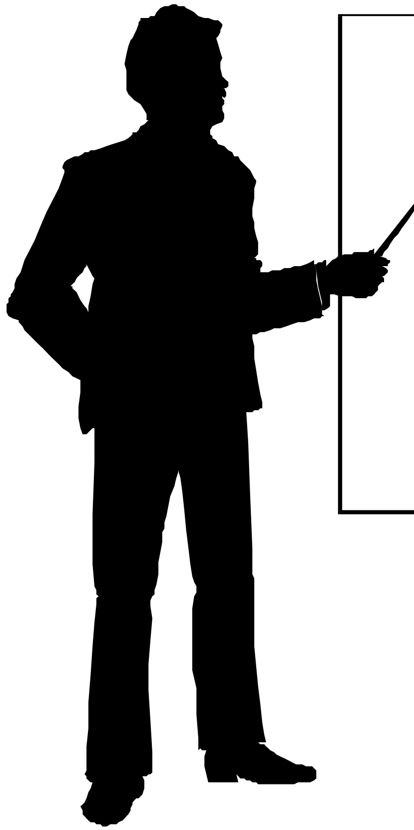


ARP

Introduction

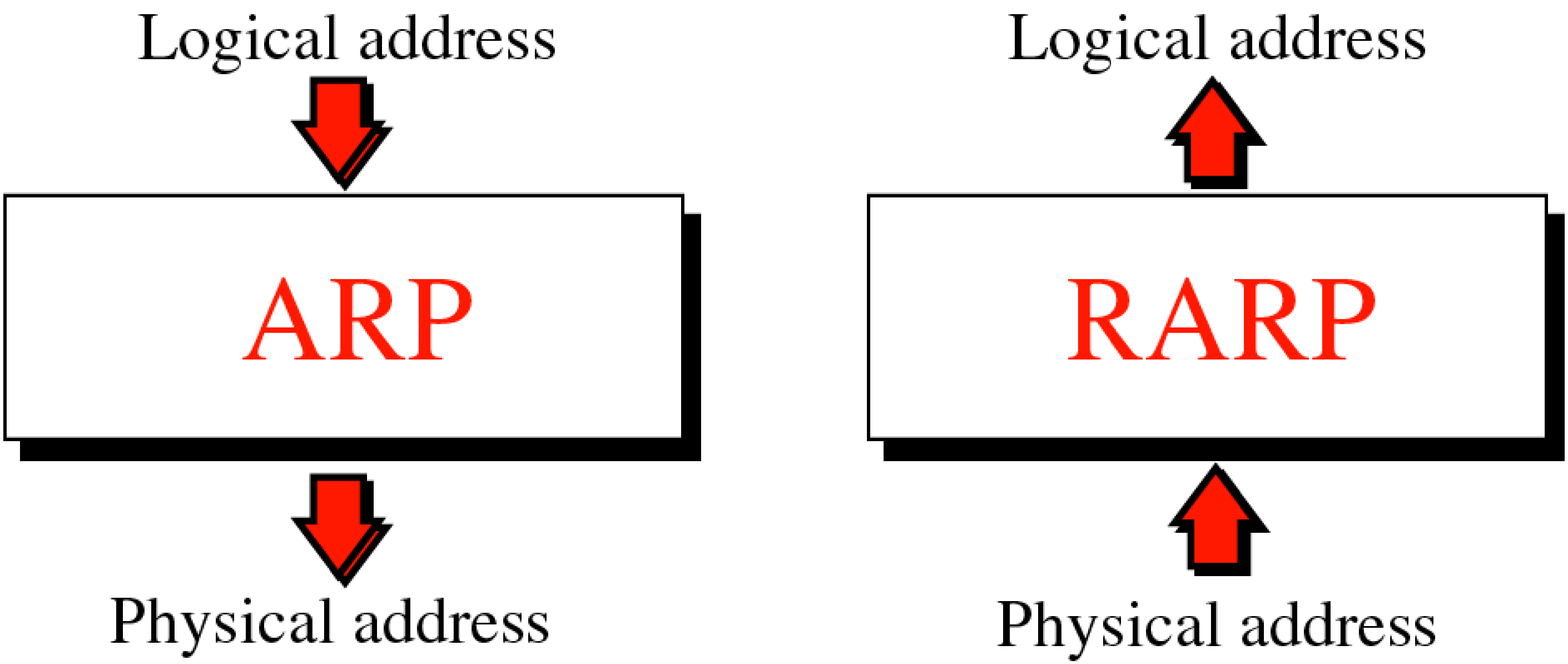
- The logical addresses in TCP/IP protocol suite are IP addresses
- Physical address is usually implemented in hardware
  - Ex) 48-bit MAC addresses in Ethernet and Token ring protocols, which are imprinted on the NIC installed in the host or router

Introduction (cont'd)

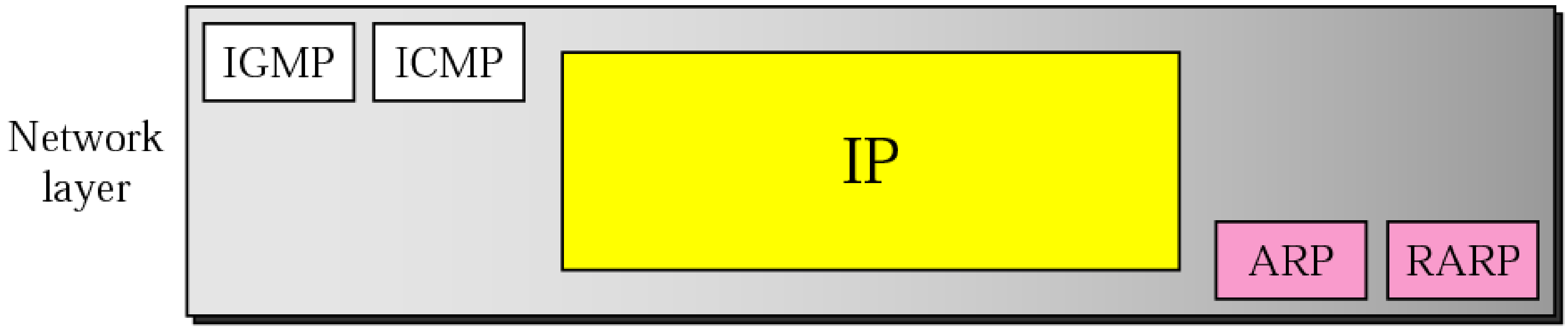
- Mapping a logical address to its corresponding physical address
  - Static mapping
    - means creating a table that associates a logical address with a physical address
    - need to update periodically
  - Dynamic mapping
    - each time a machine knows one of the two addresses (logical or physical), it can use a protocol to find the other one
    - ARP (address resolution protocol), RARP (reverse address resolution protocol)

ARP and RARP

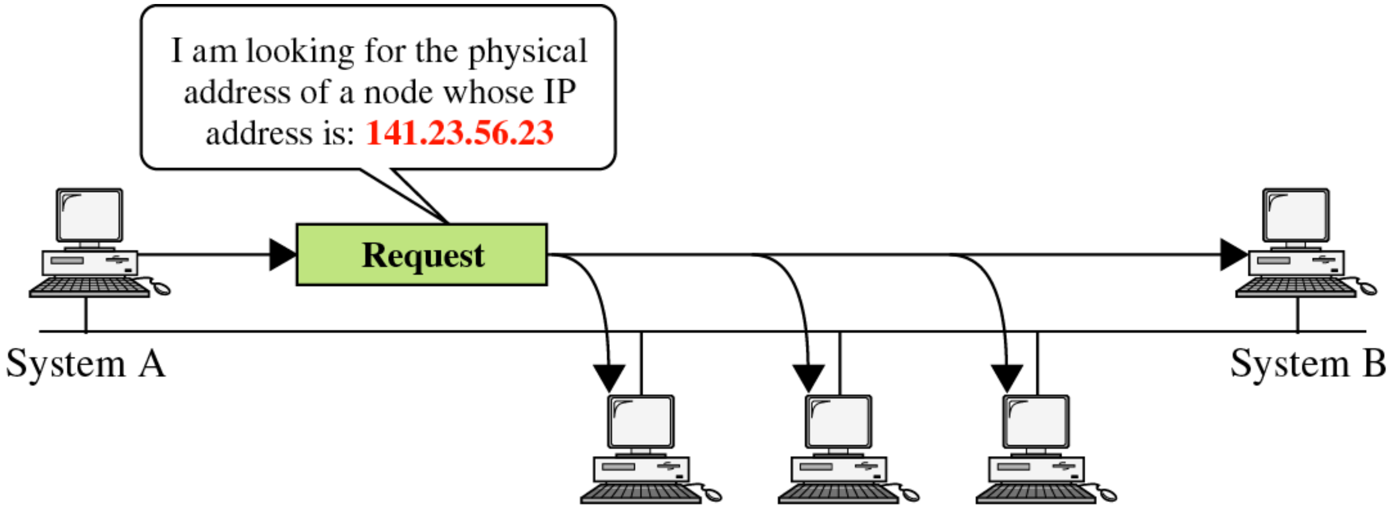
- ARP (address resolution protocol)
- RARP (reverse address resolution protocol)



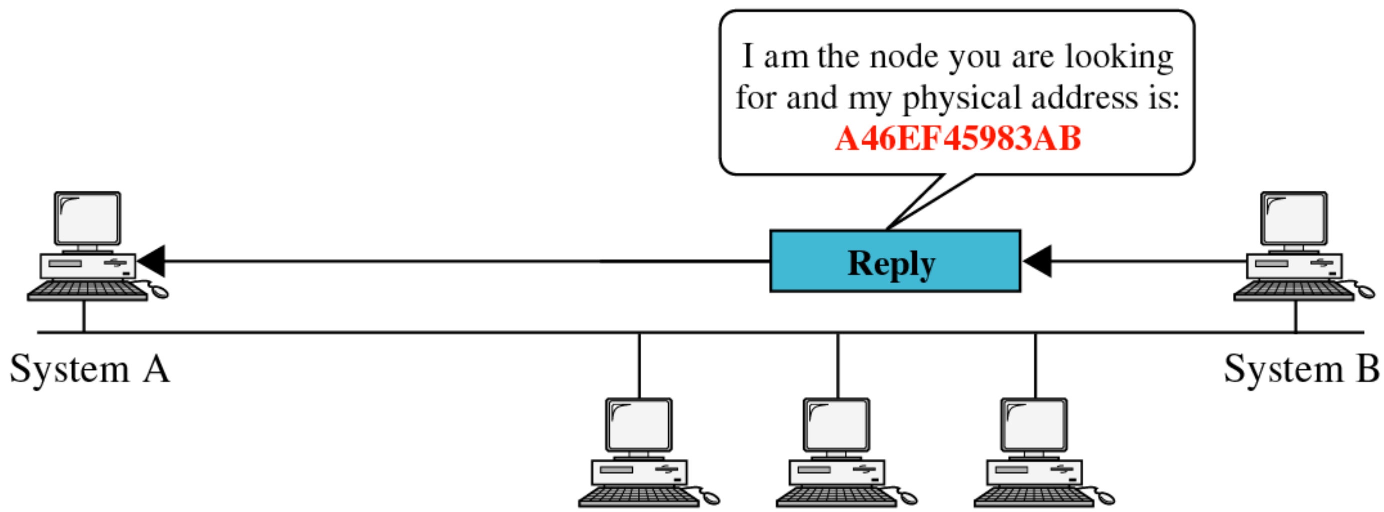
Position of ARP and RARP in TCP/IP Protocol Suite



ARP Operation



a. ARP request is broadcast



b. ARP reply is unicast

ARP Operation (cont'd)

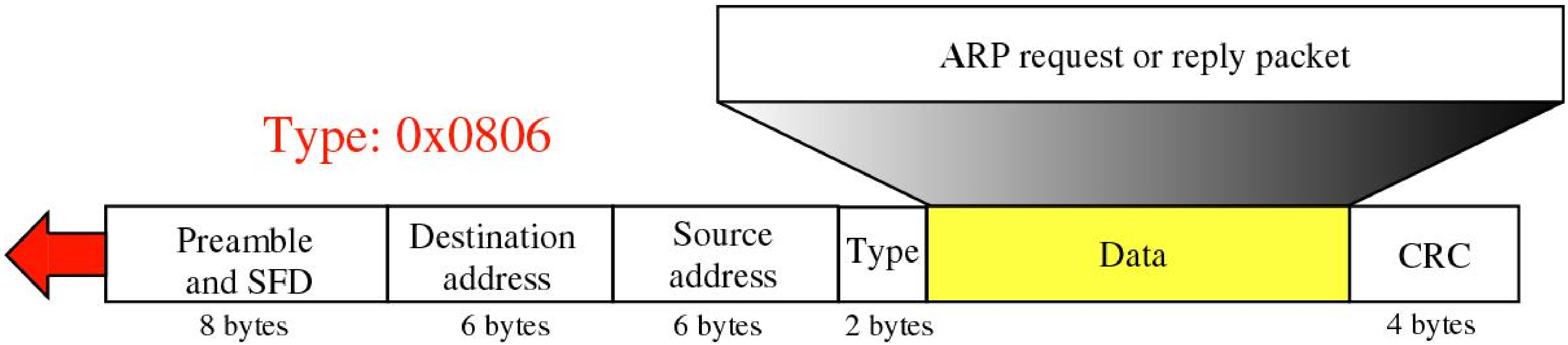
- An ARP request is **broadcast**; an ARP reply is **unicast**

The Format of ARP packet

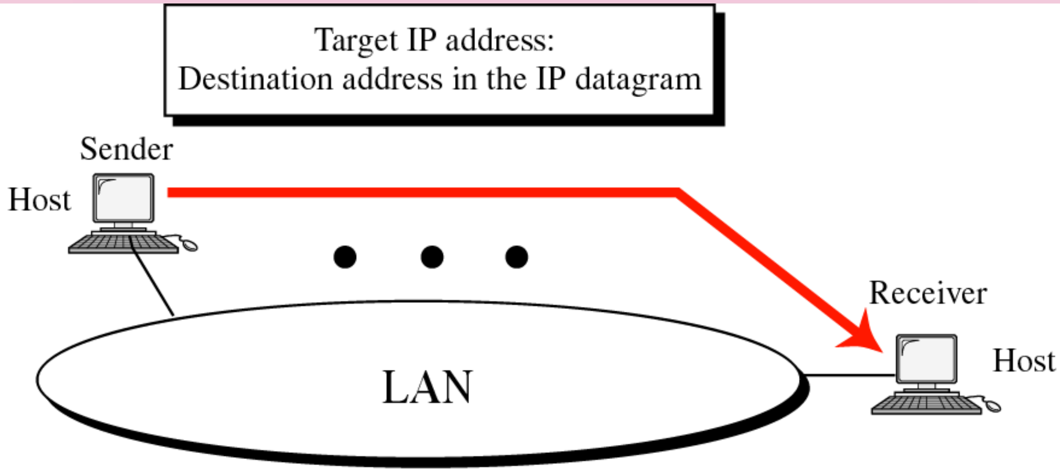
Hardware Type		Protocol Type
Hardware length	Protocol length	Operation Request 1, Reply 2
Sender hardware address (For example, 6 bytes for Ethernet)		
Sender protocol address (For example, 4 bytes for IP)		
Target hardware address (For example, 6 bytes for Ethernet) (It is not filled in a request)		
Target protocol address (For example, 4 bytes for IP)		

Encapsulation of ARP Packet

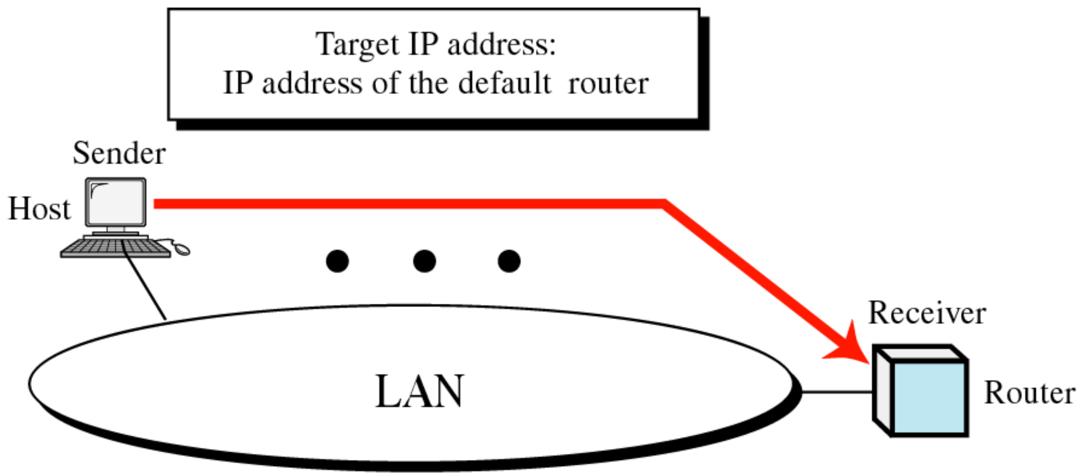
- encapsulated directly into a data link frame
- ARP packet encapsulated in an Ethernet frame



Four Cases using ARP

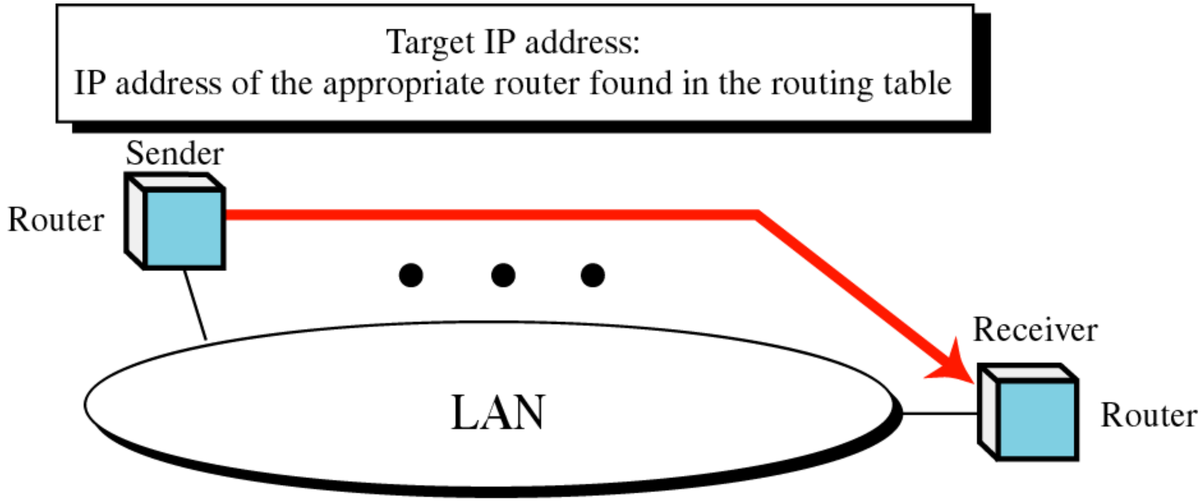


Case 1. A host has a packet to send to another host on the same network.

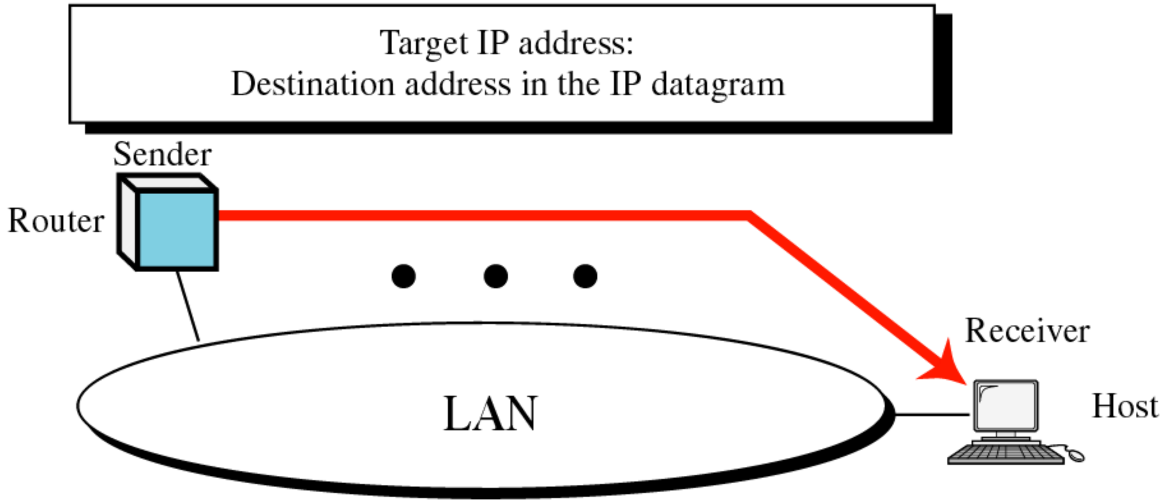


Case 2. A host wants to send a packet to another host on another network. It must first be delivered to the default router.

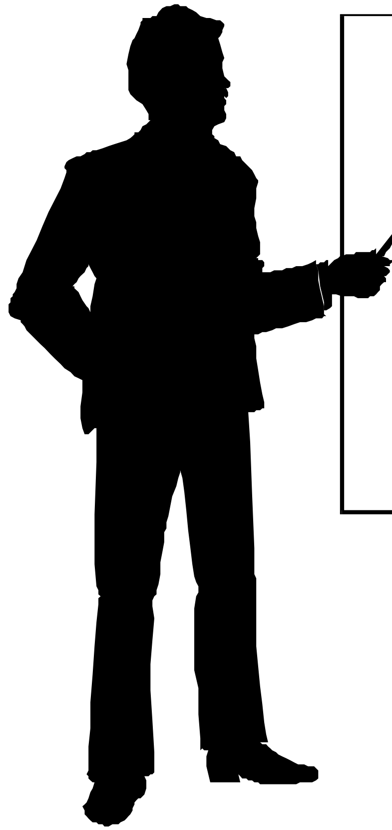
Four Cases using ARP (cont'd)



Case 3. A router receives a packet to be sent to a host on another network. It must first be delivered to the appropriate router.



Case 4. A router receives a packet to be sent to a host on the same network.



Distributed Routing

- Two standard distributed routing algorithms
  - Link State (LS) routing
  - Distance Vector (DV) routing

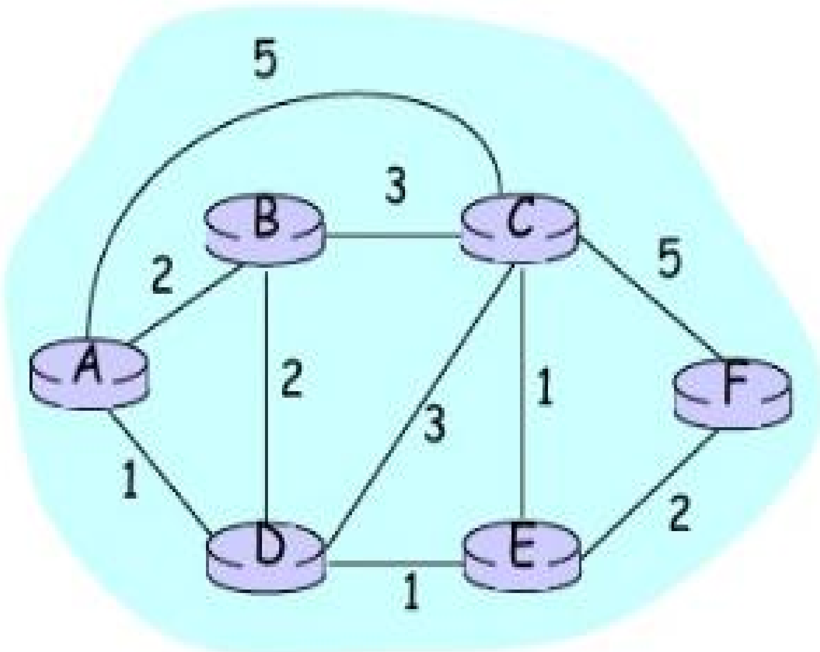
Link State *vs* Distance Vector

- Both assume that
  - The address of each neighbor is known
  - The cost of reaching each neighbor is known
- Both find global information
  - By exchanging routing info among neighbors
- Differ in the information exchanged and route computation
  - LS: tells every other node its distances to neighbors
  - DV: tells neighbors its distance to every other node

Link State Algorithm

- Basic idea: Distribute link state packet to all routers
  - Topology of the network
    - Cost of each link in the network
- Each router independently computes optimal paths
  - From itself to every destination
  - Routes are guaranteed to be loop free if
    - Each router sees the same cost for each link
    - Uses the same algorithm to compute the best path

Topology Database: Example



c(x,y)	A	B	C	D	E	F
A	0	2	5	1	∞	∞
B	2	0	3	2	∞	∞
C	5	3	0	3	1	5
D	1	2	3	0	1	∞
E	∞	∞	1	1	0	2
F	∞	∞	5	∞	2	0

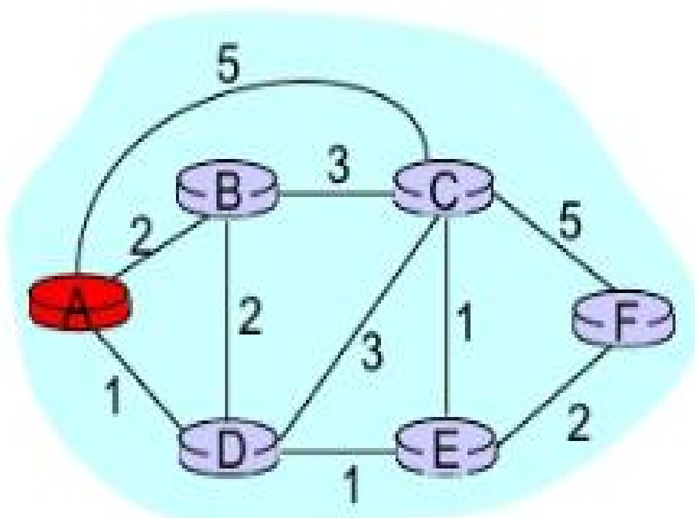
link state database

Algorithm (at Node X)

- Initialization
  - N = {X}
  - For all nodes V
    - If V adjacent to X, D(V) = C(X,V) else D(V) = ∞
- Loop
  - Find U not in N such that D(U) is the smallest
  - Add U into set N
  - Update D(V) for all V not in N
    - D(V) = min{D(V), D(U) + C(U,V)}
  - Until all nodes in N

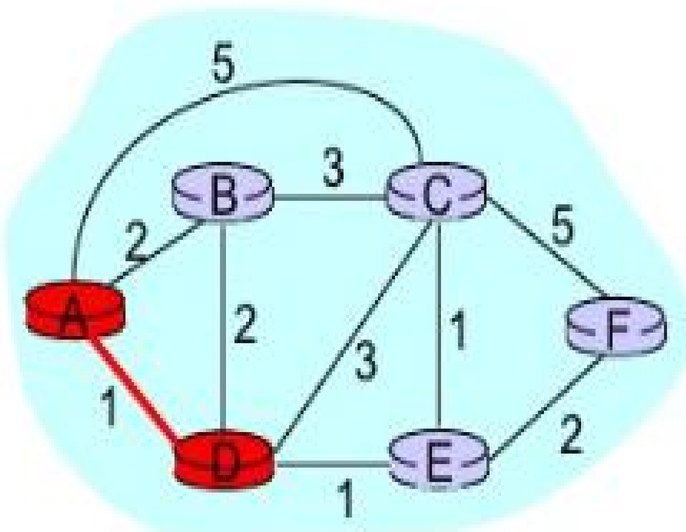
Example: Dijkstra's Algorithm

Step	start N	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(F),p(F)
0	A	2,A	5,A	1,A	∞	∞
1						
2						
3						
4						
5						



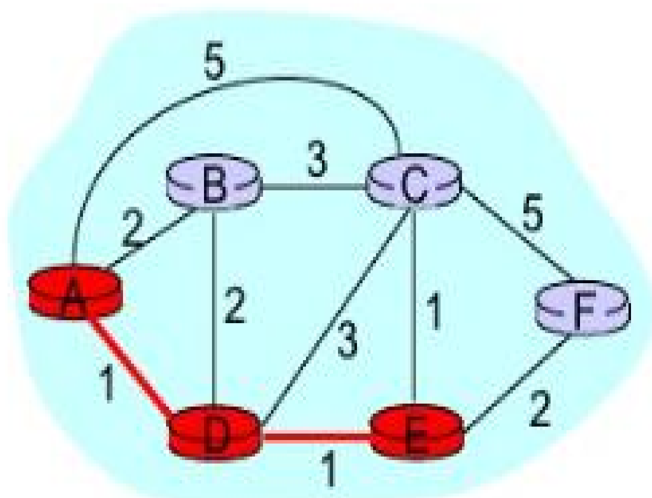

```
Initialization:
  N = {A};
  for all nodes v
    if v adjacent to A
      then D(v) = c(A,v);
      else D(v) = ∞;
...
```

Step	start N	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(F),p(F)
0	A	2,A	5,A	1,A	∞	∞
1	AD		4,D		2,D	∞
2						
3						
4						
5						



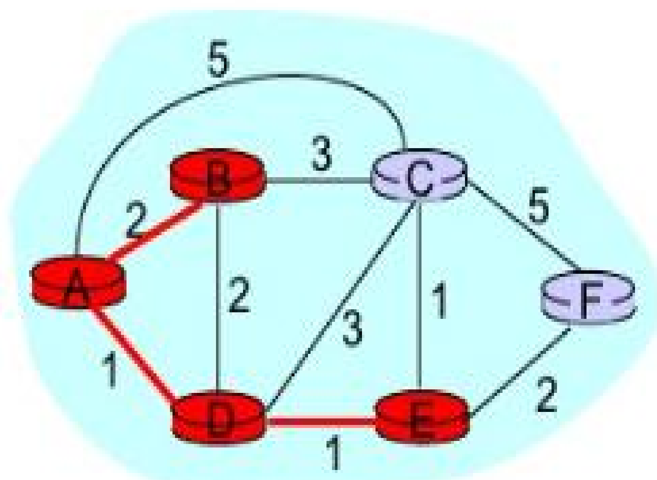

```
...
8   Loop
9     find w not in N s.t. D(w) is a minimum;
10    add w to N;
5     update D(v) for all v adjacent
      to w and not in N:
12      D(v) = min( D(v), D(w) + c(w,v) );
13  until all nodes in N;
```

Step	start N	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(F),p(F)
0	A	2,A	5,A	1,A	∞	∞
1	AD		4,D		2,D	∞
2	ADE		3,E			4,E
3						
4						
5						



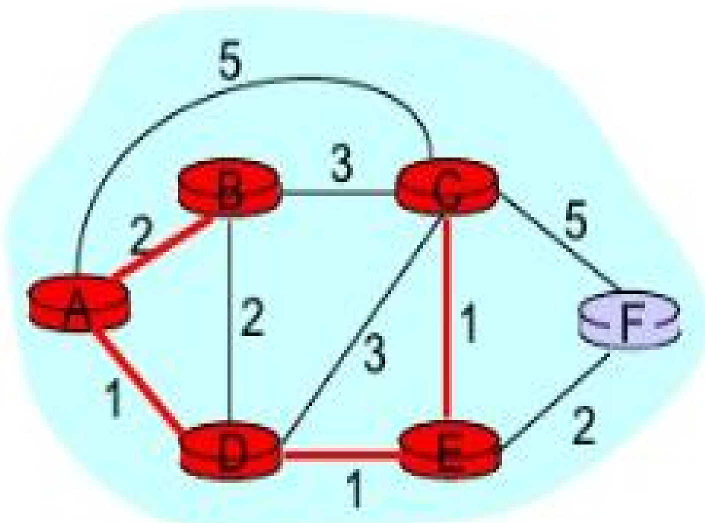

```
...
Loop
  find w not in N s.t. D(w) is a minimum;
  add w to N;
  update D(v) for all v adjacent
  to w and not in N:
    D(v) = min( D(v), D(w) + c(w,v) );
until all nodes in N;
```

Step	start N	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(F),p(F)
0	A	2,A	5,A	1,A	∞	∞
1	AD		4,D		2,D	∞
2	ADE		3,E			4,E
3	ADEB					
4						
5						



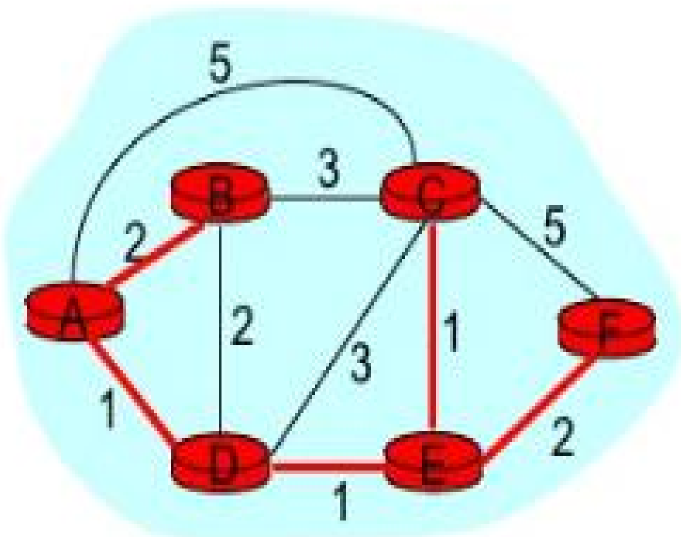

```
...
Loop
   find w not in N s.t. D(w) is a minimum;
   add w to N;
  update D(v) for all v adjacent
   to w and not in N:
     D(v) = min( D(v), D(w) + c(w,v) );
 until all nodes in N;
```

Step	start N	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(F),p(F)
0	A	2,A	5,A	1,A	∞	∞
1	AD		4,D		2,D	∞
2	ADE		3,E			4,E
3	ADEB					
4	ADEBC					
5						




```
...
Loop
  find w not in N s.t. D(w) is a minimum;
  add w to N;
  update D(v) for all v adjacent
   to w and not in N:
     D(v) = min( D(v), D(w) + c(w,v) );
until all nodes in N;
```

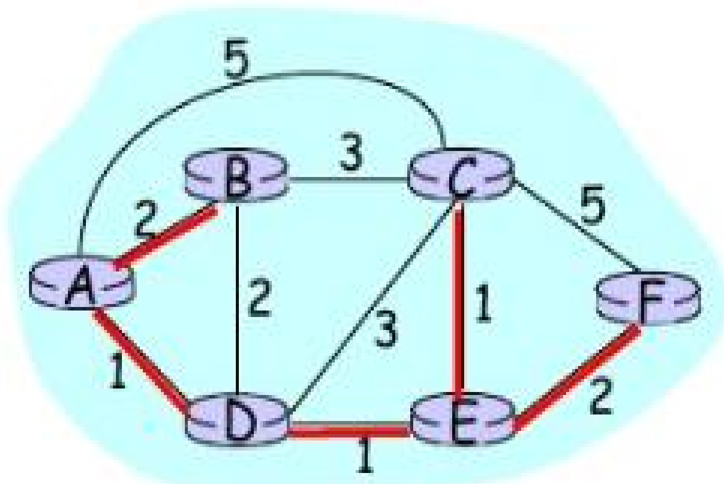
Step	start N	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(F),p(F)
0	A	2,A	5,A	1,A	∞	∞
1	AD		4,D		2,D	∞
2	ADE		3,E			4,E
3	ADEB					
4	ADEBC					
→ 5	ADEBCF					



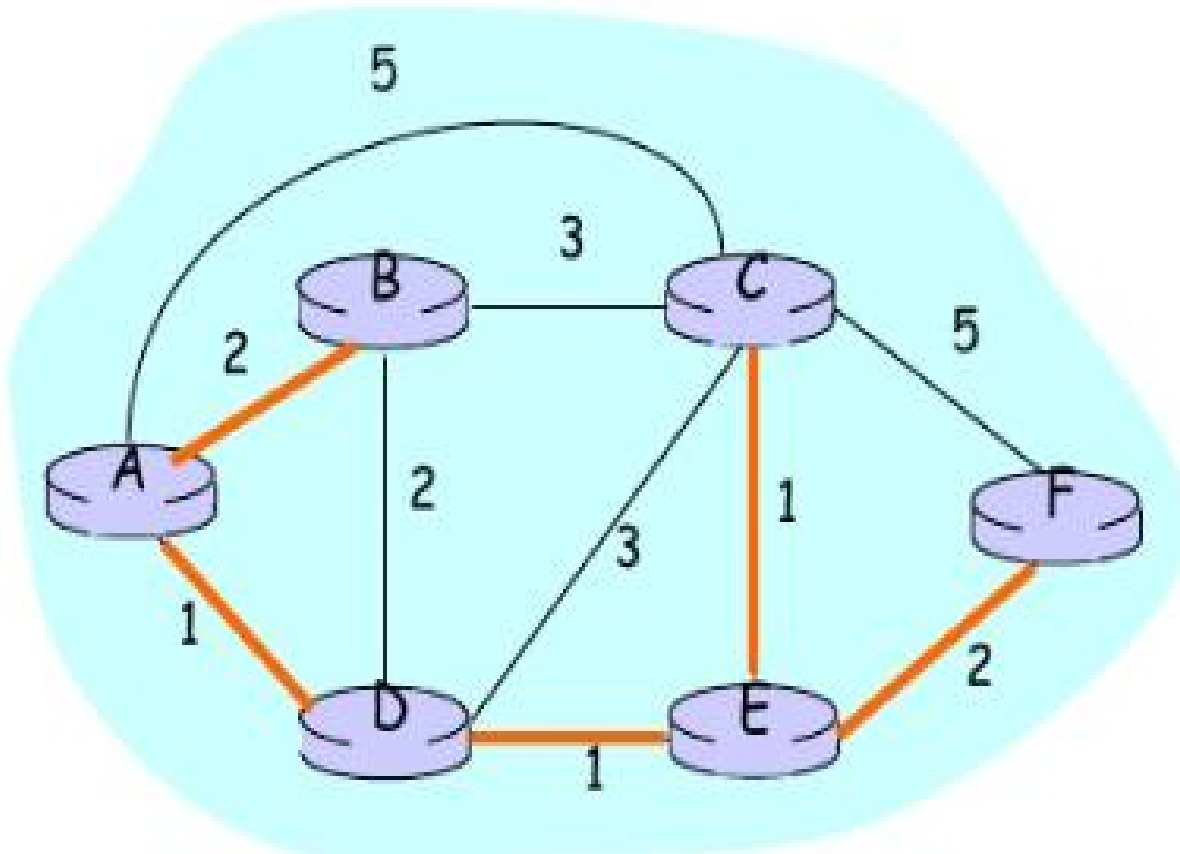

```
...
Loop
  find w not in N s.t. D(w) is a minimum;
  add w to N;
  update D(v) for all v adjacent
   to w and not in N:
     D(v) = min( D(v), D(w) + c(w,v) );
until all nodes in N;
```

Dijkstra's Algorithm: In a Nutshell

Step	start N	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(F),p(F)
0	A	2,A	5,A	1,A	infinity	infinity
1	AD	2,A	4,D		2,D	infinity
2	ADE	2,A	3,E			4,E
3	ADEB		3,E			4,E
4	ADEBC					4,E
5	ADEBCF					



dest	next
B	B
C	D
D	D
E	D
F	D



Distance Vector Routing

a) The least-cost route between any two nodes is the route with minimum distance.

b) Each node maintains a vector(table) of minimum distances to every node.

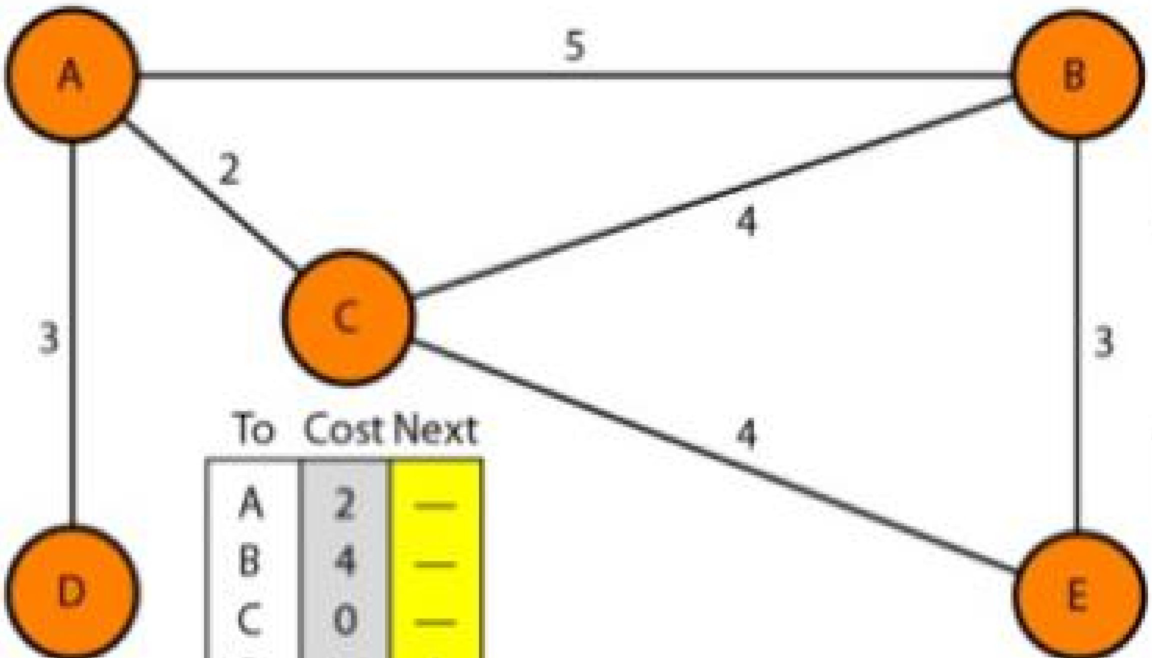
c) The table at each node also guides the packets to the desired node by showing the showing the next hop routing.

Example:

Assume each node as the cities.

Lines as the roads connecting them.

Final Distance vector routing tables



To	Cost	Next
A	0	—
B	5	—
C	2	—
D	3	—
E	6	C

A's table

To	Cost	Next
A	5	—
B	0	—
C	4	—
D	8	A
E	3	—

B's table

To	Cost	Next
A	2	—
B	4	—
C	0	—
D	5	A
E	4	—

C's table

To	Cost	Next
A	3	—
B	8	A
C	5	A
D	0	—
E	9	A

D's table

To	Cost	Next
A	6	C
B	3	—
C	4	—
D	9	C
E	0	—

E's table