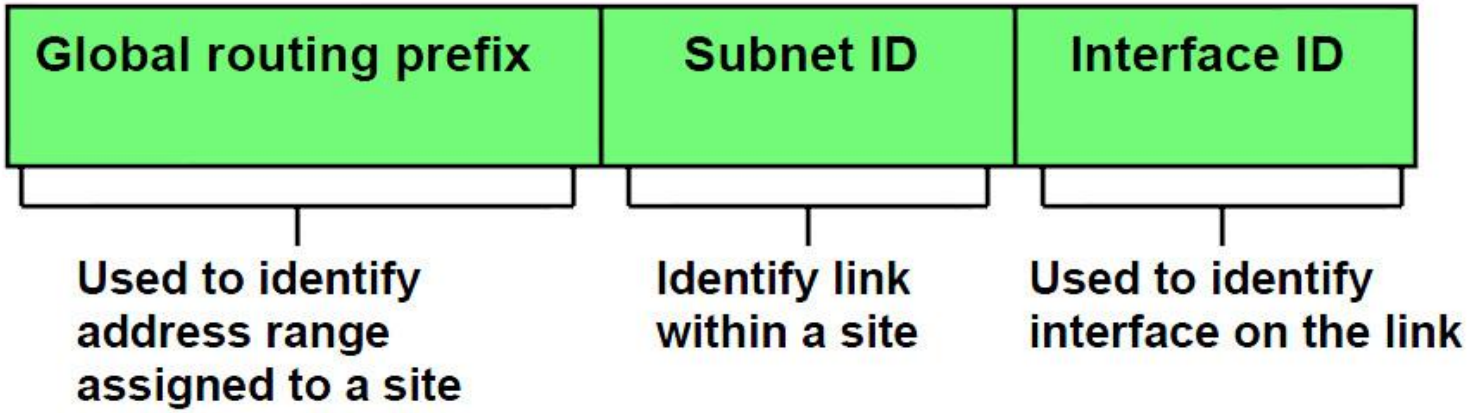
IPv6 Addressing Contd.

- Address space allocation

Block prefix	*CIDR*	*Block assignment*	*Fraction*
0000 0000	0000::/8	Special addresses	1/256
001	**2000::/3**	**Global unicast**	**1/8**
1111 110	FC00::/7	Unique local unicast	1/128
1111 1110 10	FE80::/10	Link local addresses	1/1024
1111 1111	FF00::/8	Multicast addresses	1/256

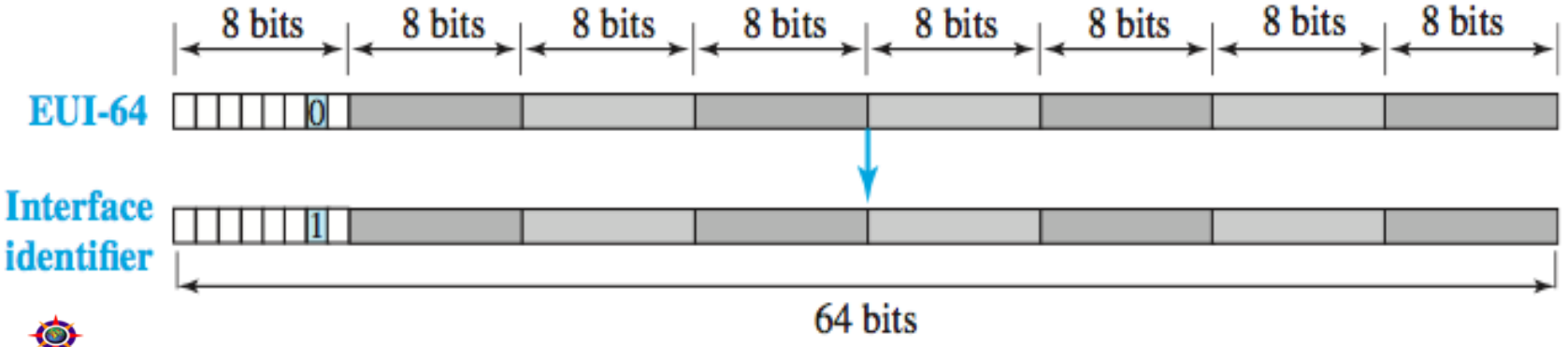
IPv6 Addressing Contd.

- Global unicast
  - Global Routing Prefix (Network Number) =48 bits
  - Subnet Bits =16 bits
  - Interface ID (Host portion)=64 bits



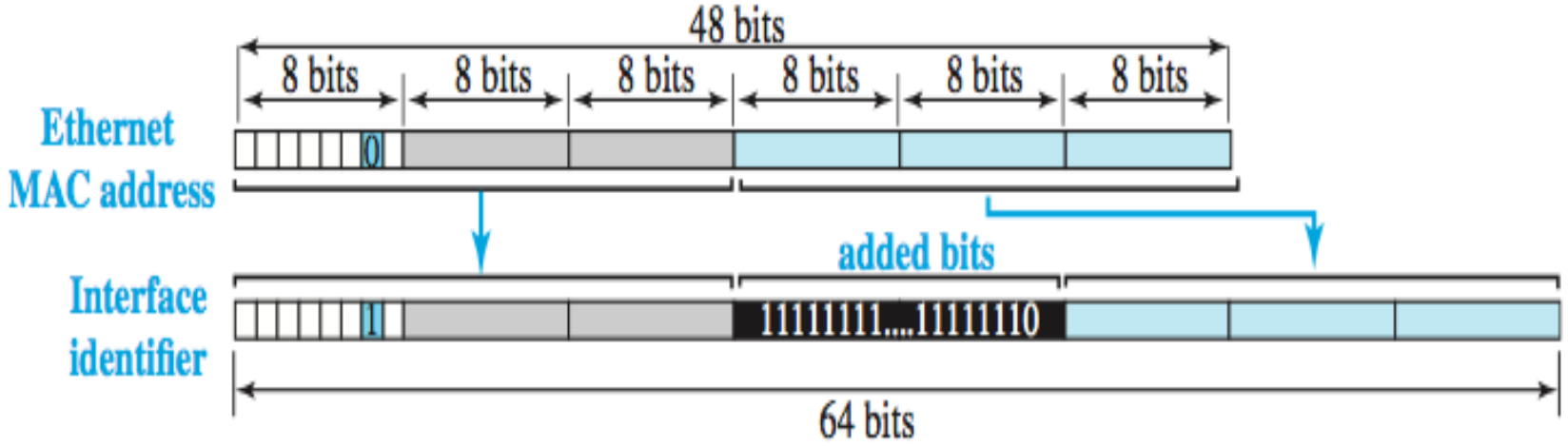
IPv6 Addressing Contd.

- EUI-64
  - Extended Unique Identifier (link level address)
    - Defined by IEEE
    - Ethernet is 48 bits
- Mapping EUI-64
  - IPv6 allows direct mapping of link layer address
    - In interface id part
  - Set the local/global bit to global (1)



IPv6 Addressing Contd.

- Mapping Ethernet MAC address
  - Change the local/global bit to 1
  - Provide additional 16 bits
    - Defined as FFFE



IPv6 Addressing Contd.

- Given the ethernet address as
  - $(F5\text{-}A9\text{-}23\text{-}14\text{-}7A\text{-}D2)_{16}$
- Find the Interface ID in IPv6
- Hint
  - Change universal/local bit
  - Insert the two octets FFFE

- Answer
  - **F7A9:23FF:FE14:7AD2**

IPv6 Addressing Contd.

An organization is assigned the block **2000:1456:2474/48**

Physical address of m/c is **F5-A9-23-14-7A-D2**

Find the IP address of 3rd subnet for this m/c

– Hint
  - Change universal/local bit
  - put the correct value (in hex) for 3rd subnet

- Answer
  – **2000:1456:2474:0003:F7A9:23FF:FE14:7AD2**

Auto-configuration

- RFC 2462
- allows a host to generate its own address
- uses local information and router advertisement info
- routers advertise the prefix identifying the subnets
- host generates the interface id within the subnet
- when router is absent
  - host generates link local address
  - permits communication on same link
- applies only to hosts and not to routers
  - routers need to be configured by some other means

Auto-configuration

- IPv6 can configure itself or use DHCP
- Autoconfiguration process/steps
  - Create a link local address for itself
    - 1111 1110 10 + 54 0bits + EUI-64(ethernet)
  - Sends a neighbour solicitation message
  - Should not receive neighbour advertisement message
    - No other node having the same address
  - Sends to router solicitation message for global unicast address prefix
  - Router responds with global unicast prefix and subnet ID
  - use interface id to generate global unicast address
    - EUI - 64

Auto-configuration

- For the info below, find out
  - a) global unicast address
  - b) Link local address
  - Ethernet address: **F5-A9-23-11-9B-E2**
  - Global Unicast prefix: **3A21:1216:2165**
  - Subnet Identifier: **A245**
- Answer
  - EUI-64: : **F*7*A9:23*FF:FE*11:9BE2**
  - Link local: FE80::**F*7*A9:23*FF:FE*11:9BE2**
  - Global unicast prefix: **3A21:1216:2165:A245**
  - Global Unicast Address: **3A21:1216:2165:A245:F7A9:23FF:FE11:9BE2**

ICMPv6 Messages

- Error Messages
  - Destination unreachable
    - no route, destination prohibited/unreachable, no port
  - Packet Too Big
    - no fragmentation is done
  - Time Exceeded
    - Hop Limit exceeded (similar to TTL Expiry)
    - fragmentation reassembly time exceeded
  - parameter problems
    - erroneous header etc
  - Note:
    - source quench eliminated
    - ICMP Redirect moved to Neighbor Discovery

4.5.3 Transition from IPv4 to IPv6

Although we have a new version of the IP protocol, how can we make the transition to stop using IPv4 and start using IPv6? The first solution that comes to mind is to define a transition day on which every host or router should stop using the old version and start using the new version.

- Dual Stack
- Tunneling
- Header Translation

Transition from IPv4 to IPv6

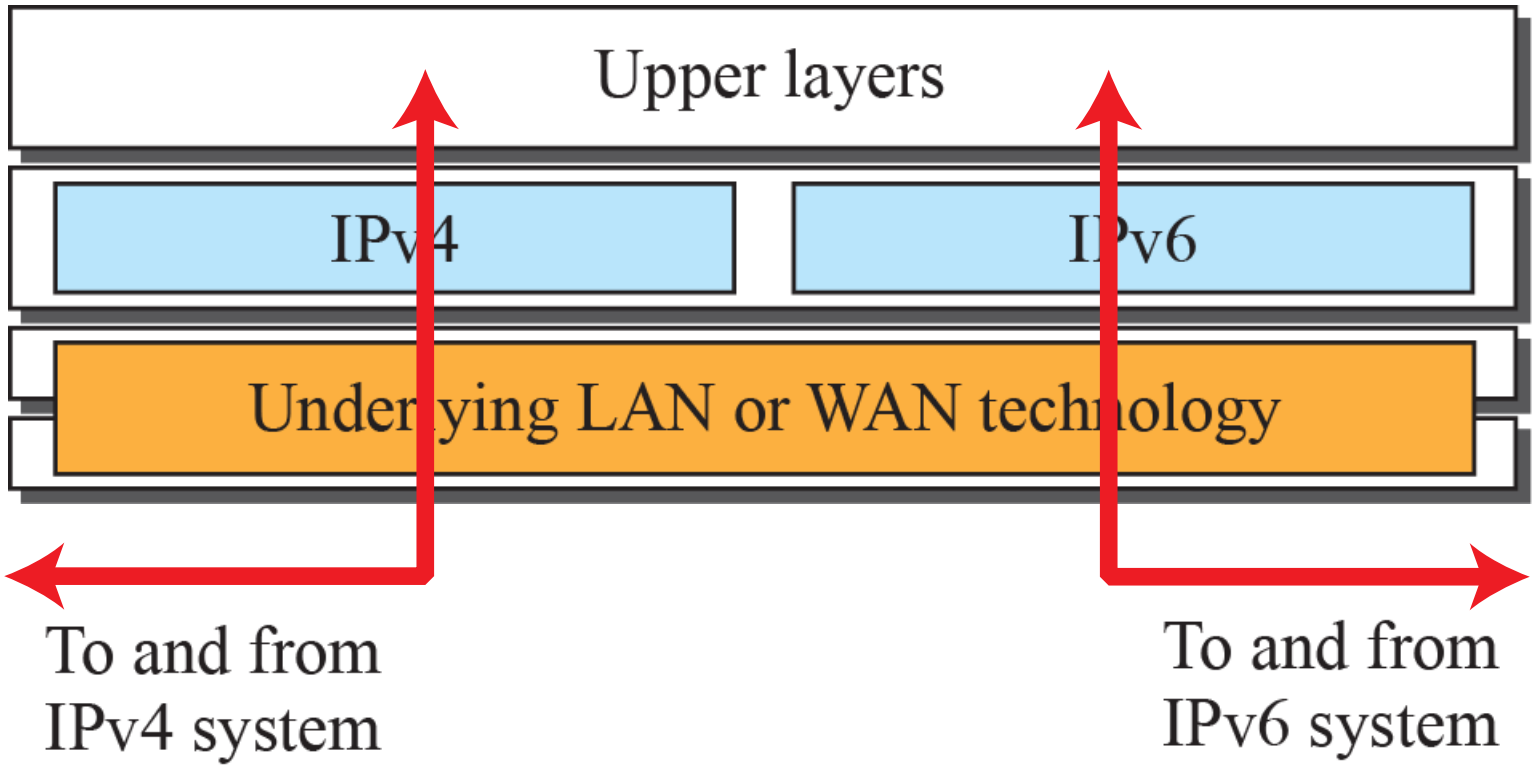
- Dual Stack
  - at network layer, both IPv4 and IPv6 present
  - DNS response determines which stack to use
    - response can IPv4 or IPv6 or both
      - If IPv6 is present, use IPv6
      - Else use IPv4
  - Two stacks co-exist indefinitely
  - Bundled with OS, no extra add on software/cost
    - Linux, Mac, Windows (including Windows XP)
  - No change in Datalink layer/Transport layer

Using Dual Stack

- Configuring DNS Server
  - install `bind9`
    - `sudo apt-get install bind9`
  - configure `named.conf`

  - How does dual stack work
    - DNS provides
      - only IPv6 address - AAAA Record
      - only IPv4 address - A record
      - either both IPv4 and IPv6 address.
        - » IPv6 is used then

Figure 4.106: Dual stack



IPv6 Tunneling over IPv4

- Why tunneling?
  - Two islands of IPv6 network
    - connected via IPv4 network
  - A transition strategy to enable communication among IPv6 network
- What is tunneling
  - Two end points are defined
  - each is aware of two types of network
  - each encapsulates and de-capsulates

IPv6 Tunneling over IPv4

- *tunneling:* IPv6 datagram carried as *payload* in IPv4 datagram among IPv4 routers
  - similar to IP in IP tunnels
- There is no virtual connection setup
  - ❿ two end points need to be configured
- intermedia IPv4 routers treat it as IPv4 packet

Figure 4.107: Tunneling strategy

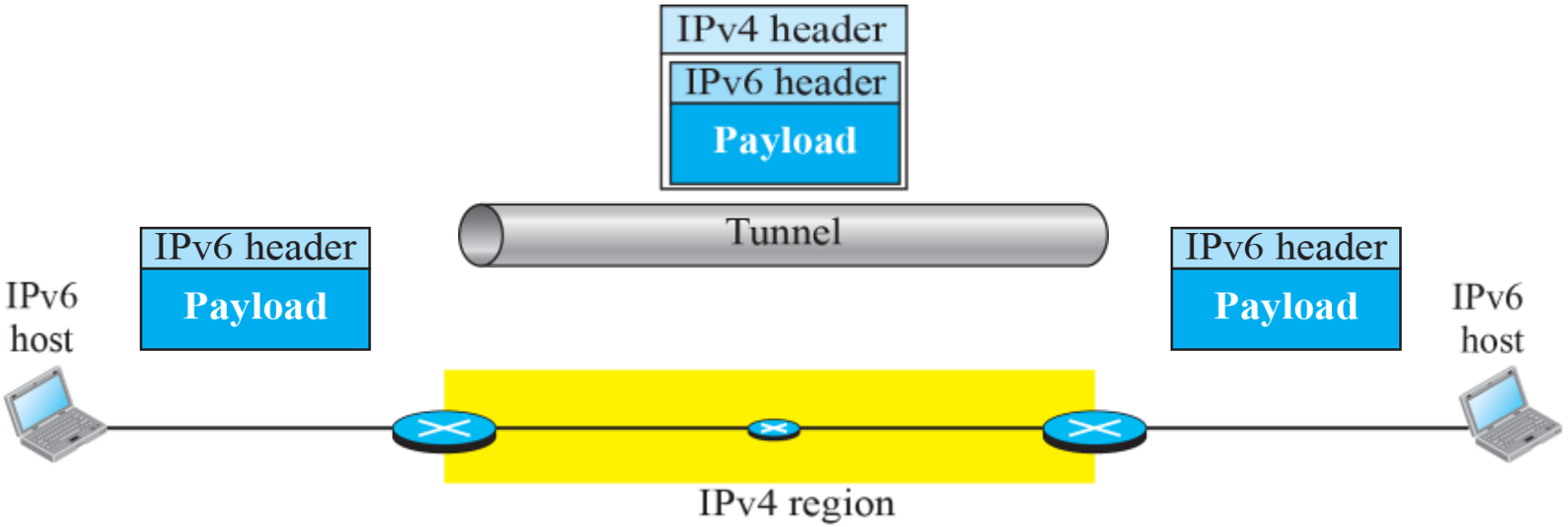


Figure 4.108: Header translation strategy

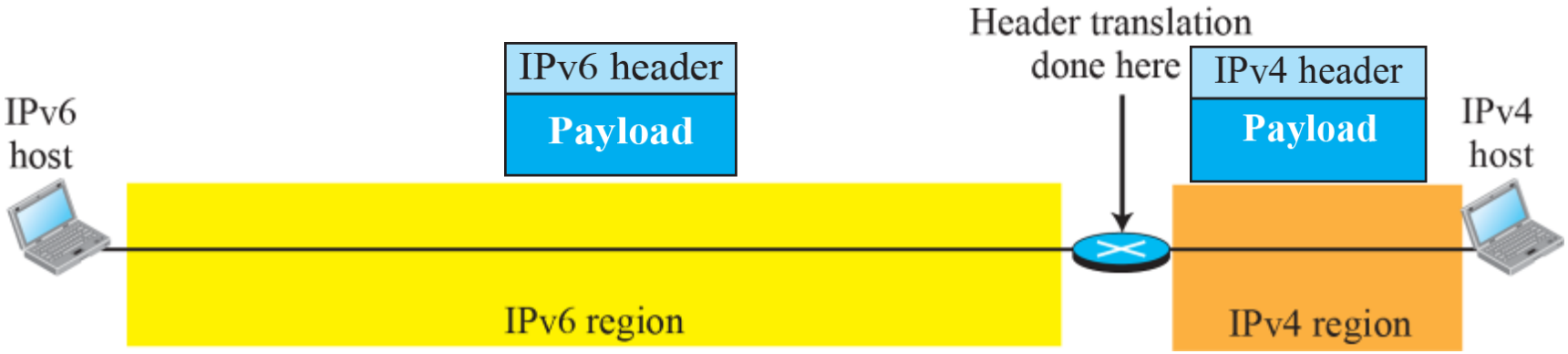


Table 20.11 *Header translation*

Header Translation Procedure
1. The IPv6 mapped address is changed to an IPv4 address by extracting the rightmost 32 bits.
2. The value of the IPv6 priority field is discarded.
3. The type of service field in IPv4 is set to zero.
4. The checksum for IPv4 is calculated and inserted in the corresponding field.
5. The IPv6 flow label is ignored.
6. Compatible extension headers are converted to options and inserted in the IPv4 header. Some may have to be dropped.
7. The length of IPv4 header is calculated and inserted into the corresponding field.
8. The total length of the IPv4 packet is calculated and inserted in the corresponding field.

- Thank You