What you feed your mind with will rule your life.

Routing Information Protocol

Two main algorithm types

- Distance vector algorithms:
  - ✓ every station must broadcast its *global* routing tables, but only to its *neighbors*.
  - ✓ Converges slowly after topology change
- Link state algorithms:
  - ✓ every station must broadcast its *local* information to all the network's junctions
  - ✓ accurate and reliable

Routing Information Protocol- RIP

- Common distance vector protocol
- Useful for small subnets
- Easy to install
- Uses Bellman-Ford Algorithm
- Distributed
- RIP v1 RFC 1058 (reverse engineering)
- RIP v2 RFC 2453

Introduction

- This algorithm has been used for routing computations in computer networks since the early days of the ARPANET
- RIP is intended for use within the IP-based Internet, though provision is made to accept other protocols as well
- It became a de facto standard for exchange of routing information among gateways and hosts
- RIP messages are carried in UDP datagram

Distance vector algorithm: Bellman-Ford

Each node:

- Estimates distance to ***every*** other node in the network (*distance vectors)*
- Transmit the information to its neighbours (& receives similar ***distance vectors*** from its neighbours)
- Updates its estimation of best path (*minimum cost*) to each destination
- Estimates the next hop for this destination

Advantages:

- adapts to traffic changes and link failures
- suitable for networks with multiple administrative entities

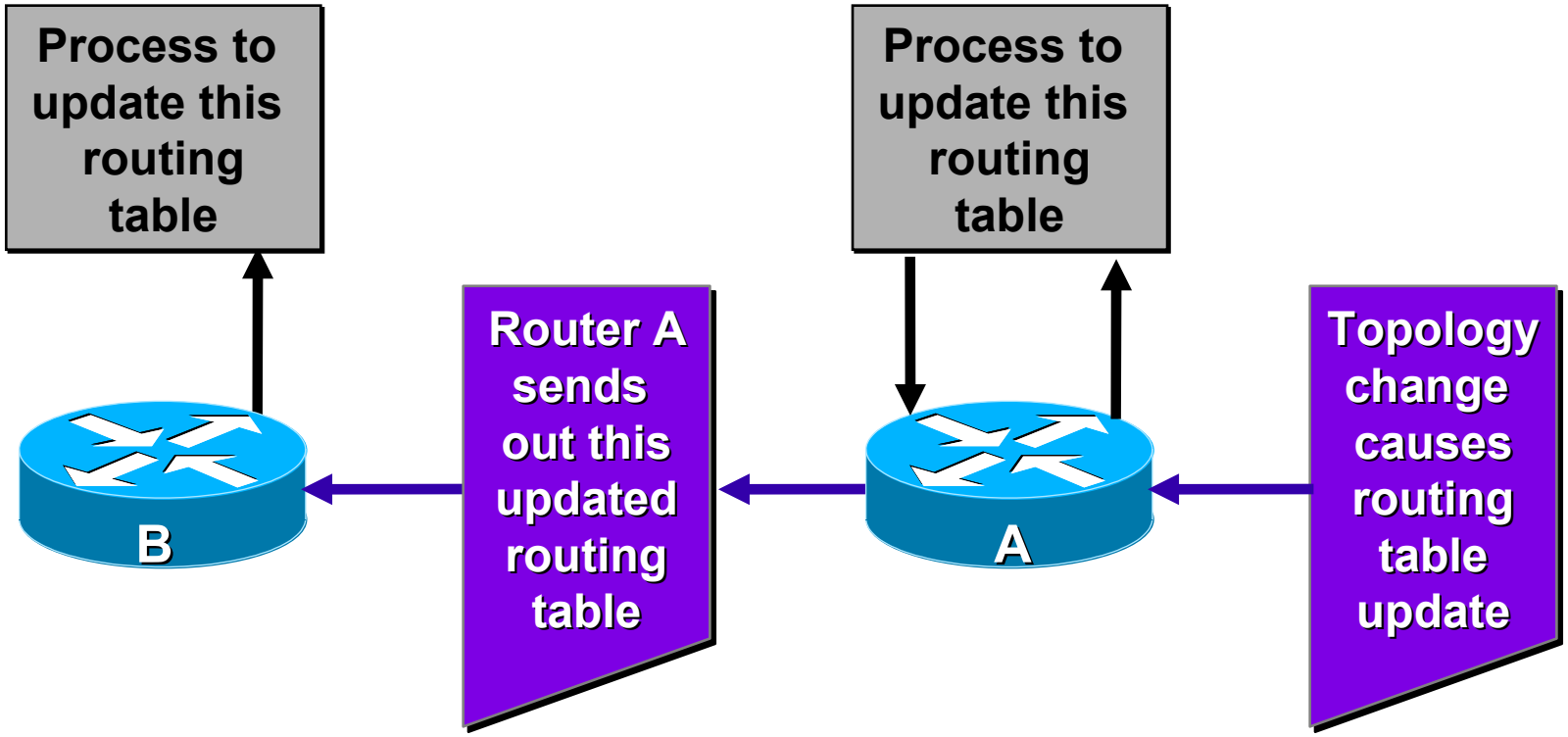
Problem:

- Run into problems when links go down or come up

Distance vector algorithm: Bellman-Ford

- Each router maintains one entry for each destination, estimated distance, outgoing line
- Distance in terms of hops, time delay, queue
- Starts with information of neighbours
- Propagates as time passes
- Will take at least ‘n’ cycles if it is ‘n’ level tree

Distance Vector Topology Changes



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

Routing Table Format

Destination IP address	Distance	Outgoing port
11.1.0.0	0	E1
11.2.0.0	2	S0
11.3.0.0	7	S3

RIPv1 packet format

- **Command**—Indicates whether the packet is a ***request*** or a ***response***
- **Version**—Specifies the RIP version used
- **Unused**—Has a value set to zero
- **Address-family identifier (AFI)**—Specifies the address family used, set to 2 for IP
- **IP address**—Specifies the IP address for the entry
- **Metric**—Indicates how many internetwork hops (routers) have been traversed in the trip to the destination

RIPv1

- Upto 25 routes (20 bytes each) can be advertised in a RIP message
- Multiple RIP messages required to send a large routing table

RIP Operation

- Initialisation
  - On bootstarp, sends request on all connected interface
  - On UDP destination port 520
  - Address family set to ‘0’
  - Metric set to 16
  - This is a request for complete routing table

RIP Operation

- Request received
  - Sends entire table if so requested
  - Otherwise, send requested info
  - If route is known, send the metric value
  - If unknown route, set the metric as 16 (infinity)
- Response received
  - Validate response and update routing table

RIPv1 Updates

- Regular Routing Updates
  - Based on ***refresh timer***
  - Normally every 30 seconds, with some random value to avoid congestion
  - routing table is sent to all the neighbours
  - Does not take care of changes between two refresh events
  - Slower convergence

RIP v2 packet format

- **Command**—Indicates whether the packet is a request or a response
- **Version**—Specifies the RIP version used
- **Routing domain**— identifier of routing domain to which packet belong
- **Address-family identifier (AFI)**—Specifies the address family used, can carry routing information of multiple protocols
- **Route tag**—Provides a method for distinguishing between internal routes (learned by RIP) and external routes (learned from other protocols). Carries autonomous system number of domain
- **IP address**—Specifies the IP address for the entry
- **Next hop**—Indicates the IP address of the next hop to which packets for the entry should be forwarded. ‘0’ means local loop
- **Metric**—Indicates how many internetwork hops (routers) have been traversed in the trip to the destination

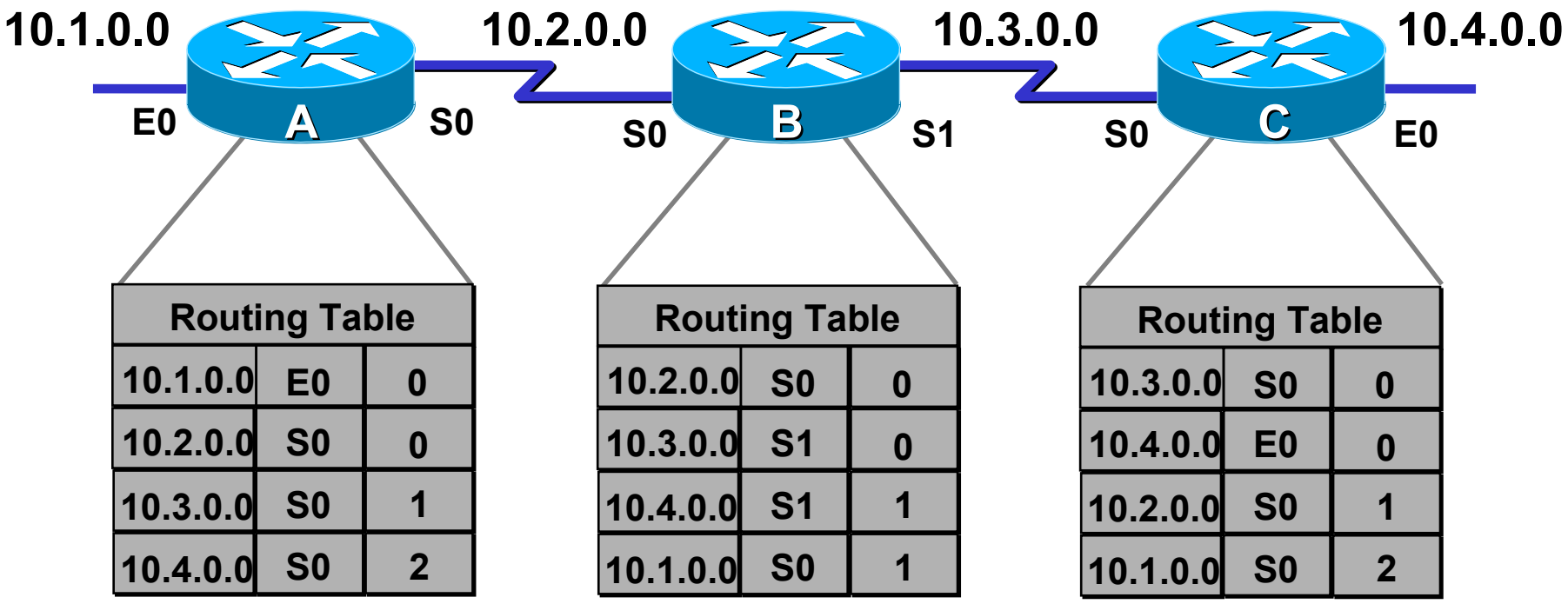
RIPv2

- If the AFI for the first entry in the message is 0xFFFF, the remainder of the entry contains authentication information.
- Currently, the only authentication type is simple password, in clear text.

RIPv2 Updates

- Regular Routing Updates
  - Every 30 seconds routing table is sent to all the neighbours
- Triggered updates
  - Sent whenever the metric for a route changes
  - Only the changed entries are sent

Problem: Routing Loops



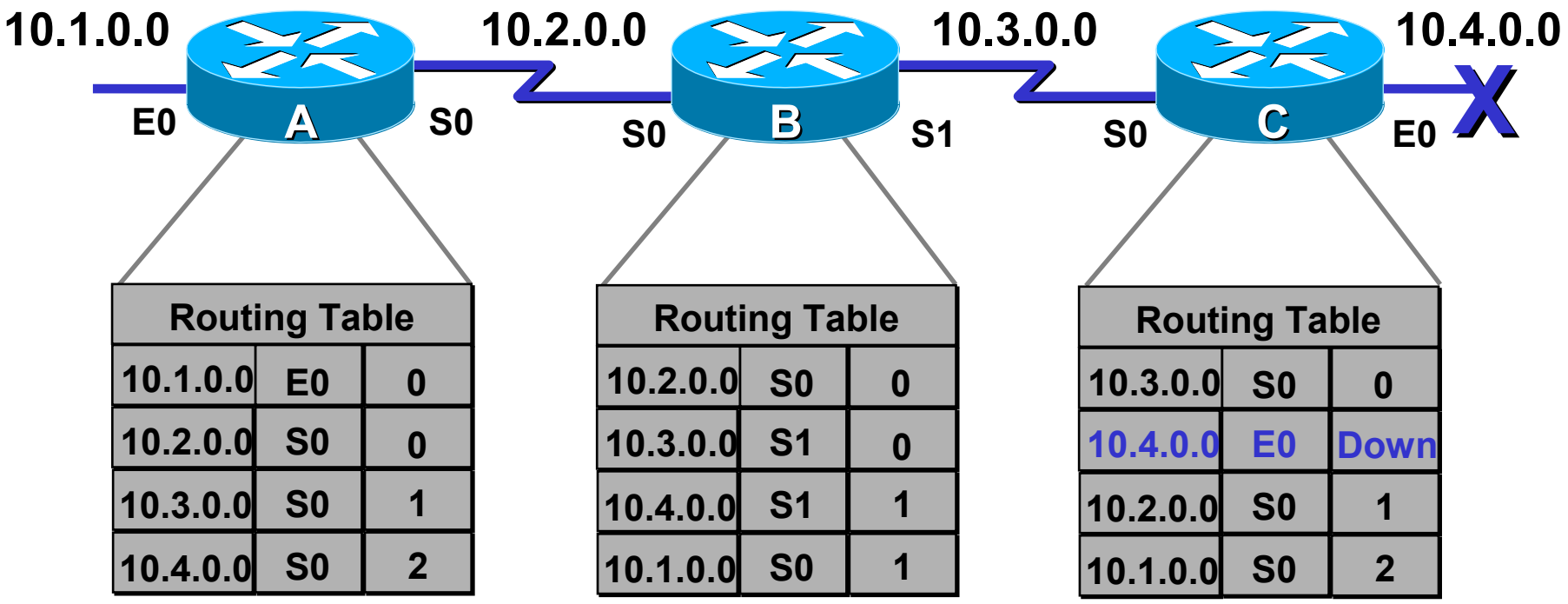
Routing Table		
10.1.0.0	E0	0
10.2.0.0	S0	0
10.3.0.0	S0	1
10.4.0.0	S0	2

Routing Table		
10.2.0.0	S0	0
10.3.0.0	S1	0
10.4.0.0	S1	1
10.1.0.0	S0	1

Routing Table		
10.3.0.0	S0	0
10.4.0.0	E0	0
10.2.0.0	S0	1
10.1.0.0	S0	2

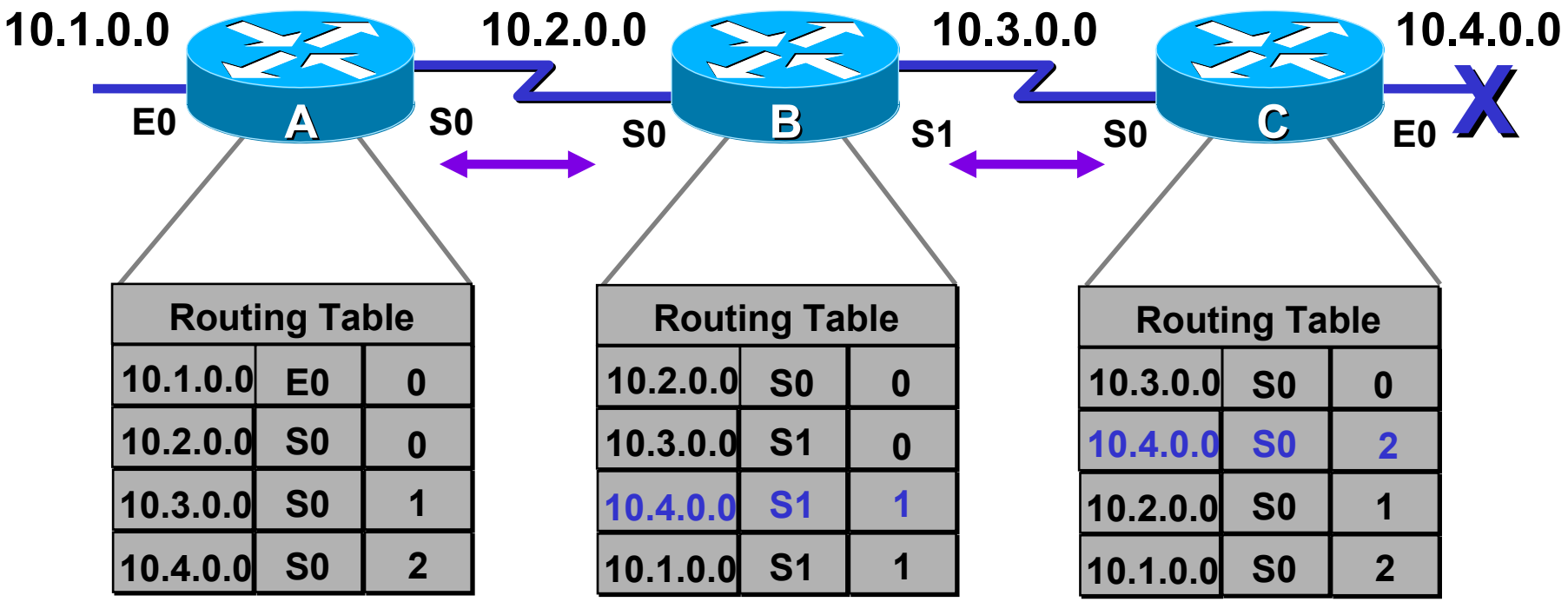
- Each node maintains the distance from itself to each possible destination network

Problem: Routing Loops



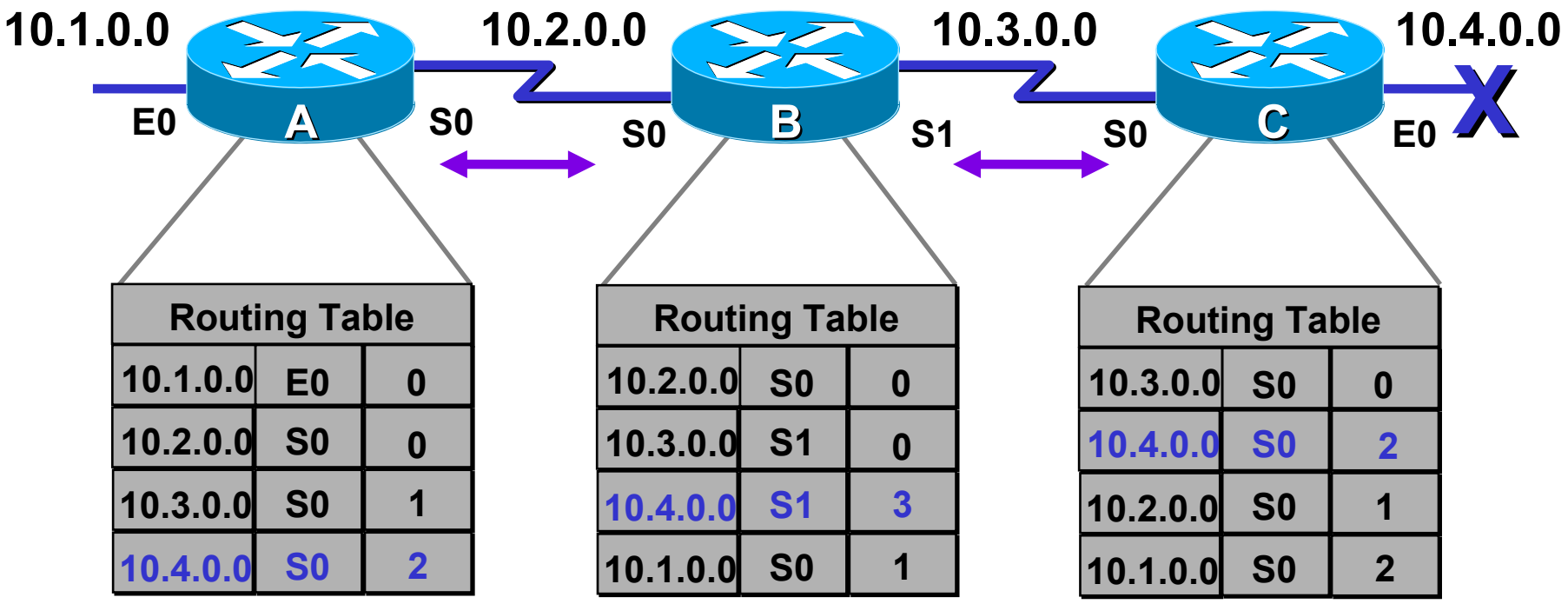
- Slow convergence produces inconsistent routing
- Negative route entry is not sent

Problem: Routing Loops



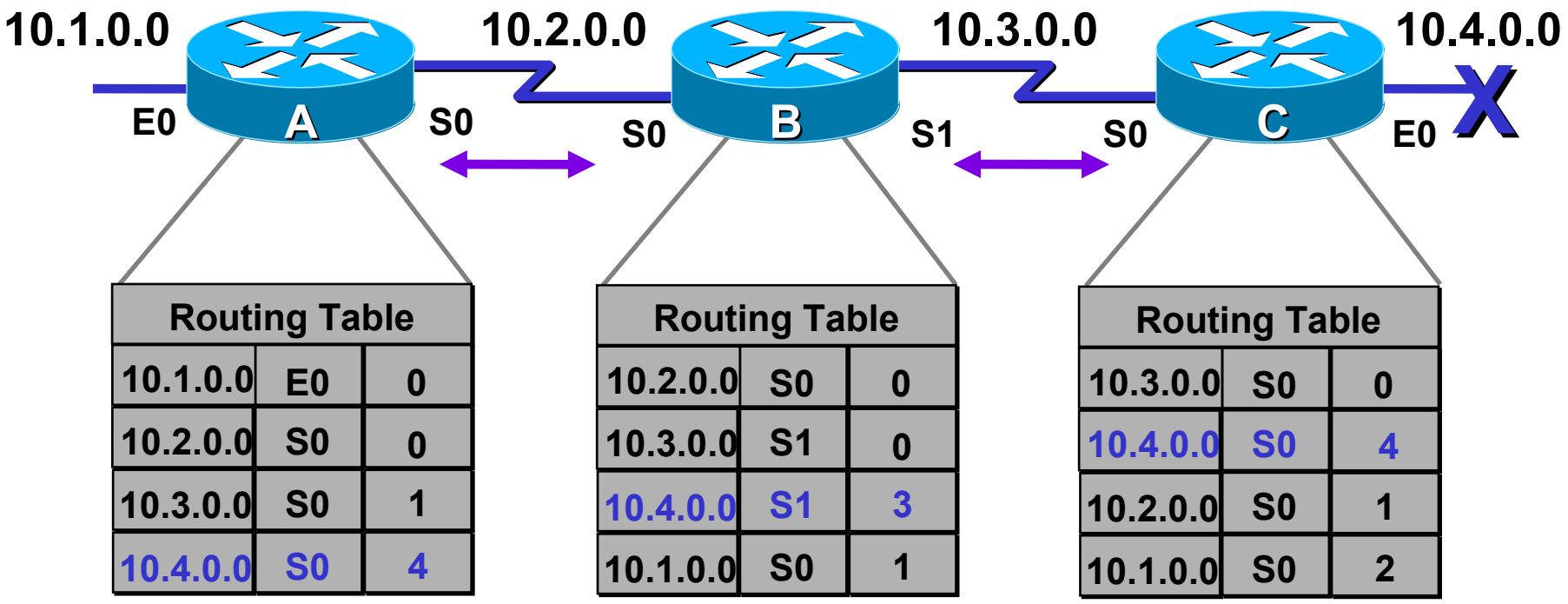
- Router C concludes that the best path to network 10.4.0.0 is through Router B

Problem: Routing Loops



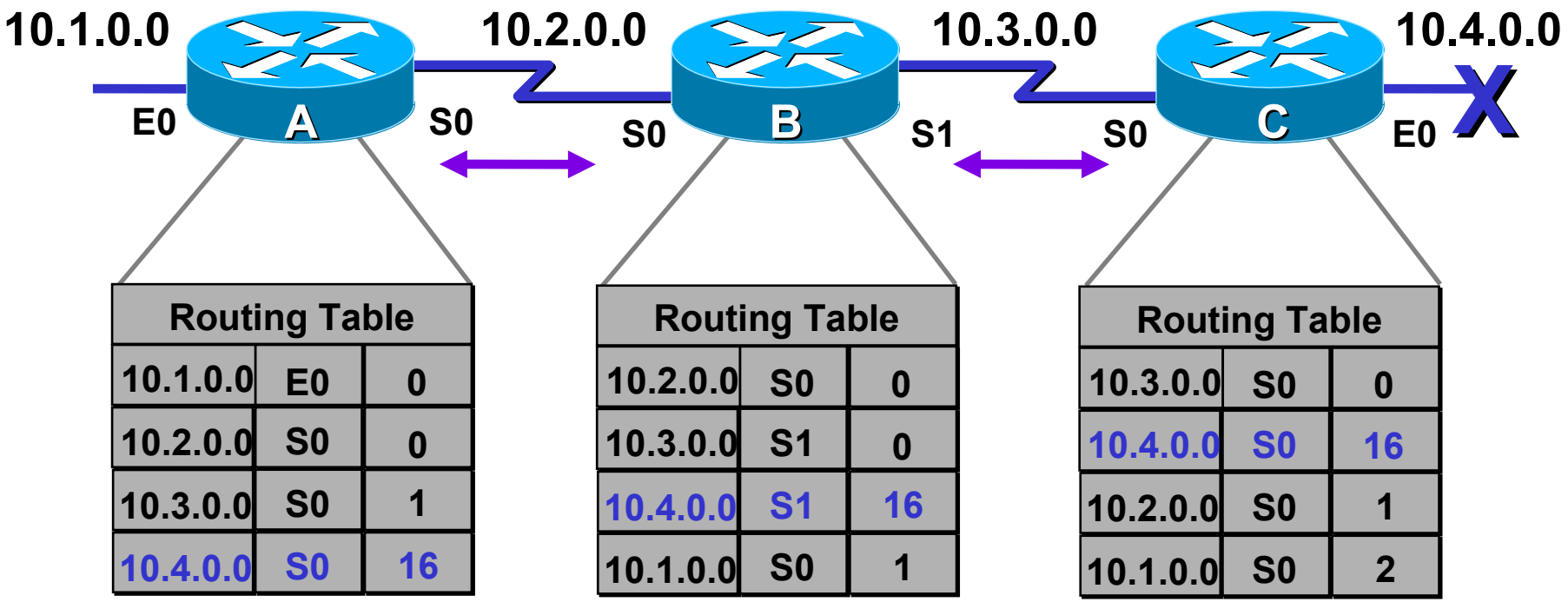
- Router A updates its table to reflect the new but erroneous hop count

Symptom: Counting to Infinity



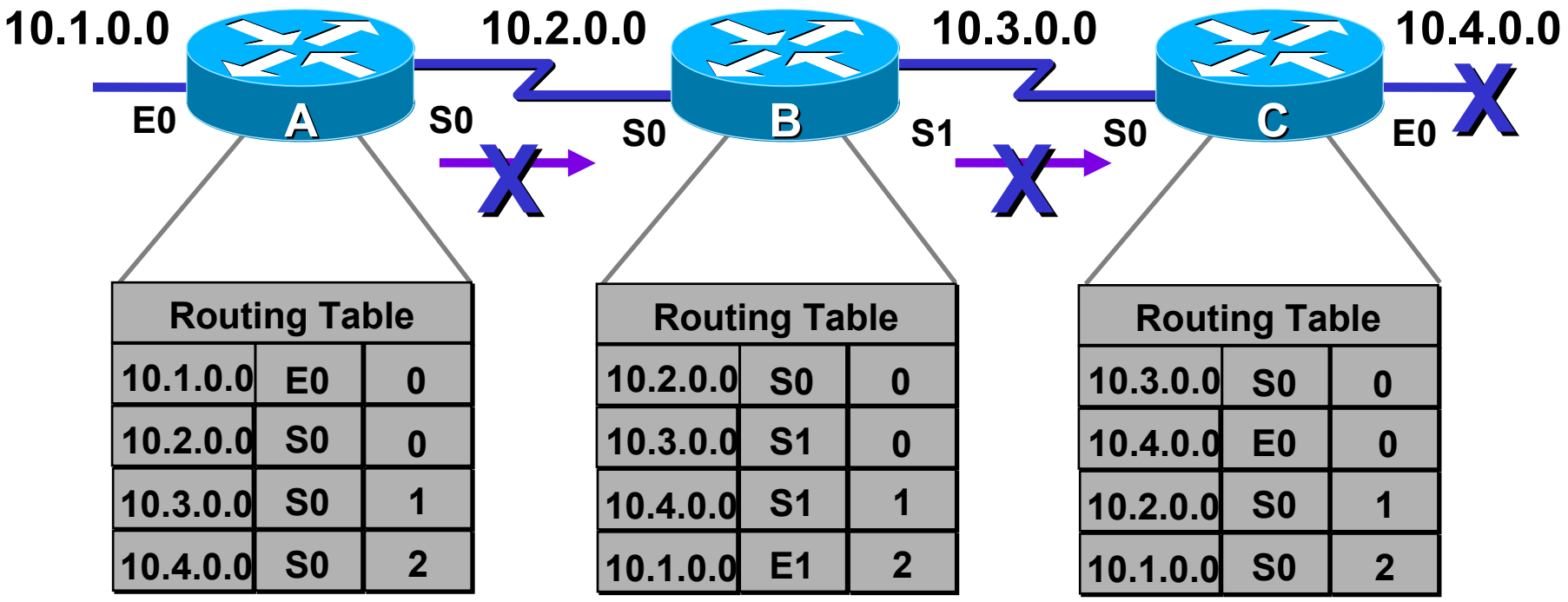
- Packets for network 10.4.0.0 bounce between routers A, B, and C, incrementing hop count

Solution: Defining a Maximum



- Define a limit on the number of hops to prevent infinite loops.
- RIP has got the maximum 15 hops and hence network can be maximum 15 hops end to end on any segment, >15 is considered to be infinite.

Solution: Split Horizon

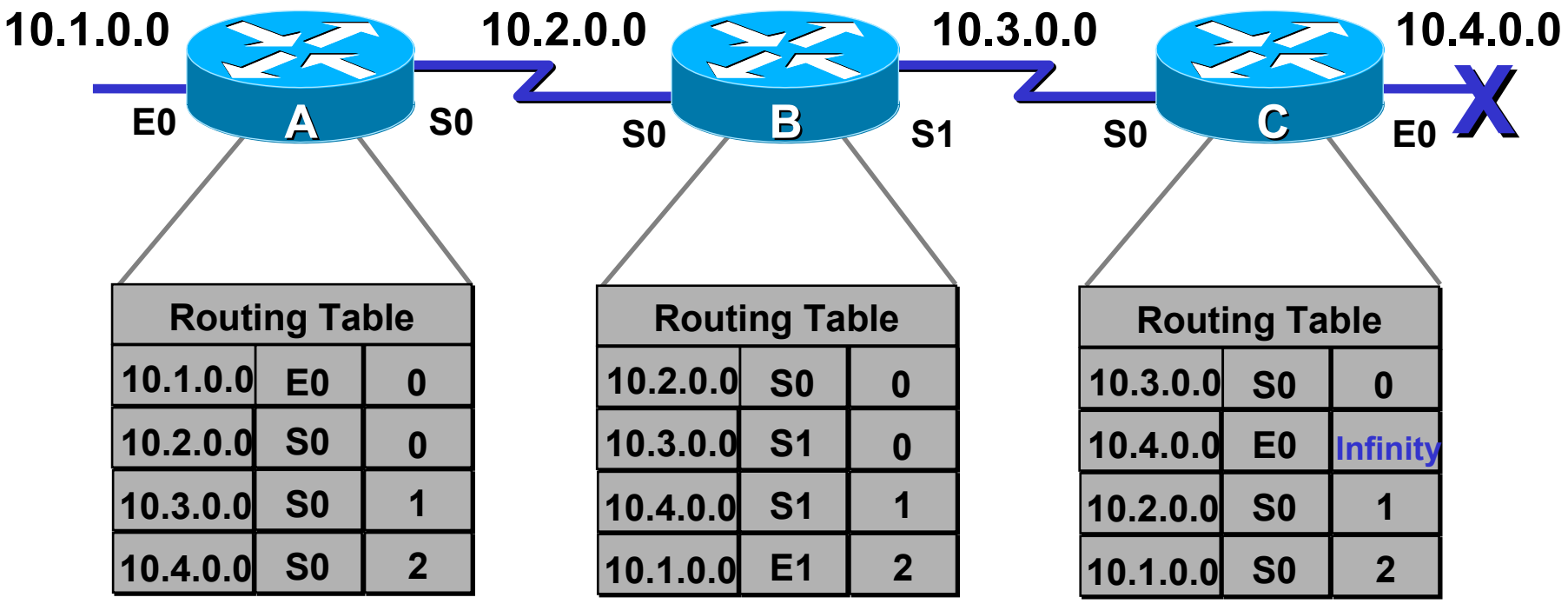


- It is never useful to send information about a route back in the direction from which the original packet came

Routing Table Format

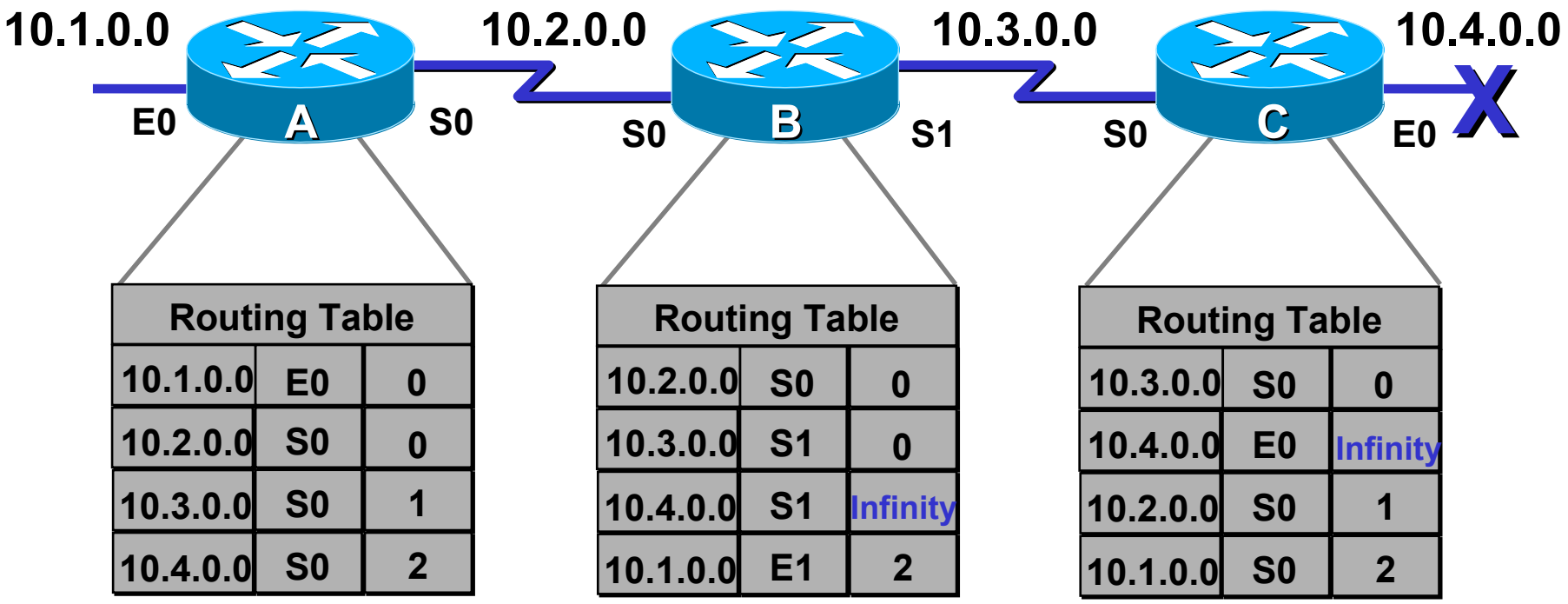
Destination IP address	Distance	Outgoing port/route learnt from
11.1.0.0	0	E1
11.2.0.0	2	S0
11.3.0.0	7	S3

Solution: Poison Reverse



- Routers set the distance to infinity if the destination is routed on that link
- Send negative route entry

Solution: Poison Reverse



- Routers set the distance to infinity if the destination is routed on that link

Timers

- Routing Update timer
- Route Timeout timer
- Route Flush timer

Few Observations

- Does not make use of bandwidth information
- RIPv1 supports class based routing
- RIPv2 supports CIDR (classless inter domain routing)

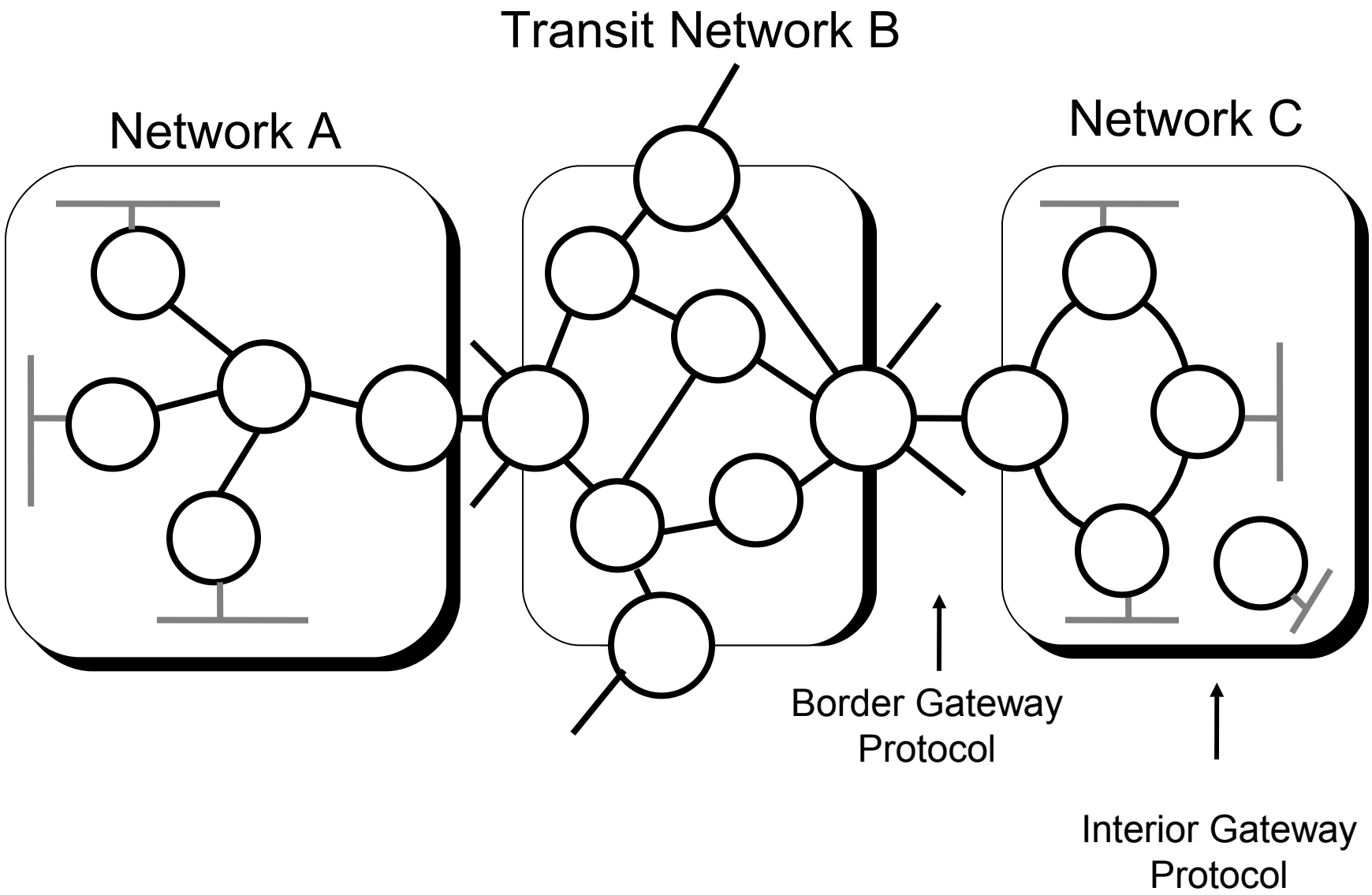
Few Observations

- Can define static routes
  - This has priority over other learnt/calculated routes
- Can locally store collected information from neighbouring routers
  - Multiple tables
  - Helps in calculating alternate route immediately
  - Lot of memory overhead
- Can disable/enable routing protocol on specific ports
- May have entry for subnet mask in routing table

Few Observations

- Multiple routing tables
  - In the router primary memory of the router
    - ❖ Calculated route
  - In the communication processor (ASIC/FPGA) cache/buffer
    - ❖ Subset of calculated route
    - ❖ Based on usage
  - In the flash memory
    - ❖ Static route

Routing


Transit Network B
Network A
Network C
Border Gateway Protocol
Interior Gateway Protocol