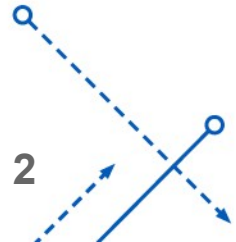
The University of Danang
University of Science and Technology

# CHAPTER 4
# NETWORK LAYER

FACULTY OF INFORMATION TECHNOLOGY
PhD. LE TRAN DUC

# OUTLINE

1. Overview of Network Layer
2. IPv4
3. IPv6
4. NAT – Network Address Translation
5. Subnet Addressing
6. Routing protocols



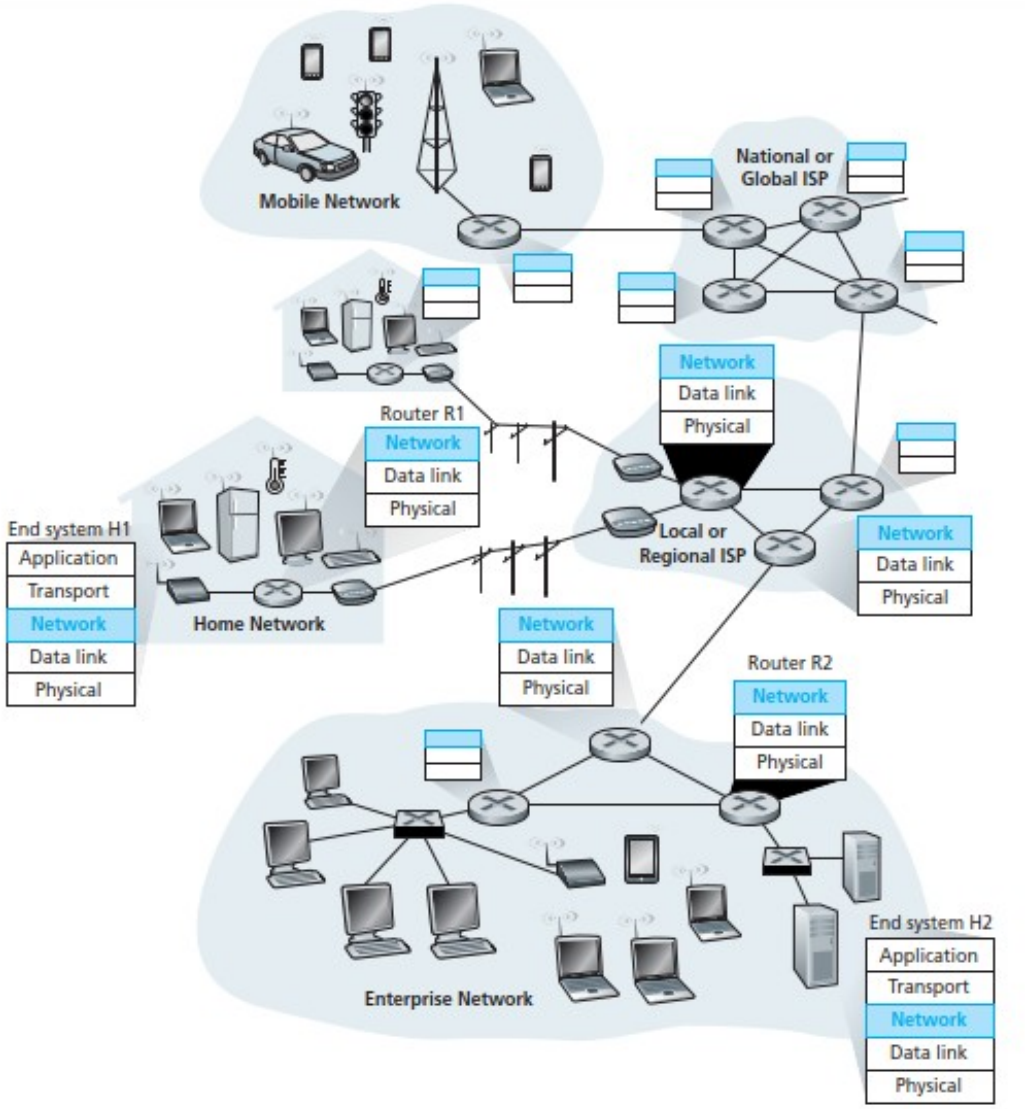
The University of Danang
University of Science and Technology

# 1. OVERVIEW OF NETWORK LAYER


Faculty of Information Technology
PhD. Le Tran Duc

# NETWORK LAYER

- Transport segment from sending to receiving host
- On sending side encapsulates segments into IP packet
- On receiving side, delivers segments to transport layer
- Network layer protocols in *every* host, router
- Router examines header fields in all IP packets passing through it



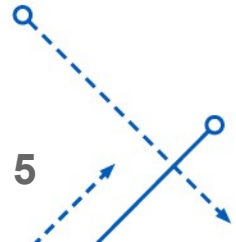
# TWO KEY NETWORK LAYER FUNCTIONS

***Network-layer functions:***

- ***Forwarding****:* move packets from router's input to appropriate router output
- ***Routing****:* determine route taken by packets from source to destination
  - ***Routing algorithms***

***Analogy: taking a trip***

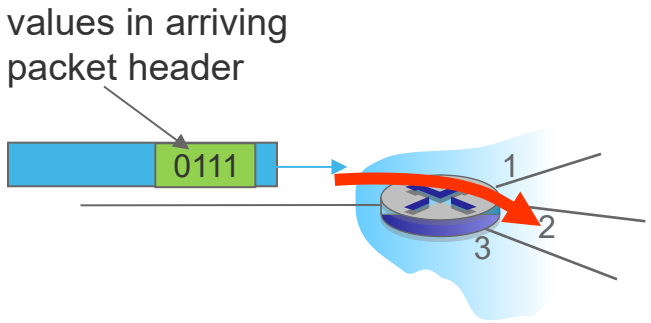
- *Forwarding:* process of getting through single interchange
- *Routing:* process of planning trip from source to destination



# NETWORK LAYER: DATA PLANE, CONTROL PLANE

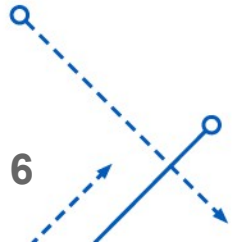
## *Data plane*

- Local, per-router function
- Determines how packet arriving on router-input-port is forwarded to router-output-port
- Forwarding function



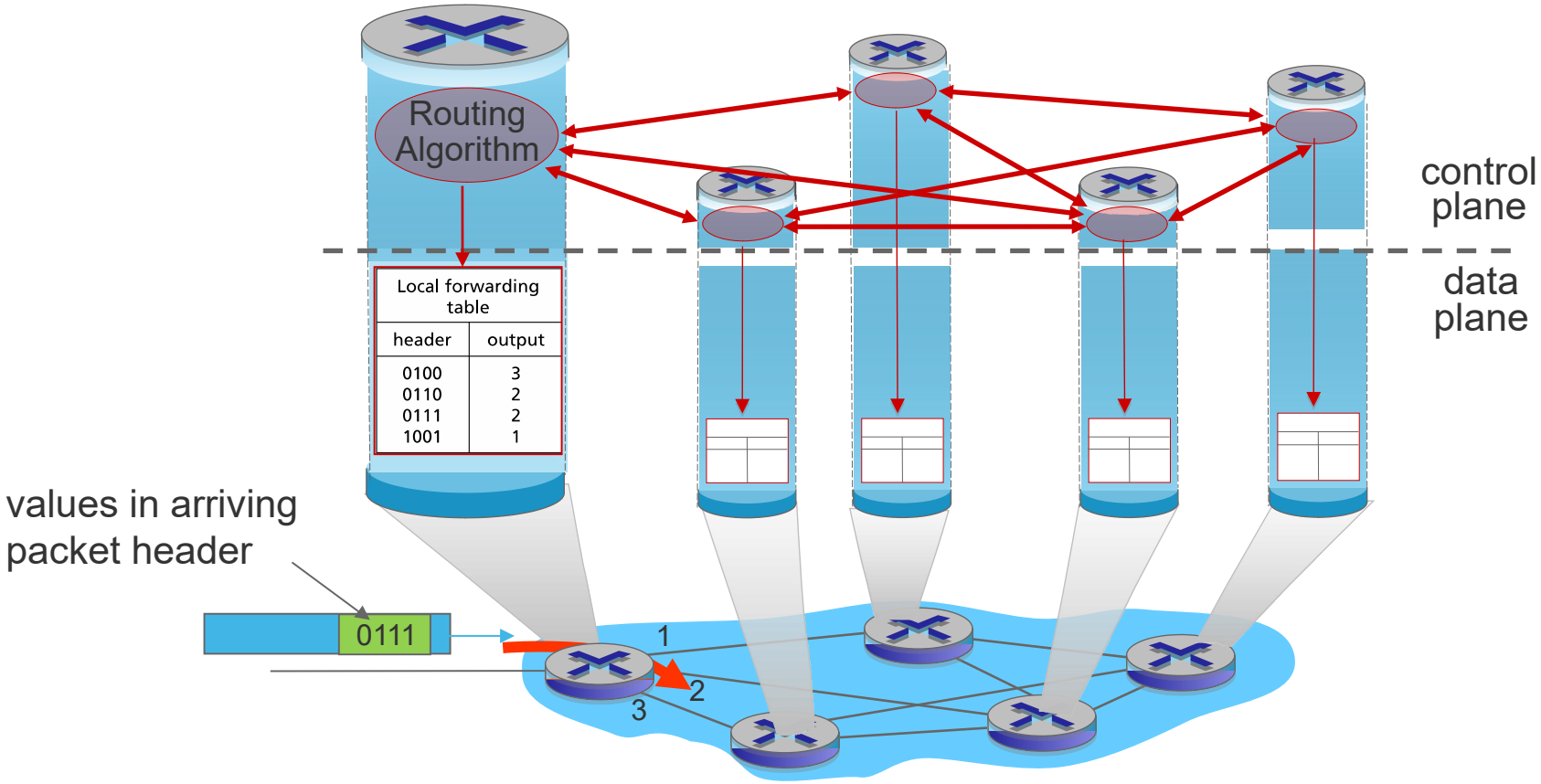

## *Control plane*

- Network-wide logic
- Determines how packet is routed among routers along end-to-end path from source host to destination host
- Two control-plane approaches:
  - ***Traditional routing algorithms***: implemented in routers
  - ***Software-defined networking (SDN)***: implemented in (remote) servers



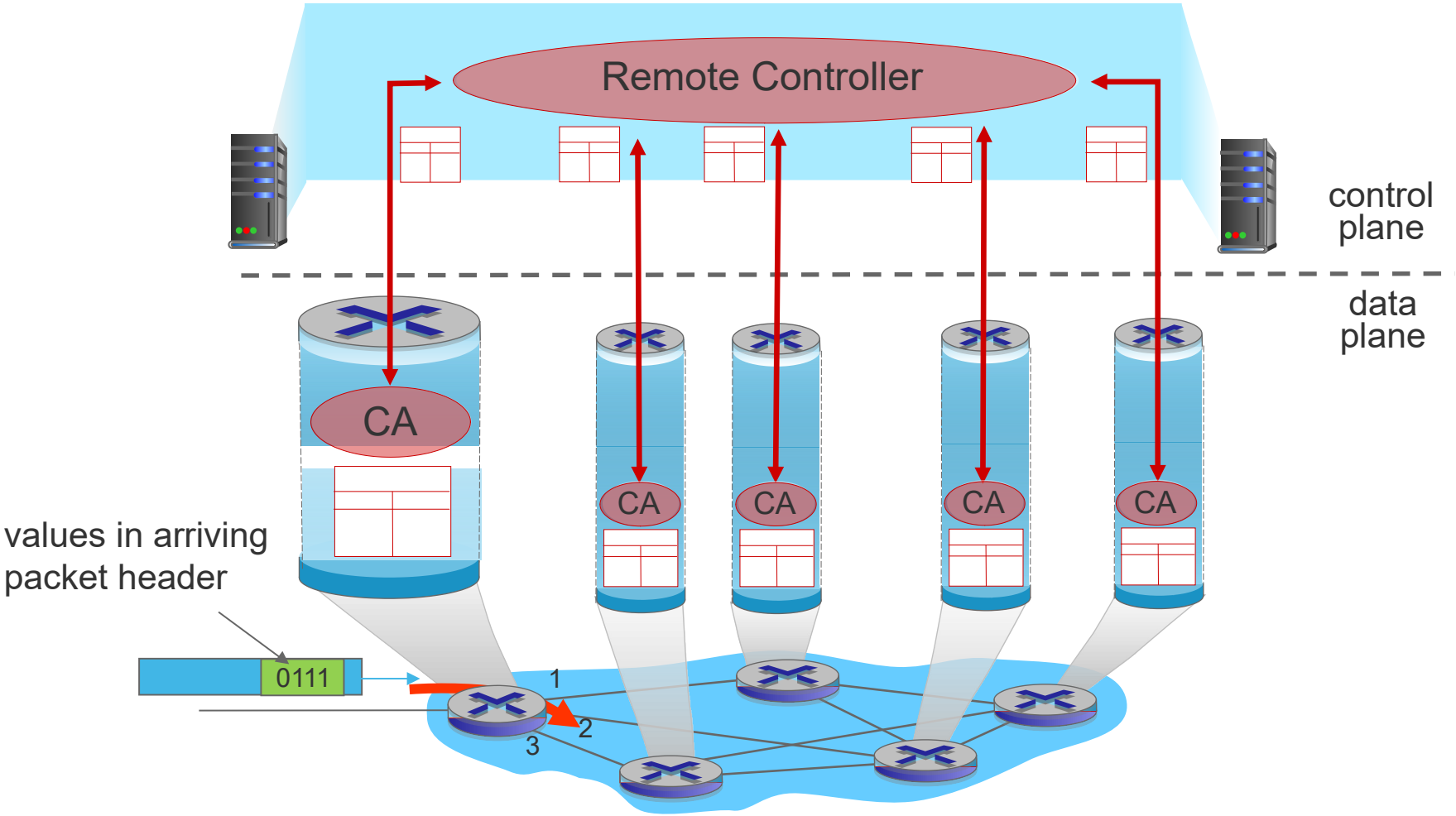
# PER-ROUTER CONTROL PLANE

Individual **routing algorithm** components ***in each and every router*** interact in the control plane



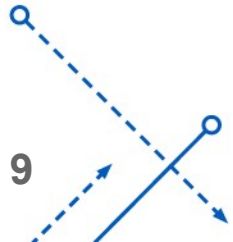
# LOGICALLY CENTRALIZED CONTROL PLANE

A **distinct** (typically remote) **controller** interacts with local control agents (CAs)



# NETWORK SERVICE MODEL

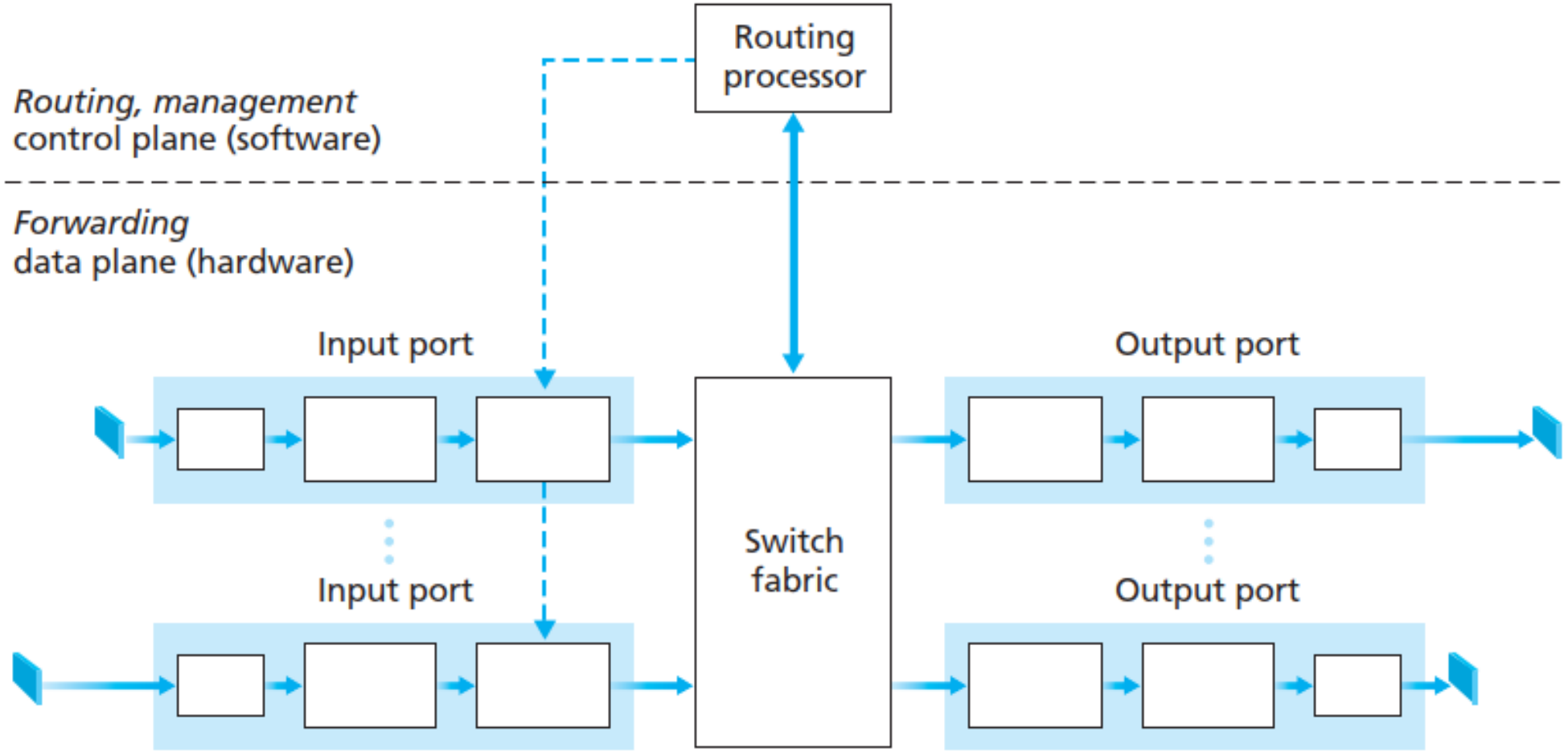
- The network service model defines the characteristics of end-to-end delivery of packets between sending and receiving hosts.
- Network layer could provide following services:
  - **Guaranteed delivery** → Guarantees that a packet will arrive at the destination host
  - **Guaranteed delivery with bounded delay** → Guarantees that a packet delivered within a specified delay bound
  - **In-order packet delivery** → Guarantees that packets arrive at the destination in the order that they were sent
  - **Guaranteed minimal bandwidth** → Reserves bandwidth
  - **Security** → The network layer could encrypt all IP packets at the source and decrypt them at the destination, thereby providing confidentiality to all transport-layer segments



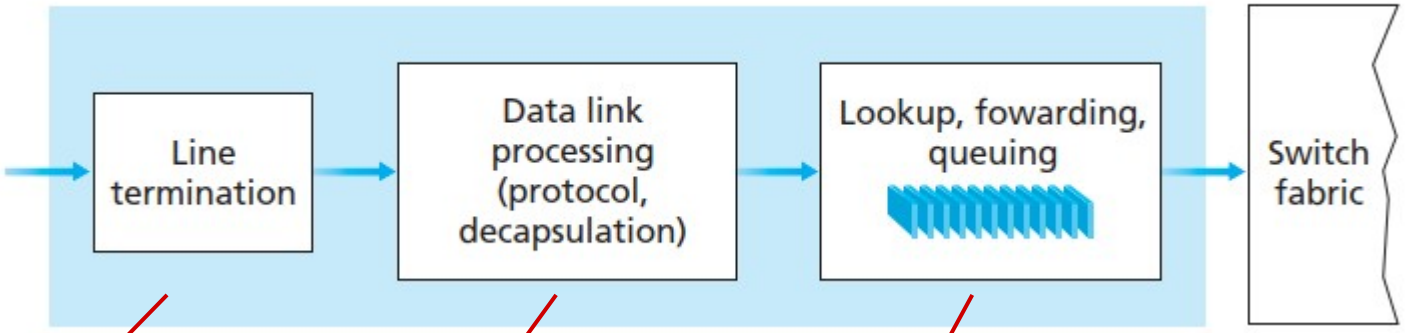
# QOS REQUIREMENTS

| S. No | Service Type | General QoS | | | Medical QoS | | |
|---|---|---|---|---|---|---|---|
| | | Data Rate (kbps) | Delay (ms) | Loss Rate | Data Rate (kbps) | Delay (ms) | Loss Rate |
| 1 | Audio | 56 - 64 kbps | <150 ms | <0.1% | 4 - 26 kbps | 150 - 400 ms | 3% |
| 2 | Video | 4 - 60 Mbps | <150 ms | <0.0001% | 32 - 384 kbps | 150 - 400 ms | 1% |
| 3 | FTP | 11.8 kbps | ~10 sec. | 0 (zero) | 16.99 kbps | 177.6 ms | 0 (zero) |
| 4 | VOIP | 64 kbps | <150 ms, phone to phone delay | <1% | 500 kbps | 150 - 240 ms (downlink) 200 ms (uplink) | 0.01% |
| 5 | Video conferencing | 24 - 1920 kbps | <150 ms | <0.01% | 640 kbps - 5 Mbps | <250 ms E2E | 1% |
| 6 | Images | <100 kbps | <10 sec. | NA | <1 Mbps | ~10 sec. | NA |
| 7 | Web browsing | ~10 kbps | <4 sec., per page | 0 (zero) | 10 kbps | ~2 sec. | 0 (zero) |
| 8 | Email | <10 kbps | <4 sec. | 0 (zero) | <30.5 kbps | <400 ms | 0 (zero) |

# ROUTER ARCHITECTURE OVERVIEW



# INPUT PORT FUNCTIONS




**physical layer**:
bit-level reception

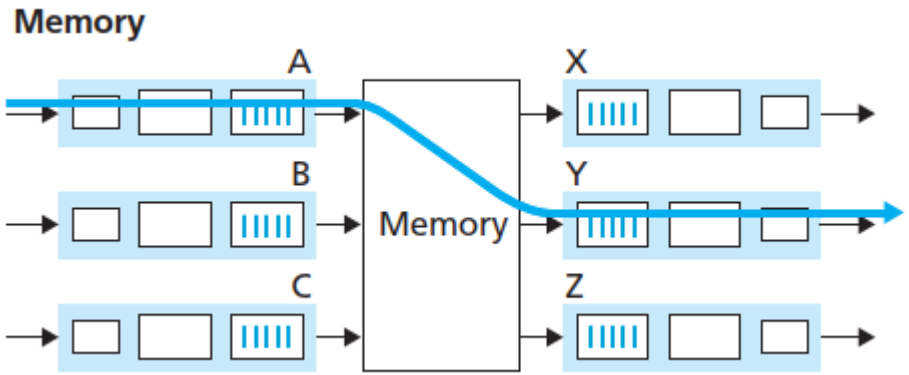
**data link layer**:
e.g., Ethernet

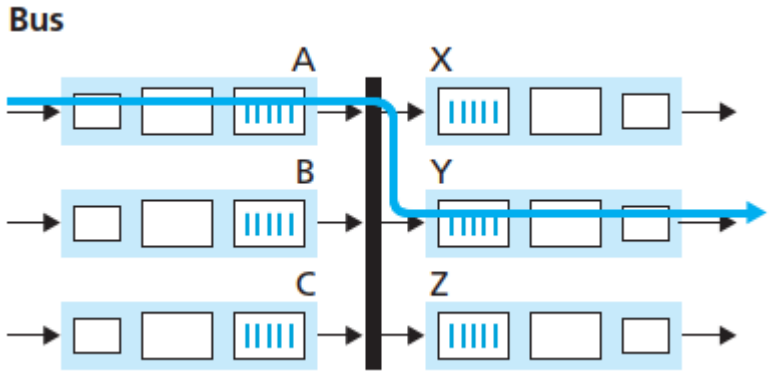
**Decentralized switching**:

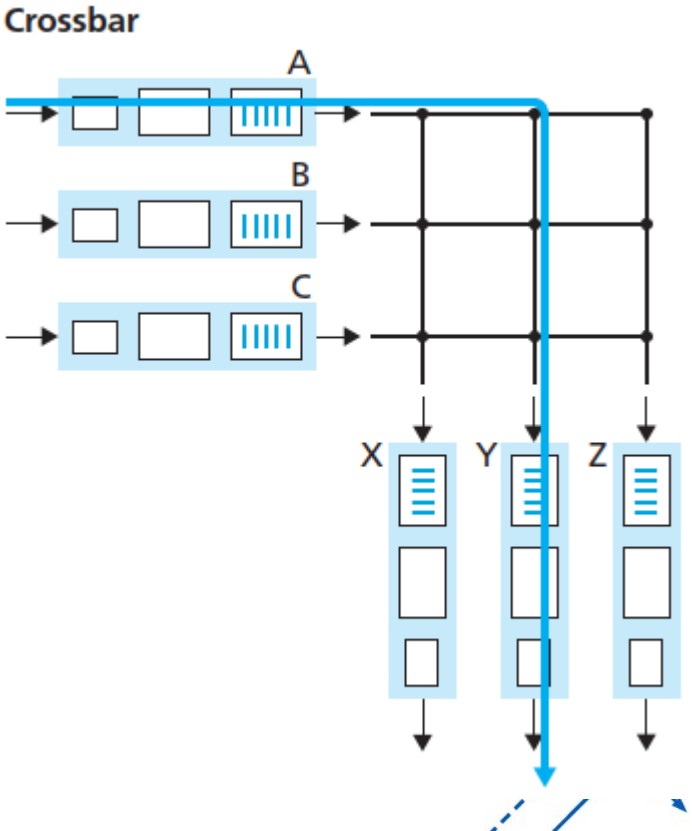
- Using header field values, **lookup** output port using forwarding table in input port memory *("match plus action")*
- Goal: complete input port processing at 'line speed'
- Queuing: if packet arrive faster than forwarding rate into switch fabric
- ***Destination-based forwarding***: forward based only on destination IP address (traditional)
- ***Generalized forwarding***: forward based on any set of header field values

# SWITCHING FABRICS

- Transfer packet from input buffer to appropriate output buffer
- Switching rate: rate at which packets can be transfer from inputs to outputs
  - Often measured as multiple of input/output line rate
  - N inputs: switching rate N times line rate desirable
- Three types of switching fabrics:

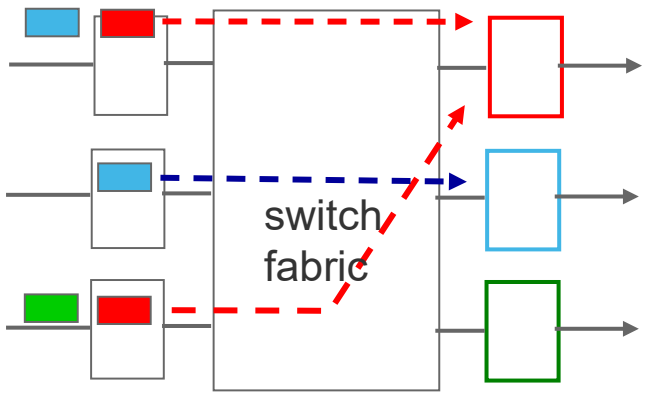


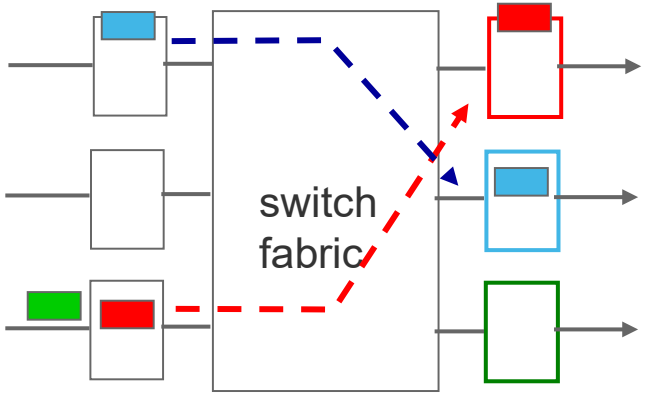




# INPUT PORT QUEUEING

- Queueing may occur at input queues and output queues.
  - *Queueing delay and loss due to **input** buffer overflow!*
- **Head-of-the-Line (HOL) blocking**: queued packet at front of queue prevents others in queue from moving forward

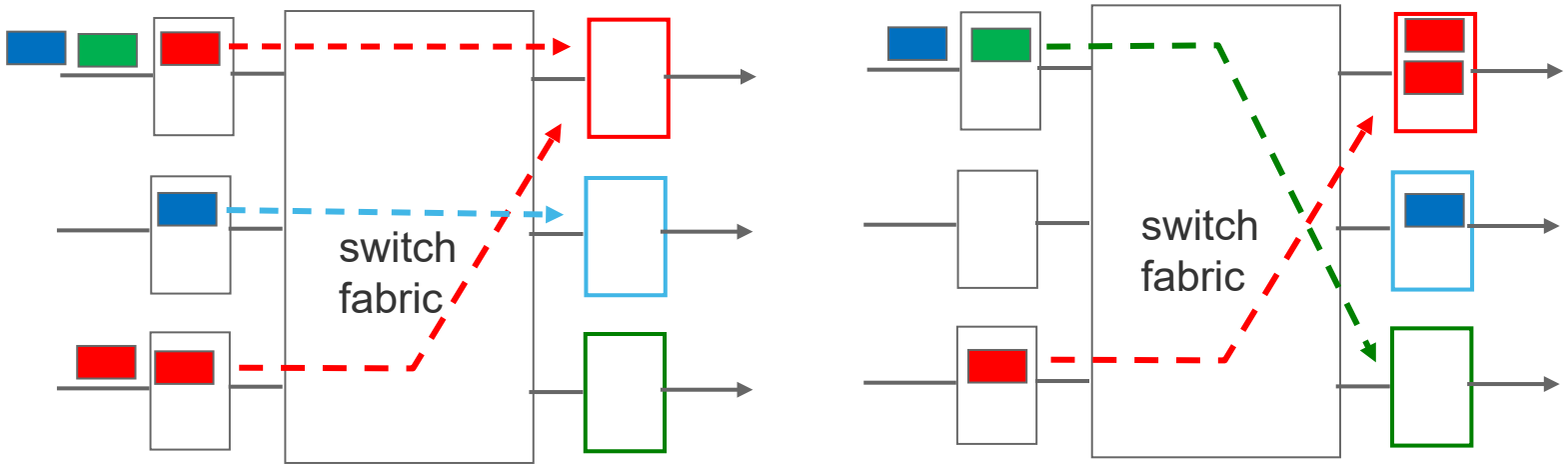



output port contention:
only one red packet can be transferred.
*lower red packet is blocked*




one packet time later: green packet experiences HOL blocking

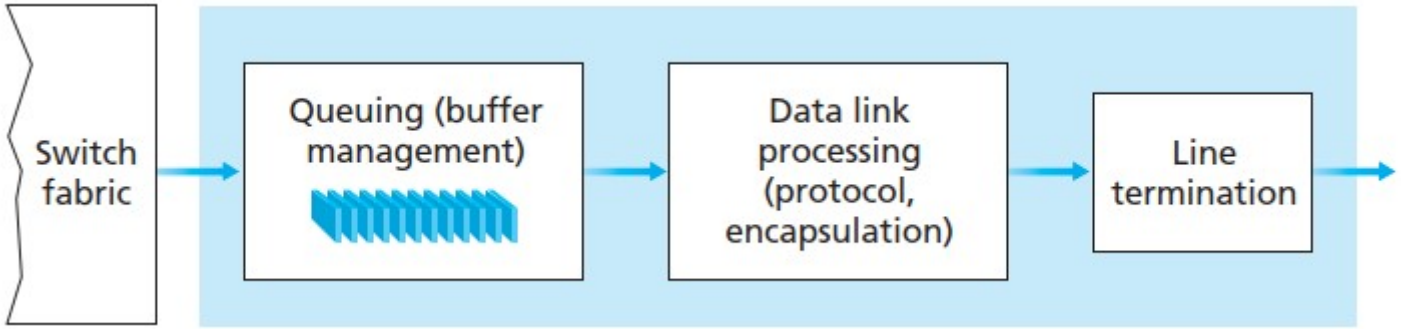
# OUTPUT PORT QUEUEING




at *t*, packets more from input to output

one packet time later

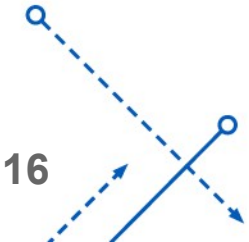
- Buffering when arrival rate via switch exceeds output line speed
- *Queueing (delay) and loss due to **output** port buffer overflow!*



# HOW MUCH BUFFERING?

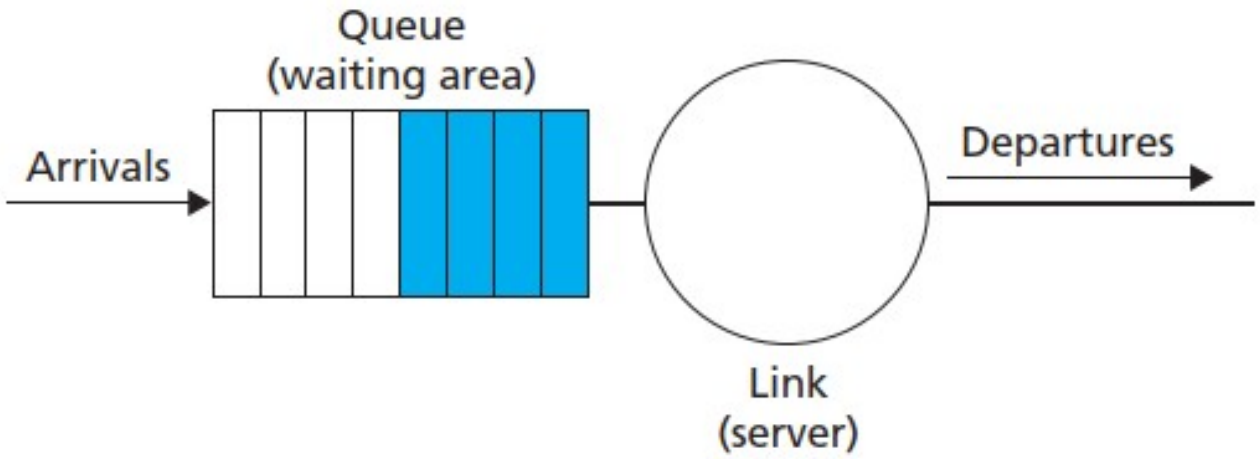
- RFC 3439 rule: average buffering equal to “typical” RTT (say 250 msec) times link capacity C
  - e.g., C = 10 Gpbs link: 2.5 Gbit buffer
- Recent recommendation: with *N* flows, buffering equal to

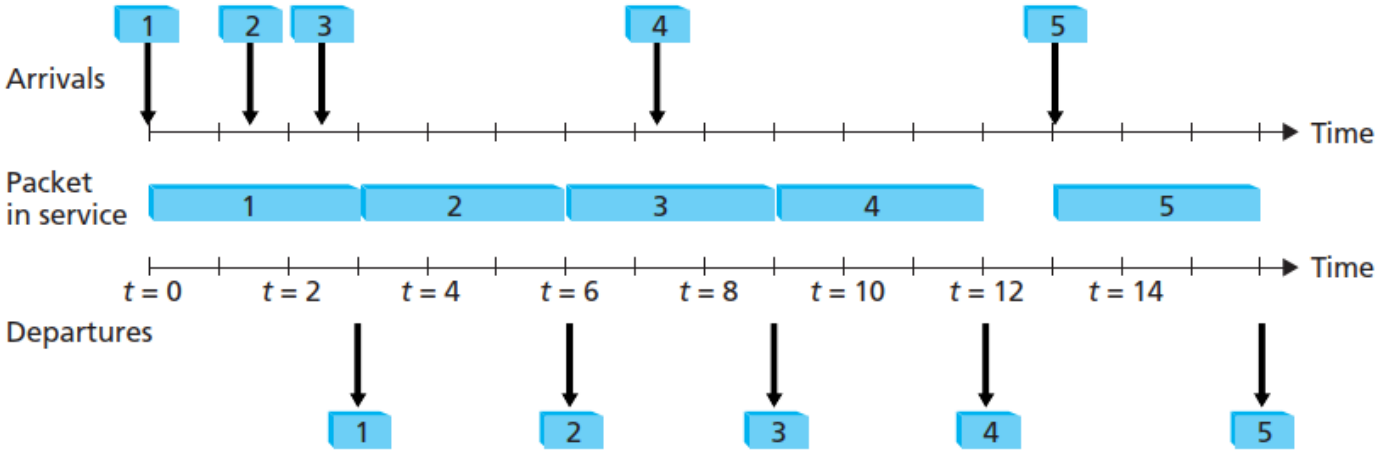
$$\frac{RTT \cdot C}{\sqrt{N}}$$



# PACKET SCHEDULING MECHANISMS

- ***Scheduling****:* choose next packet to send on link
- ***FIFO (first in first out) scheduling****:* send in order of arrival to queue
  - *Discard policy:* if packet arrives to full queue: who to discard?
    - ***tail drop****:* drop arriving packet
    - ***priority****:* drop/remove on priority basis
    - ***random****:* drop/remove randomly

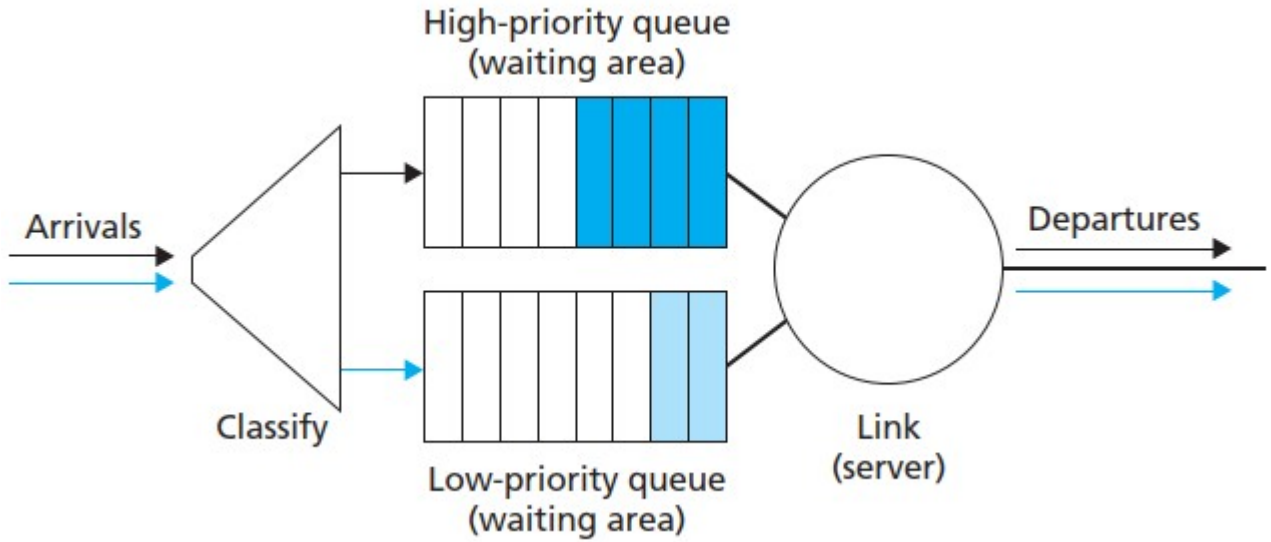


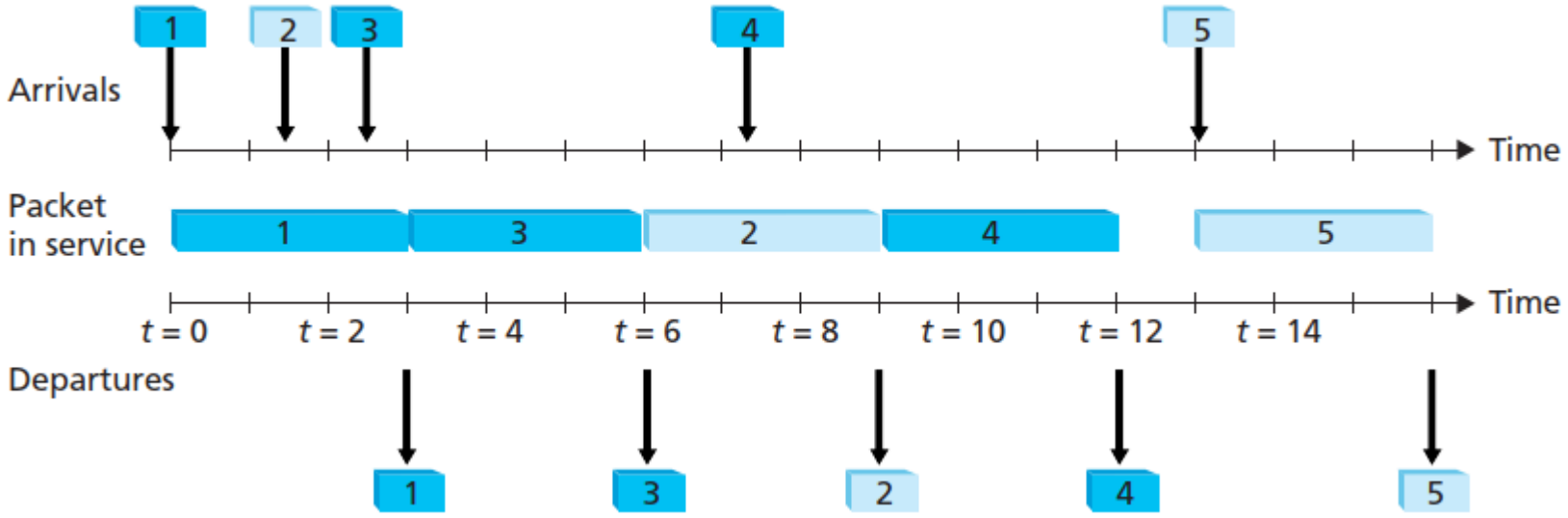



# PRIORITY SCHEDULING

***Priority scheduling**:* send highest priority queued packet

- Multiple *classes*, with different priorities
  - Class may depend on marking or other header info, e.g. IP source/dest, port numbers, etc.

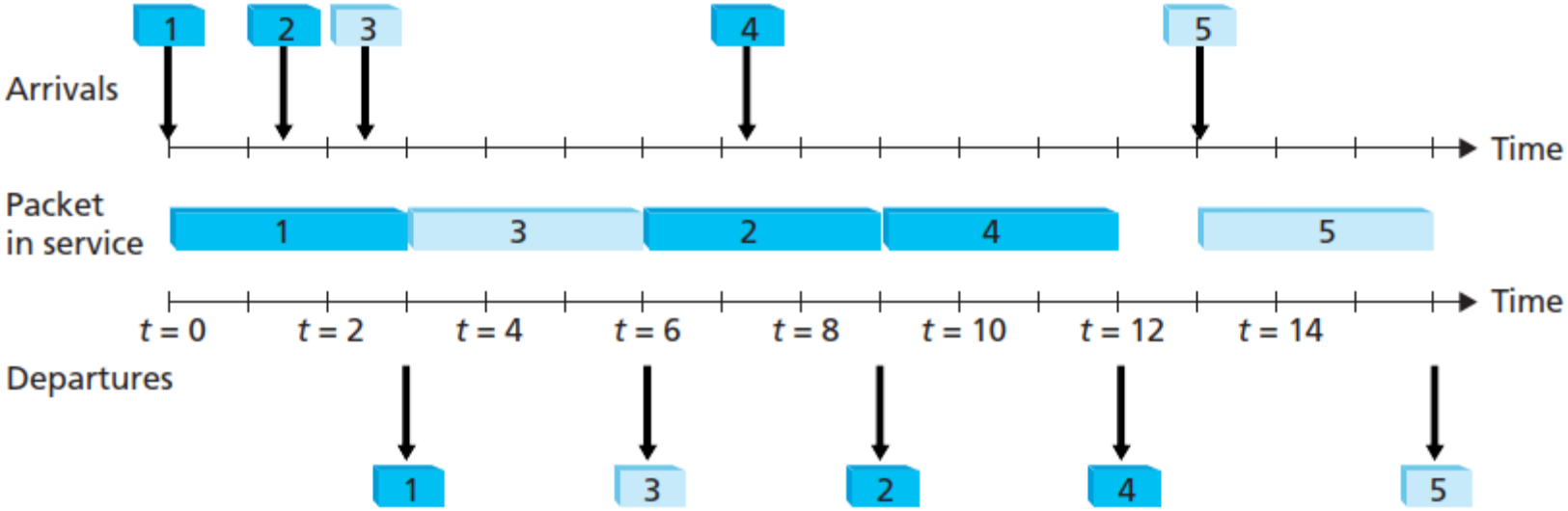


# ROUND ROBIN (RR) SCHEDULING

***Round Robin (RR) scheduling:***

- Multiple classes
- Cyclically scan class queues, sending one complete packet from each class (**if available**)

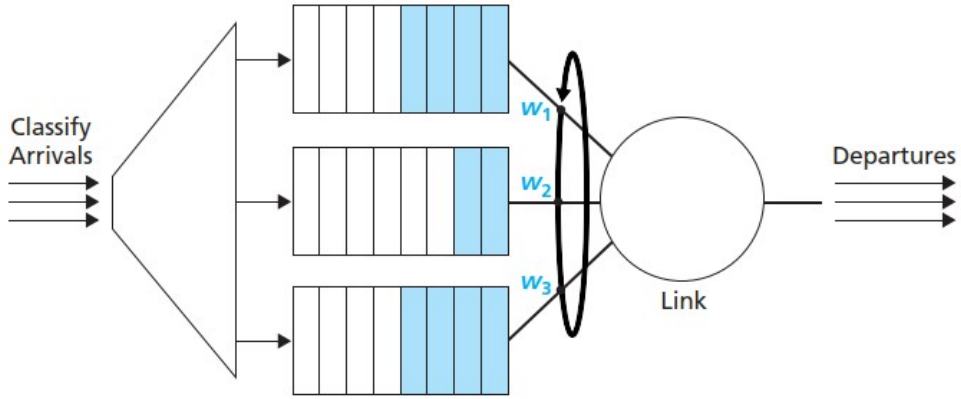


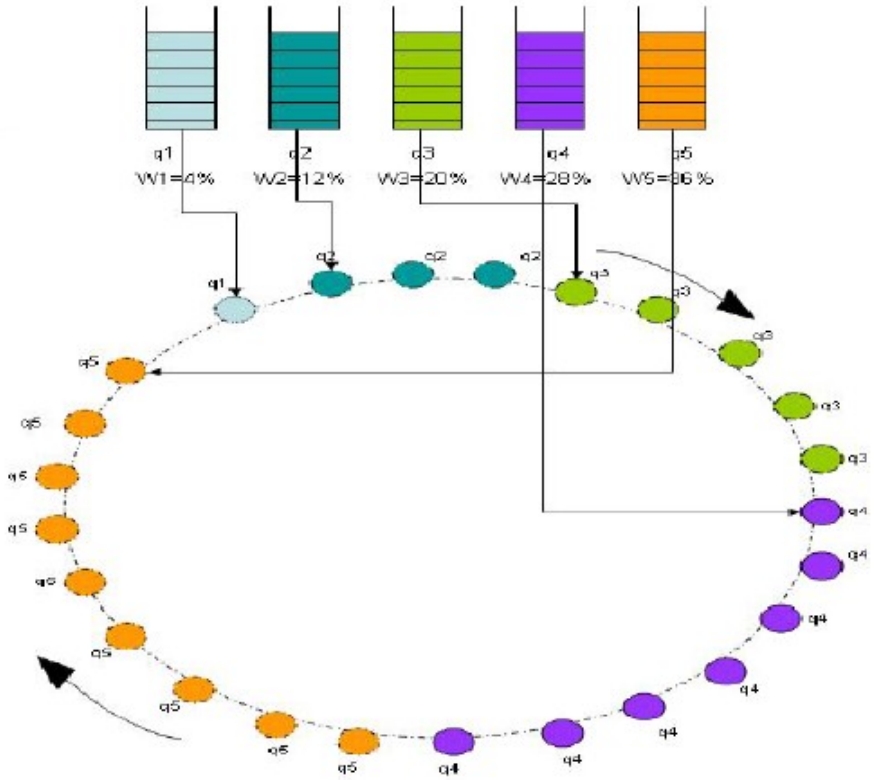

The two-class robin queue in operation
(**work-conserving queuing**)

# WEIGHTED FAIR QUEUING (WFQ)

***Weighted Fair Queuing (WFQ):***

- Arriving packets are classified and queued in the appropriate per-class waiting area
- As in round robin scheduling, a WFQ scheduler will serve classes in a circular manner
- Each class gets weighted amount of service in each cycle:
  - Each class, $i$, is assigned a weight, $w_i$
  - During any interval of time during which there are class $i$ packets to send, fraction of service equal to $w_i/(\sum w_j)$
  - For a link with transmission rate $R$, class $i$ will always achieve a throughput of at least $R.w_i/(\sum w_j)$



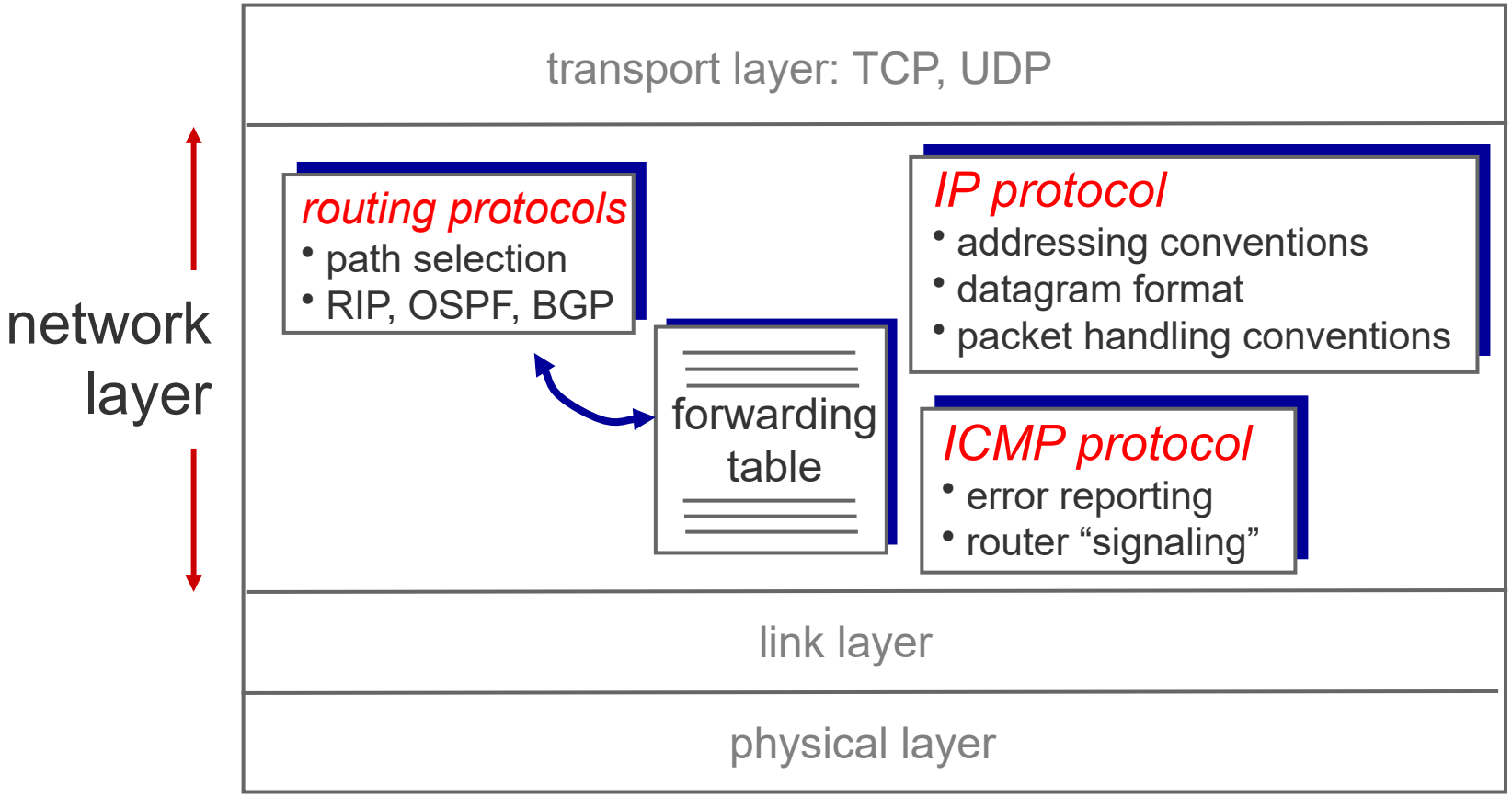



The University of Danang
University of Science and Technology

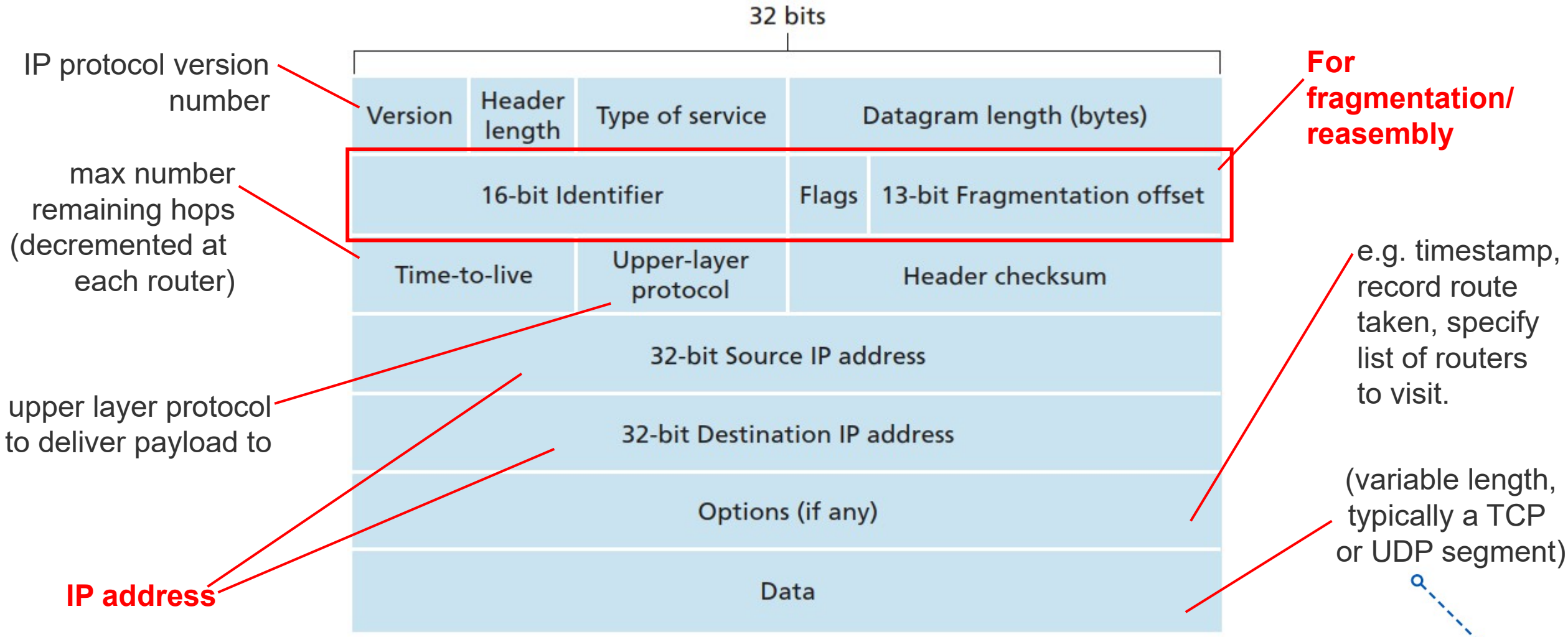
# 2. IPv4 (INTERNET PROTOCOL)


Faculty of Information Technology
PhD. Le Tran Duc

# THE INTERNET NETWORK LAYER

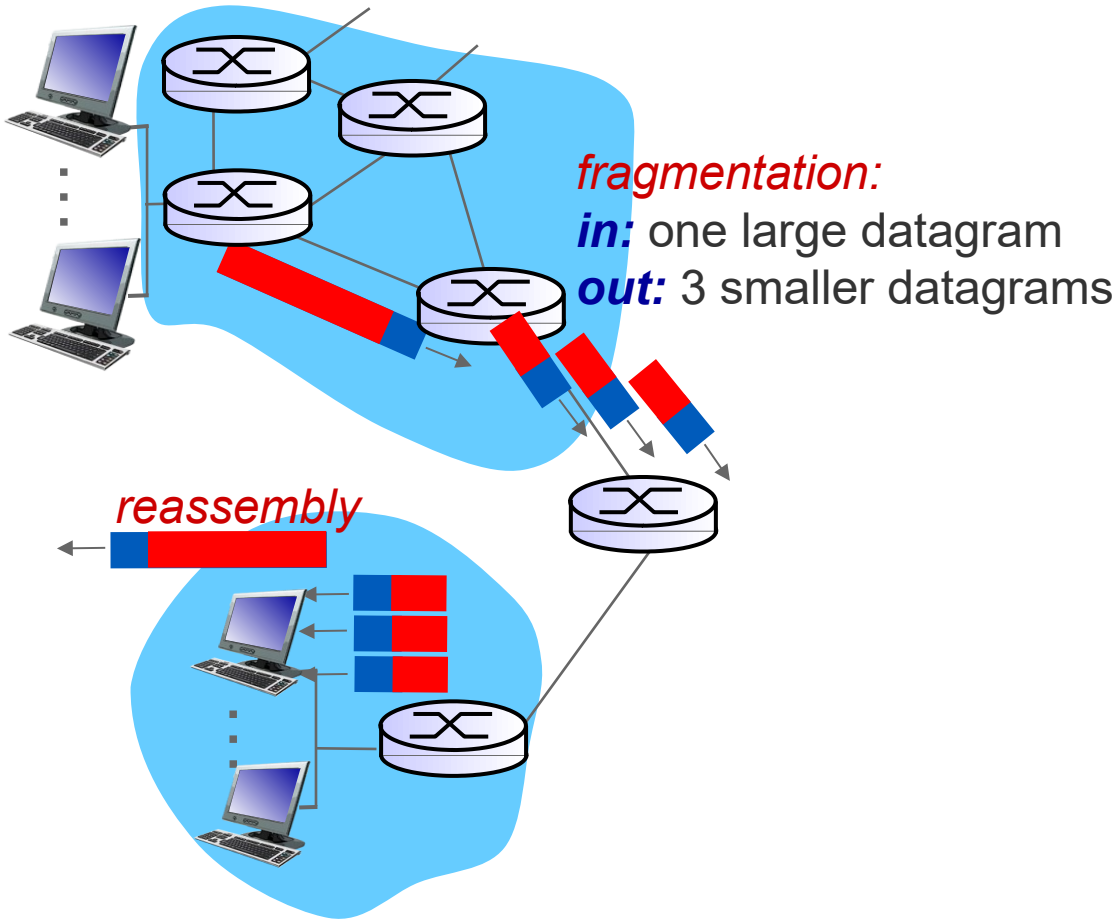


# IP PACKET (DATAGRAM) FORMAT



# IP FRAGMENTATION, REASSEMBLY

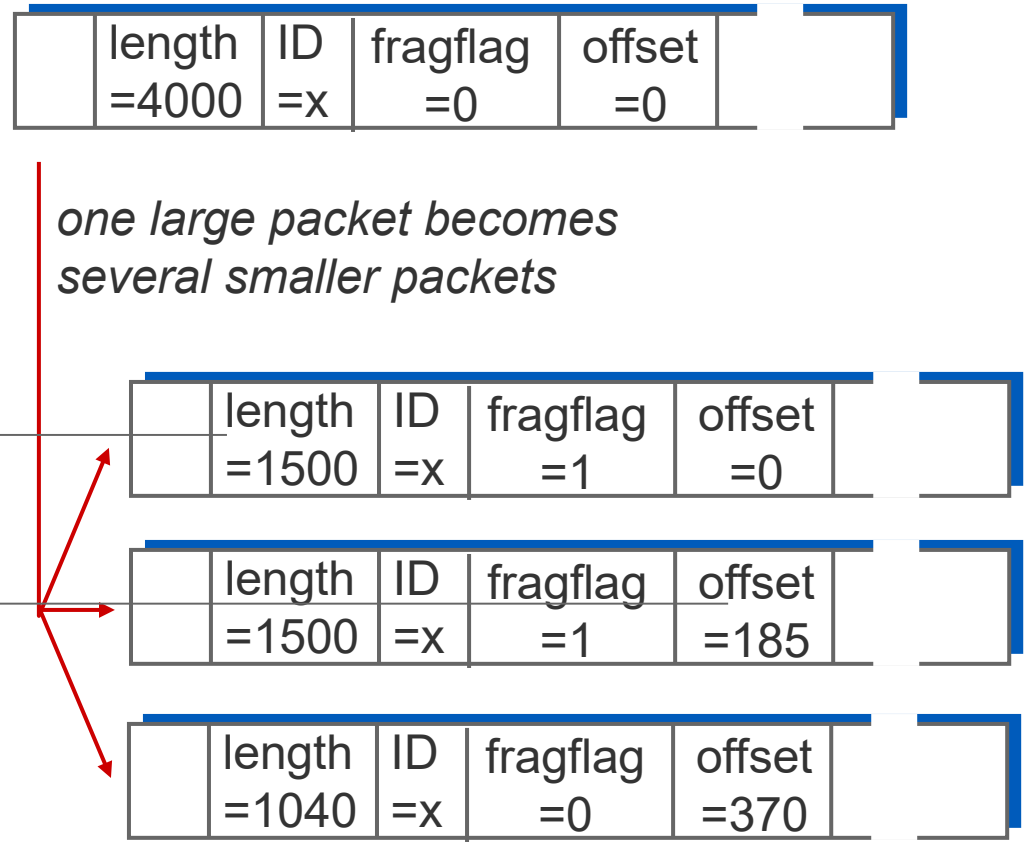
- Network links have **MTU** (**Maximum Transmission Unit**) - largest possible link-layer frame can carry.
  - Different link types, different MTUs
- Large IP packet divided ("***fragmented***") within net
  - One packet becomes several small packets
  - "***Reassembled***" only at final destination
  - IP header bits used to identify, order related fragments



# IP FRAGMENTATION, REASSEMBLY

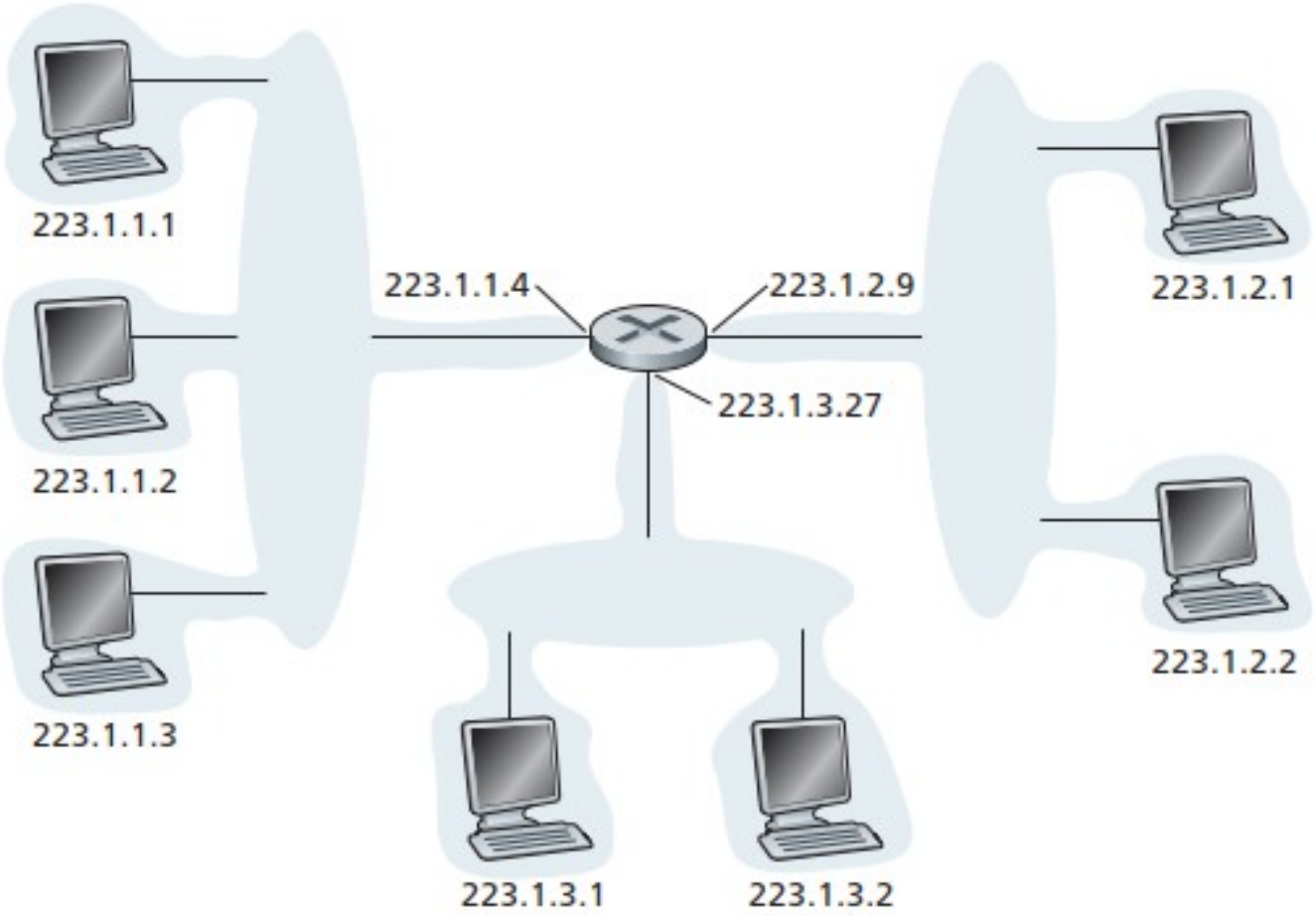
*Example:*

- 4000 byte datagram
- MTU = 1500 bytes



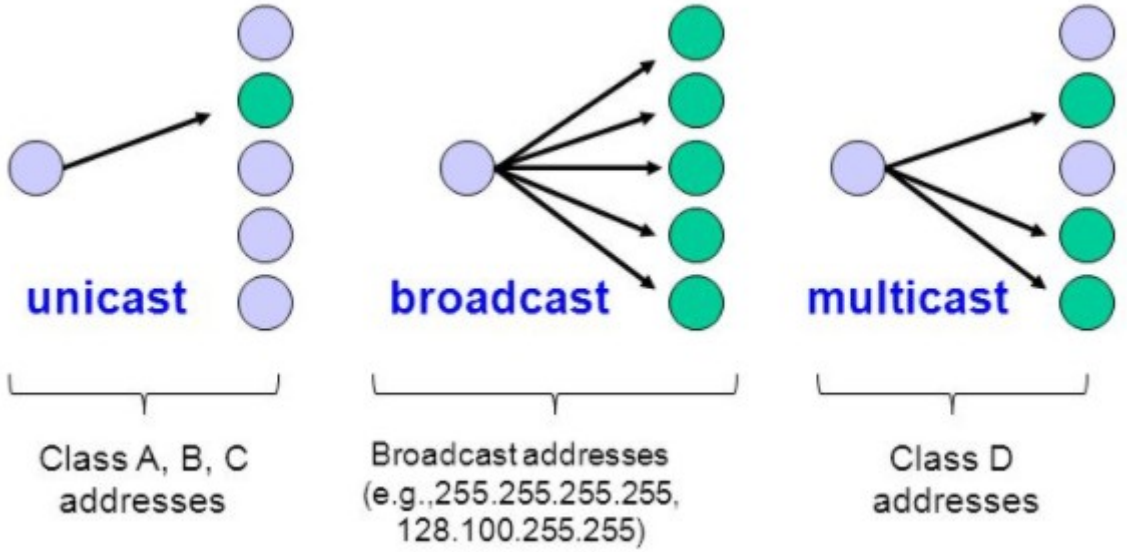
# IPv4 ADDRESSING

- ***IP address**:* 32-bit identifier for host, router ***interface***
- ***Interface**:* connection between host/router and physical link
  - router's typically have multiple interfaces
  - host typically has one or two interfaces (e.g., wired Ethernet, wireless 802.11)
- ***IP addresses associated with each interface***



# IPv4 ADDRESSING

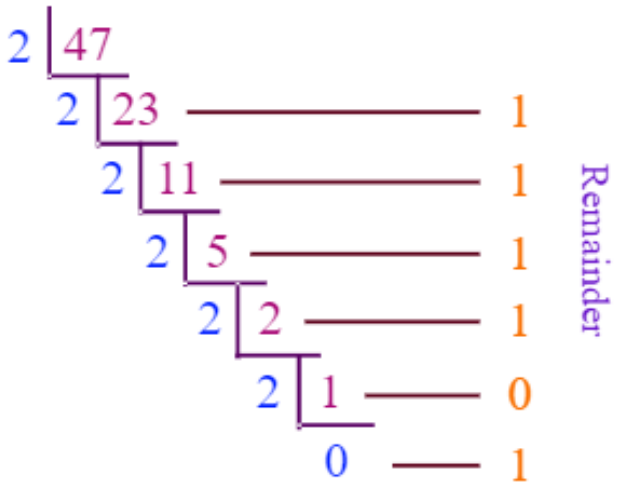
- ***IP address***: 32-bit (*4 octets, each octet 8 bits*)
- ***Notation:***
  - Binary notation
  - Dotted-decimal notation
- Value in each octet: **0 – 255**.
- Type of IPv4 address: **Unicast**, **Multicast**, **Broadcast**




**IP address: 192.168.21.76**

# SUPPLEMENT: USEFUL MATHEMATIC

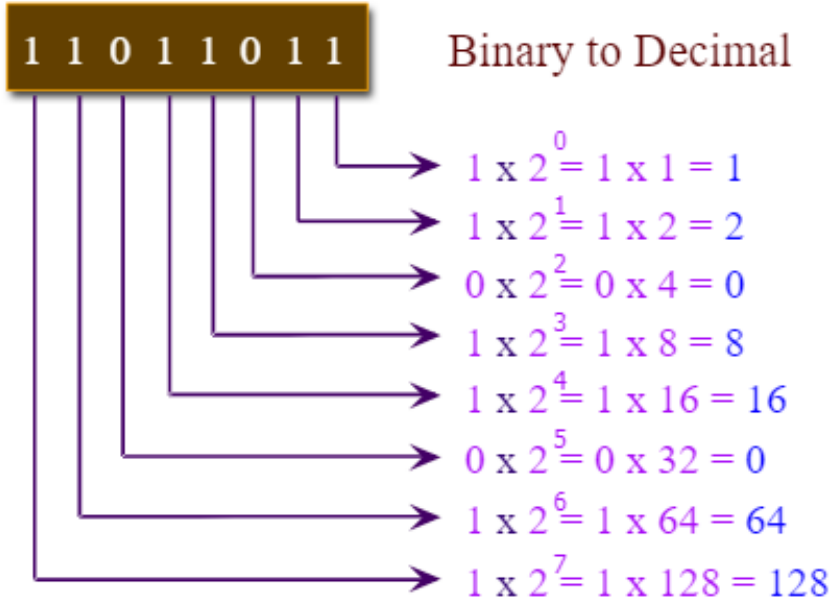
## Decimal to Binary




$(47)_{10} = (101111)_2$



## Binary to Decimal




$1 + 2 + 8 + 16 + 64 + 128 = 219$

$(11011011)_2 = (219)_{10}$



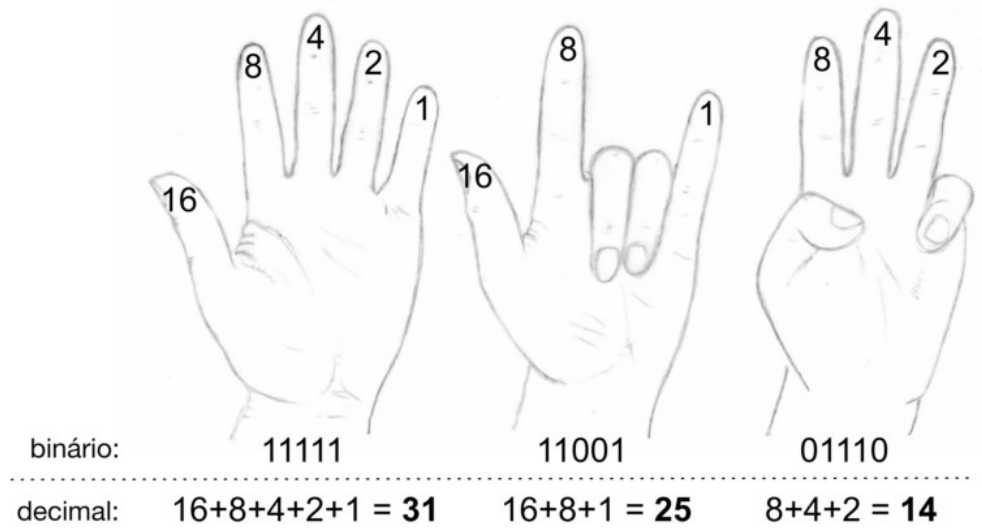
## Must remember table (1)

| | | | |
|---|---|---|---|
| $2^0$ | 1 | $2^9$ | 512 |
| $2^1$ | 2 | $2^{10}$ | 1024 |
| $2^2$ | 4 | $2^{11}$ | 2048 |
| $2^3$ | 8 | $2^{12}$ | 4096 |
| $2^4$ | 16 | $2^{16}$ | 65536 |
| $2^5$ | 32 | | |
| $2^6$ | 64 | | |
| $2^7$ | 128 | | |
| $2^8$ | 256 | | |

# SUPPLEMENT: USEFUL MATHEMATIC

| Must remember table (2) | |
|---|---|
| 0000 0000 | 0 |
| 1000 0000 | 128 |
| 1100 0000 | 192 |
| 1110 0000 | 224 |
| 1111 0000 | 240 |
| 1111 1000 | 248 |
| 1111 1100 | 252 |
| 1111 1110 | 254 |
| 1111 1111 | 255 |

| Must remember table (3) | |
|---|---|
| Borrowed bits (k) | Hops ($2^{8-k}$) |
| 1 | 128 |
| 2 | 64 |
| 3 | 32 |
| 4 | 16 |
| 5 | 8 |
| 6 | 4 |
| 7 | 2 |
| 8 | 1 |




**Given sequence of n binary-bits**
**→ Can create $2^n$ binary with n-bits**

$2^n$ binary numbers

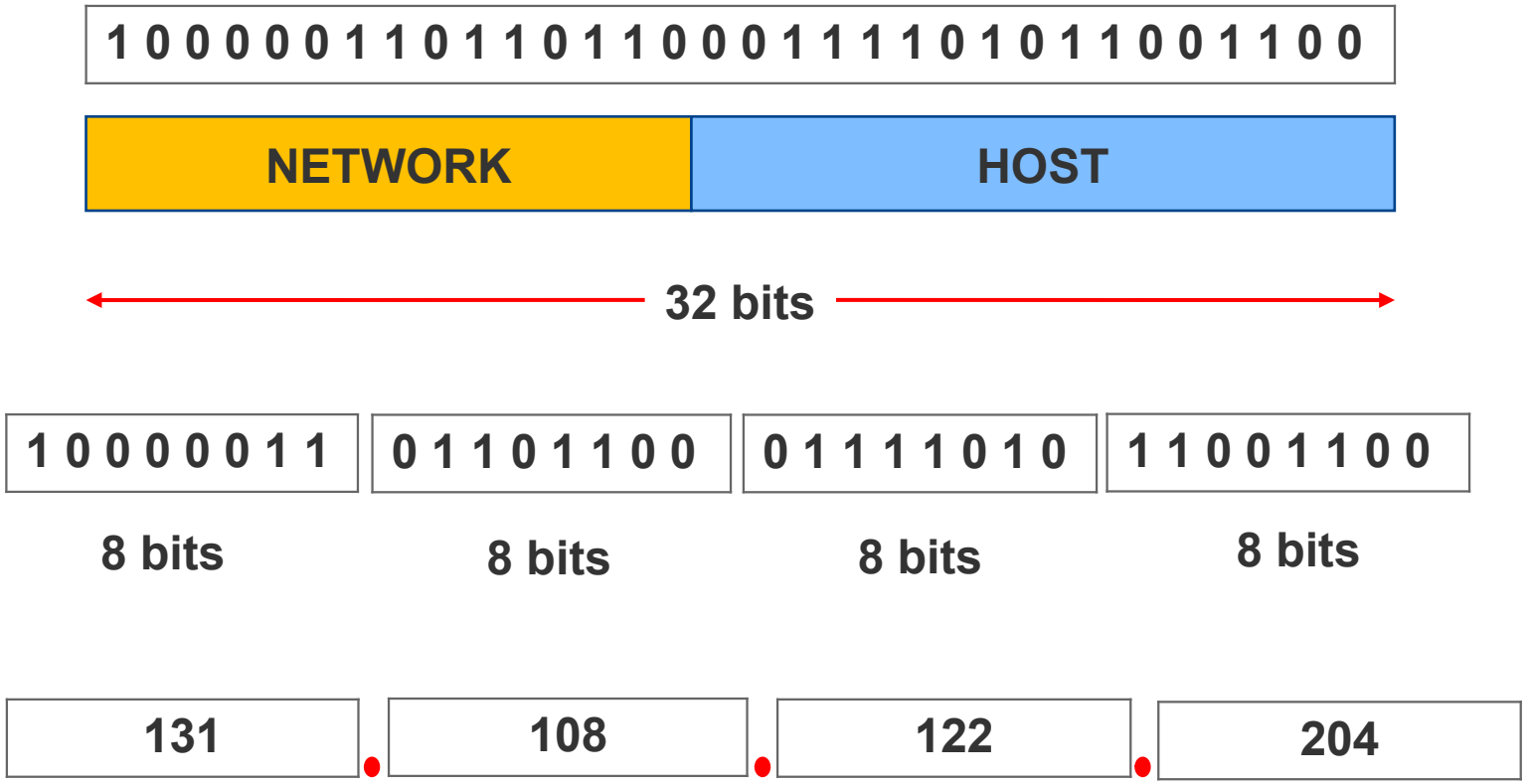
0 0 0 0…0 0 0 → 0

⋮

1 1 1 1…1 1 1 → $2^n$ - 1

n bits

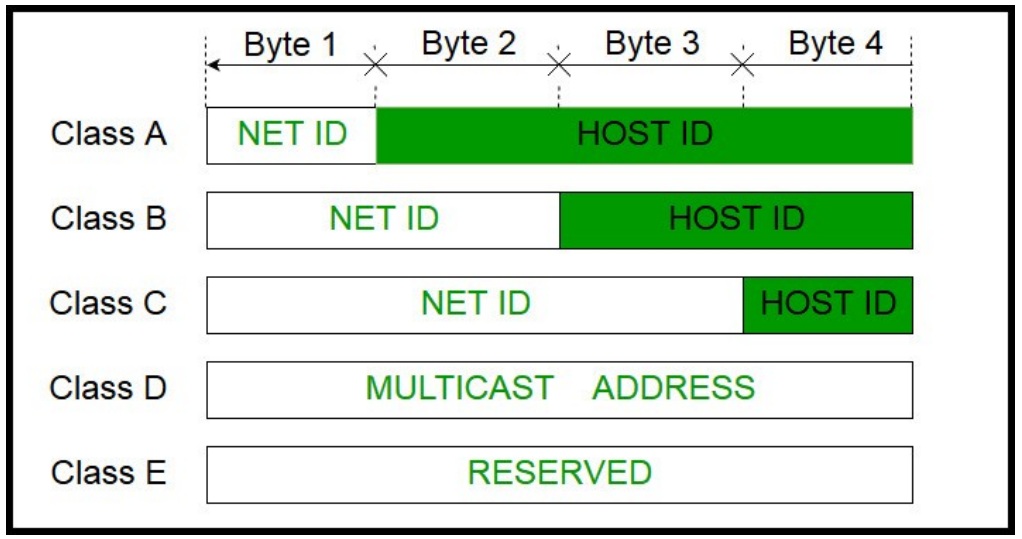
# IP ADDRESS STRUCTURE




- Network-bits cannot be zero at the same time
- If all Host-bits = 0 → Network Address
- If all Host-bits = 1 → Broadcast Address

# CLASSFUL ADDRESSING

- Different class addresses reserve different amounts of bits for Network and Host portions of the address
- Provide flexibility required to support different size networks
- In classful addressing, the address space is divided into five classes: **A, B, C, D, and E**




| Class | High Order Bits (First Octet) | Start Address | End Address |
|---|---|---|---|
| Class A | 0xxx xxxx | 1.0.0.0 | 127.255.255.255 |
| Class B | 10xx xxxx | 128.0.0.0 | 191.255.255.255 |
| Class C | 110x xxxx | 192.0.0.0 | 223.255.255.255 |
| Class D | 1110 xxxx | 224.0.0.0 | 239.255.255.255 |
| Class E | 1111 xxxx | 240.0.0.0 | 255.255.255.255 |

# CLASS A

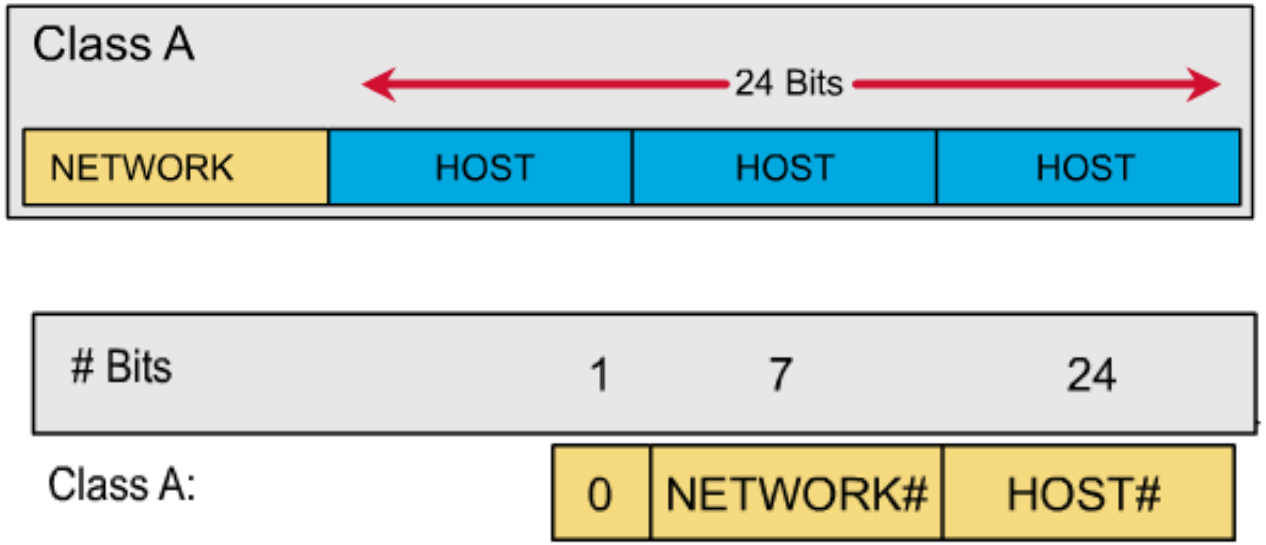
- **First bit** of a Class A address is **always 0**.
- The remaining **7 bits** in first octet identify the network ID → $2^7 - 1 = 127$ networks of Class A

→Possible **network address** from 1.0.0.0 to 127.0.0.0.

→However, 127.0.0.0 is used for **loopback network**

→**Can be used: 126 network address of Class A**

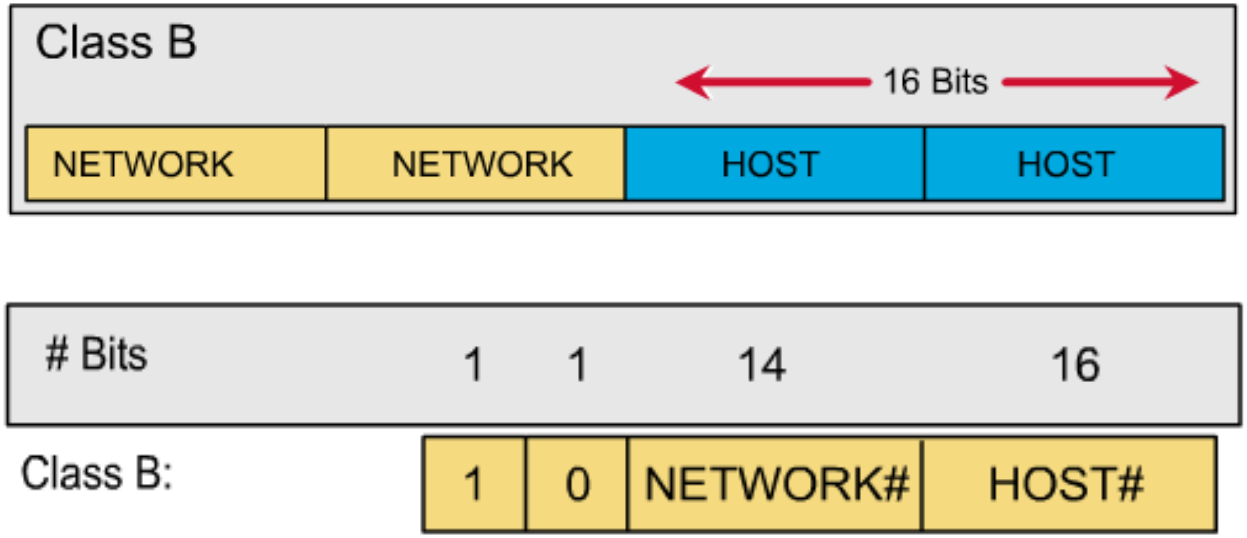
- Remaining three octets used for the host portion of the address.
- Each class A network have up to **$2^{24}$-2 = 16,777,214** possible IP addresses.




**First Octet: 1 – 126 → Class A**

# CLASS B

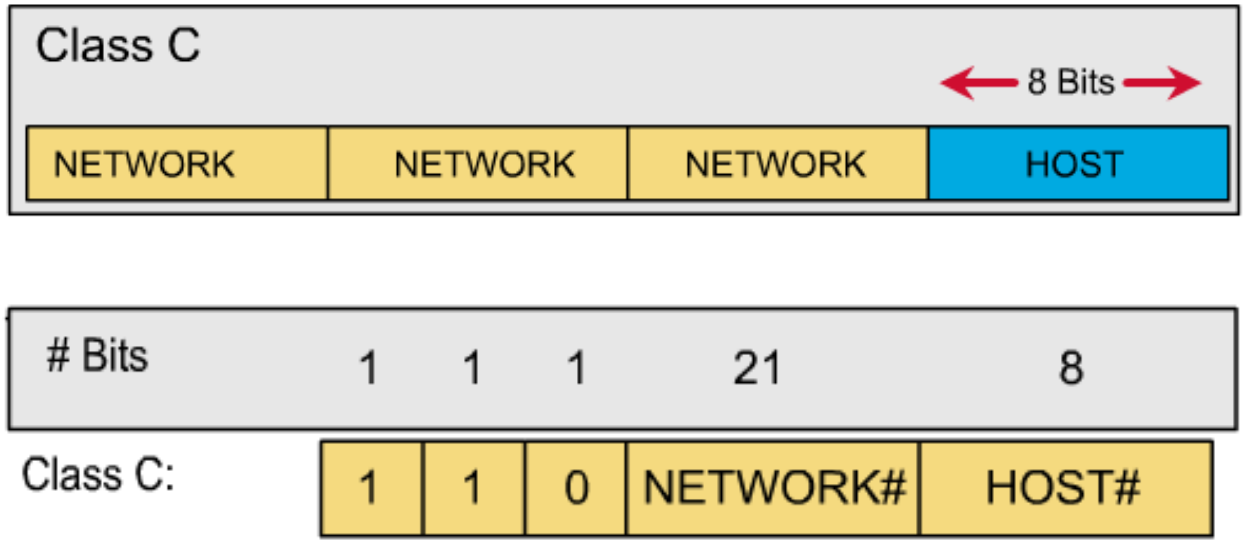
- **First 2 bits** of Class B address is **always 10**.
- The remaining **14 bits** in two first octets identify the network ID → $2^{14}$ **= 16,384** networks of Class B
- Possible **network address** from 128.0.0.0 to 191.255.0.0.
- Remaining two octets used for host portion of the address.
- Class B network have up to $2^{16} - 2$ **= 65,534** possible IP addresses.




**First Octet: 128 – 191 → Class B**

# CLASS C

- **First 3 bits** of Class C address is **always 110**.
- The remaining **21 bits** in three first octets identify the network ID → $2^{21}$ **= 2,097,152** networks of Class C
- Possible **network address** from 192.0.0.0 to 223.255.255.0.
- Remaining octet used for host portion of the address.
- Class C network have up to $2^8$ **– 2 = 254** possible IP addresses.

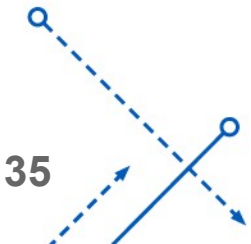



**First Octet: 192 – 223 → Class C**

# CLASS D

- IP addresses of Class D are used for Multicast Address
- Example:
  - **224.0.0.5**: used for OSPF protocol
  - **239.255.255.255**: used for RIPv2 protocol

**First Octet: 224 – 239 → Class D**

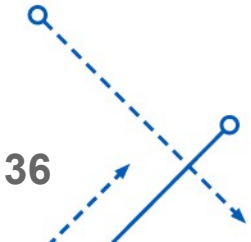


# SUMMARY

## NETWORK ID

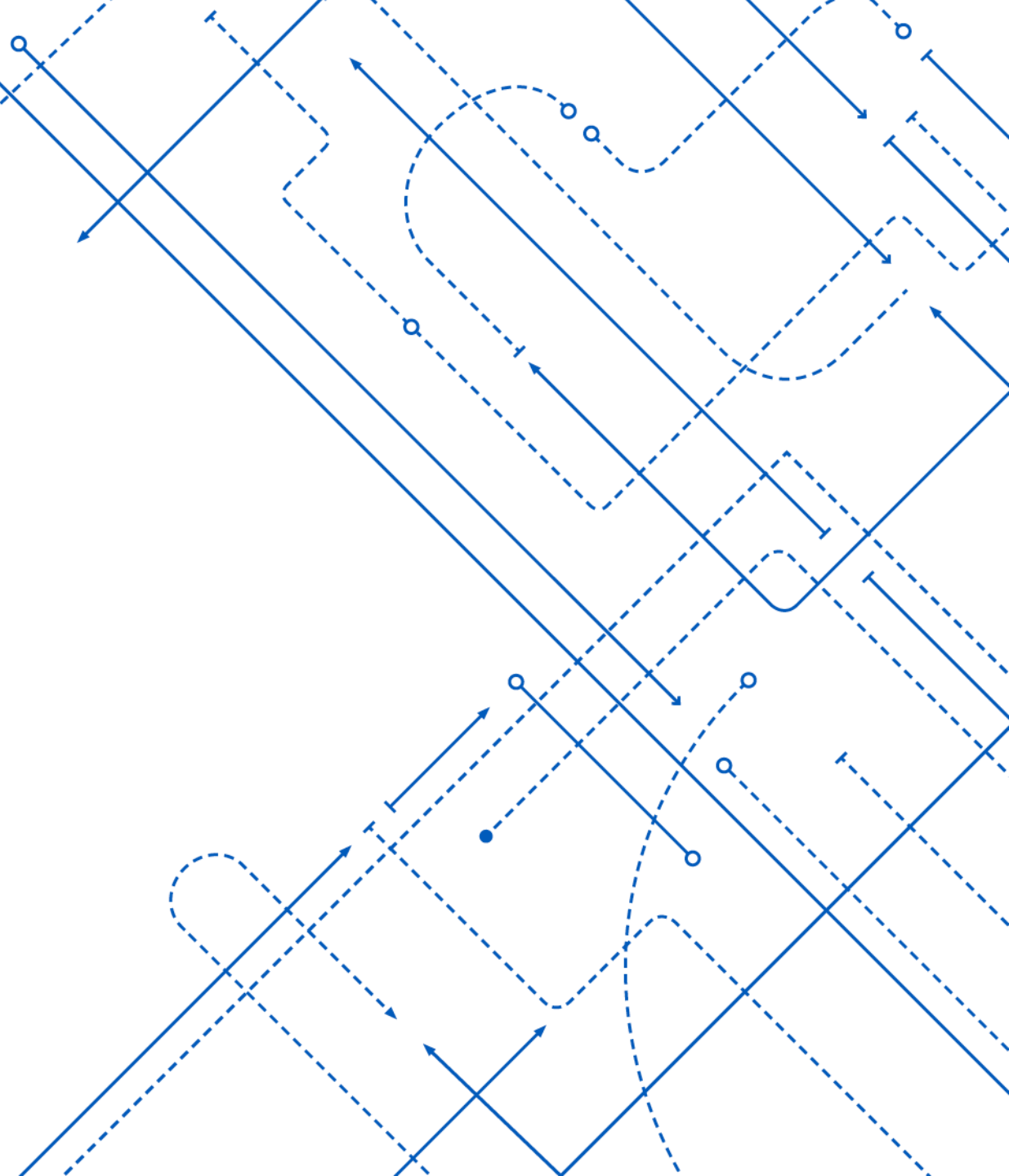
- 1.0.0.0 - 126.0.0.0 : Class A.
- 127.0.0.0 : Loopback network.
- 128.0.0.0 - 191.255.0.0 : Class B.
- 192.0.0.0 - 223.255.255.0 : Class C.
- 224.0.0.0 - 239.255.255.255: Class D, multicast.
- >= 240.0.0.0 : Class E, reserved.

| Class | High Order Bits (First Octet) | Start Address | End Address |
|---|---|---|---|
| Class A | 0xxx xxxx | 1.0.0.0 | 127.255.255.255 |
| Class B | 10xx xxxx | 128.0.0.0 | 191.255.255.255 |
| Class C | 110x xxxx | 192.0.0.0 | 223.255.255.255 |
| Class D | 1110 xxxx | 224.0.0.0 | 239.255.255.255 |
| Class E | 1111 xxxx | 240.0.0.0 | 255.255.255.255 |



The University of Danang
University of Science and Technology

# 3. IPv6

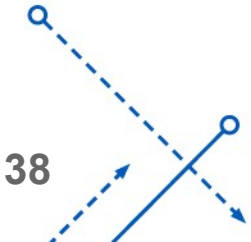


Faculty of Information Technology
PhD. Le Tran Duc

# IPv6 MOTIVATION

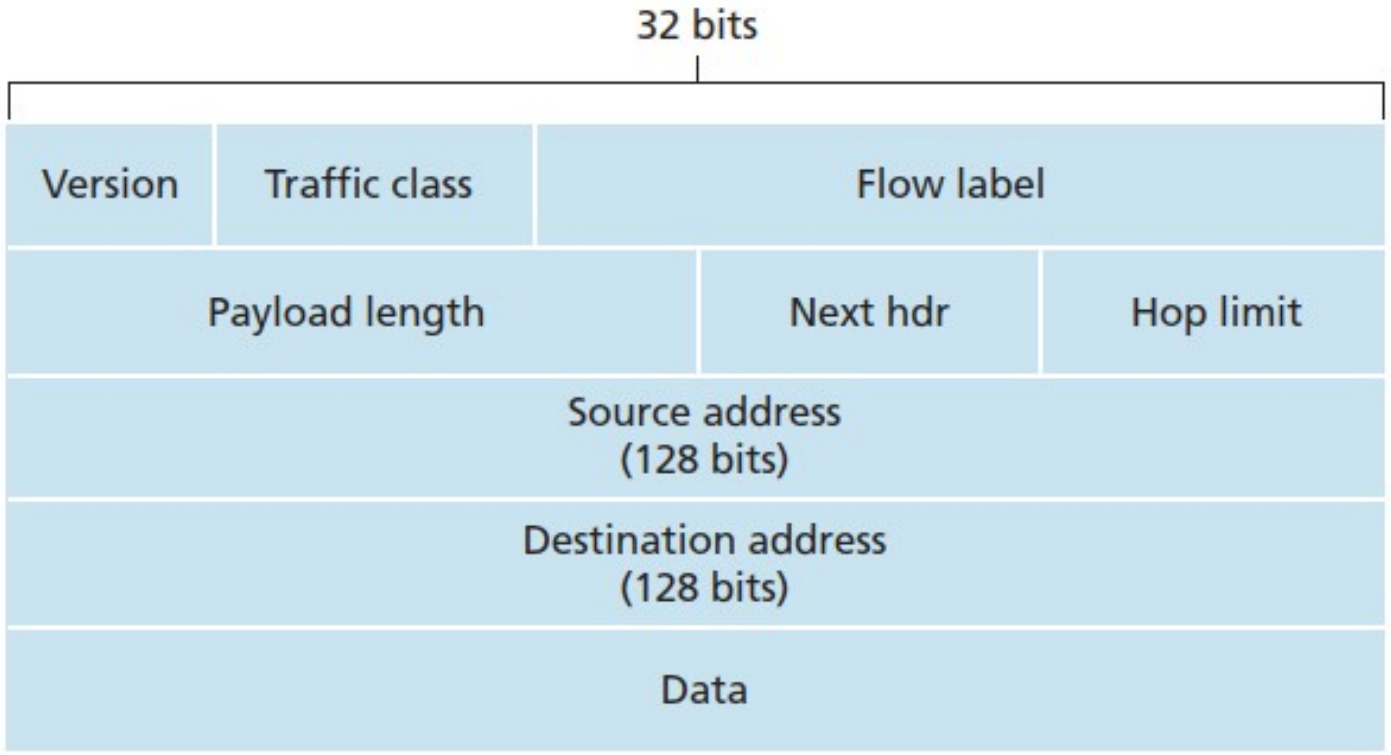
- ***Initial motivation****:* 32-bit address space soon to be completely allocated.
- Additional motivation:
  - Header format helps speed processing/forwarding
  - Header changes to facilitate QoS

- ***IPv6 packet format:***
  - Fixed-length 40 byte header (no length variability in header)
  - **No fragmentation** allowed

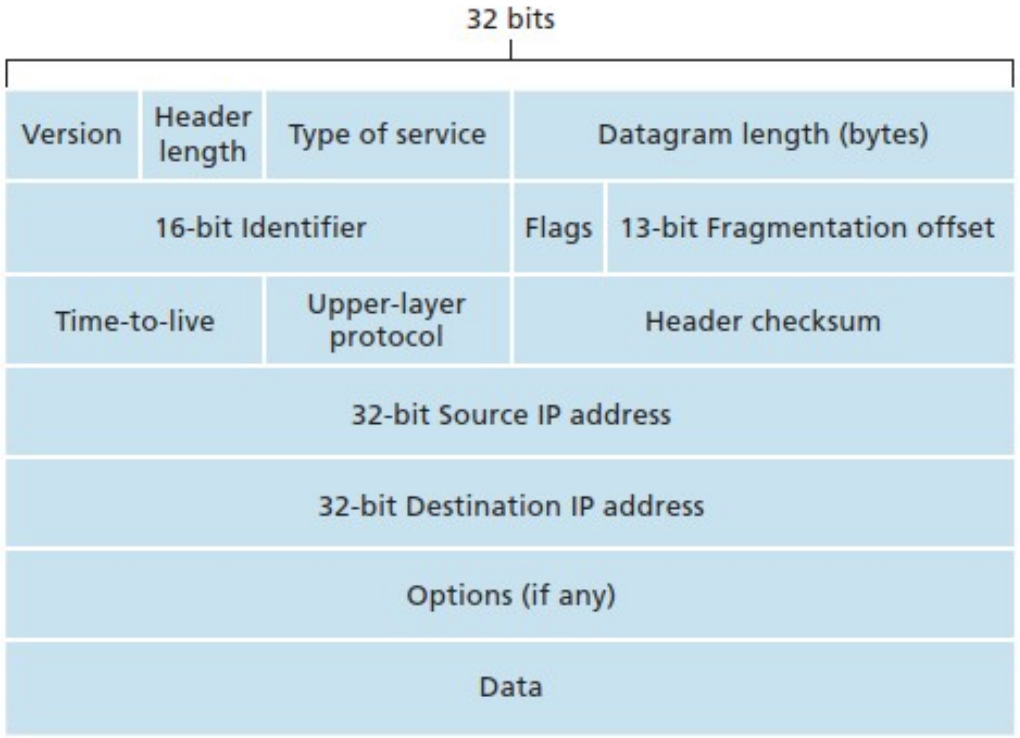


# IPv6 PACKET FORMAT

- ***Traffic Class*** *(Priority):* identify priority among packets in flow
- ***Flow Label****:* identify packets in same "flow."
  (concept of"flow" not well defined).
- ***Next header****:* identify upper layer protocol for data

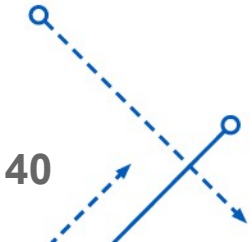



## IPv4 Packet



# OTHER CHANGES FROM IPv4

- ***Fragmentation/Reassembly:*** IPv6 does not allow for fragmentation and reassembly at intermediate routers; these operations can be performed only by the source and destination.
- ***Checksum***: removed entirely to reduce processing time at each hop
- ***Options****:* allowed, but outside of header, indicated by "Next Header" field
- ***ICMPv6****:* new version of ICMP
  - Additional message types, e.g. "**Packet Too Big**"→ send back to sender to inform that the IPv6 packet received by a router is too large to be forwarded
  - Multicast group management functions



# IPv6 REPRESENTATION

- IPv6 is 128bit while IPv4 only has 32 bits. The 128 bits of an IPv6 address are represented in **8 groups of 16 bits each**.
- IPv6 is written in HEX (Each group is written as **four hexadecimal digits**) while IPv4 is written in decimal
- The groups are separated by colons (**:**).

An IPv6 address (in hexadecimal)

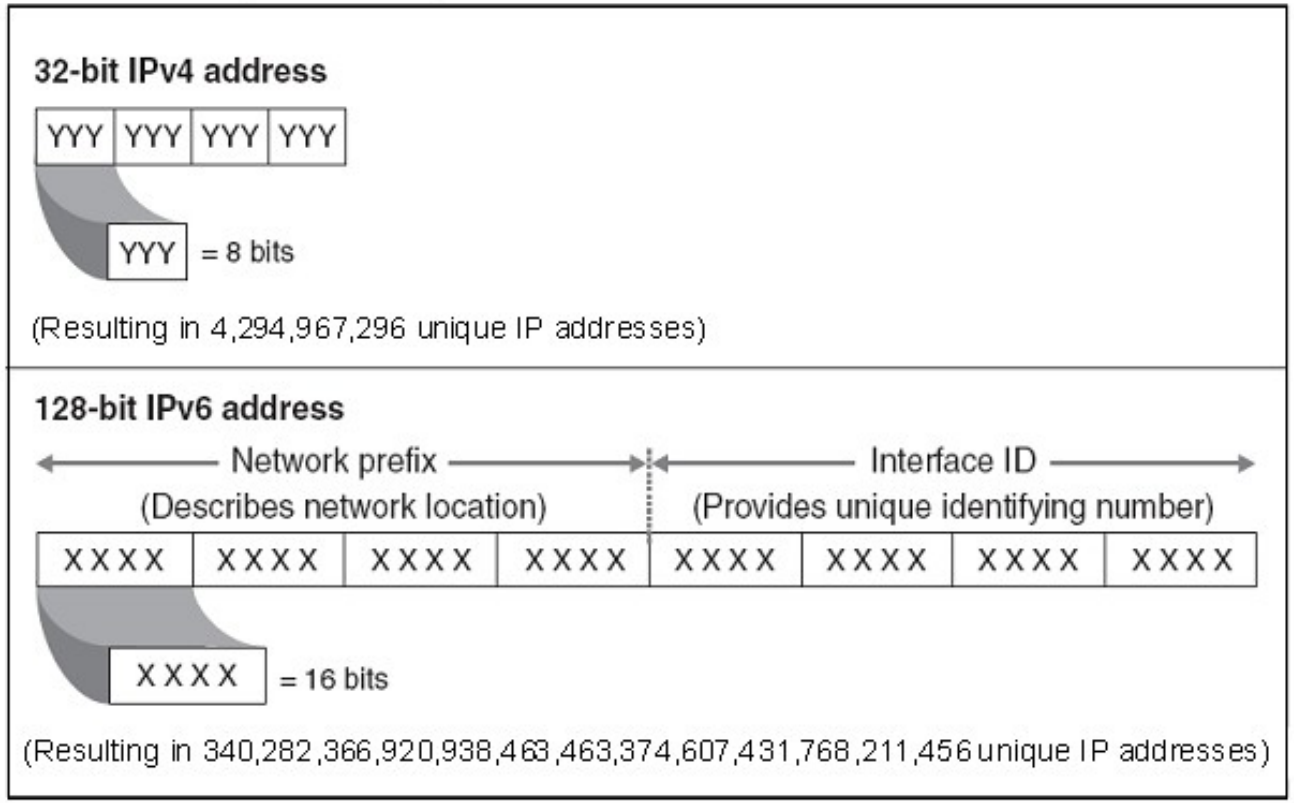
**2001:0DB8:AC10:FE01:0000:0000:0000:0000**

**2001:0DB8:AC10:FE01::** Zeroes can be omitted

0010000000000001:0000110110111000:1010110000010000:1111111000000001:

0000000000000000:0000000000000000:0000000000000000:0000000000000000

Comparison of IPv6 and IPv4 Address Scheme




Source: GAO.

# IPv6 REPRESENTATION EXAMPLE

- IPv^6: **2001:0db8:85a3:0000:0000:8a2e:0370:7334**
- Some IPv6 shortening rules:
  - Omit Leading 0s

fe80: **000**0: **000**0: **000**0:a299:9bff:fe18:50d1

fe80:**0**:**0**:**0**:a299:9bff:fe18:50d1

2001: **0**db8: 1111:**000**a:**00**b0:**000**0:9000:**0**200

2001:db8:1111:a:b0:0:9000:200

  - Omit All-0s Hextets (only 1 time/address)

ff02:**0000**:**0000**:**0000**:**0000**:**0000**:**0000**:0001

ff02::0001

2001:0db8:**0000**:**0000**:abcd:0000:0000:1234

2001:0db8::abcd:0000:0000:1234

# IPv6 REPRESENTATION EXAMPLE

Find possible IPv6 addresses: **2001::abcd::1234**

2001:**0000:0000:0000:0000**:abcd:**0000**:1234

2001:**0000:0000:0000**:abcd:**0000:0000**:1234

2001:**0000:0000**:abcd:**0000:0000:0000**:1234

2001:**0000**:abcd:**0000:0000:0000:0000**:1234

Combining rule 1 and rule 2: 2001:**0**db8:1111:**000**a:**00**b0:**000**0:9000:**0**200

2001:db8:1111:a:b0::9000:200

The University of Danang
University of Science and Technology

# 4. NAT (NETWORK ADDRESS TRANSLATION)

Faculty of Information Technology
PhD. Le Tran Duc

# NETWORK ADDRESS TRANSLATION (NAT)

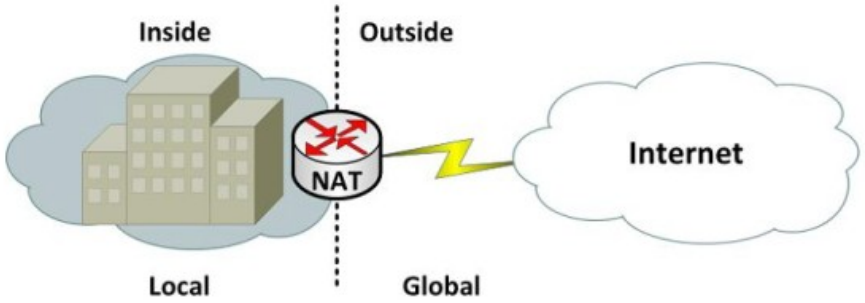
***Motivation**:* local network uses just one IP address as far as outside world is concerned:

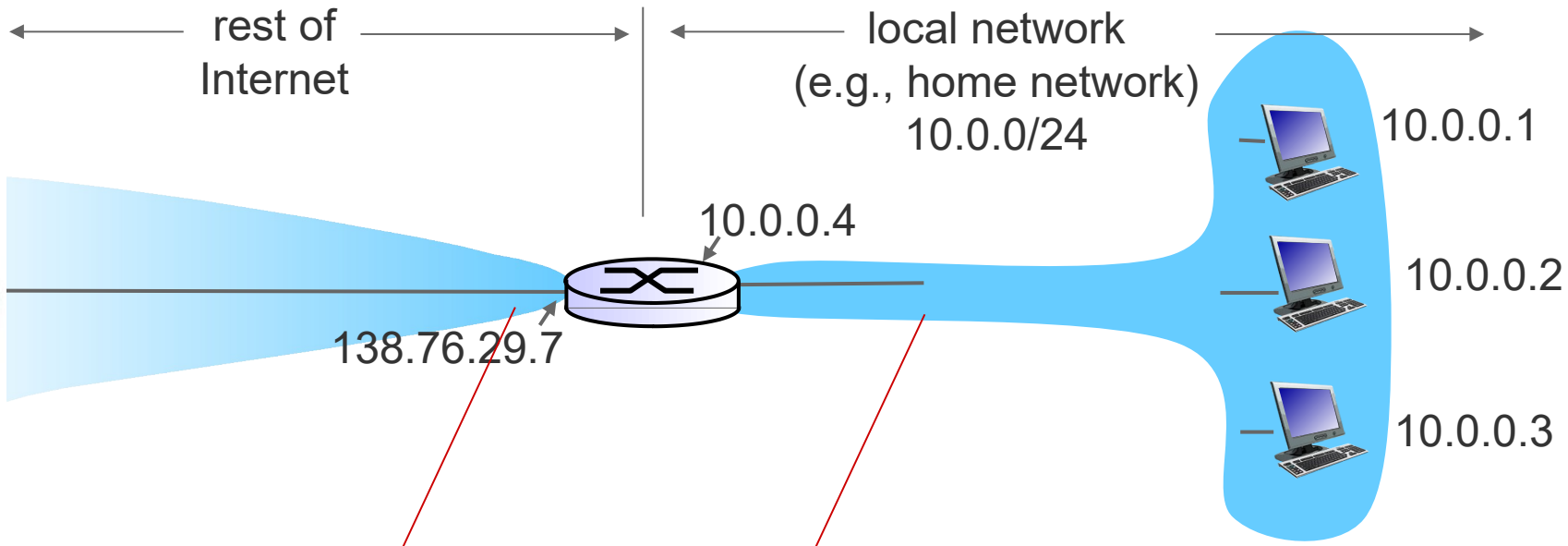
- Range of addresses not needed from ISP:  just one IP address for all devices
- Can change addresses of devices in local network without notifying outside world
- Can change ISP without changing addresses of devices in local network
- Devices inside local network not explicitly addressable, visible by outside world (a security plus)

***Implementation***: NAT router must:

- ***Outgoing datagrams**: replace* (source IP address, port #) of every outgoing datagram to (NAT IP address, new port #)
  . . . remote clients/servers will respond using (NAT IP address, new port #) as destination addr
- ***Remember (in NAT translation table)*** every (source IP address, port #)  to (NAT IP address, new port #) translation pair
- ***Incoming datagrams**: replace* (NAT IP address, new port #) in dest fields of every incoming datagram with corresponding (source IP address, port #) stored in NAT table

# NETWORK ADDRESS TRANSLATION (NAT)

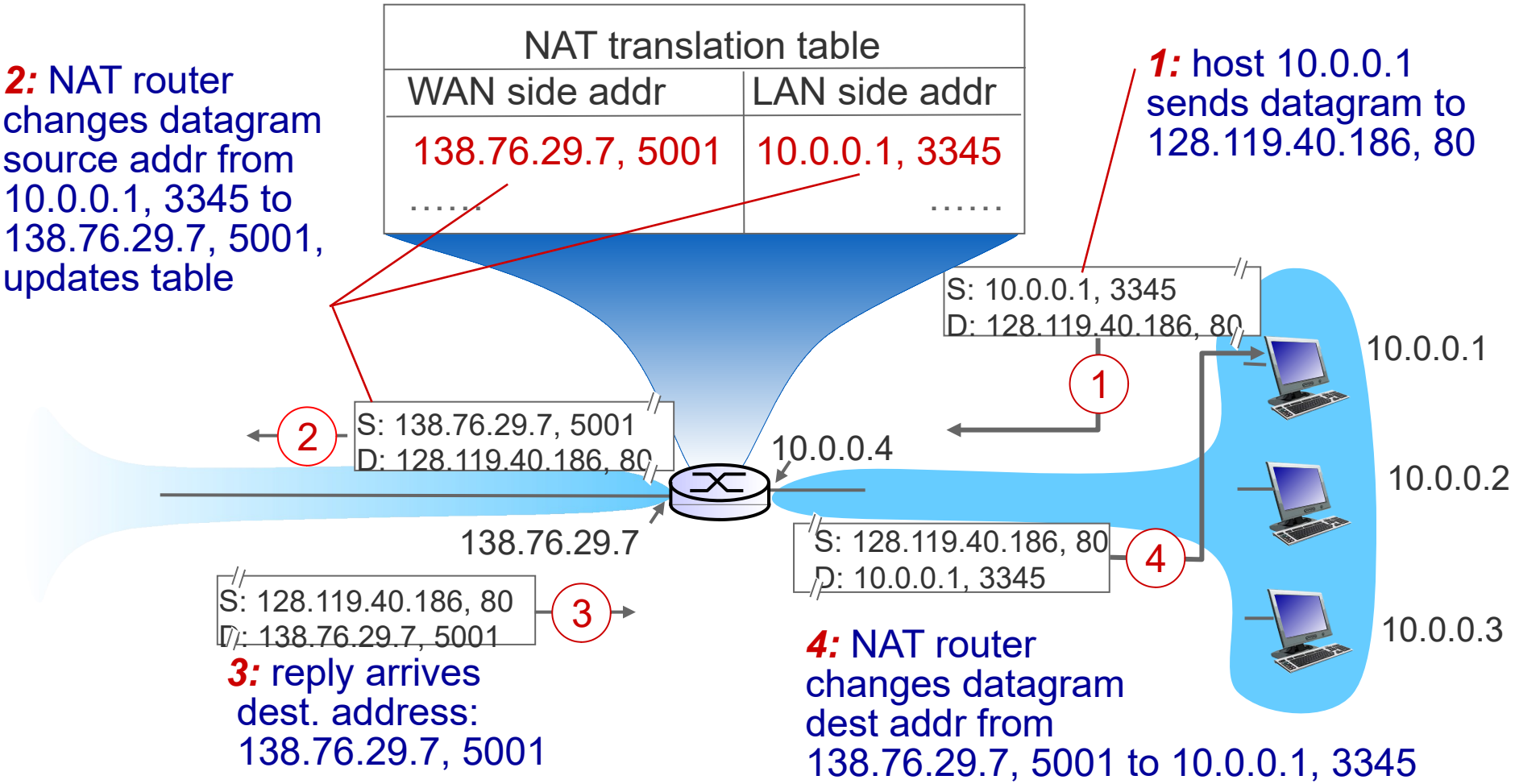


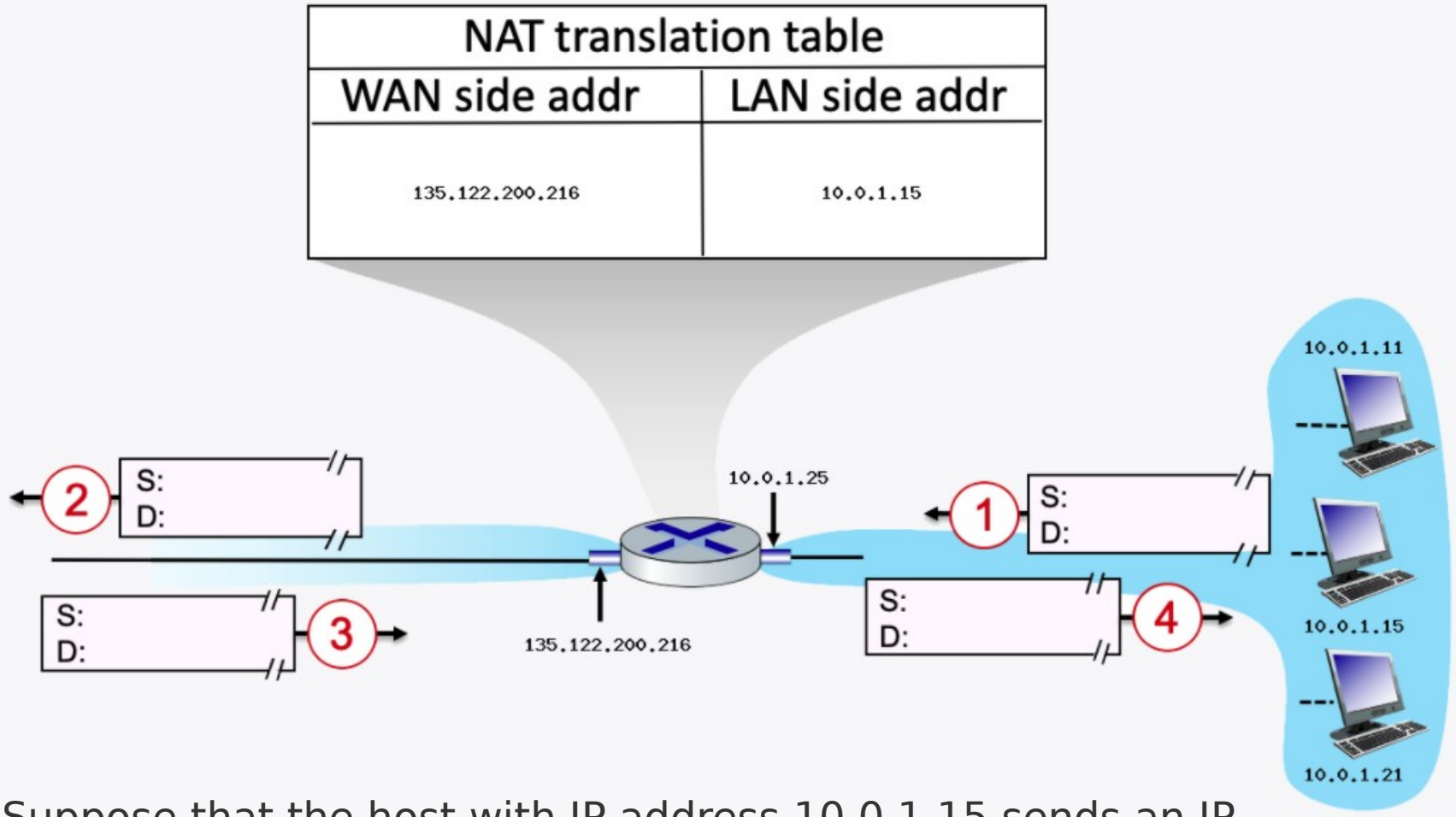

*all* datagrams *leaving* local network have *same* single source NAT IP address: 138.76.29.7,different source port numbers

datagrams with source or destination in this network have 10.0.0/24 address for source, destination (as usual)

# NETWORK ADDRESS TRANSLATION (NAT)



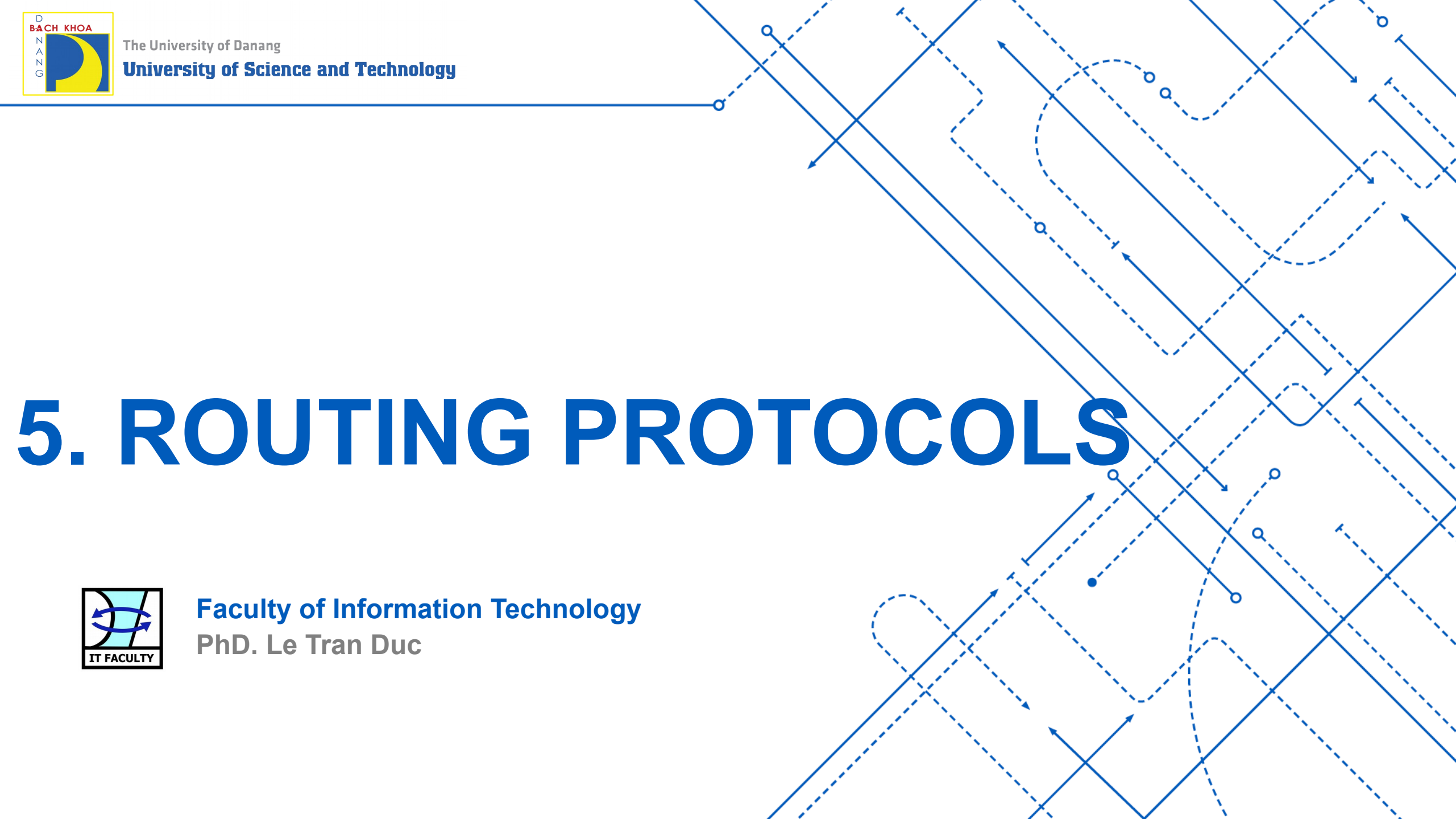
# NETWORK ADDRESS TRANSLATION (NAT)




Consider the scenario below in which three hosts, with private IP addresses 10.0.1.11, 10.0.1.15, 10.0.1.21 are in a local network behind a NAT'd router that sits between these three hosts and the larger Internet. IP datagrams being sent from, or destined to, these three hosts must pass through this NAT router. The router's interface on the LAN side has IP address 10.0.1.25, while the router's address on the Internet side has IP address 135.122.200.216

Suppose that the host with IP address 10.0.1.15 sends an IP datagram destined to host 128.119.167.186. The source port is 3421, and the destination port is 80.

The University of Danang
University of Science and Technology



# 5. ROUTING PROTOCOLS

Faculty of Information Technology
PhD. Le Tran Duc

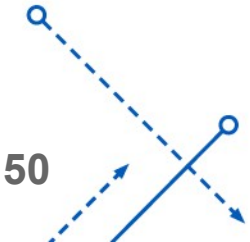
# RECALL NETWORK-LAYER FUNCTIONS

***Recall**: two network-layer functions:*

- ***Forwarding**:* move packets from router's input to appropriate router output — ***data plane***
- ***Routing**:* determine route taken by packets from source to destination — ***control plane***

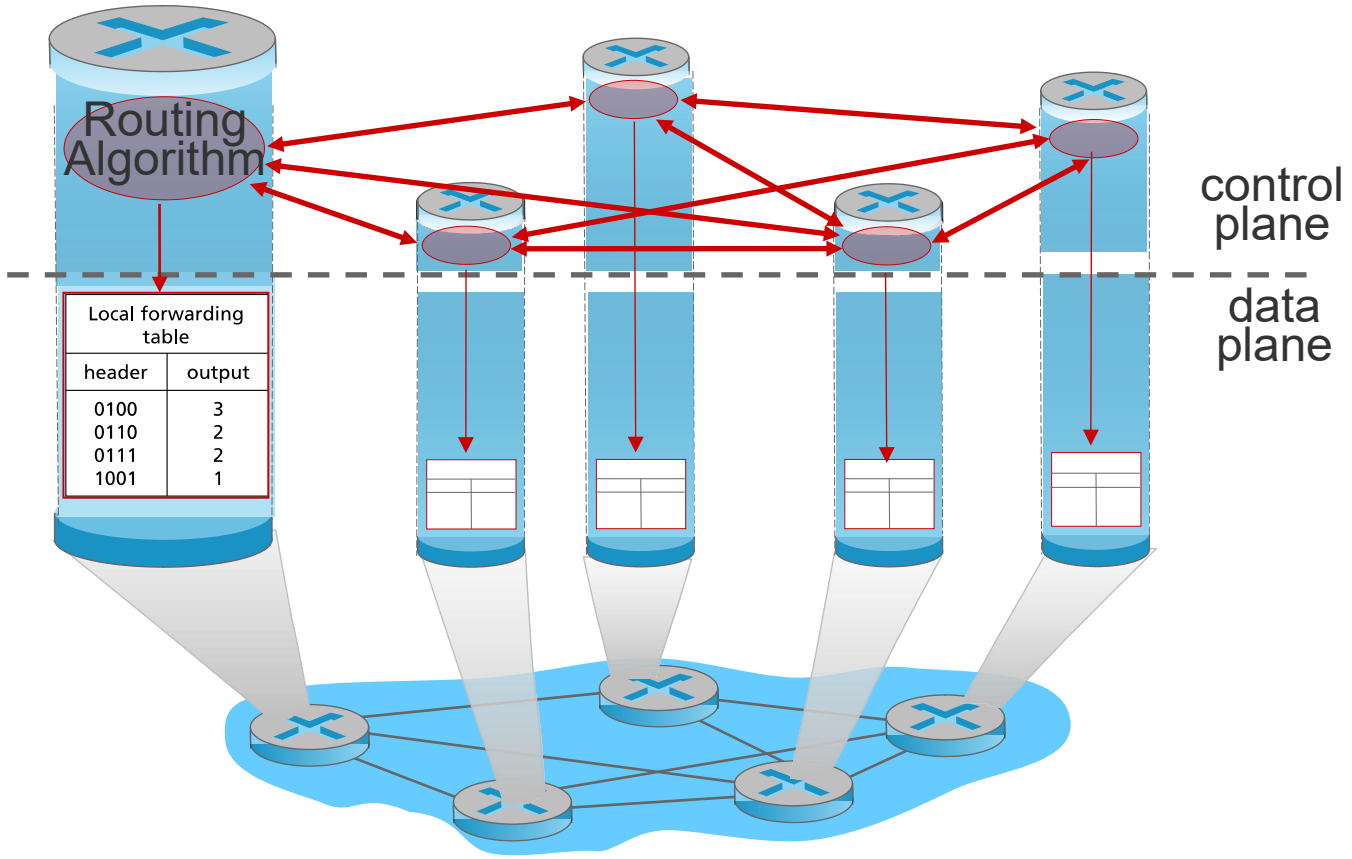
*Two approaches to structuring network control plane:*

- Per-router control (traditional)
- Logically centralized control (software defined networking)



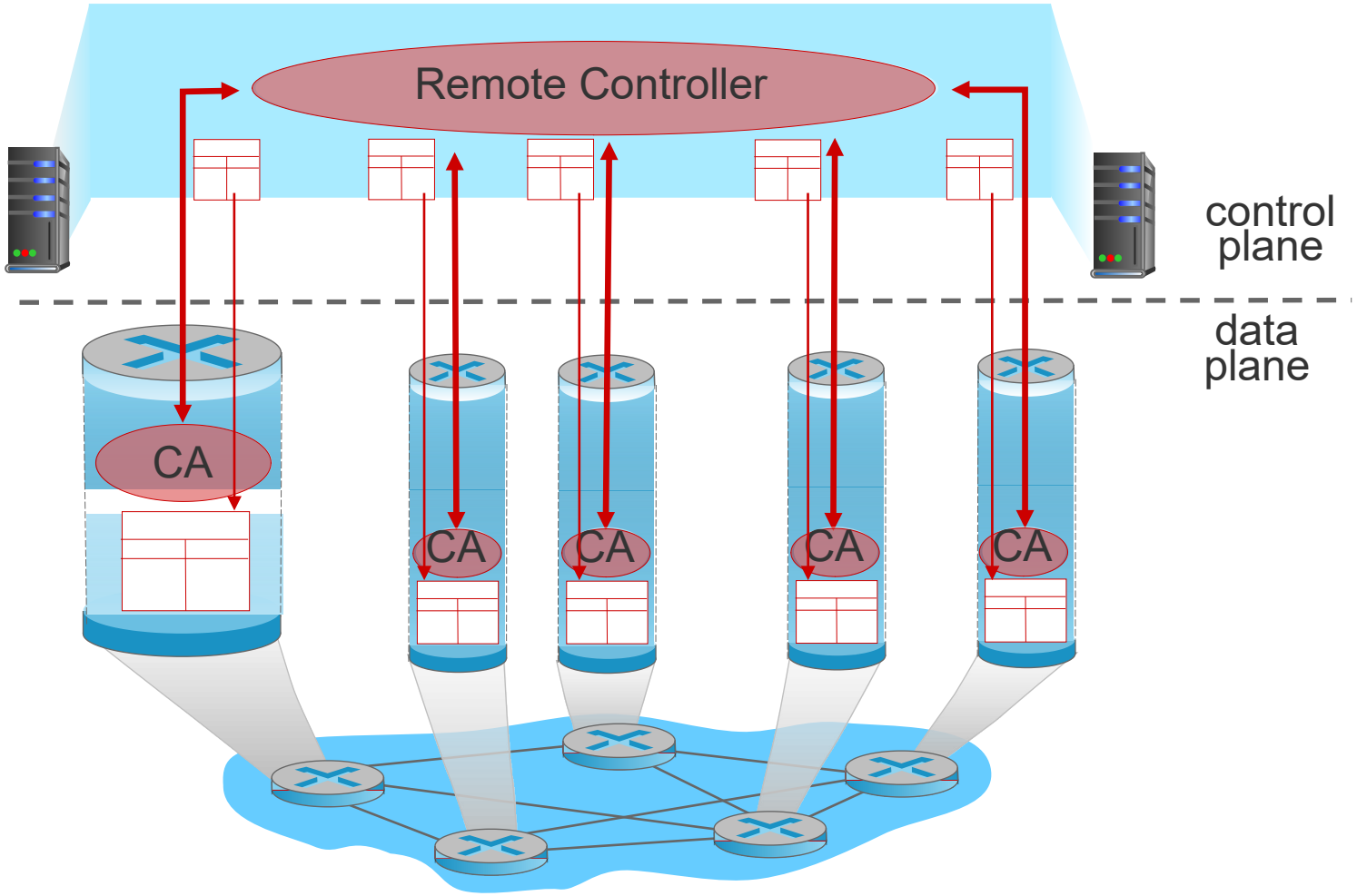
# PER-ROUTER CONTROL PLANE

Individual routing algorithm components ***in each and every router*** interact with each other in control plane to compute forwarding tables



# LOGICALLY CENTRALIZED CONTROL PLANE

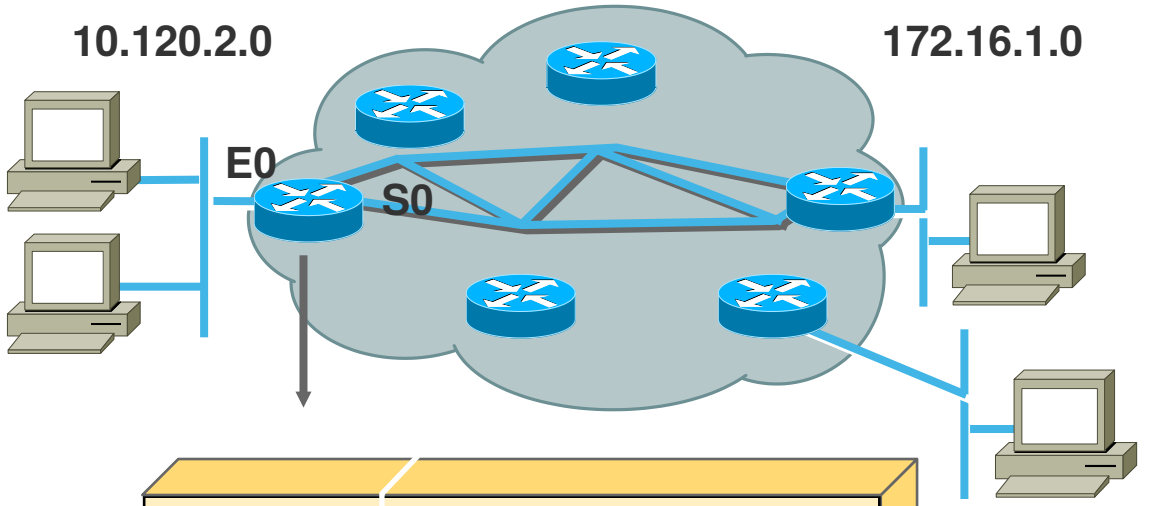
A distinct (typically remote) controller interacts with local control agents (CAs) in routers to compute forwarding tables



# ROUTING PROTOCOLS

**Routing protocols** are used between routers to determine good paths and maintain routing tables.

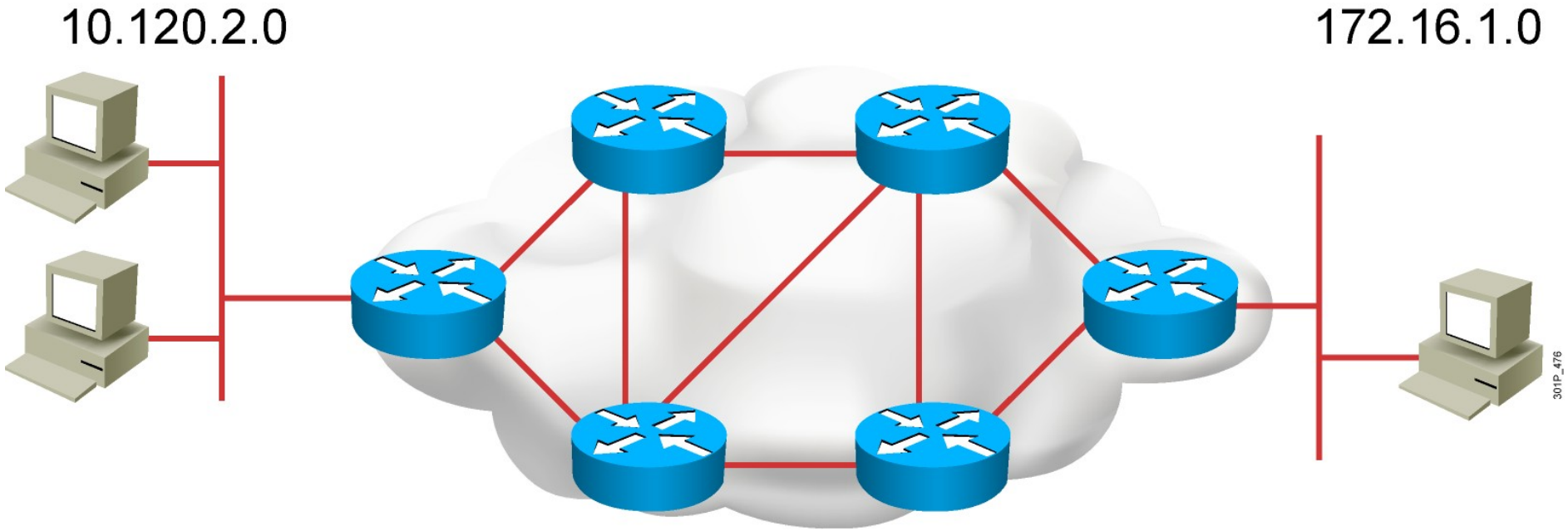
- **Path**: sequence of routers, packets will traverse in going from given initial source host to given final destination host
- "**good**": least "cost", "fastest", "least congested"
- Once the path is determined a router can route a **routed** protocol.




| Network Protocol | Destination Network | Exit Interface |
|---|---|---|
| Connected | 10.120.2.0 | E0 |
| RIP | 172.16.2.0 | S0 |
| EIGRP | 172.17.3.0 | S1 |

**Routed Protocol: IP**
**Routing protocol: RIP, EIGRP…**

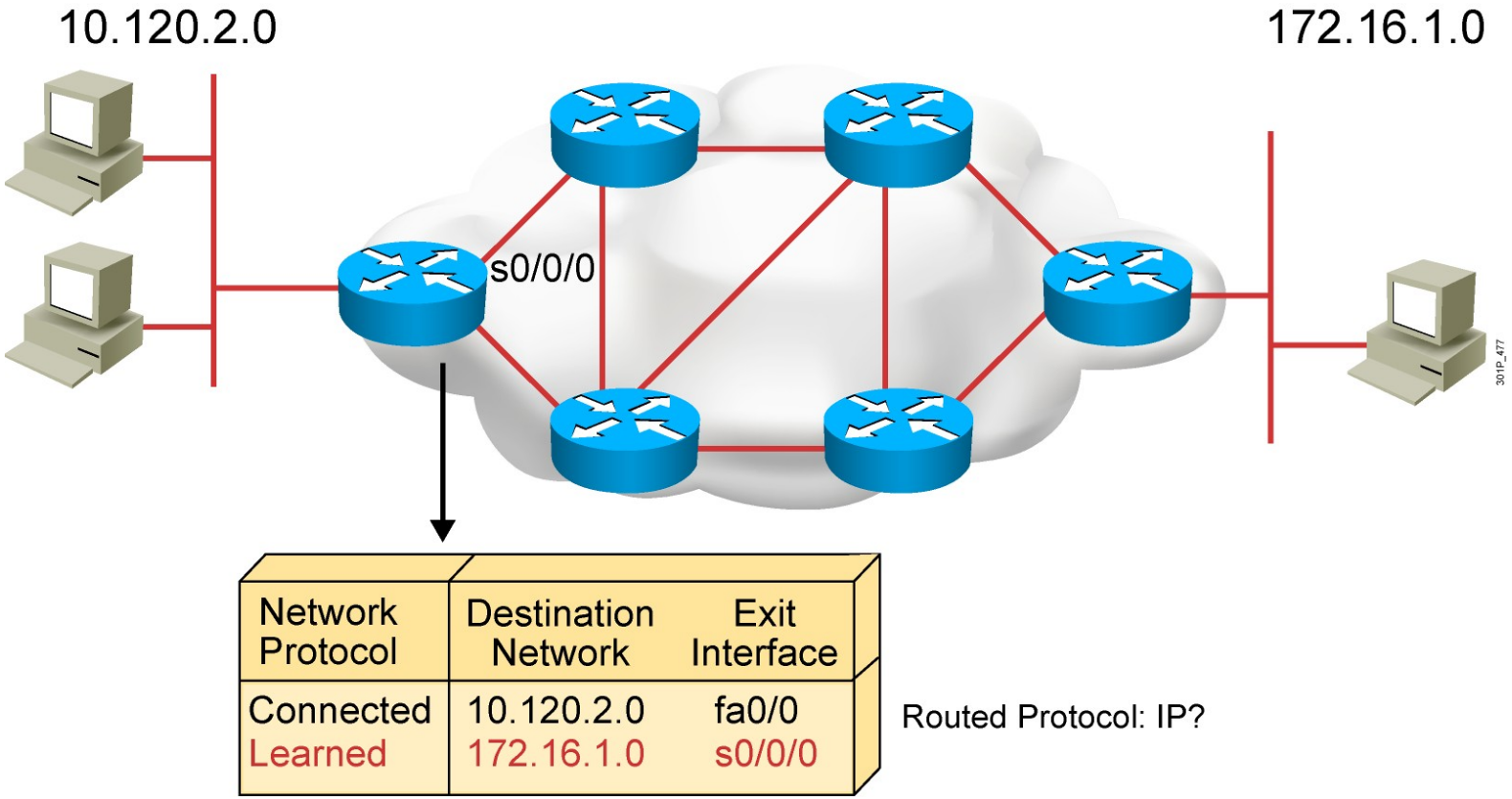
# ROUTER OPERATIONS




**A router needs to do the following:**

- Know the destination address.
- Identify the sources from which the router can learn.
- Discover possible routes to the intended destination.
- Select the best route.
- Maintain and verify routing information.

# ROUTER OPERATIONS




- **Routers must learn destinations that are not directly connected.**

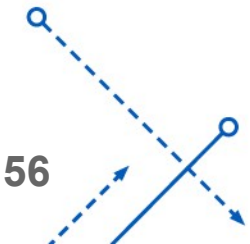
# ROUTING ALGORITHM CLASSIFICATION

***Q: global or decentralized information?***

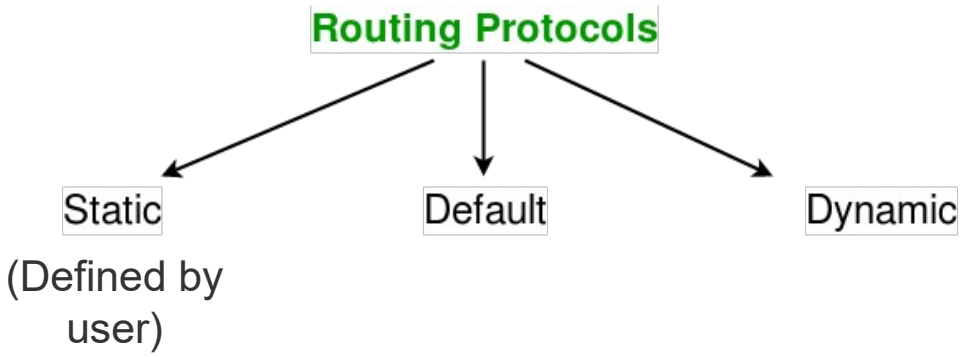
- ***Global***:
  - All routers have complete topology, link cost info
  - "**Link state**" algorithms
- ***Decentralized***:
  - Router knows physically-connected neighbors, link costs to neighbors
  - Iterative process of computation, exchange of info with neighbors
  - "**Distance vector**" algorithms

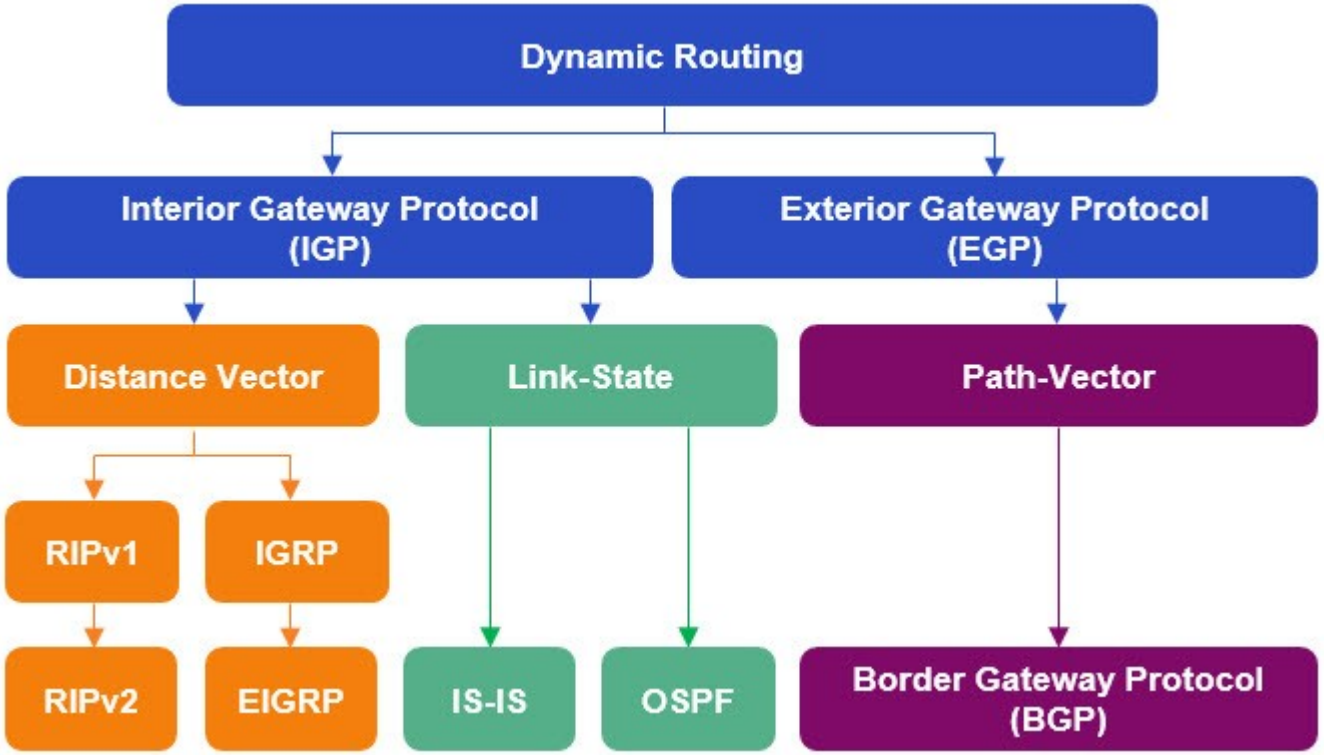
***Q: static or dynamic?***

- ***Static***:
  - Uses a route that a network administrator enters into the router manually
  - Routes change slowly over time
- ***Dynamic***:
  - Uses a route that a network routing protocol adjusts automatically for topology or traffic changes
  - Routes change more quickly
    - periodic update
    - in response to link cost changes



# ROUTING ALGORITHM CLASSIFICATION



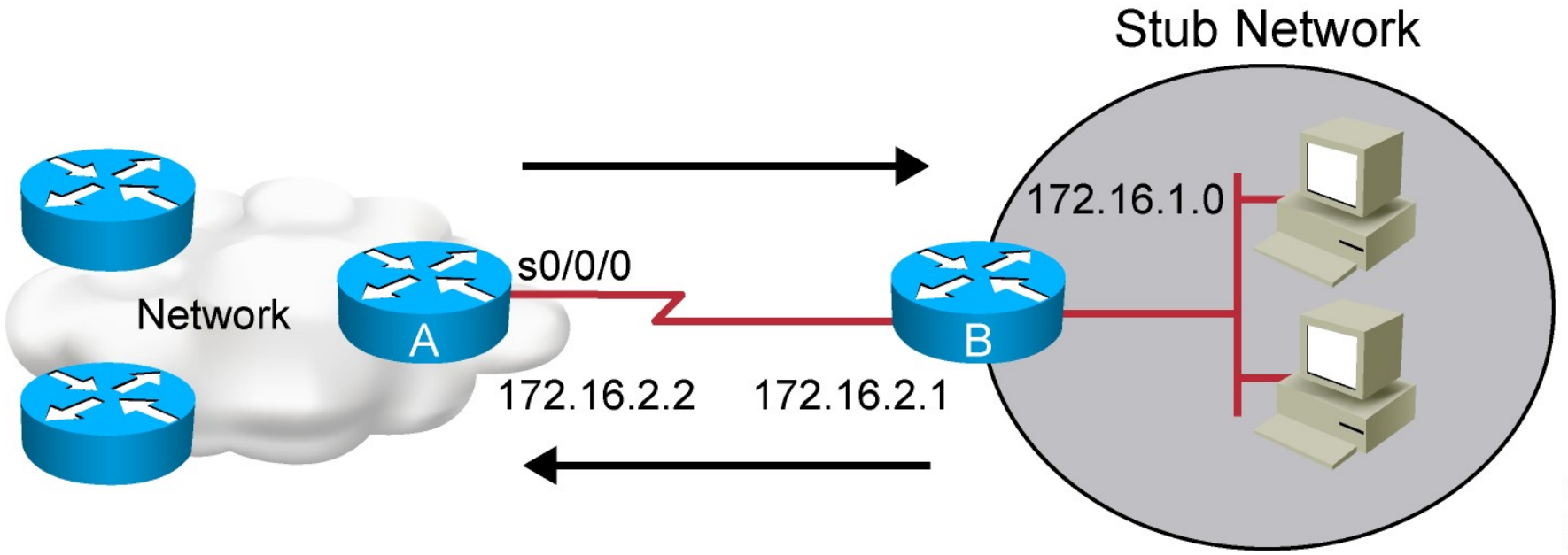




| | | | | |
|---|---|---|---|---|
| **Classful** | **RIP** | IGRP | | EGP |
| **Classless** | **RIPv2** | **EIGRP** | **OSPFv2** IS-IS | BGPv4 |
| **IPv6** | RIPng | EIGRP for IPv6 | OSPFv3 | IS-IS for IPv6 | BGPv4 for IPv6 |

# CRITERIA USED TO COMPARE ROUTING PROTOCOLS

- **Time to convergence**
  - Time to convergence defines how quickly the routers in the network topology share routing information and reach a state of consistent knowledge.
  - The faster the convergence, the more preferable the protocol.
- **Scalability**
  - Scalability defines how large a network can become based on the routing protocol that is deployed.
  - The larger the network is, the more scalable the routing protocol needs to be.
- **Resource usage**
  - Resource usage includes the requirements of a routing protocol such as memory space, CPU utilization, and link bandwidth utilization.
  - Higher resource requirements necessitate more powerful hardware to support the routing protocol operation
- **Classless (Use of VLSM) or Classful**
  - Classless routing protocols include the subnet mask in the updates.
  - This feature supports the use of Variable Length Subnet Masking (VLSM) and better route summarization.
- **Implementation & maintenance**
  - Implementation and maintenance describes the level of knowledge that is required for a network administrator to implement and maintain the network based on the routing protocol deployed.

# STATIC ROUTES

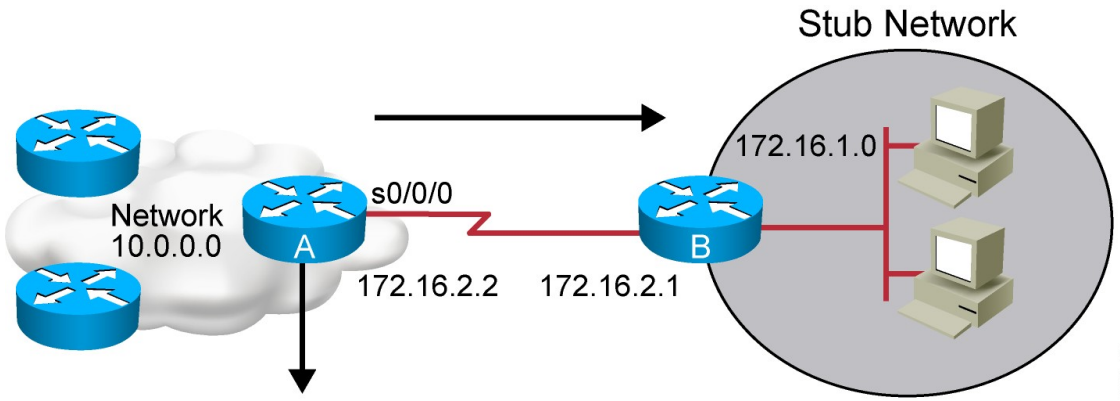



Configure unidirectional static routes to and from a stub network to allow communications to occur.

# STATIC ROUTE CONFIGURATION

```
RouterX(config)# ip route network  [mask]
{address | interface}[distance] [permanent]
```

- Defines a path to an IP destination network or subnet or host
- Address = IP address of the next hop router
- Interface = outbound interface of the local router



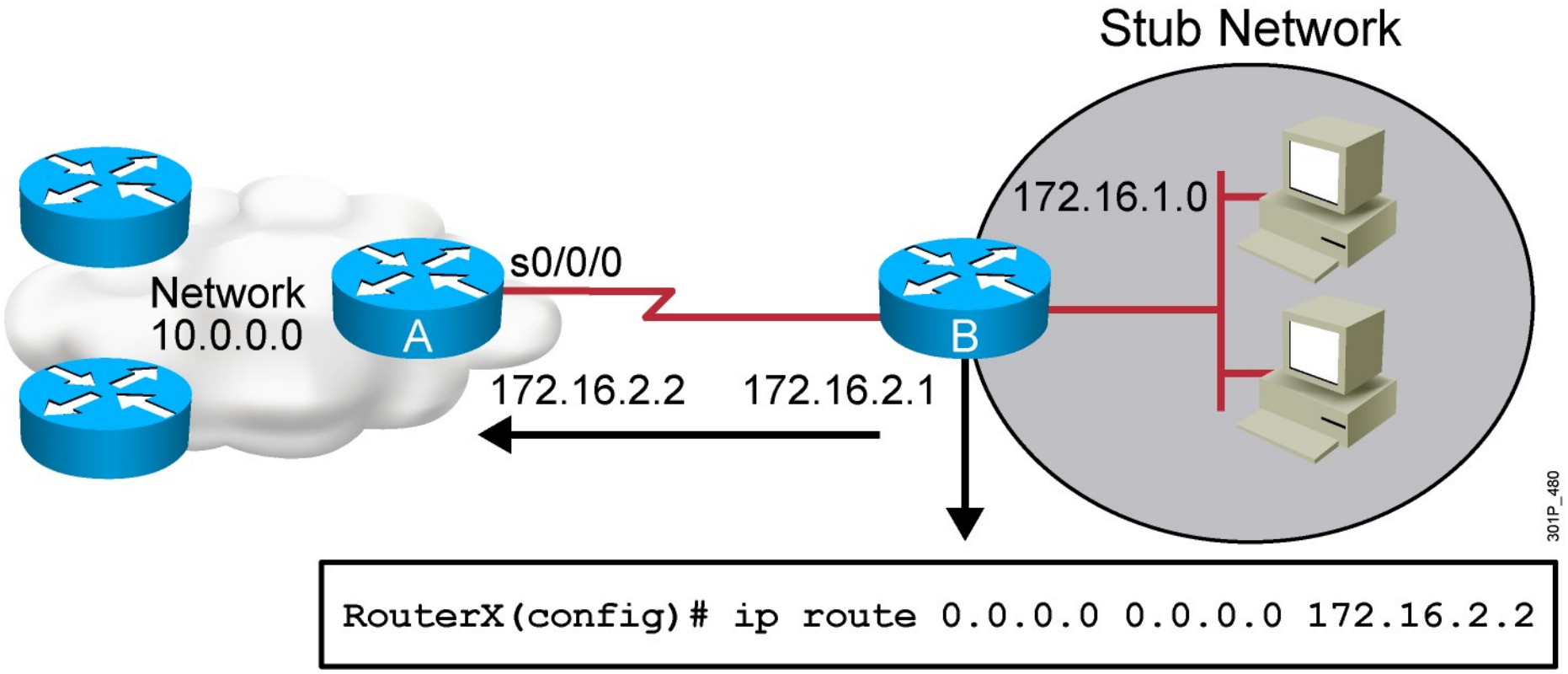

```
RouterX(config)# ip route 172.16.1.0 255.255.255.0 172.16.2.1
```

or

```
Router(config)#ip route 172.16.1.0 255.255.255.0 s0/0/0
```

- This is a unidirectional route. You must have a route configured in the opposite direction.

# DEFAULT ROUTES




```
RouterX(config)# ip route 0.0.0.0 0.0.0.0 172.16.2.2
```

- Default routes are usually applied to **stub networks**
- This route allows the stub network to reach all known networks beyond Router A.

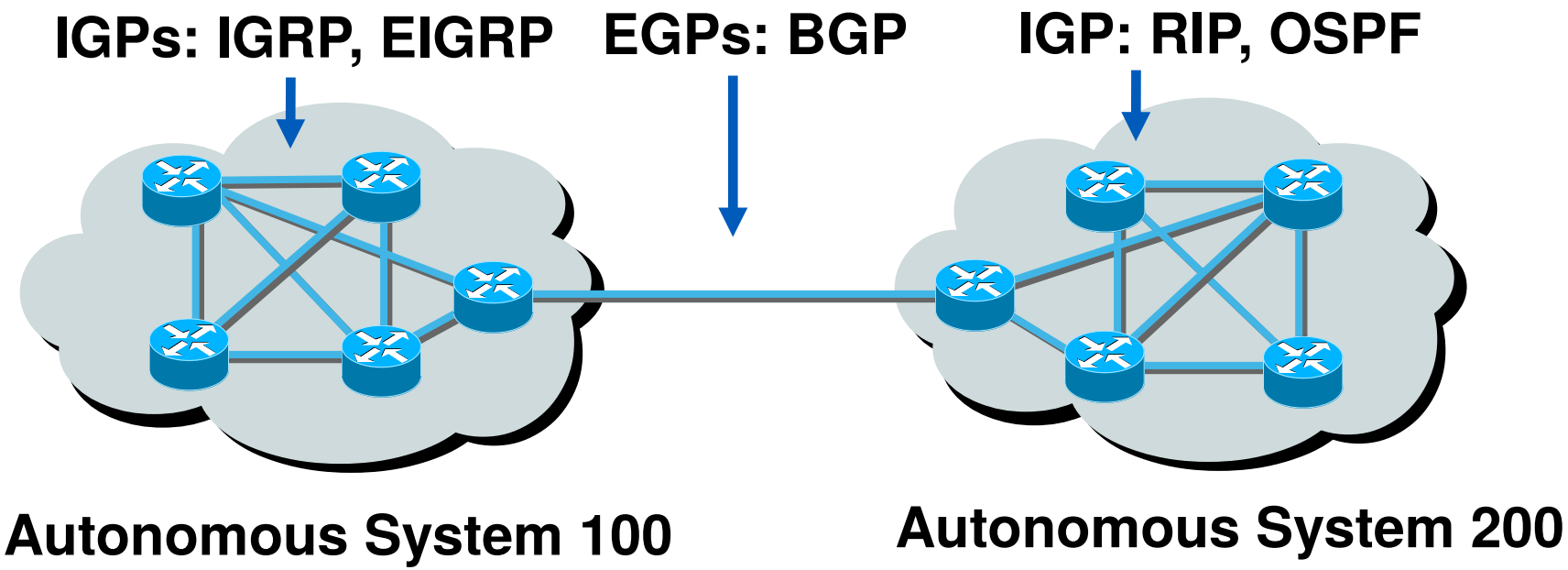
# VERIFYING THE STATIC ROUTE CONFIGURATION

```
RouterX# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route

Gateway of last resort is 0.0.0.0 to network 0.0.0.0

     10.0.0.0/8 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, Serial0/0/0
S*   0.0.0.0/0 is directly connected, Serial0
```

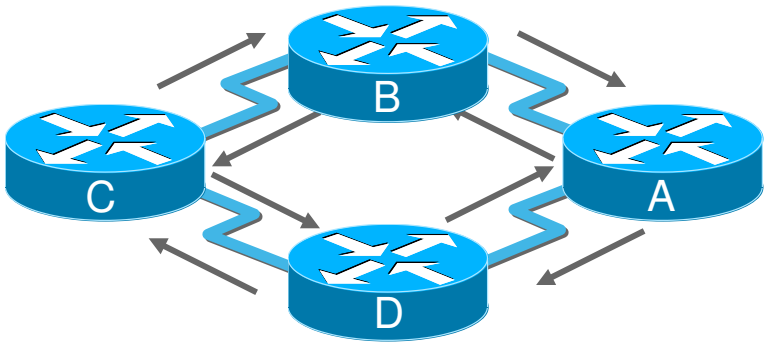
# AUTONOMOUS SYSTEMS: INTERIOR OR EXTERIOR ROUTING PROTOCOLS



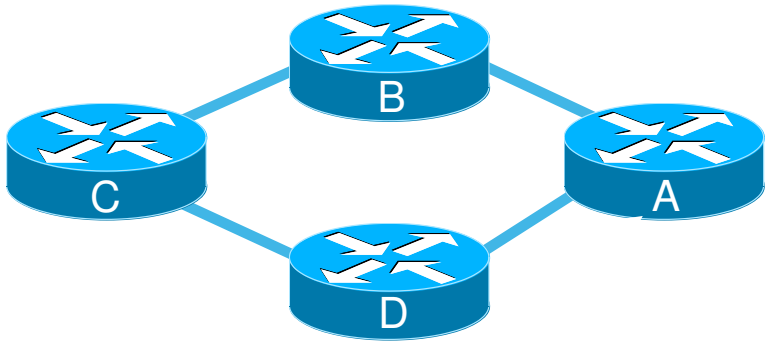

- An autonomous system is a collection of networks under a common administrative domain/protocol
- IGPs operate within an autonomous system
- EGPs connect different autonomous systems

# DISTANCE VECTOR vs LINK STATE

Distance Vector (RIP)



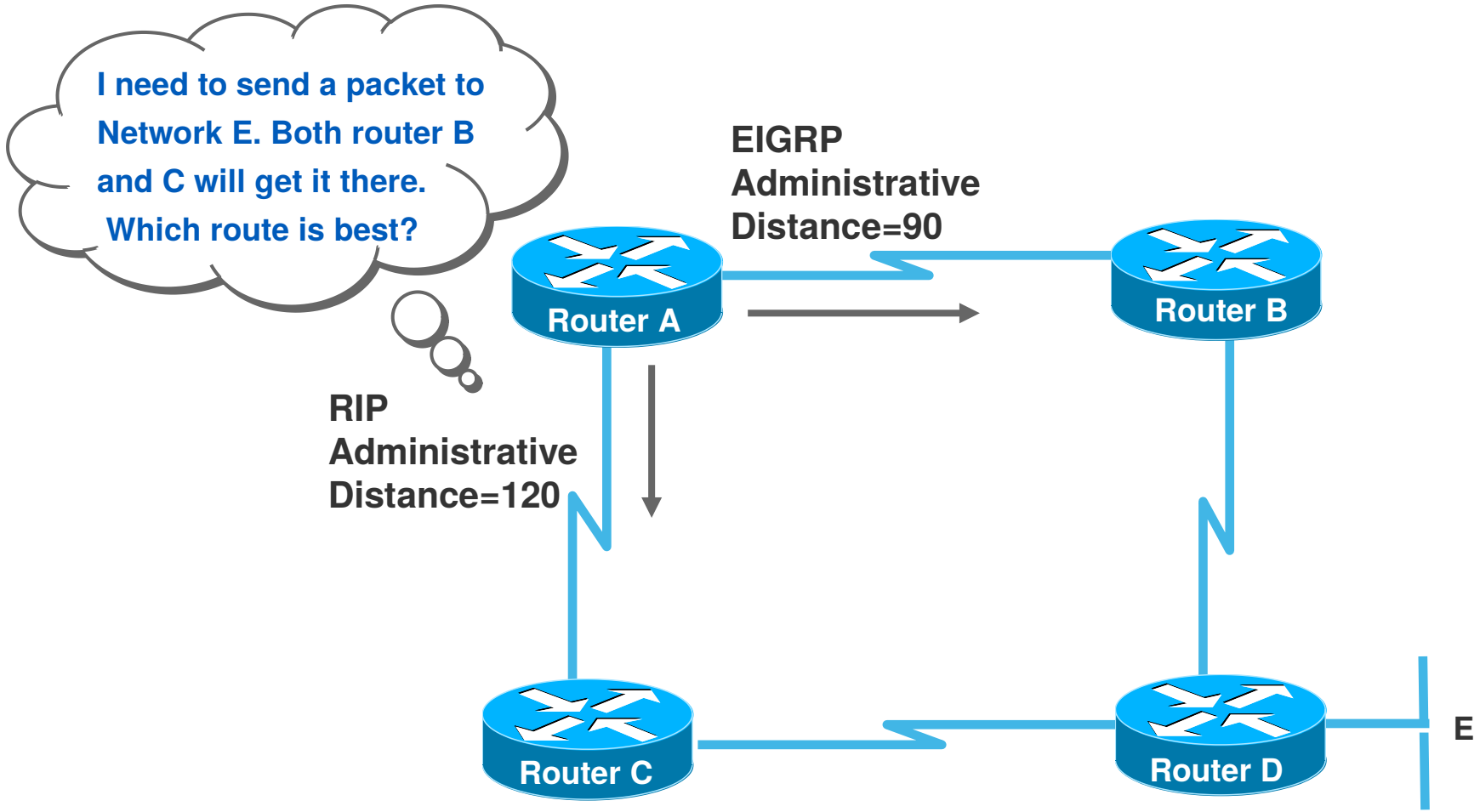

Advanced Distance Vector/Hybrid Routing (EIGRP)



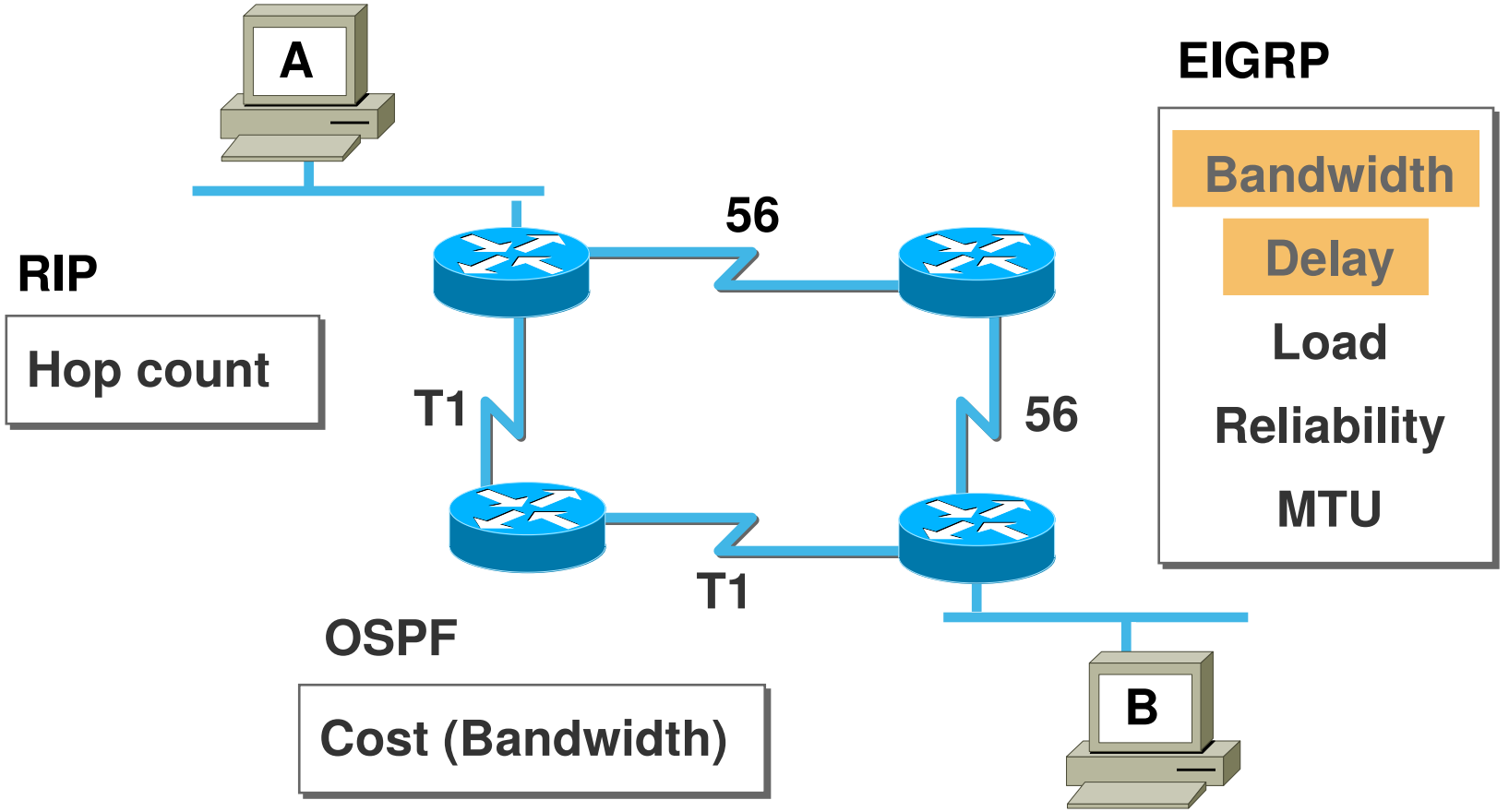

Link State (OSPF)

- **Distance-Vector Protocols (RIP, IGRP)**:
  - View network topology from neighbor's perspective.
  - Add distance vectors from router to router.
  - Frequent, periodic updates.
  - Pass copy of routing tables to neighbor routers.
- **Link State Protocols (OSPF):**
  - Gets common view of entire network topology.
  - Calculates the shortest path to other routers.
  - Event-triggered updates.
  - Passes link state routing updates to other routers.

# ADMINISTRATIVE DISTANCE (AD): RANKING ROUTE INFORMATION



# ROUTING METRICS – SELECTING THE BEST ROUTE WITH METRICS




**RIP**

Hop count

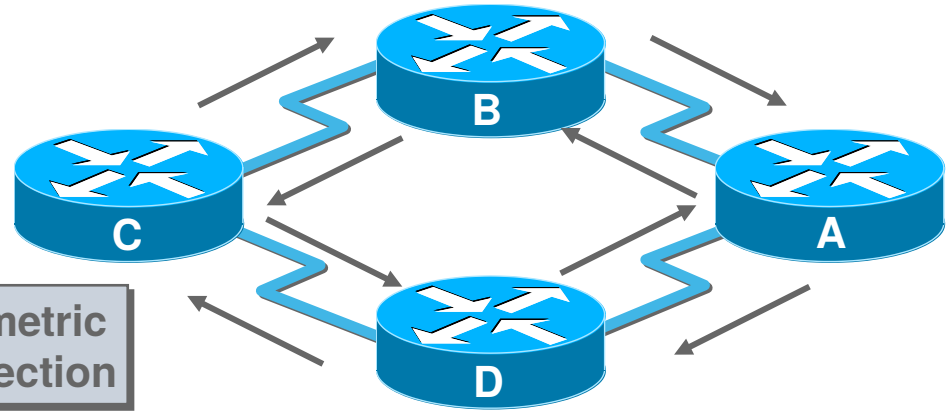
**OSPF**

Cost (Bandwidth)

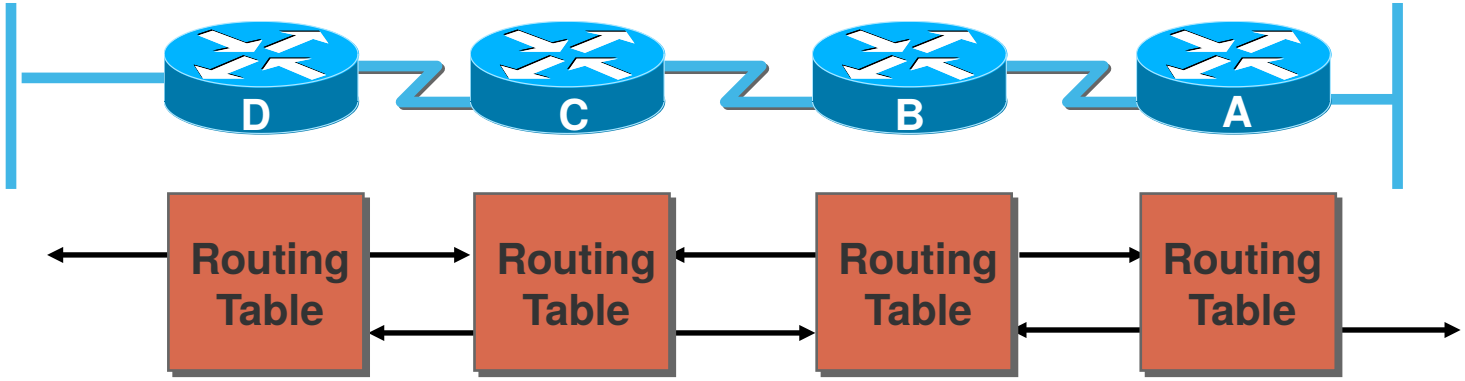
**EIGRP**

- Bandwidth
- Delay
- Load
- Reliability
- MTU

# DISTANCE VECTOR PROTOCOL – SOURCES OF INFORMATION & DISCOVERING ROUTES



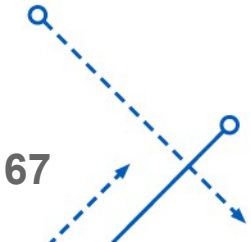

Distance—How far/metric
Vector—In which direction




**The Meaning of Distance Vector**:

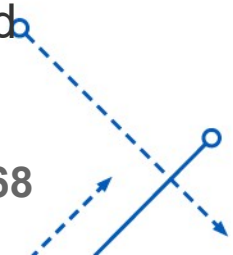
- A router using distance vector routing protocols knows 2 things:
  - Distance to final destination
    - The distance or how far it is to the destination network
  - Vector, or direction, traffic should be directed
    - The direction or interface in which packets should be forwarded

- Pass periodic copies of routing table to neighbor routers and accumulate distance vectors

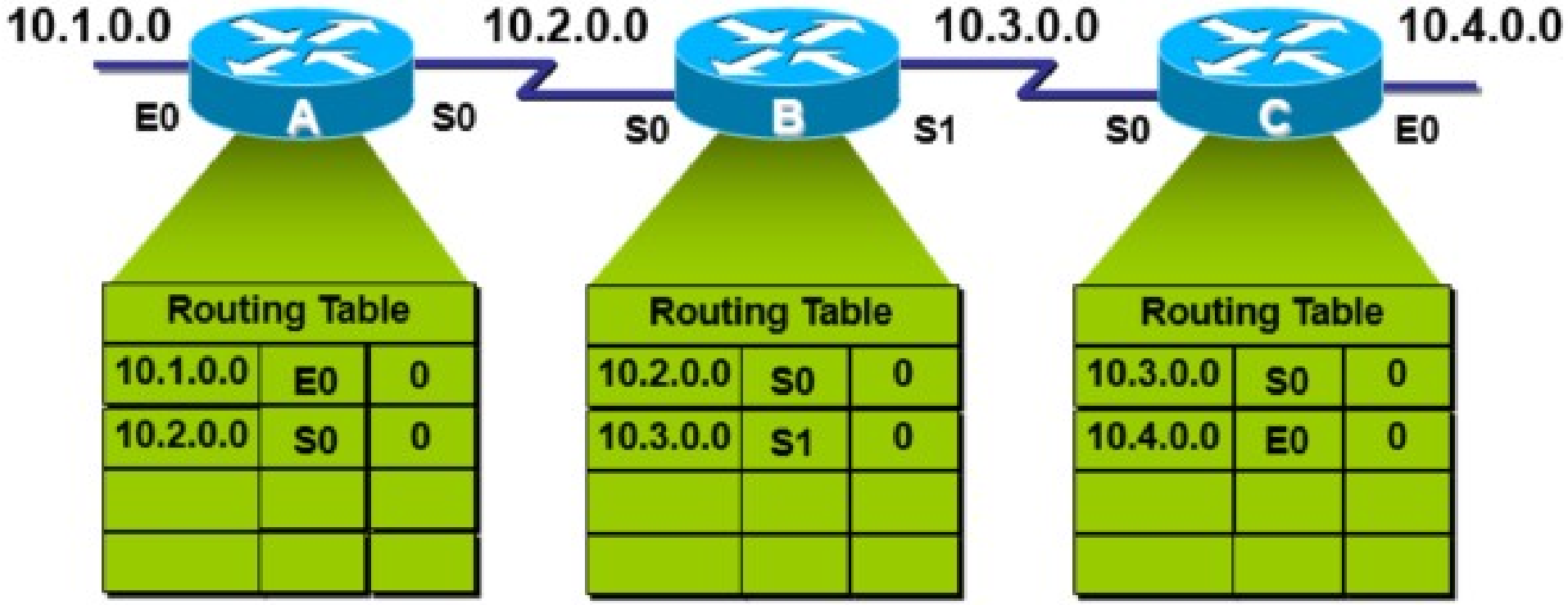


# CHARACTERISTICS OF DISTANCE VECTOR ROUTING PROTOCOLS

- **Periodic updates**
  - Periodic Updates sent at regular intervals (**30 seconds for RIP**). Even if the topology has not changed in several days,
- **Neighbors**
  - The router is only aware of the network addresses of its own interfaces and the remote network addresses it can reach through its neighbors.
  - It has no broader knowledge of the network topology
- **Broadcast updates**
  - Broadcast Updates are sent to 255.255.255.255.
  - Some distance vector routing protocols use multicast addresses instead of broadcast addresses.
- **Entire routing table is included with routing update**
  - Entire Routing Table Updates are sent, with some exceptions to be discussed later, periodically to all neighbors.
  - Neighbors receiving these updates must process the entire update to find pertinent information and discard the rest.
  - Some distance vector routing protocols like EIGRP do not send periodic routing table updates.

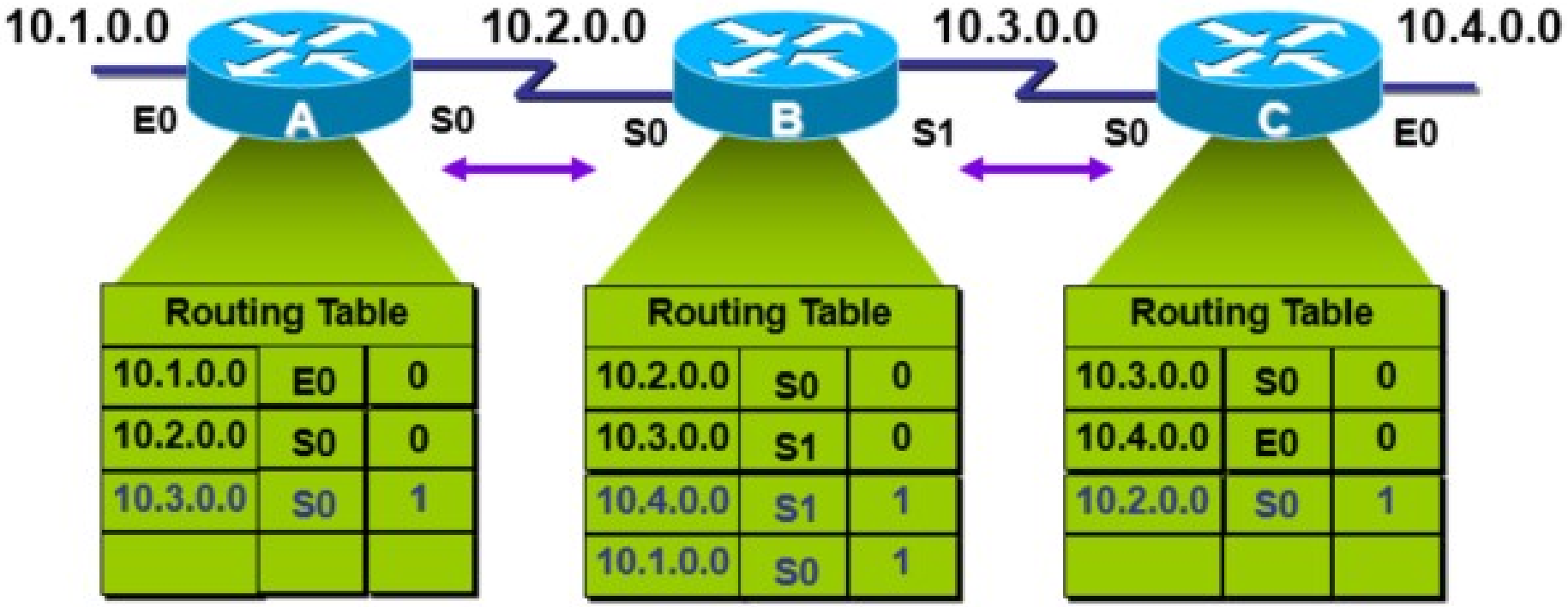


# DISTANCE VECTOR – DISCOVERING ROUTES



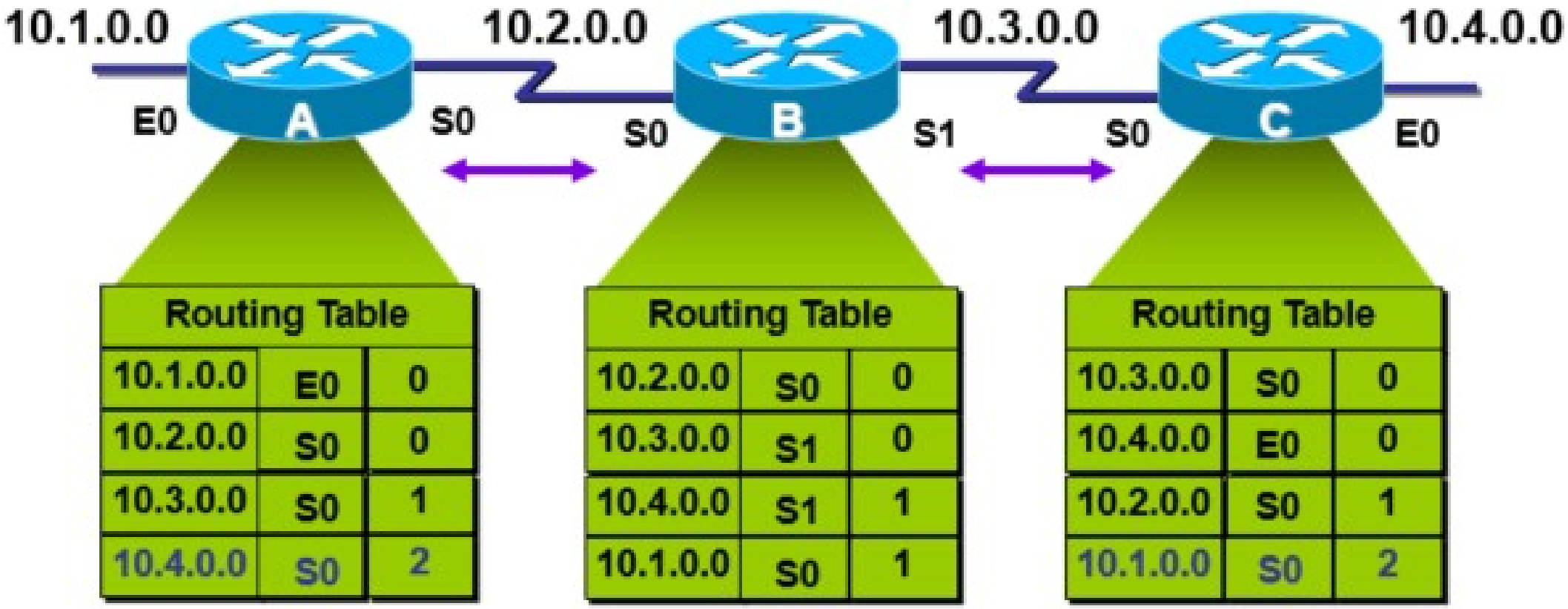

- Routers discover the best path to destinations from each neighbor → **RIP protocol**

# DISTANCE VECTOR – DISCOVERING ROUTES



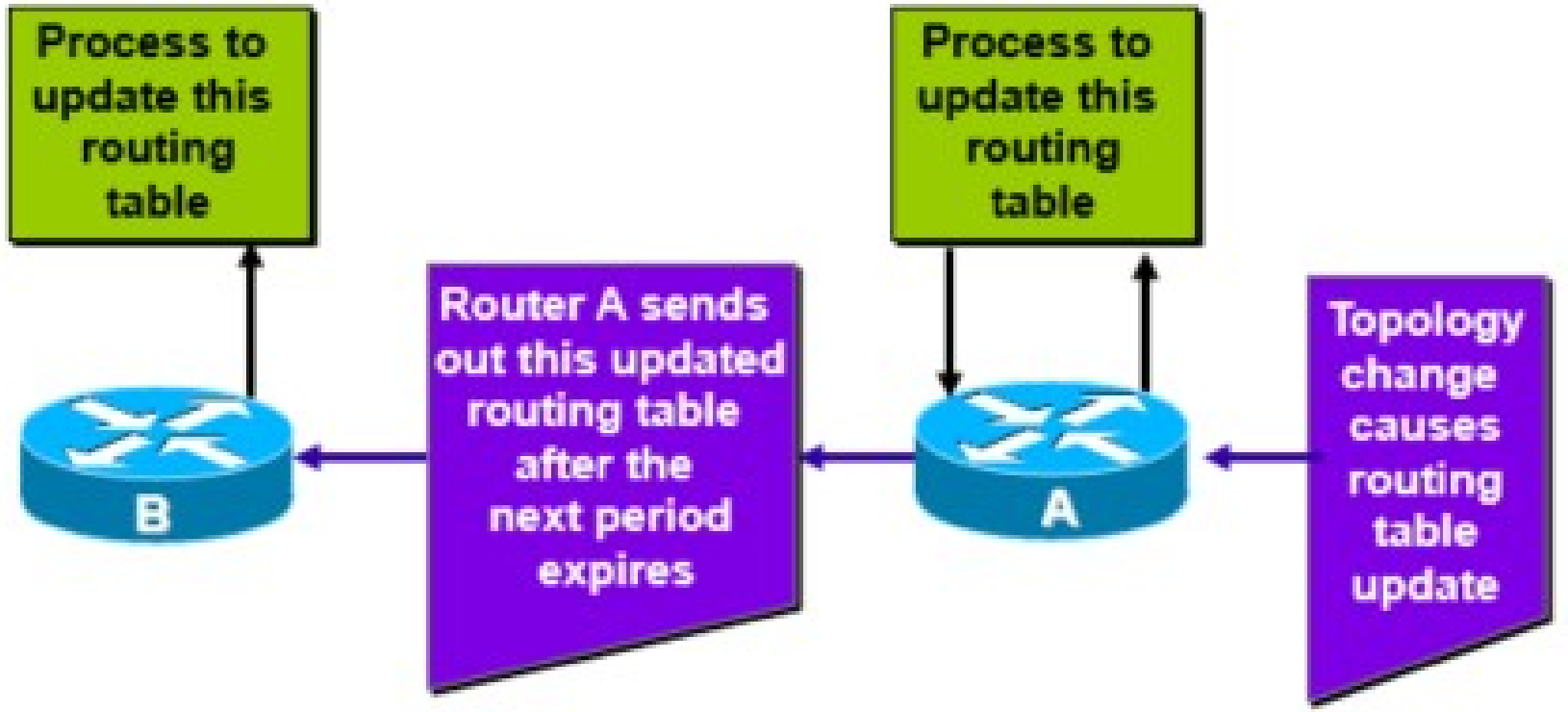

- Routers discover the best path to destinations from each neighbor

# DISTANCE VECTOR – DISCOVERING ROUTES



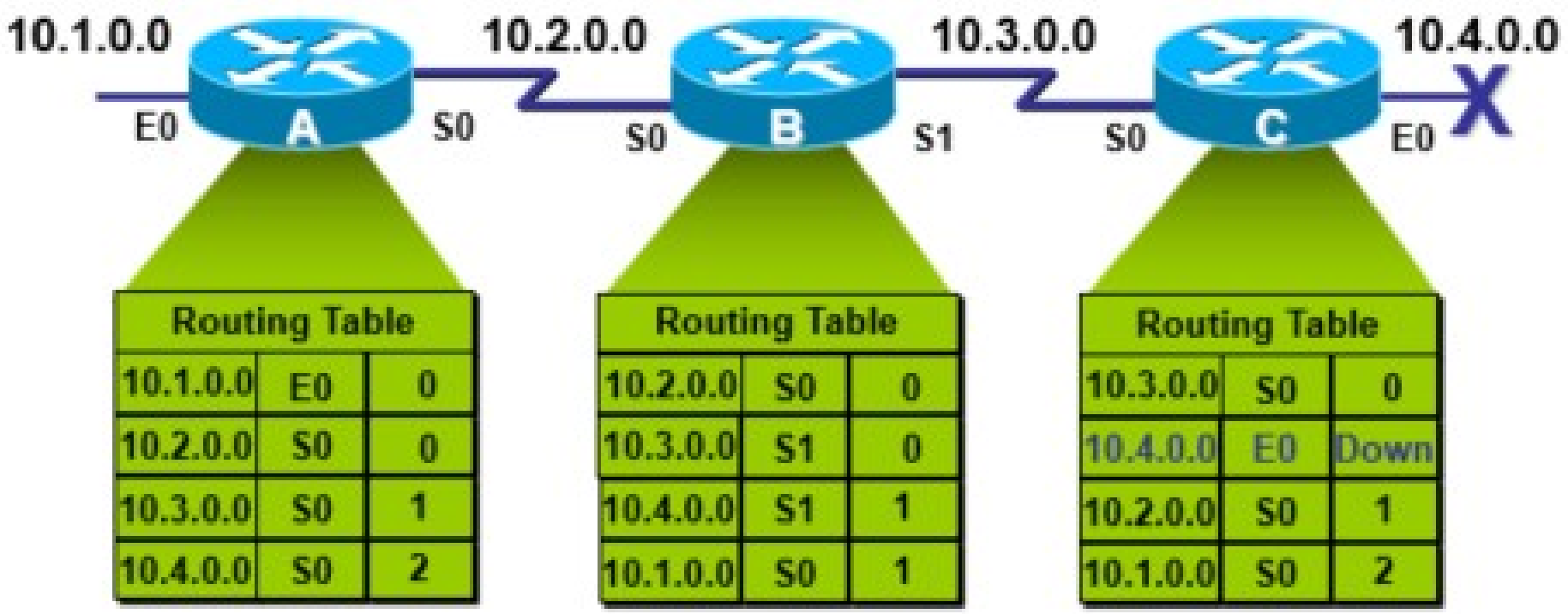

- Each node maintains the distance from itself to each possible destination network

# MAINTAINING ROUTING INFORMATION



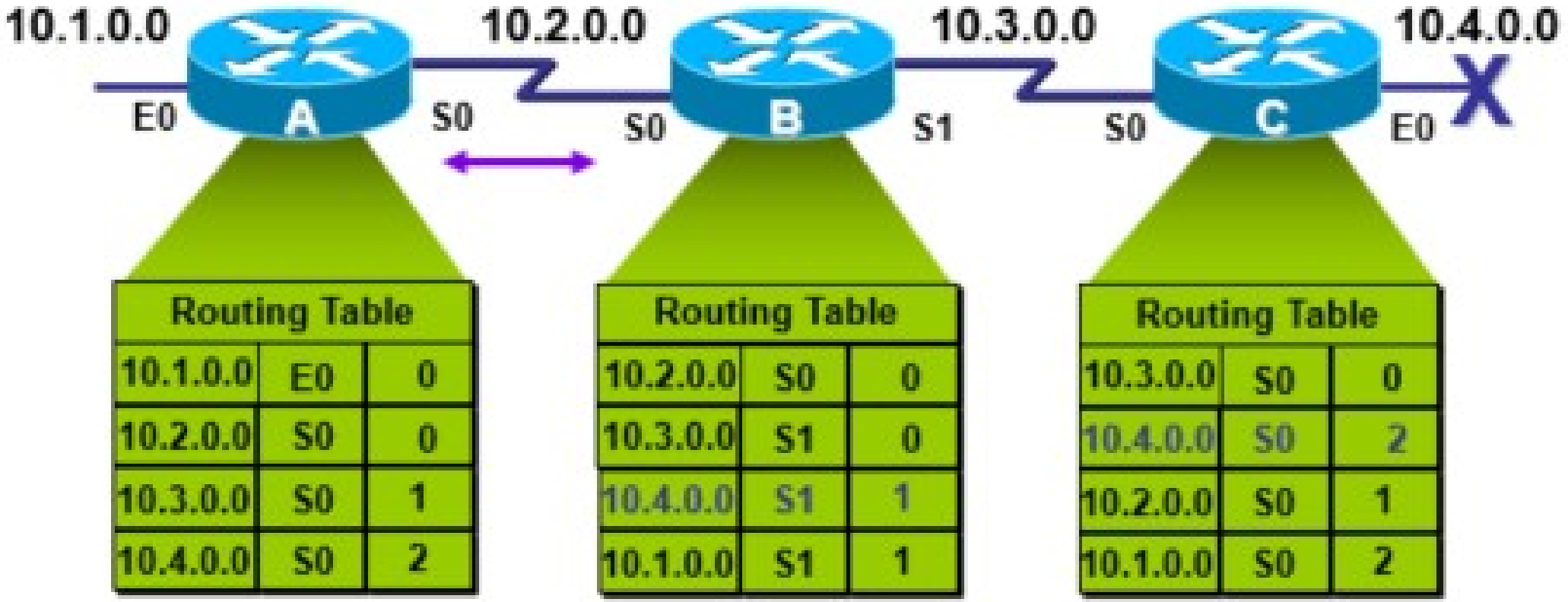

- Updates proceed step-by-step from router to router

# PROBLEM – ROUTING LOOPS




- Slow convergence produces inconsistent routing

# PROBLEM – ROUTING LOOPS




Routing Table

| | | |
|---|---|---|
| 10.1.0.0 | E0 | 0 |
| 10.2.0.0 | S0 | 0 |
| 10.3.0.0 | S0 | 1 |
| 10.4.0.0 | S0 | 2 |

Routing Table

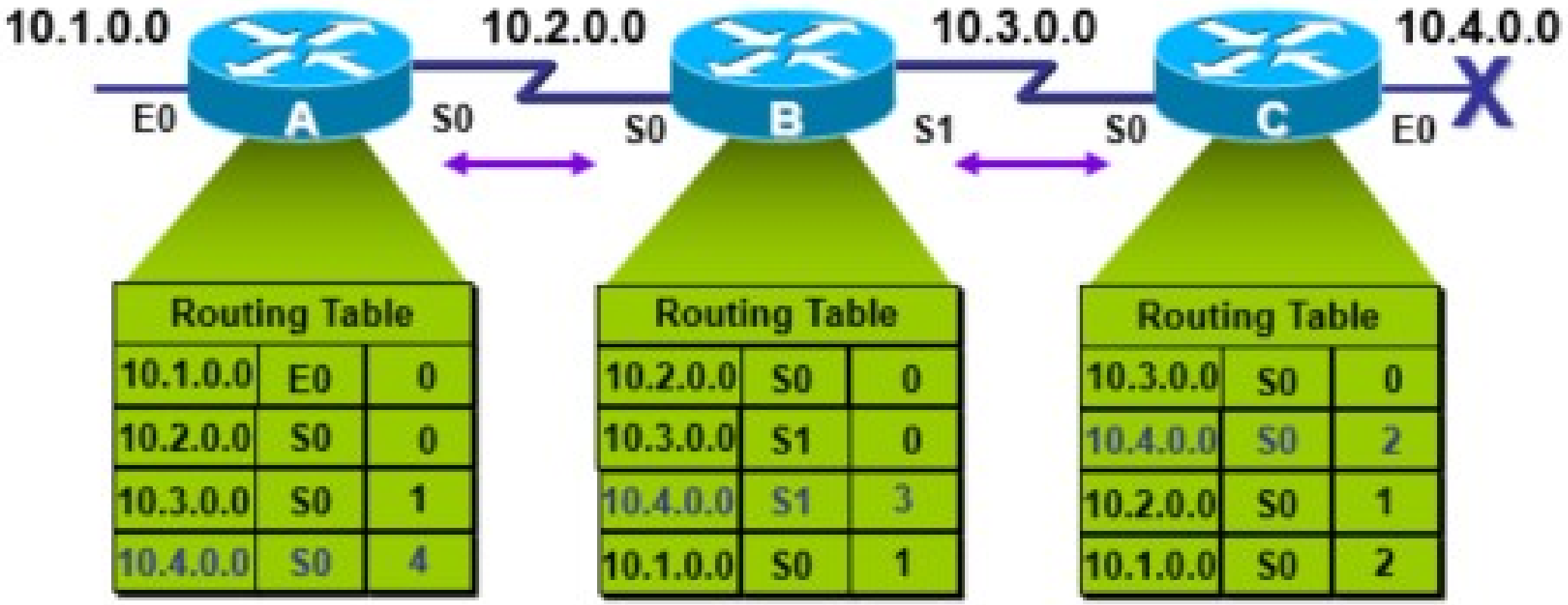
| | | |
|---|---|---|
| 10.2.0.0 | S0 | 0 |
| 10.3.0.0 | S1 | 0 |
| 10.4.0.0 | S1 | 1 |
| 10.1.0.0 | S1 | 1 |

Routing Table

| | | |
|---|---|---|
| 10.3.0.0 | S0 | 0 |
| 10.4.0.0 | S0 | 2 |
| 10.2.0.0 | S0 | 1 |
| 10.1.0.0 | S0 | 2 |

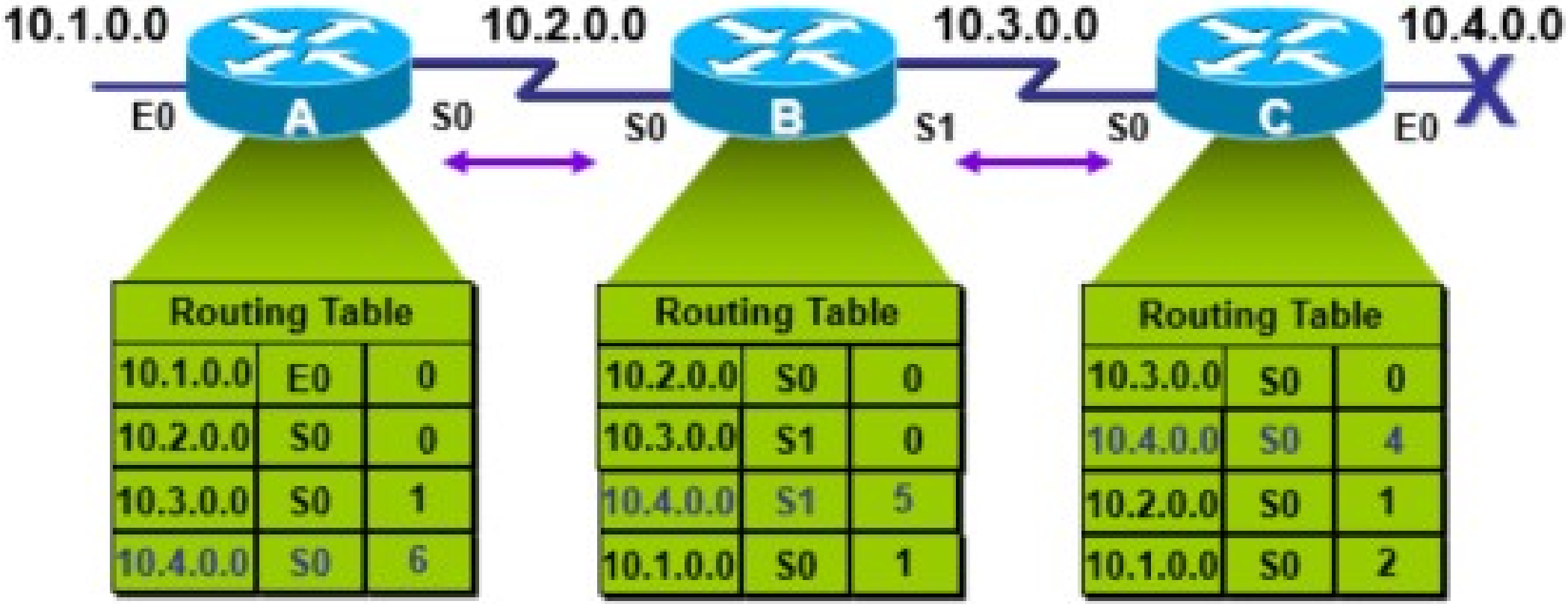
**Router C concludes that the best path to network 10.4.0.0 is through Router B**

# PROBLEM – ROUTING LOOPS



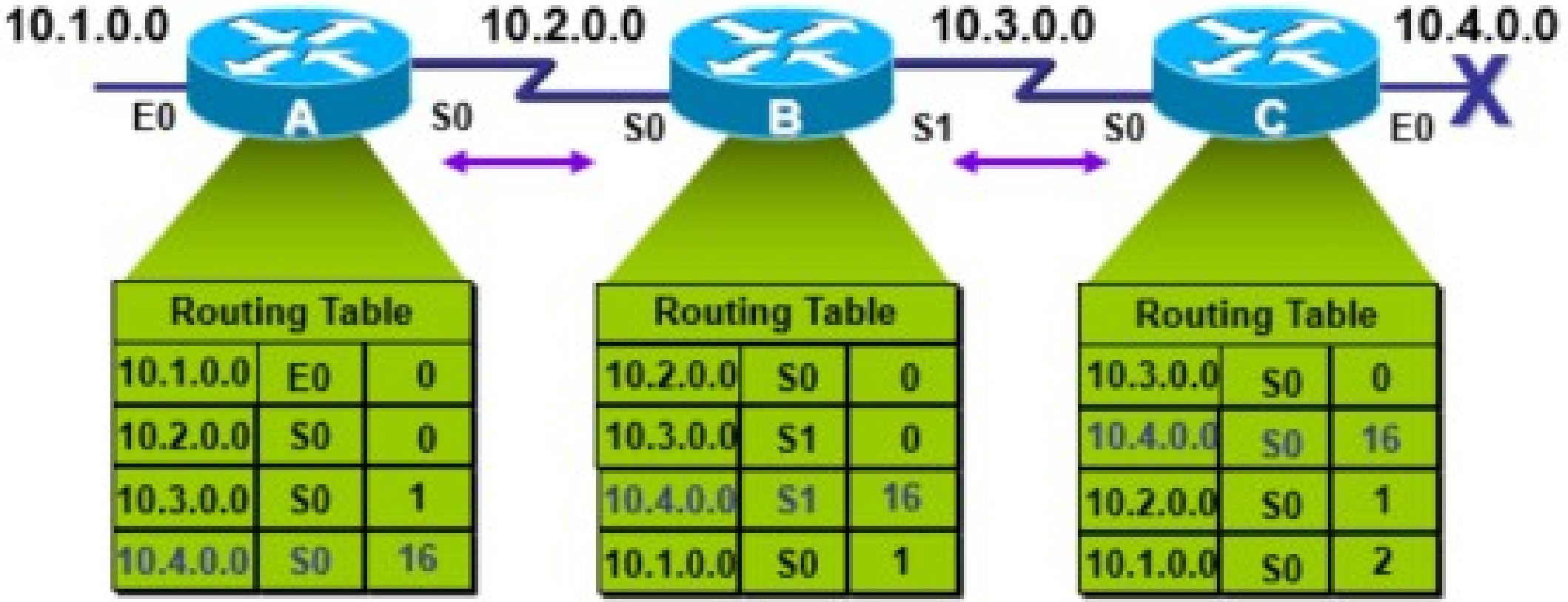

**Router A updates its table to reflect the new but erroneous hop count**

# SYMPTOM: COUNTING TO INFINITY



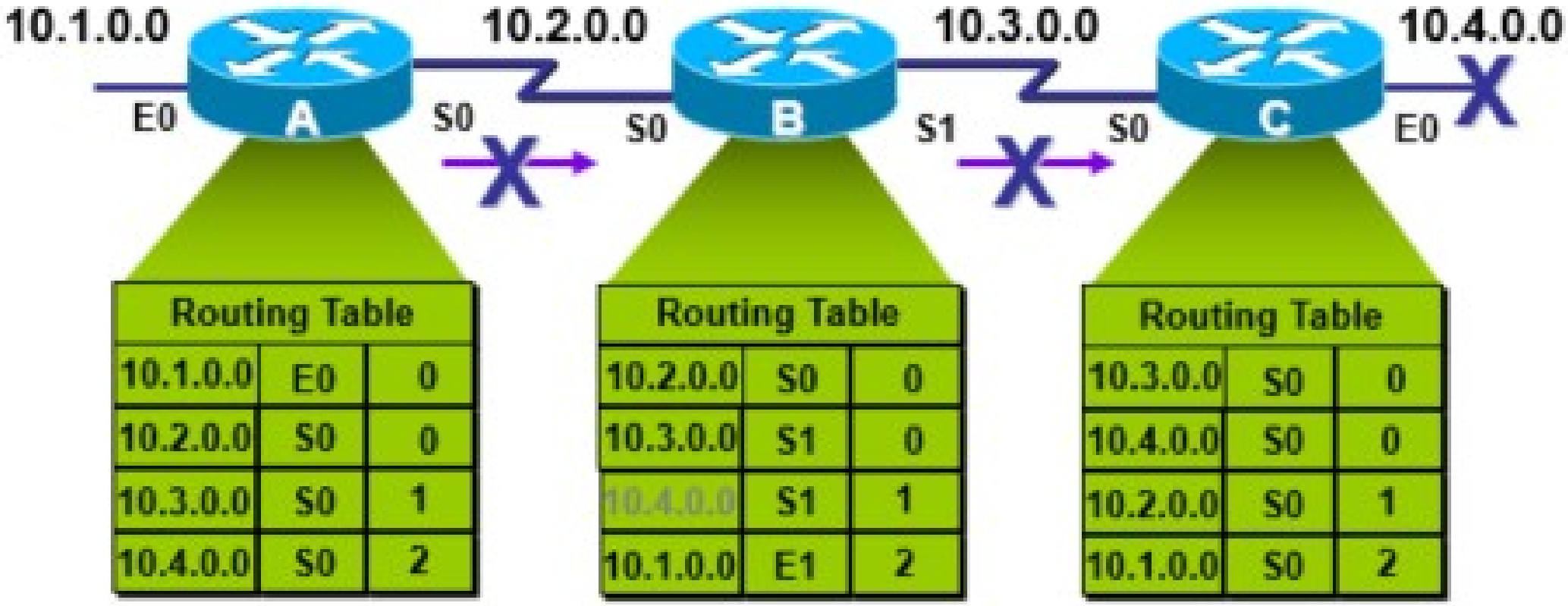

- Packets for network 10.4.0.0 bounce between routers A, B, and C
- Hop count for network 10.4.0.0 counts to infinity

# SOLUTION: DEFINING A MAXIMUM



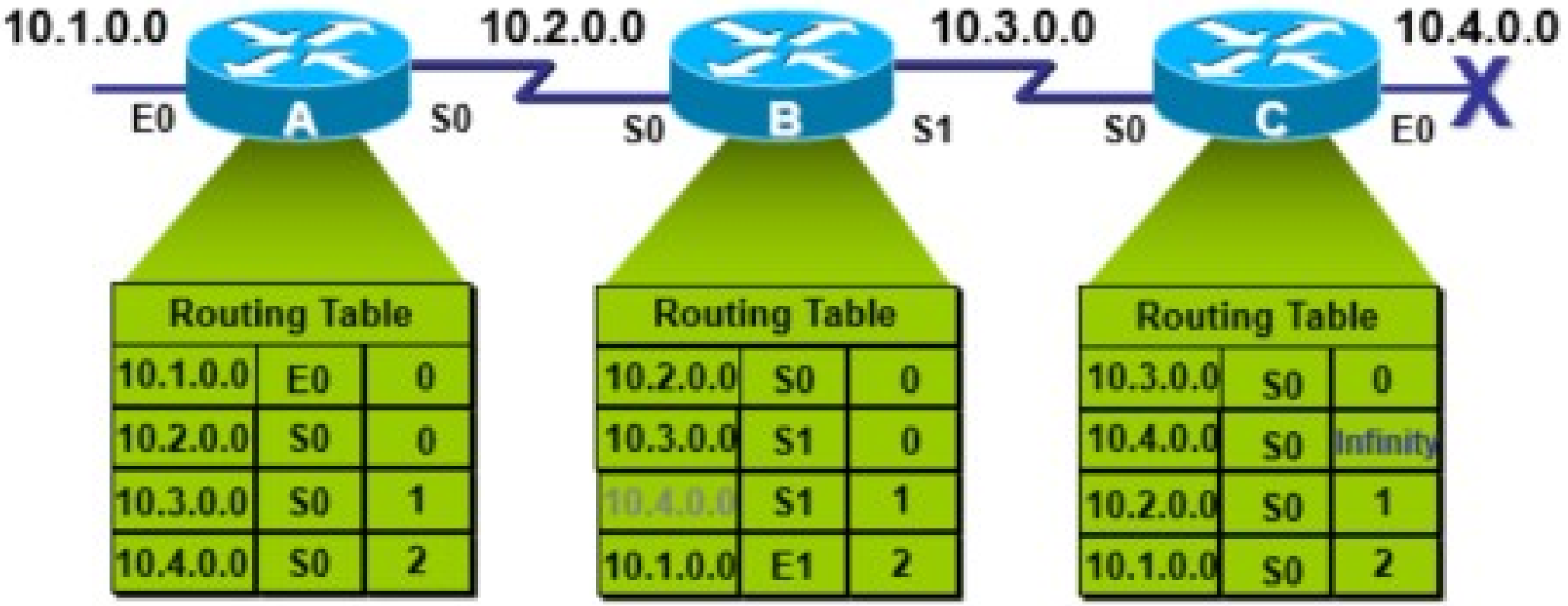

- Define a limit on the number of hops to prevent infinite loops

# SOLUTION: SPLIT HORIZON



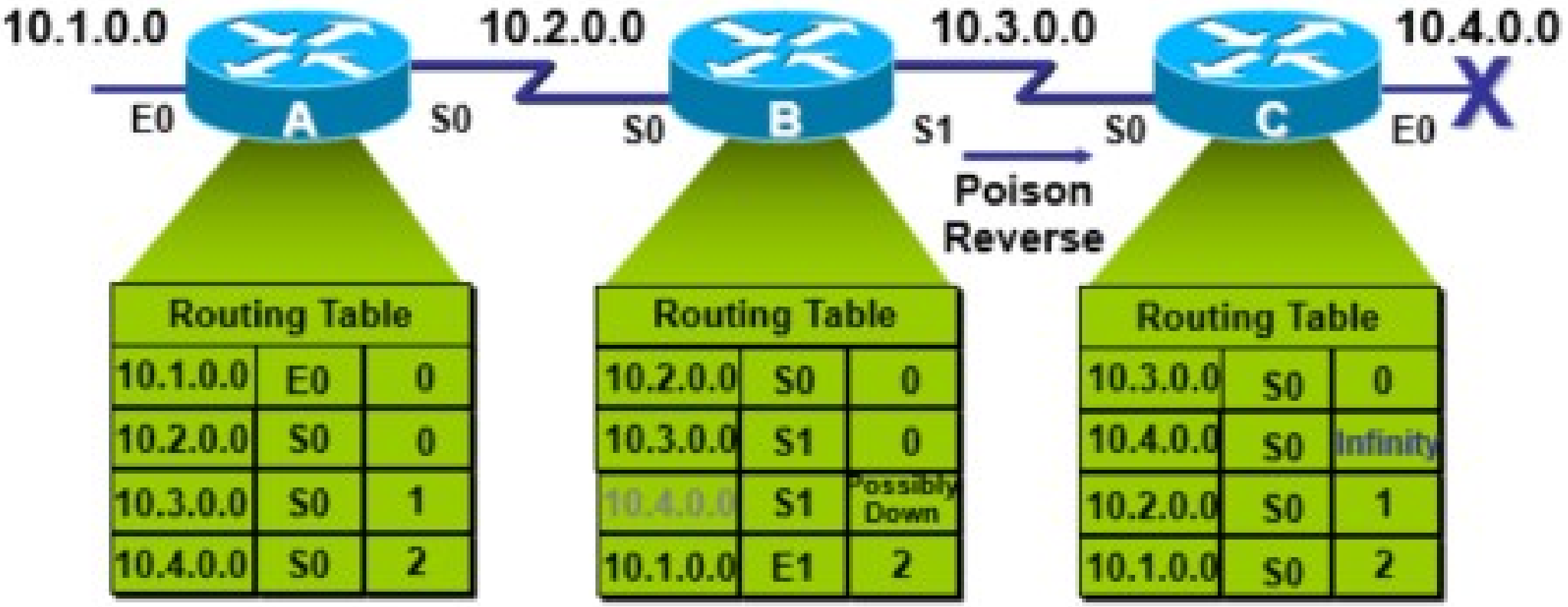

- It is never useful to send information about a route back in the direction from which the original packet came

# SOLUTION: ROUTE POISONING



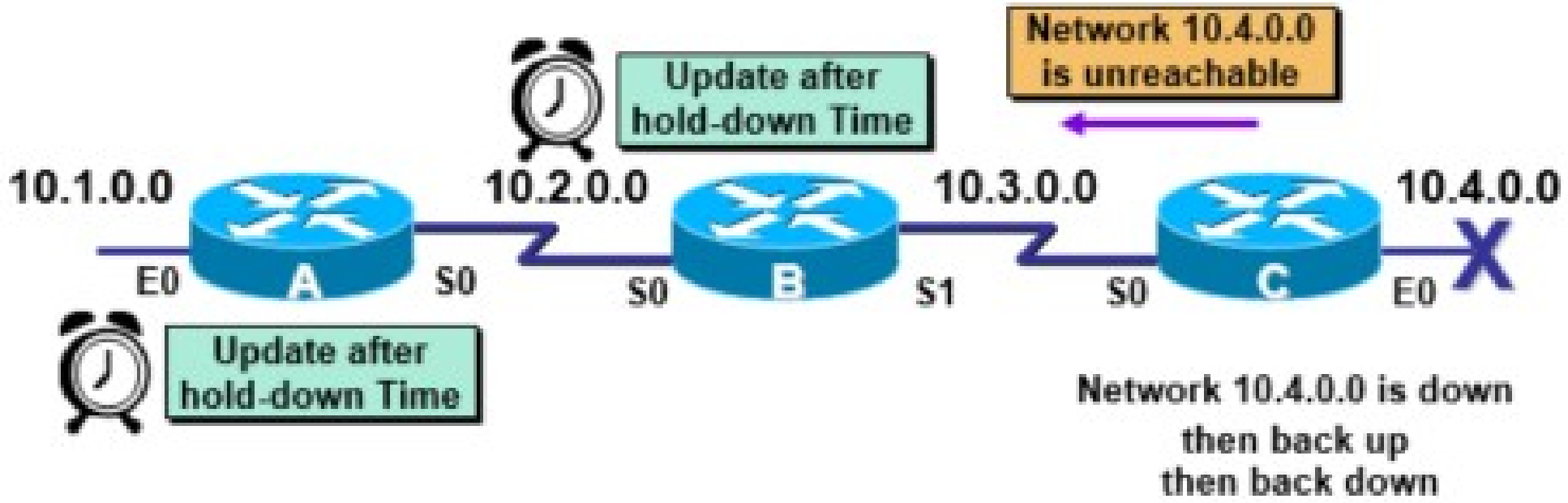

- Routers set the distance of routes that have gone down to infinity

# SOLUTION: POISON REVERSE



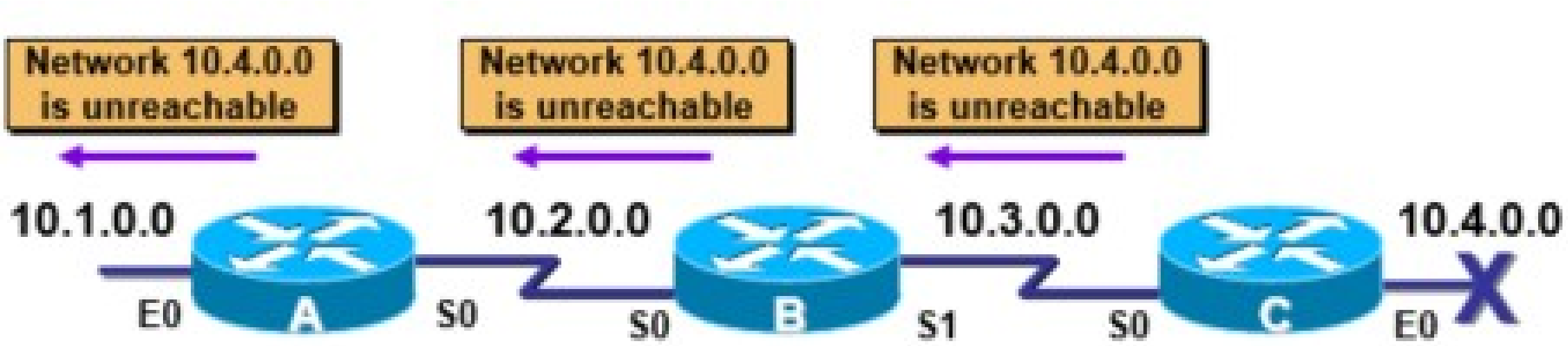

- Poison Reverse overrides split horizon

# SOLUTION: HOLD-DOWN TIMERS



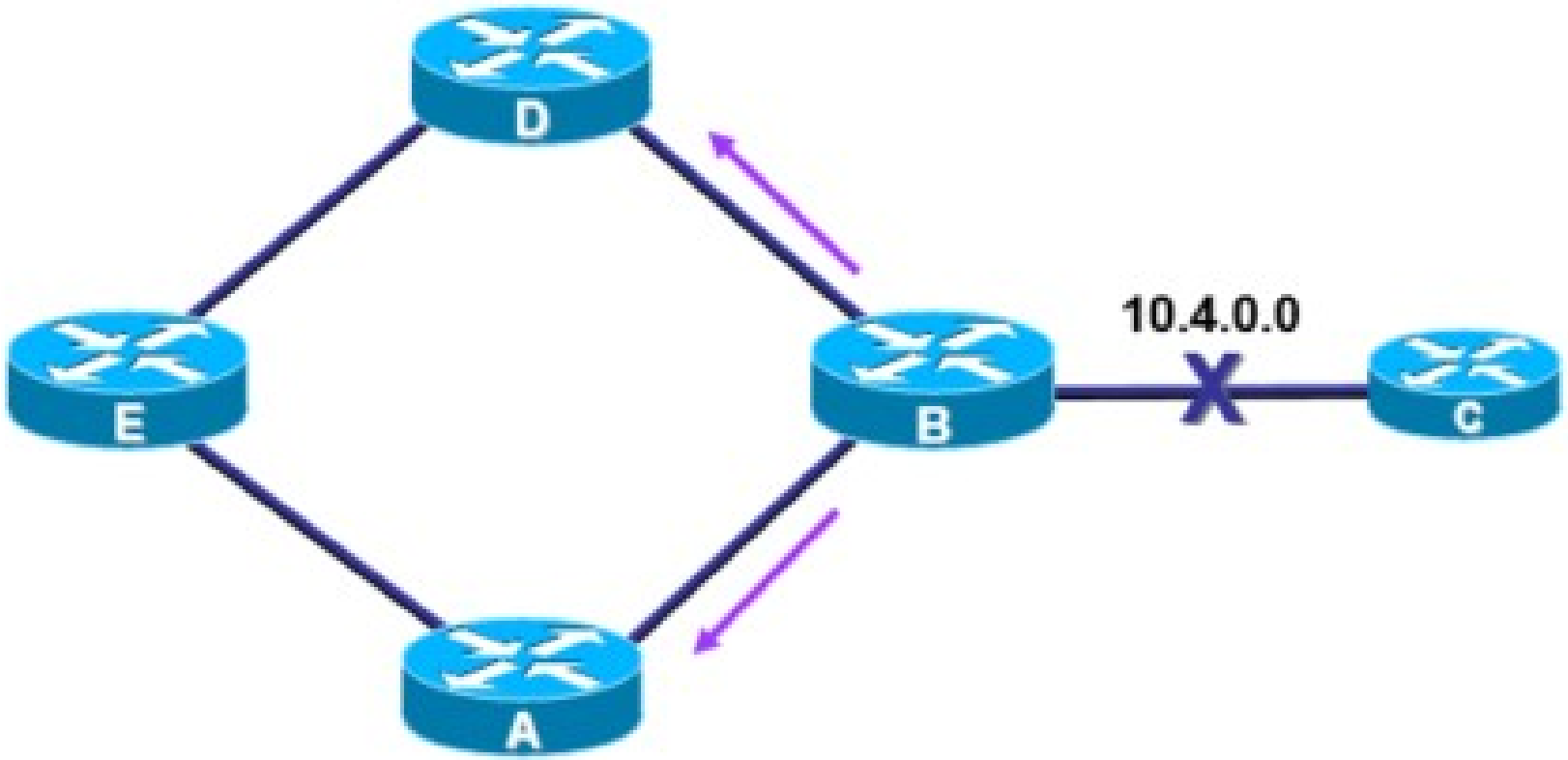

- Router keeps an entry for the network possibly down state, allowing time for other routers to recompute for this topology change

# SOLUTION: TRIGGERED UPDATES

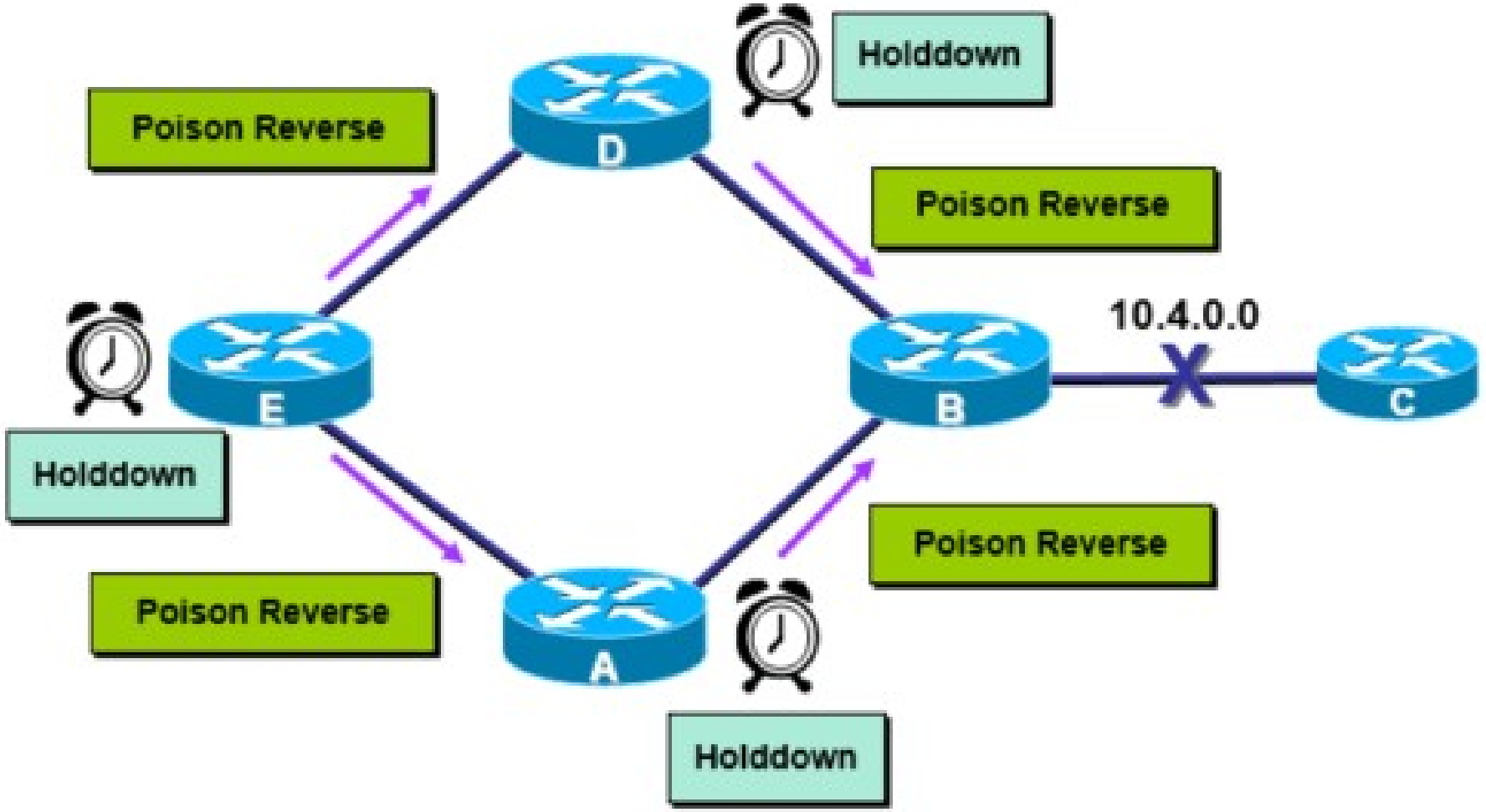



- Router sends updates when a change in its routing table occurs

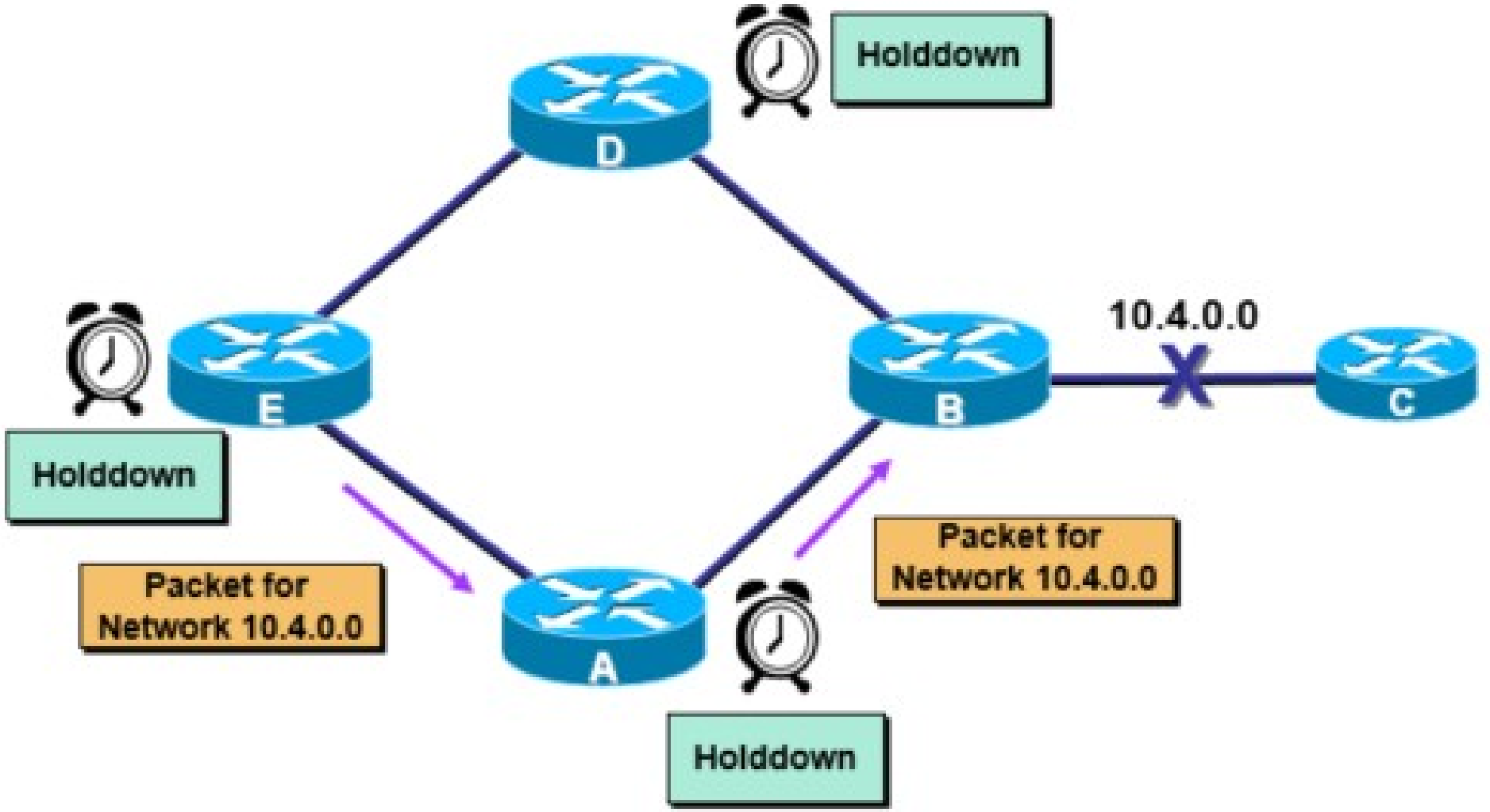
# IMPLEMENTING SOLUTIONS IN MULTIPLE ROUTES



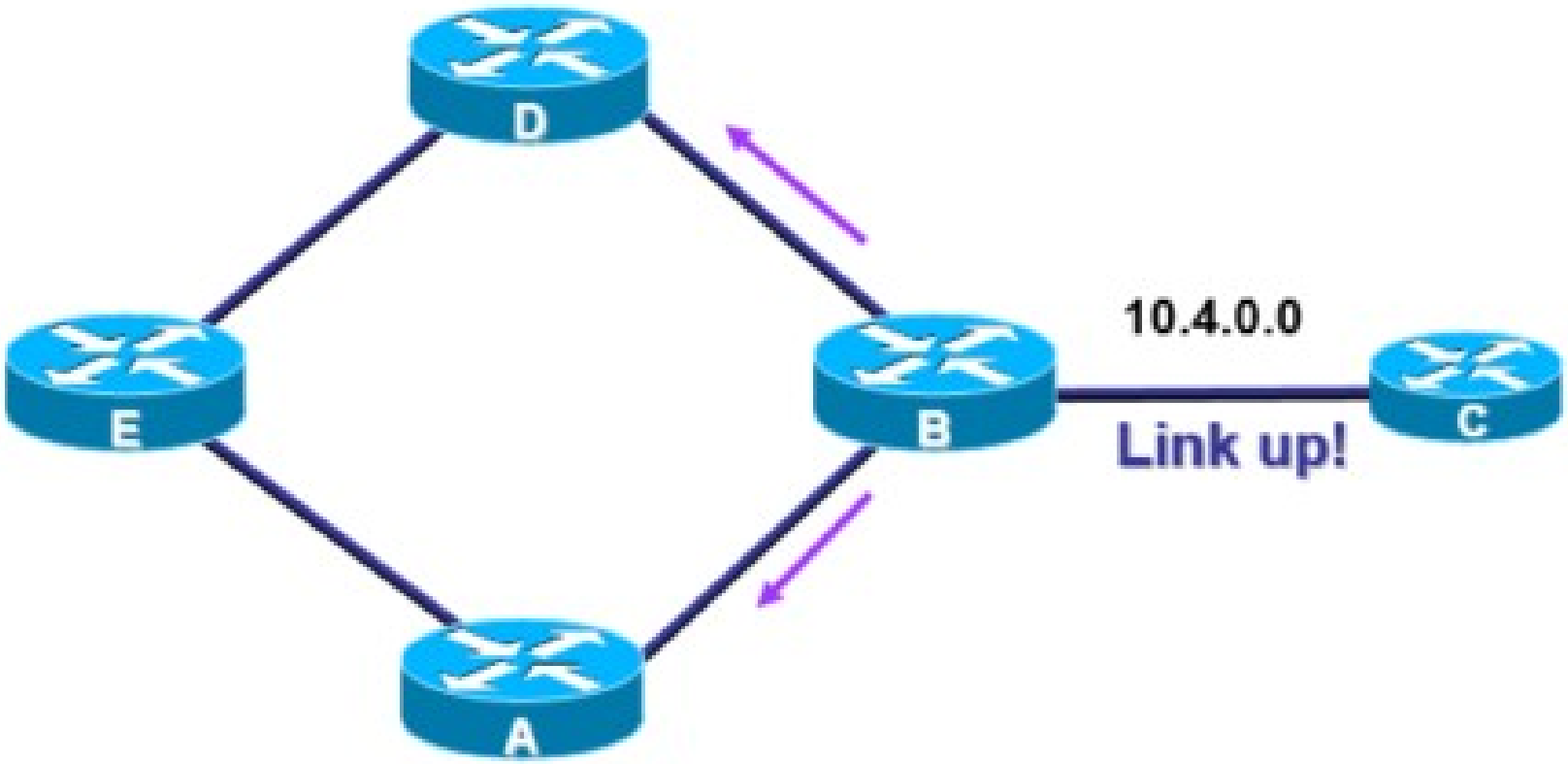
# IMPLEMENTING SOLUTIONS IN MULTIPLE ROUTES



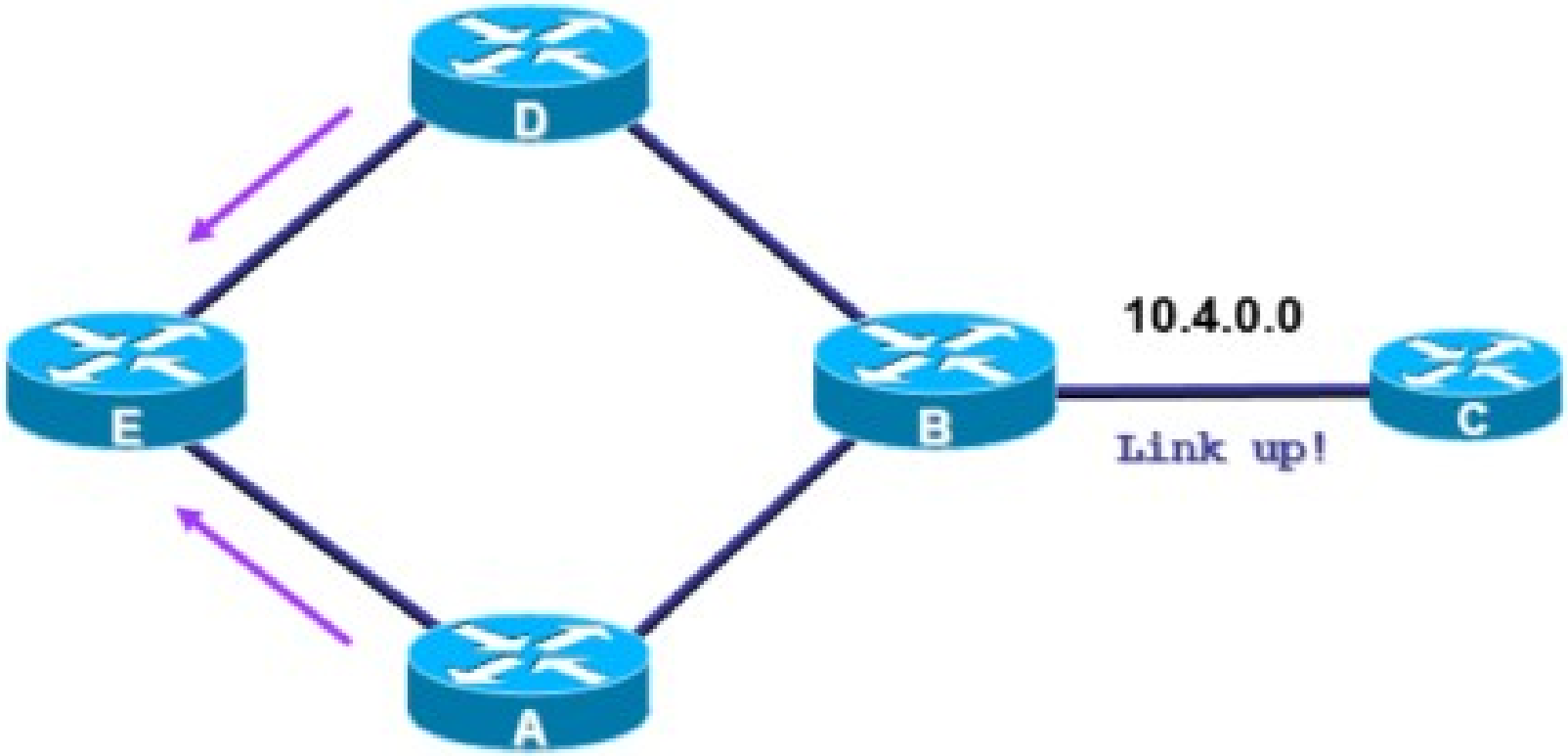
# IMPLEMENTING SOLUTIONS IN MULTIPLE ROUTES



# IMPLEMENTING SOLUTIONS IN MULTIPLE ROUTES

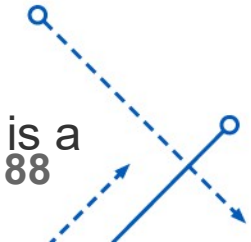


# IMPLEMENTING SOLUTIONS IN MULTIPLE ROUTES



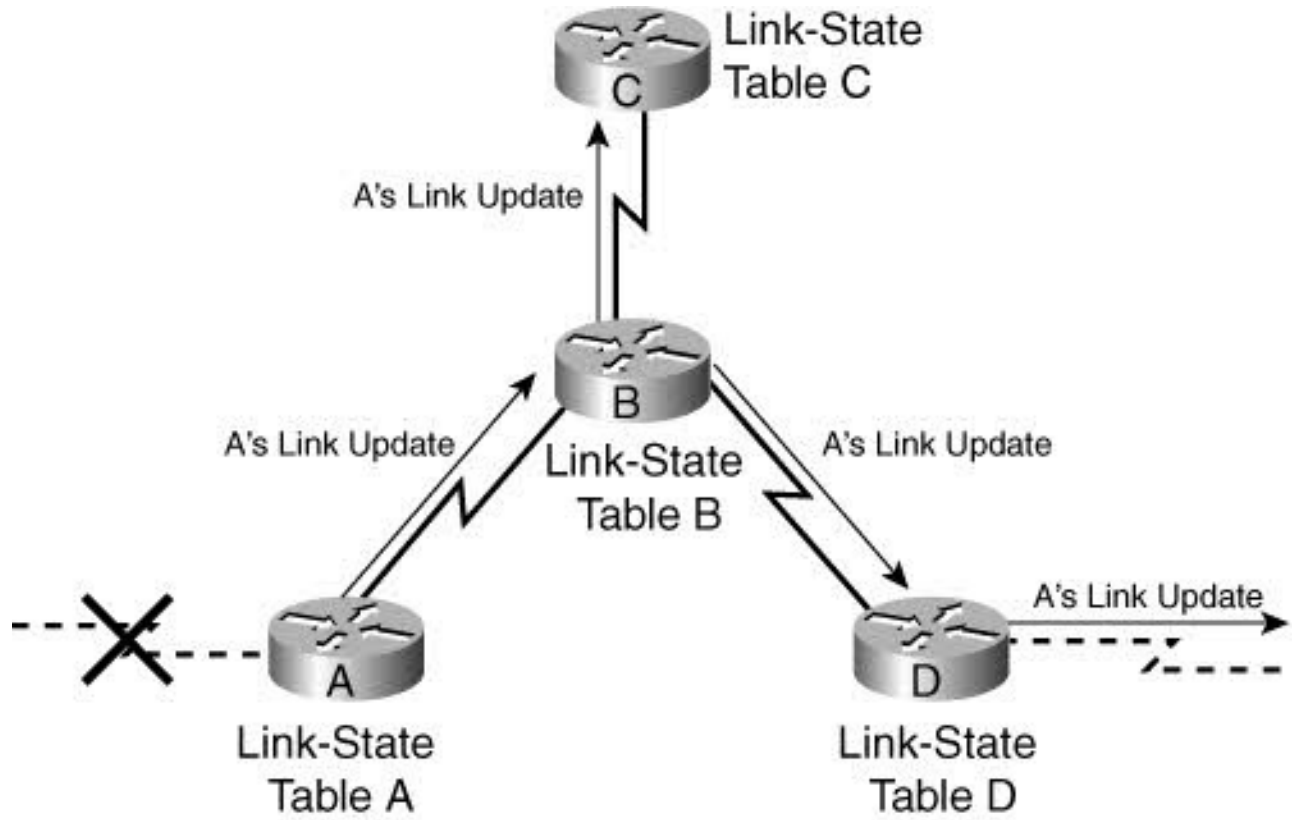
## SUPPLEMENT: DIFFERENT DISTANCE-VECTOR ROUTING PROTOCOLS

- Routing Information Protocol (RIP)
  - RFC 1058.
  - Hop count is used as the metric for path selection.
  - If the hop count for a network is greater than 15, RIP cannot supply a route to that network.
  - Routing updates are broadcast or multicast every 30 seconds, by default.
- Interior Gateway Routing Protocol (IGRP)
  - Proprietary protocol developed by Cisco.
  - Bandwidth, delay, load and reliability are used to create a composite metric.
  - Routing updates are broadcast every 90 seconds, by default.
  - IGRP is the predecessor of EIGRP and is now obsolete.
- Enhanced Interior Gateway Routing Protocol (EIGRP)
  - Cisco proprietary distance vector routing protocol.
  - It can perform unequal cost load balancing.
  - It uses Diffusing Update Algorithm (DUAL) to calculate the shortest path.
  - There are no periodic updates as with RIP and IGRP. Routing updates are sent only when there is a change in the topology.



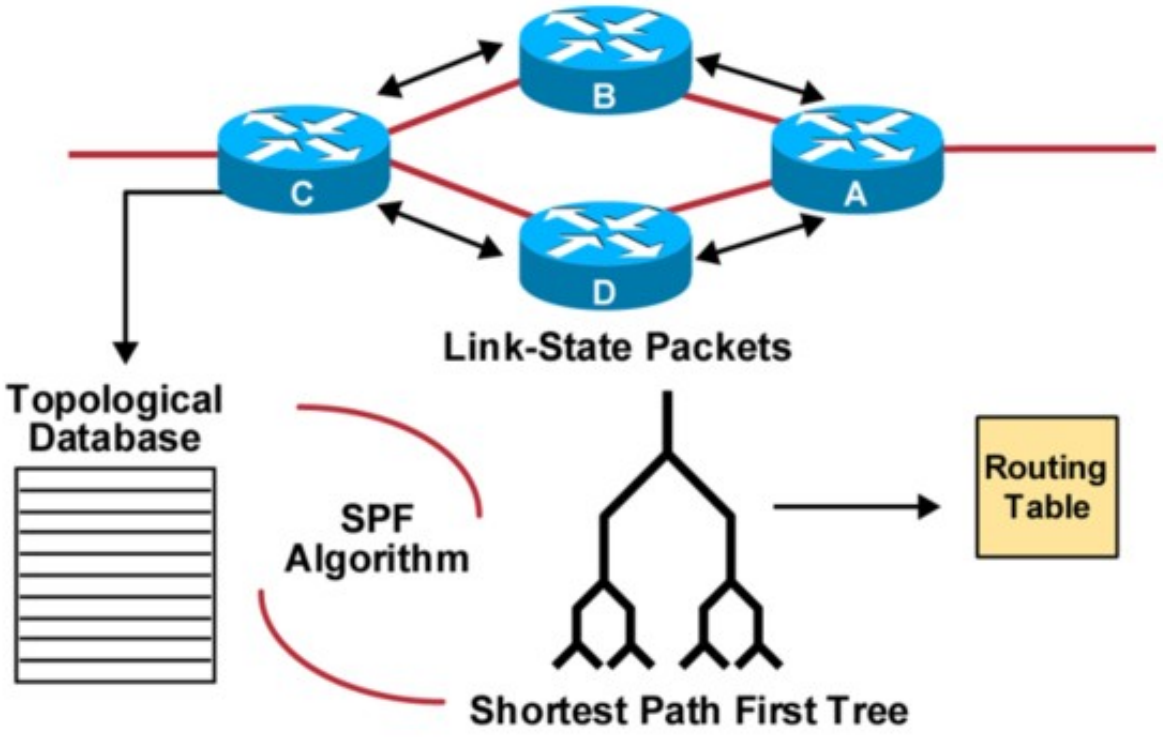
# LINK-STATE ROUTING PROTOCOL

- Each router contains a database containing a map of the whole topology
  - **Links** (an interface on a router)
  - **Link state** (including cost, information about the state of the links )
- All routers have the same information
- All routers calculate the best path to every destination
- Any link state changes are flooded across the network
  - "*Global spread of local knowledge*"



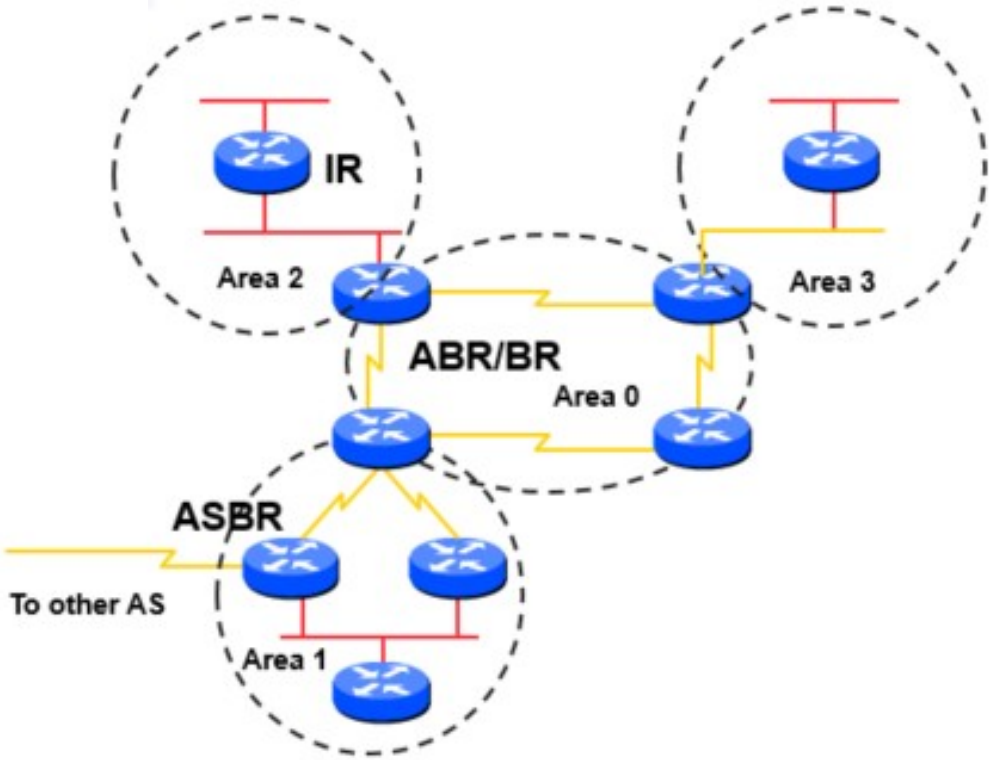
# LINK-STATE ROUTING PROTOCOL

- Automatic neighbour discovery
  - Neighbours are physically connected routers
- Each router constructs a **Link State Packet** (**LSP**)
  - Distributes the LSP to neighbours…
  - …using an **LSA** (**Link State Announcement**)
- Each router computes its best path to every destination (**SPF algorithm = Dijkstra algorithm**)
- On network failure
  - New LSPs are flooded
  - All routers recompute routing table



# OSPF AREAS CONCEPT

- Group of contiguous hosts and networks
- Per area topological database
  - Invisible outside the area
  - Reduction in routing traffic
- Backbone area contiguous
  - All other areas must be connected to the backbone
- **Virtual Links**: If topology is such that an area cannot have a physical connection to a device in area 0, then a virtual link must be configured



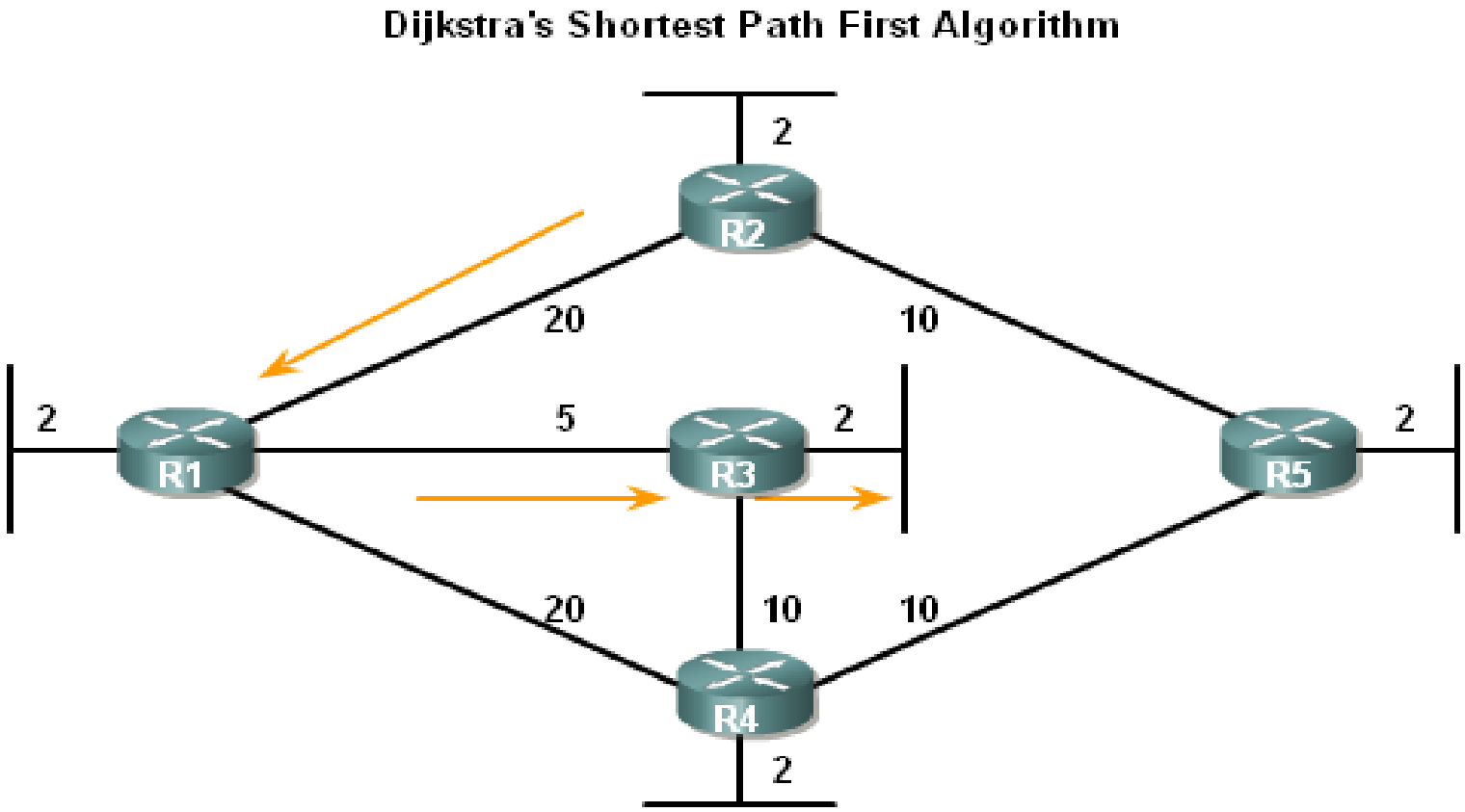

- Internal Router (IR)
- Area Border Router (ABR)
- Backbackbone Router (BR)
- Autonomous System Border Router (ASBR)

# LINK-STATE ROUTING PROTOCOL

## Actions of routers:

- Select Router-id
- Establish neighbor relations
- Exchange LSDB
- Build routing table

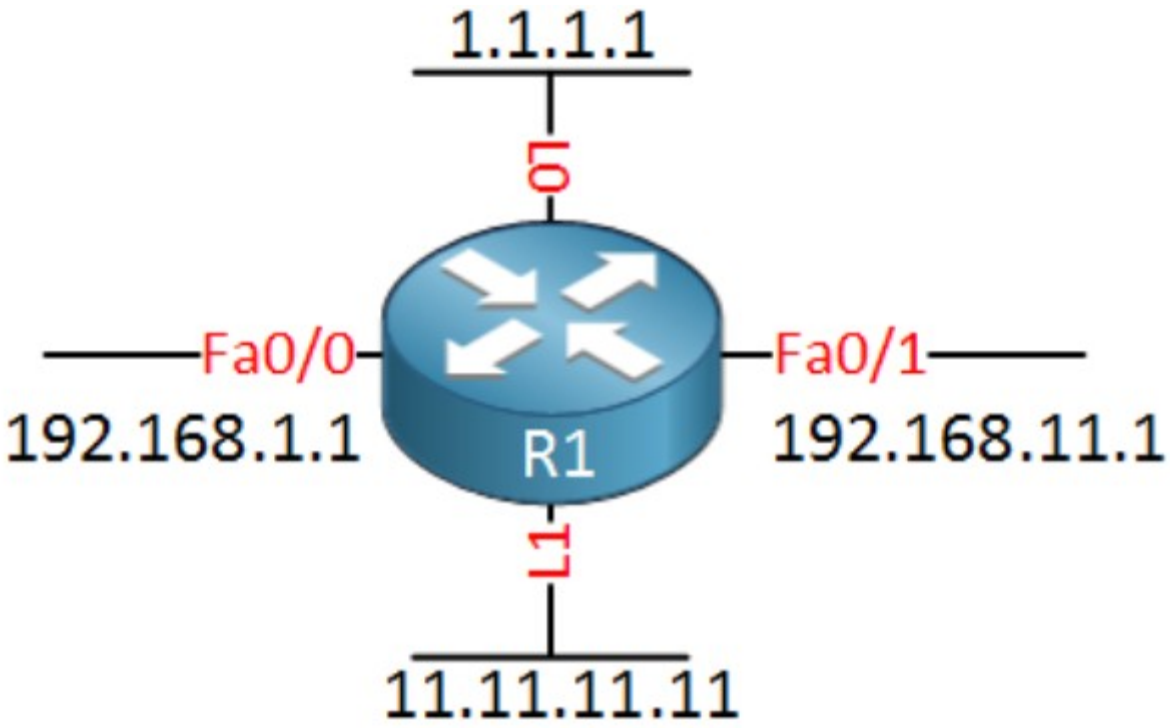
$AD_{OSPF}$ = 110
Metric = cost ← {Bandwidth}



# LINK-STATE PROTOCOL – OPEN SHORTEST PATH FIRST(OSPF) – HOW IT WORKS

## Step 0 – Select Router-id

- Router-id is the value used to identify the Router when participating in the OSPF routing environment
- Router-id has IP format (**NOT IP ADDRESS**)
- **Priority** to select Router-id:
  - Manually configured
  - If there is no manually configured router ID → Router-id = loopback interface IP address
  - If there is no configured loopback interface → Router-id = highest IP address on active interfaces

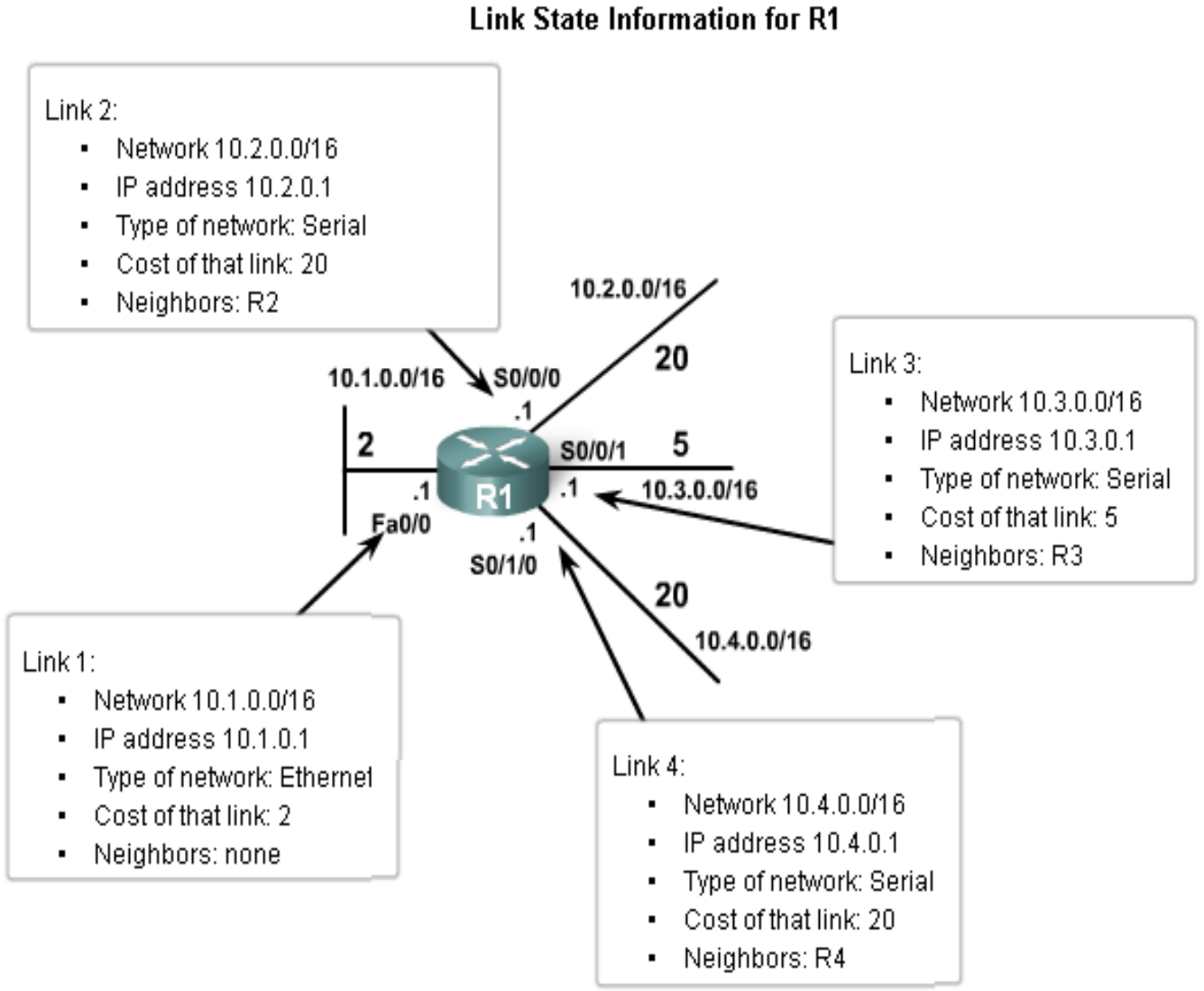



**Router ID 11.11.11.11**

# LINK-STATE PROTOCOL – OPEN SHORTEST PATH FIRST(OSPF) – HOW IT WORKS

## Step 1 – Learn about directly connected Networks

**AD = 110**
**Metric = cost ← {Bandwidth}**



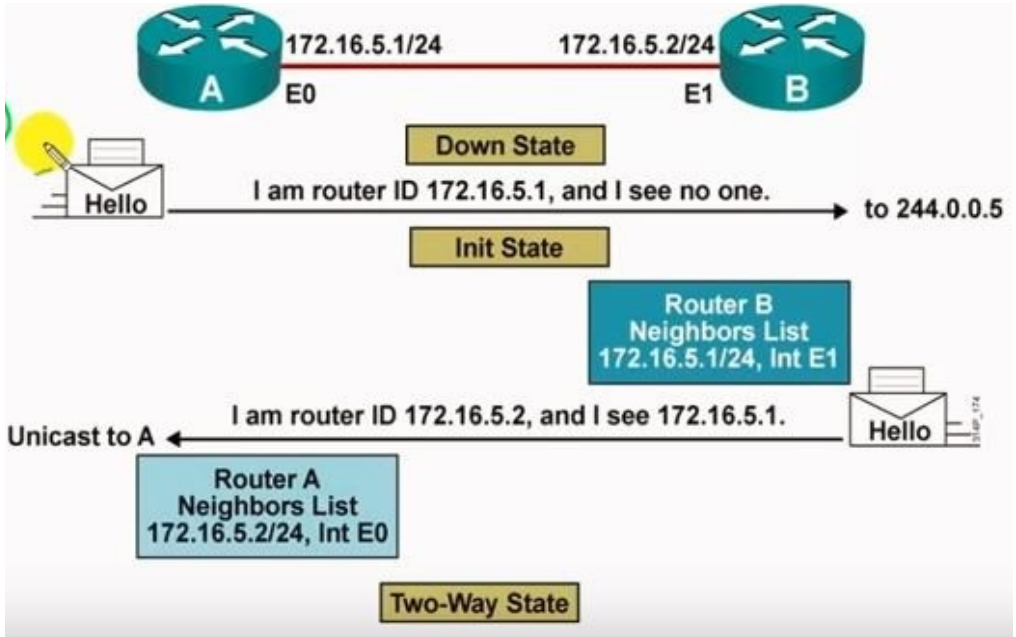
# LINK-STATE PROTOCOL – OPEN SHORTEST PATH FIRST(OSPF) – HOW IT WORKS

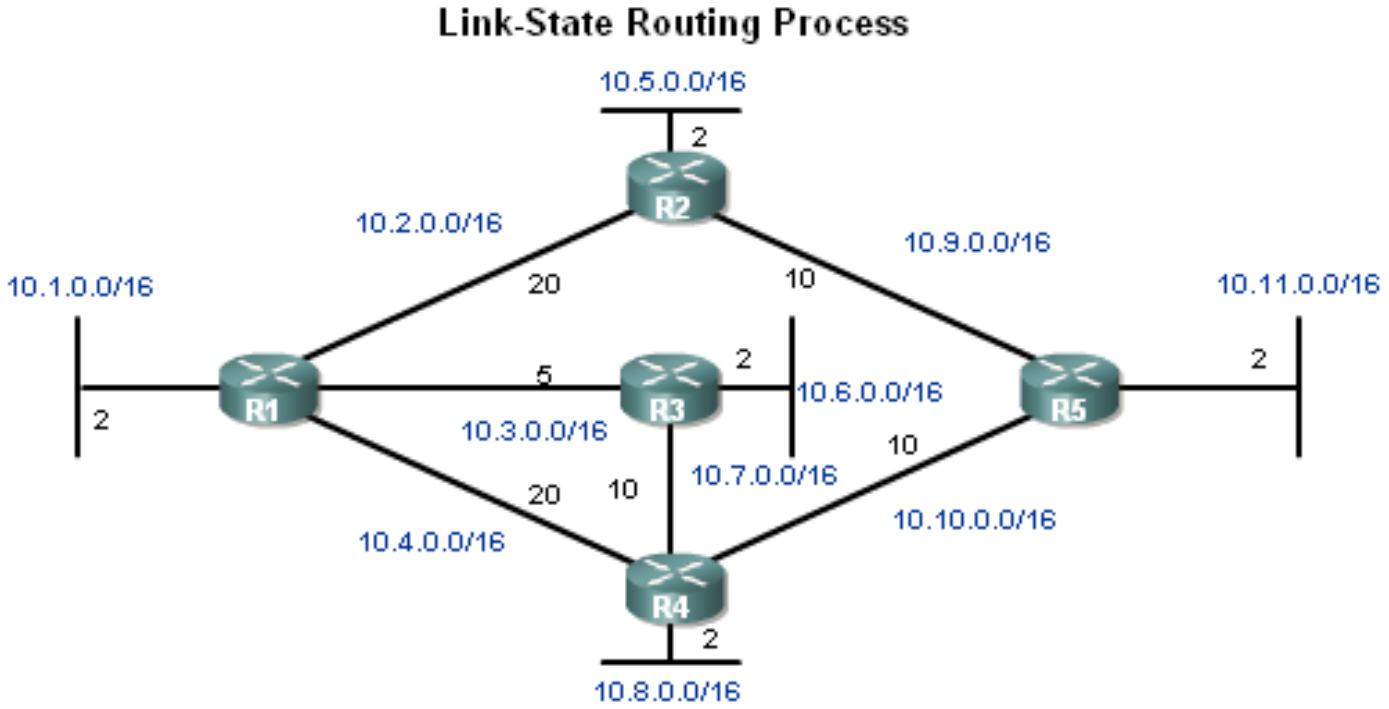
## Step 2 – Sending Hello Packets to Neighbors (period: 10s)

- Link state routing protocols use a **hello protocol**

  Purpose of a hello protocol:

  - To discover neighbors (that use the same link state routing protocol) on its link



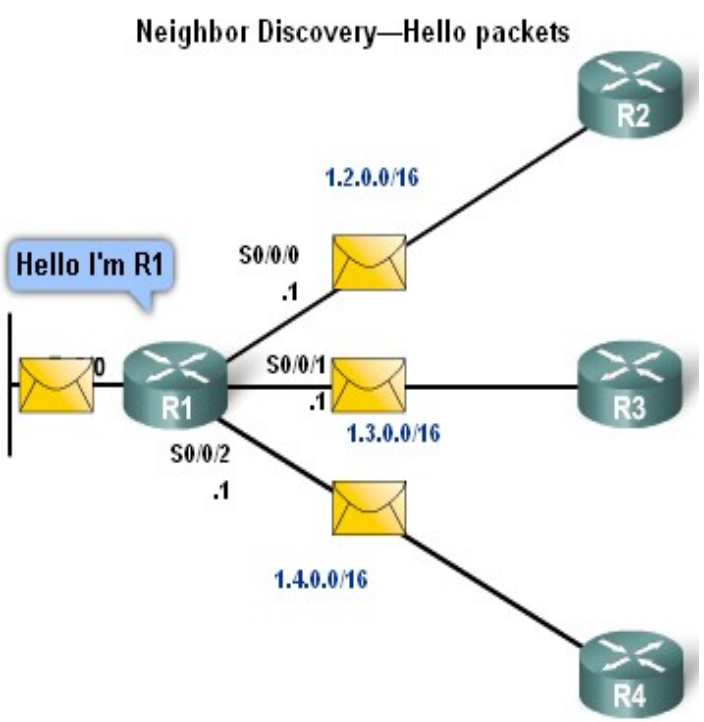



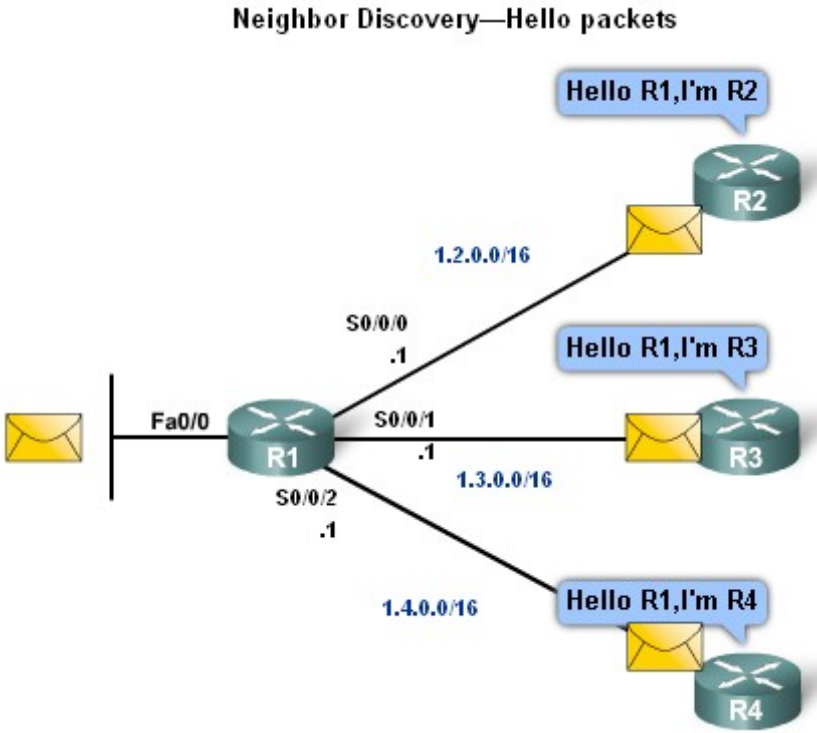

1. Each router learns about each of its own directly connected networks.
2. Each router is responsible for "saying hello" to its neighbors on directly connected networks.

# LINK-STATE PROTOCOL – OPEN SHORTEST PATH FIRST(OSPF) – HOW IT WORKS

## Step 2 – Sending Hello Packets to Neighbors

- Connected interfaces that are using the same link state routing protocols will exchange hello packets.
- Once routers learn it has neighbors they form an adjacency
  - 2 adjacent neighbors will exchange hello packets
  - These packets will serve as a keep alive function



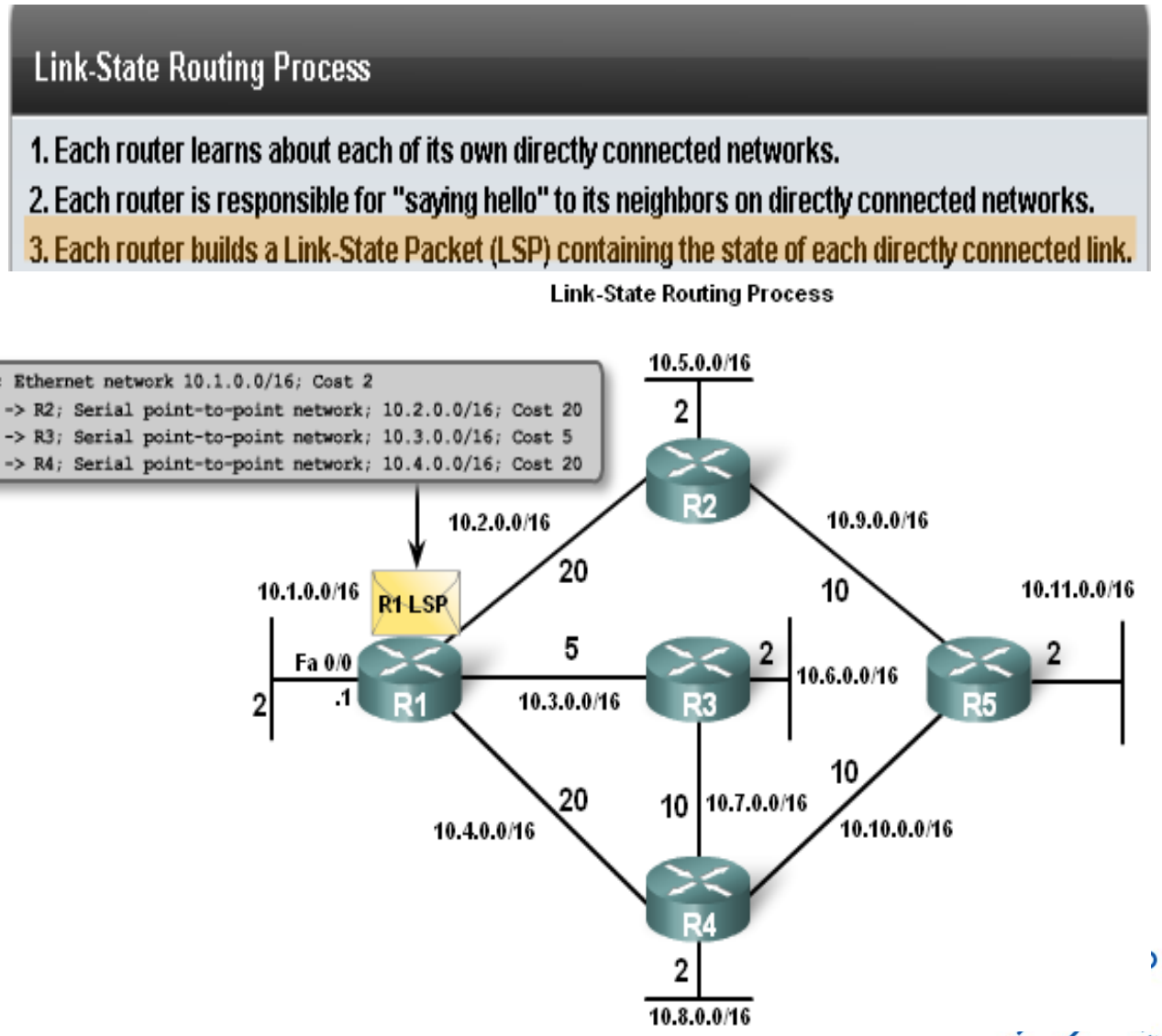
# LINK-STATE PROTOCOL – OPEN SHORTEST PATH FIRST(OSPF) – HOW IT WORKS

## Step 3 – Building the Link State Packet (LSP)

- Contents of LSP:
  - State of each directly connected link
  - Includes information about neighbors such as neighbor ID, link type, & bandwidth.

- A simplified version of the LSPs from R1 is:

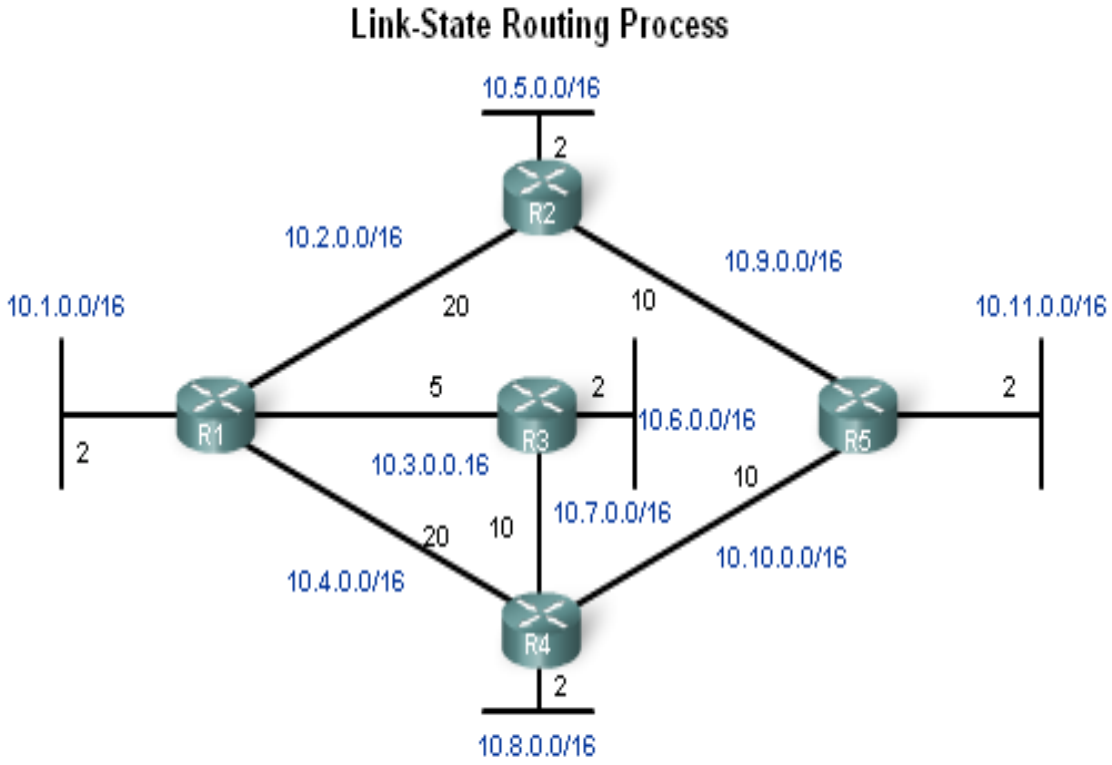
  1. R1; Ethernet network 10.1.0.0/16; Cost 2
  2. R1 -> R2; Serial point-to-point network; 10.2.0.0/16; Cost 20
  3. R1 -> R3; Serial point-to-point network; 10.3.0.0/16; Cost 5
  4. R1 -> R4; Serial point-to-point network; 10.4.0.0/16; Cost 20



# LINK-STATE PROTOCOL – OPEN SHORTEST PATH FIRST(OSPF) – HOW IT WORKS

## Step 4 – Flooding LSPs to Neighbors

- Once LSP are created they are forwarded out to neighbors.
  - -Each router floods its link-state information to all other link-state routers in the routing area.
  - -Whenever a router receives an LSP from a neighboring router, it immediately sends that LSP out all other interfaces except the interface that received the LSP.
  - -This process creates a flooding effect of LSPs from all routers throughout the routing area.



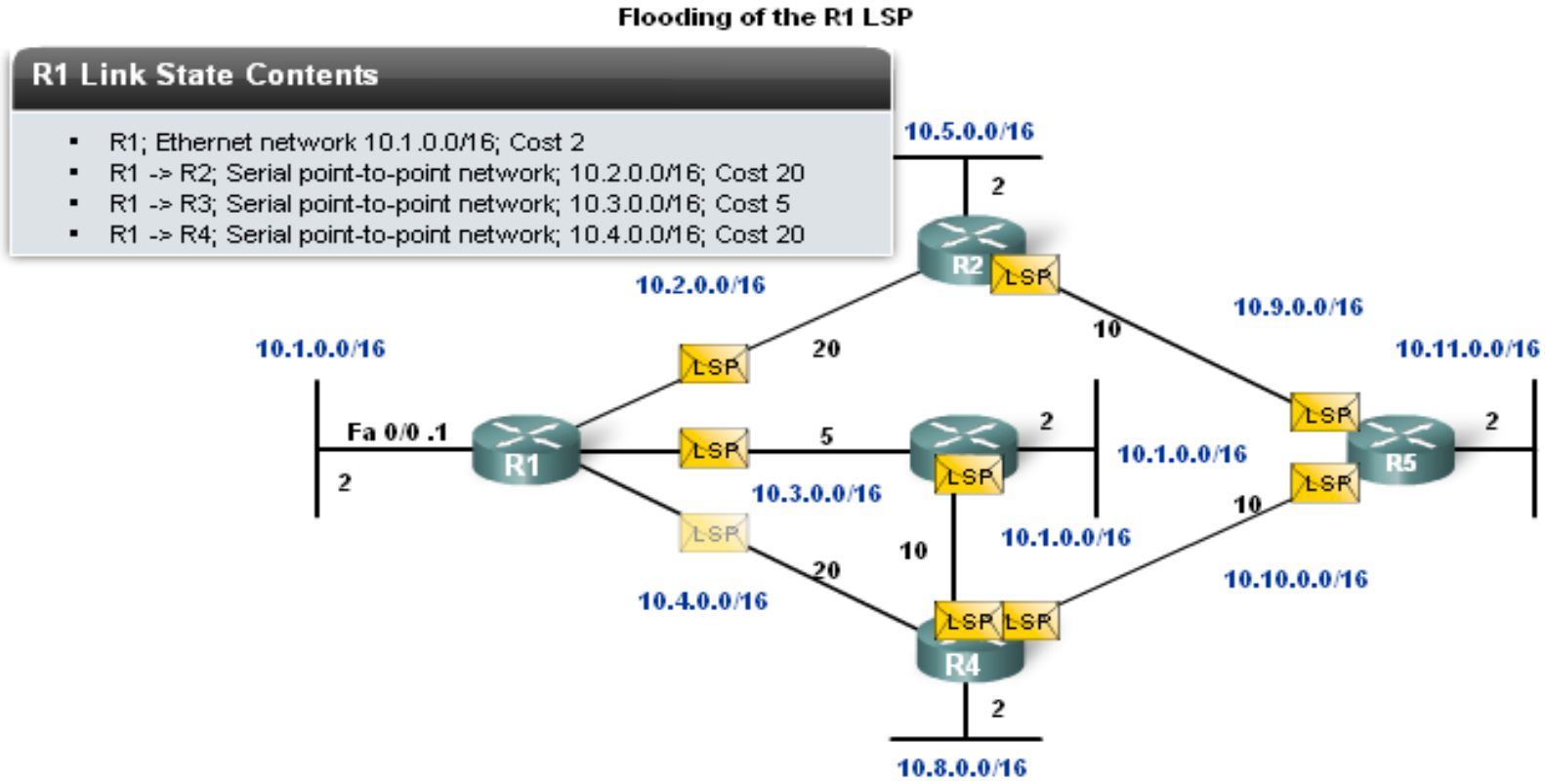

**Link-State Routing Process**

1. Each router learns about each of its own directly connected networks.
2. Each router is responsible for "saying hello" to its neighbors on directly connected networks.
3. Each router builds a Link-State Packet (LSP) containing the state of each directly connected link.
4. Each router floods the LSP to all neighbors, who then store all LSPs received in a database.

# LINK-STATE PROTOCOL – OPEN SHORTEST PATH FIRST(OSPF) – HOW IT WORKS

## Step 4 – Flooding LSPs to Neighbors

- LSPs are sent out under the following conditions
  - Initial router start up or routing process
  - When there is a change in topology
    - including a link going down or coming up, or a neighbor adjacency being established or broken



# LINK-STATE PROTOCOL – OPEN SHORTEST PATH FIRST(OSPF) – HOW IT WORKS

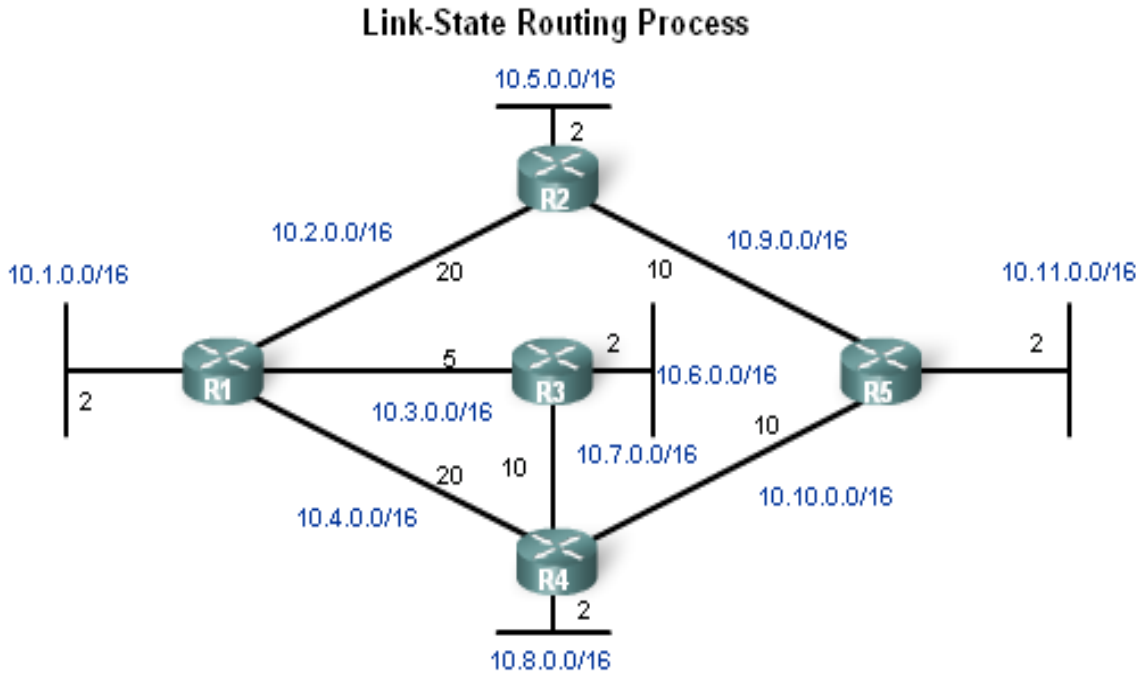
## Step 5 – Constructing a link state data base

- Routers use a database to construct a topology map of the network

  -After each router has propagated its own LSPs using the link-state flooding process, each router will then have an LSP from every link-state router in the routing area.

  -These LSPs are stored in the link-state database.

  -Each router in the routing area can now use the SPF algorithm to construct the SPF trees that you saw earlier.



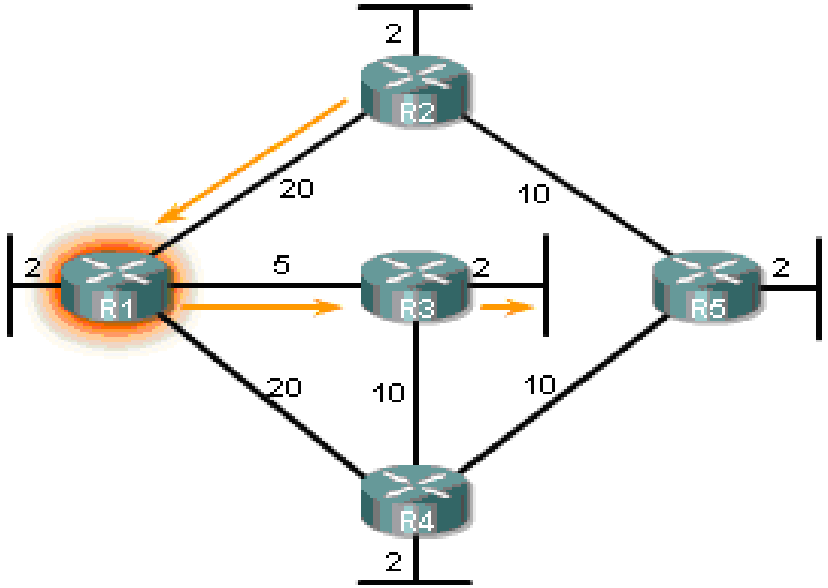

**Link-State Routing Process**

1. Each router learns about each of its own directly connected networks.
2. Each router is responsible for "saying hello" to its neighbors on directly connected networks.
3. Each router builds a Link-State Packet (LSP) containing the state of each directly connected link.
4. Each router floods the LSP to all neighbors, who then store all LSPs received in a database.
5. Each router uses the database to construct a complete map of the topology and computes the best path to each destination network.

# LINK-STATE PROTOCOL – OPEN SHORTEST PATH FIRST(OSPF) – HOW IT WORKS

## Step 5 – Constructing a link state data base

Router R1 has learned the link-state information for each router in its routing area.




**R1 Link-State Database**

R1s Link-State DatabaseLSPs from R2:
- Connected to neighbor R1 on network 10.2.0.0/16, cost of 20
- Connected to neighbor R5 on network 10.9.0.0/16, cost of 10
- Has a network 10.5.0.0/16, cost of 2

LSPs from R3:
- Connected to neighbor R1 on network 10.3.0.0/16, cost of 5
- Connected to neighbor R4 on network 10.7.0.0/16, cost of 10
- Has a network 10.6.0.0/16, cost of 2

LSPs from R4:
- Connected to neighbor R1 on network 10.4.0.0/16, cost of 20
- Connected to neighbor R3 on network 10.7.0.0/16, cost of 10
- Connected to neighbor R5 on network 10.10.0.0/16, cost of 10
- Has a network 10.8.0.0/16, cost of 2

LSPs from R5:
- Connected to neighbor R2 on network 10.9.0.0/16, cost of 10
- Connected to neighbor R4 on network 10.10.0.0/16, cost of 10
- Has a network 10.11.0.0/16, cost of 2

R1 Link-states:
- Connected to neighbor R2 on network 10.2.0.0/16, cost of 20
- Connected to neighbor R3 on network 10.3.0.0/16, cost of 5
- Connected to neighbor R4 on network 10.4.0.0/16, cost of 20

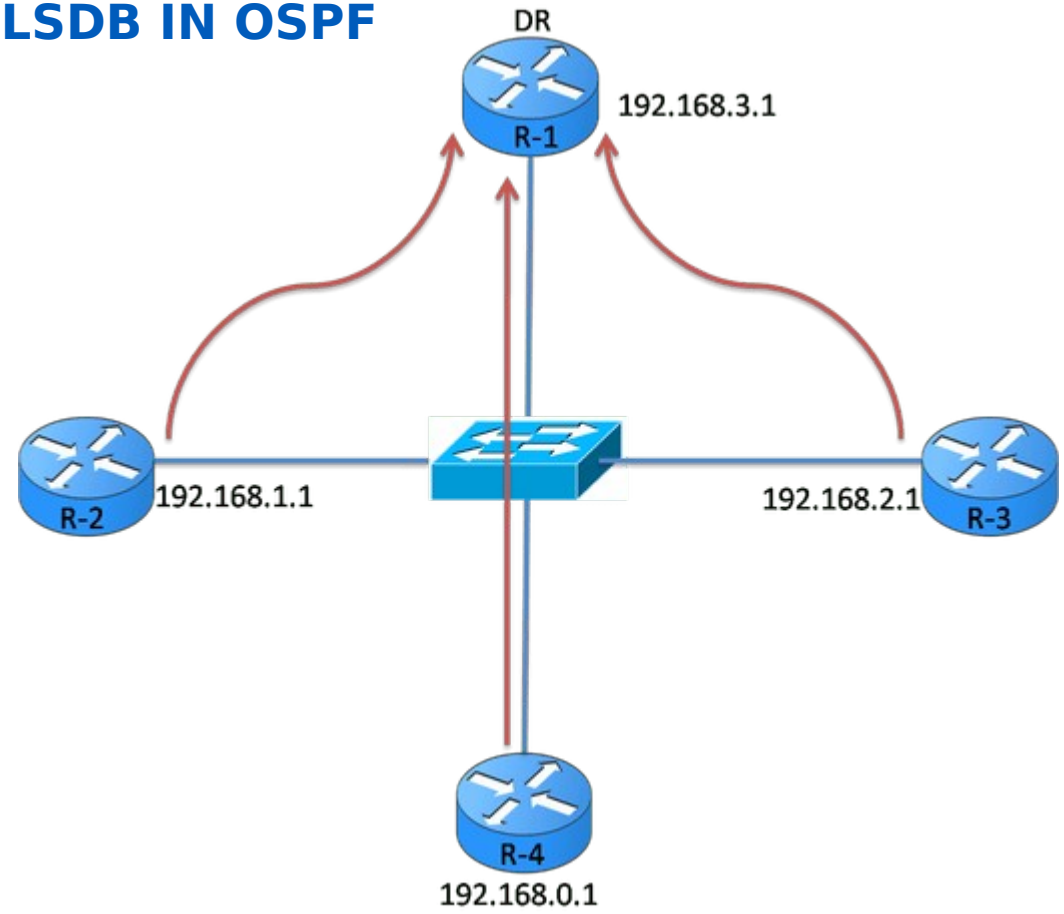
With a complete link-state database, R1 can now use the database and the shortest path first (SPF) algorithm to calculate the preferred path or shortest path to each network.

| Destination | Shortest Path | Cost |
|---|---|---|
| R2 LAN | R1 to R2 | 22 |
| R3 LAN | R1 to R3 | 7 |
| R4 LAN | R1 to R3 to R4 | 17 |
| R5 LAN | R1 to R3 to R4 to R5 | 27 |

# ADVANCED OSPF – TWO APPROACHES TO EXCHANGE LSDB IN OSPF

There are 2 important approaches to exchange LSDB in OSPF:

- Point – to – Point (as presented above)
- Broadcast Multi-access → Need to elect **designated router (DR)** and elect **Backup designated router** (**BDR**), other Routers called **DRother**



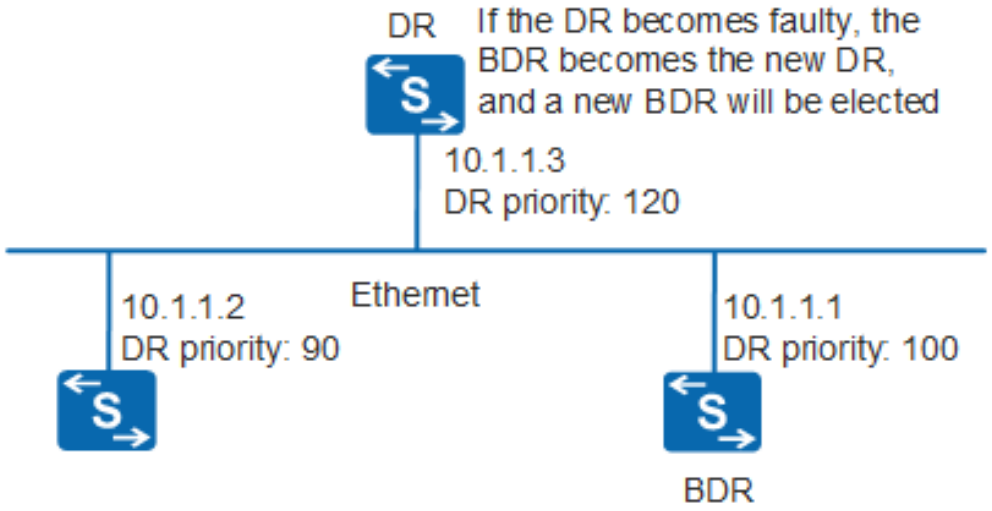

- DR is to be the collection & distribution point for LSAs sent and received. The DR is responsible for forwarding the LSAs from DROthers to all other routers using **multicast address 224.0.0.5**

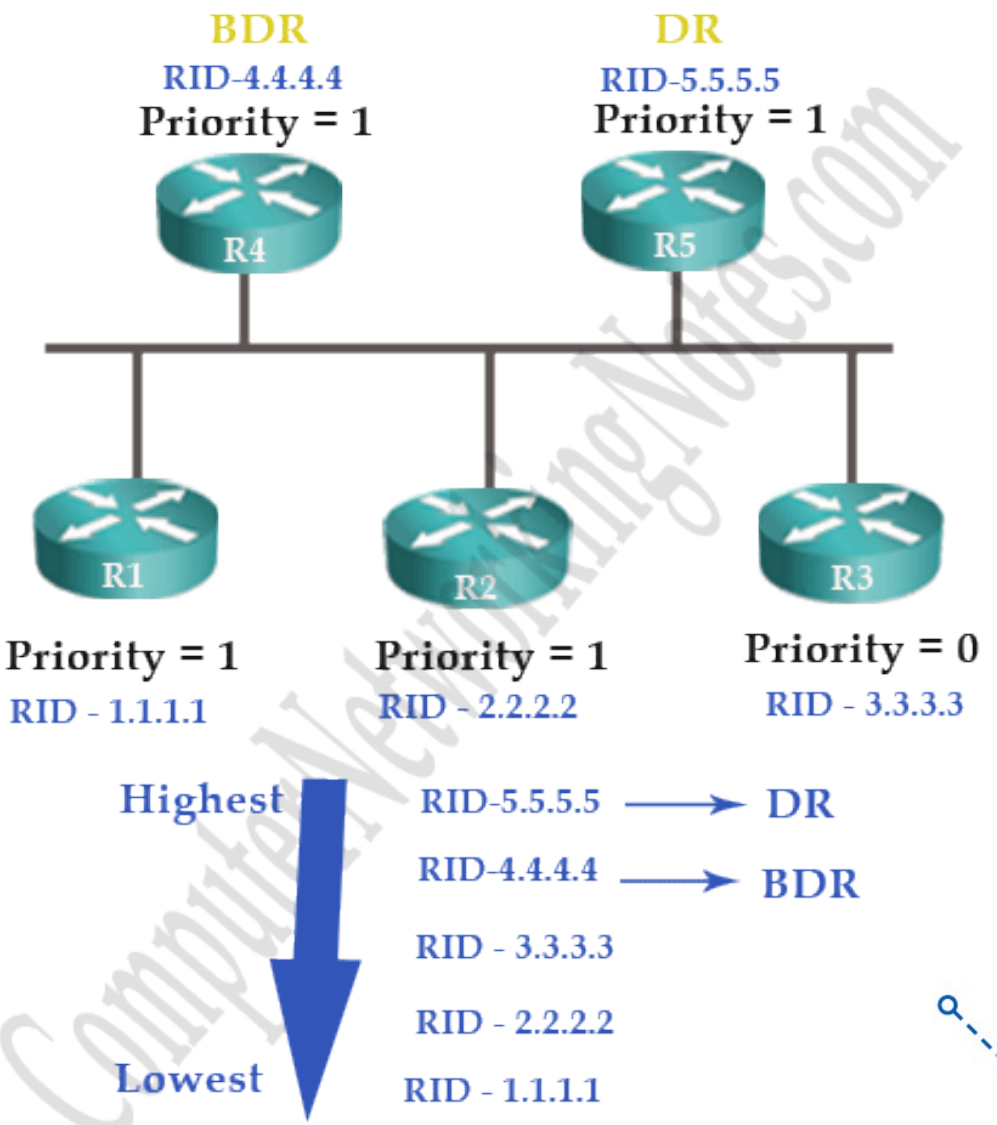
- BDR is used to backup for BR

- DROthers only form full adjacencies with the DR & BDR in the network. They send information to DR using **multicast address 224.0.0.6**

# CRITERIA TO ELECT DR/BDR ROUTERS

- On each interface connected to multi-access environment has a **priority** parameter (0 – 255). Default priority = 1
- Router has highest priority → DR, then BDR…
- If there are routers have the same priority → Use Router ID (RID)



# HOW TO CALCULATE METRIC IN OSPF

- Metric = cost → Cost =
- Metric = cost → Cost = $\sum cost$

→ Cumulative cost = *Sum of all outgoing interfaces cost in route*

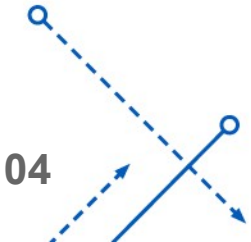
→ **Cumulative cost = *Sum of all outgoing interfaces cost in route***

→ Best route for routing table = *Route which has the lowest cumulative cost*

→ **Best route for routing table = *Route which has the lowest cumulative cost***

- $Cost = \frac{10^8}{BW(bps)}$

| Bandwidth | OSPF Cost |
|---|---|
| 100 Gbps | 1 |
| 40 Gbps | 1 |
| 10 Gbps | 1 |
| 1 Gbps | 1 |
| 100 Mbps | 1 |
| 10 Mbps | 10 |
| 1.544 Mbps | 64 |
| 768 Kbps | 133 |
| 384 Kbps | 266 |
| 128 Kbps | 781 |



# COMPARISONS WITH DISTANCE VECTOR

## Advantages of a Link-State Routing Protocol

| Routing protocol | Builds Topological map | Router can independently determine the shortest path to every network. | Convergence | Event driven routing updates | Use of LSP |
|---|---|---|---|---|---|
| Distance vector | No | No | Slow | Generally No | No |
| Link State | Yes | Yes | Fast | Generally Yes | Yes |

# DRAWBACKS & REQUIREMENTS OF LINK-STATE PROTOCOL

* **Drawback**

- Initial discovery may cause flooding.

- Link-state routing is memory- and processor-intensive.

* **Requirements for using**

Memory requirements

*Typically link state routing protocols use more memory*

Processing Requirements

*More CPU processing is required of link state routing protocols*

Bandwidth Requirements

*Initial startup of link state routing protocols can consume lots of bandwidth*

*This should only occur during initial startup of routers, but can also be an issue on unstable networks.*