Network Layer

Dr. Xiqun Lu

College of Computer Science

Zhejiang University

Outline

- Overview of network layer
- Routing algorithms
- The network layer in the Internet
- MPLS (Multiprotocol Label Switching)

Outline

- Overview of network layer
- Routing algorithms
- The network layer in the Internet
- MPLS (Multiprotocol Label Switching)

The Network Layer

- The Network Layer
  - is concerned with getting packet from source to destination
- The Data Link Layer
  - is concerned with moving frames from one end of wire to the other.

Two Important Network Layer Functions [8]

- The role of the network layer is deceptively simple — to move packets from a sending host to a receiving host.
  - Each router has **a forwarding (internal, routing) table**.
  - The forwarding table is indexed by either **the destination address** in the packet header or **an indication of connection** to which the packet belongs.
- **Forwarding** (the main function of a router)
  - Forwarding involves the transfer of a packet from an incoming link to an outgoing link within a single router.
  - Forwarding refers the router-local action.
- **Routing** (to build *the forwarding table* for each router)
  - Routing involves all of the network's routers, whose collective interactions via **routing protocols** determine the paths that packets take on their trips from source to destination node. The routing algorithm determines the values that are inserted into the routers' forwarding tables.
  - Routing refers to the network-wide process.
    - Centralized or decentralized.

Network Layer Design Issues

- The issues include the service provided to the transport layer and the internal design of the network.
- Store-and-forward Packet Switching

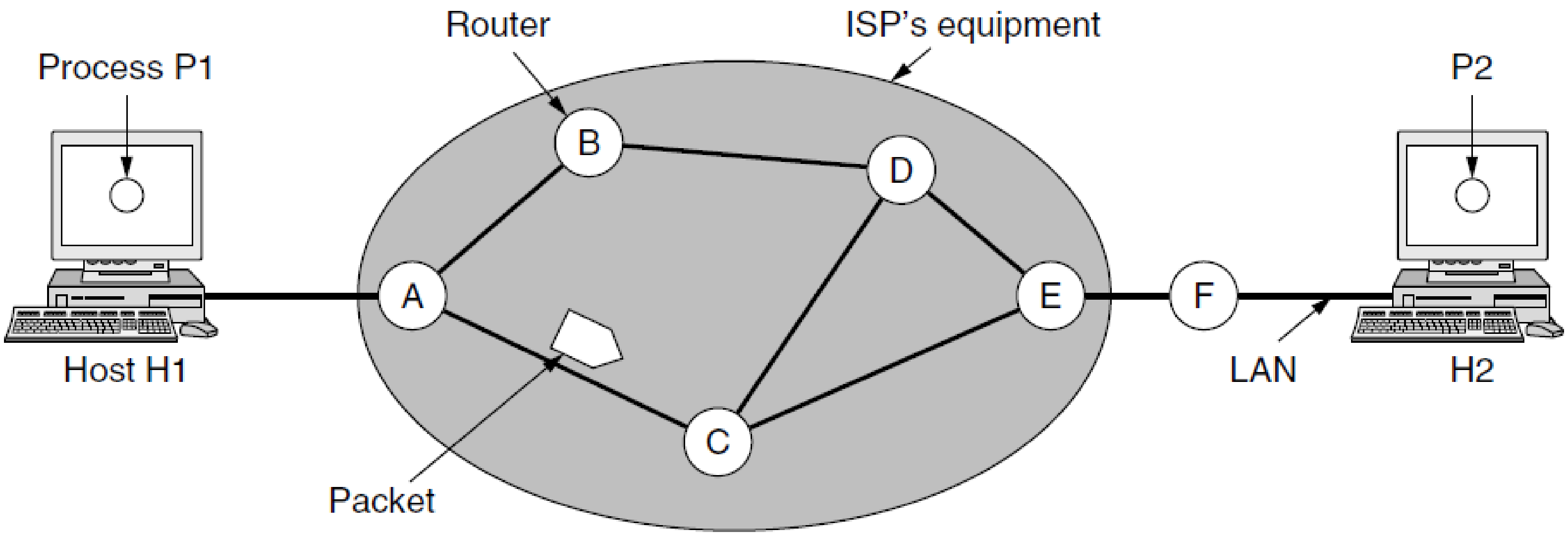


Figure 5-1. The environment of the network layer protocols.

A packet is stored a router until it has fully arrived and the link has finished its processing by verifying the checksum. Then it is **forwarded** to the next router along the path until it reaches the destination host, where it is delivered.

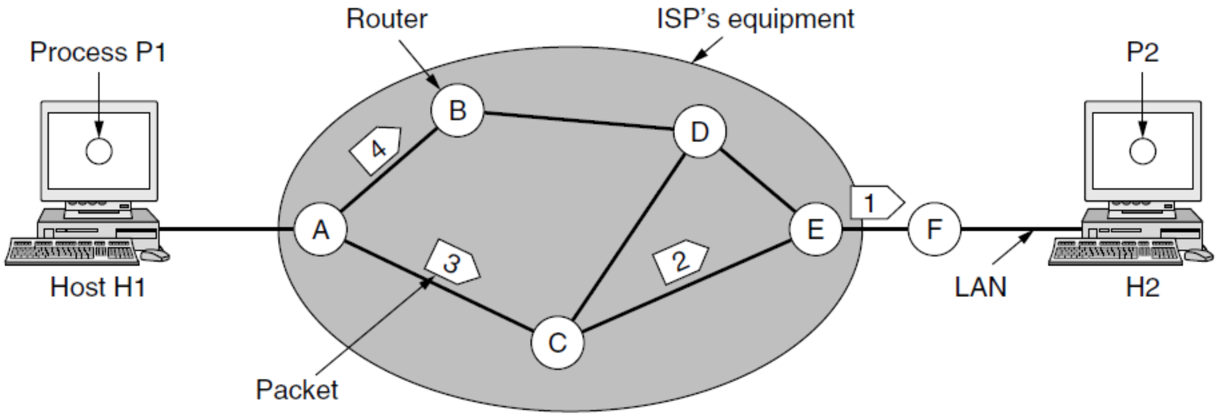
Services Provided to the Transport Layer

- Design goals
  - The services should be independent of the router technology
  - The transport layer should be shielded from the number, type, and topology of the routers present
  - The network addresses should use a uniform numbering plan.

- Connection-oriented & Connectionless service
  - Two warring factions
    - The internet community: Connectionless service
    - The telephone company: Connection-oriented
  - Dispute focus is: network layer should provide connection-oriented service or connectionless service.

Implementation of Connectionless Service

- Packets are injected into the network *individually* and routed *independently* of each other.
  - The *packets* are frequently called **datagrams**.



A's table (initially)

Dest.	Line
A	–
B	B
C	C
D	B
E	C
F	C

A's table (later)

A	–
B	B
C	C
D	B
E	B
F	B

C's table

A	A
B	A
C	–
D	E
E	E
F	E

E's table

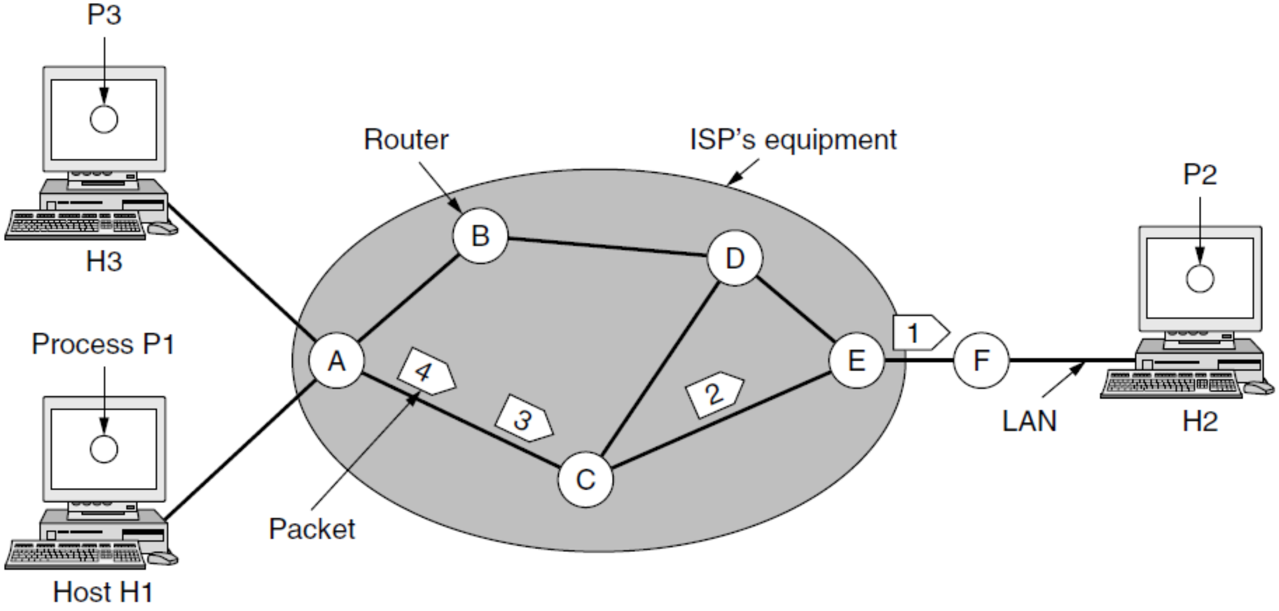
A	C
B	D
C	C
D	D
E	–
F	F

Figure 5-2. Routing within a datagram network.

Every router has **a forwarding table**. Each table entry is a pair consisting of *a destination* and *the outgoing line* to use for that destination.

Implementation of Connection-Oriented Service

- With connection-oriented service, each packet carries **an identifier** telling which virtual circuit it belongs to.
  - Router A will assign different identifies to different connections although they may use the same virtual circuit. — Label Switching (**MPLS**, MultiProtocol LS)



A's table

In		Out	
H1	1	C	1
H3	1	C	2

C's table

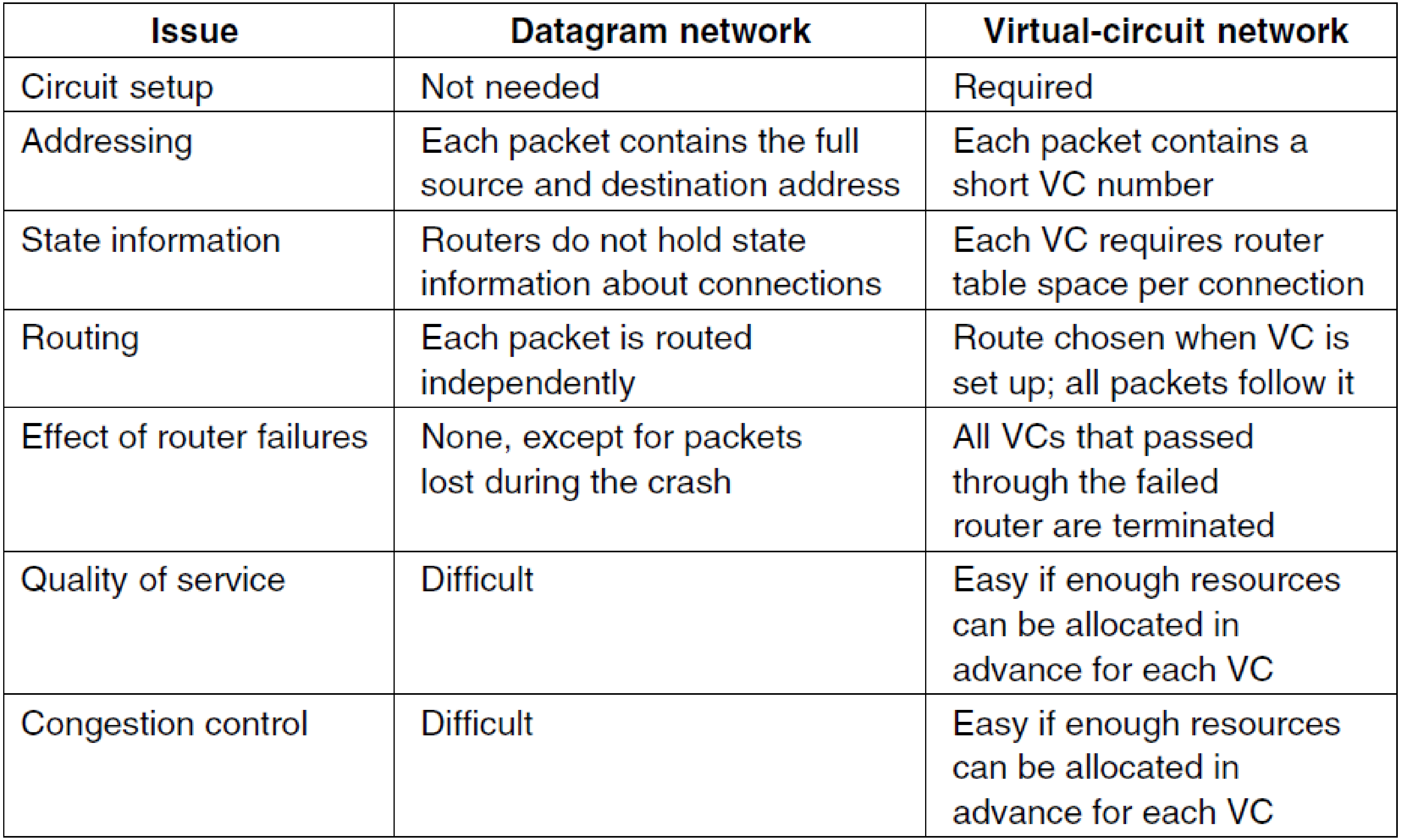
A	1	E	1
A	2	E	2

E's table

C	1	F	1
C	2	F	2

Figure 5-3. Routing within a virtual-circuit network.

Virtual-Circuit vs. Datagram Networks



Issue	Datagram network	Virtual-circuit network
Circuit setup	Not needed	Required
Addressing	Each packet contains the full source and destination address	Each packet contains a short VC number
State information	Routers do not hold state information about connections	Each VC requires router table space per connection
Routing	Each packet is routed independently	Route chosen when VC is set up; all packets follow it
Effect of router failures	None, except for packets lost during the crash	All VCs that passed through the failed router are terminated
Quality of service	Difficult	Easy if enough resources can be allocated in advance for each VC
Congestion control	Difficult	Easy if enough resources can be allocated in advance for each VC

Figure 5-4. Comparison of datagram and virtual-circuit networks.

Outline

- Overview of network layer
- Routing algorithms
- The network layer in the Internet
- MPLS (Multiprotocol Label Switching)

Outline

- Overview of network layer
- Routing algorithms
- The network layer in the Internet
- MPLS (Multiprotocol Label Switching)

Routing Algorithms

- The main function of the network layer is routing packets from the source machine to the destination machine.
- Two functions of a router:
  - Forwarding: to handle each incoming packet, look up the route table, then forward to an output line.
  - Filling and updating the route (forwarding, or internal) table.
- Routing algorithms
  - Difference in Datagram and Virtual Circuit.
    - In datagram, the best route may have changed since last time
    - In virtual circuit, routing decisions are made only when a new virtual circuit is being set up.
  - Desirable properties: **correctness**, **simplicity**, **robustness**, **stability**, **fairness** and **efficiency**.

Classification of Routing Algorithms [8]

- We can classify routing algorithms into global routing algorithms and decentralized algorithms.
  - **Global** routing algorithms compute the least-cost path between a source and destination using complete, global knowledge about the network.
    - Link-state (LS) algorithms (may be appropriate for small-scale network)
  - In **decentralized** routing algorithms, the calculation of the least-cost path is carried out in an *iterative*, *distributed* manner.
    - Distance-vector (DV) algorithms (much better for large-scale network)
- A second broad way to classify routing algorithms is according to whether they are static or dynamic.
  - Static routing algorithms (non-adaptive, Lab4)
    - For example, a human manually edit a router’s forwarding table
  - Dynamic routing algorithms (adaptive)
    - Responsive to network changes, but also more susceptible to problems such as *routing loops* and *oscillation* in routes.

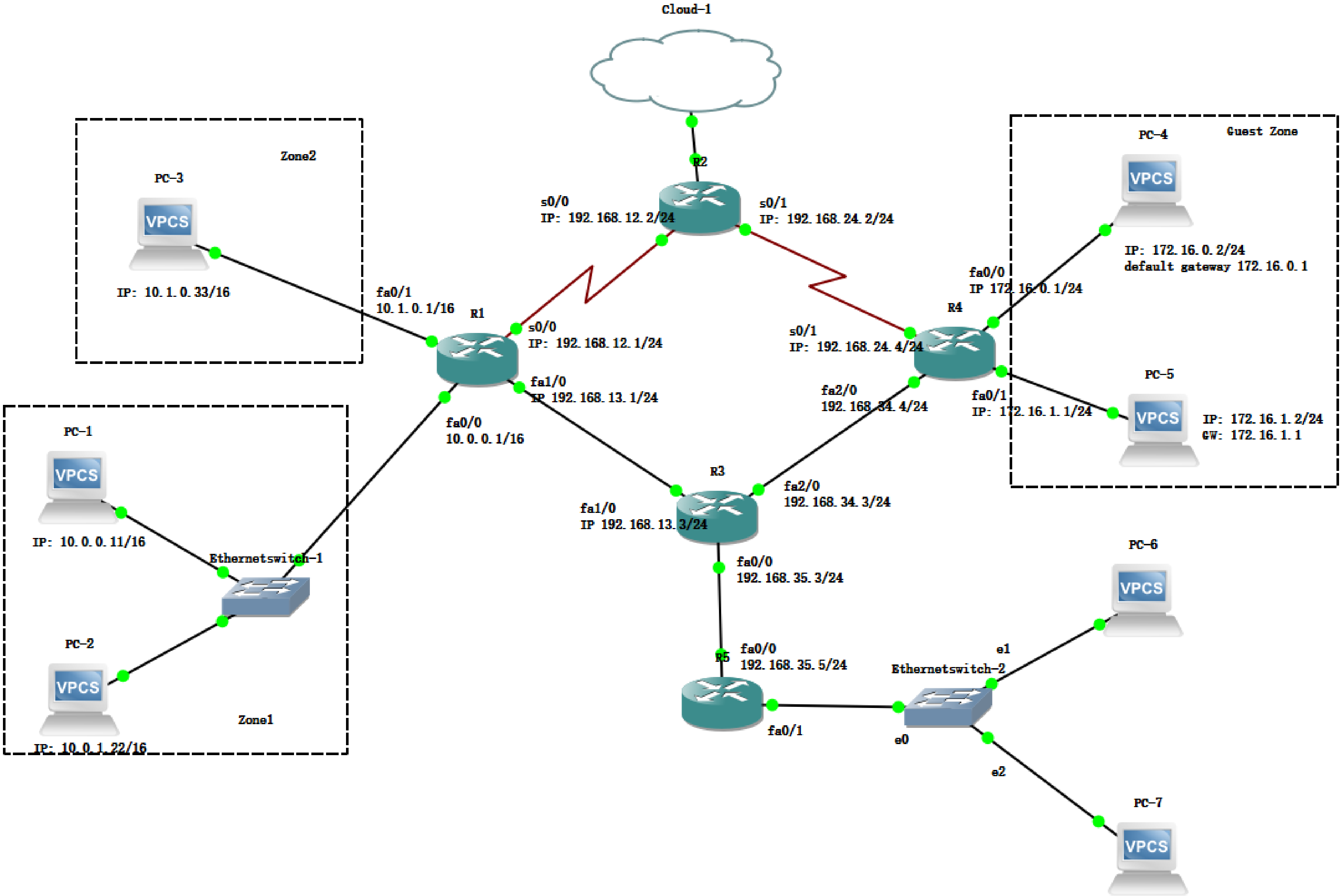
Outline

- Overview of network layer
- Routing algorithms
- The network layer in the Internet
- MPLS (Multiprotocol Label Switching)

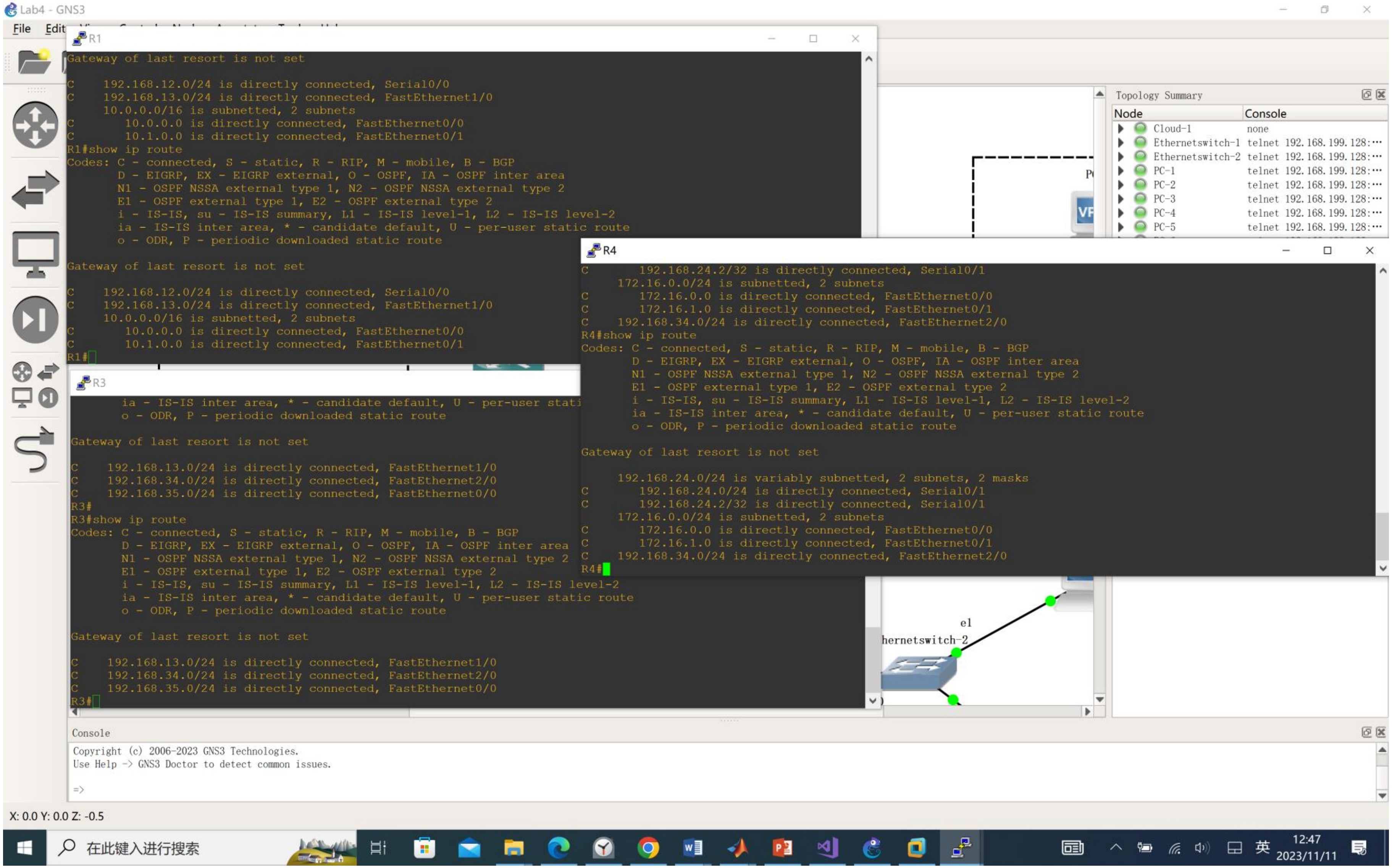
Outline

- Overview of network layer
- Routing algorithms
  - Static routing
  - Link-state (LS) routing algorithms
  - Distance-vector (DV) routing algorithms
  - Hierarchical routing
  - Broadcast routing
  - Multicast routing
  - Anycast routing
- The network layer in the Internet
- MPLS (Multiprotocol Label Switching)

Static Routing Example: **Lab4**



Static Routing Example: **Lab4**



R1

```
Gateway of last resort is not set

C    192.168.12.0/24 is directly connected, Serial0/0
C    192.168.13.0/24 is directly connected, FastEthernet1/0
     10.0.0.0/16 is subnetted, 2 subnets
C       10.0.0.0 is directly connected, FastEthernet0/0
C       10.1.0.0 is directly connected, FastEthernet0/1
R1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

C    192.168.12.0/24 is directly connected, Serial0/0
C    192.168.13.0/24 is directly connected, FastEthernet1/0
     10.0.0.0/16 is subnetted, 2 subnets
C       10.0.0.0 is directly connected, FastEthernet0/0
C       10.1.0.0 is directly connected, FastEthernet0/1
R1#
```

R4

```
C       192.168.24.2/32 is directly connected, Serial0/1
     172.16.0.0/24 is subnetted, 2 subnets
C       172.16.0.0 is directly connected, FastEthernet0/0
C       172.16.1.0 is directly connected, FastEthernet0/1
C    192.168.34.0/24 is directly connected, FastEthernet2/0
R4#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     192.168.24.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.24.0/24 is directly connected, Serial0/1
C       192.168.24.2/32 is directly connected, Serial0/1
     172.16.0.0/24 is subnetted, 2 subnets
C       172.16.0.0 is directly connected, FastEthernet0/0
C       172.16.1.0 is directly connected, FastEthernet0/1
C    192.168.34.0/24 is directly connected, FastEthernet2/0
R4#
```

R3

```
       ia - IS-IS inter area, * - candidate default, U - per-user stati
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

C    192.168.13.0/24 is directly connected, FastEthernet1/0
C    192.168.34.0/24 is directly connected, FastEthernet2/0
C    192.168.35.0/24 is directly connected, FastEthernet0/0
R3#
R3#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

C    192.168.13.0/24 is directly connected, FastEthernet1/0
C    192.168.34.0/24 is directly connected, FastEthernet2/0
C    192.168.35.0/24 is directly connected, FastEthernet0/0
R3#
```

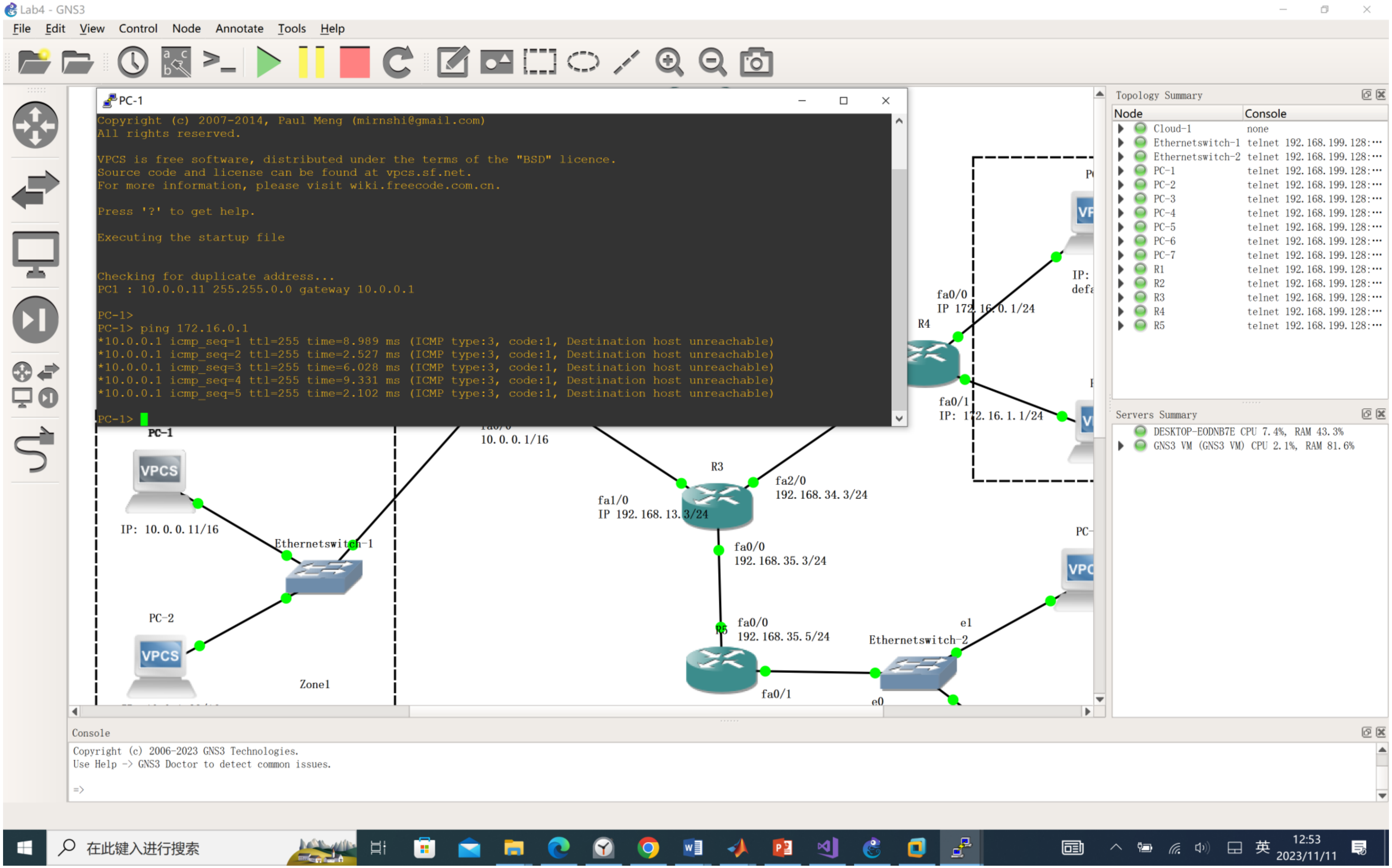
Topology Summary

Node	Console
Cloud-1	none
Ethernetswitch-1	telnet 192.168.199.128:…
Ethernetswitch-2	telnet 192.168.199.128:…
PC-1	telnet 192.168.199.128:…
PC-2	telnet 192.168.199.128:…
PC-3	telnet 192.168.199.128:…
PC-4	telnet 192.168.199.128:…
PC-5	telnet 192.168.199.128:…

Console

```
Copyright (c) 2006-2023 GNS3 Technologies.
Use Help -> GNS3 Doctor to detect common issues.

=>
```

Static Routing Example: **Lab4**



```
Copyright (c) 2007-2014, Paul Meng (mirnshi@gmail.com)
All rights reserved.

VPCS is free software, distributed under the terms of the "BSD" licence.
Source code and license can be found at vpcs.sf.net.
For more information, please visit wiki.freecode.com.cn.

Press '?' to get help.

Executing the startup file


Checking for duplicate address...
PC1 : 10.0.0.11 255.255.0.0 gateway 10.0.0.1

PC-1>
PC-1> ping 172.16.0.1
*10.0.0.1 icmp_seq=1 ttl=255 time=8.989 ms (ICMP type:3, code:1, Destination host unreachable)
*10.0.0.1 icmp_seq=2 ttl=255 time=2.527 ms (ICMP type:3, code:1, Destination host unreachable)
*10.0.0.1 icmp_seq=3 ttl=255 time=6.028 ms (ICMP type:3, code:1, Destination host unreachable)
*10.0.0.1 icmp_seq=4 ttl=255 time=9.331 ms (ICMP type:3, code:1, Destination host unreachable)
*10.0.0.1 icmp_seq=5 ttl=255 time=2.102 ms (ICMP type:3, code:1, Destination host unreachable)

PC-1>
```

Topology Summary

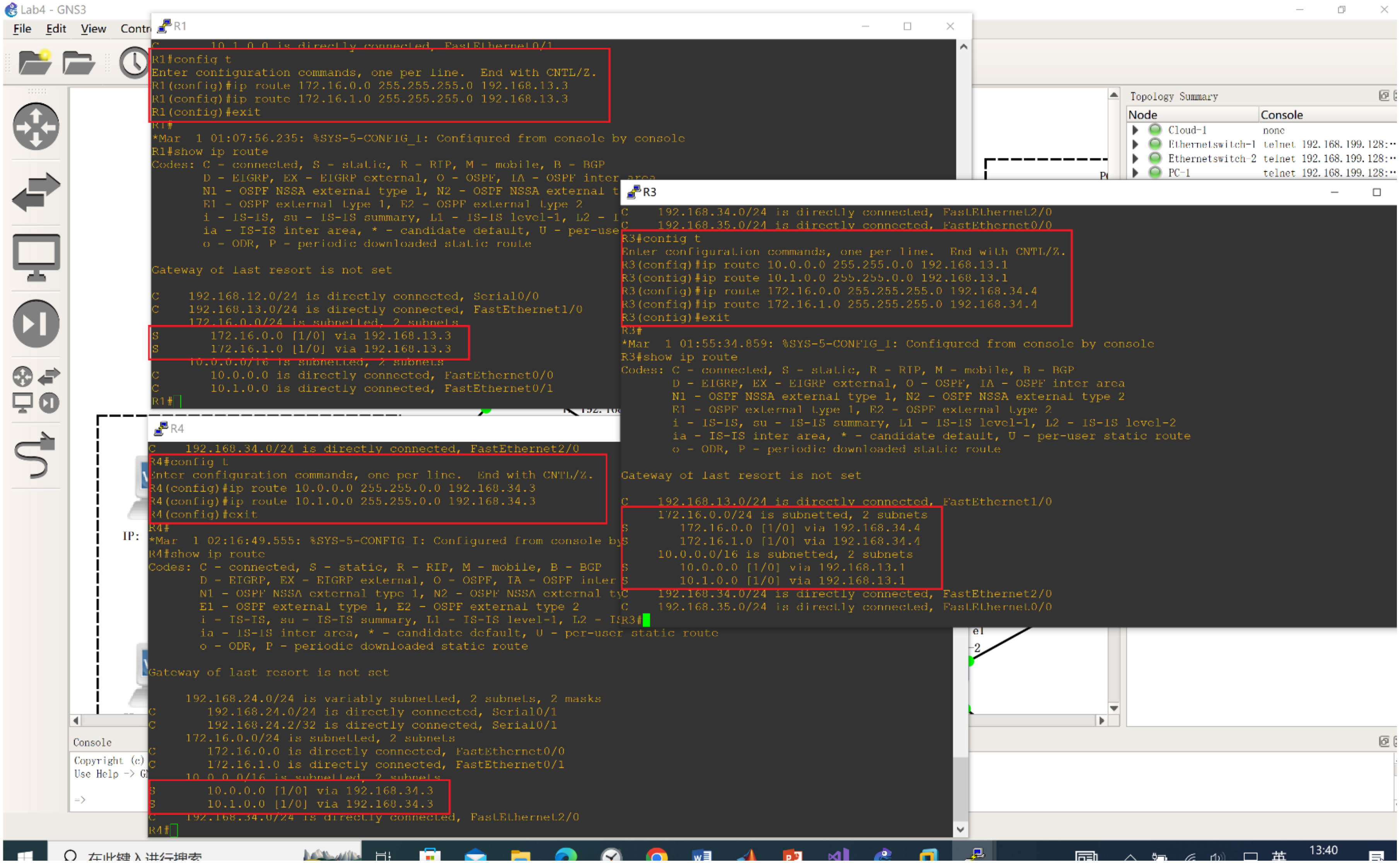
Node	Console
Cloud-1	none
Ethernetswitch-1	telnet 192.168.199.128:…
Ethernetswitch-2	telnet 192.168.199.128:…
PC-1	telnet 192.168.199.128:…
PC-2	telnet 192.168.199.128:…
PC-3	telnet 192.168.199.128:…
PC-4	telnet 192.168.199.128:…
PC-5	telnet 192.168.199.128:…
PC-6	telnet 192.168.199.128:…
PC-7	telnet 192.168.199.128:…
R1	telnet 192.168.199.128:…
R2	telnet 192.168.199.128:…
R3	telnet 192.168.199.128:…
R4	telnet 192.168.199.128:…
R5	telnet 192.168.199.128:…

Servers Summary

```
DESKTOP-EODNB7E CPU 7.4%, RAM 43.3%
GNS3 VM (GNS3 VM) CPU 2.1%, RAM 81.6%
```

```
Copyright (c) 2006-2023 GNS3 Technologies.
Use Help -> GNS3 Doctor to detect common issues.

=>
```

Static Routing Example: **Lab4**



R1:

```
C       10.1.0.0 is directly connected, FastEthernet0/1
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#ip route 172.16.0.0 255.255.255.0 192.168.13.3
R1(config)#ip route 172.16.1.0 255.255.255.0 192.168.13.3
R1(config)#exit
R1#
*Mar  1 01:07:56.235: %SYS-5-CONFIG_I: Configured from console by console
R1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external t
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - I
       ia - IS-IS inter area, * - candidate default, U - per-use
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

C    192.168.12.0/24 is directly connected, Serial0/0
C    192.168.13.0/24 is directly connected, FastEthernet1/0
     172.16.0.0/24 is subnetted, 2 subnets
S       172.16.0.0 [1/0] via 192.168.13.3
S       172.16.1.0 [1/0] via 192.168.13.3
     10.0.0.0/16 is subnetted, 2 subnets
C       10.0.0.0 is directly connected, FastEthernet0/0
C       10.1.0.0 is directly connected, FastEthernet0/1
R1#
```

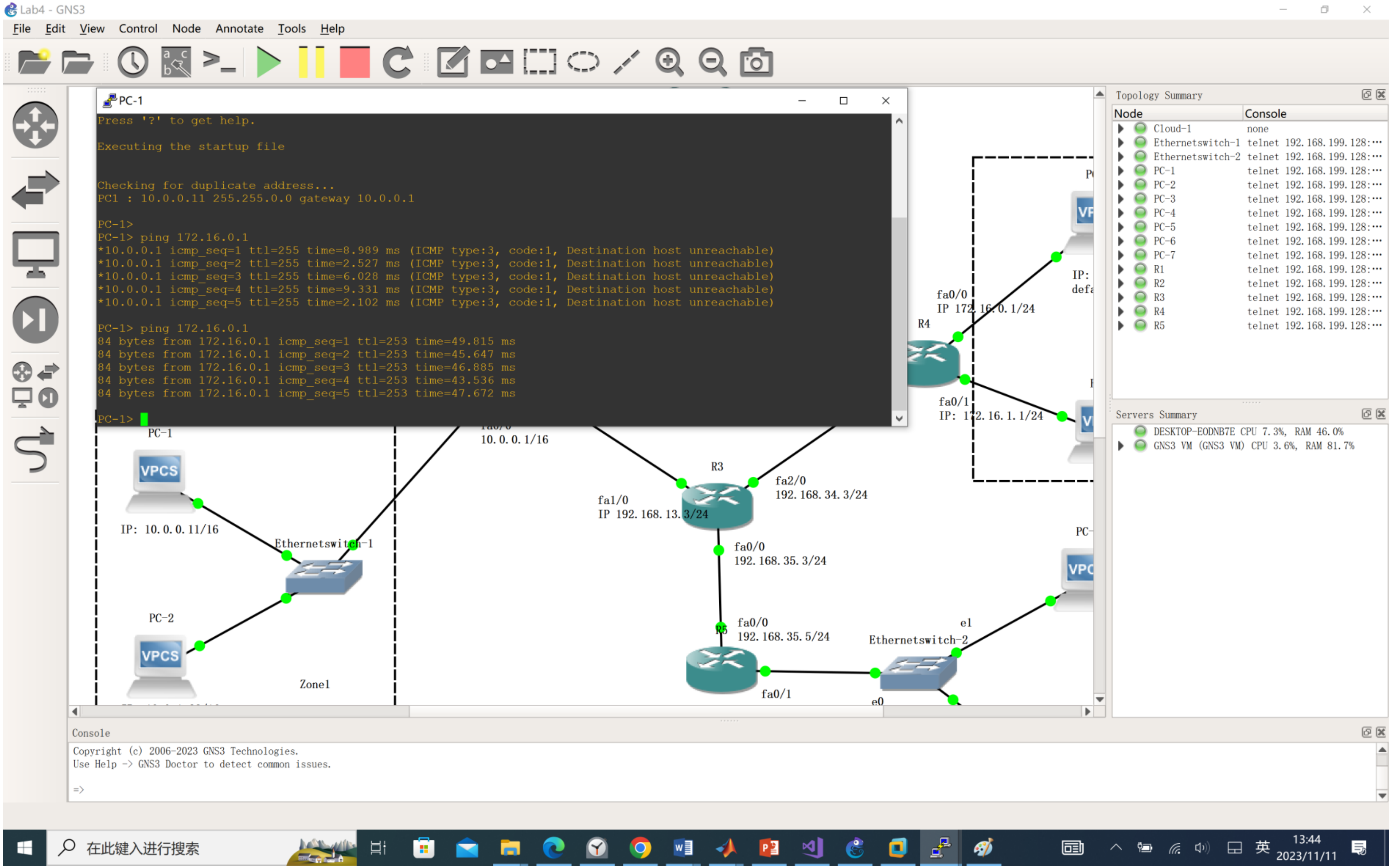
R3:

```
C    192.168.34.0/24 is directly connected, FastEthernet2/0
C    192.168.35.0/24 is directly connected, FastEthernet0/0
R3#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#ip route 10.0.0.0 255.255.0.0 192.168.13.1
R3(config)#ip route 10.1.0.0 255.255.0.0 192.168.13.1
R3(config)#ip route 172.16.0.0 255.255.255.0 192.168.34.4
R3(config)#ip route 172.16.1.0 255.255.255.0 192.168.34.4
R3(config)#exit
R3#
*Mar  1 01:55:34.859: %SYS-5-CONFIG_I: Configured from console by console
R3#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

C    192.168.13.0/24 is directly connected, FastEthernet1/0
     172.16.0.0/24 is subnetted, 2 subnets
S       172.16.0.0 [1/0] via 192.168.34.4
S       172.16.1.0 [1/0] via 192.168.34.4
     10.0.0.0/16 is subnetted, 2 subnets
S       10.0.0.0 [1/0] via 192.168.13.1
S       10.1.0.0 [1/0] via 192.168.13.1
C    192.168.34.0/24 is directly connected, FastEthernet2/0
C    192.168.35.0/24 is directly connected, FastEthernet0/0
R3#
```

R4:

```
C    192.168.34.0/24 is directly connected, FastEthernet2/0
R4#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R4(config)#ip route 10.0.0.0 255.255.0.0 192.168.34.3
R4(config)#ip route 10.1.0.0 255.255.0.0 192.168.34.3
R4(config)#exit
R4#
*Mar  1 02:16:49.555: %SYS-5-CONFIG_I: Configured from console by
R4#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external ty
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     192.168.24.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.24.0/24 is directly connected, Serial0/1
C       192.168.24.2/32 is directly connected, Serial0/1
     172.16.0.0/24 is subnetted, 2 subnets
C       172.16.0.0 is directly connected, FastEthernet0/0
C       172.16.1.0 is directly connected, FastEthernet0/1
     10.0.0.0/16 is subnetted, 2 subnets
S       10.0.0.0 [1/0] via 192.168.34.3
S       10.1.0.0 [1/0] via 192.168.34.3
C    192.168.34.0/24 is directly connected, FastEthernet2/0
R4#
```

Static Routing Example: **Lab4**



```
Press '?' to get help.

Executing the startup file


Checking for duplicate address...
PC1 : 10.0.0.11 255.255.0.0 gateway 10.0.0.1

PC-1>
PC-1> ping 172.16.0.1
*10.0.0.1 icmp_seq=1 ttl=255 time=8.989 ms (ICMP type:3, code:1, Destination host unreachable)
*10.0.0.1 icmp_seq=2 ttl=255 time=2.527 ms (ICMP type:3, code:1, Destination host unreachable)
*10.0.0.1 icmp_seq=3 ttl=255 time=6.028 ms (ICMP type:3, code:1, Destination host unreachable)
*10.0.0.1 icmp_seq=4 ttl=255 time=9.331 ms (ICMP type:3, code:1, Destination host unreachable)
*10.0.0.1 icmp_seq=5 ttl=255 time=2.102 ms (ICMP type:3, code:1, Destination host unreachable)

PC-1> ping 172.16.0.1
84 bytes from 172.16.0.1 icmp_seq=1 ttl=253 time=49.815 ms
84 bytes from 172.16.0.1 icmp_seq=2 ttl=253 time=45.647 ms
84 bytes from 172.16.0.1 icmp_seq=3 ttl=253 time=46.885 ms
84 bytes from 172.16.0.1 icmp_seq=4 ttl=253 time=43.536 ms
84 bytes from 172.16.0.1 icmp_seq=5 ttl=253 time=47.672 ms

PC-1>
```

Topology Summary

Node	Console
Cloud-1	none
Ethernetswitch-1	telnet 192.168.199.128:…
Ethernetswitch-2	telnet 192.168.199.128:…
PC-1	telnet 192.168.199.128:…
PC-2	telnet 192.168.199.128:…
PC-3	telnet 192.168.199.128:…
PC-4	telnet 192.168.199.128:…
PC-5	telnet 192.168.199.128:…
PC-6	telnet 192.168.199.128:…
PC-7	telnet 192.168.199.128:…
R1	telnet 192.168.199.128:…
R2	telnet 192.168.199.128:…
R3	telnet 192.168.199.128:…
R4	telnet 192.168.199.128:…
R5	telnet 192.168.199.128:…

Servers Summary

DESKTOP-EODNB7E CPU 7.3%, RAM 46.0%
GNS3 VM (GNS3 VM) CPU 3.6%, RAM 81.7%

Console

```
Copyright (c) 2006-2023 GNS3 Technologies.
Use Help -> GNS3 Doctor to detect common issues.

=>
```

The Link-State (LS) Routing Algorithm

- In a link-state algorithm, the network topology and all link costs are known, that is, available as input to the LS algorithm.
  - In practice, this is accomplished by having each node **broadcast** link state packets to all other nodes in the network, with each link-state packet containing the identities and costs of its attached links.
    - For example, the Internet's **OSPF** routing protocol is accomplished by a link-state broadcast algorithm.
- All nodes have an identical and complete view of the network. Each node can then run the LS algorithm and compute the same set of least-cost paths as every other node.
  - The well-known LS algorithm is **Dijkstra's algorithm**.
    - Dijkstra's algorithm computes the least-cost path from one node (the source) to all other nodes in the network.

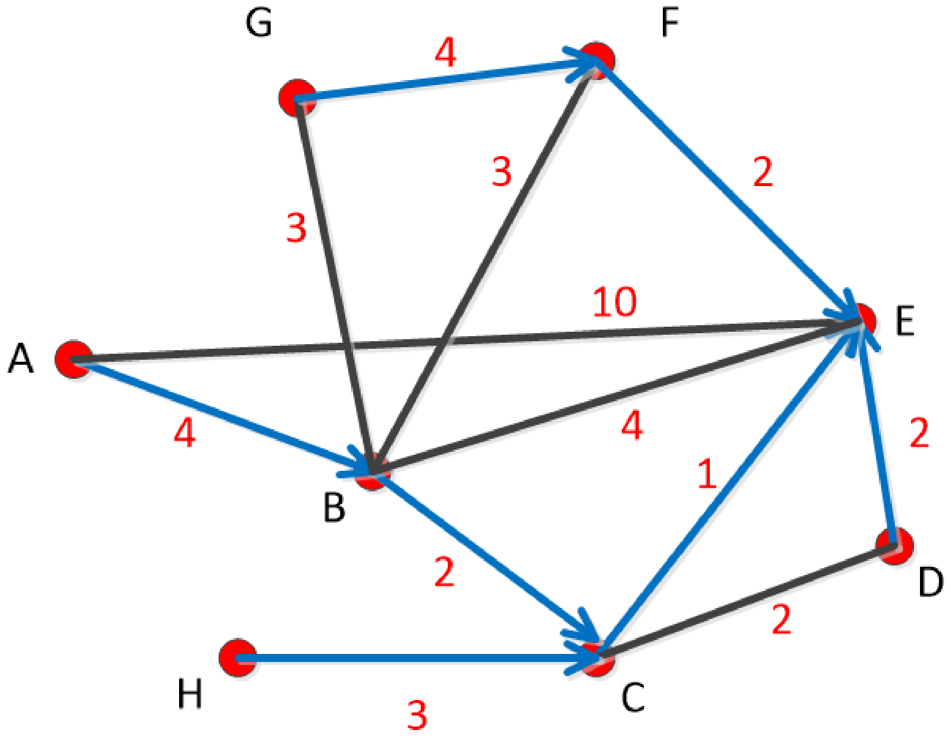
The Shortest Path Algorithm

- The shortest path is one that has the least cost.
- Measure path cost (length)
  - Number of hops
  - Delay
  - distance
  - Bandwidth
  - Communication cost
  - Average traffic
- The optimality principle
  - If router *J* is on the optimal path from router *I* to router *K*, then the optimal path from *J* to *K* also falls along the same route.

Sink Tree (I)

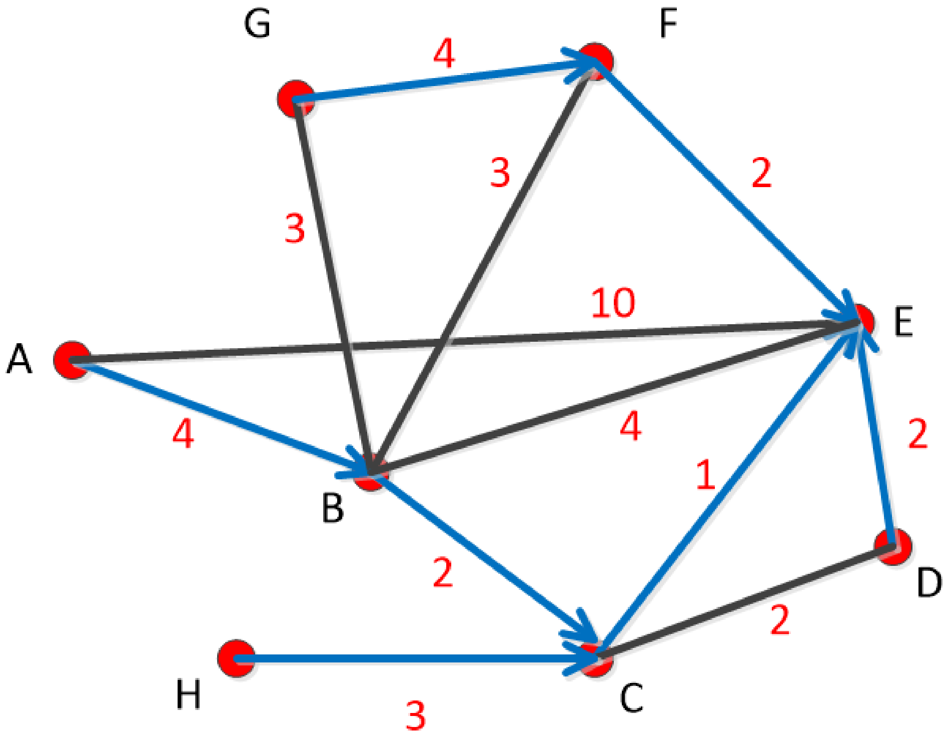
- Sink tree for a **destination** is the union of all shortest paths towards the destination
  - Similarly source tree

- Find a sink tree for E.
  - A→B→C→E
  - B→C→E
  - C→E
  - H→C→E
  - D→E
  - F→E
  - G→F→E (G→B→C→E)



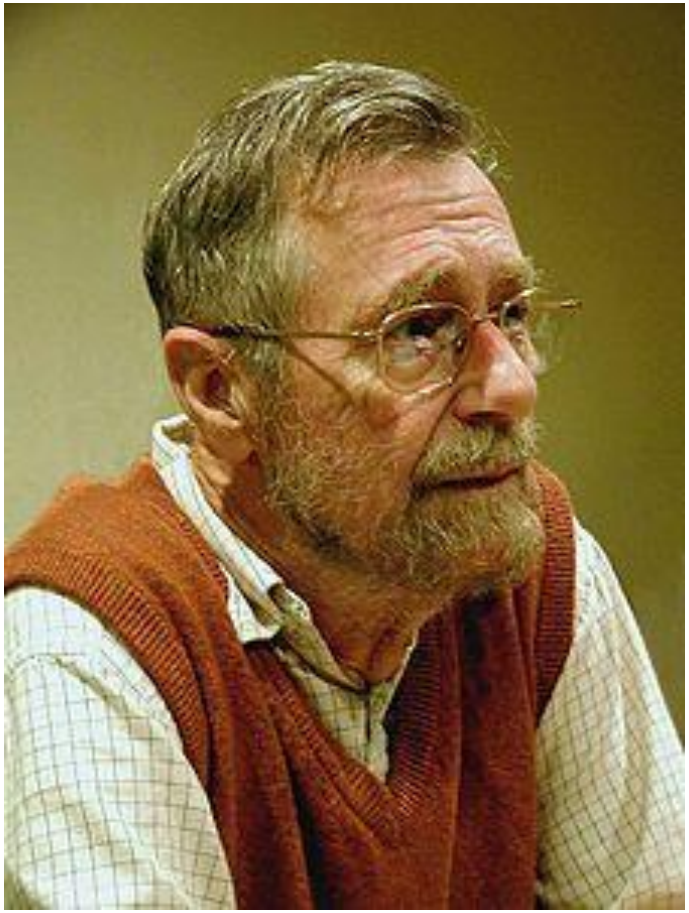
Sink Tree (II)

- Implications:
  - Only need to use destination to follow shortest paths
  - Each node only need to send to the next hop
- Forwarding table at a node
  - List next hop for each destination



Edsger Wybe Dijkstra [2]

- Received the 1972 A. M. Turing Award.
- The Schlumberger Centennial Chair of Computer Sciences at the University of Texas at Austin from 1984 until 2000.
- Made a strong case against use of the GOTO statement in programming languages and helped lead to its deprecation.



May 11, 1930 – Aug. 6, 2002

Dijkstra Algorithm (1959) [2]

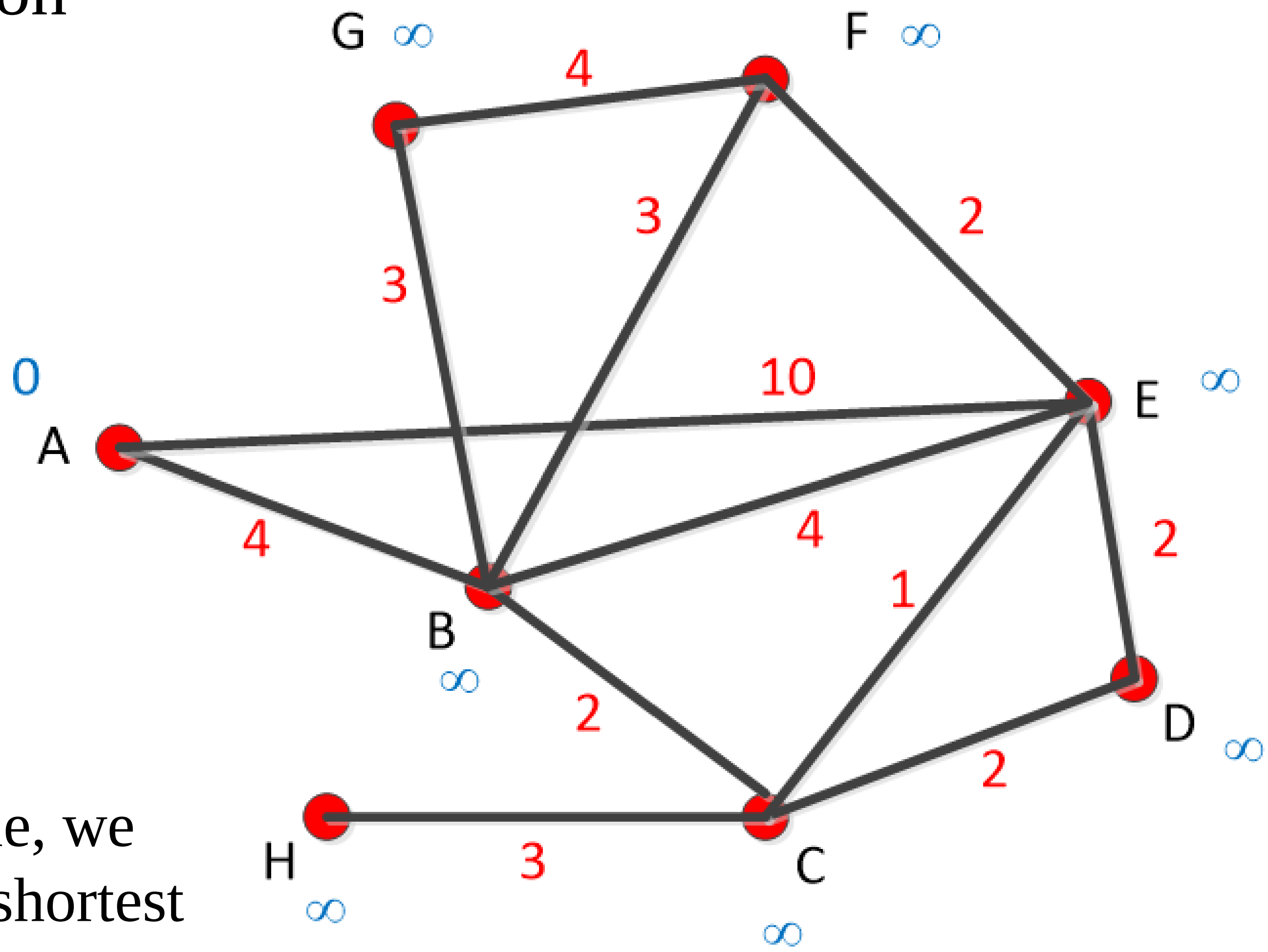
- Single source shortest path problem — The problem of finding shortest paths from a source vertex ***s*** to all other vertices in the graph.
- Dijkstra's algorithm is a solution to the single source shortest path problem in graph theory.
  - Works on both directed and undirected graphs. However, all edges must have *nonnegative* weights.
  - Approach: **Greedy**.
  - Input: weighted graph G = {V, E} and source vertex ***s*** ∈ V, such that all edge weights are nonnegative.
  - Output: lengths of shortest paths (or the shortest paths themselves) from a given source vertex ***s*** ∈ V to all other vertices.

Dijkstra Algorithm (1959) [2]

- Algorithm:
  - Mark all nodes tentative, set distances from source to 0 for source, and ∞ (infinity) for all other nodes.
  - While tentative nodes remain:
    - Extract *n*, a node with lowest distance
    - Add link to *n* to the shortest path tree
    - Relax (or updating) the distances of neighbors of *n* by lowering any better distance estimates.

Dijkstra Algorithm (1959) [2] (I)

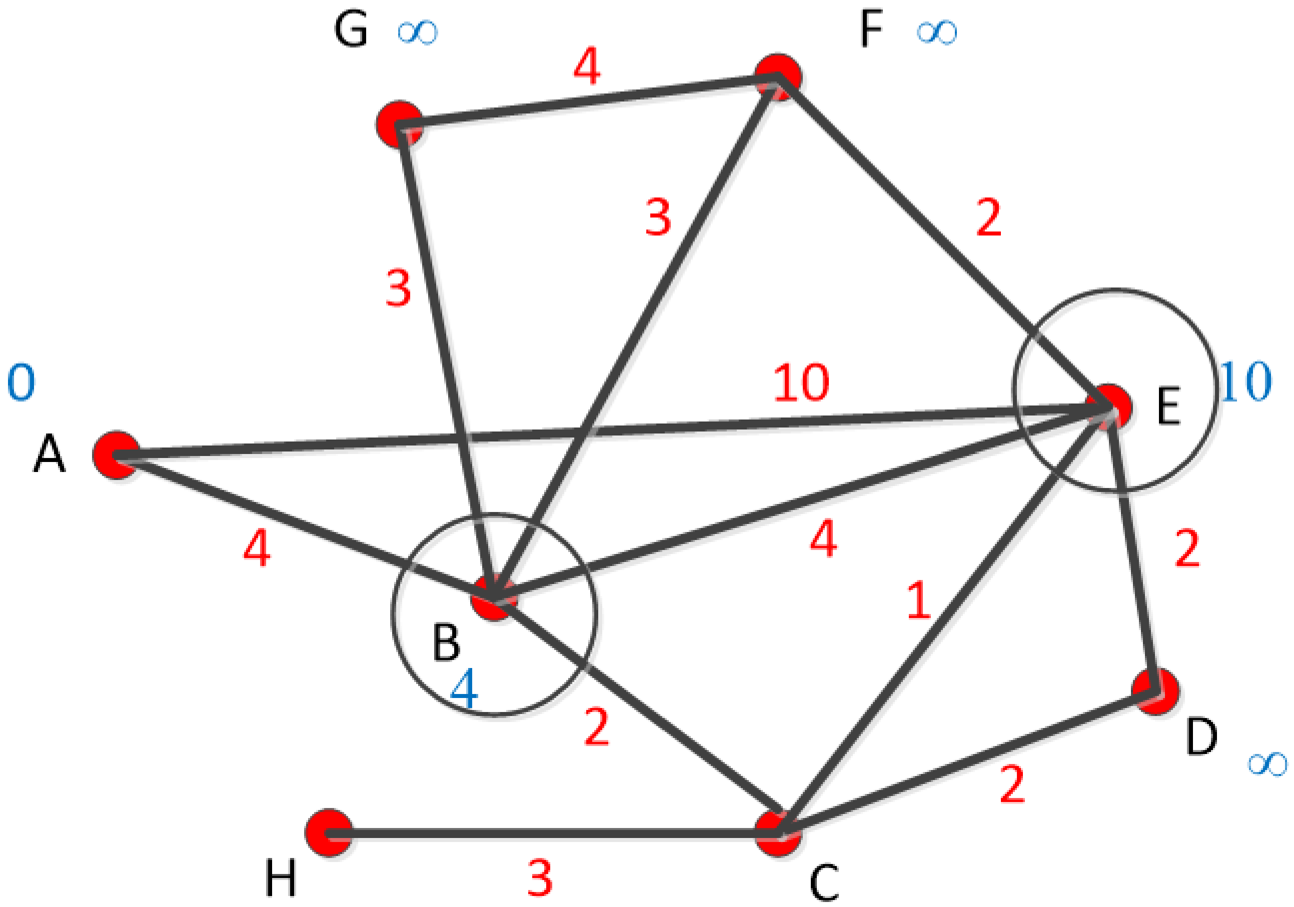
- Initialization



In this example, we will compute shortest paths **from** A

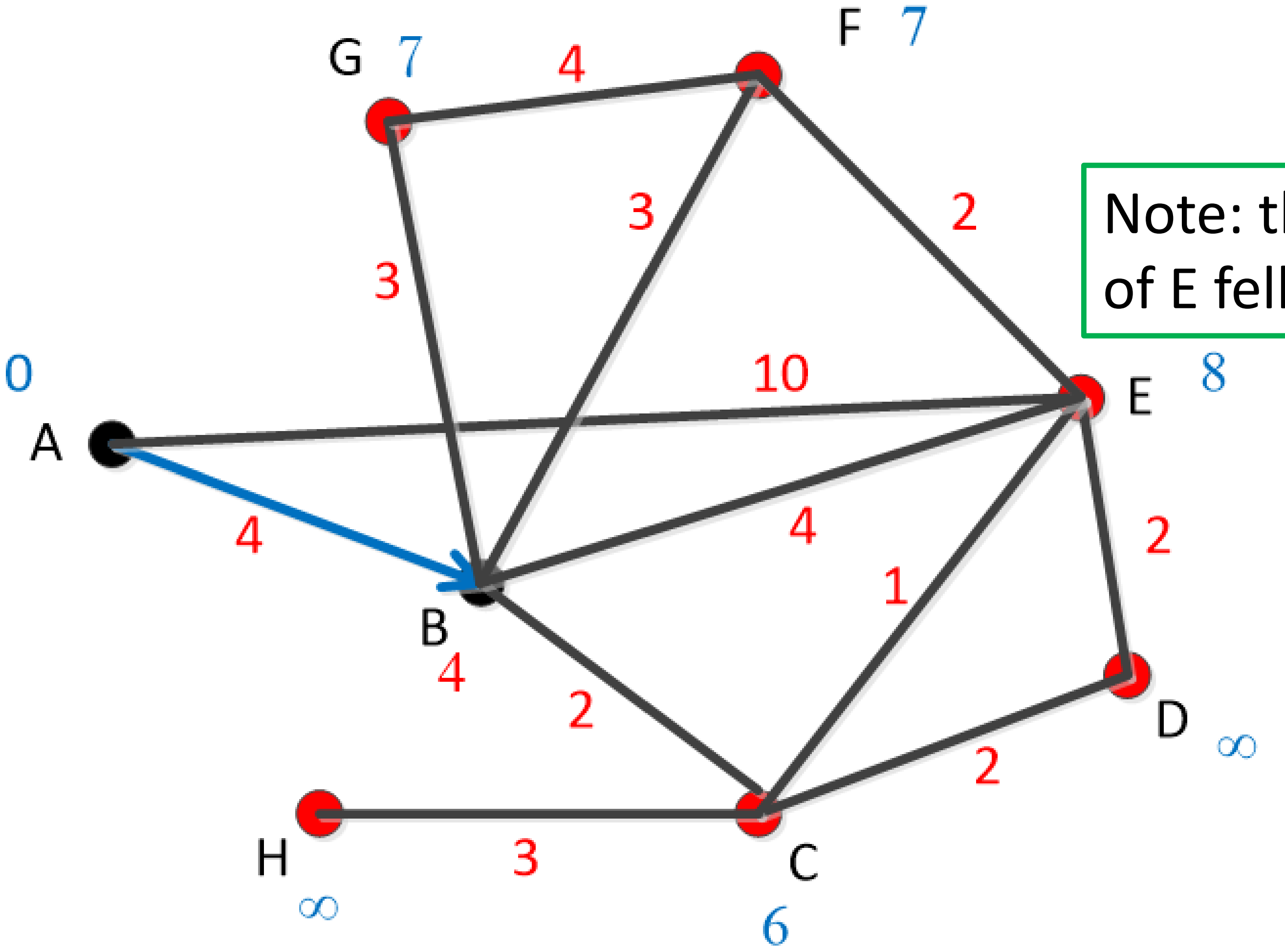
Dijkstra Algorithm (1959) [2] (II)

- Relax around A: B and E are neighbors of A



Dijkstra Algorithm (1959) [2] (III)

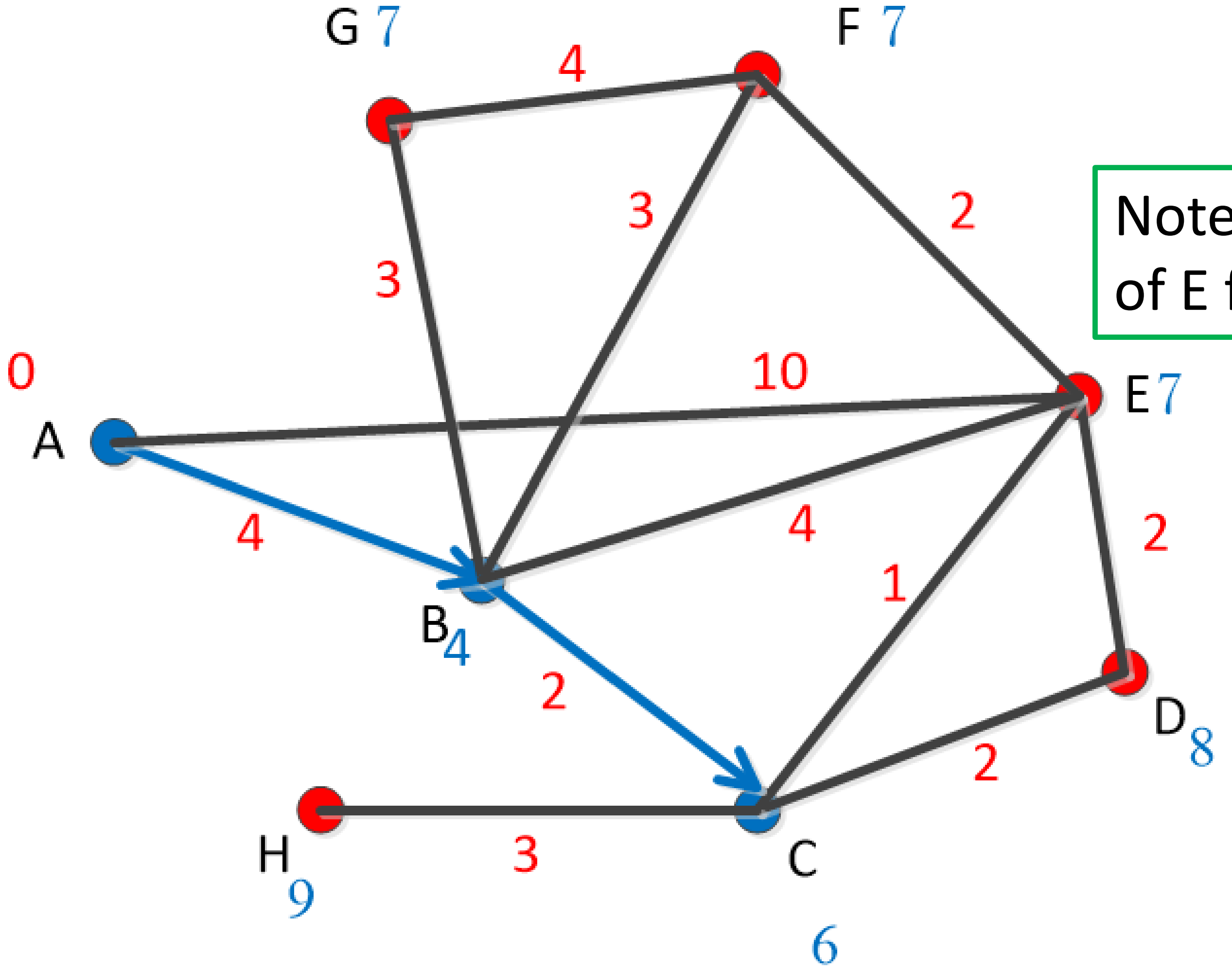
- Relax around B: C, E, F, and G are neighbors of B



Note: the distance of E fell from 10 to 8!

Dijkstra Algorithm (1959) [2] (IV)

- Relax around C: D, E and H are neighbors of C

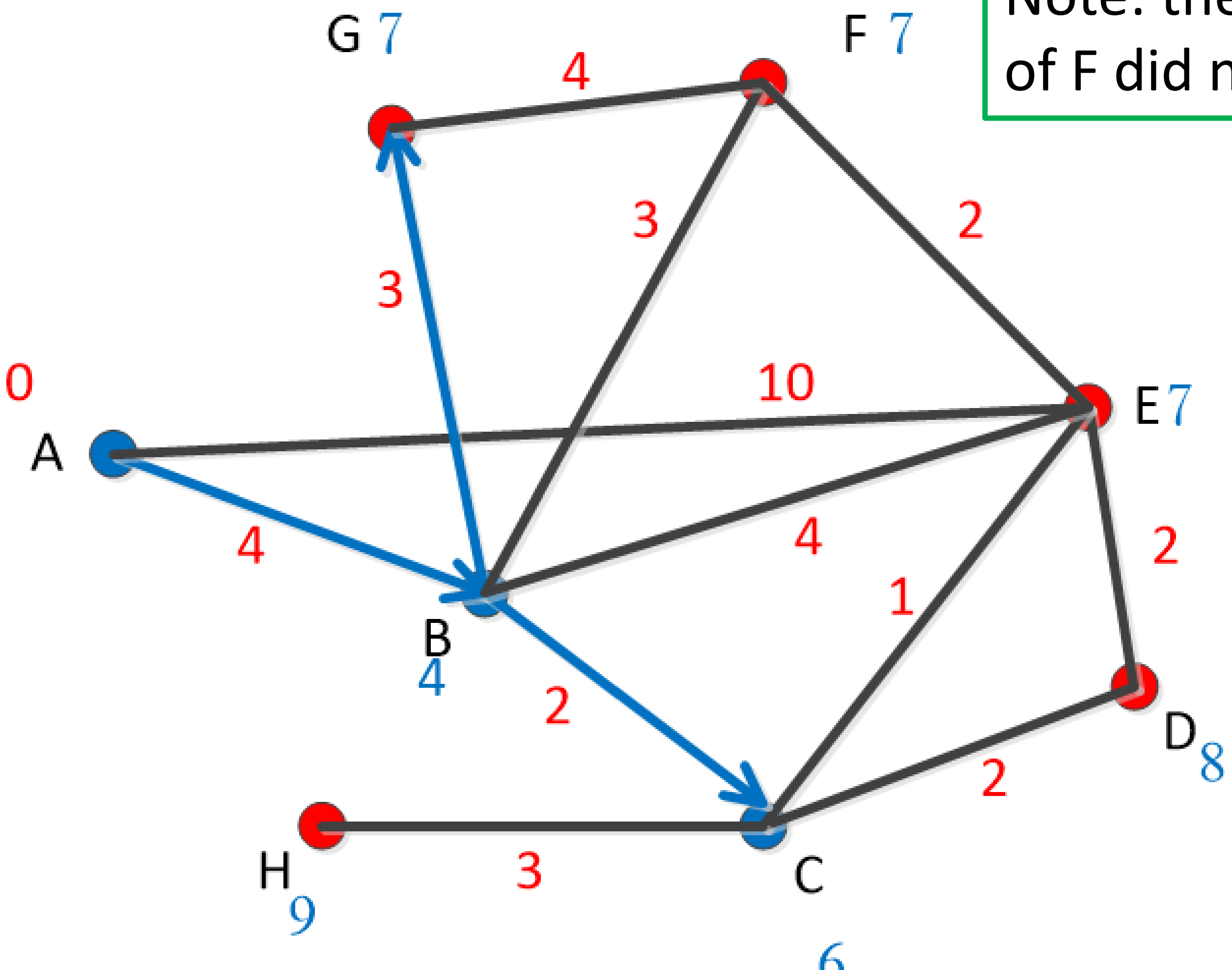


Note: the distance of E fell from 8 to 7!

Dijkstra Algorithm (1959) [2] (V)

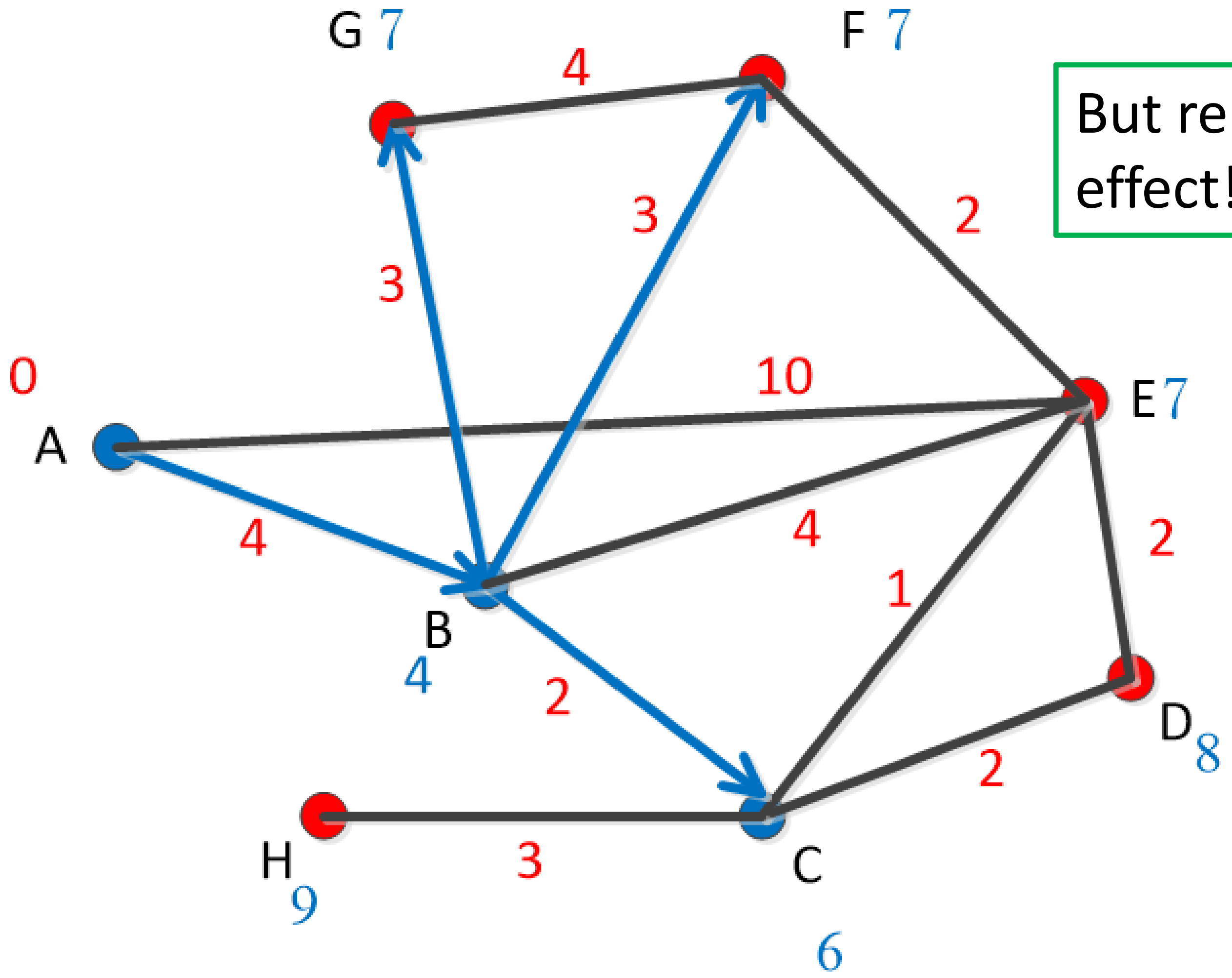
- Relax around G: F is neighbor of G

Note: the distance of F did not fall!



Dijkstra Algorithm (1959) [2] (VI)

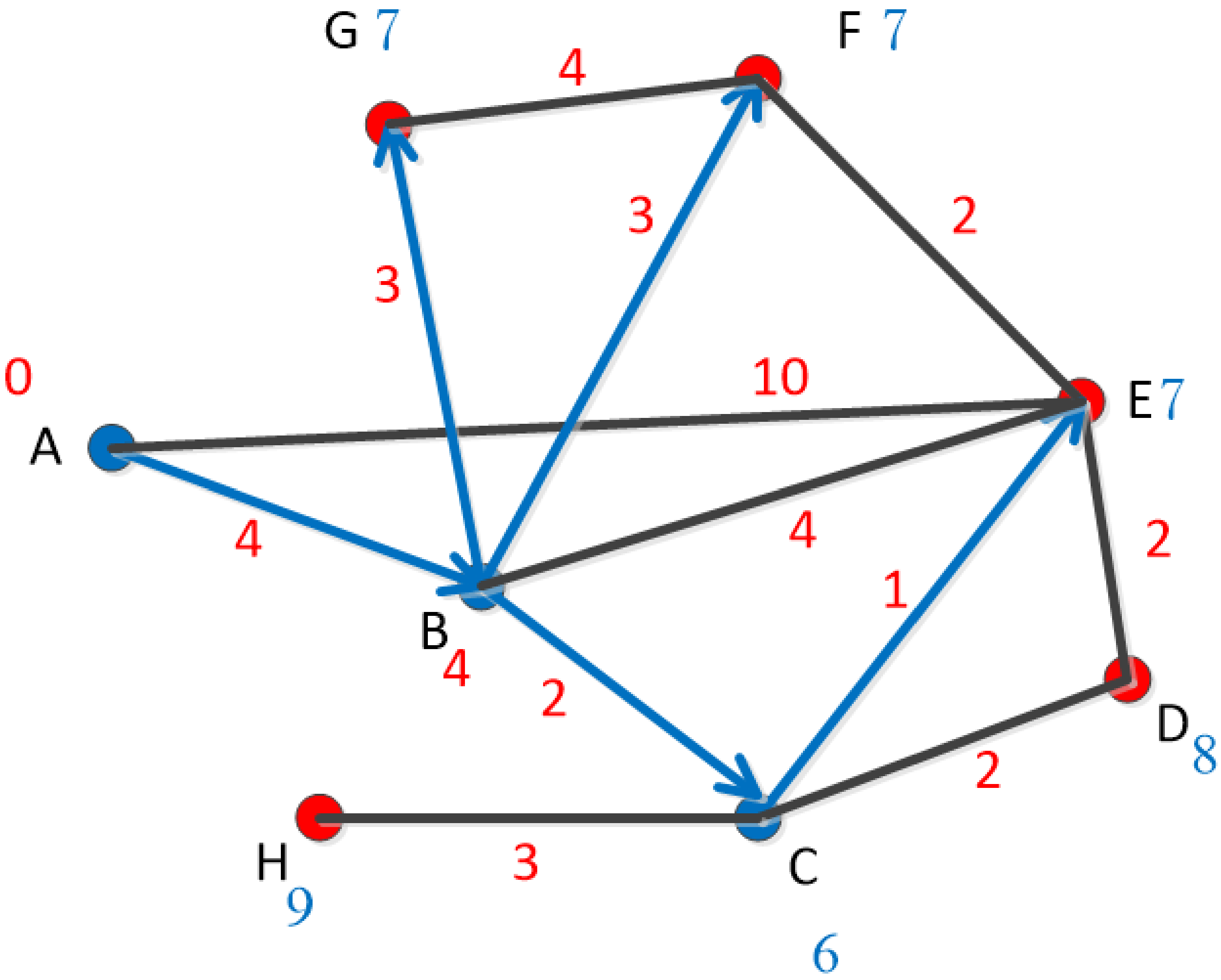
- Relax around F: G and E are neighbors of F



But relax has no effect!

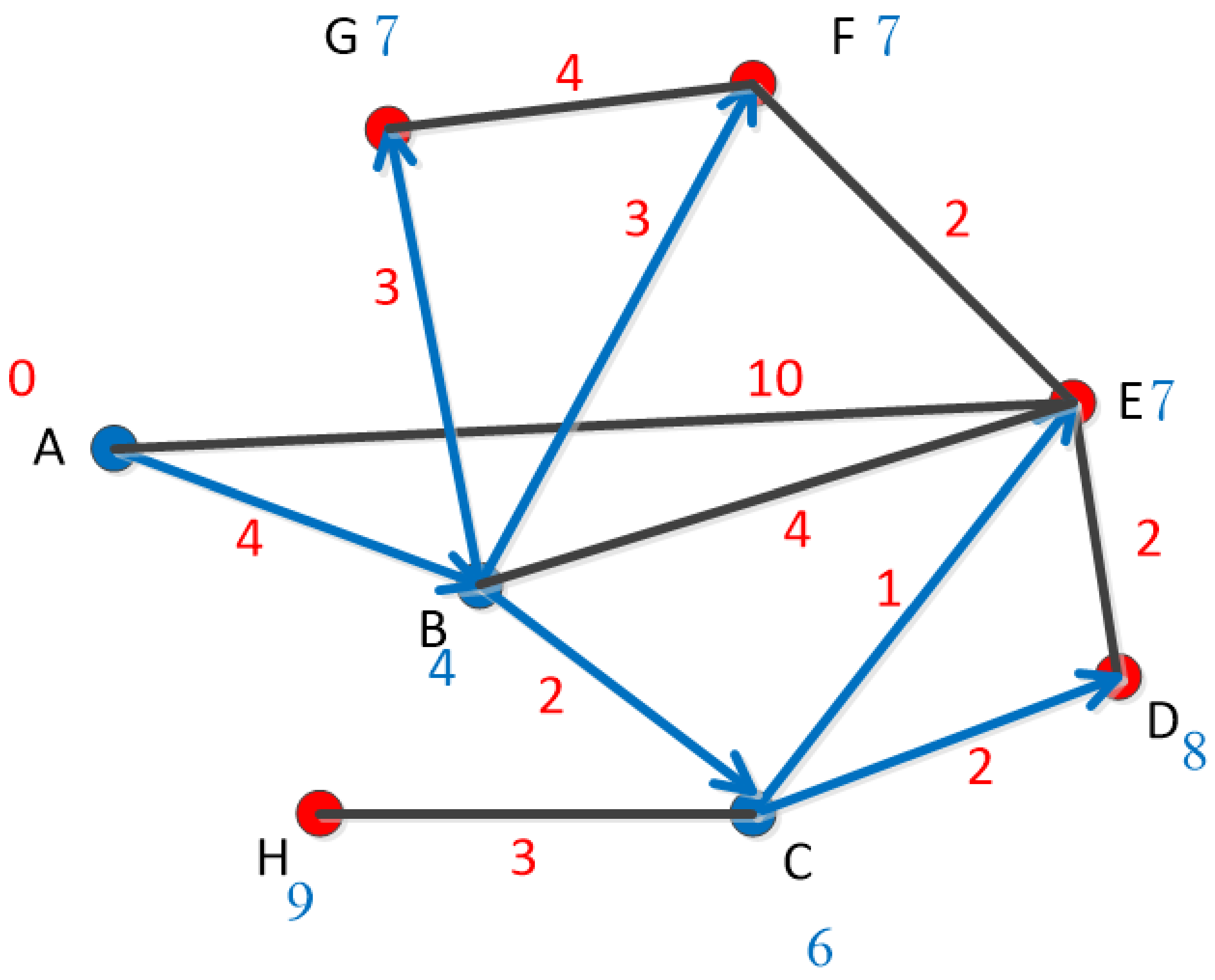
Dijkstra Algorithm (1959) [2] (VII)

- Relax around E



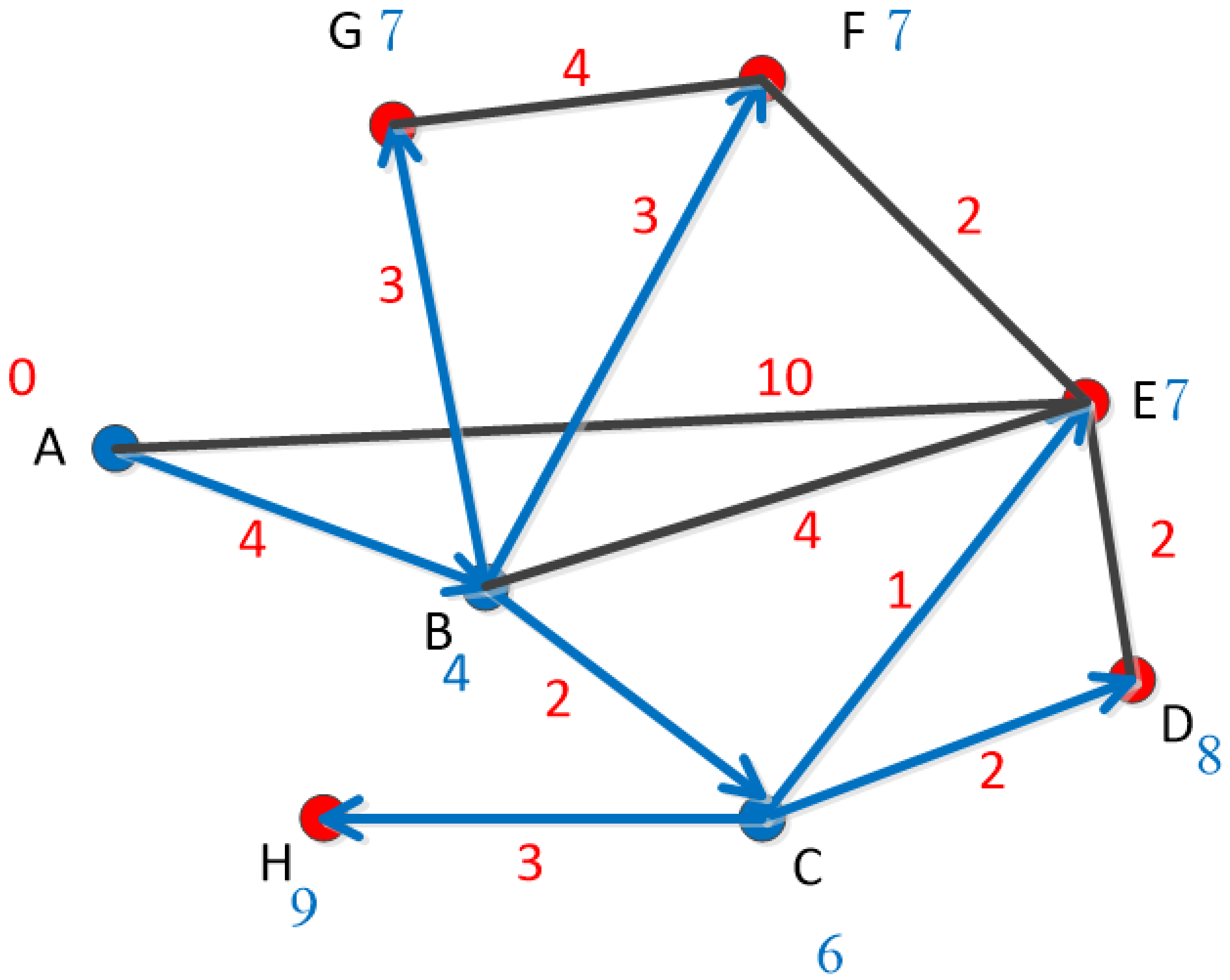
Dijkstra Algorithm (1959) [2] (VIII)

- Relax around D

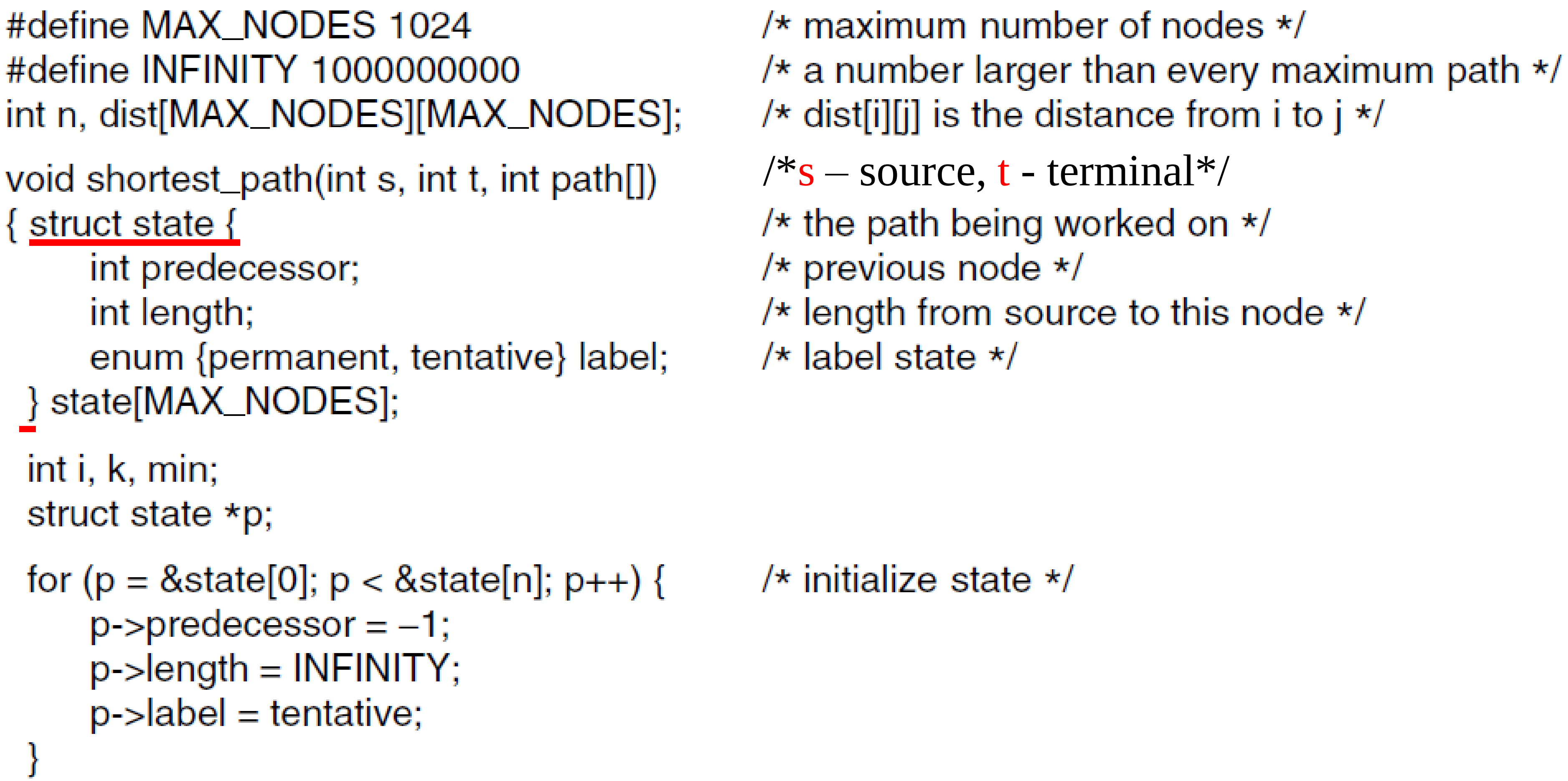


Dijkstra Algorithm (1959) [2] (IX)

- Finally, relax around H

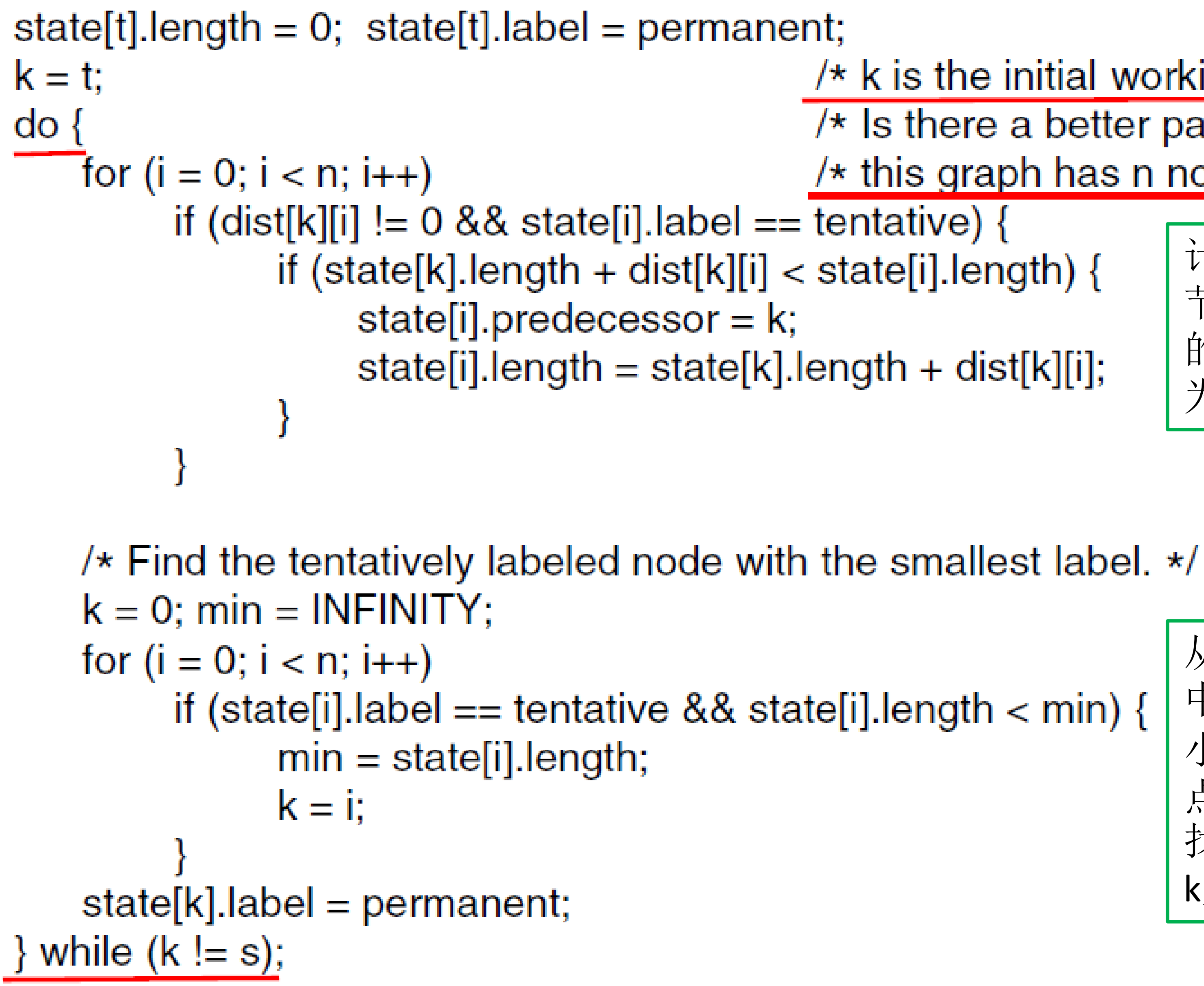


Dijkstra Algorithm (1959) [2] (X)



```
#define MAX_NODES 1024                       /* maximum number of nodes */
#define INFINITY 1000000000                  /* a number larger than every maximum path */
int n, dist[MAX_NODES][MAX_NODES];           /* dist[i][j] is the distance from i to j */

void shortest_path(int s, int t, int path[]) /*s – source, t - terminal*/
{ struct state {                             /* the path being worked on */
      int predecessor;                       /* previous node */
      int length;                            /* length from source to this node */
      enum {permanent, tentative} label;     /* label state */
 } state[MAX_NODES];

 int i, k, min;
 struct state *p;

 for (p = &state[0]; p < &state[n]; p++) {   /* initialize state */
      p->predecessor = –1;
      p->length = INFINITY;
      p->label = tentative;
 }
```

Dijkstra Algorithm (1959) [2] (XI)



```
state[t].length = 0;  state[t].label = permanent;
k = t;                                        /* k is the initial working node */
do {                                          /* Is there a better path from k? */
    for (i = 0; i < n; i++)                   /* this graph has n nodes */
         if (dist[k][i] != 0 && state[i].label == tentative) {
              if (state[k].length + dist[k][i] < state[i].length) {
                   state[i].predecessor = k;
                   state[i].length = state[k].length + dist[k][i];
              }
         }

    /* Find the tentatively labeled node with the smallest label. */
    k = 0; min = INFINITY;
    for (i = 0; i < n; i++)
         if (state[i].label == tentative && state[i].length < min) {
              min = state[i].length;
              k = i;
         }
    state[k].label = permanent;
} while (k != s);
```

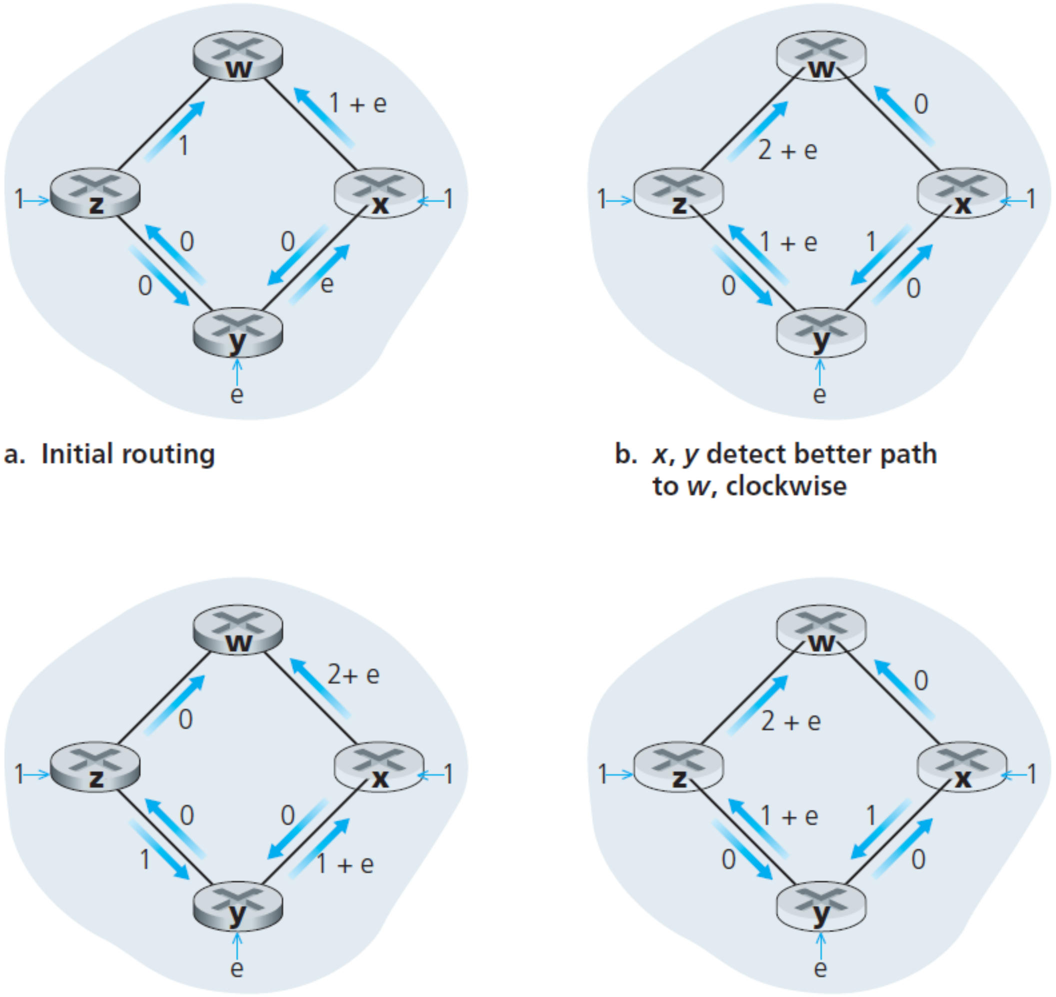
计算凡是与k相邻节点到节点k之间的距离。k初始化为terminal。

从与k相邻的节点中选取到节点k最小距离的那个节点作为下一次寻找的起点k，直至k为源节点。

The Complexity of Dijkstra Algorithm

- In the 1st iteration, we need to search through all *n* nodes to determine the node that has the minimum cost.
- In the 2nd iteration, we need to check $n - 1$ nodes to determine the minimum cost.
- In the 3rd iteration, we need to check $n - 2$ nodes
- …
- The total number of nodes we need to search through over all the iterations is $(n+1)n/2$.
- The worst-case complexity is $O(n^2)$.

The Oscillation Problem with the LS Algorithm



a. Initial routing

b. *x*, *y* detect better path to *w*, clockwise

c. *x*, *y*, *z* detect better path to *w*, counterclockwise

d. *x*, *y*, *z*, detect better path to *w*, clockwise

The Distance Vector (DV) Routing Algorithm [8]

- The distance vector routing algorithm is *iterative*, *asynchronous*, and *distributed*.
  - **Distributed** — each node receives some information from one or more of its directly neighbors, performs a calculation, and then distributes the results of its calculation back to neighbors.
  - **Iterative** — this process continues on until no more information is exchanged between neighbors. It is self-terminating.
  - **Asynchronous** — it does not require all of the nodes to operate in lockstep with each other.
- DV-like algorithms are used in many routing protocols in practice, including the Internet's **RIP**, **BGP**, and so on.

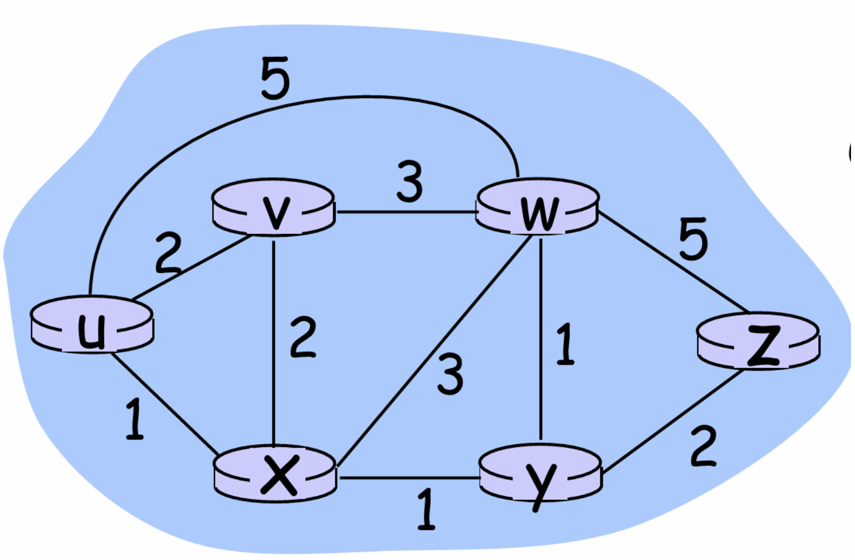
Bellman-Ford Equation

- Let $d_x(y)$ be *the cost of the least-cost path* from node *x* to node *y*. Then the least costs are related by the celebrated Bellman-Ford equation, namely,

$$\boxed{d_x(y) = \min_v\{c(x,v) + d_v(y)\}}$$

  - where the $\min_v$ in the equation is taken over all of *x*'s neighbors.

Bellman-Ford Example



- Clearly, $d_v(z) = 5$, $d_x(z) = 3$, $d_w(z) = 3$
- Bellman-Ford Equation says:

$$\begin{aligned} d_u(z) = \min\{ & c(u,x) + d_x(z), \\ & c(u,v) + d_v(z), \\ & c(u,w) + d_w(z) \} \\ = \min\{ & 1 + 3, \\ & 2 + 5, \\ & 5 + 3 \} \\ = 4 & \end{aligned}$$

The Distance Vector Routing Algorithm

- The basic idea: each node x begins with $D_x(y)$, an *estimate* of the cost of the least-cost path from itself to node y, for all nodes in N (the number of nodes in the network). Let $\boldsymbol{D}_x(y) = [D_\mathrm{x}(y) : y \text{ in } N]$ be node x's distance vector, which is the vector of cost estimates from x to all other nodes, y, in N.
- With the DV algorithm, each node x maintains the following information
  - For each neighbor node v, the cost $c(x, v)$ from x to directly attached neighbor v.
  - Node x's distance vector, that is $\boldsymbol{D}_x(y) = [D_x(y) : y \text{ in } N]$, containing x's estimate of its cost to all destinations, y, in N.
  - The distance vectors of each of its neighbors, that is, $\boldsymbol{D}_v(y) = [D_v(y) : y \text{ in } N]$ for each neighbor v of x.

The Distance Vector Routing Algorithm

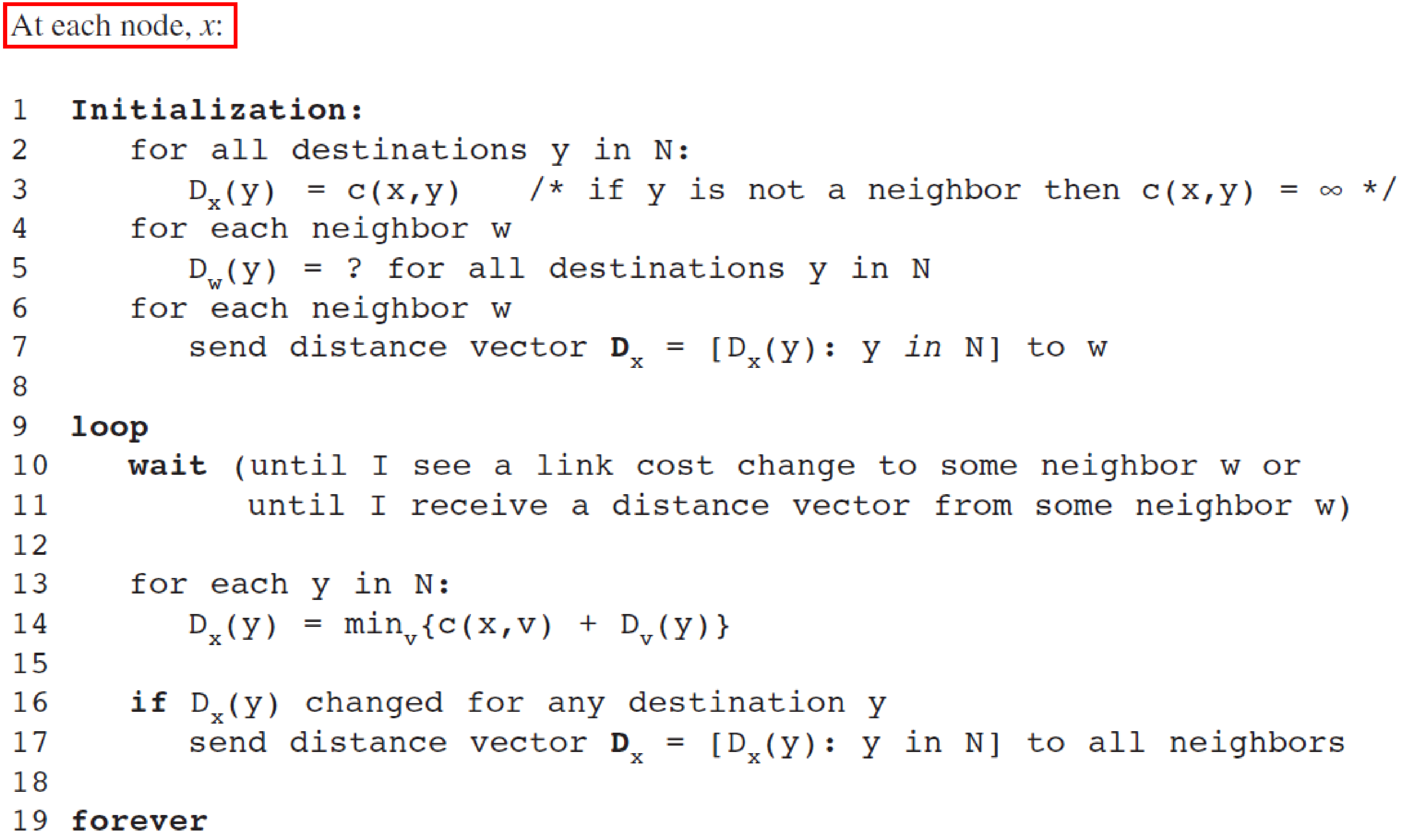
- When a node *x* receives a new distance vector from any of its neighbors *v*, it save *v*'s distance vector, and then use the Bellman-Ford equation to **update** its own distance vector as follows

$$\boxed{D_x(y) = \min_v \{c(x,v) + D_v(y)\}}$$

for each node *y* in *N*

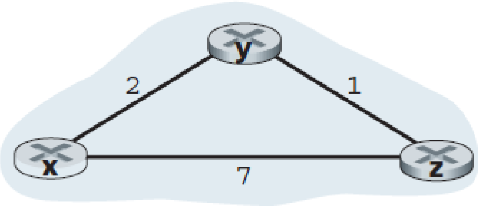
- If node *x*'s distance vector has changed as a result of this update step, node *x* will then send its updated distance vector to each of its neighbors, which can in turn update their own distance vectors.
- As long as all the nodes continue to exchange their distance vectors in an *asynchronous* fashion, each cost estimation $D_x(y)$ will **converges** to $d_x(y)$.

The Distance Vector Routing Algorithm



At each node, *x*:

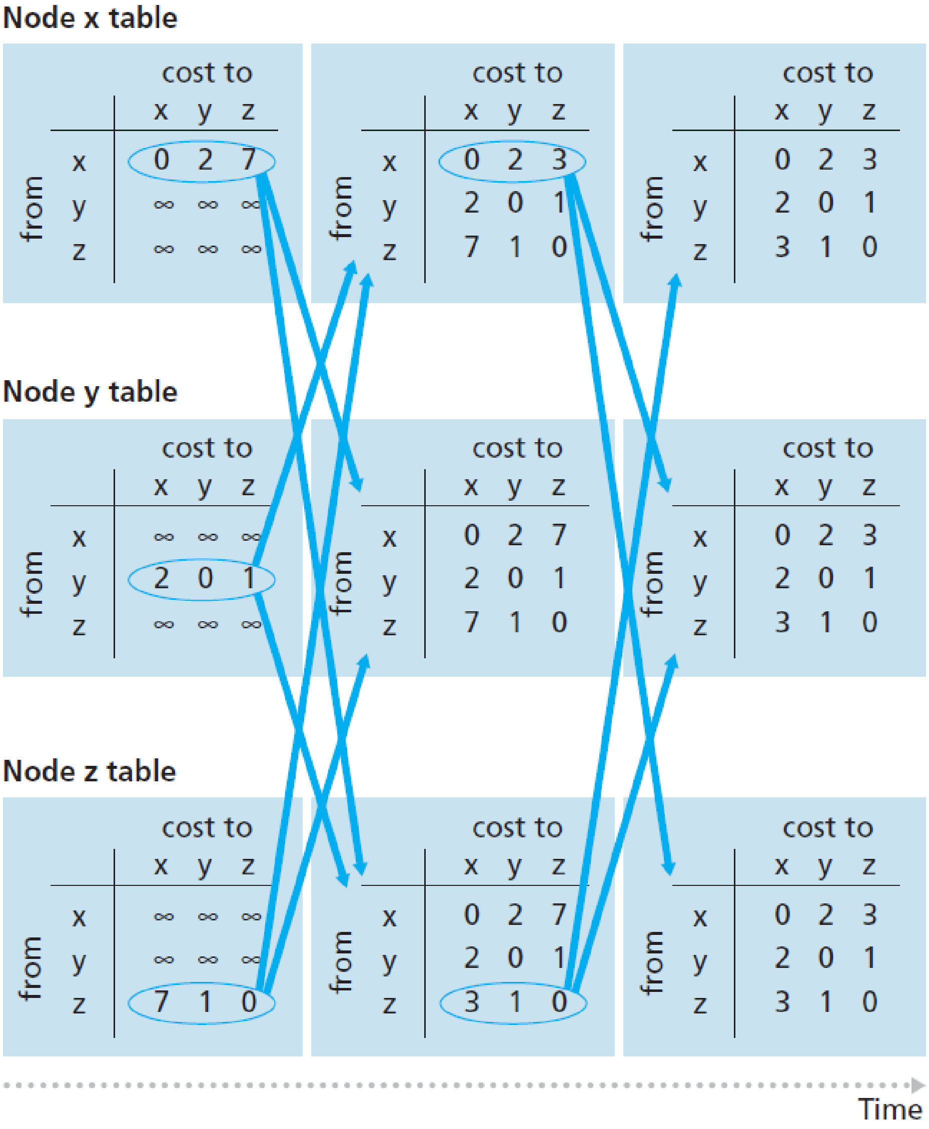
```
Initialization:
   for all destinations y in N:
      $D_x(y) = c(x,y)$   /* if y is not a neighbor then $c(x,y) = \infty$ */
   for each neighbor w
      $D_w(y) = ?$ for all destinations y in N
   for each neighbor w
      send distance vector $\mathbf{D}_x = [D_x(y)$: y in N] to w

loop
   wait (until I see a link cost change to some neighbor w or
         until I receive a distance vector from some neighbor w)

   for each y in N:
      $D_x(y) = \min_v\{c(x,v) + D_v(y)\}$

   if $D_x(y)$ changed for any destination y
      send distance vector $\mathbf{D}_x = [D_x(y)$: y in N] to all neighbors

forever
```



$$D_x(x) = 0$$

$$D_x(y) = \min\{c(x,y) + D_y(y),\ c(x,z) + D_z(y)\} = \min\{2 + 0,\ 7 + 1\} = 2$$

$$D_x(z) = \min\{c(x,y) + D_y(z),\ c(x,z) + D_z(z)\} = \min\{2 + 1,\ 7 + 0\} = 3$$



Node x table

		cost to		
		x	y	z
from	x	0	2	7
	y	∞	∞	∞
	z	∞	∞	∞

		cost to		
		x	y	z
from	x	0	2	3
	y	2	0	1
	z	7	1	0

		cost to		
		x	y	z
from	x	0	2	3
	y	2	0	1
	z	3	1	0

Node y table

		cost to		
		x	y	z
from	x	∞	∞	∞
	y	2	0	1
	z	∞	∞	∞

		cost to		
		x	y	z
from	x	0	2	7
	y	2	0	1
	z	7	1	0

		cost to		
		x	y	z
from	x	0	2	3
	y	2	0	1
	z	3	1	0

Node z table

		cost to		
		x	y	z
from	x	∞	∞	∞
	y	∞	∞	∞
	z	7	1	0

		cost to		
		x	y	z
from	x	0	2	7
	y	2	0	1
	z	3	1	0

		cost to		
		x	y	z
from	x	0	2	3
	y	2	0	1
	z	3	1	0

♠ The process of receiving updated distance vectors from neighbors, re-computing routing table entries, and informing neighbors of changed costs of the least-cost path to a destination continues until no update message are sent.

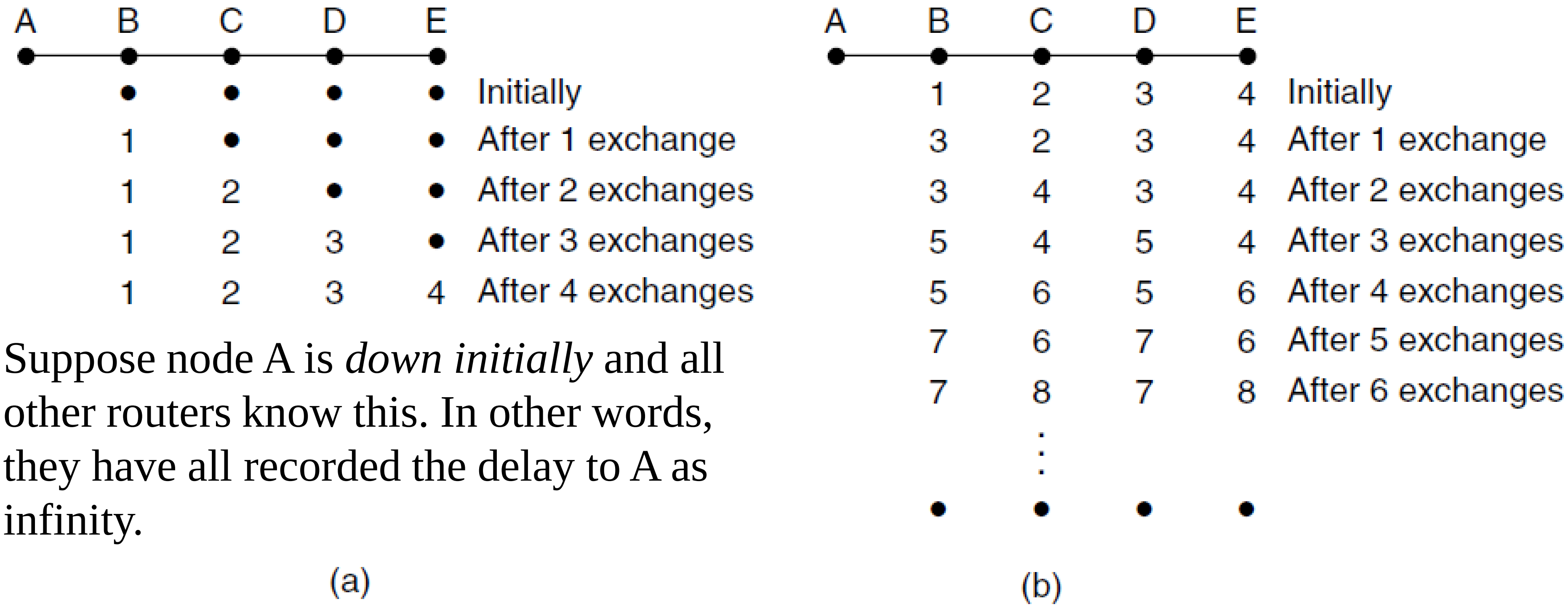
— **A quiescent state**

♠ Although the distance vector routing is a simple technique by which routers can collectively compute shortest paths, it has a serious drawback in practice: it converges to the correct answer, but it may do so slowly.

— **The Count-to-Infinity Problem**

The Count-to-Infinity Problem

- The core of the problem is that when *x* tells *y* that it has a path somewhere, *y* has no way of knowing whether it itself is on the path.



A	B	C	D	E	
	•	•	•	•	Initially
	1	•	•	•	After 1 exchange
	1	2	•	•	After 2 exchanges
	1	2	3	•	After 3 exchanges
	1	2	3	4	After 4 exchanges

Suppose node A is *down initially* and all other routers know this. In other words, they have all recorded the delay to A as infinity.

(a)

A	B	C	D	E	
	1	2	3	4	Initially
	3	2	3	4	After 1 exchange
	3	4	3	4	After 2 exchanges
	5	4	5	4	After 3 exchanges
	5	6	5	6	After 4 exchanges
	7	6	7	6	After 5 exchanges
	7	8	7	8	After 6 exchanges
		⋮			
	•	•	•	•	

(b)

Figure 5-10. The count-to-infinity problem.

In the second row of (b), B does not hear any thing from A, but C says it has a path to B, so now B to A is 3, and B does not know C’s path runs through B itself.

Link State Routing *vs.* Distance Vector Routing

♦ Message Complexity

♥ LS: with *n* nodes, *E* Links, *O*(*nE*) messages sent

♥ DV: exchange between neighbors only

♦ Speed of Convergence

♥ LS: O(n^2) algorithm, requires *O*(*nE*) messages

♠ may have oscillations

♥ DV: convergence time varies

♠ count-to-infinite 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

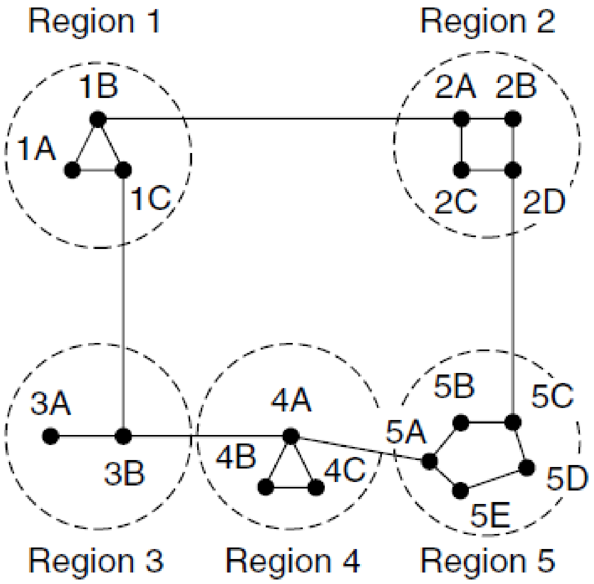
♥ each node's table used by others

♠ errors propagate through network

Hierarchical Routing

- At a certain point, the network may grow to the point where it is no longer feasible for every router to have an entry for every other router.
  - The routing will have to be done **hierarchically**.
- When hierarchical routing is used, the routers are divided into what we will call **regions**.
  - "Autonomous Systems" (AS)
  - Each router knows all the details about how to route packets to destination within its own AS but knows nothing about the internal structure of other ASes.

Hierarchical Routing



(a)

Full table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

(b)

Hierarchical table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

(c)

When routing is done hierarchically, there are entries for all the local routers, but all other ASes are condensed into **a single router**.

Figure 5-14. Hierarchical routing.

Hierarchical Routing

- Routers in same AS run same routing protocol
  - “intra-AS” routing protocol
  - Routers in different AS can different intra-AS routing protocol
- Gateway router
  - Direct link to router in another AS
  - For the previous example, routers B and C are gateway routers.
- When a single network becomes very large, an interesting question is “how many levels should the hierarchy have?”
  - Kamoun and Kleinrock (1979) discovered that the optimal number of levels of an *N* router networks is log*N*.

Broadcast Routing

- In broadcasting routing, the network layer provides a service of delivering a packet sent from a source node to all other nodes in the network.
- 1. Flooding: when node receives broadcast packet, sends copy to all neighbors
  - Problems: cycles & broadcast storm
- 2. Controlled flooding: node only broadcast packet if it hasn’t broadcast the same packet before.
  - Node keeps track of packet ids already broadcasted
  - Or Reverse Path Forwarding (**RPF**): only forward packet if it arrived on shortest path between node and source
- 3. Spanning tree
  - Nodes forward copes only along spanning tree
  - No redundant packets received by any node

Broadcast Routing: Flooding

- 1. **Uncontrolled flooding**: One of the most simple and straightforward way to accomplish broadcast communication is for the sending node to send a separate copy of packet to each destination.
  - – Advantages: simple, no new network-layer routing protocol is needed.
  - – Drawbacks:
    - 1) inefficiency;
    - 2) the source should have a complete list of all destinations.
    - 3) Broadcast storm

Broadcast Routing: Packet ID

- 2. Controlled flooding: **Sequence-number-controlled flooding**
  - A source node puts its address (or other unique identifier) as well as **a broadcast sequence number** into a broadcast packet, then sends the packet to all its neighbors.
  - Each node maintains a list of the source address and sequence number of each broadcast packet it has already received, duplicated, and forwarded.
  - When a node receives a broadcast packet, it first checks whether the packet is in this list. If so, the packet is dropped; if not, the packet is duplicated and forwarded to all the node's neighbors.

Broadcast Routing: RPF

- 2. Controlled flooding: **Reverse Path Forwarding** (**RPF**)
  - When a broadcast packet arrives at a router, the router checks to see if the packet arrived on the link that is normally used for sending packets toward the source of the broadcast. If so, there is an excellent chance that the broadcast packet itself followed **the best route** from the router and is therefore the first copy to arrive at the router. This being the case, the router forwards copies of it onto all links except the one it arrived on. If, however, the broadcast packet arrived on the link other than the preferred one for reaching the source, the packet is discarded as a likely duplicate.
  - Advantage: RPF need only know the next neighbor on its unicast shortest path to the sender. It uses this neighbor’s identity only to determine whether or not to flood a received broadcast packet, without needing to remember sequence numbers.

Broadcast Routing: RPF

- A simple **RPF** example
  - Node B will forward the source-A packet it has received from A (since A is on its least-cost path to A). B will ignore (drop, without forwarding) any source-A packets it receives from any other nodes.

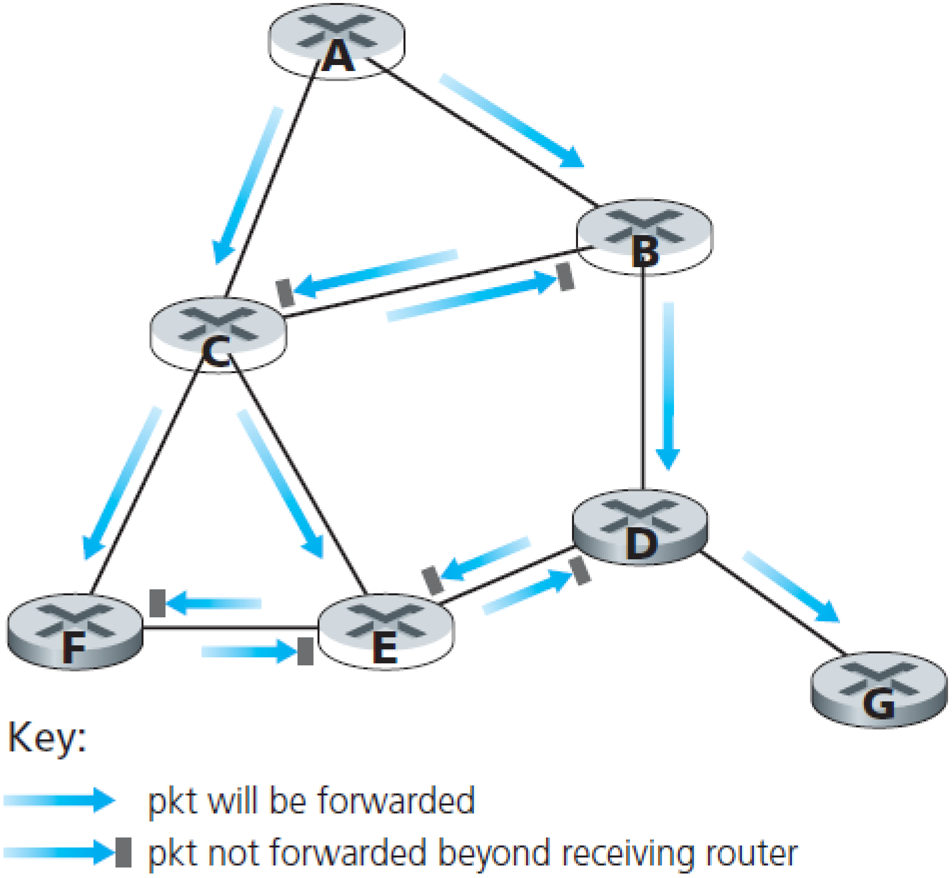


Figure 4.44 ♦ Reverse path forwarding

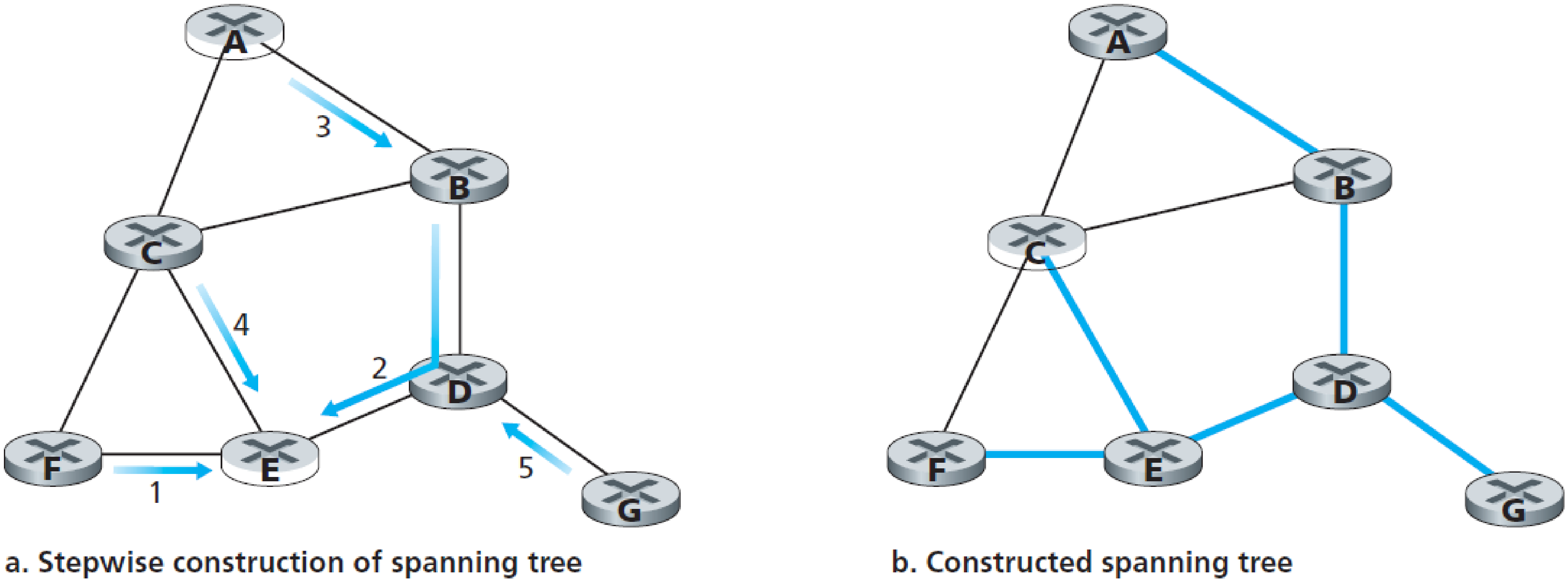
Broadcast Routing: Spanning-tree

- While sequence-number-controlled flooding and RPF avoid broadcast storms, they do **not** completely avoid the transmission of *redundant* broadcast packets. Ideally, every node should receive only one copy of the broadcast packet.
- 3. **Spanning-Tree Broadcast**
  - A tree contains each and every node in a graph but contains no cycles.
  - When a source node wants to send a broadcast packet, it sends the packet out on all of the incident links that belong to the spanning tree.
  - A node receiving a broadcast packet then forwards the packet to all its neighbors in the spanning tree.
  - A node need not be aware of the entire tree; it simply needs to know which of its neighbors in G are spanning-tree neighbors.

Spanning-tree: Creation

- The main complexity associated with the spanning-tree approach is the creation and maintenance of the spanning tree.
- The **center-based approach** to building a spanning tree
  - 1) A center node (also known as **a rendezvous point** or **a core**) is defined.
  - 2) Nodes then unicast tree-join messages addressed to the center node. A tree-join message is forwarded using unicast routing toward the center until it either arrives at a node that already belongs to the spanning tree or arrives at the center.
    - In either case, the path that the tree-join message has followed defines the branch of the spanning tree between the edge node that initiated the tree-join message and the center.
    - Graft (嫁接)

Spanning-tree: Creation



a. Stepwise construction of spanning tree

b. Constructed spanning tree

Figure 4.46 ♦ Center-based construction of a spanning tree

1) Suppose that node F first joints the tree and forwards a tree-join message to E. link EF becomes the initial spanning tree.
2) Node B then joins the spanning tree by sending its tree-join message to E. Suppose that the unicast path route to E from B is via D. The path BDE is grafted onto the spanning tree.
3) If A's unicast path to E is through B, then since B has already joined the spanning tree, the arrival of A's tree-join message at B will result in the AB link being grafted onto the tree.

Multicast Routing: Problem Statement

- To send messages to well-defined groups that are numerically large in size but small compared to the network as a whole.
  - Web cache updating
  - Interactive gaming
- In multicast communication, we are faced with two problems
  - How to identify the receivers of a multicast packet?
  - How to address a packet sent to these receivers?

Multicast Routing: two approaches

- In practice, two approaches have been adopted for determining the multicast routing tree.
- Multicast routing using a source-based tree
  - To construct a multicast routing tree for each source in the multicast group.
  - An RPF algorithm (with source node *x*) is used to construct a multicast forwarding tree for multicast datagrams originating at source *x*.
- Multicast routing using a group-shared tree.
  - As in the case of spanning-tree broadcast, multicast routing over a group-shared tree is based on building a tree that includes all edge routers with attached hosts belonging to the multicast group.
  - All routers along the path that the join message follows will then forward received multicast packets to the edge router that initiated the multicast join. (center-based tree)

Internet Multicast Routing

- In the Internet, the single identifier that represents a group of receivers is a class D multicast IP address.
- How does a group get started and how does it terminate? How is the group address chosen? How are new hosts added to the group (either as senders or receivers)…
  - The Internet Group Management Protocol (**IGMP**) [RFC3376]

Internet Multicast Routing

- Network-layer multicast in the Internet consists of two complementary components: **IGMP** and multicast routing protocols.
- IGMP has only *three* messages types. Like ICMP, IGMP messages are carried with an IP datagram, with an IP protocol number of 2.
  - 1) The *membership-query message* is sent by a router to all hosts on an attached interface to determine the set of all multicast groups that have been joined by the hosts on that interface. Membership-query messages can also be generated by a host when an application first joins a multicast group without waiting for a membership-query message from the router.
  - 2) Hosts respond with an IGMP *membership-report message*.
  - 3) The *leave-group message* is optional.
    - A host is no longer in the multicast group if it no longer responds to a membership-query message with the given group address.

Internet Multicast Routing

- Distance-vector multicast routing protocol (**DVMRP**) [RFC 1075]
  - DVMRP implements source-based trees with reverse path forwarding and pruning.
- The Protocol-Independent Multicast (**PIM**) routing protocol, which explicitly recognizes two multicast distribution scenarios.
  - PIM **dense** mode is a flood-and-prune reverse path forwarding technique similar in spirit to DVMRP.
  - PIM **sparse** mode uses rendezvous points to set up the multicast distribution tree.

Anycast Routing

- Unicast —— a single destination
- Broadcast —— to all destinations
- Multicast —— to a group of destinations
- In anycast, a packet is delivered to the nearest member of a group.
- Why would we want anycast? Sometimes nodes provide a service, such as time of day or content distribution for which it is getting the right information all that matters, not the node that is contacted; any node will do.
  - Anycast is used in the Internet as part of DNS (Chapter 7)
  - Robust to DDoS attack (Distributed Denial of Service) (Chapter 8)

Anycast Routing

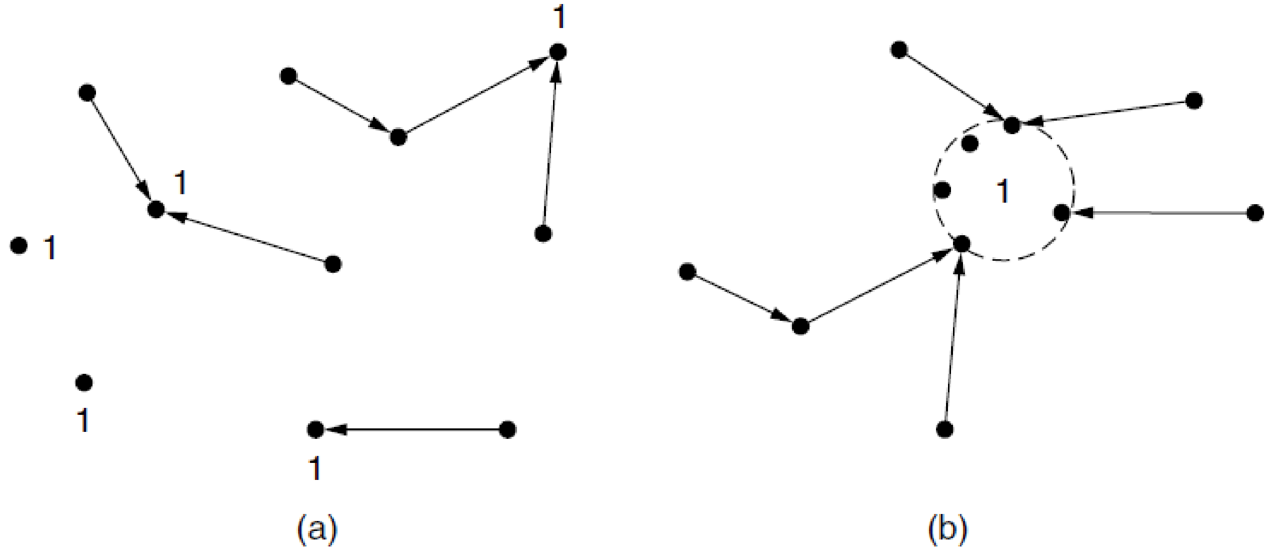


Figure 5-18. (a) Anycast routes to group 1. (b) Topology seen by the routing protocol.

- Not necessary to devise new routing schemes for anycast because regular distance vector and link state routing can produce anycast routes.
- For example, we want to anycast to the members of group 1. They will all be given the address “1”, instead of different addresses. Distance vector routing will distribute vectors as usual, and nodes will choose the shortest path to destination 1. This will result in nodes sending to the nearest instance of destination 1.
  - This procedure works because the routing protocol does not realize that there are multiple instances of destination 1.

Outline

- Overview of network layer
- Routing algorithms
- The network layer in the Internet
- MPLS (Multiprotocol Label Switching)

Outline

- Overview of network layer
- Routing algorithms
- The network layer in the Internet
  - IP Protocol
  - Control Protocols
  - Routing Protocols
- MPLS (Multiprotocol Label Switching)

Outline

- Overview of network layer
- Routing algorithms
- The network layer in the Internet
  - IP Protocol
    - IPv4 datagram
    - Fragment
    - IPv4 addressing
      - Subnet & subnet mask
      - Subnetting & route aggregation
      - nat box
      - DHCP
    - IPv6 datagram
  - Control Protocols
  - Routing Protocols
- MPLS (Multiprotocol Label Switching)

The Network Layer in the Internet

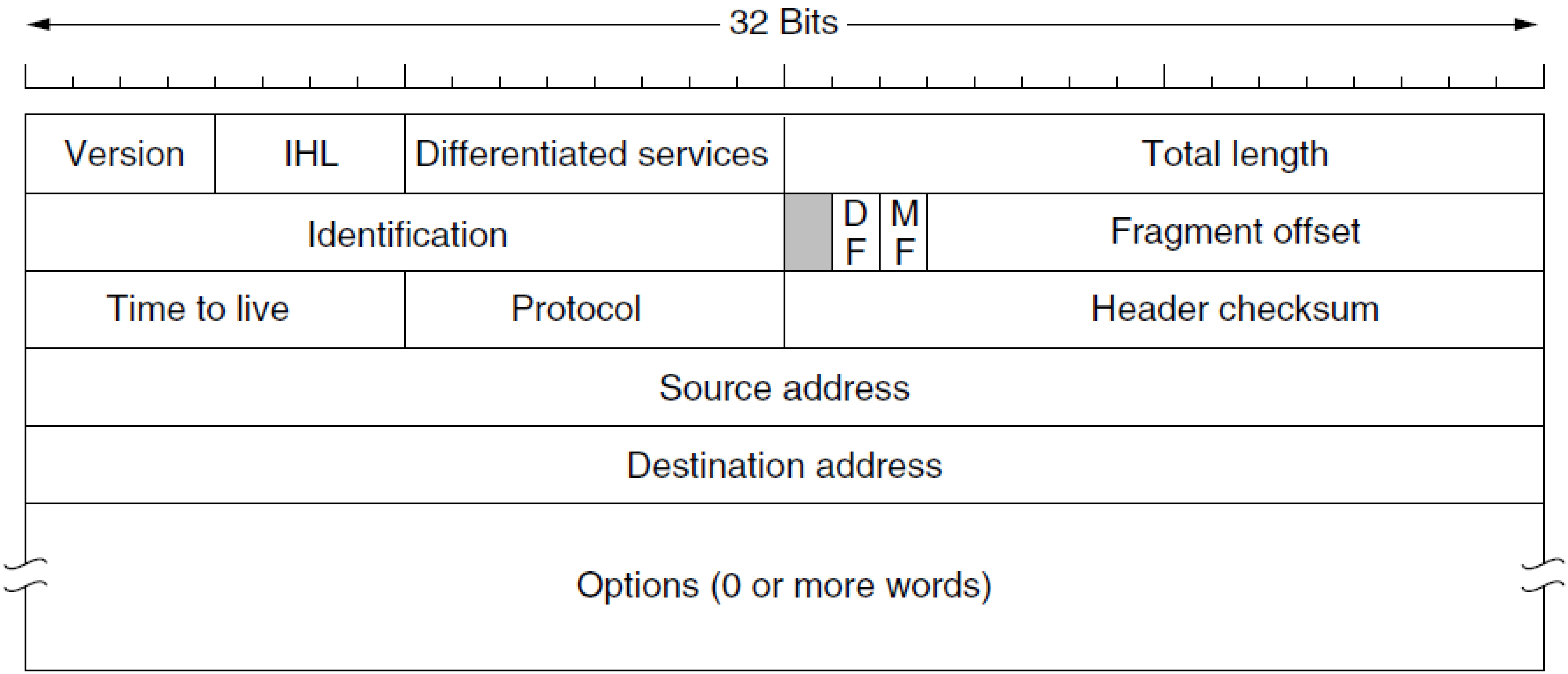
- There are two basic choices for connecting different networks:
  - We can build devices that *translate or convert packets* from each kind of network into packets for each other network.
  - Or, we can try to solve the problem by *building a common layer* on top of the different networks.
- History story:
  - Cerf and Kahn (1974) argued for **a common layer** to hide the differences of existing networks. The layer they proposed was eventually separated into the TCP and IP protocols.
    - **IP** is the foundation of the modern Internet. IP provides **a universal packet format** that all routers recognize and that can be passed through almost every network. There are two versions of IP in use today: **IPv4** and **IPv6**.
  - Cerf and Kahn were awarded the 2004 Turing Award.

The Network Layer of the Internet [8]

- The network layer of the internet has three main components:
  - The IP protocol
  - The Internet control protocols (including ICMP, DHCP, ARP)
  - The Internet routing protocols (including RIP, OSPF and BGP)

The IPv4 Datagram

- The header has a 20-byte fixed part and a variable-length optional part.
- The bits are transmitted from left to right and top to bottom. This is “big-endian” network byte order.



32 Bits

Version | IHL | Differentiated services | Total length

Identification | DF | MF | Fragment offset

Time to live | Protocol | Header checksum

Source address

Destination address

Options (0 or more words)

Figure 5-46. The IPv4 (Internet Protocol) header.

The IPv4 Datagram Format (I)

- 1. The Version field (4 bits)
  - By including the version at the start of each datagram, it becomes possible to have a transition between versions over a long period of time.

- 2. IHL field (Header Length) (4 bits)
  - To tell how long the header is in 32-bit words (a word = 4 bytes). The maximum value of this 4-bit field is 15, which limits the header to 60 bytes, and thus the options field to 40 bytes.
  - Most IP datagrams do not contain options, so the typical IP datagram has a 20-bytes header.

The IPv4 Datagram Format (II)

- 3. The Different Service field (Type of Service) (8 bits)
  - The top 6 bits are used to mark the packet with its service class, The bottom 2 bits are used to signal explicit congestion indications.
    - Real-time traffic (an IP telephone application) or non-real-time traffic (FTP)

- 4. The Total Length field (16 bits)
  - The total length of header and data.
  - The theoretical maximum length is 65,535 bytes. (the maximum payload of an IP packet is 65,515 = 65,535 – 20)
  - However, datagrams are rarely larger than **1,500 bytes**. (why?)

The IPv4 Datagram Format (III)

- 5. The Identification field (16 bits), flags (DF, MF), and fragment offset (13 bits)
  - These three fields have to do with so-called **IP fragmentation**.
  - <u>All the fragments of a packet contain the same identification field</u>.
  - 1) The Unused bit
  - 2) DF — Don’t Fragment
    - Now it is used as part of the process to discover the path MTU, which is the largest packet that can travel along a path without being fragmented.
  - 3) MF — More Fragments
    - All fragments except the last have this bit set (**MF = 0 means this the last fragment**).
  - 4) The Fragment Offset field (13 bits)
    - There is a maximum of 8192 fragments per datagram
    - How to implement datagram fragmentation will be discussed immediately.

The IPv4 Datagram Format (IV)

- 6. The TtL (Time to live) field
  - This field is decremented by one each time the datagram is processed by a router.
  - In practice, it just counts hops. When it hits zero, the packet is discarded and a warning packet is sent back to the source host.
- 7. The Protocol field (8 bit)
  - The protocol field tells is which transport process to give the packet to.
    - A value of 6 indicates that the data portion is passed to TCP.
    - A value of 17 indicates that the data is passed to UDP.
  - The protocol number is glue that binds the network and transport layer together, whereas the port number is the glue that binds the transport and application layers together.

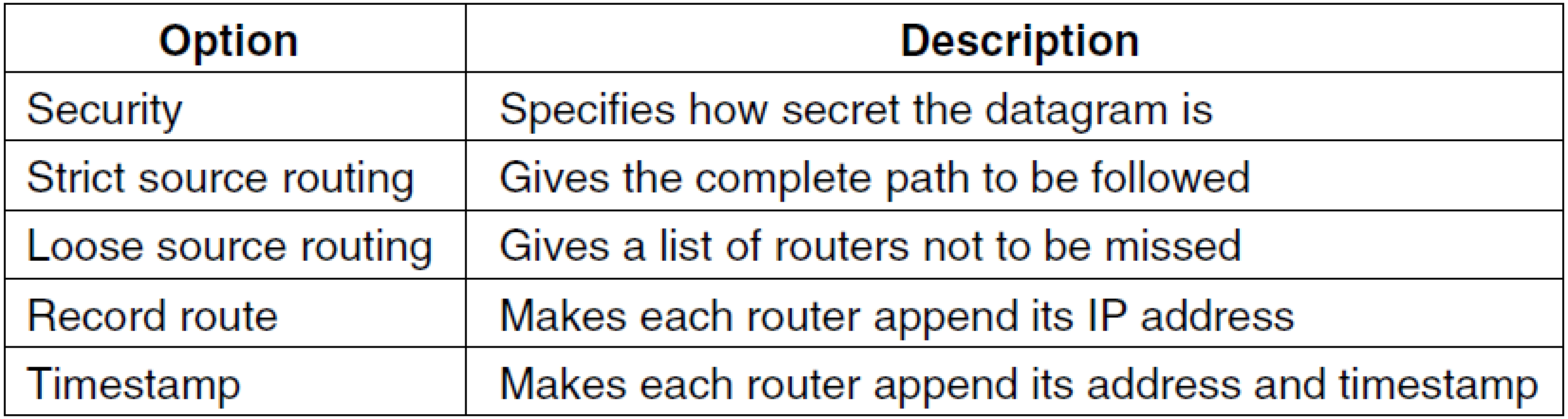
The IPv4 Datagram Format (V)

- 8. The header checksum
  - The header checksum is computed by treating each 2 bytes in the header as a number and summing these numbers using **one’s complement arithmetic**.
  - The Header checksum is assumed to be **zero** upon arrival.
  - A router computes the header checksum for *each received IP datagram* and detects an error condition if the checksum carried in the datagram header does not equal the computed checksum.
    - Routers typically discard datagrams for which an error has been detected.
  - Note that the checksum must be **recomputed** and stored again at each router, as **the TTL field**, and possibly the options fields as well, may change.

- 9. The Source address and Destination address (each with 32 bit)

The IPv4 Datagram Format (VI)

- 10. The Options field



Option	Description
Security	Specifies how secret the datagram is
Strict source routing	Gives the complete path to be followed
Loose source routing	Gives a list of routers not to be missed
Record route	Makes each router append its IP address
Timestamp	Makes each router append its address and timestamp

Figure 5-47. Some of the IP options.

- 11. Data (payload)

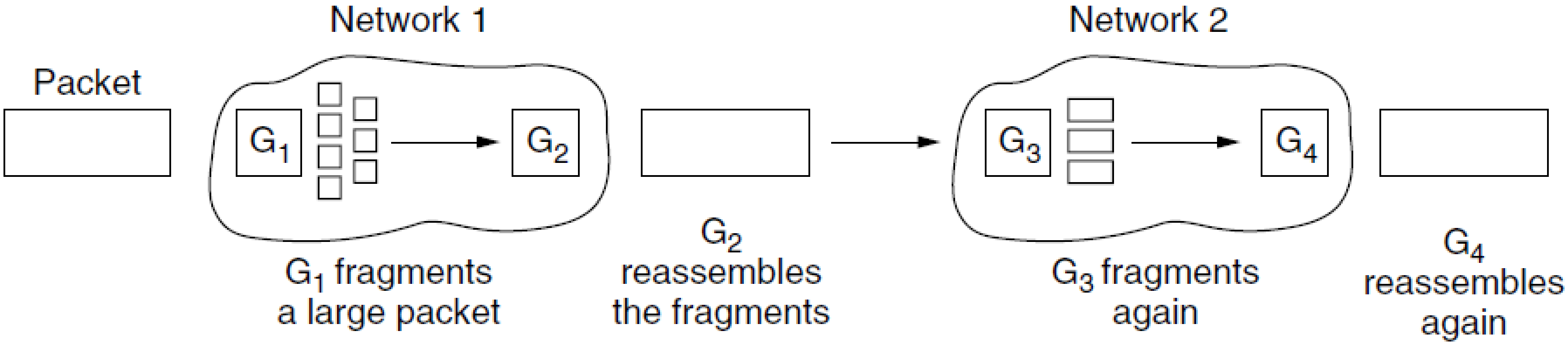
Packet Fragmentation (I)

- The maximum payloads of different networks
  - Ethernet — 1500 bytes
  - 802.11 — 2304 bytes (the maximum size of the frame body before encryption)
  - IP — 65,515 bytes

- Two Solutions
  - 1. To make sure the packet fragmentation does not occur in the 1st place.
    - Path MTU (Path Maximum Transmission Unit)
  - 2. To break up packets into fragments, sending each fragment as a separate network layer packet.
    - Two opposing strategies exist for recombining the fragments back into the original packet
      - **Transparent fragmentation**
      - **Nontransparent fragmentation**.

Packet Fragmentation (II)

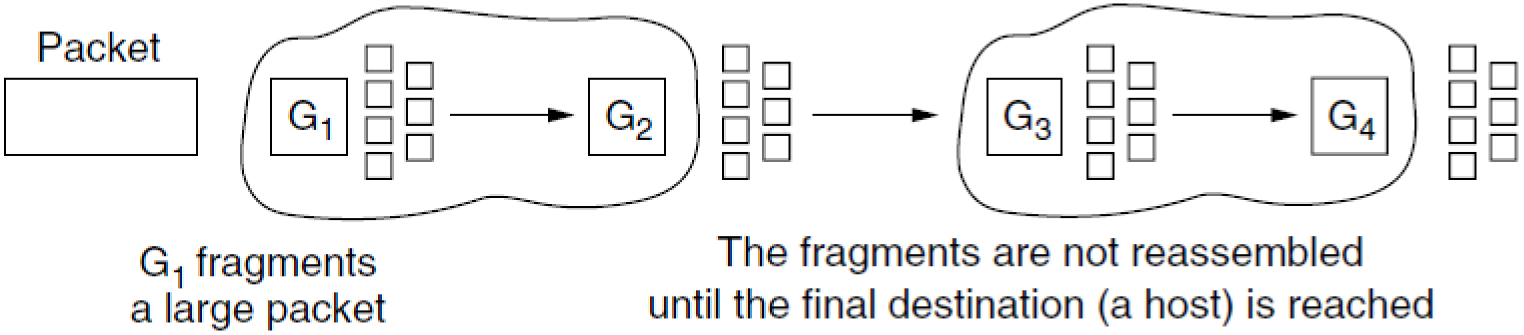
- **Transparent fragmentation** is straightforward but has some problems:
  - The exit router must know when it has received all the pieces (a count field or an "end of packet" bit (such as the "MF" bit in a IP datagram))
  - All packets must exit via the same router so that they can be reassembled, the routers are constrained.



(a)

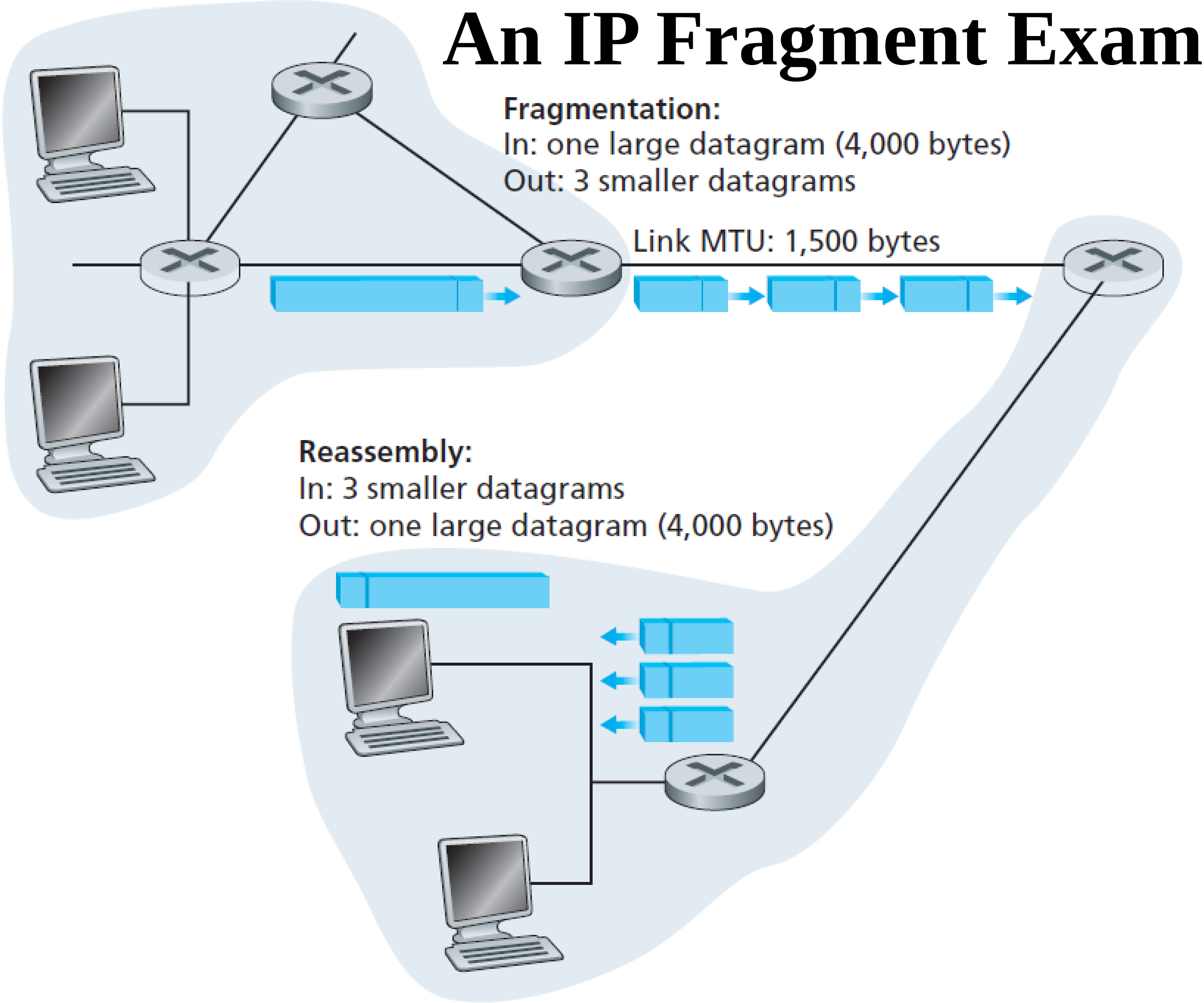
Packet Fragmentation (III)

- **Nontransparent fragmentation** is to refrain from recombining fragments at any intermediate routers.
  - Reassembly is performed only at the destination host
- The main advantage of nontransparent fragmentation is that it requires routers to do less work.
  - IPv4 works in this way.
  - A complete design requires that the fragments to be **numbered** in such a way that the original data stream can be reconstructed.


Packet
G_1
G_2
G_3
G_4
G_1 fragments a large packet
The fragments are not reassembled until the final destination (a host) is reached

(b)

An IP Fragment Example

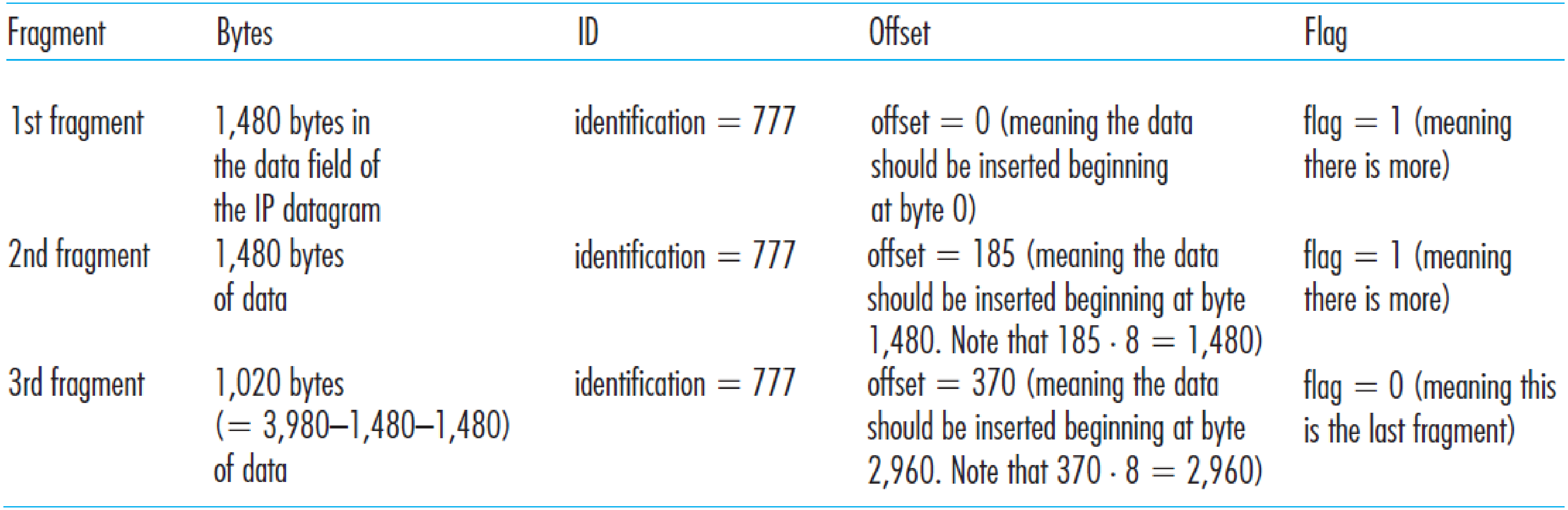


Fragmentation:
In: one large datagram (4,000 bytes)
Out: 3 smaller datagrams

Link MTU: 1,500 bytes

Reassembly:
In: 3 smaller datagrams
Out: one large datagram (4,000 bytes)

An IP Fragment Example [8]



Fragment	Bytes	ID	Offset	Flag
1st fragment	1,480 bytes in the data field of the IP datagram	identification = 777	offset = 0 (meaning the data should be inserted beginning at byte 0)	flag = 1 (meaning there is more)
2nd fragment	1,480 bytes of data	identification = 777	offset = 185 (meaning the data should be inserted beginning at byte 1,480. Note that 185 · 8 = 1,480)	flag = 1 (meaning there is more)
3rd fragment	1,020 bytes (= 3,980–1,480–1,480) of data	identification = 777	offset = 370 (meaning the data should be inserted beginning at byte 2,960. Note that 370 · 8 = 2,960)	flag = 0 (meaning this is the last fragment)

- 分片偏移就是某片在原分组的相对位置，以8个字节为偏移单位。这就是说，每个分片的长度一定是8字节（64位）的整数倍。
- 每个分片都要加上IP头部。
- MF = 0 means this the last fragment (MF is a flag bit in a IP datagram, MF — More Fragments)

Packet Fragmentation (IV)

- **Path MTU discovery** — the strategy used in the modern Internet.
- The advantage of path MTU discovery is that the source how know what length packet to send.
  - If the routers and path MTU change, new error packets will be triggered and the source will adapt to the new path.
- The disadvantage of path MTU discovery is that there may be **added startup delays** simply to send a packet, more than one round-trip delay may be needed to probe the path.

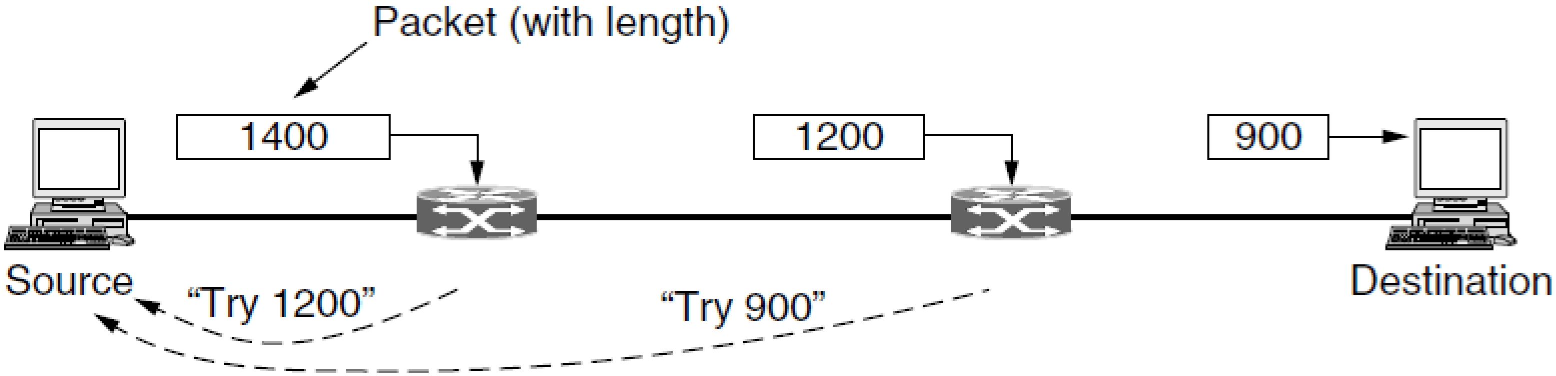


Figure 5-44. Path MTU discovery.

IPv4 Addressing (I)

- A defining feature of IPv4 is its 32-bit addresses.
- It is important to note that an IP address does not actually refer to a host. It really refers to *a network interface*, so if a host is on two networks, it must have two IP addresses.
  - In practice, most hosts are on one network and thus have one IP address.
  - In contrast, routers have multiple interfaces and thus multiple IP addresses.
- IP addresses are hierarchical, unlike Ethernet addresses.
- Each 32-bit address is compromised of a variable-length **network portion** in the top bits and **a host portion** in the bottom bits.

IPv4 Addressing (II)

- IP addresses are written in **dotted decimal notation**.
  - In this format, each of the 4 bytes is written in decimal, from 0 to 255.
  - Example: 128.208.2.151
- IP addresses can also be expressed in hexadecimal
  - Example: 128.208.2.151 = 80D00297
- Addresses are allocated in blocks called **prefixes**.
  - Addresses in an L-bit prefix have the same top L bits
  - There are $2^{32–L}$ addresses aligned on $2^{32–L}$ boundary.

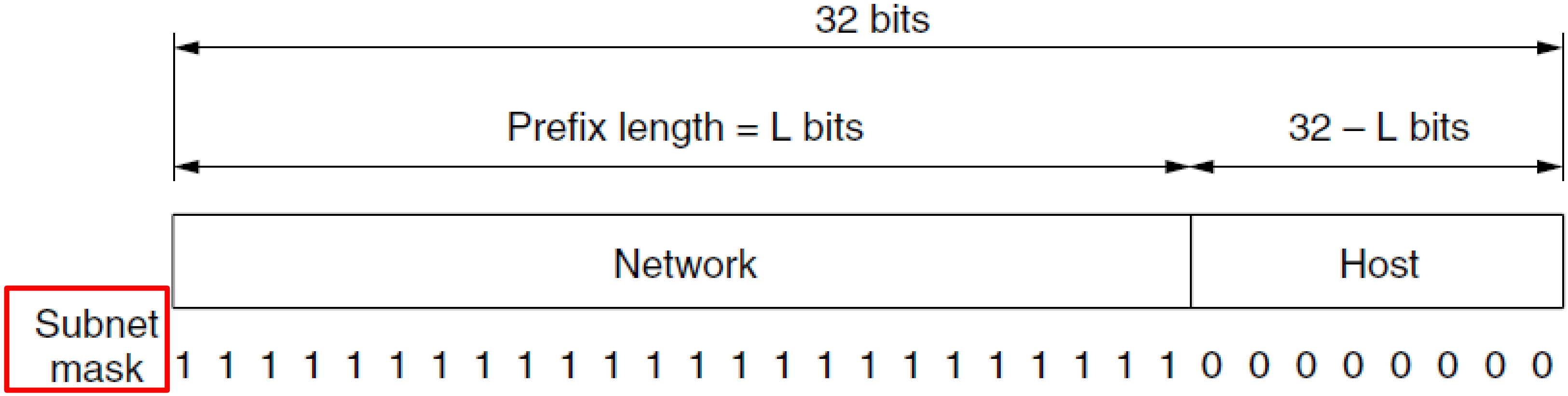


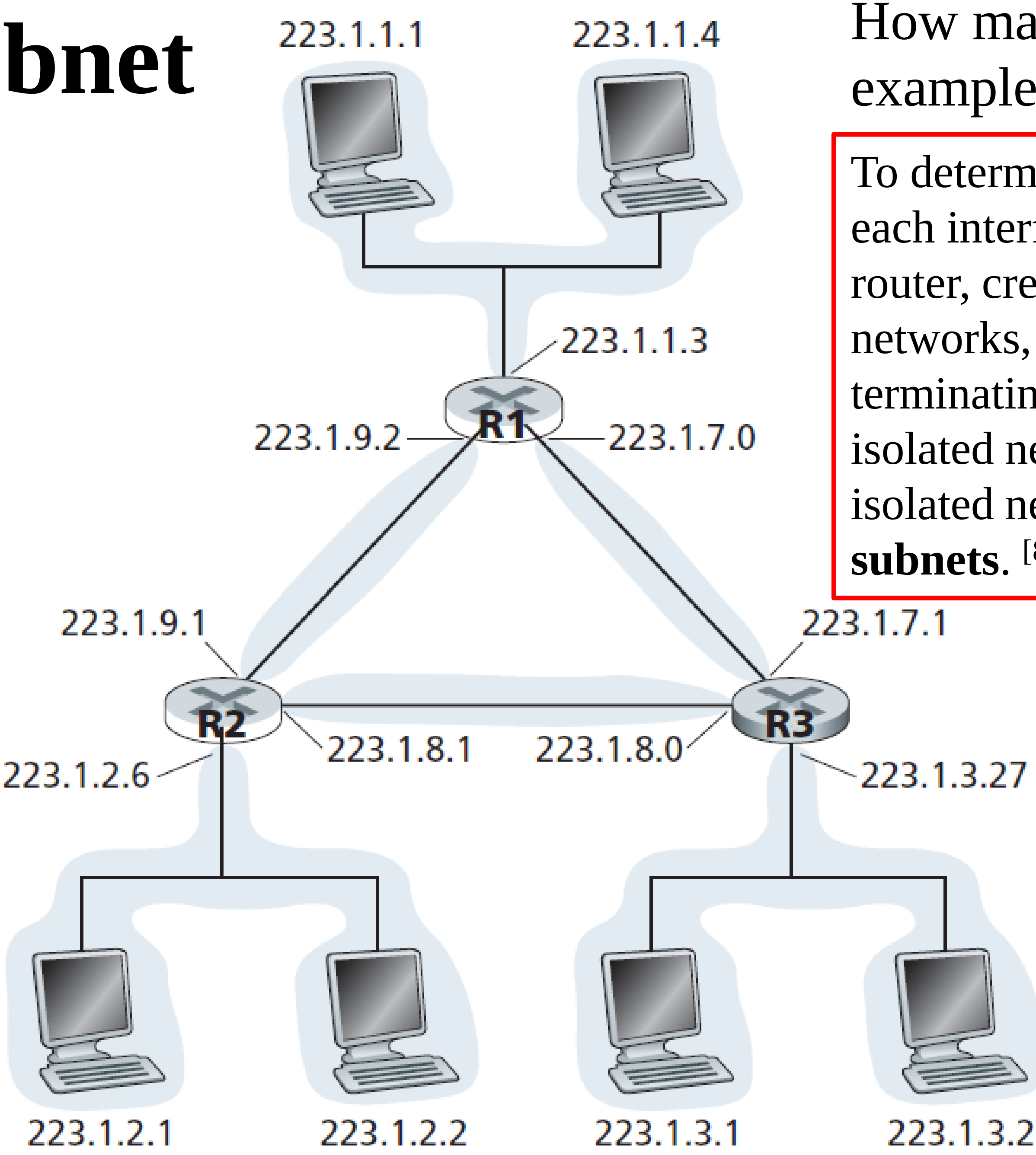
Figure 5-48. An IP prefix and a subnet mask.

IP Prefixes (Network portion)

- Written in "IP address/length" notation
  - Address is lowest address in the prefix, length is prefix bits. The /N sometimes known as **a subnet mask**.
    - /24 → The subnet mask is 255.255.255.0
  - E.g., 128.13.0.0/16 is 128.13.0.0 to 128.13.255.255
  - So a/24 has 256 addresses, and a/32 has only one address.

- The key advantage of prefixes is that routers can forward packets based on only the network portion of the address, as long as each of the networks has a unique address block.
  - More specific prefix has longer prefix, hence a smaller number of IP addresses.
  - Less specific prefix has shorter prefix, hence a larger number of IP addresses.

Subnet



How many **subnets** in this examples?

To determine the subnets, detach each interface from its host or router, creating islands of isolated networks, with interfaces terminating the end points of the isolated networks. Each of these isolated networks is called **subnets**. [8]

Subnets

- Routing by prefix requires all the hosts in a network to have the same network number.
- This property can cause problems as networks grows.
- The solution is to allow the block of addresses to be split into several parts for internal use as multiple networks, while still acting like a single network to the outside world. — **subnetting**

Subnets

Computer Science:	10000000	11010000	1|xxxxxxx	xxxxxxxx	a/17
Electrical Eng.:	10000000	11010000	00|xxxxxx	xxxxxxxx	a/18
Art:	10000000	11010000	011|xxxxx	xxxxxxxx	a/19

Here, the vertical bar (|) shows the boundary between the subnet number and the host portion.

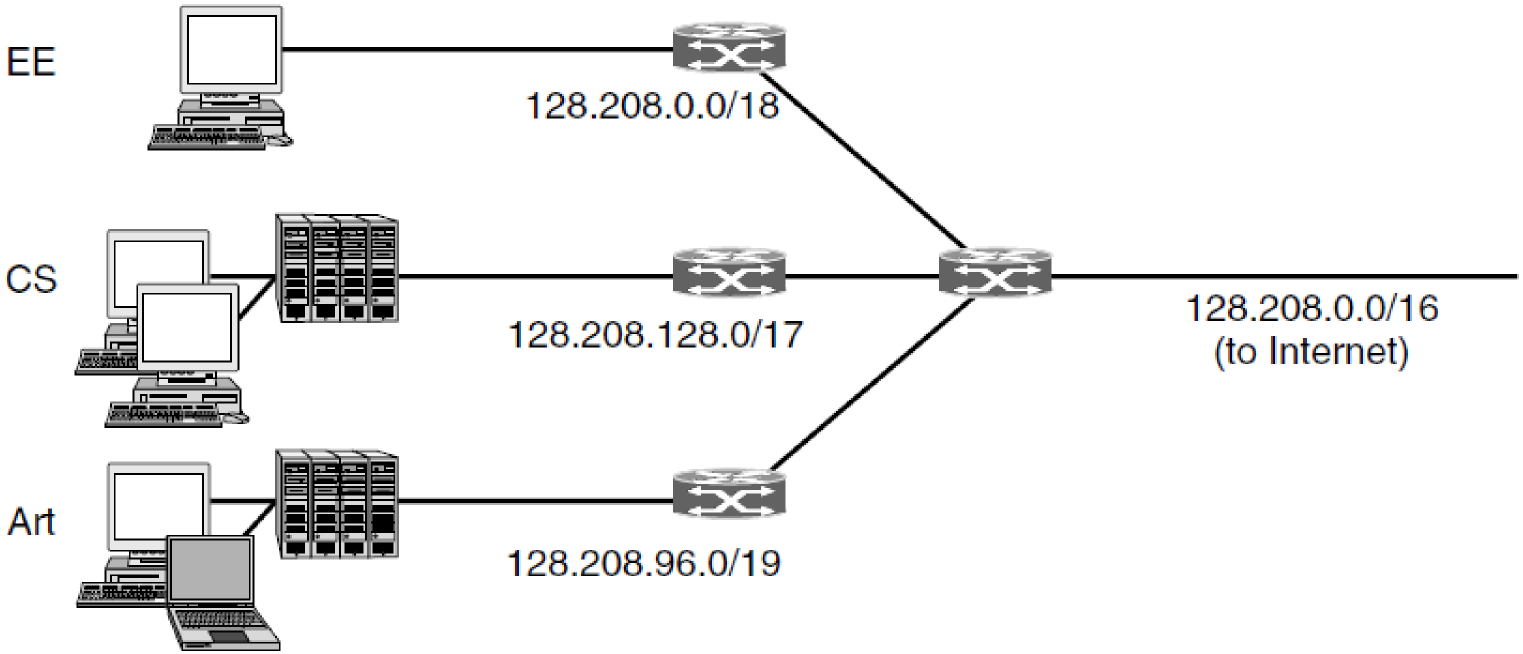


Figure 5-49. Splitting an IP prefix into separate networks with subnetting.

Subnets

- When a packet comes into the main router, how does the router know which subnet to give it to?
  - One solution is that for each router to have a table with 65536 entries telling it which outgoing line to use for each host on campus.
  - The other way is that the router can do this by **ANDing** the destination address with the mask for each subnet and checking to see if the result is the corresponding prefix.

IP Address Classes - Historical

- Before CIDR (Classless InterDomain Routing) was adopted, the network portions of an IP address were constrained to be 8, 16, or 24 bits in length, and addressing scheme known as **classful addressing**.

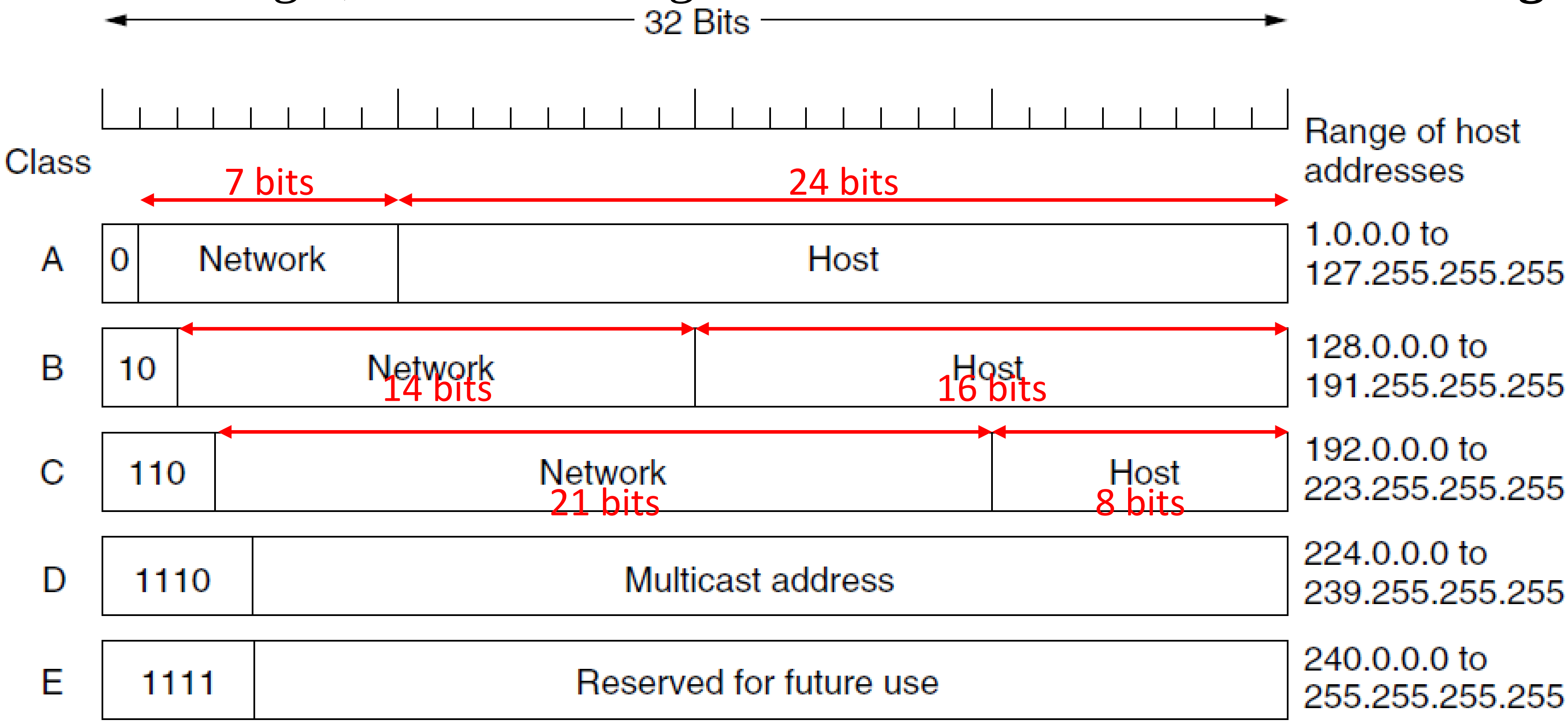


Figure 5-53. IP address formats.

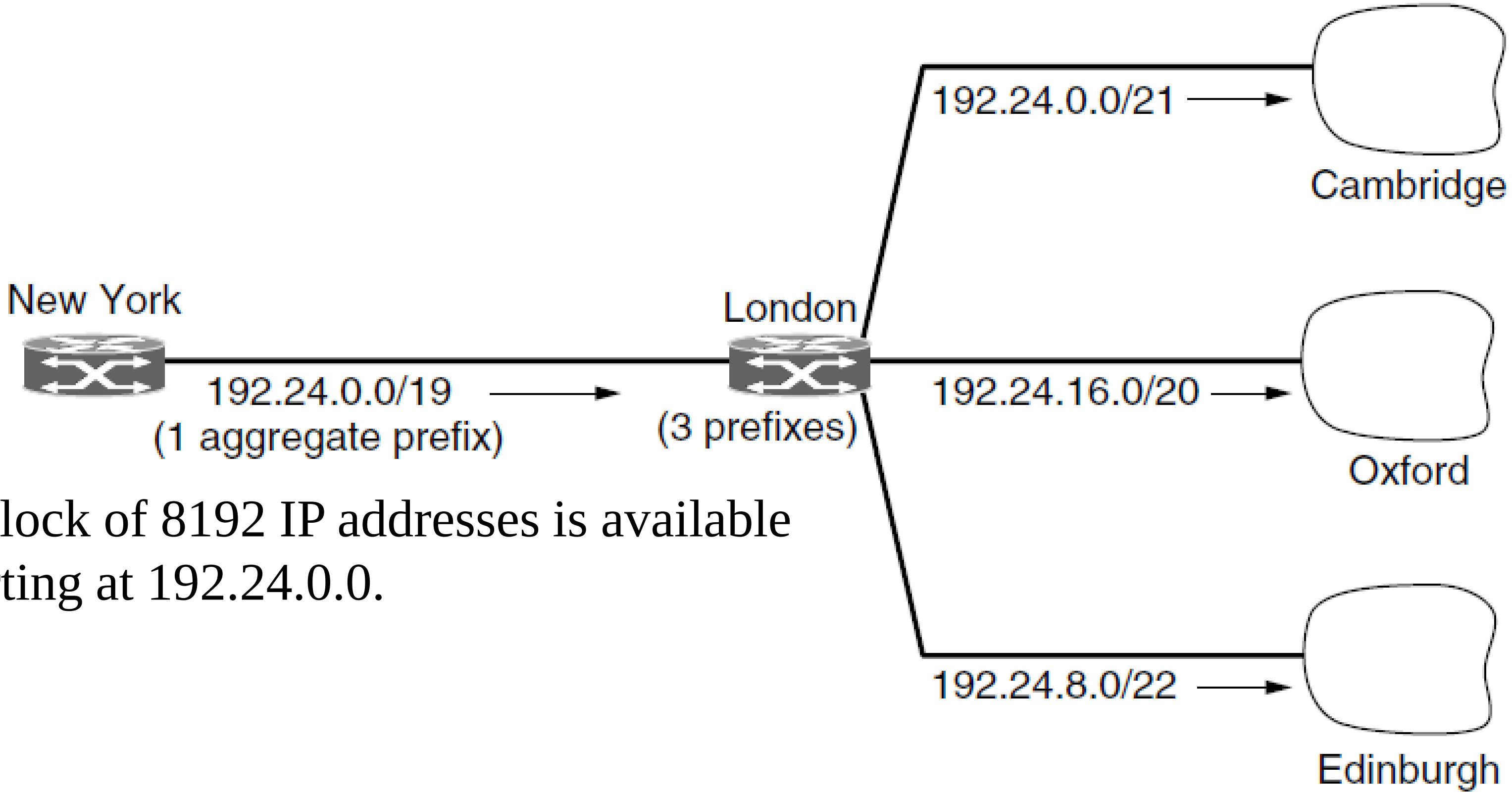
Allocating Public IP Addresses [8]

- IP addresses are managed under the authority of the Internet Corporation for Assigned Names and Numbers (**ICANN**) following *a hierarchical process*
  - RFC2050
  - The role of the nonprofit ICANN organization is not only to allocate IP addresses, but also to manage the DNS root servers.
  - The ICANN allocates addresses to regional Internet registries (for example, ARIN, RIPE, APNIC and LACNIC, which together form the Address Supporting Organization of ICANN)
  - Companies assign to their customers /computers (**DHCP**)

CIDR — Classless InterDomain Routing

- Even if blocks of IP addresses are allocated so that the addresses are used efficiently, there is still a problem that remains: **routing table explosion**. [RFC 4632]
- There is something we can do to reduce routing table sizes — By adjusting the size of IP prefixes
- **Subnetting**: split IP prefixes
- **Route aggregation**: combine multiple small prefixes into a single larger prefix.
- The design work of subnetting and route aggregation is called CIDR (Classless InterDomain Routing).

University	First address	Last address	How many	Prefix
Cambridge	192.24.0.0	192.24.7.255	2048	192.24.0.0/21
Edinburgh	192.24.8.0	192.24.11.255	1024	192.24.8.0/22
(Available)	192.24.12.0	192.24.15.255	1024	192.24.12.0/22
Oxford	192.24.16.0	192.24.31.255	4096	192.24.16.0/20



A block of 8192 IP addresses is available starting at 192.24.0.0.

Figure 5-51. Aggregation of IP prefixes.

The Longest Matching Prefix

- Prefixes are allowed to overlap.
- The rule is that packets are sent in the direction of the most specific route or **the longest matching prefix** that has the fewest IP address.

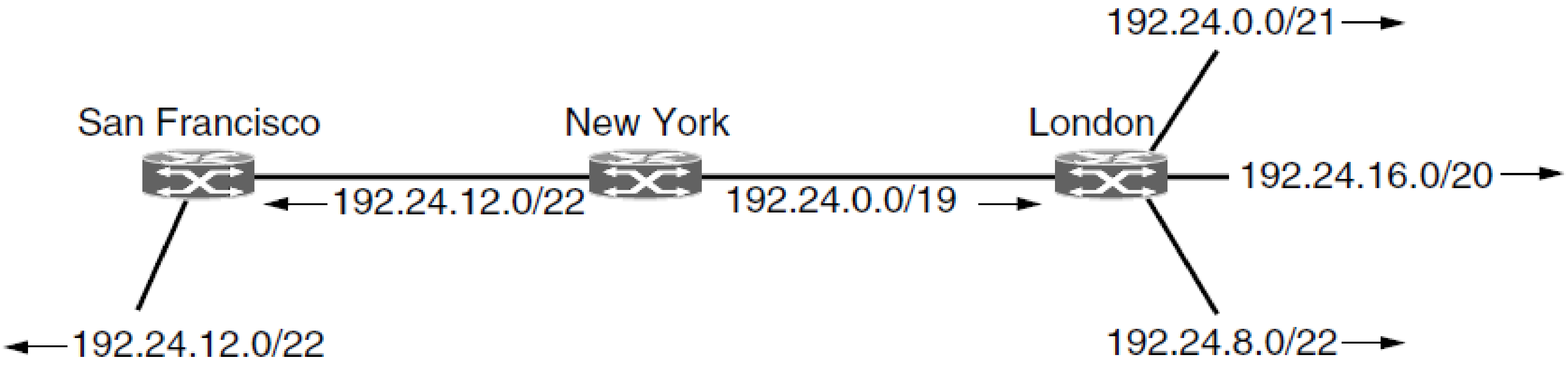


Figure 5-52. Longest matching prefix routing at the New York router.

Subnetting *vs.* Aggregation

- Two use cases for adjusting the size of IP prefixes, both reduce routing table.
- **Subnetting**
  - Internally split one less specific prefix into multiple more specific prefixes.
- **Aggregation**
  - Externally join multiple more specific prefixes into one large prefix.

Example

- A large number of consecutive IP addresses are available starting at 172.16.0.0. Suppose that four organizations: A, B, C and D, request 2000, 4000, 4000, and 8000 addresses, respectively, and in that order. For each of these, give the first IP address assigned, the last IP address assigned and the mask in the w.x.y.z/s notation. Assign addresses from small to large in the order of A to D.

Organization	First IP address	Last IP address	mask
A	172.16.0.0	172.16.7.255	172.16.0.0/21
B			
C			
D			

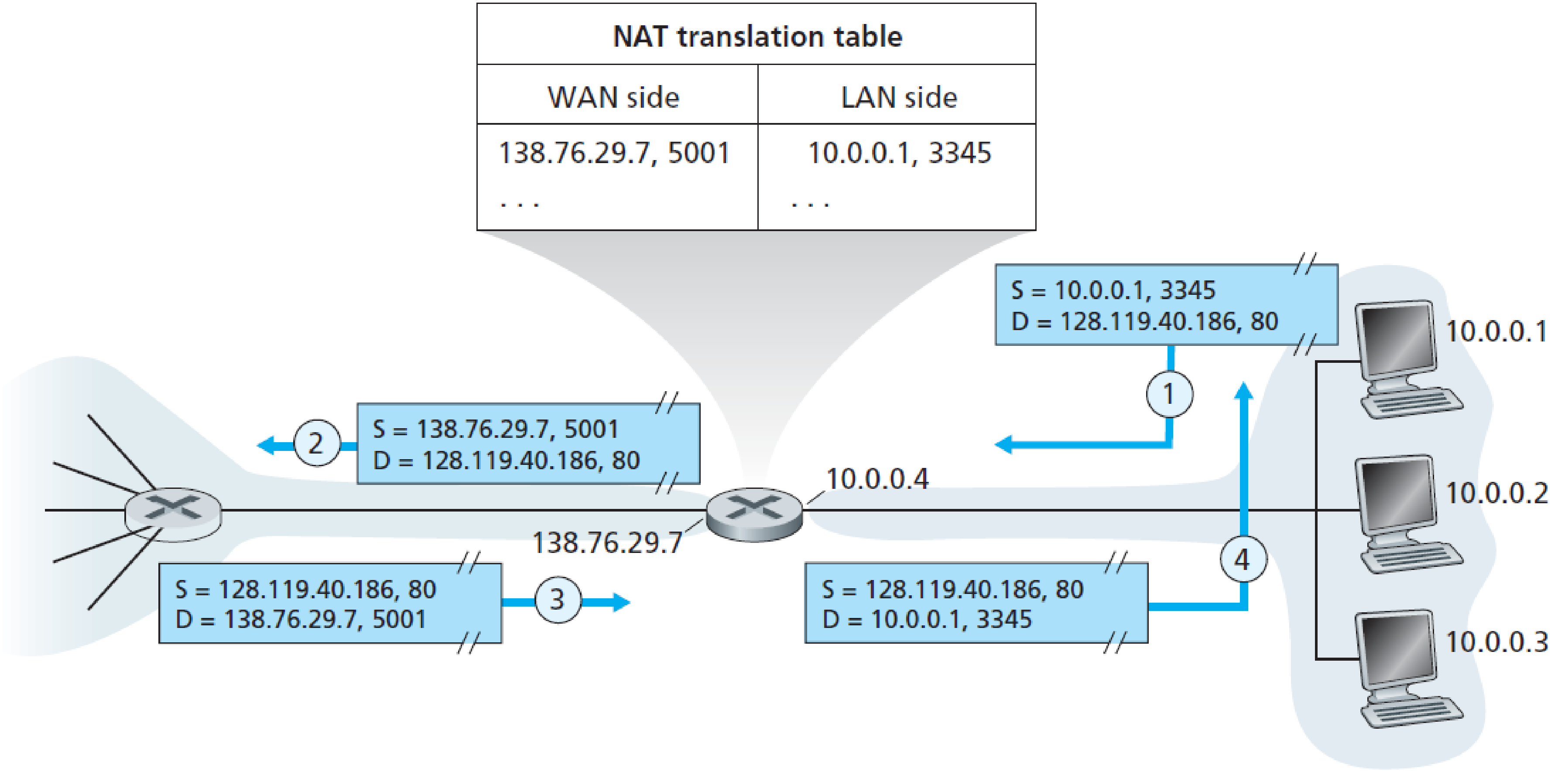
Special IP Addresses

- The IP address 0.0.0.0, the lowest address, is used by hosts when they are being booted. It means “this network” or “this host”.
- The IP address 255.255.255.255, the highest address, is used to mean all hosts on the indicated network. It allows **broadcasting** on the local network, typically a LAN.
- The IP address 127.0.0.1 (本机地址), and 127.xx.yy.zz are reserved for loopback testing.

NAT — Network Address Translation [5]

- IP addresses are scarce.
- 1) One solution is to dynamically assign an IP address to a compute when it is on and using the network, and to take the IP address back when it becomes inactive — **DHCP**
- 2) **NAT** box (Network Address Translation box) connects an internal network to an external network
  - Many internal hosts are connected using few external IP addresses.
  - The NAT box is often combined in a single device with a **firewall**, which provides security by carefully controlling what goes into the customer network and what comes out of it.
  - RFC 2663; RFC 3022

How NAT works [8]



NAT translation table	
WAN side	LAN side
138.76.29.7, 5001	10.0.0.1, 3345
. . .	. . .

Figure 4.22 ♦ Network address translation

The NAT translation table includes port numbers as well as IP addresses in the table entries. The NAT router can behave to the outside world as a single device with a single IP address.

How NAT works (II)

- Example

Internal IP : Port	External IP : Port
10.0.1.2 : 5544	128.143.71.21 : 3344
10.0.1.3 : 1234	128.143.71.21 : 3345
10.0.1.4 : 1234	128.143.71.21 : 3346

Private IP addresses

Public IP address

- Ports are effectively an extra 16 bits of addressing that identify which process gets which incoming packet.
- Ports 0-1023 are reserved for well-known services
  - Port 80 is the port used by Web servers

Problems with NAT

- Many purists in the IETF community loudly object to NAT:
  - Port numbers are meant to be used for addressing processes, not for addressing hosts.
  - Routers are supposed to process packets only up to layer 3.
  - The NAT protocol violates the so-called end-to-end argument; that is, hosts should be talking directly with each other, without interfering nodes modifying IP addresses and port numbers.
  - NAT changes the Internet from a *connectionless* network to a peculiar kind of *connection-oriented* network.
    - If the NAT box crashes and its mapping table is lost, all its connections are destroyed.
  - NAT violates the most fundamental rule of protocol layering: layer *k* may not make any assumptions about what layer *k*+1 has put into the payload field.

Dynamic Host Configuration Protocol (**DHCP**) [8] (I) *

- Host address can be configured manually, but more often this task is now done using the Dynamic Host Configuration Protocol (**DHCP**).
  - RFC 2131
  - In addition to host IP address assignment, DHCP also allows a host to learn additional information, such as its subnet mask, the address of its first-hop router (often called the default gateway), and the address of its local DNS server.
  - DHCP is often referred to as **a plug-and-play protocol**.
    - Each time a host joins, the DHCP server allocates an arbitrary address from its current pool of available addresses; each time a host leaves, its address is returned to the pool.

Dynamic Host Configuration Protocol (**DHCP**) [8] (II) *

- Host address can be configured manually, but more often this task is now done using the Dynamic Host Configuration Protocol (**DHCP**).
  - DHCP is a *client-server* protocol.
    - In the simplest case, each subnet will have a DHCP server. If no server is present on the subnet, a DHCP relay agent (typically a router) that knows the address of a DHCP server for that network is needed.
  - For a newly arriving host, the DHCP protocol is **a four-step process**.

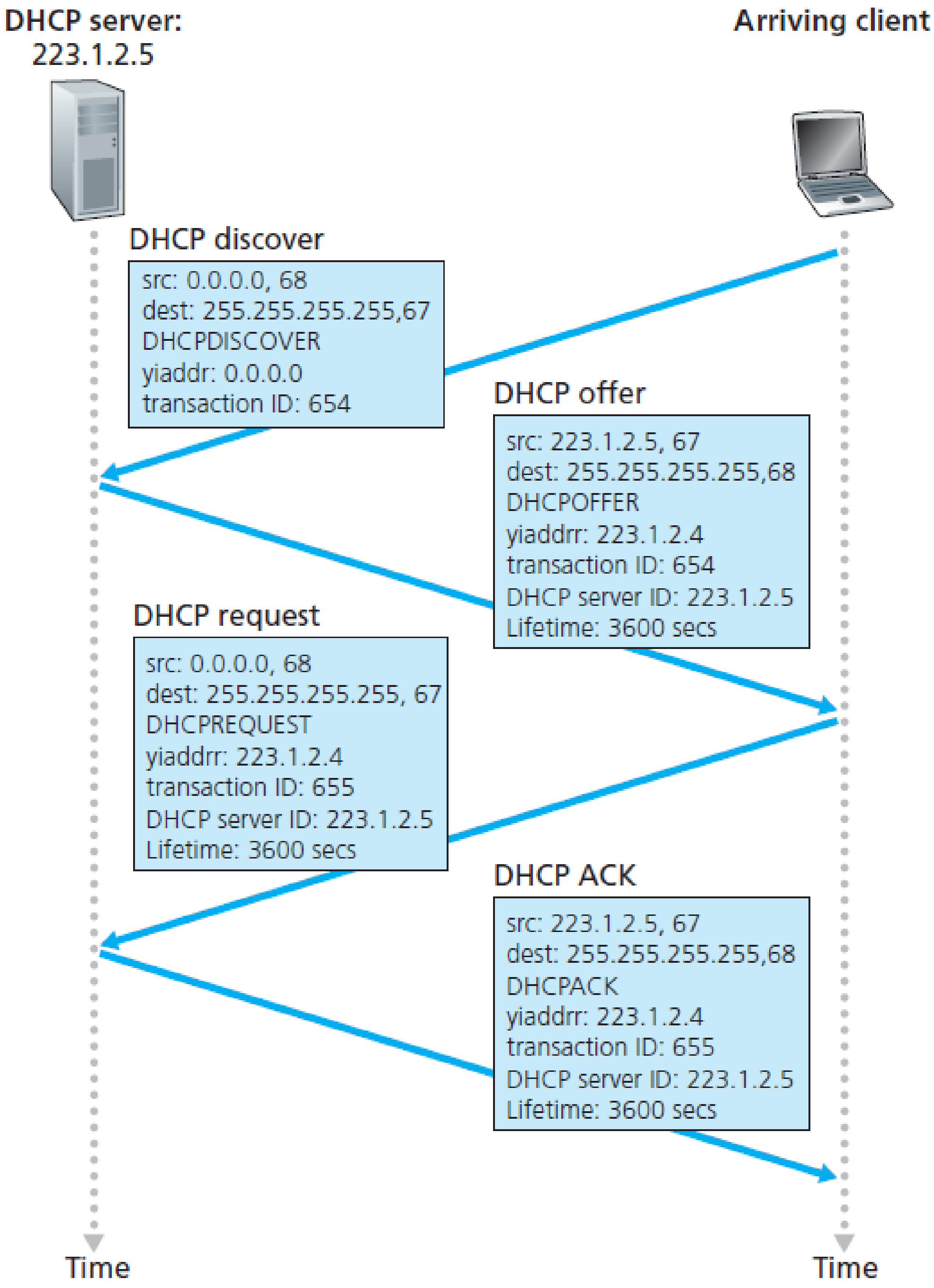
The Four-step Process of DHCP (I) *

- 1) DHCP server discovery.
  - This is done using **a DHCP discover message**, which a *client* sends within a UDP packet to port **67**.
  - The UDP packet is encapsulated in an IP datagram with the **broadcast** destination IP address of 255.255.255.255 and a “this host” source IP address of 0.0.0.0.
- 2) DHCP server offer(s)
  - A DHCP *server* receiving a DHCP discovery message responds to the client with **a DHCP offer message** that is **broadcast** to all nodes on the subnet.
  - Several DHCP servers can be present on the subnet, the client may choose from among several offers.

The Four-step Process of DHCP (II) *

- 3) DHCP request
  - The newly arriving client will choose from among one or more server offers and respond to its selected offer with **a DHCP request message** echoing back the configuration parameters.
- 4) DHCP ACK
  - The server responds to the DHCP request message with **a DHCP ACK message**, confirming the requested parameters.

- Once the client receives the DHCP ACK, the interaction is complete and the client can use the DHCP-allocated IP address for the lease duration.
  - renew



DHCP discover

src: 0.0.0.0, 68
dest: 255.255.255.255,67
DHCPDISCOVER
yiaddr: 0.0.0.0
transaction ID: 654

DHCP offer

src: 223.1.2.5, 67
dest: 255.255.255.255,68
DHCPOFFER
yiaddrr: 223.1.2.4
transaction ID: 654
DHCP server ID: 223.1.2.5
Lifetime: 3600 secs

DHCP request

src: 0.0.0.0, 68
dest: 255.255.255.255, 67
DHCPREQUEST
yiaddrr: 223.1.2.4
transaction ID: 655
DHCP server ID: 223.1.2.5
Lifetime: 3600 secs

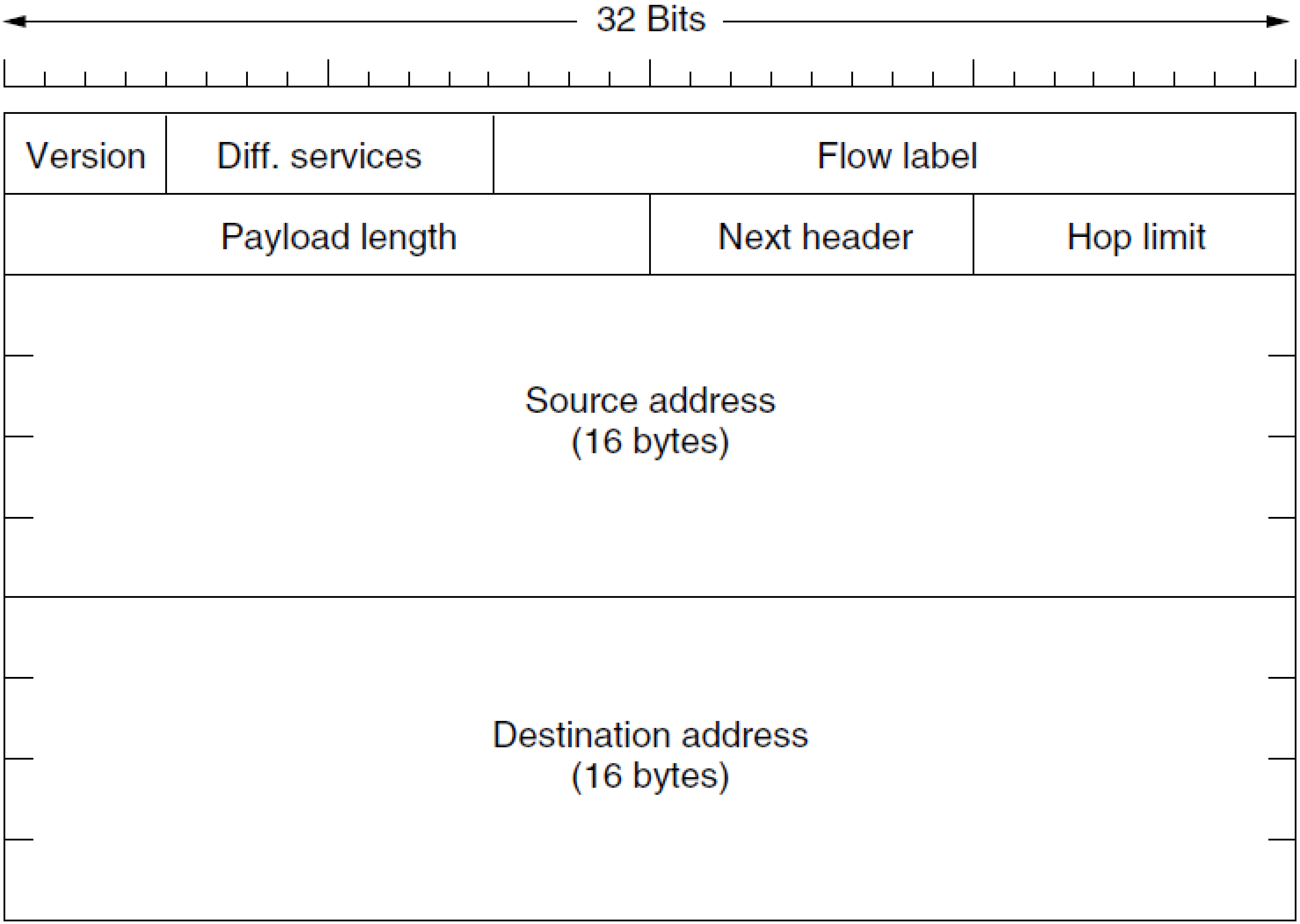
DHCP ACK

src: 223.1.2.5, 67
dest: 255.255.255.255,68
DHCPACK
yiaddrr: 223.1.2.4
transaction ID: 655
DHCP server ID: 223.1.2.5
Lifetime: 3600 secs

IPv6 Addressing

- IPv6 uses 128-bit addresses. [RFC2460]
    - An Internet Standard since 1998.
    - IPv6 is **not** compatible with IPv4, but it is compatible with other auxiliary Internet protocols, including TCP, UDP, ICMP, IGMP, OSPF, BGP, and DNS
    - The main features:
        - 1. IPv6 has longer address than IPv4.
        - 2. The simplification of the header contains only seven fields (vs. 13 in IPv4)
            - This change allows routers to process packets faster and thus improve throughput and delay.
        - 3. Better support for options → speeds up packet processing time
        - 4. Security
        - 5. Quality of services

The IPv6 Datagram Header (I)



32 Bits

Version | Diff. services | Flow label

Payload length | Next header | Hop limit

Source address (16 bytes)

Destination address (16 bytes)

Figure 5-56. The IPv6 fixed header (required).

The IPv6 Header (II)

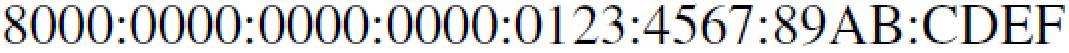
- 1. the Version field (4 bits)
  - **6** for IPv6 or 4 for IPv4 (Note that putting a 4 in this field does not create a valid IPv4 datagram.)
- 2. the Difference Services field (8 bits)
  - The low-order 2 bits are used to signal explicit congestion indications
- 3. the Flow Label field (20 bits)
  - When a packet with a nonzero flow label shows up, all the routers can look it up in internal tables to see what kind of special treatment it requires.
  - The flexibility of a datagram network and the guarantees of *a virtual-circuit network*
  - RFC 1752 and RFC 2460

IPv6 Header (III)

- 4. the Payload length field (16 bits)
  - Tells how many bytes follow the 40-byte header
- 5. the Next Header field (8 bits)
  - The next header field tells which transport protocol handler (e.g. TCP, UDP) to pass the packet to.
  - This field uses the same values as **the protocol field** in the IPv4 header.
- 6. the Hop Limit field (8 bits)
  - The same as **the Time to Live** field in IPv4.
- 7. the Source address and Destination address fields (each with 128 bits or 16 bytes)
  - RFC 4291

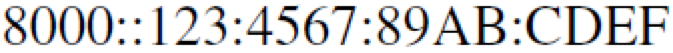
The IPv6 Address

- IPv6 addresses are written as *eight* groups of four hexadecimal digits with colons between the groups



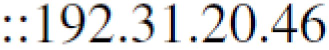
8000:0000:0000:0000:0123:4567:89AB:CDEF

- Since many addresses will have many zeros inside them, three optimization have been authorized:
  - Leading zeros with a group can be omitted, so 0123 can be written as 123.
  - One or more groups of 16 zero bits can be replaced by a pair of colons



8000::123:4567:89AB:CDEF

  - IPv4 addresses can be written as a pair of colons and an old dotted decimal number



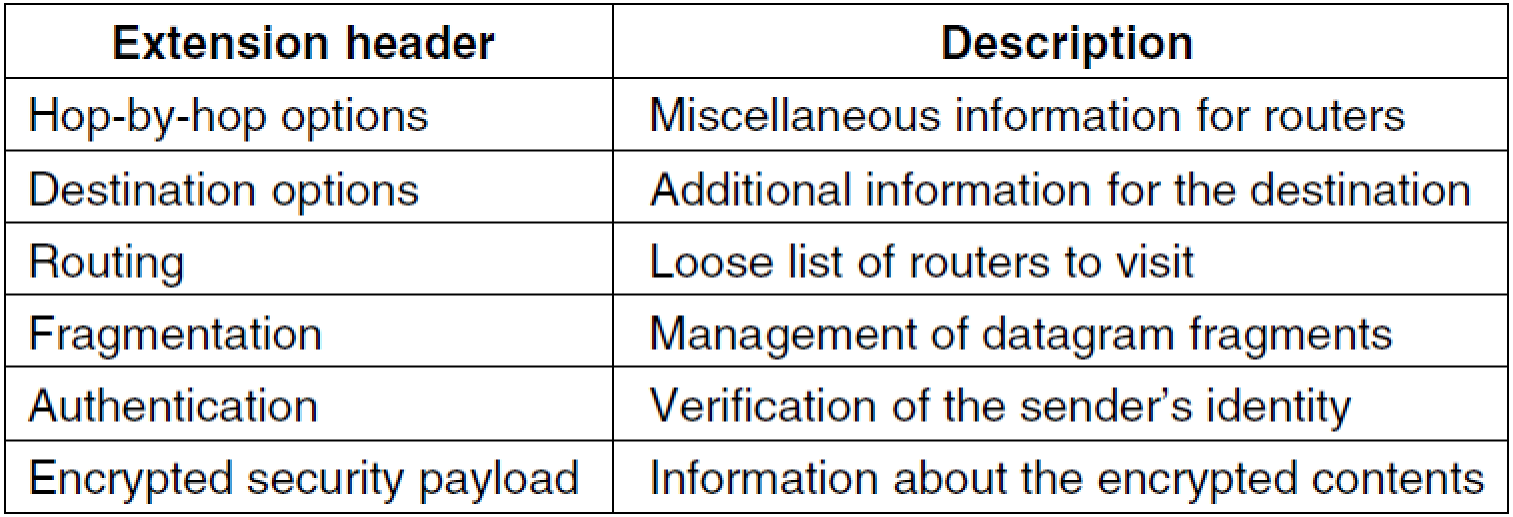
::192.31.20.46

Several Fields in IPv4 are no longer present in the IPv6 datagram [8]

- 1) Fragmentation/reassembly. IPv6 does **not** allow for fragmentation and reassembly at intermediate routers; these operations can be performed only the source and destination.
  - Fragmentation and reassembly is a time-consuming operation; removing this functionality from the routers can speeds up IP forwarding within the network.
  - If an IPv6 datagram received by a router is too large to be forwarded over the outgoing link, the router simply drops the datagram and sends a "Packet Too Big" ICMP error message back to the sender.
- 2) No header checksum
  - Because the transport-layer (for example, TCP and UDP) and data link layer (for example, Ethernet) protocols in the Internet layer perform checksum. So this functionality was redundant in the network layer.
  - Since IPv4 header contains a TTL field, the IPv4 header checksum need to be recomputed at every router. Without checksum, it is faster to process IP packets.
- 3) Options. The removal of the options field results in a **fixed**-length, 40-byte IP header.

IPv6 Extension Headers (I)

- Some of the missing IPv4 fields are occasionally still needed, so IPv6 introduces the concept of (optional) extension headers.
  - Each one is optional, but if more than one is present they must appear directly after the fixed header, and preferably in the order listed.



Extension header	Description
Hop-by-hop options	Miscellaneous information for routers
Destination options	Additional information for the destination
Routing	Loose list of routers to visit
Fragmentation	Management of datagram fragments
Authentication	Verification of the sender’s identity
Encrypted security payload	Information about the encrypted contents

Figure 5-57. IPv6 extension headers.

IPv6 Extension Headers (II)

- Each item is encoded as a (Type, Length, Value) tuple.
- The Type is a 1-byte field telling which option this is.
  - The first 2 bits tell routers that do not know how to process the option what to do. The choices are: skip the option, discard the packet; discard the packet and send back an ICMP packet; and discard the packet but do not send ICMP packets for multicast addresses
- The Length is a 1-byte field. It tells how long the value is (0 to 255 bytes)
- The Value is any information required, up to 255 bytes.

IPv6 Extension Headers (III)

- **The hop-by-hop header** is used for information that all routers along the path must examine.
- **The destination options header**
- **The routing header** lists one or more routers that must be visited on the way to the destination. It is very similar to the IPv4 loose source routing.
  - The Next header, Header extension length, Routing type, Fragment left
- **The fragmentation header** deals with fragmentation similarly to the way IPv4 does.
  - In IPv6, unlike in IPv4, only the source host can fragment a packet.
- **The authentication header** provides a mechanism by which the receiver of a packet can be sure of who sent it.
- **The encrypted security payload** makes it possible to encrypt the contents of a packet so that only the intended recipient can read it.

Transitioning from IPv4 to IPv6 (I) [8]

- RFC 4213 describes two approaches.
  - 1) **IPv6-capable nodes** is a **dual-stack** approach, where IPv6 nodes also have a complete IPv4 implementation.
    - Some IPv6-specific fields in the IPv6 datagram will be **missed**.

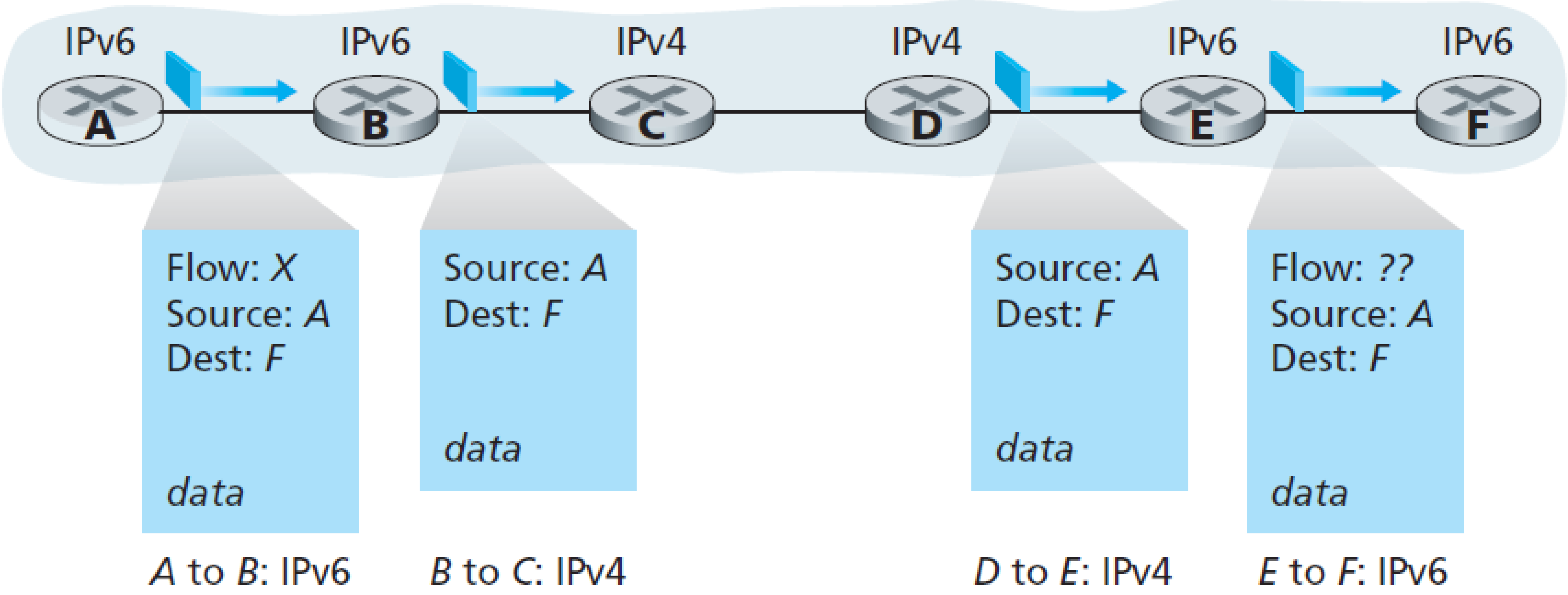


Figure 4.25 ♦ A dual-stack approach

Transitioning from IPv4 to IPv6 (II) [8]

- RFC 4213 describes two approaches.
  - 2) **Tunneling**. To take the entire IPv6 datagram into the data (payload) field of an IPv4 datagram.

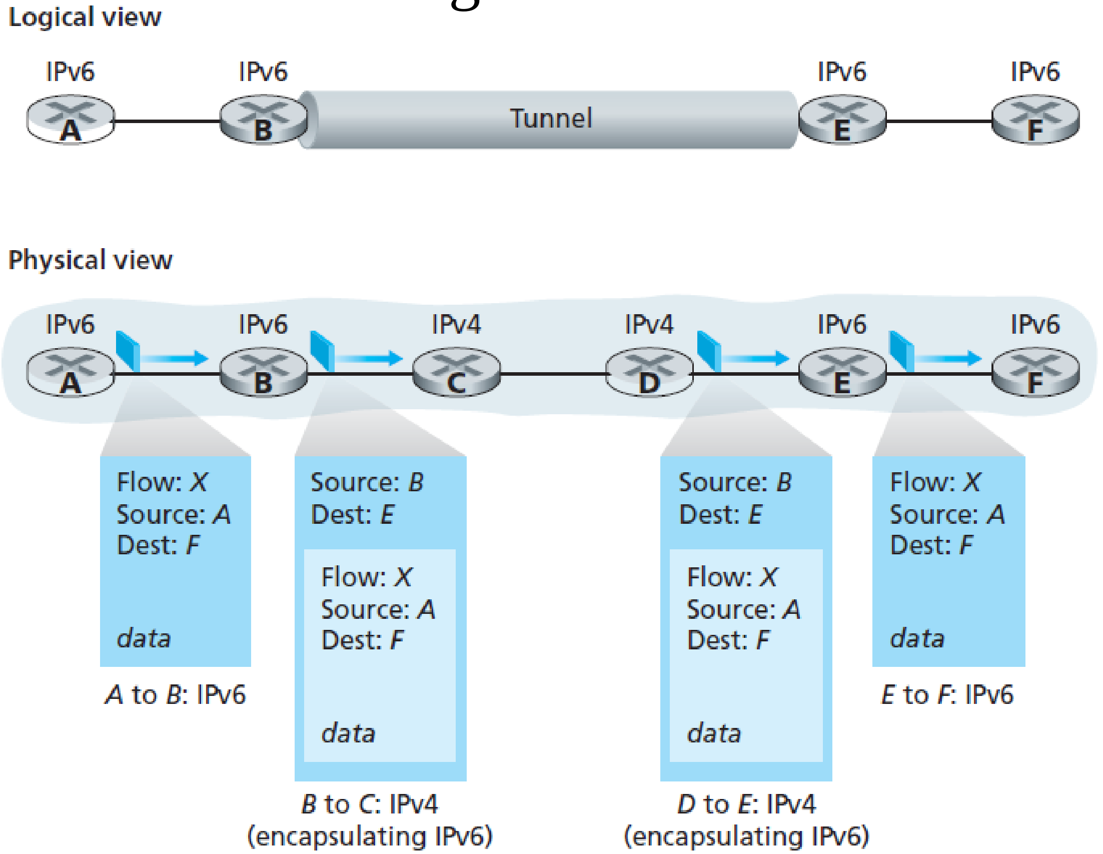


Figure 4.26 ♦ Tunneling

Tunneling

- This case is where the source and destination hosts are on the same type of network, but there is a different network in between.
  - The disadvantage of tunneling is that none of the hosts on the network that is tunneled over can be reached because the packets cannot escape in the middle of the tunnel.
  - But this limitation is turned into an advantage with **VPNs** (Virtual Private Networks, Chapter 8)
-

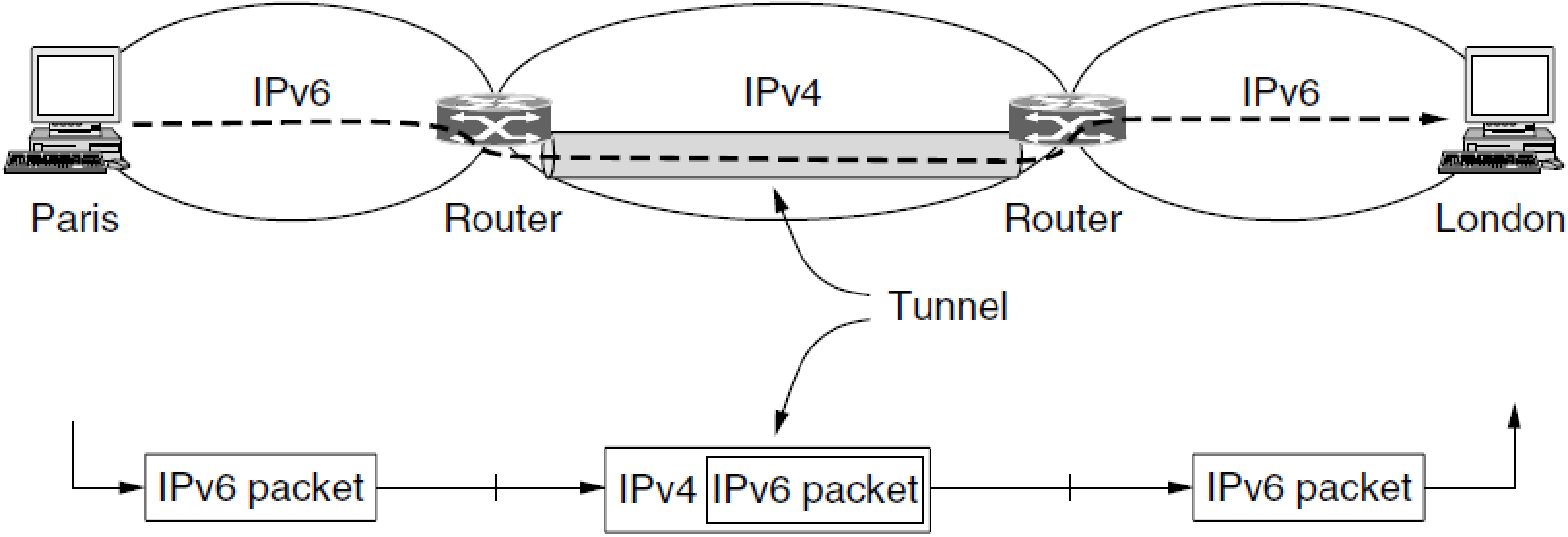


Figure 5-40. Tunneling a packet from Paris to London.

Outline

- Overview of network layer
- Routing algorithms
- The network layer in the Internet
  - IP Protocol
  - Control Protocols
  - Routing Protocols
- MPLS (Multiprotocol Label Switching)

Outline

- Overview of network layer
- Routing algorithms
- The network layer in the Internet
  - IP Protocol
  - Control Protocols
    - ICMP
    - DHCP
    - ARP
  - Routing Protocols
- MPLS (Multiprotocol Label Switching)

Internet Control Protocols

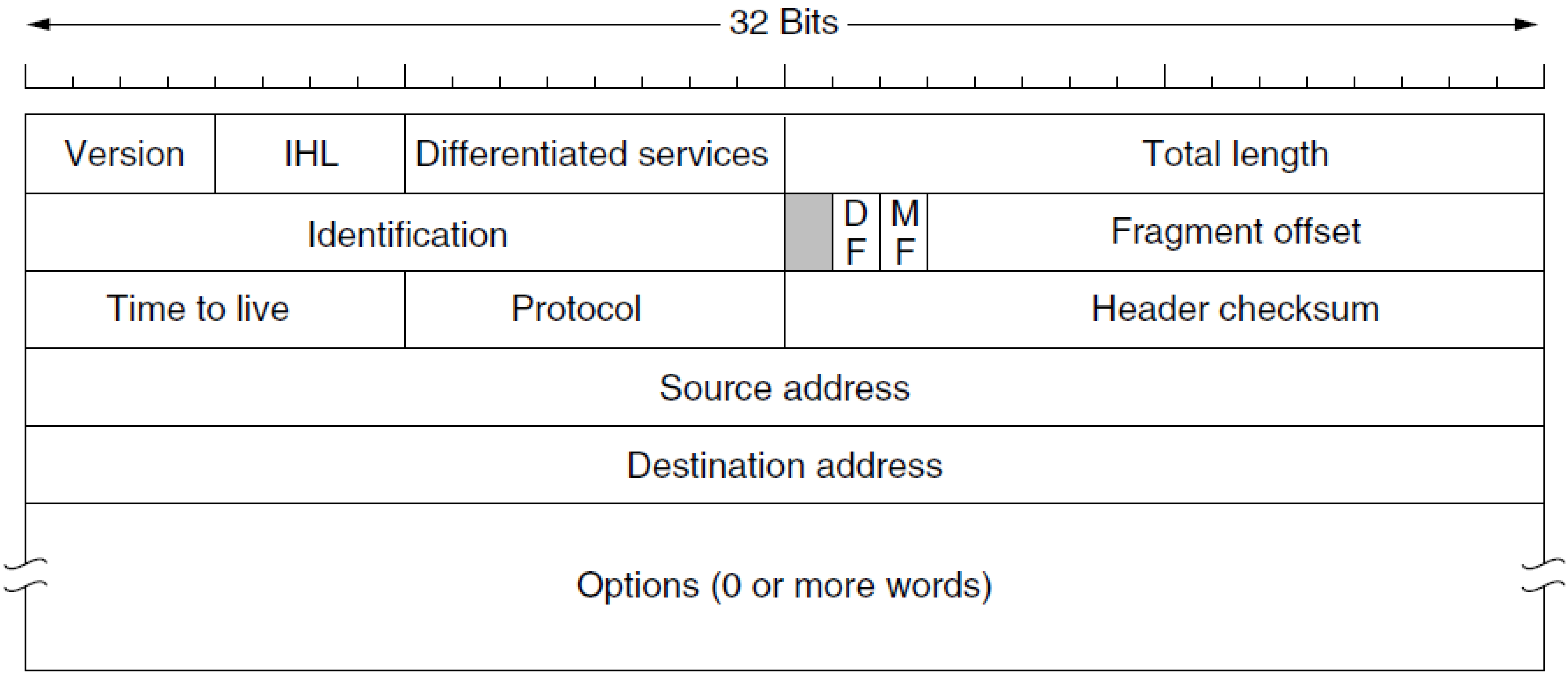
- In addition to IP, which is used for data transfer, the Internet has several companion control protocols that are used in the network layer.
  - IPv4: ICMP, ARP, and DHCP
  - IPv6: ICMP, NDP (Neighbor Discovery Protocol) and DHCP

Internet Control Message Protocol (**ICMP**) [8]

- ICMP is specified in RFC 792.
- The most typical use of ICMP is for **error reporting**.
  - For example, when running a Telnet, FTP, or HTTP session, you may have encountered an error message such as "Destination network unreachable".
- ICMP is often considered part of IP but architecturally it lies just above IP, as ICMP messages are carried inside IP datagrams.
  - The value in the protocol field will be 1.
- ICMP messages have a type and a code field, and contain the header and the first 8 bytes of the IP datagram.

The IPv4 Datagram

- The header has a 20-byte fixed part and a variable-length optional part.
- The bits are transmitted from left to right and top to bottom. This is "big-endian" network byte order.



32 Bits

Version | IHL | Differentiated services | Total length

Identification | D F | M F | Fragment offset

Time to live | Protocol | Header checksum

Source address

Destination address

Options (0 or more words)

Figure 5-46. The IPv4 (Internet Protocol) header.

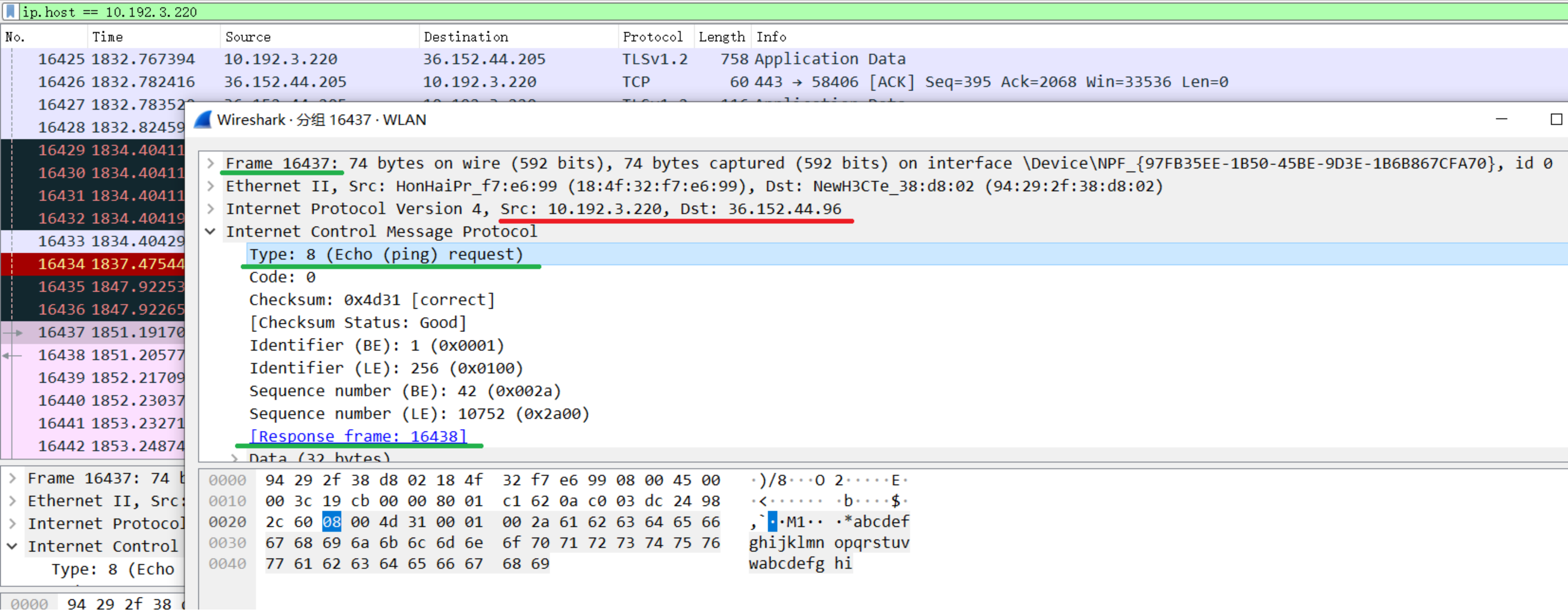
ICMP Type	Code	Description
0	0	echo reply (to ping)
3	0	destination network unreachable
3	1	destination host unreachable
3	2	destination protocol unreachable
3	3	destination port unreachable
3	6	destination network unknown
3	7	destination host unknown
4	0	source quench (congestion control)
8	0	echo request
9	0	router advertisement
10	0	router discovery
11	0	TTL expired
12	0	IP header bad

Figure 4.23 ♦ ICMP message types

Use of ICMP — "ping"

- The well-known "**ping**" program sends an ICMP **type 8 code 0** message (echo request) to the specified host.
- The destination host, seeing the echo request, sends back a **type 0 code 0** ICMP echo reply.

“Ping” Example — Echo Request



```
ip.host == 10.192.3.220

No.    Time          Source          Destination     Protocol  Length Info
 16425 1832.767394   10.192.3.220    36.152.44.205   TLSv1.2      758 Application Data
 16426 1832.782416   36.152.44.205   10.192.3.220    TCP           60 443 → 58406 [ACK] Seq=395 Ack=2068 Win=33536 Len=0
 16427 1832.78352
 16428 1832.82459
 16429 1834.40411
 16430 1834.40411
 16431 1834.40411
 16432 1834.40419
 16433 1834.40429
 16434 1837.47544
 16435 1847.92253
 16436 1847.92265
 16437 1851.19170
 16438 1851.20577
 16439 1852.21709
 16440 1852.23037
 16441 1853.23271
 16442 1853.24874
```

```
Wireshark · 分组 16437 · WLAN

> Frame 16437: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{97FB35EE-1B50-45BE-9D3E-1B6B867CFA70}, id 0
> Ethernet II, Src: HonHaiPr_f7:e6:99 (18:4f:32:f7:e6:99), Dst: NewH3CTe_38:d8:02 (94:29:2f:38:d8:02)
> Internet Protocol Version 4, Src: 10.192.3.220, Dst: 36.152.44.96
v Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0x4d31 [correct]
    [Checksum Status: Good]
    Identifier (BE): 1 (0x0001)
    Identifier (LE): 256 (0x0100)
    Sequence number (BE): 42 (0x002a)
    Sequence number (LE): 10752 (0x2a00)
    [Response frame: 16438]
  > Data (32 bytes)

0000  94 29 2f 38 d8 02 18 4f  32 f7 e6 99 08 00 45 00   ·)/8···O 2·····E·
0010  00 3c 19 cb 00 00 80 01  c1 62 0a c0 03 dc 24 98   ·<······ ·b····$·
0020  2c 60 08 00 4d 31 00 01  00 2a 61 62 63 64 65 66   ,`··M1·· ·*abcdef
0030  67 68 69 6a 6b 6c 6d 6e  6f 70 71 72 73 74 75 76   ghijklmn opqrstuv
0040  77 61 62 63 64 65 66 67  68 69                     wabcdefg hi
```

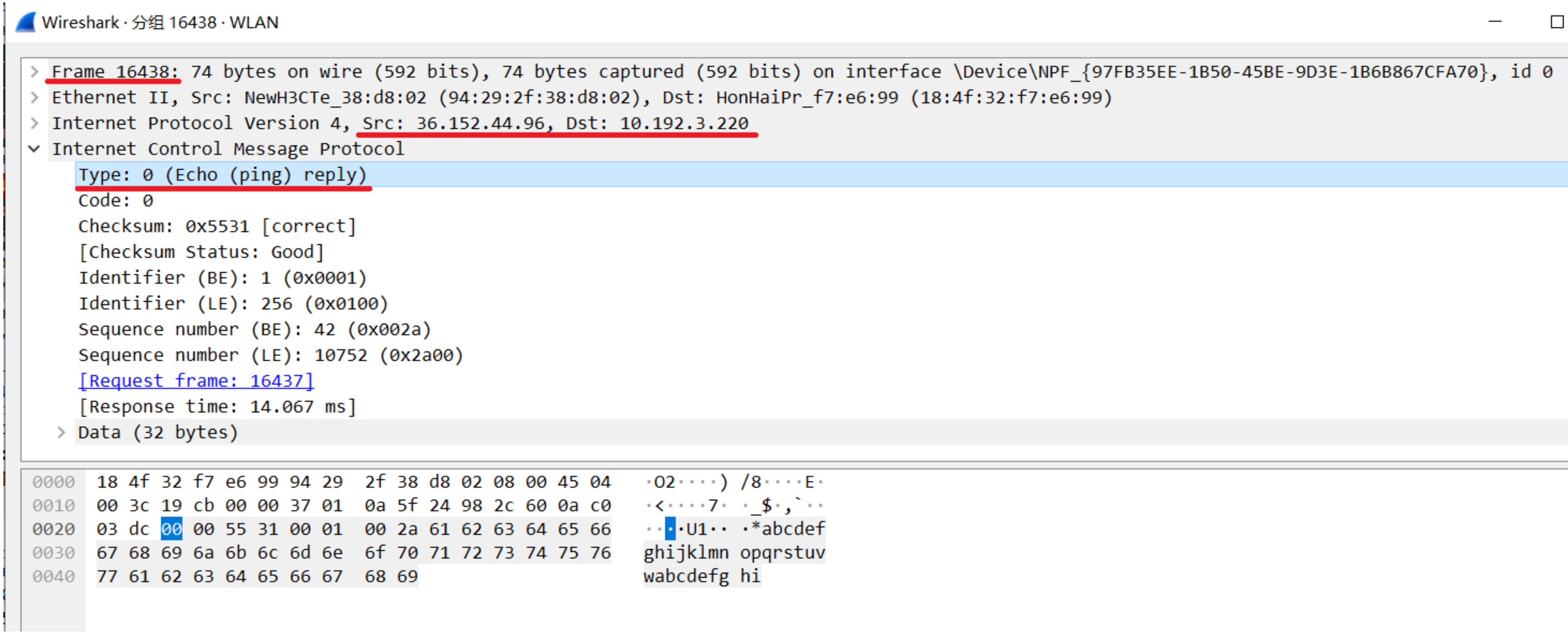
```
> Frame 16437: 74 b
> Ethernet II, Src:
> Internet Protocol
v Internet Control
    Type: 8 (Echo

0000  94 29 2f 38
```

这是我2020年9月26日晚上做的实验：我当时的IP地址为10.192.3.220，命令为“ping 36.152.44.96”（36.152.44.96是nslookup命令返回的百度域名服务器的ip地址之一，这个ip地址可能时时变化的）。

Type: 8 (Echo (ping) request)

Code: 0

“Ping” Example — Echo Reply



```
> Frame 16438: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{97FB35EE-1B50-45BE-9D3E-1B6B867CFA70}, id 0
> Ethernet II, Src: NewH3CTe_38:d8:02 (94:29:2f:38:d8:02), Dst: HonHaiPr_f7:e6:99 (18:4f:32:f7:e6:99)
> Internet Protocol Version 4, Src: 36.152.44.96, Dst: 10.192.3.220
v Internet Control Message Protocol
    Type: 0 (Echo (ping) reply)
    Code: 0
    Checksum: 0x5531 [correct]
    [Checksum Status: Good]
    Identifier (BE): 1 (0x0001)
    Identifier (LE): 256 (0x0100)
    Sequence number (BE): 42 (0x002a)
    Sequence number (LE): 10752 (0x2a00)
    [Request frame: 16437]
    [Response time: 14.067 ms]
  > Data (32 bytes)
```

```
0000  18 4f 32 f7 e6 99 94 29  2f 38 d8 02 08 00 45 04   ·O2····) /8····E·
0010  00 3c 19 cb 00 00 37 01  0a 5f 24 98 2c 60 0a c0   ·<····7· ·_$·,`··
0020  03 dc 00 00 55 31 00 01  00 2a 61 62 63 64 65 66   ····U1·· ·*abcdef
0030  67 68 69 6a 6b 6c 6d 6e  6f 70 71 72 73 74 75 76   ghijklmn opqrstuv
0040  77 61 62 63 64 65 66 67  68 69                     wabcdefg hi
```

这是我2020年9月26日晚上做的实验：我当时的IP地址为10.192.3.220，命令为“ping 36.152.44.96”（36.152.44.96是nslookup命令返回的百度域名服务器的ip地址之一，这个ip地址可能时时变化的）。

Type: 0 (Echo (ping) reply)

Code: 0

Use of ICMP — "Tracert" (I)

- In Lab1, we introduced the **Tracert** program, which allows us to trace a route from a host to any other host in the world.
- Tracert is implemented with ICMP messages, to determine the names and addresses of the routers between source and destination,
  - 1) Tracert in the source sends *a series of ordinary IP datagrams* to the destination.
  - Each of these datagrams carries a **UDP** segment with an unlikely UDP port number.
  - The 1st of these datagrams has a TTL of 1, the 2nd of 2, the 3rd of 3, and so on. The source also starts timers for each of the datagrams.

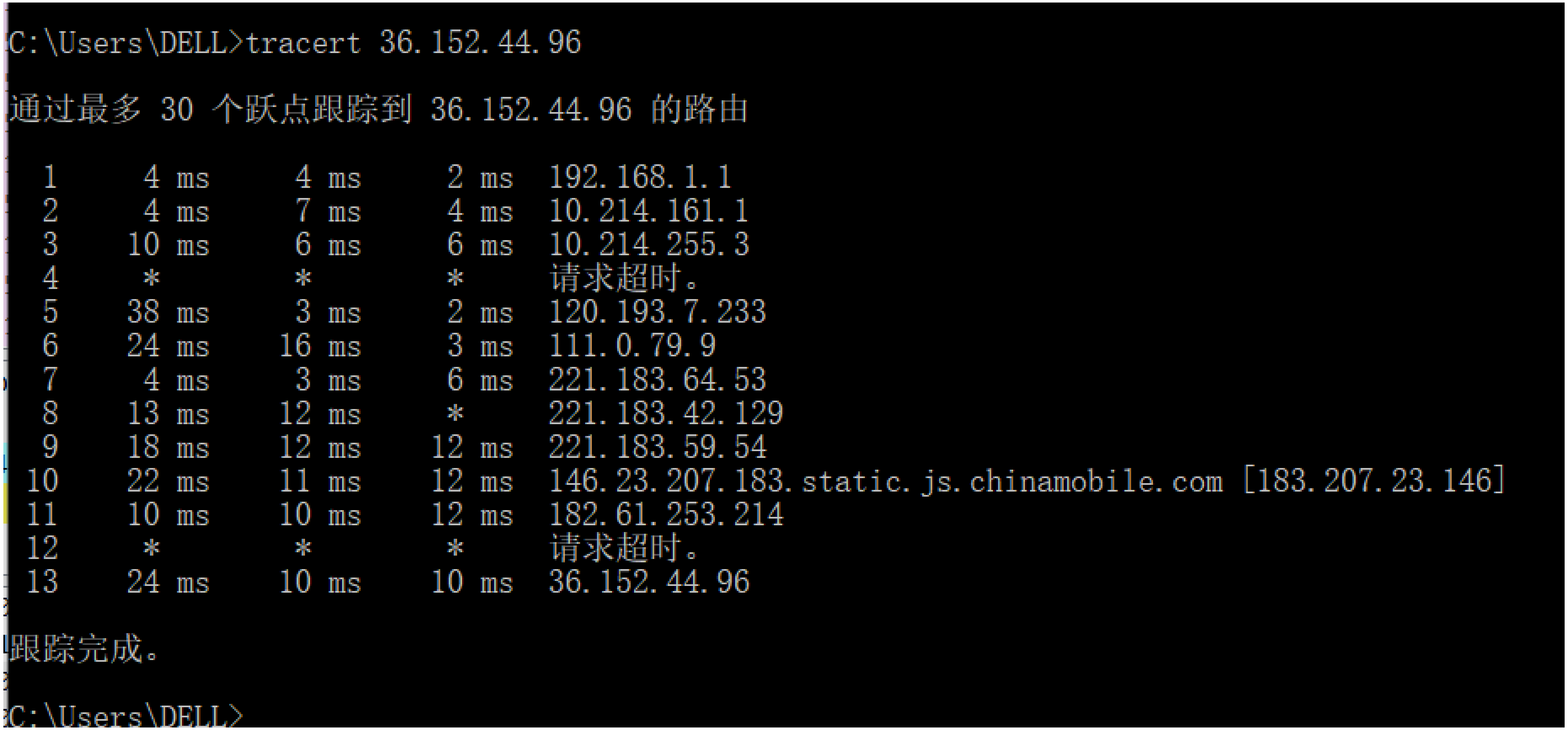
Use of ICMP — "Tracert" (II)

- Tracert is implemented with ICMP messages, to determine the names and addresses of the routers between source and destination,
  - 2) When the *n*th datagram arrives at the *n*th router, the *n*th router observes that *the TTL of the datagram has just expired*.
  - According to the rules of the IP protocol, the router discards the datagram and sends an ICMP warning message to the source (type 11 code 0)
  - This warning message includes the name of the router and its IP address.
  - 3) When this ICMP message arrives back at the source, the source obtains **the round-trip time** from the timer and the name and IP address of the *n*th router from the ICMP message

Use of ICMP — "Tracert" (III)

- Tracert is implemented with ICMP messages, to determine the names and addresses of the routers between source and destination,
  - 4) How does a Tracert source know when to **stop** sending UDP segments?
  - Recall that the source increments the TTL field for each datagram it sends. Thus, one of the datagrams will eventually make it all the way to the destination host.
  - Because this datagram contains a UDP segment with an unlikely port number, the destination host sends **a port unreachable ICMP message** (type 3 code 3) back to the source.
  - When the source host receives this particular ICMP message, it knows it does not need to send additional probe packets.
  - The standard Tracert program actually sends sets of **three** packets with the same TTL; thus the Tracert output provides three results for each TTL.

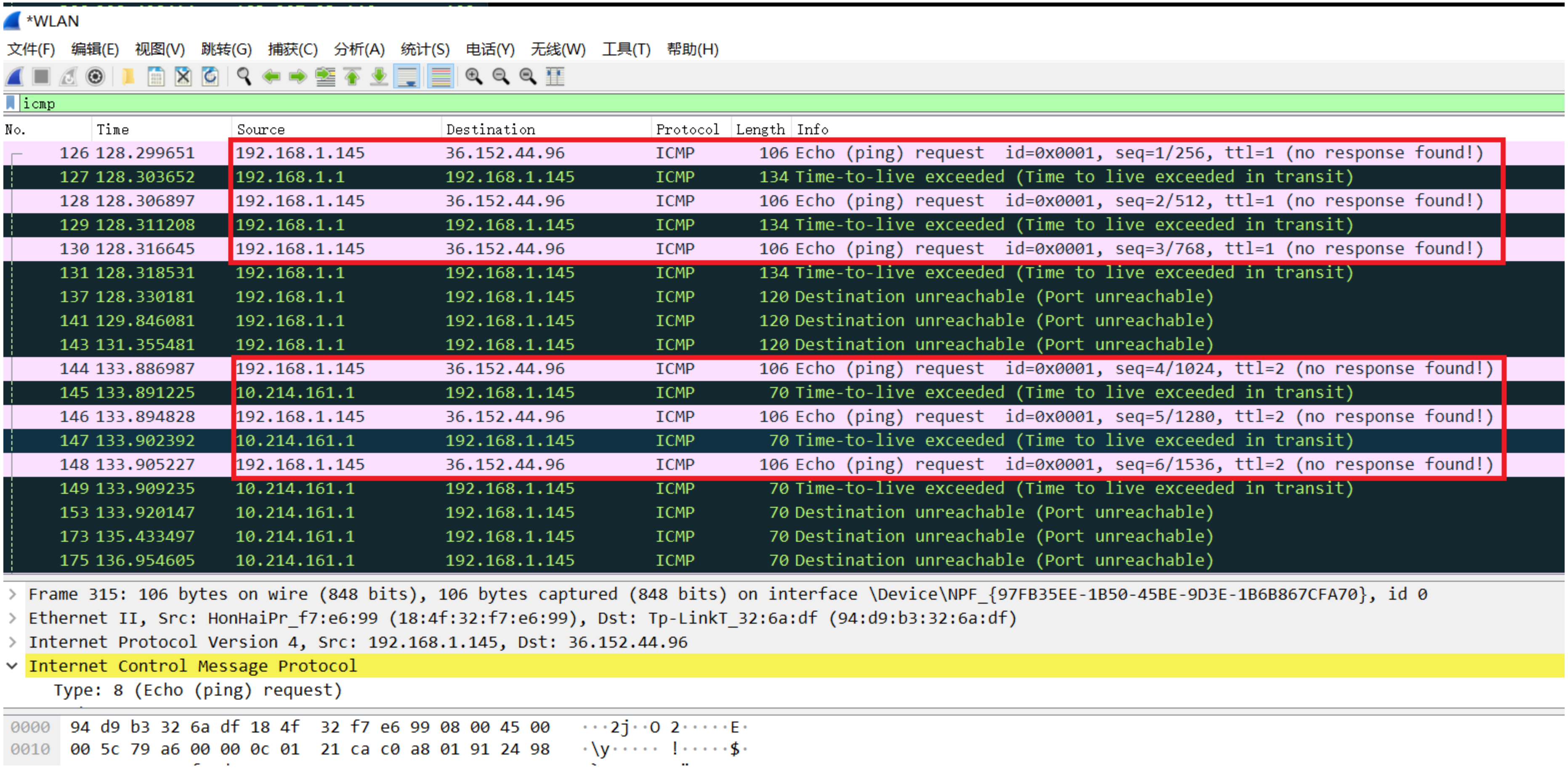
“tracert” Example (I)



```
C:\Users\DELL>tracert 36.152.44.96

通过最多 30 个跃点跟踪到 36.152.44.96 的路由

  1     4 ms     4 ms     2 ms  192.168.1.1
  2     4 ms     7 ms     4 ms  10.214.161.1
  3    10 ms     6 ms     6 ms  10.214.255.3
  4     *        *        *     请求超时。
  5    38 ms     3 ms     2 ms  120.193.7.233
  6    24 ms    16 ms     3 ms  111.0.79.9
  7     4 ms     3 ms     6 ms  221.183.64.53
  8    13 ms    12 ms     *     221.183.42.129
  9    18 ms    12 ms    12 ms  221.183.59.54
 10    22 ms    11 ms    12 ms  146.23.207.183.static.js.chinamobile.com [183.207.23.146]
 11    10 ms    10 ms    12 ms  182.61.253.214
 12     *        *        *     请求超时。
 13    24 ms    10 ms    10 ms  36.152.44.96

跟踪完成。

C:\Users\DELL>
```

这是我2020年10月27日晚上做的实验：我当时的IP地址为192.168.1.145，命令为“tracert 36.152.44.96”（ 36.152.44.96是nslookup命令返回的百度域名服务器的ip地址之一）。注意第一跳是网关192.168.1.1，每一IP address前面有三个RRT数值。

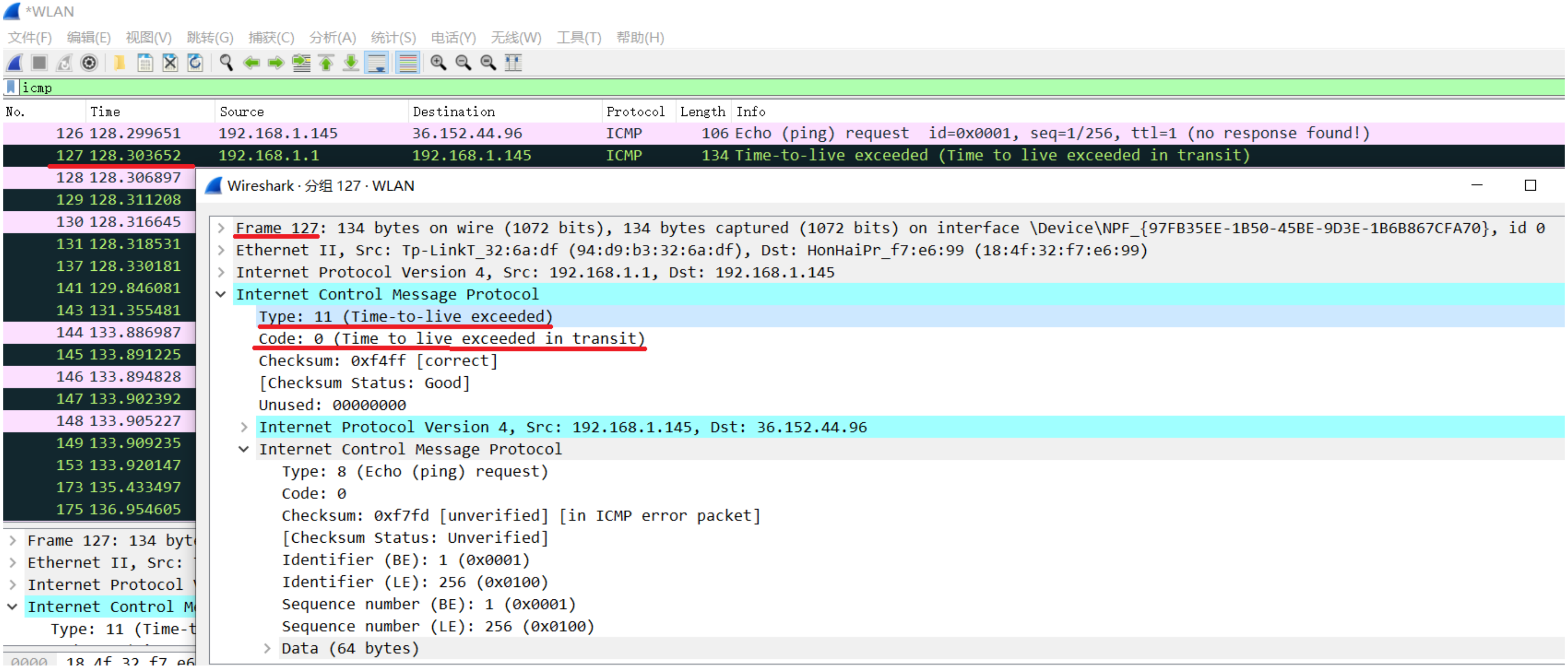
“tracert” Example (II)


*WLAN
文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)

```
icmp

No.  Time        Source         Destination    Protocol Length Info
126 128.299651  192.168.1.145  36.152.44.96   ICMP     106 Echo (ping) request  id=0x0001, seq=1/256, ttl=1 (no response found!)
127 128.303652  192.168.1.1    192.168.1.145  ICMP     134 Time-to-live exceeded (Time to live exceeded in transit)
128 128.306897  192.168.1.145  36.152.44.96   ICMP     106 Echo (ping) request  id=0x0001, seq=2/512, ttl=1 (no response found!)
129 128.311208  192.168.1.1    192.168.1.145  ICMP     134 Time-to-live exceeded (Time to live exceeded in transit)
130 128.316645  192.168.1.145  36.152.44.96   ICMP     106 Echo (ping) request  id=0x0001, seq=3/768, ttl=1 (no response found!)
131 128.318531  192.168.1.1    192.168.1.145  ICMP     134 Time-to-live exceeded (Time to live exceeded in transit)
137 128.330181  192.168.1.1    192.168.1.145  ICMP     120 Destination unreachable (Port unreachable)
141 129.846081  192.168.1.1    192.168.1.145  ICMP     120 Destination unreachable (Port unreachable)
143 131.355481  192.168.1.1    192.168.1.145  ICMP     120 Destination unreachable (Port unreachable)
144 133.886987  192.168.1.145  36.152.44.96   ICMP     106 Echo (ping) request  id=0x0001, seq=4/1024, ttl=2 (no response found!)
145 133.891225  10.214.161.1   192.168.1.145  ICMP      70 Time-to-live exceeded (Time to live exceeded in transit)
146 133.894828  192.168.1.145  36.152.44.96   ICMP     106 Echo (ping) request  id=0x0001, seq=5/1280, ttl=2 (no response found!)
147 133.902392  10.214.161.1   192.168.1.145  ICMP      70 Time-to-live exceeded (Time to live exceeded in transit)
148 133.905227  192.168.1.145  36.152.44.96   ICMP     106 Echo (ping) request  id=0x0001, seq=6/1536, ttl=2 (no response found!)
149 133.909235  10.214.161.1   192.168.1.145  ICMP      70 Time-to-live exceeded (Time to live exceeded in transit)
153 133.920147  10.214.161.1   192.168.1.145  ICMP      70 Destination unreachable (Port unreachable)
173 135.433497  10.214.161.1   192.168.1.145  ICMP      70 Destination unreachable (Port unreachable)
175 136.954605  10.214.161.1   192.168.1.145  ICMP      70 Destination unreachable (Port unreachable)

> Frame 315: 106 bytes on wire (848 bits), 106 bytes captured (848 bits) on interface \Device\NPF_{97FB35EE-1B50-45BE-9D3E-1B6B867CFA70}, id 0
> Ethernet II, Src: HonHaiPr_f7:e6:99 (18:4f:32:f7:e6:99), Dst: Tp-LinkT_32:6a:df (94:d9:b3:32:6a:df)
> Internet Protocol Version 4, Src: 192.168.1.145, Dst: 36.152.44.96
v Internet Control Message Protocol
    Type: 8 (Echo (ping) request)

0000  94 d9 b3 32 6a df 18 4f  32 f7 e6 99 08 00 45 00   ···2j··O 2·····E·
0010  00 5c 79 a6 00 00 0c 01  21 ca c0 a8 01 91 24 98   ·\y····· !·····$·
```

从WireShark抓包的结果来看：TTL= 1, ICMP包发了三次， 同样TTL= 2, ICMP包发了三次，其它设置的TTL ICMP包也一样。

“tracert” Example (III)


*WLAN
文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)
icmp
Wireshark · 分组 127 · WLAN

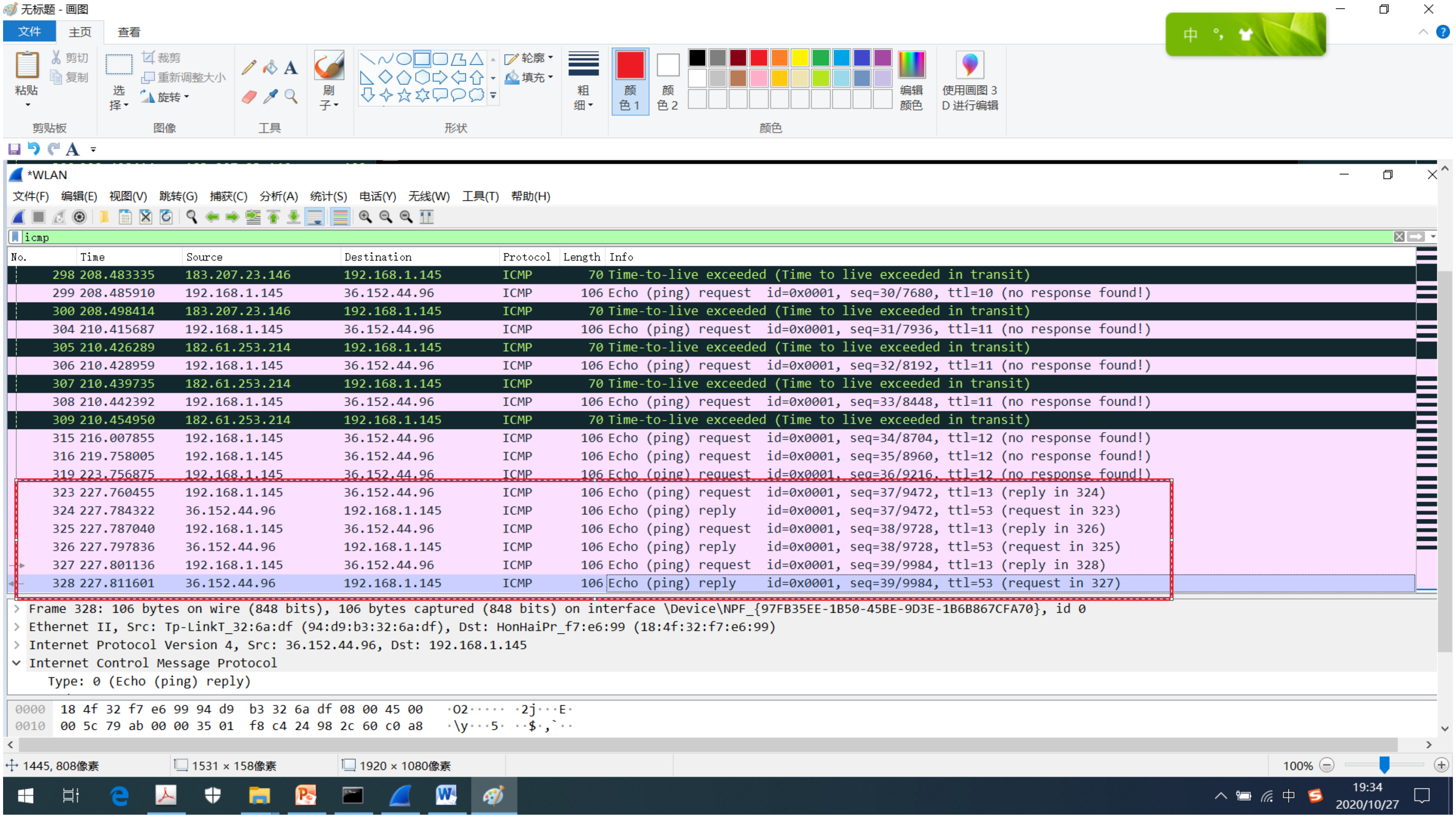
```
No.   Time         Source          Destination     Protocol  Length  Info
  126 128.299651   192.168.1.145   36.152.44.96    ICMP       106 Echo (ping) request  id=0x0001, seq=1/256, ttl=1 (no response found!)
  127 128.303652   192.168.1.1     192.168.1.145   ICMP       134 Time-to-live exceeded (Time to live exceeded in transit)
  128 128.306897
  129 128.311208
  130 128.316645
  131 128.318531
  137 128.330181
  141 129.846081
  143 131.355481
  144 133.886987
  145 133.891225
  146 133.894828
  147 133.902392
  148 133.905227
  149 133.909235
  153 133.920147
  173 135.433497
  175 136.954605
```

```
> Frame 127: 134 bytes on wire (1072 bits), 134 bytes captured (1072 bits) on interface \Device\NPF_{97FB35EE-1B50-45BE-9D3E-1B6B867CFA70}, id 0
> Ethernet II, Src: Tp-LinkT_32:6a:df (94:d9:b3:32:6a:df), Dst: HonHaiPr_f7:e6:99 (18:4f:32:f7:e6:99)
> Internet Protocol Version 4, Src: 192.168.1.1, Dst: 192.168.1.145
v Internet Control Message Protocol
    Type: 11 (Time-to-live exceeded)
    Code: 0 (Time to live exceeded in transit)
    Checksum: 0xf4ff [correct]
    [Checksum Status: Good]
    Unused: 00000000
  > Internet Protocol Version 4, Src: 192.168.1.145, Dst: 36.152.44.96
  v Internet Control Message Protocol
      Type: 8 (Echo (ping) request)
      Code: 0
      Checksum: 0xf7fd [unverified] [in ICMP error packet]
      [Checksum Status: Unverified]
      Identifier (BE): 1 (0x0001)
      Identifier (LE): 256 (0x0100)
      Sequence number (BE): 1 (0x0001)
      Sequence number (LE): 256 (0x0100)
    > Data (64 bytes)
```

```
> Frame 127: 134 byt
> Ethernet II, Src: 
> Internet Protocol 
v Internet Control M
    Type: 11 (Time-t
0000  18 4f 32 f7 e6
```

这里把序号为127的包打开，具体看里面的分组信息：TTL为1的路由器地址为192.168.1.1，当这个TTL为1的ICMP包达到192.168.1.1时，已经expired，所以192.168.1.1发送回给source (即我的电脑192.168.1.145) ICMP包中type为11，code为0，该message为：Time to live exceed in transit 。

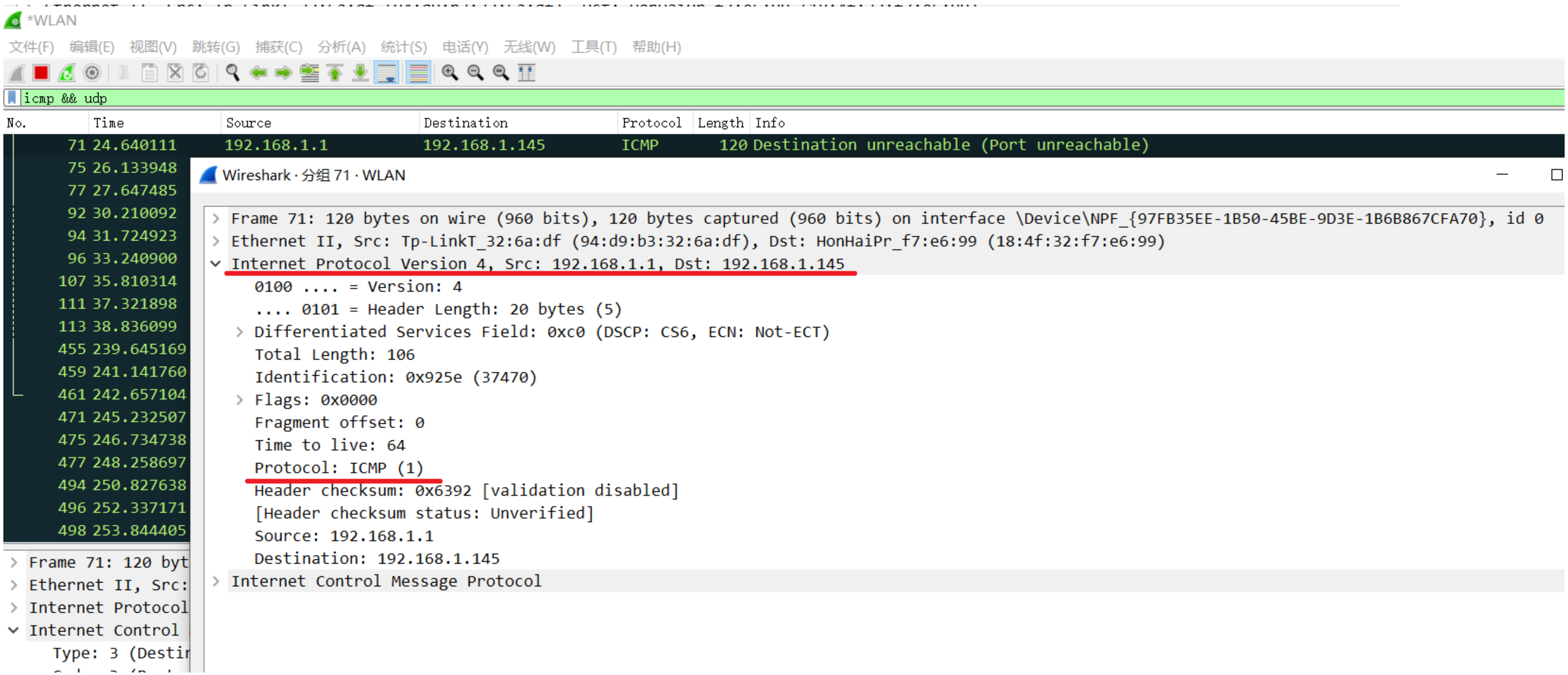
“tracert” Example (IV)



```
No.   Time        Source           Destination      Protocol Length Info
  298 208.483335  183.207.23.146   192.168.1.145    ICMP       70 Time-to-live exceeded (Time to live exceeded in transit)
  299 208.485910  192.168.1.145    36.152.44.96     ICMP      106 Echo (ping) request  id=0x0001, seq=30/7680, ttl=10 (no response found!)
  300 208.498414  183.207.23.146   192.168.1.145    ICMP       70 Time-to-live exceeded (Time to live exceeded in transit)
  304 210.415687  192.168.1.145    36.152.44.96     ICMP      106 Echo (ping) request  id=0x0001, seq=31/7936, ttl=11 (no response found!)
  305 210.426289  182.61.253.214   192.168.1.145    ICMP       70 Time-to-live exceeded (Time to live exceeded in transit)
  306 210.428959  192.168.1.145    36.152.44.96     ICMP      106 Echo (ping) request  id=0x0001, seq=32/8192, ttl=11 (no response found!)
  307 210.439735  182.61.253.214   192.168.1.145    ICMP       70 Time-to-live exceeded (Time to live exceeded in transit)
  308 210.442392  192.168.1.145    36.152.44.96     ICMP      106 Echo (ping) request  id=0x0001, seq=33/8448, ttl=11 (no response found!)
  309 210.454950  182.61.253.214   192.168.1.145    ICMP       70 Time-to-live exceeded (Time to live exceeded in transit)
  315 216.007855  192.168.1.145    36.152.44.96     ICMP      106 Echo (ping) request  id=0x0001, seq=34/8704, ttl=12 (no response found!)
  316 219.758005  192.168.1.145    36.152.44.96     ICMP      106 Echo (ping) request  id=0x0001, seq=35/8960, ttl=12 (no response found!)
  319 223.756875  192.168.1.145    36.152.44.96     ICMP      106 Echo (ping) request  id=0x0001, seq=36/9216, ttl=12 (no response found!)
  323 227.760455  192.168.1.145    36.152.44.96     ICMP      106 Echo (ping) request  id=0x0001, seq=37/9472, ttl=13 (reply in 324)
  324 227.784322  36.152.44.96     192.168.1.145    ICMP      106 Echo (ping) reply    id=0x0001, seq=37/9472, ttl=53 (request in 323)
  325 227.787040  192.168.1.145    36.152.44.96     ICMP      106 Echo (ping) request  id=0x0001, seq=38/9728, ttl=13 (reply in 326)
  326 227.797836  36.152.44.96     192.168.1.145    ICMP      106 Echo (ping) reply    id=0x0001, seq=38/9728, ttl=53 (request in 325)
  327 227.801136  192.168.1.145    36.152.44.96     ICMP      106 Echo (ping) request  id=0x0001, seq=39/9984, ttl=13 (reply in 328)
  328 227.811601  36.152.44.96     192.168.1.145    ICMP      106 Echo (ping) reply    id=0x0001, seq=39/9984, ttl=53 (request in 327)

> Frame 328: 106 bytes on wire (848 bits), 106 bytes captured (848 bits) on interface \Device\NPF_{97FB35EE-1B50-45BE-9D3E-1B6B867CFA70}, id 0
> Ethernet II, Src: Tp-LinkT_32:6a:df (94:d9:b3:32:6a:df), Dst: HonHaiPr_f7:e6:99 (18:4f:32:f7:e6:99)
> Internet Protocol Version 4, Src: 36.152.44.96, Dst: 192.168.1.145
v Internet Control Message Protocol
    Type: 0 (Echo (ping) reply)

0000  18 4f 32 f7 e6 99 94 d9  b3 32 6a df 08 00 45 00   ·O2·····  ·2j···E·
0010  00 5c 79 ab 00 00 35 01  f8 c4 24 98 2c 60 c0 a8   ·\y···5·  ··$·,`··
```

但是在实验中我并没有发现最后有**a port unreachable ICMP message** (type 3 code 3)， 我只看到最后TTL = 13 的三组ping ICMP数据包：Echo request数据包和echo reply数据包。traceroute在Windows下实现机制不同，所以我在ICMP包中也没有看到port信息。

“tracert” Example (V)


*WLAN
文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)
Wireshark · 分组 71 · WLAN

icmp && udp

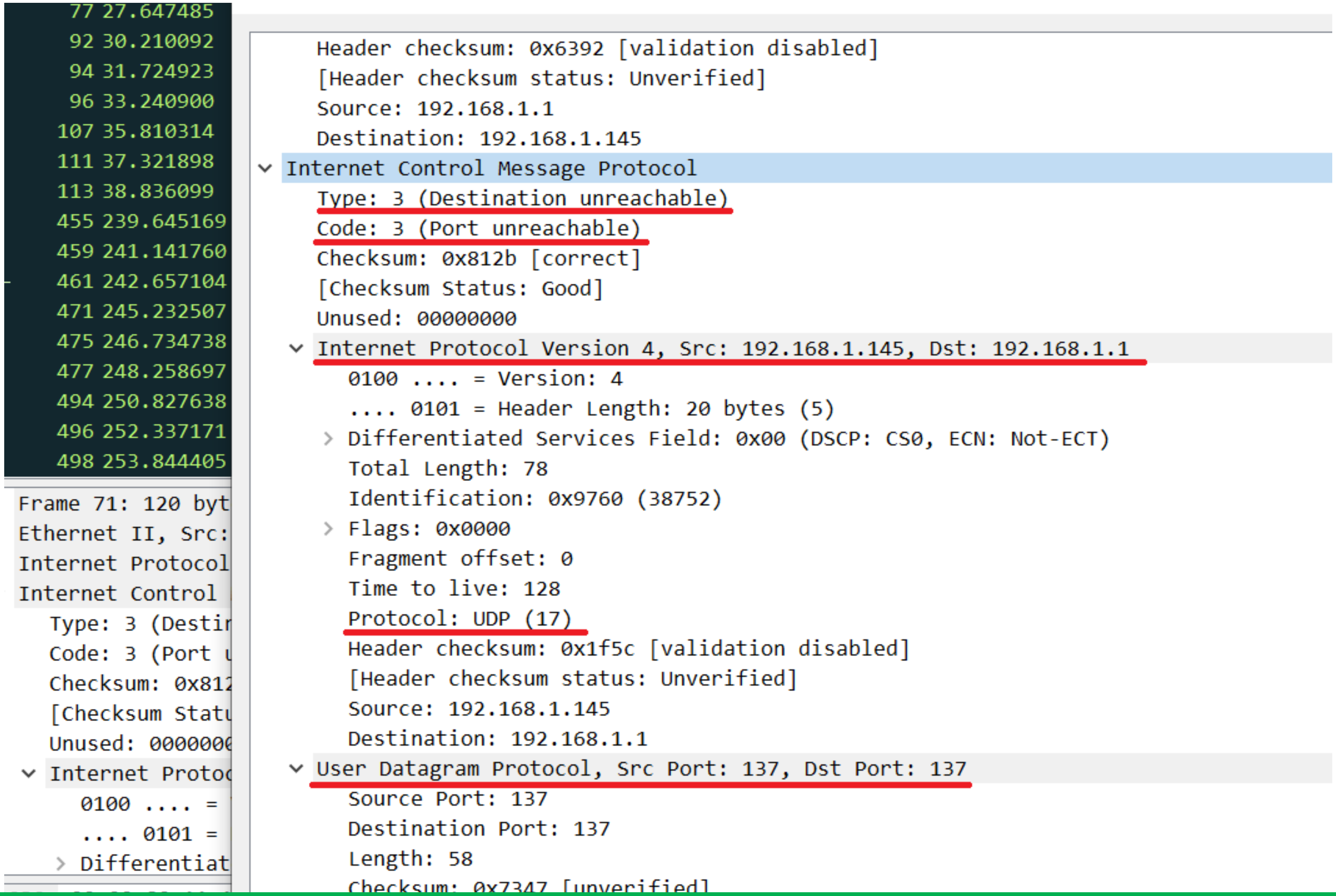
```
No.    Time         Source          Destination      Protocol  Length Info
    71 24.640111    192.168.1.1     192.168.1.145    ICMP         120 Destination unreachable (Port unreachable)
    75 26.133948
    77 27.647485
    92 30.210092
    94 31.724923
    96 33.240900
   107 35.810314
   111 37.321898
   113 38.836099
   455 239.645169
   459 241.141760
   461 242.657104
   471 245.232507
   475 246.734738
   477 248.258697
   494 250.827638
   496 252.337171
   498 253.844405
```

```
> Frame 71: 120 byt
> Ethernet II, Src:
> Internet Protocol
v Internet Control
    Type: 3 (Destir
```

```
> Frame 71: 120 bytes on wire (960 bits), 120 bytes captured (960 bits) on interface \Device\NPF_{97FB35EE-1B50-45BE-9D3E-1B6B867CFA70}, id 0
> Ethernet II, Src: Tp-LinkT_32:6a:df (94:d9:b3:32:6a:df), Dst: HonHaiPr_f7:e6:99 (18:4f:32:f7:e6:99)
v Internet Protocol Version 4, Src: 192.168.1.1, Dst: 192.168.1.145
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0xc0 (DSCP: CS6, ECN: Not-ECT)
    Total Length: 106
    Identification: 0x925e (37470)
  > Flags: 0x0000
    Fragment offset: 0
    Time to live: 64
    Protocol: ICMP (1)
    Header checksum: 0x6392 [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.1.1
    Destination: 192.168.1.145
> Internet Control Message Protocol
```

如果我把过滤器设置为”icmp && udp”，这时能抓到**a port unreachable ICMP message** (type 3 code 3)，因为分组信息很长，所以我这里分成两页来展示。可以看到ICMP信息是放在一个IPv4数据包中，IPv4数据包中协议域是 ”ICMP(1)”。

“tracert” Example (VI)



```
 77 27.647485
 92 30.210092
 94 31.724923
 96 33.240900
107 35.810314
111 37.321898
113 38.836099
455 239.645169
459 241.141760
461 242.657104
471 245.232507
475 246.734738
477 248.258697
494 250.827638
496 252.337171
498 253.844405
```

```
Frame 71: 120 byt
Ethernet II, Src:
Internet Protocol
Internet Control
   Type: 3 (Destir
   Code: 3 (Port u
   Checksum: 0x812
   [Checksum Statu
   Unused: 0000000
 ˅ Internet Protoc
      0100 .... =
      .... 0101 =
    › Differentiat
```

```
    Header checksum: 0x6392 [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.1.1
    Destination: 192.168.1.145
˅ Internet Control Message Protocol
    Type: 3 (Destination unreachable)
    Code: 3 (Port unreachable)
    Checksum: 0x812b [correct]
    [Checksum Status: Good]
    Unused: 00000000
  ˅ Internet Protocol Version 4, Src: 192.168.1.145, Dst: 192.168.1.1
      0100 .... = Version: 4
      .... 0101 = Header Length: 20 bytes (5)
    › Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
      Total Length: 78
      Identification: 0x9760 (38752)
    › Flags: 0x0000
      Fragment offset: 0
      Time to live: 128
      Protocol: UDP (17)
      Header checksum: 0x1f5c [validation disabled]
      [Header checksum status: Unverified]
      Source: 192.168.1.145
      Destination: 192.168.1.1
  ˅ User Datagram Protocol, Src Port: 137, Dst Port: 137
      Source Port: 137
      Destination Port: 137
      Length: 58
      Checksum: 0x7347 [unverified]
```

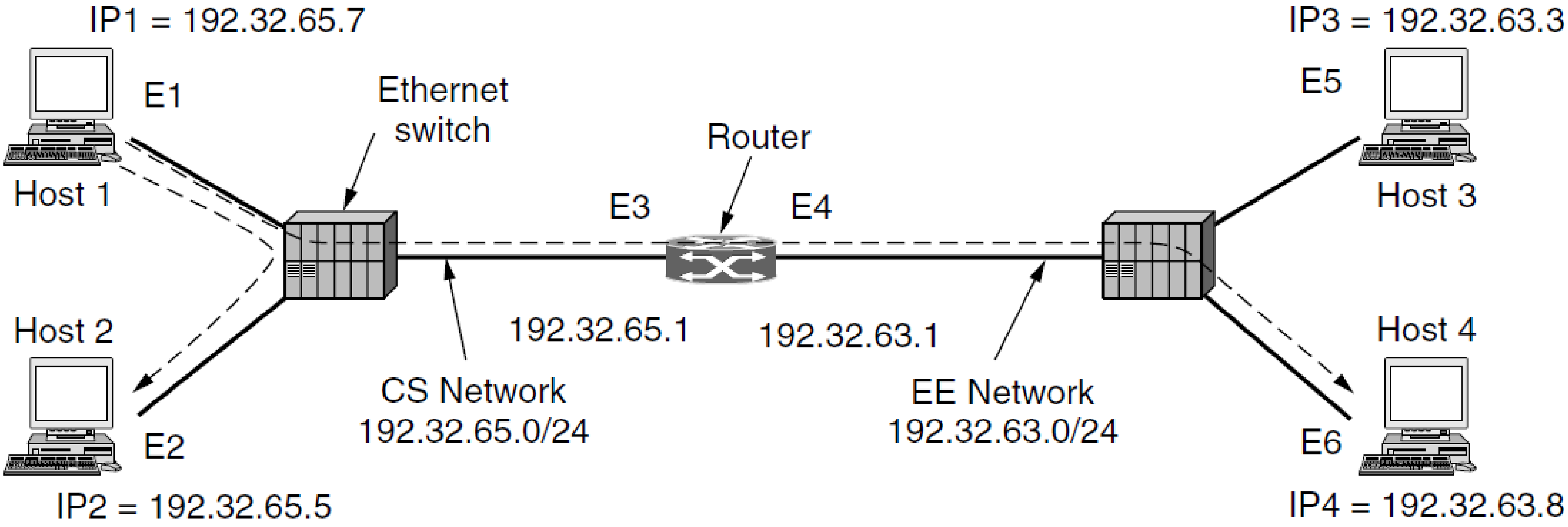
展开ICMP报文信息， 能看到ICMP type: 3, code: 3 **port unreachable**， 有意思的是ICMP报文中还藏着一个IPv4数据包，这时IPv4数据包中协议部分是”UDP (17)”，和一个UDP数据包：Src Port: 137, Dst Port: 137。(137端口的主要作用是在局域网中提供计算机的名字或IP地址查询服务)

ARP (The Address Resolution Protocol)

- Although every machine on the Internet has one or more IP addresses, these addresses are not sufficient for sending packets.
  - Data link layer NICs (Network Interface Cards) such as Ethernet cards do **NOT** understand Internet addresses.
  - The NICs send and receive frames based on 48-bit **Ethernet addresses** (the link layer addresses, that is **MAC addresses**).
- Now the question is: how do IP addresses get mapped onto data link layer addresses, such as Ethernet?
  - For the Internet, this is the job of the **Address Resolution Protocol** (**ARP**) [RFC826].
  - The purpose of the ARP query packet is to query all the other nodes *on the subnet* to determine the MAC address corresponding to the IP address that is being resolved.

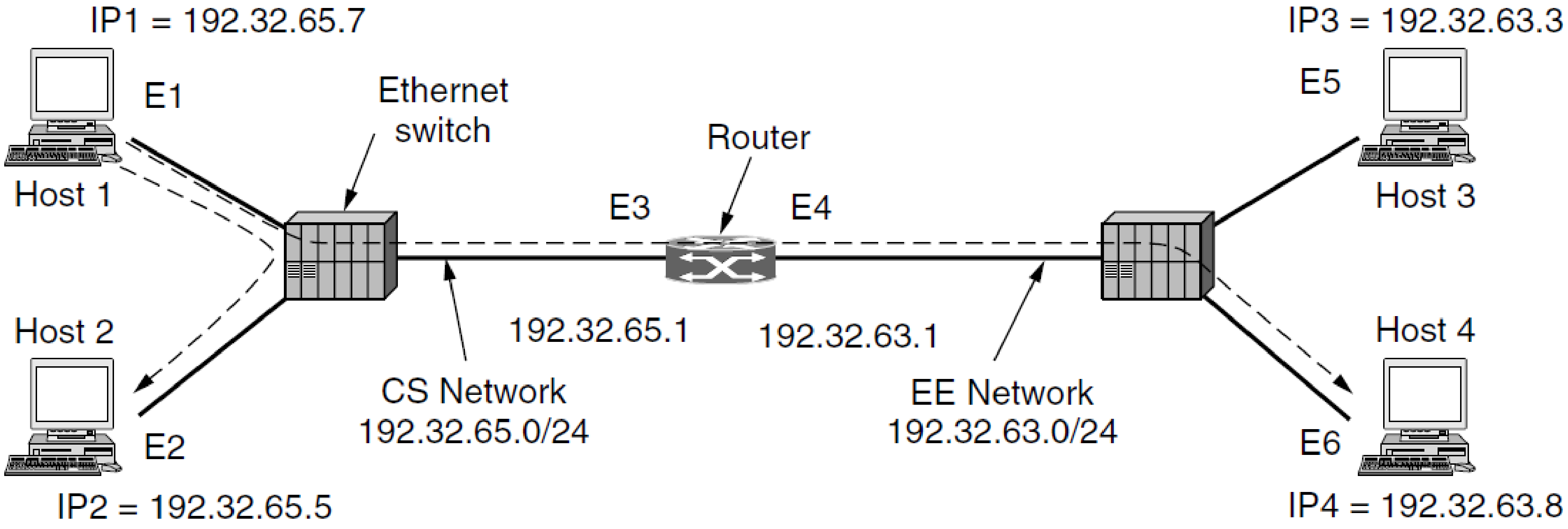
ARP: Example (I)

- Each machine on an Ethernet and each interface on the router has a **unique** Ethernet address (MAC address), and a unique IP address on the Internet (when not consider the NAT).



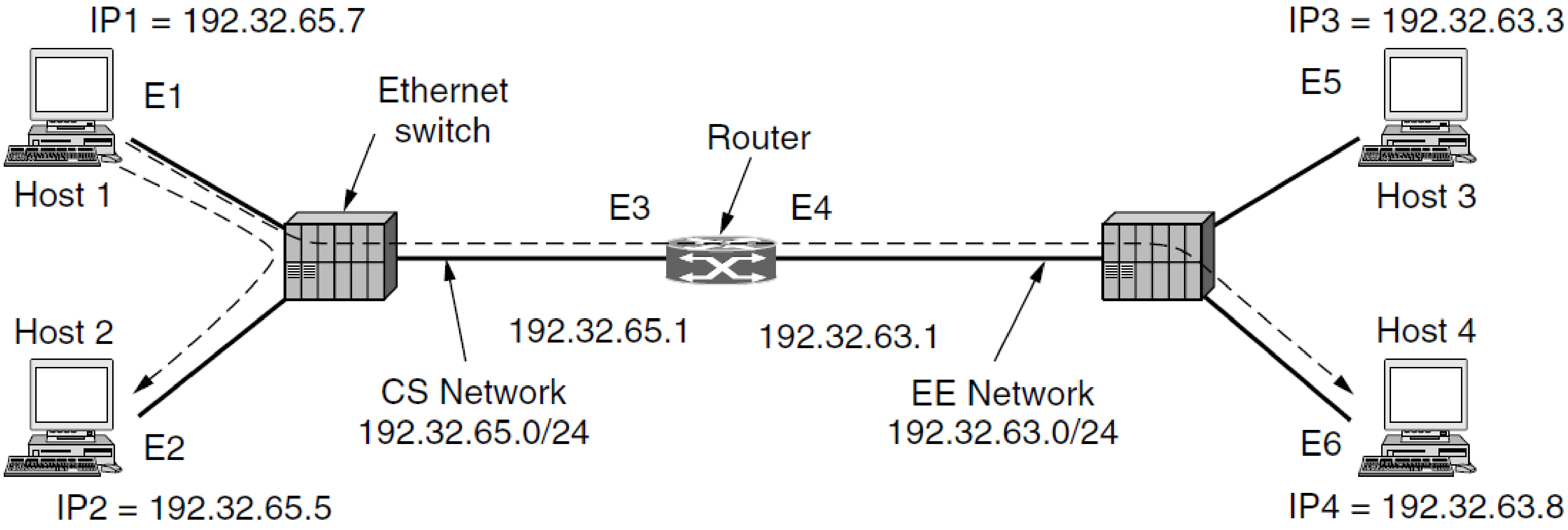
ARP: Example (II)

- How a user on host 1 sends a packet to a user on host 2 on the CS network?
  - 1) To find the IP address for host 2, suppose host 1 knows the name of host 2. This lookup is performed by DNS (chapter 7).
    - For example in Lab 1: **nslookup** www.baidu.com
  - 2) Host 1 output a **broadcast** packet on the Ethernet asking who owns IP address 192.32.65.5.
    - A special MAC broadcast address: ff:ff:ff:ff:ff:ff

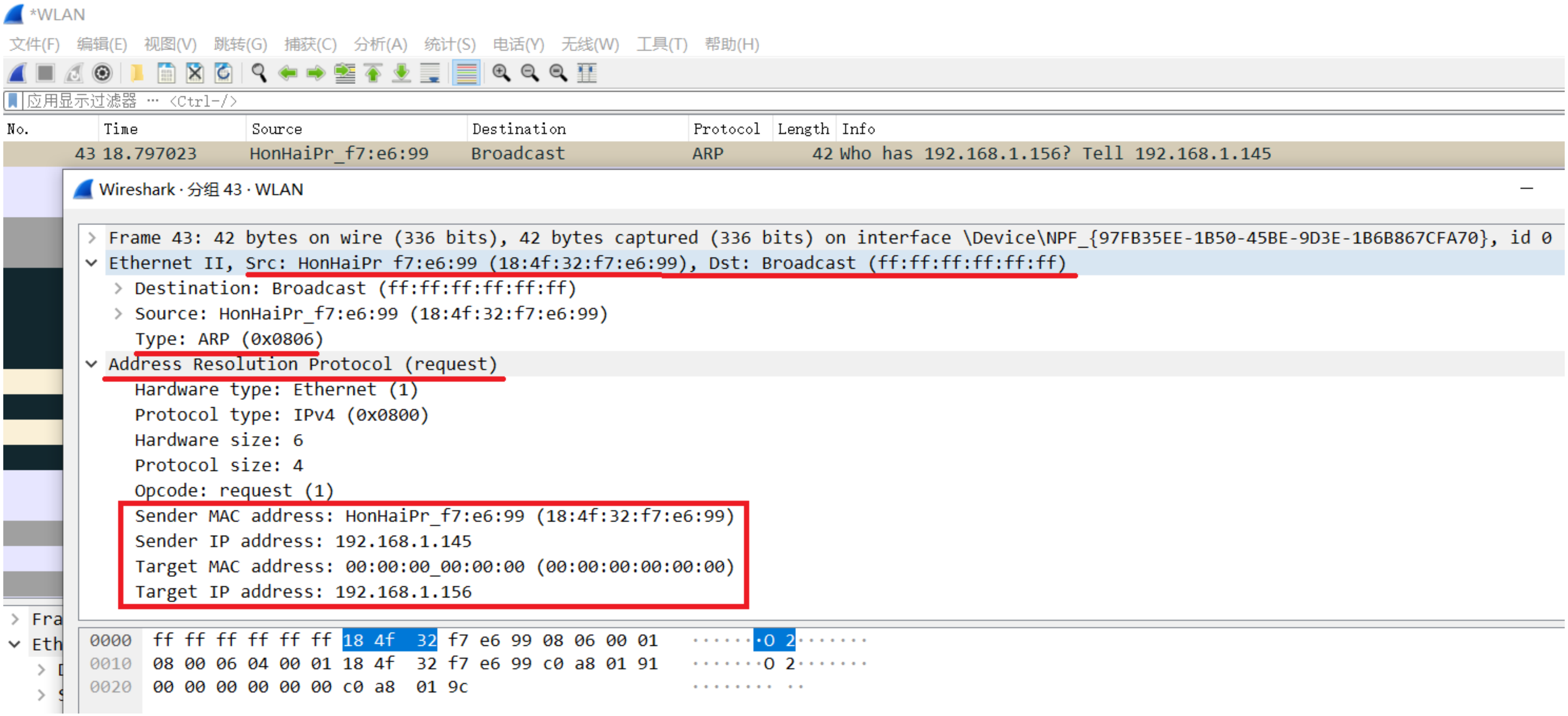


ARP: Example (III)

- How a user on host 1 sends a packet to a user on host 2 on the CS network?
  - 3) Host 2 alone will respond with its Ethernet address (E2). In this way host 1 learns that IP address 192.32.65.5 is on the host with Ethernet address (E2).
  - 4) Host 1 builds an Ethernet frame address to E2, puts the IP packet (addressed to 192.32.65.5) in the payload field, and dumps it onto the Ethernet)



ARP Request Example


*WLAN
文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)
应用显示过滤器 … <Ctrl-/>
Wireshark · 分组 43 · WLAN
Fra
Eth

```
No.    Time        Source               Destination      Protocol  Length  Info
    43 18.797023   HonHaiPr_f7:e6:99    Broadcast        ARP           42  Who has 192.168.1.156? Tell 192.168.1.145
```

```
> Frame 43: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface \Device\NPF_{97FB35EE-1B50-45BE-9D3E-1B6B867CFA70}, id 0
v Ethernet II, Src: HonHaiPr_f7:e6:99 (18:4f:32:f7:e6:99), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  > Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  > Source: HonHaiPr_f7:e6:99 (18:4f:32:f7:e6:99)
    Type: ARP (0x0806)
v Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: HonHaiPr_f7:e6:99 (18:4f:32:f7:e6:99)
    Sender IP address: 192.168.1.145
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 192.168.1.156
```

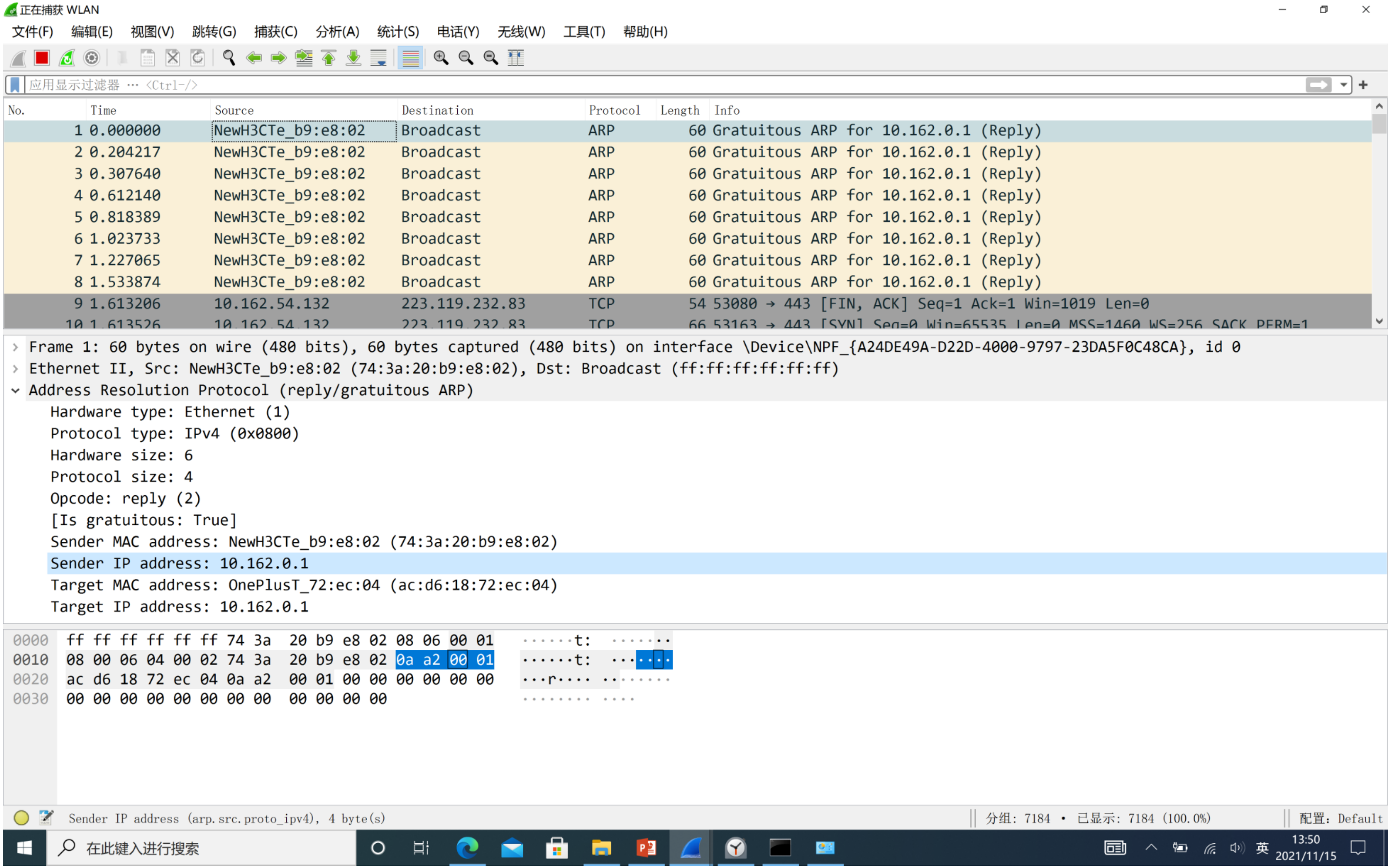
```
0000  ff ff ff ff ff ff 18 4f  32 f7 e6 99 08 06 00 01   ······O 2·······
0010  08 00 06 04 00 01 18 4f  32 f7 e6 99 c0 a8 01 91   ·······O 2·······
0020  00 00 00 00 00 00 c0 a8  01 9c                     ········ ··
```

在本次实验中电脑发出ARP广播包“Who has 192.168.1.156?”实验中发问的host’s IP address 是192.168.1.145， 其Ethernet address (MAC address)为18:4f:32:f7:e6:99。目标host’s IP address是192.168.1.156， 其MAC地址初始化为00:00:00:00:00:00。

Various Optimizations of ARP (I)

- 1) Cache
  - Once a machine as run ARP, it caches the result in case it needs to contact the same machine shortly. Each node (host and router) has **an ARP table** in its memory, which contains mappings of IP addresses to MAC addresses. The ARP table contains a time-to-live (**TTL**) value, which indicates when each mapping will be deleted from the table.
  - In many cases, host 2 will need to send back a reply, forcing it, too, to run ARP to determine the sender's Ethernet address. This ARP broadcast can be avoided by having host 1 include its IP-to-Ethernet mapping in the ARP packet. When the ARP broadcast arrives at host 2, the pair (192.32.65.7, E1) is entered into host 2's ARP cache. In fact, all machines on the Ethernet can enter this mapping into their ARP caches.
  - To allow mappings to change, for example, when a host is configured to use a new IP address (but keeps its Ethernet address), it broadcast an ARP looking for its own IP address. There should be no response, but a side effect of the broadcast is to make or update an entry in everyone's ARP cache.
    - Gratuitous ARP (无谓的ARP)

Gratuitous ARP



```
No.   Time      Source              Destination     Protocol  Length Info
    1 0.000000  NewH3CTe_b9:e8:02   Broadcast       ARP           60 Gratuitous ARP for 10.162.0.1 (Reply)
    2 0.204217  NewH3CTe_b9:e8:02   Broadcast       ARP           60 Gratuitous ARP for 10.162.0.1 (Reply)
    3 0.307640  NewH3CTe_b9:e8:02   Broadcast       ARP           60 Gratuitous ARP for 10.162.0.1 (Reply)
    4 0.612140  NewH3CTe_b9:e8:02   Broadcast       ARP           60 Gratuitous ARP for 10.162.0.1 (Reply)
    5 0.818389  NewH3CTe_b9:e8:02   Broadcast       ARP           60 Gratuitous ARP for 10.162.0.1 (Reply)
    6 1.023733  NewH3CTe_b9:e8:02   Broadcast       ARP           60 Gratuitous ARP for 10.162.0.1 (Reply)
    7 1.227065  NewH3CTe_b9:e8:02   Broadcast       ARP           60 Gratuitous ARP for 10.162.0.1 (Reply)
    8 1.533874  NewH3CTe_b9:e8:02   Broadcast       ARP           60 Gratuitous ARP for 10.162.0.1 (Reply)
    9 1.613206  10.162.54.132       223.119.232.83  TCP           54 53080 → 443 [FIN, ACK] Seq=1 Ack=1 Win=1019 Len=0
   10 1.613526  10.162.54.132       223.119.232.83  TCP           66 53163 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
```

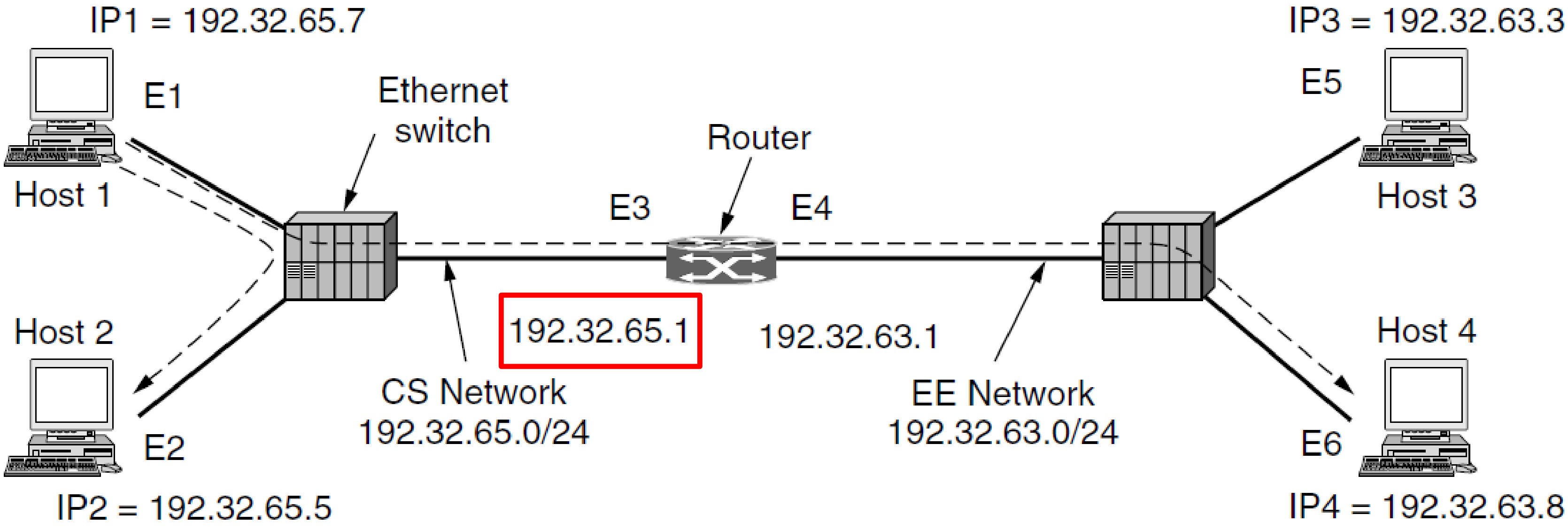
```
> Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{A24DE49A-D22D-4000-9797-23DA5F0C48CA}, id 0
> Ethernet II, Src: NewH3CTe_b9:e8:02 (74:3a:20:b9:e8:02), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
v Address Resolution Protocol (reply/gratuitous ARP)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    [Is gratuitous: True]
    Sender MAC address: NewH3CTe_b9:e8:02 (74:3a:20:b9:e8:02)
    Sender IP address: 10.162.0.1
    Target MAC address: OnePlusT_72:ec:04 (ac:d6:18:72:ec:04)
    Target IP address: 10.162.0.1
```

```
0000  ff ff ff ff ff ff 74 3a  20 b9 e8 02 08 06 00 01   ······t:  ······
0010  08 00 06 04 00 02 74 3a  20 b9 e8 02 0a a2 00 01   ······t:  ······
0020  ac d6 18 72 ec 04 0a a2  00 01 00 00 00 00 00 00   ···r····  ········
0030  00 00 00 00 00 00 00 00  00 00 00 00               ········  ····
```

Sender IP address (arp.src.proto_ipv4), 4 byte(s)   分组: 7184 · 已显示: 7184 (100.0%)   配置: Default

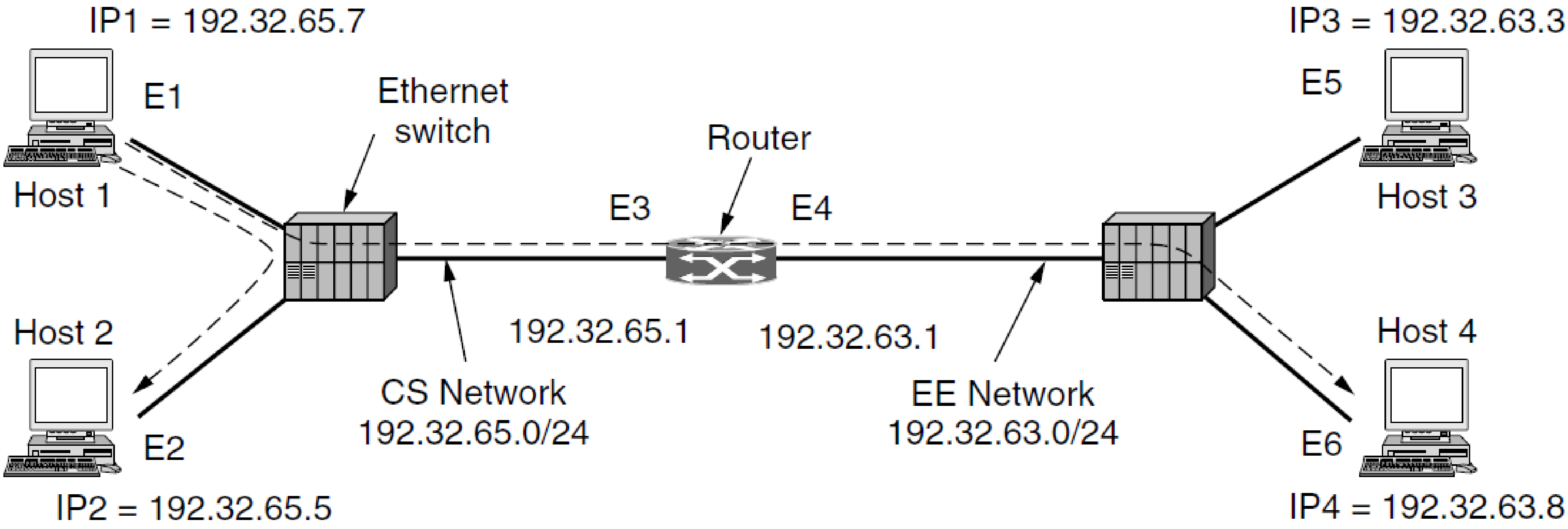
Various Optimizations of ARP (II)

- 2) The default gateway
  - For example, this time assume that host 1 wants to send a packet to host 4 (192.32.63.8) on the EE network. Host 1 will see that the destination IP address is not on the CS network. It knows to send all such off-network traffic to **the default gateway** (a router).
  - By convention, the default gateway has the lowest address on the network (192.32.65.1)



Through the Default Gateway

- Step 1: If host 1 does not know the MAC address of the default gateway (router) with the IP address (192.32.65.1), it discovers it by sending an ARP broadcasting packet, and find the MAC address (the Ethernet address) of the default gateway is E3.
- Step 2: Host 1 sends a frame (Src 192.32.65.7, E1; Dst 192.32.63.8, E3) to **the default gateway**. When the router gets this frame, it gives the packet to the IP software. It knows from the network masks that the packet should be sent onto the EE network where it will reach host 4.
- Step 3: If the router does not know the MAC address of host 4, it will use ARP again (The MAC address of host 4 is E6).
- Step 4: The router sends a frame (Src 192.32.65.7, E4; Dst 192.32.63.8, E6)



Frame	Source IP	Source Eth.	Destination IP	Destination Eth.
Host 1 to 2, on CS net	IP1	E1	IP2	E2
Host 1 to 4, on CS net	IP1	E1	IP4	E3
Host 1 to 4, on EE net	IP1	E4	IP4	E6

Figure 5-61. Two switched Ethernet LANs joined by a router.

Observe that the Ethernet address change with the frame on each network while the IP addresses remain constant (because they indicate the endpoints across all of the interconnected networks).

Various Optimizations of ARP (III)

- 3) Proxy ARP
  - It is also possible to send a packet from host 1 to host 4 without host 1 knowing that host 4 is on a different network.
  - The solution is to have the router answer ARPs on the CS network for host 4 and give its MAC address, E3, as the response.
  - It is not possible to have host 4 reply directly because it will not see the ARP request (as **routers do not forward Ethernet-level broadcasts**). The router will then receive frames sent to 192.32.63.8 and forward them onto the EE network.

ARP vs. DNS

- ARP vs. DNS
  - ARP resolves an IP address to a MAC address only for nodes on the same subnet.
  - DNS resolves host names to IP addresses for hosts anywhere in the Internet.
- ARP is probably best considered a protocol that straddles the boundary *between the link and network layers*.

Outline

- Overview of network layer
- Routing algorithms
- The network layer in the Internet
  - IP Protocol
  - Control Protocols
  - Routing Protocols
- MPLS (Multiprotocol Label Switching)

Outline

- Overview of network layer
- Routing algorithms
- The network layer in the Internet
  - IP Protocol
  - Control Protocols
  - Routing Protocols
    - RIP
    - OSPF
    - BGP
- MPLS (Multiprotocol Label Switching)

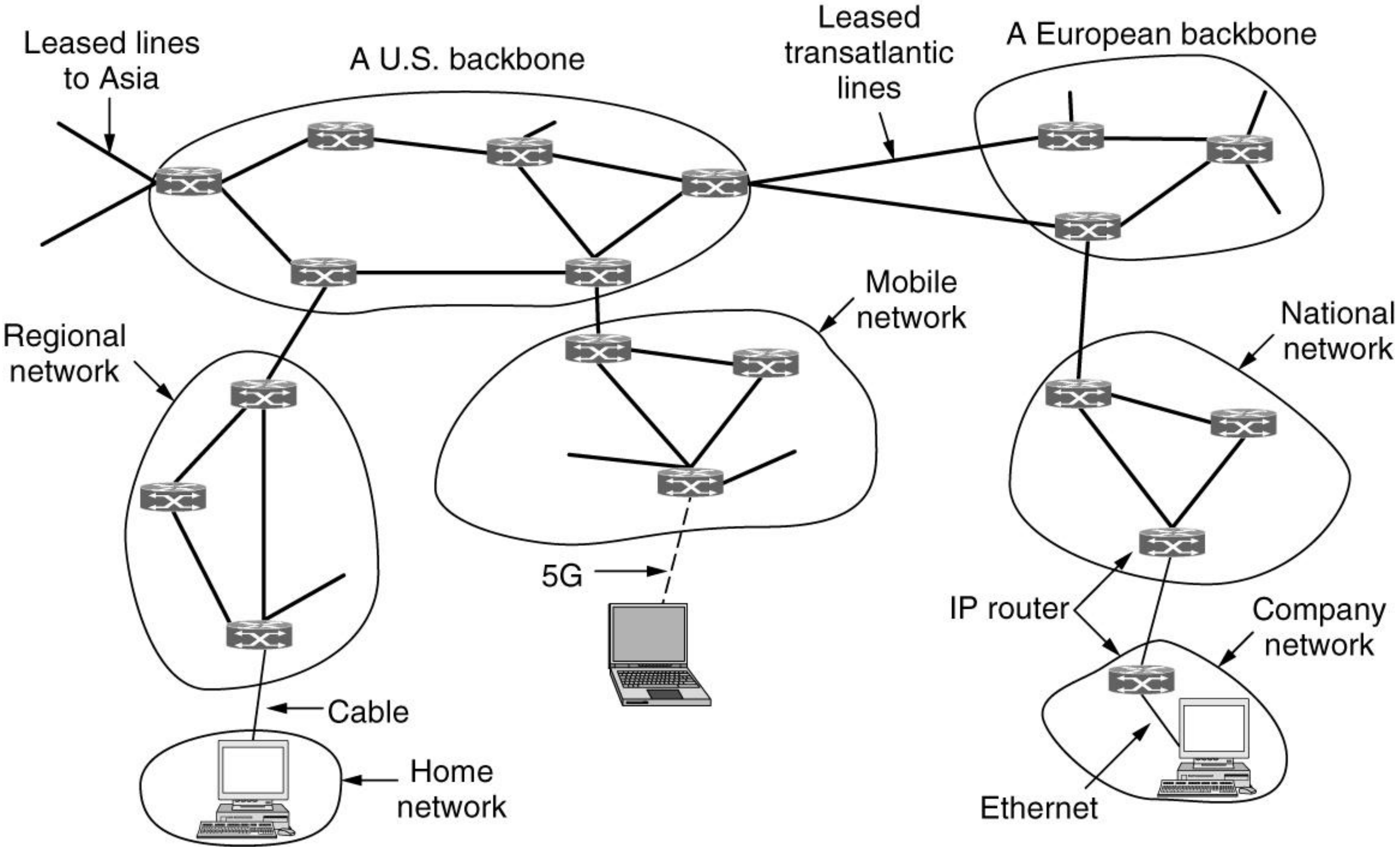
Routing in the Internet

- A two-level routing algorithm
  - Within each network, an intradomain or interior gateway protocol is used for routing.
    - Distance vector routing
    - Link state routing
  - Across the networks that make up the internet, an interdomain or exterior gateway protocol is used.
  - The networks may all use different intradomain protocols, but they must use the same interdomain protocol.
    - In the Internet, the interdomain routing protocol is called **BGP** (Border Gateway Protocol)
    - Each ISP may charge or receive money from the other ISPs for carrying traffic.

The Internet

- In the network layer, the Internet can be viewed as a collection of networks or **AS**es (Autonomous Systems) that are interconnected.
  - There is no real structure, but several major backbones exist.
  - These are constructed from high-bandwidth lines and fast routers.
  - The biggest of these backbones are called **the Tier 1 networks**.
  - Attached to the backbones are ISPs (Internet Service Providers) that provide Internet access to homes and businesses, data centers and colocation facilities full of server machines, and regional (mid-level) networks.
  - Attached to the region networks are more ISPs, LANs at many universities and companies, and other edge networks.
- The glue that holds the whole Internet together is the network layer protocol, **IP** (**Internet Protocol**).
  - In theory, IP packets can be up to 64 KB each, but in practice they are usually not more than 1500 bytes (so they fit in one Ethernet frame).

The Internet



Different Devices Connect Networks

- Repeater and hubs — physical layer
  - Analogy devices and just move bits from one wire to another.
- Bridges and switches — data link layer
  - Only minor protocol translation in the process
- Routers — network layer
- Gateways — transportation layer

Routers vs. Switches (Bridges)

- With a router, the packet is extracted from the frame and **the network address** in the packet is used for deciding where to send it.
- With a switch (or bridge), the entire frame is transported on the basis of **its MAC address**.
- Switches do not have to understand the network layer protocol being used to switch packet. Routers do.
- Today, bridges are predominantly used to connect the same kind of network at the link layer, and routers connect different networks at the network layer.

Routing in the Internet [8]

- In our study of LS and DV algorithms, we have viewed the network simply as a collection of interconnected routers. One router was indistinguishable from another in the sense that all routers executed the same routing algorithm to computing routing paths through the entire network.
- In practice, this model and its view of a homogeneous set of routers all executing the same routing algorithm is too simplistic for at least **two important causes**:
  - **Scale**: A LS algorithm needs to store all routing information and to broadcast LS updates among all of the routers; while a DV algorithm iterated among a large number of routers would surely never converged.
  - **Administrative autonomy**: an organization should be able to run and administer its network as it wishes, while still being able to connect its network to other outside networks.

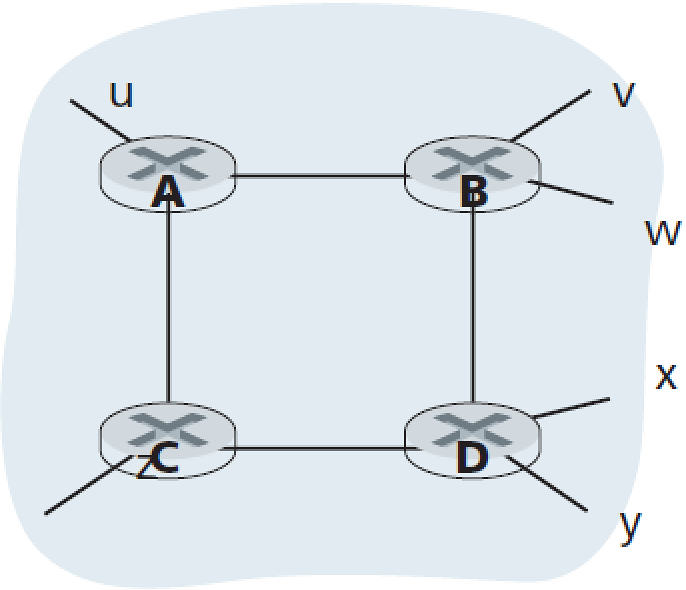
Autonomous System

- Both the scale and administrative autonomy can be solved by organizing routers into autonomous systems (ASes).
  - Hierarchical routing
- Routers *within the same autonomous system* (AS) all run the same routing algorithm (for example, an LS or DV algorithm) and have information about each other.
  - Intra-autonomous system routing protocol
- To connect ASs to each other, one or more of the routers in an AS will have the added task of being responsible for forwarding packets to destination outside the AS.
  - These routers are called **gateway routers**.
- In the following, we will examine two intra-AS routing protocols (**RIP** and **OSPF**) and the inter-AS routing protocol (**BGP**) that are used in today's Internet.

Intra-AS Routing in the Internet: RIP

- RIP (Routing Information Protocol) is **a distance-vector protocol**. [RFC 1058, RFC 2453]
- RIP uses **hop count** as a cost metric.
  - RIP uses the term **hop**, which is **the number of subnets** traversed along the shortest path from source router to destination subnet, including the destination subnet.
  - The maximum cost of a path is limited to **15**, thus limiting the use of RIP to autonomous systems that are fewer than 15 hops in diameter.
  - In RIP, routing updates are exchanged between neighbors approximately every 30 seconds using a **RIP response message**.
    - The response message sent by a router or host contains a list of up to 15 destination subnets within the AS, as well as the sender's distance to each of those subnets.
    - The response messages are also known as **RIP advertisement**.
  - RIP is implemented as an application-layer protocol running over **UDP**. Routers send RIP request and response message to each other over **UDP** using port number **520**.

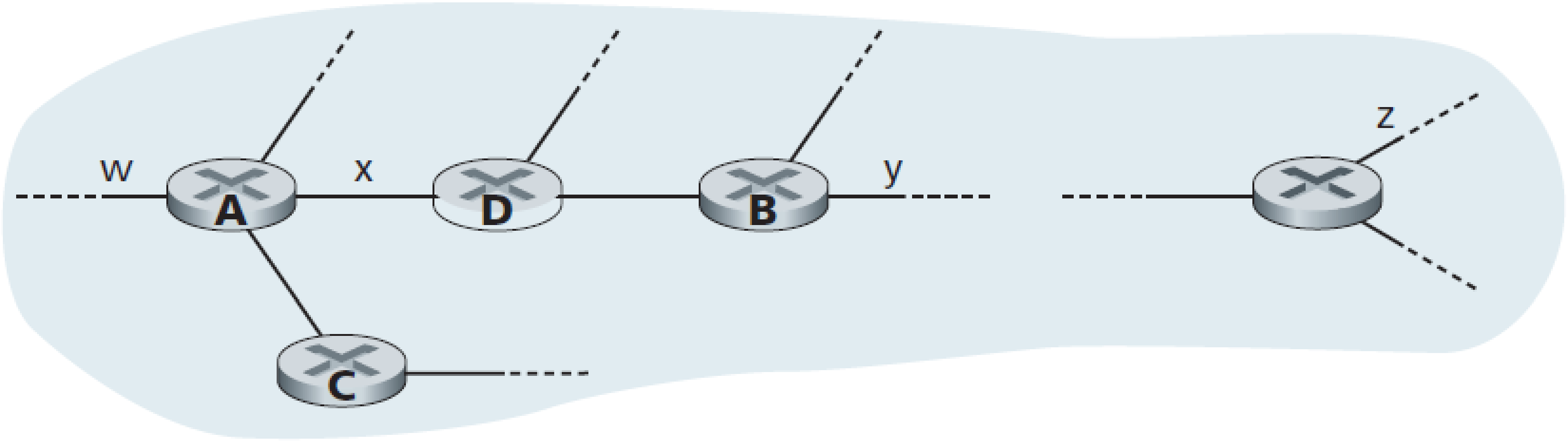
An Example



Destination	Hops
u	1
v	2
w	2
x	3
y	3
z	2

Figure 4.34 ♦ Number of hops from source router A to various subnets

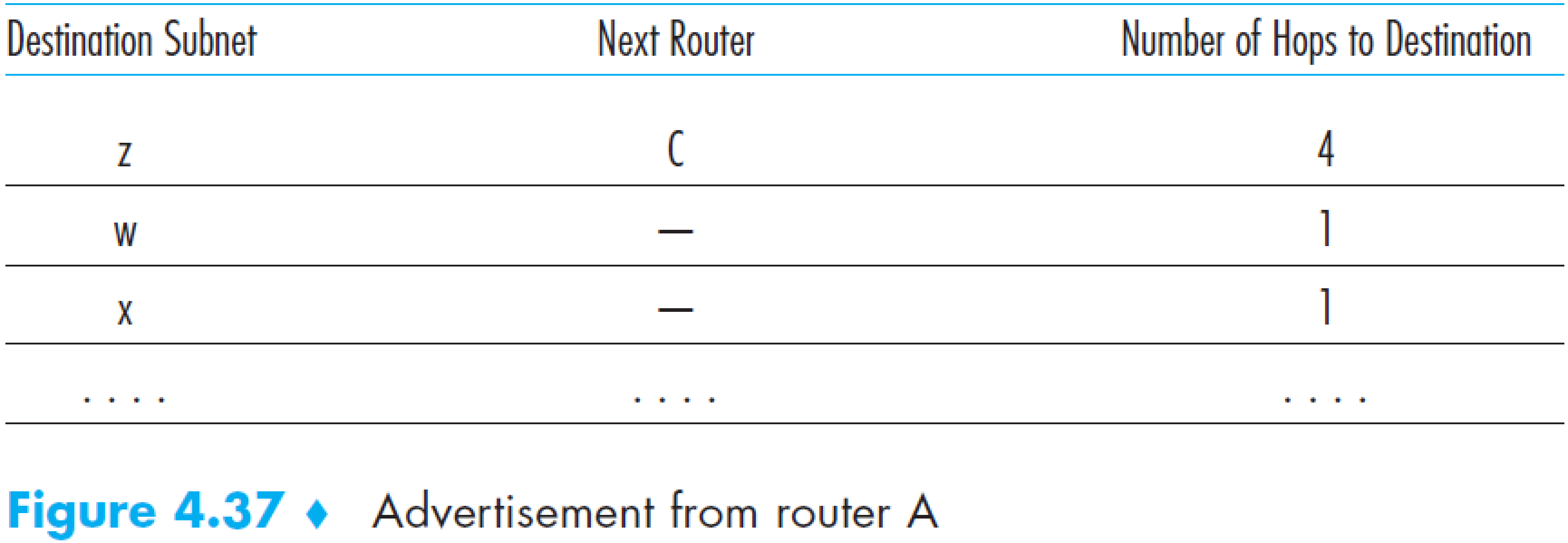
An AS with six leaf subnets. The table indicates the number of hops from the source A to each of the leaf subnets.



Each router maintains a RIP table known as a routing table. A router's routing table includes both the router's distance vector and the router's forwarding table.

Destination Subnet	Next Router	Number of Hops to Destination
w	A	2
y	B	2
z	B	7
x	—	1
. . . .	. . . .	. . . .

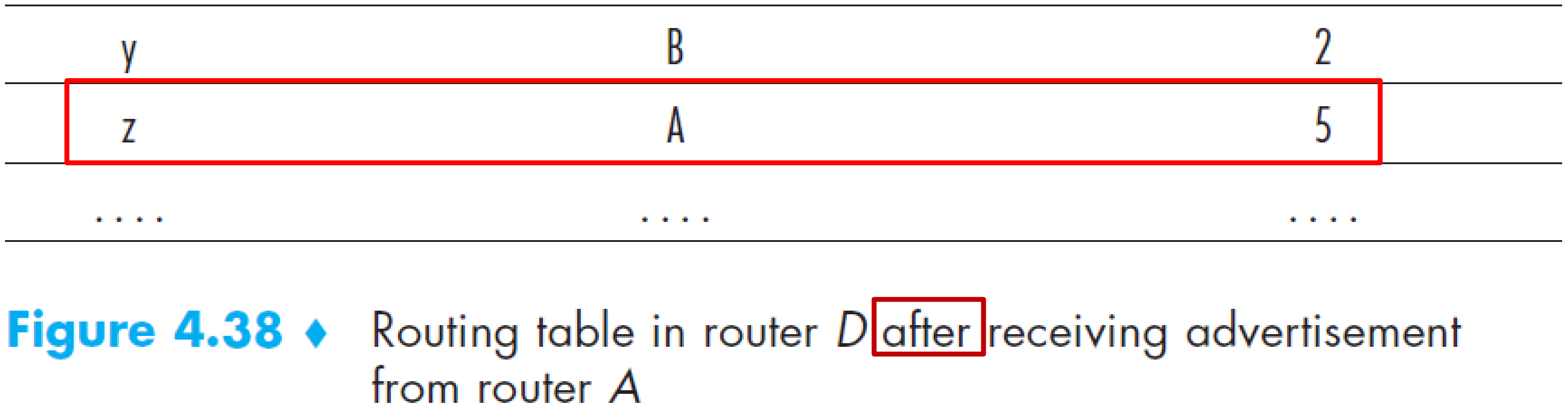
Figure 4.36 ♦ Routing table in router *D* before receiving advertisement from router *A*



Destination Subnet	Next Router	Number of Hops to Destination
z	C	4
w	—	1
x	—	1
....	....	....

Figure 4.37 ♦ Advertisement from router A

Now suppose that 30 seconds later, router D receives from router A the advertisement. Note this advertisement is nothing other than the routing table from router A! This information indicates that subnet z is only four hops from router A. Router D then updates its routing table to account for the shorter shortest path.



y	B	2
z	A	5
....	....	....

Figure 4.38 ♦ Routing table in router *D* after receiving advertisement from router *A*

OSPF — An Interior Gateway Routing Protocol

- OSPF (Open Shortest Path First) and its closely related cousin, IS-IS, are typically deployed in upper-tier ISPs whereas RIP is deployed in lower-tier ISPs and enterprise networks.
- At its heart, OSPF is **link-state protocol** that uses flooding of link state information and a Dijkstra least-cost path algorithm. [RFC2328]
  - With OSPF, a router constructs a complete topological map of the entire autonomous system. The router then locally runs Dijkstra's shortest-path algorithm to determine a shortest-path tree to all *subnets*, with itself as the root node.
  - It also broadcasts a link's state periodically (at least once every 30 minutes), even if the link's state has not changed.
  - OSPF advertisements are contained in OSPF messages that are carried directly by **IP**, with an upper-layer protocol of 89 for OSPF.

OSPF

- If multiple paths are found that are equally short. In this case, OSPF remembers the set of shortest paths and during packet forwarding, traffic is split across them. — **ECMP** (Equal Cost MultiPath)
  - Load balance

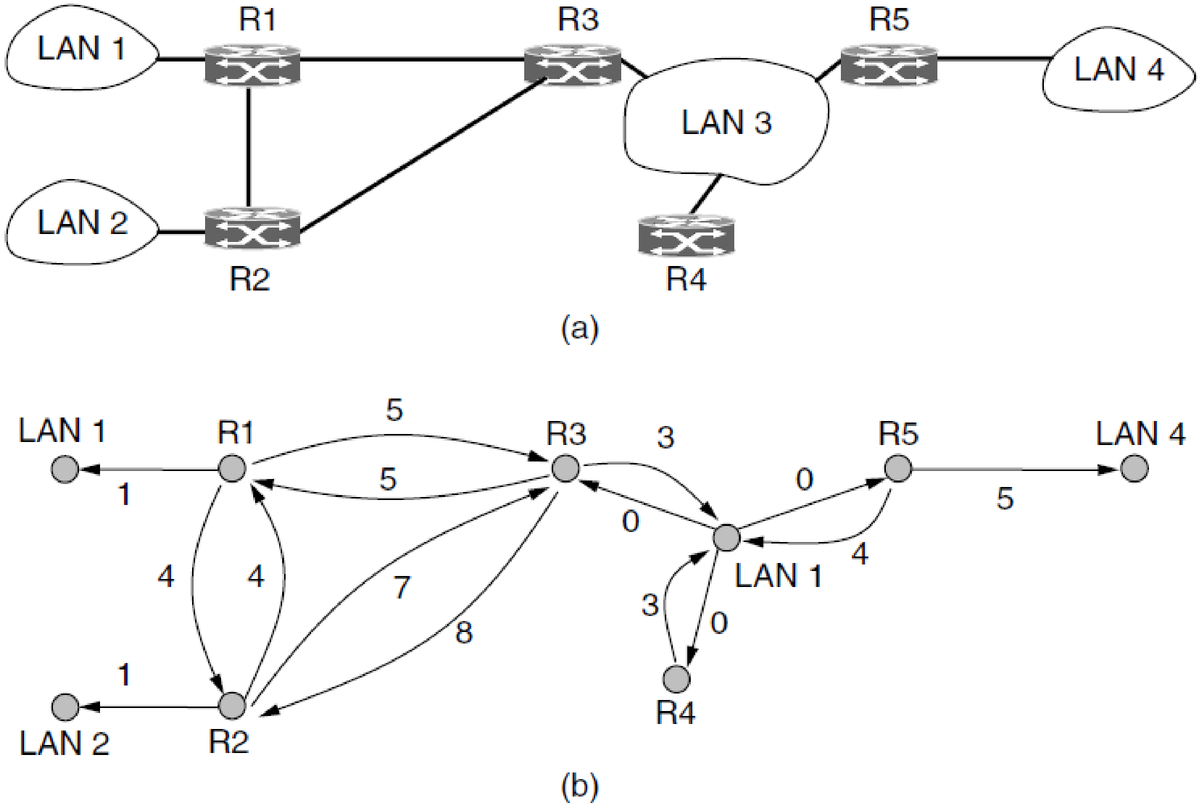
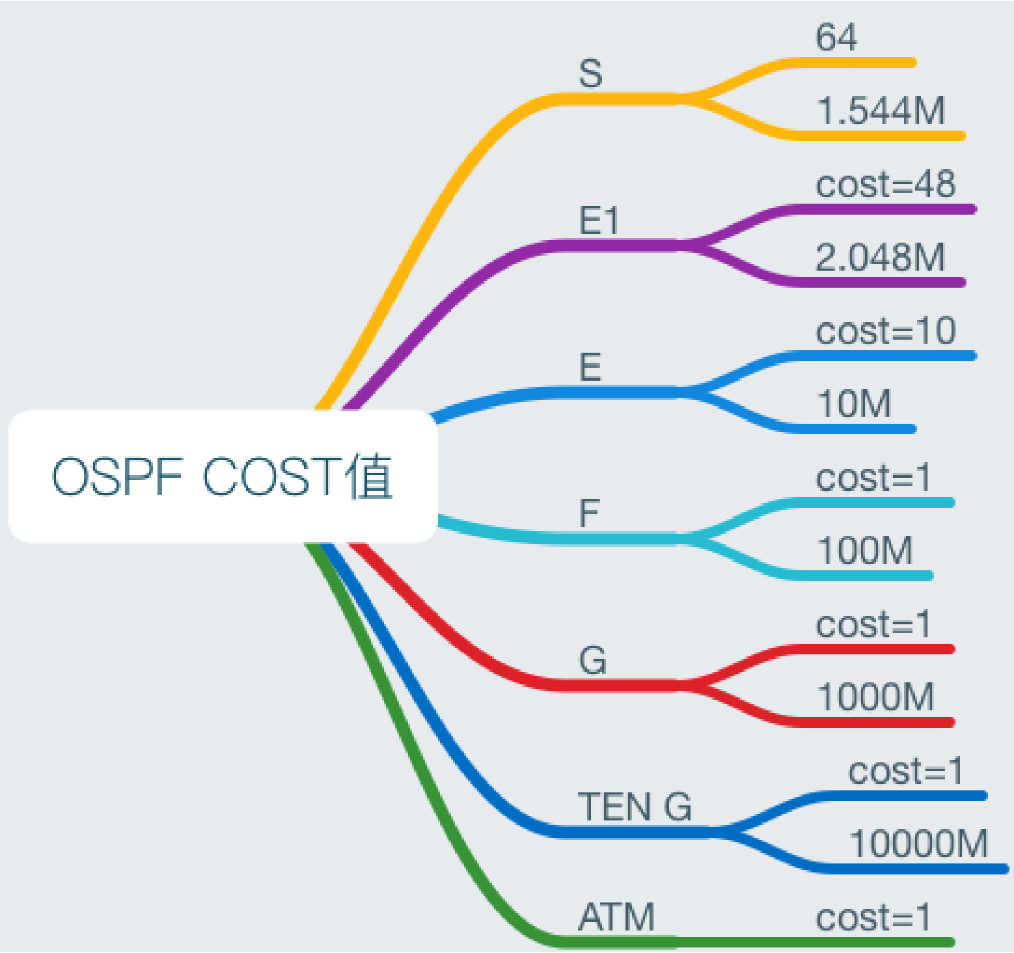


Figure 5-64. (a) An autonomous system. (b) A graph representation of (a).

OSPF Cost Table



The higher data rate, the lower cost of the channel.

OSPF

- Many of the ASes in the Internet are themselves large and nontrivial to manage.
- To work at this scale, OSPF allows an AS to be divided into **numbered areas**, where **an area** is a network or a set of contiguous networks. — hierarchical OSPF routing
  - An area is a generalization of an individual network.
  - Outside an area, its destinations are visible but not its topology.
  - Routers that lie wholly within an area are called **internal routers**.
- Every AS has **a backbone area**, called **area 0**.
  - The routers in this area are called **backbone routers**.
  - All areas are connected to the backbone, possibly by tunnels, so it is possible to go from any area in the AS to any other are in the AS via the backbone.
  - As with other areas, the topology of the backbone is not visible outside the backbone.

OSPF

- Each router that is connected to two or more areas is called **an area border router**. It must also be part of the backbone.
  - The job of an area border router is to **summarize** the destinations in one area and to **inject** this summary into the other areas to which it is connected.
  - This summary includes cost information but **not** all the details of the topology within an area.
  - If there is only one border router out of an area, even the summary does not need to be passed. This kind of area is called a **stub area**.
- **An AS boundary router** injects routes to external destinations on other ASes into the area.
  - The external routers then appear as destinations that can be reached via the AS boundary router with some cost.

OSPF

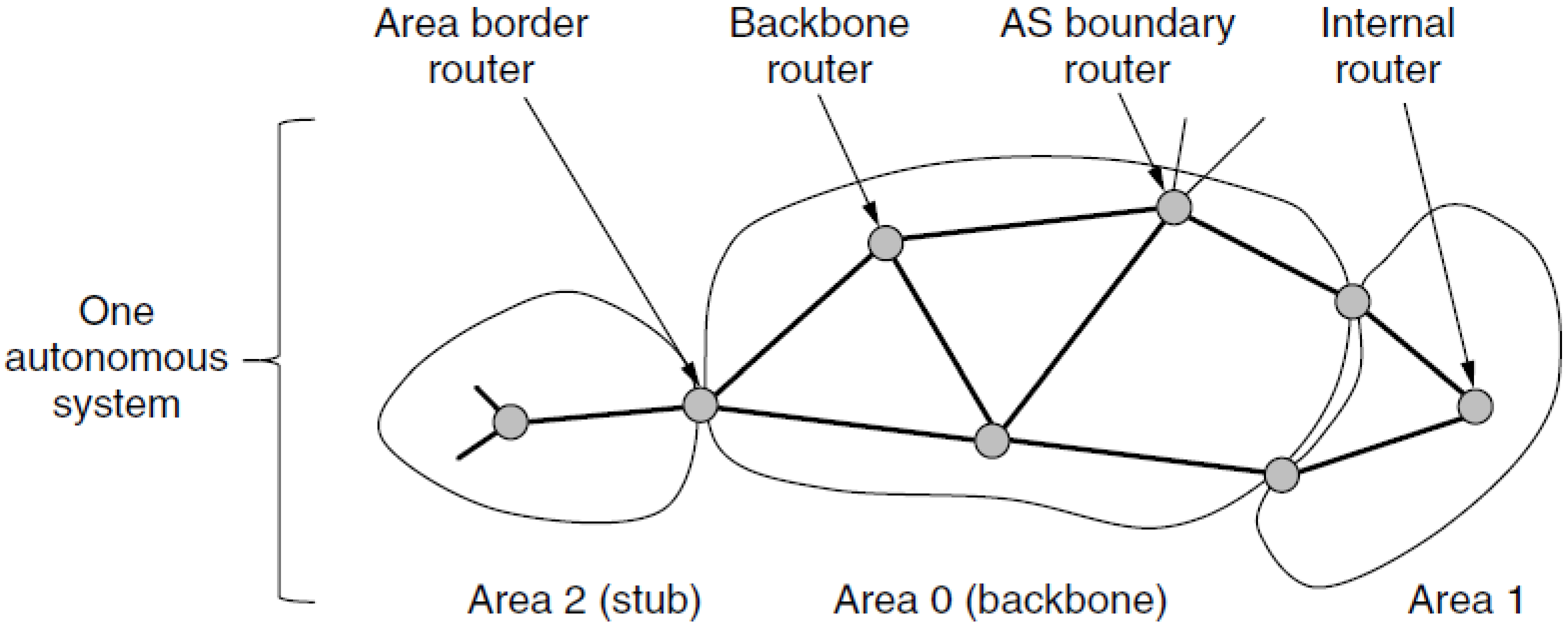


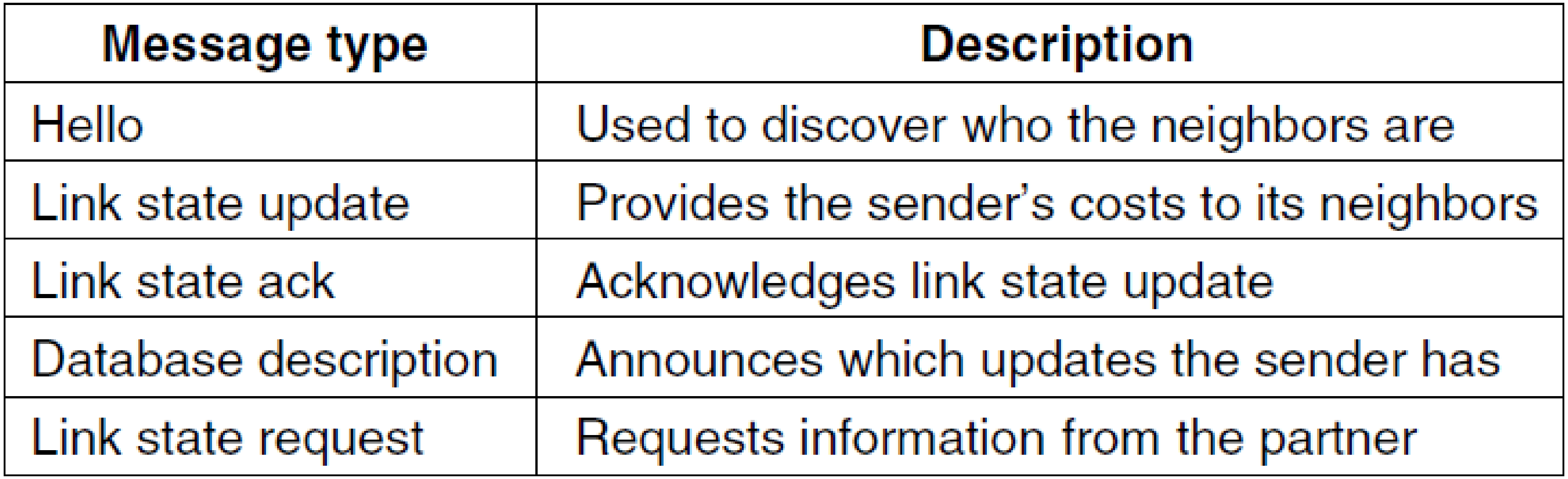
Figure 5-65. The relation between ASes, backbones, and areas in OSPF.

One router may play multiple roles, for example a border router is also a backbone routers.

OSPF

- Each router within an area has **the same link state database** and runs the same shortest path algorithm.
- **An area border router** needs the databases for all the areas to which it is connected and must run the shortest path algorithm for each area separately.
- For a source and a destination in different areas, the inter-area route must go from the source to the backbone, across the backbone to the destination are, and then to destination.
  - A star configuration on OSPF, with the backbone being the hub and other areas being spokes (车轮辐条).
- Routers to external destinations may include the external cost from **the AS boundary router** over the external path.
- It is inefficient to have every router on a LAN talk to every other router on the LAN.
  - To avoid this situation, one router is elected as **the designed router**. It acts as **the single node** that represents the LAN.
  - **A backup designed router** is always kept up to date to ease the transition should the primary designed router crash and need to be replaced immediately.

OSPF



Message type	Description
Hello	Used to discover who the neighbors are
Link state update	Provides the sender's costs to its neighbors
Link state ack	Acknowledges link state update
Database description	Announces which updates the sender has
Link state request	Requests information from the partner

Figure 5-66. The five types of OSPF messages.

During normal operation, each router **periodically floods** LINK STATE UPDATE messages to each of its adjacent routers.
The flooding messages are **acknowledged**, to make them reliable.
Each message has **a sequence number**, so a router can see whether an incoming LINK STATE UPDATE is older or newer than what it currently has.

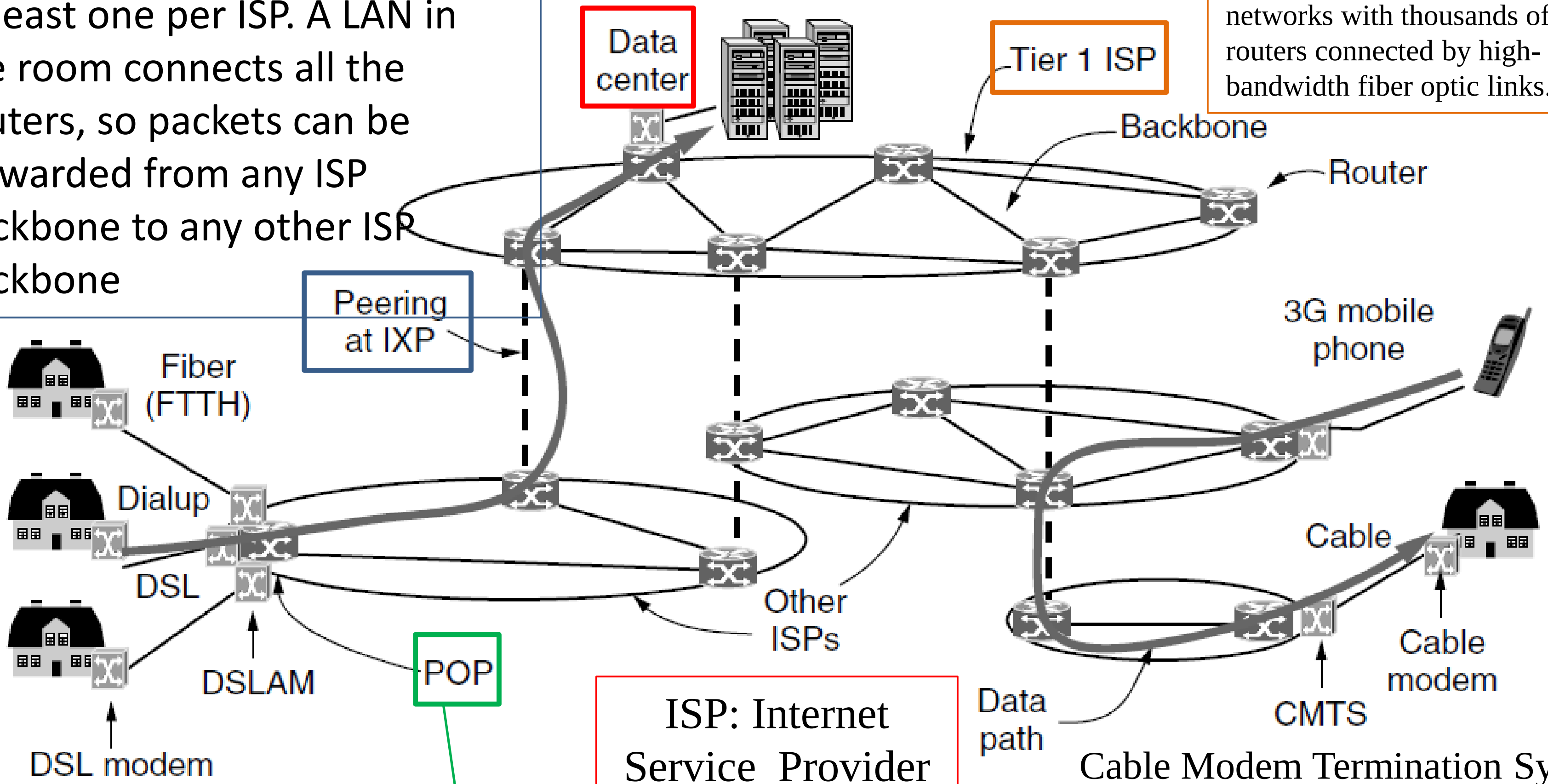
BGP — The Exterior Gateway Routing Protocol

- The BGP (Border Gateway Protocol) is the de facto standard inter-AS routing protocol in today's Internet. [RFC 4271; RFC 4274; RFC 4276]
  - BGP is a form of **distance vector routing protocol** based on **policies** rather than on minimum distance
- The connection is often made with a link at IXP (Internet eXchange Points), facilities to which many ISPs have a link for the purpose of connecting with other ISPs.
- If a customer connects to one and only one ISP, it does not need to run BGP because it is **a stub network** that is connect to the rest of the Internet by only one link.
- However, some company networks are connected to multiple ISPs (**multihoming**). In this case, the company network is likely to run an interdomain routing protocol (e.g. BGP) to tell other ASs which addresses should be reached via which ISP links.
- BGP chooses a path to follow at the AS level and OSPF chooses paths within each of the ASs.

The **path** a packet takes through the internet depends on the peering choices of the ISPs.

An IXP is a room full of routers, at least one per ISP. A LAN in the room connects all the routers, so packets can be forwarded from any ISP backbone to any other ISP backbone

Like AT&T and Sprint, operate large international backbone networks with thousands of routers connected by high-bandwidth fiber optic links.



ISP: Internet Service Provider

Cable Modem Termination System

Figure 1-29. Overview of the Internet architecture.

Telephone line:

1) Dial up
2) DSL (Digital Subscriber Line) **broadband**
3) Fiber to the Home

POP (Point of Presence): the location at which customer packets enter the ISP network for service. From this point on, the system is fully **digital** and **packet switched**.

BGP — The Exterior Gateway Routing Protocol

- BGP is an absolutely critical protocol for the Internet — in essence, it is the protocol that glues the whole thing together.
- In BGP, pairs of routers exchange routing information over **semipermanent TCP connections** using port **179**.
  - A mesh of TCP connections within each AS.
  - In BGP, **destinations** are **not** hosts but instead are **CIDRized prefixes**, with each prefix representing a subnet or a collection of subnets.

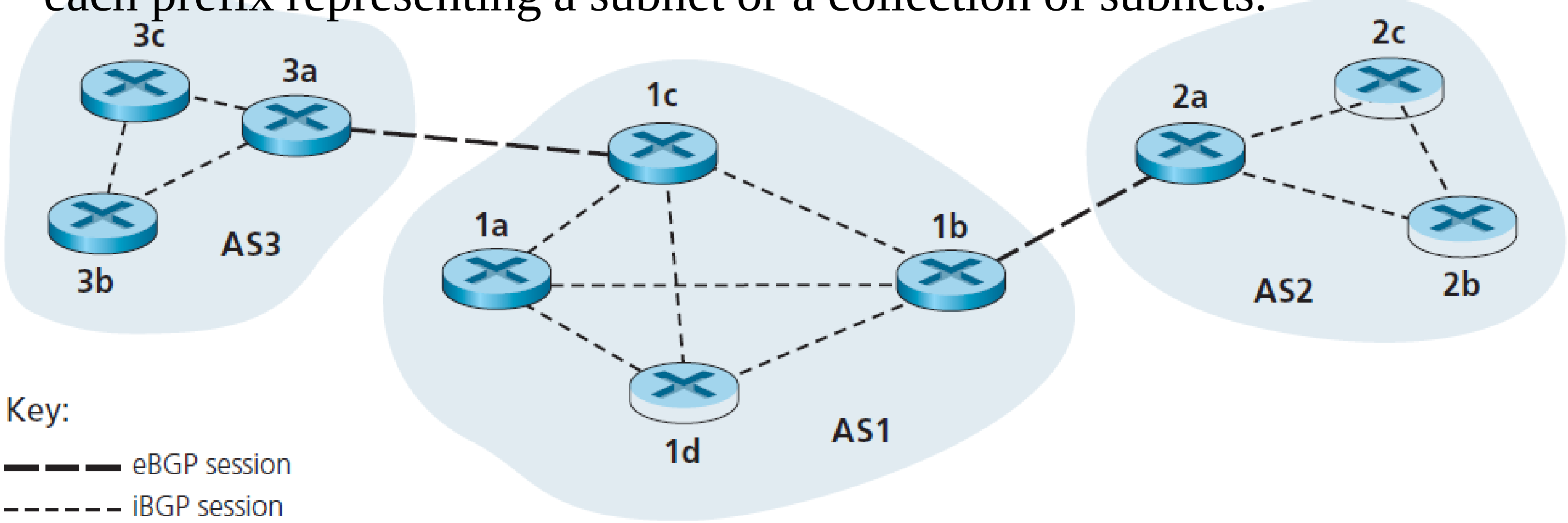


Figure 4.40 ♦ eBGP and iBGP sessions

BGP — The Exterior Gateway Routing Protocol

- When a gateway router (in any AS) receives eBGP-learned prefixes, the gateway router uses its iBGP sessions to distribute the prefixes to the other routers in the AS.
- When a router (gateway or not) learns about a new prefix, it creates an entry for the prefix in its forwarding table.
- In BGP, an autonomous system is identified by <u>its globally unique autonomous system number</u> (**ASN**) [RFC 1930].
  - AS numbers, like IP addresses, are assigned by ICANN regional registries.
- In BGP jargon, *a prefix along with its attributes* is called **a route**. Thus BGP peers advertise routes to each other.

BGP Route Advertising (I)

- Different parties like ISPs are called AS (Autonomous Systems)
- Border routers of ASs announce BGP routes to each other.
- Route advertisements contain **an IP prefix**, **AS-path**, **next hop**.
  - AS-Path is list of ASs on the way to the prefix; list is to find loops
    - When a router receives a route, it checks to see if its own AS number is already in the AS path. If it is, **a loop** has been detected and the advertisement is discarded.
- Route advertisements move **in the opposite direction to traffic**.

BGP Route Advertising (II)

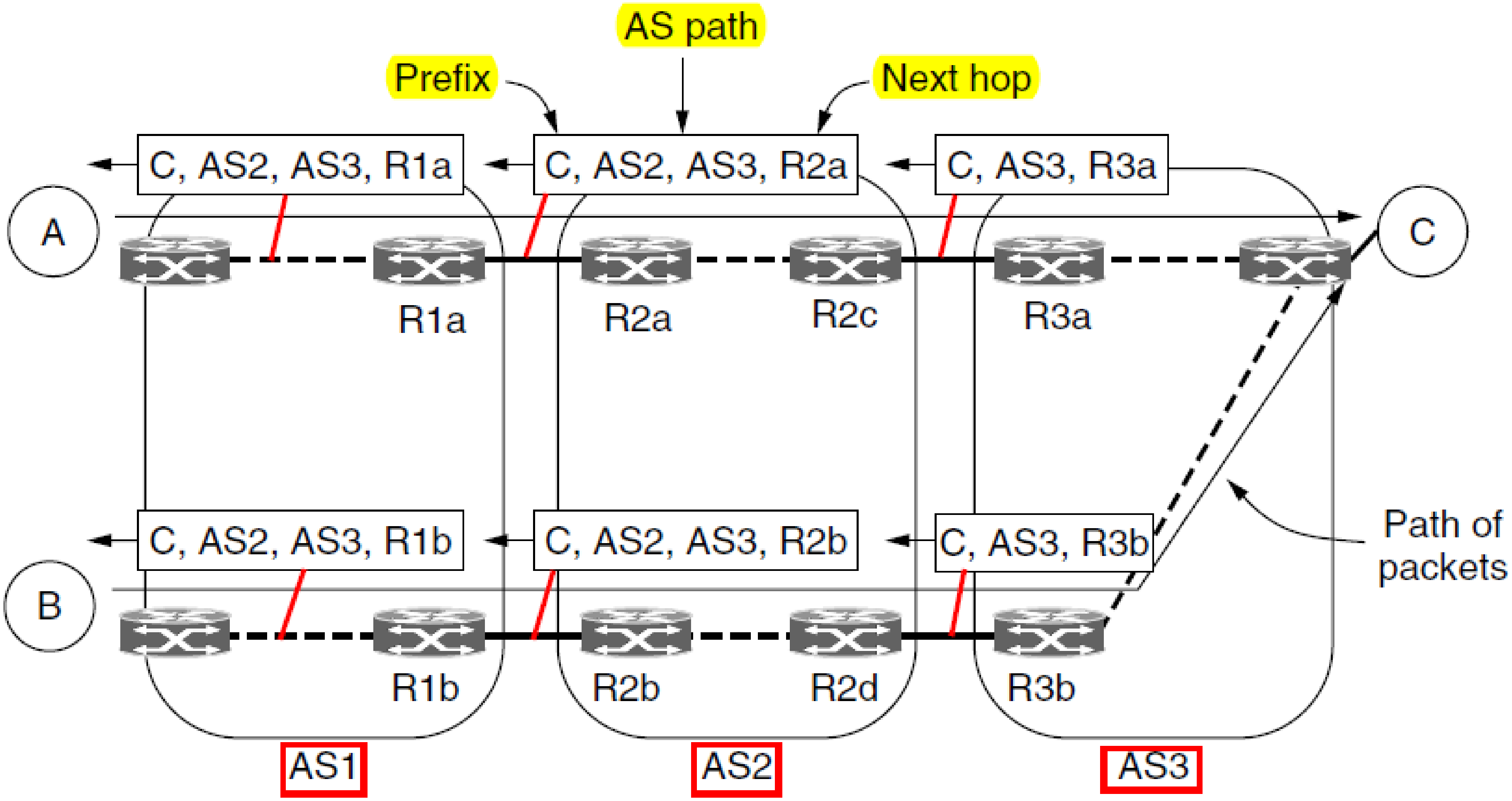


Figure 5-68. Propagation of BGP route advertisements.

BGP Example (I)

- AS2, AS3, and AS4 are customers of AS1. They buy **transit service** from it.
- When source A sends to destination C, the packets travel from AS2 to AS1 and finally to AS4.
- The routing advertisements travel *in the opposite direction to the packets*.
  - AS4 advertises C as a destination to its transit provider, AS1, to let sources reach C via AS1. Later, AS1 advertises a route to C to its other customers, including AS2, to let the customers know that they can send traffic to C via AS1.

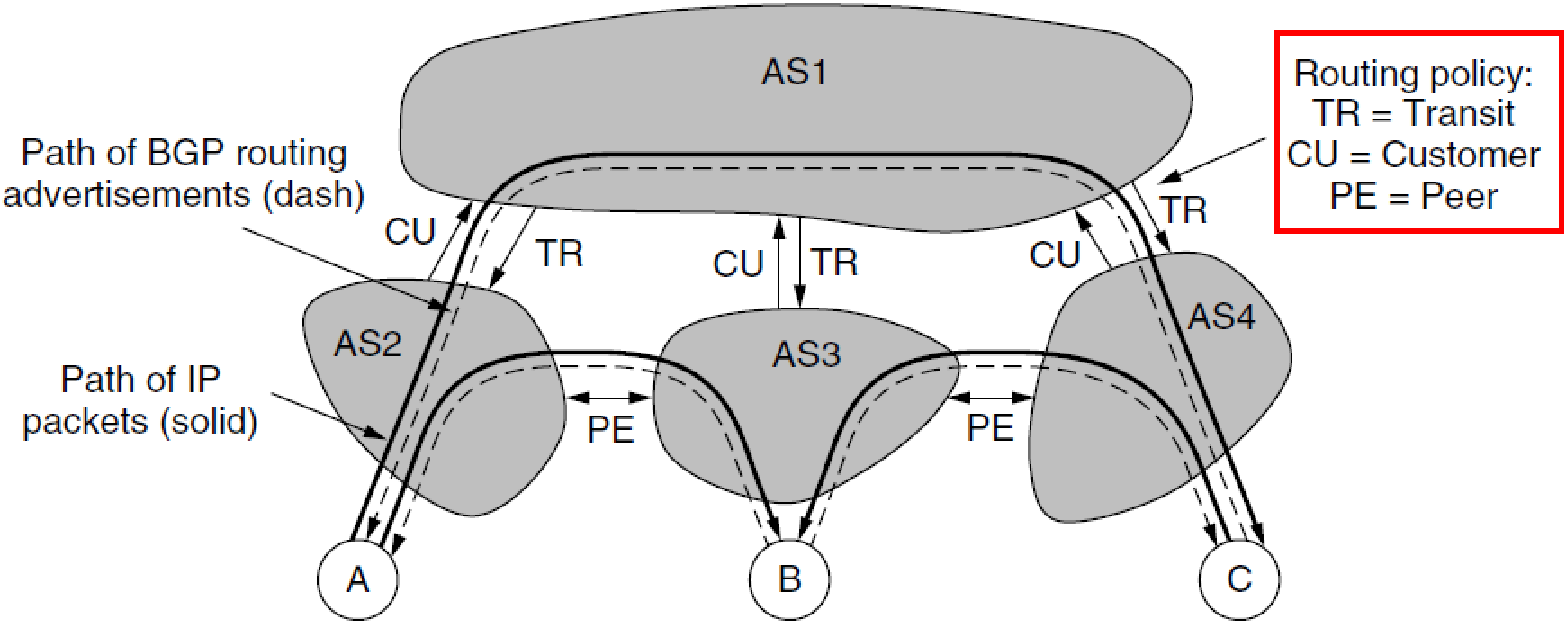


Figure 5-67. Routing policies between four autonomous systems.

BGP Example (II)

- AS2 buys **TRANSIT service** (TR) from AS1 and **PEER service** (PE) from AS3.

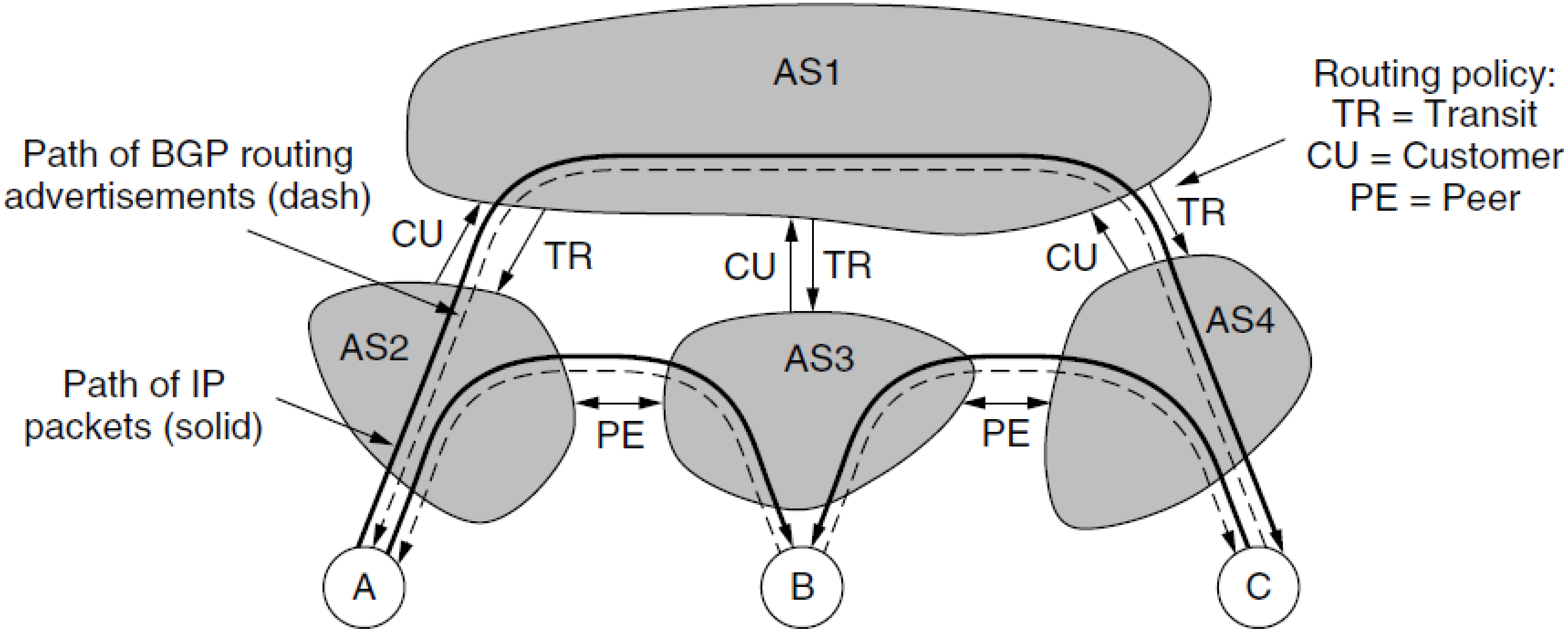


Figure 5-67. Routing policies between four autonomous systems.

Note that **peering is not transit**. Though AS2 is peering with AS3, and AS3 is peering with AS4, this does not mean AS2 is peering with AS4 through AS3, even though a physical path exists.

BGP Policy

- Policy is implemented in two ways:
  - Border routers of ISP announce paths only to other parties who may use those paths
    - Filter out paths others cannot use
  - Border routers of ISP select the best path of the ones they hear in any, non-shortest way
  - Hot-potato routing (early exit)

Outline

- Overview of network layer
- Routing algorithms
- The network layer in the Internet
- MPLS (Multiprotocol Label Switching)

Outline

- Overview of network layer
- Routing algorithms
- The network layer in the Internet
- MPLS (Multiprotocol Label Switching)

Label Switching and MPLS

- So far, we have focused exclusively on packets as **datagrams** that are forwarded by IP routers.
- MPLS (Multiprotocol Label Switching) is perilously close to circuit switching.
  - To improve *the forwarding speed* of IP routers by adopting a key concept from the world of virtual-circuit networks: **a fixed-length label**.
  - The goal was not to abandon the destination-based IP datagram-forwarding infrastructure for one based on fixed-length labels and virtual circuits, but to augment it by selectively labeling datagrams and allowing routers to forward datagrams based on fixed-length labels (rather than destination IP addresses) when possible.
  - The MPLS protocol [RFC 3031, RFC 3032]

MPLS

- The 1st question to ask is **where** does the label go?
  - Since IP packets were not designed for virtual circuits, there is no field available for virtual-circuit numbers within the IP header.
  - A new MPLS header had to be added between the layer-2 (i.e., PPP or Ethernet) header and layer-3 (i.e., IP) header.

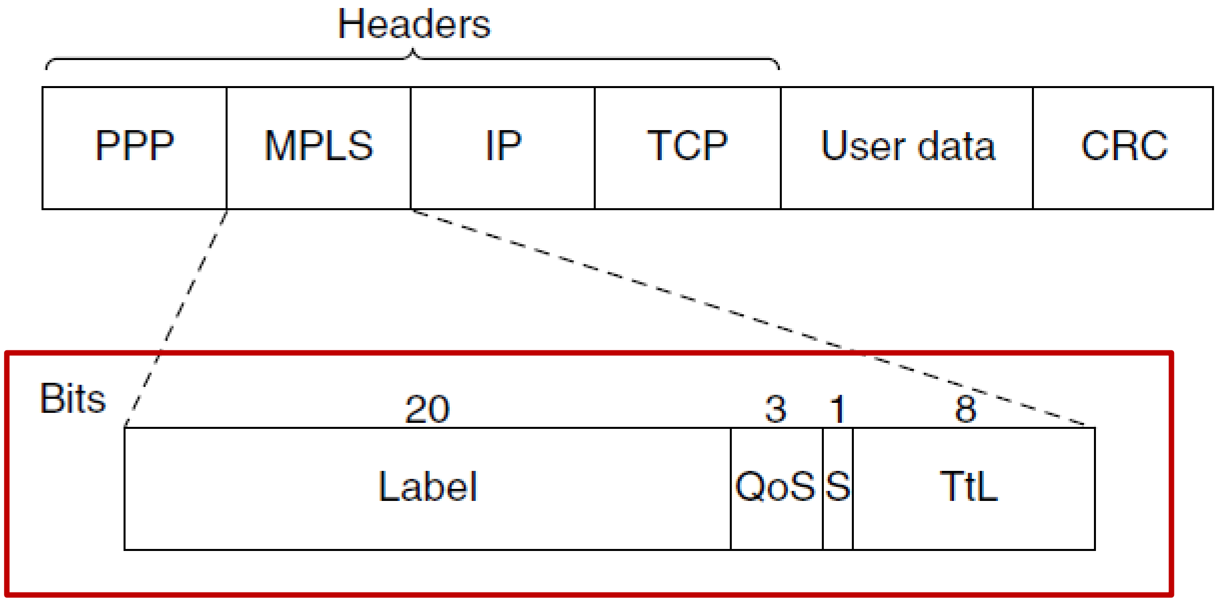


Figure 5-62. Transmitting a TCP segment using IP, MPLS, and PPP.

MPLS

- The generic MPLS header is 4 bytes long and has four fields
  - The Label field
  - The QoS field
  - The S field (relates to stacking multiple labels)
  - The TtL field (decremented at each router, and if it hits 0, the packet is discarded.)
- MPLS falls between the network layer protocol and the data link layer protocol.
  - It is not really a layer 3 protocol because it depends on IP or other network layer addresses to set up label paths.
  - It is not really a layer 2 protocol either because it forwards packets across multiple hops, not a single link.

MPLS

- MPLS is sometimes described as a layer 2.5 protocol.
- When an MPLS-enhanced packet arrives at a LSR (**Label Switched Router**), the label is used as an index into a table to determine the outgoing line to use and also the new label to use.
  - Labels have to be **remapped** at every hop.
    - Labels have only local significance
  - The label swapping is used in all virtual-circuit networks.
  - The longest matching prefix algorithm is used for IP forwarding.
  - In contrast, switching using a label taken from the packet as an index into a forwarding table. It is simpler and faster.

MPLS

- Since most hosts and routers do not understand MPLS, the 2nd question is to ask **when and how** the labels are attached to packets?
  - This happens when an IP packet reaches the edge of an MPLS network. The **LER** (Label Edge Router) inspects the destination IP address and other fields to see which MPLS path the packet should follow, and puts the right label on the front of the packet.
  - Within the MPLS network, **the label** is used to forward the packet.
  - At the other edge of the MPLS network, the label has served its purpose and is removed, revealing the IP packet again for the next network.

MPLS

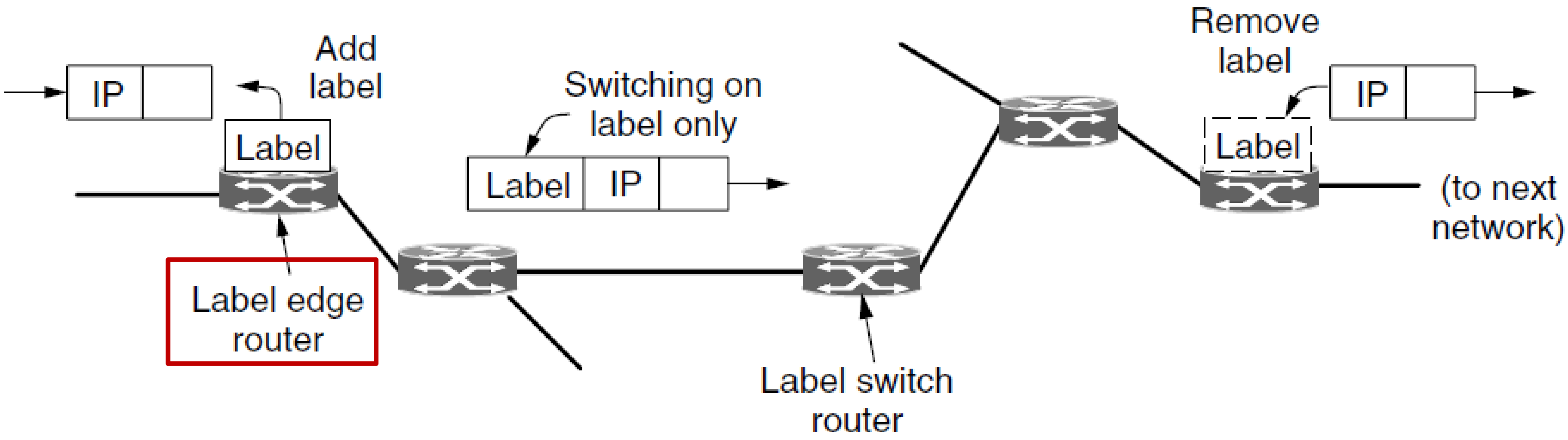


Figure 5-63. Forwarding an IP packet through an MPLS network.

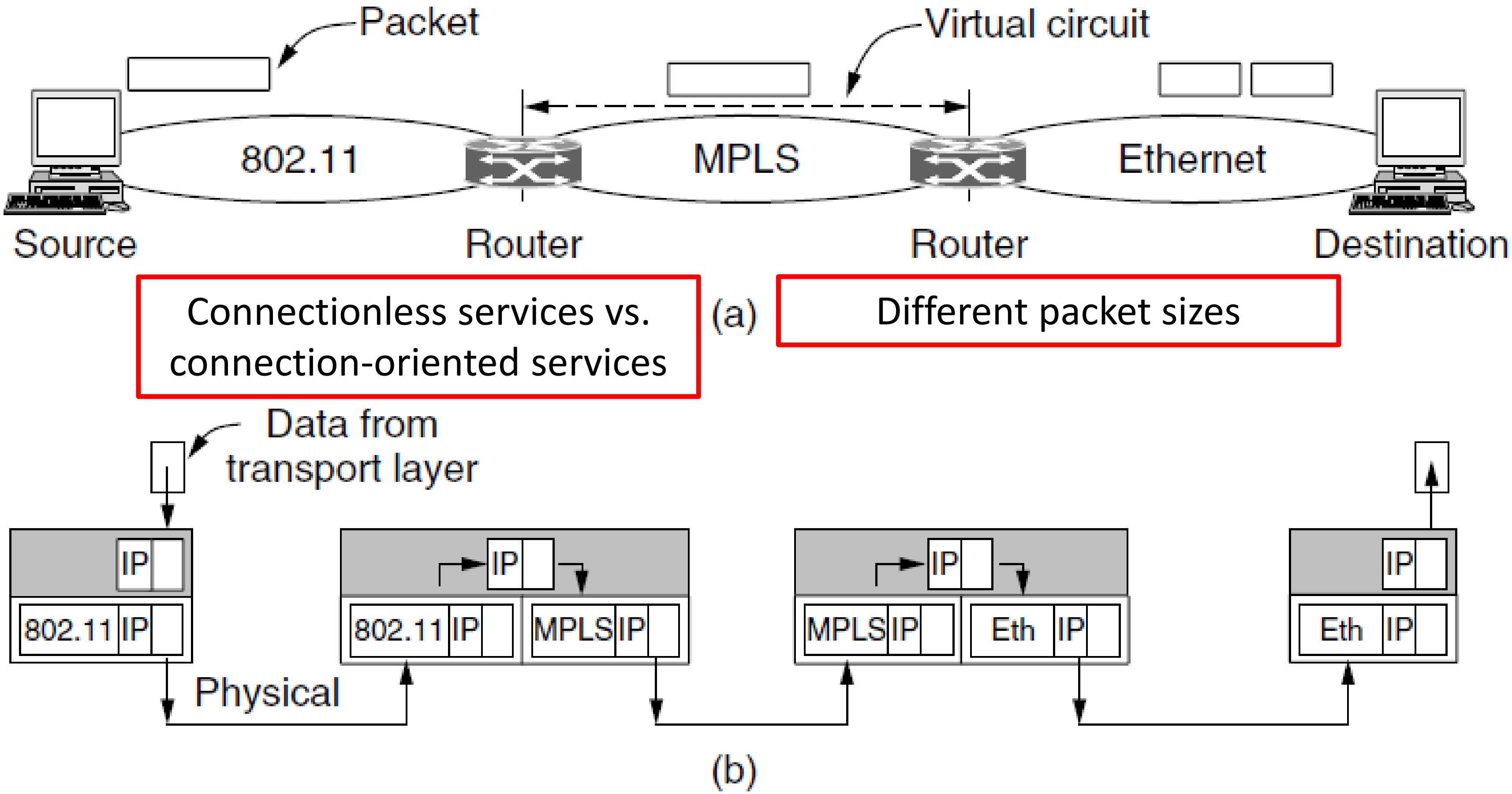
- One difference from traditional virtual circuits is the level of **aggregation**.
- It is common for routers to group multiple flows that end at a particular router or LAN and use a single label for them. But with traditional virtual-circuit routing, it is not possible to group several distinct paths with different endpoints onto the same virtual-circuit identifier because there would be no way to distinguish them at the final destination. *With MPLS, the packets still contain their final destination address, in addition to the label.*

MPLS

- MPLS can operate at multiple levels at once by adding more than one label to the front of a packet.
  - For example, suppose that there are many packets that already have different labels (because we want to treat the packets differently somewhere in the network) that should follow a common path to some destination.
  - Instead of setting up many label switching paths, one for each of the different labels, we can set up a single path. When the already-labeled packets reach the start of this path, another label is added to the front.
    - A stack of labels
  - The S bit in Fig.5-62 allows a router removing a label to know if there are any additional labels left. It is set to 1 for the bottom label and 0 for all the other labels.

MPLS

- The final question we will ask is **how** the label forwarding tables are set up so that packets follow them.
  - In traditional virtual-circuit networks, when a user wants to establish a connection, **a setup packet** is launched into the network to create the path and make the forwarding table entries.
  - MPLS dos not involve users in the setup phase. The forwarding information is setup by protocols that are a combination of routing protocols and connection setup protocols.
  - When a router is booted, it checks to see which routes it is the final destination for (e.g., which prefixes belong to its interfaces). It then creates one or more FECs (Forwarding Equivalent Class) for them, allocates a label for each one and passes the labels to its neighbors. They, in turn, enter the labels in their forwarding tables and send new labels to their neighbors, until all the routers have acquired the path.



Connectionless services vs. connection-oriented services

Different packet sizes

Figure 5-39. (a) A packet crossing different networks. (b) Network and link layer protocol processing.

The packet carries a common network layer header (e.g. IP) that can identify any host across the 3 networks. The network header contains the ultimate destination address.

Main Points (I)

- Two main functions of the network layer
  - Forwarding
  - Routing
- The routing algorithms
  - Link-state routing algorithms
    - Dijsktra’s algorithm
    - may have oscillations
  - Distance-vector routing algorithms (iterative, distributed, asynchronous)
    - Bellman-Ford equation
    - The count-to-infinity problem
  - Hierarchical routing
  - Broadcast routing
  - Multicast routing
  - Anycast routing

Main Points (II)

- IP addressing
  - IPv4 (32 bits)
    - Subnetting & aggregation
    - NAT
    - DHCP
  - IPv6 (128 bits)
- The network layer of the internet has three main components:
  - The IP protocol
  - The Internet routing protocols (including RIP, OSPF and BGP)
  - The Internet control protocols (including ICMP, DHCP, ARP, DNP)
- Connectionless services and Connection-oriented services
  - Datagrams vs. Virtual-Circuits (MPLS)

References

- [1] A.S. Tanenbaum, and D.J. Wetherall, Computer Networks, 5th Edition, Prentice Hall, 2011.
- [2] E. W. Dijkstra. “A Note on Two Problems in Connection with Graphs,” Numerische Mathematik, 1. 269-271, 1959.
  - http://www.cs.utexas.edu/~EWD/
- [3] https://tools.ietf.org/html/rfc1958
- [4] https://tools.ietf.org/html/rfc1700
- [5] https://www.ietf.org/rfc/rfc3022.txt
- [6] http://www.iana.org/assignments/icmp-parameters/icmp-parameters.xhtml
- [7] https://baike.baidu.com/item/%E8%B7%9F%E8%B8%AA%E8%B7%AF%E7%94%B1/8971154?fromtitle=tracert&fromid=7578188&fr=aladdin (tracert == trace router)

References

- [8] J. F. Kurose and K.W. Ross, Computer Networking — A Top-down Approach, 5th Edition, Pearson Education Inc., 2010.
- [9] D. Kreutz, F.M.V. Ramos, P.E. Verissimo, C.E. Rothenberg, D.Azonolmolky, and E. Uhlig, “Software-defined networking: a comprehensive survey,” Proc. Of the IEEE, vol.103, no.1, 2015. (截至2021年11月1日论文引用次数2391)
- [10] https://opennetworking.org/sdn-definition/ (ONF)