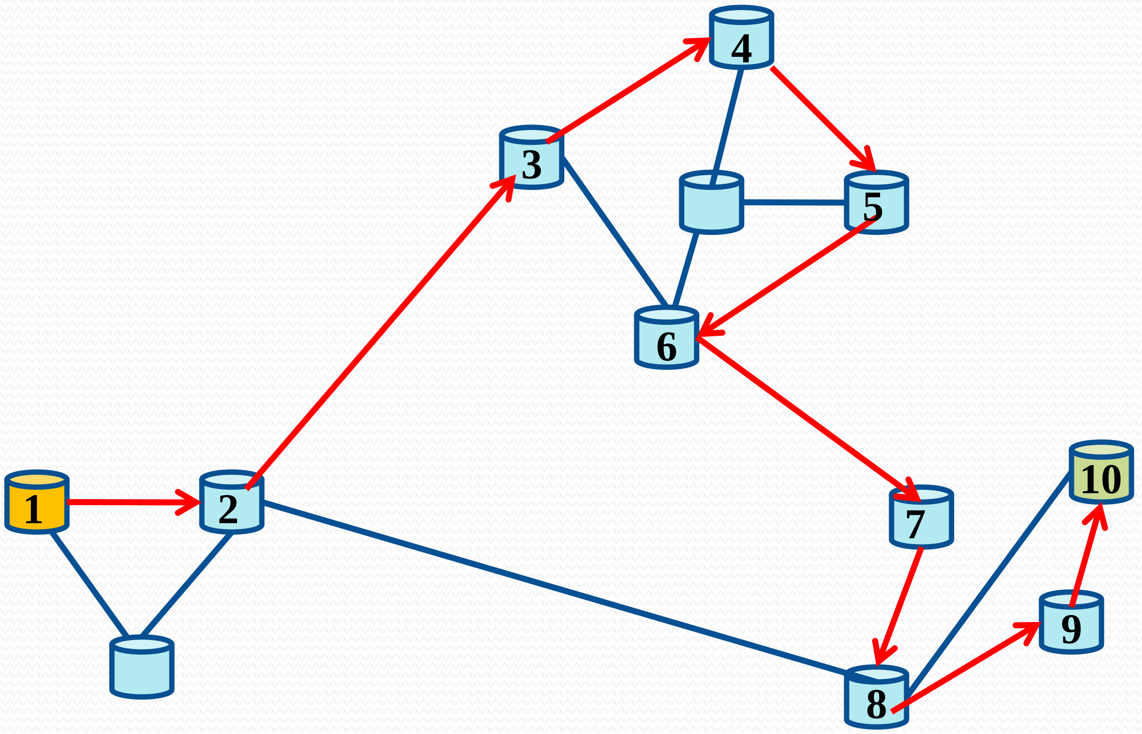
PI-Grau
(Internet Protocols)

José M. Barceló Ordinas
Departamento de Arquitectura de Computadores
(UPC)

- **Topic 3: Intra-domain Routing.**
  - Objectives
    - Introduce basic **intra-domain routing** concepts and Link State protocols
    - Understand **flooding** techniques
    - Understand **Dijkstra (minimum cost)** algorithms
    - Learn the basic components of **OSPF** as intra-domain routing protocol

- **Routing**
  - Process of selecting **paths** (≡ **routes**) along a network of nodes
  - **Path:** a walk without cycles



- **Routing strategies (assume N nodes):**
  - **Source Routing**
    - source keeps a cache with the whole path towards each destination
    - depending on cache size → flood a message if the destination is not in the cache
    - If cache holds paths towards all destination → at least once each source has to flood a message to learn the path → O(N) messages
  - **Shortest Path Routing** (distance-vector or link state)
    - source finds a next neighbor towards each destination
    - needs routing table entries of the order of O(N)
  - **Delivery semantic**: defines the set of nodes that will receive the packet. Routing protocols aim to the following delivery semantics:
    - Unicast, multicast, broadcast, anycast, geocast

- **Current Internet**:
  - N~millions → is not practical either Source Routing or Shortest Path Routing
  - Solution → hierarchical routing.
- **Hierarchical Routing:**
  - Organizations designs networks to provide services
  - **Intra-domain routing**: those routing mechanisms that are internally run by any organization.
    - Allow to construct internal paths from any to any internal node
  - **Inter-domain routing**: those routing mechanisms that connects different organizations
    - Allow to construct routes from any-to-any organization

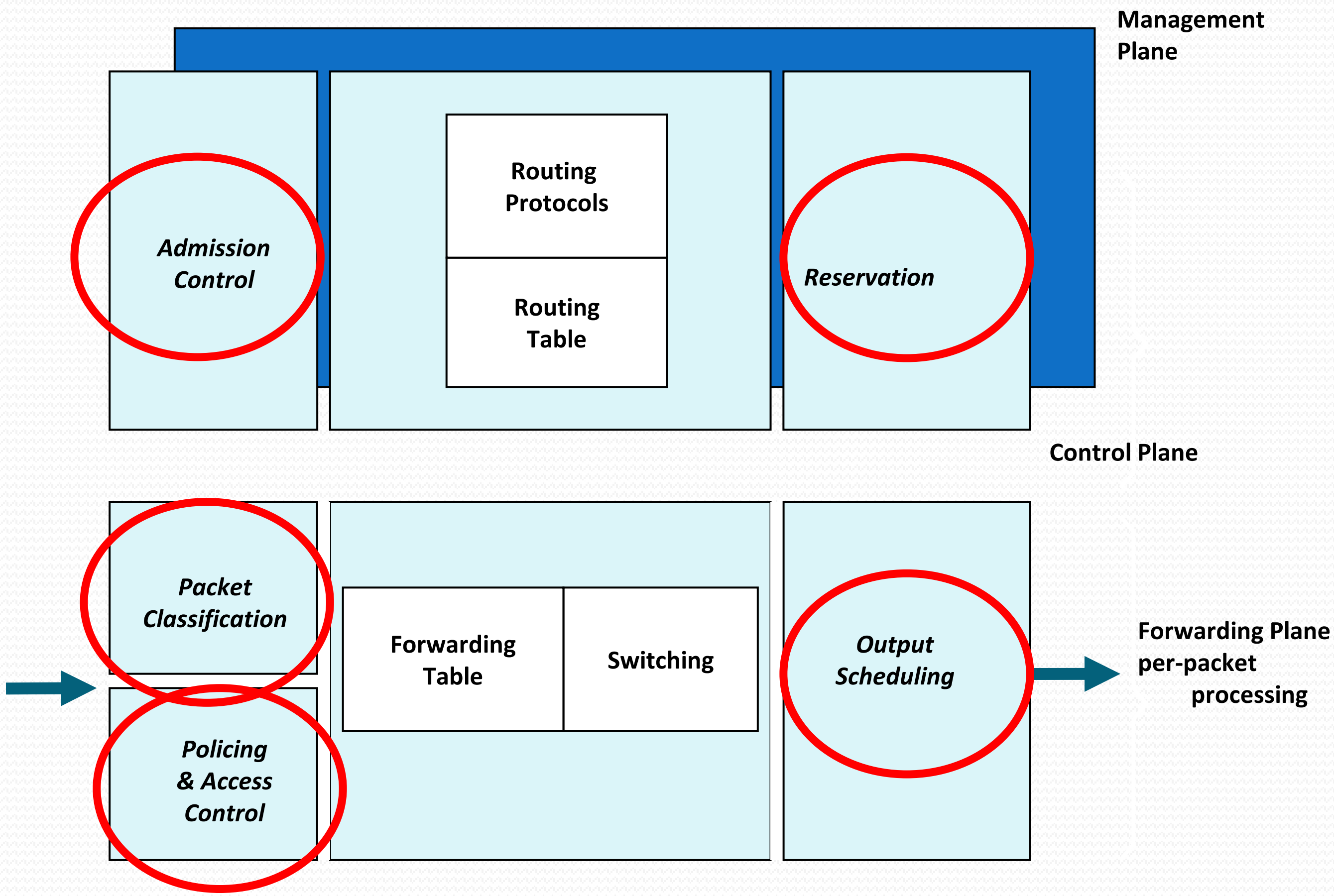
- **Path determination**
  - Process in which a router determines the possible paths in which may forward a datagram towards destination
  - The path may be chosen from information introduced by network administrators (static) or from automatic information (metrics) received by the routers (dynamic)
    - The **metrics** can be any of the following: hops, delay, load, bandwidth, link reliability, ...
  - The information exchanged by the routers to discover paths is particular to any protocol. **In general a routing protocol is characterized by**:
    - The **format and content** of routing packets exchanged between routers and how is done this exchange (e.g.; unicast, broadcast, multicast, ...)
    - The **periodicity** with which the packets are exchanged
    - **Associated algorithms** that allow to calculate the best path towards destination and thus allows to decide the leaving interface (e.g., minimum cost algorithms and associated metrics)

Convergence of routing protocols

- Under topological changes, routers should re-calculate routes and update routing tables
- Then, when all routers reach a common knowledge of the network is called **convergence time**
- Large convergence times imply that the routers will have more difficulties to sent information in the correct interface and this packets will be dropped
- Convergence depends on:
  - Distance in hops from the failure point
  - Number of routers in the network
  - Bandwidth and traffic load in the network
  - Router load (CPU)
  - The routing protocol chosen (minimum cost algorithm)
  - Network administrator configuration and topology (e.g.; bad designed network, loops, …)
- RIP is in the order of seconds (or even some minute), OSPF is in the order of miliseconds, and BGP can be on orders larger than minutes

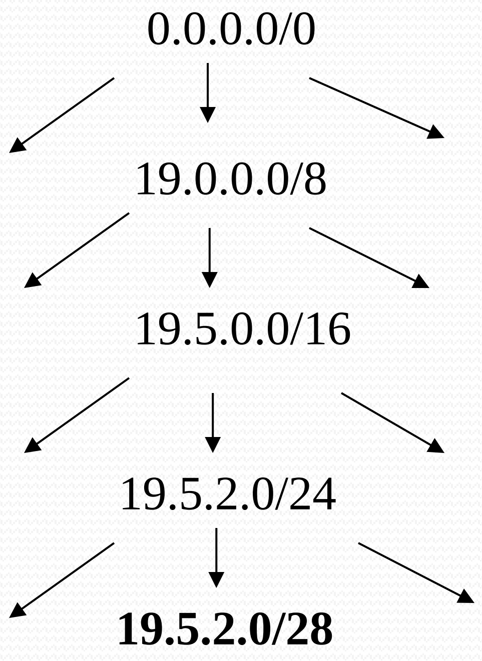
- **Router Architecture -**Three planes:
  - **Management plane:** offers an API (Application Programming Interface) that allows to configure any feature offered by the router.
  - **Control plane:** any protocol or algorithm that draws network maps. For example, the routing protocols (RIP, BGPv4, OSPF, …), signaling protocols, admission control functions, etc
  - **Forwarding Plane:** is that part of the router that decides what to do with the incoming data packets. Most of the time, these actions imply packet manipulation. Functions like forwarding, scheduling (FIFO, priorities, WFQ), policing (Leaky bucket), packet classification, etc

- Router Architecture



- **What does a router when receives a datagram?**
  - Performs all the functions related to data forwarding.
  - The main function is extracting the destination @IP and **look up at the routing table** in order to decide the output interface.
    - **Patricia tree algorithm**: looks for the Longest Prefix Match in the routing table → that entry that has the longest mask.

@IPdestino/Másk	Gateway	Interficie
19.0.0.0/8	0.0.0.0	eth0
19.5.0.0/16	0.0.0.0	eth0
19.5.2.0/24	0.0.0.0	eth0
19.5.2.0/28	0.0.0.0	eth0



Routing Protocols

Static

- Those ones that network administrators manually set the routing entries
- Useful if the network is small or when the network is a stub network (only reachable from one point)

Dynamic protocols

- Those ones that automatically set the routing table
  - Useful in medium to large networks
- Classified in three groups
  - **Vector-distance protocols:** determine direction and distance towards any subnet in the network (e.g. RIP, IGRP, BGP, ...)
  - **Link-state protocols:** use the network topology (e.g.; OSPF, IS-IS)
  - **Hybrids:** combination of the other two

- **Class-full/Class-less routing:**
  - **Class-full routing**: those protocols that do not advertise masks (e.g.; RIPv1, IGRP)
    - Subneting not allowed
    - Take care of discontinous networks since RIPv1 and IGRP summarize
  - **Class-less routing:** those protocols that advertise masks (e.g.; RIPv2, OSPF, BGP, EIGRP, etc)
    - Subneting is allowed using VLSM (Variable Length Subnet Mask)
    - Take care of discontinous networks since most protocols (RIPv2, OSPF, EIGRP) summarize

- **Summarization versus aggregation:**
  - **Major network**: the class (A,B,C) of an IP network
  - Summarization: a network is driven towards its major network.
    - E.g. 143.56.78.0/25→ 143.56.0.0/16 → 143.56/16
  - **Aggregation** (suppernetting): two networks are aggregated forming one with Net-ID. Prefixes have to be contiguous.
    - E.g. 143.56.78.0/25 and 143.56.78.128/25 → 143.45.78.0/24
  - <u>**Intra-domain routing**</u> (RIP, OSPF, IS-IS) **summarizes** by default if the networks are separated by different major networks but never aggregates except that the IETF standard specifies the contrary.
  - <u>**Inter-domain routing**</u> (BGPv4) allows **aggregation** <u>and</u> then also **summarization** as a particular case of aggregation.

Link State Routing Protocols:

- Those routing protocols that react to changes in the link (up/down) sending connectivity information in contrast to vector-distance that sends the current distance to that node (i.e., routing table information).

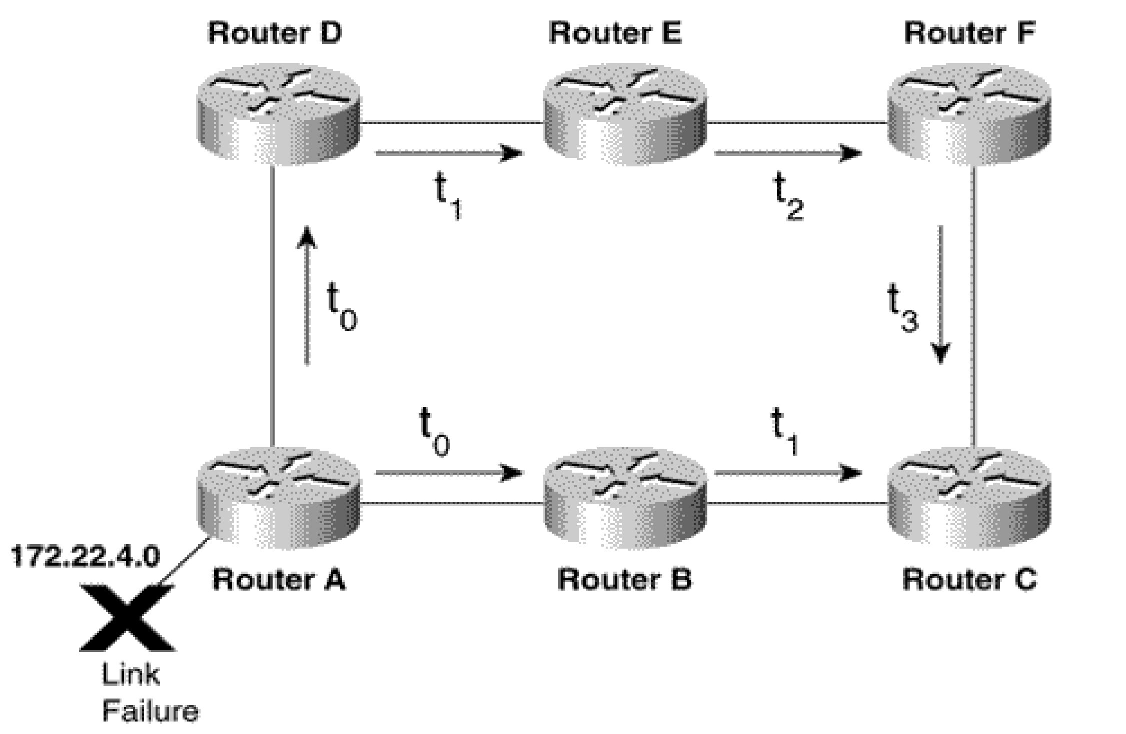
- Link Connectivity is translated as a name and a cost.

- A link state protocol is thus characterized, in general, by:

  - **Discovering of neighbors** (e.g., using a protocol called HELLO)
  - Every node learns the topology of the network (Link State Database) **flooding** Link State Packets (LSP)
  - A **minimum cost algorithm** (e.g., Dijkstra) that calculates the best next hop (routing table) using the data base

- **Flooding Link State information:**
  - Every router k sends Link State Packets (LSPs) to all of its neighbors N(k) (neighborhood set of router k)
  - LSPs arrive and wait in buffers to be "accepted"
  - If node j receives a LSP from node k it compares the **sequence numbers**. If this is the most recent one from k, send to N(j)-{k}.
  - Furthermore, each LSP travels with an **aging field** that adds reliability → each router that floods a message adds a value to the age of the LSP (1 s). Ages are included in the database.
    - the protocol defines an MaxAgeDiff. If a router receives an LSP with different ages and same sequence number and the difference between ages is lower than MaxAgeDiff (e.g. 15 min) then the LSP is not flooded, otherwise, the newest LSP is recorded and flooded.

Flooding Link State example:



- Route 172.22.4.0 fails
- R_A floods an LSP with SeqNum=x
- The LSP arrives to R_C from R_B at t_1
- A delayed LSP arrives to R_C from R_F at t_3
- R_C uses the SeqNuym=x to learn that both LSP are the same and since t_3-t_1<MaxAgeDiff, the first LSP is keept in the database with its age and the new LSP is not flooded.

Dijkstra algorithm:

- Complexity: originally $O(|V|^2)$, actually $O(|E|+|V| \log |V|)$
- C=node identifiers={1,2,…,N}
- F(j) shortest path from node 1 to j
- D(j,i) cost from node i to node j
- $A(j)=\{i \in C: D(j,i)<\infty\}$ set of successor nodes from j

---

Initialization:

$j=1$; $F(1)=0$; $F(i)=\infty$; $i \in \{2,\dots,n\}$; $U=C$

Iteration:

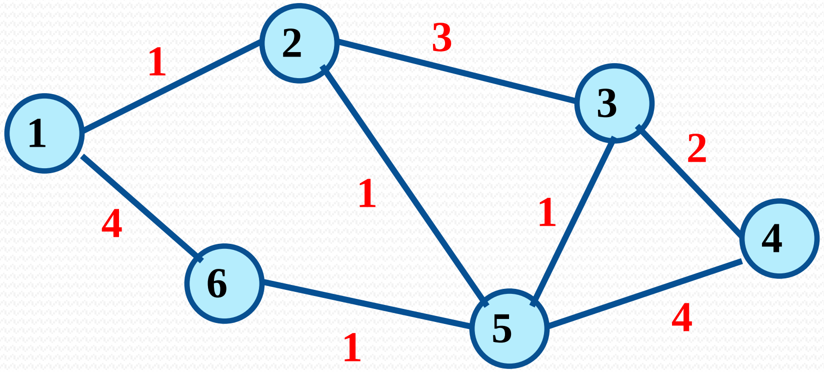
While($j \neq n$ and $F(j) < \infty$) do:

Update $U=U-\{j\}$

Update F: $F(i)=\min\{F(i), F(j)+D(j,i)\}$, $i \in A(j) \cap U$

Update j: $j=\arg\min\{F(i); i \in U\}$

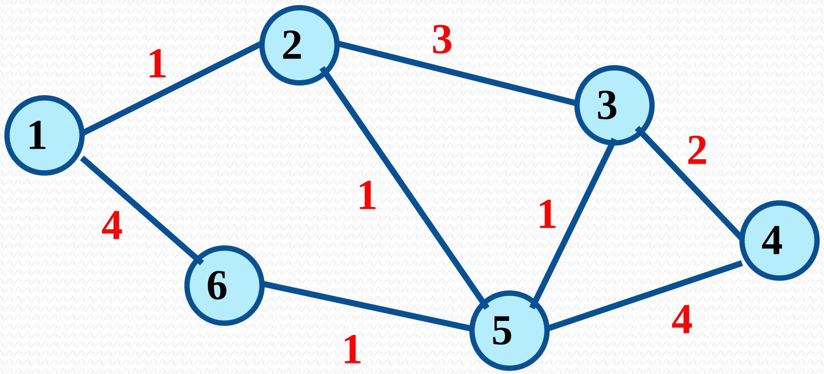
- **Dijkstra example:**



Step 0:

U=C={1,2,3,4,5,6}
F(1)=0
F(i)=∞ ; i={2,3,4,5,6}

- **Dijkstra example:**



Step 1:

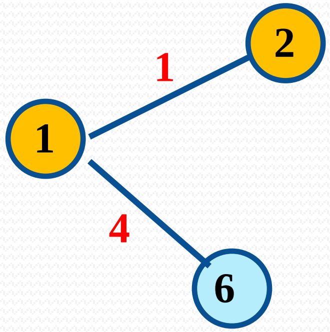
U=U-{1}={2,3,4,5,6}

A(1)={2,6}

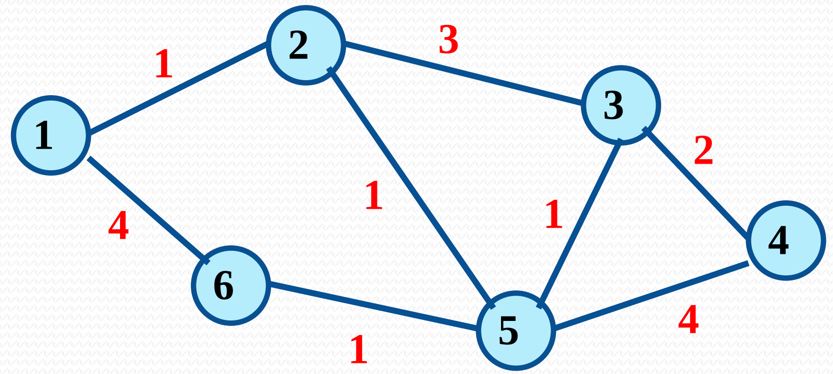
F(2) = min [F(2), F(1)+D(2,1)] = min [∞,0+1] = 1

F(6) = min [F(6), F(1)+D(6,1)] = min [∞,0+4] = 4

j=2 → A(2) = {1,3,5} ; U=U-{2}={3,4,5,6} → A(2)∩U={3,5}



- **Dijkstra example:**



Step 2:

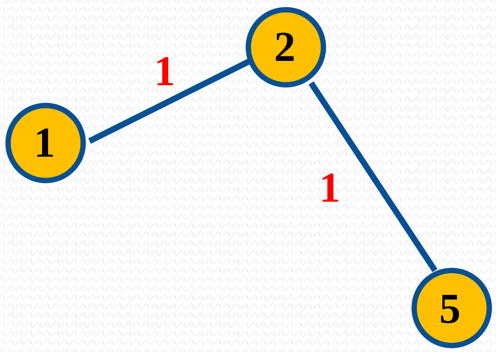
U=U-{2}={3,4,5,6}

A(2)={1,3,5} → A(2)∩U={3,5}

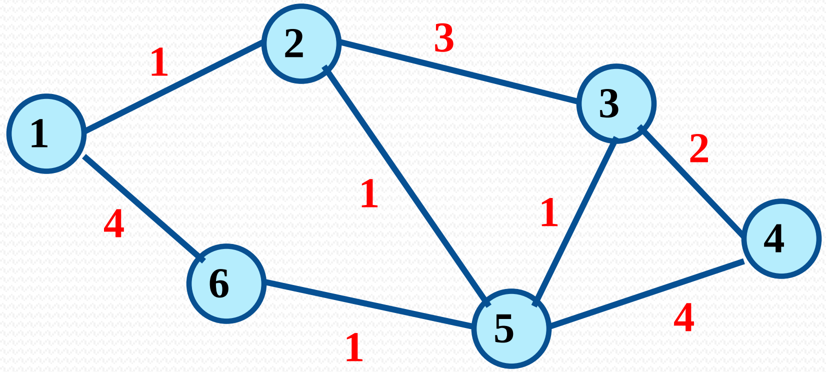
F(3) = min [F(3), F(2)+D(3,2)] = min [∞,1+3] = 4

F(5) = min [F(5), F(2)+D(5,2)] = min [∞,1+1] = 2

j=5 → A(5) = {2,3,4,6} ; U=U-{5}={3,4,6} → A(5)∩U={3,4,6}



- **Dijkstra example:**



Step 3:

U=U-{5}={3,4,6}

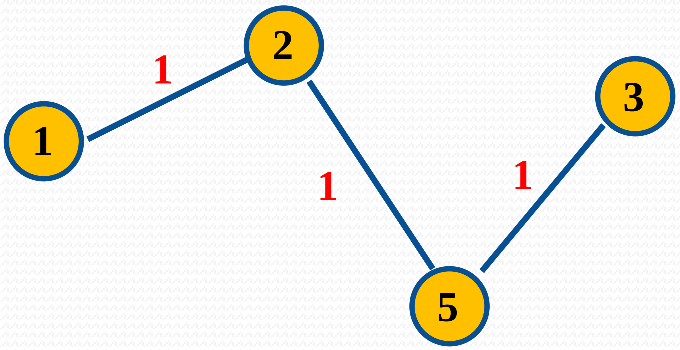
A(5) = {2,3,4,6} → A(5)∩U={3,4,6}

F(3) = min [F(3), F(5)+D(3,5)] = min [∞,2+1] = 3

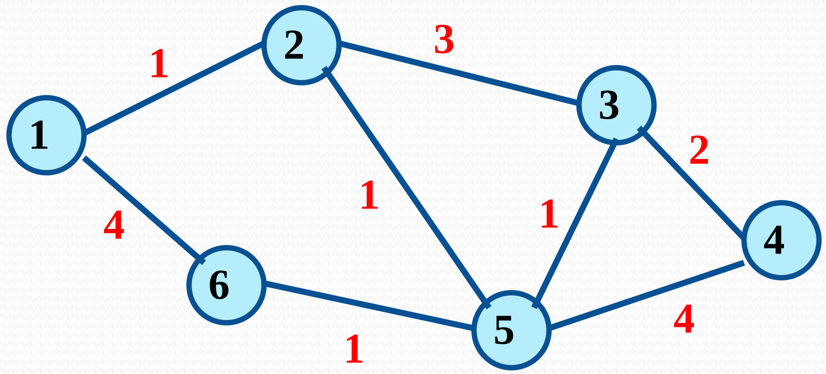
F(4) = min [F(4), F(5)+D(4,5)] = min [∞,2+4] = 6

F(6) = min [F(6), F(5)+D(6,5)] = min [∞,2+1] = 3

j=3 → A(3) = {2,4,5} ; U=U-{3}={4,6} → A(3)∩U={4}



- **Dijkstra example:**



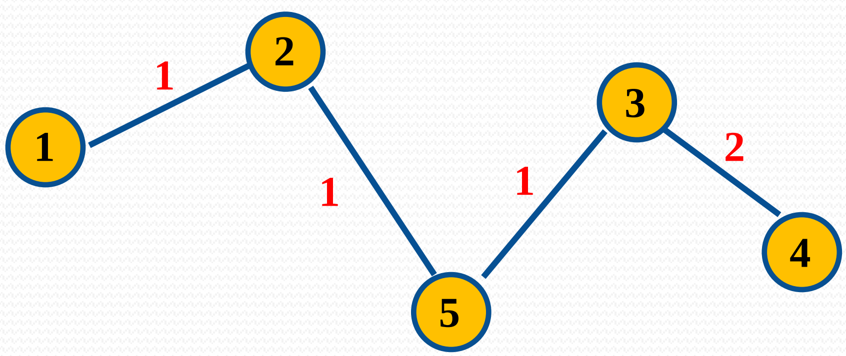
Step 4:

U=U-{3}={4,6}

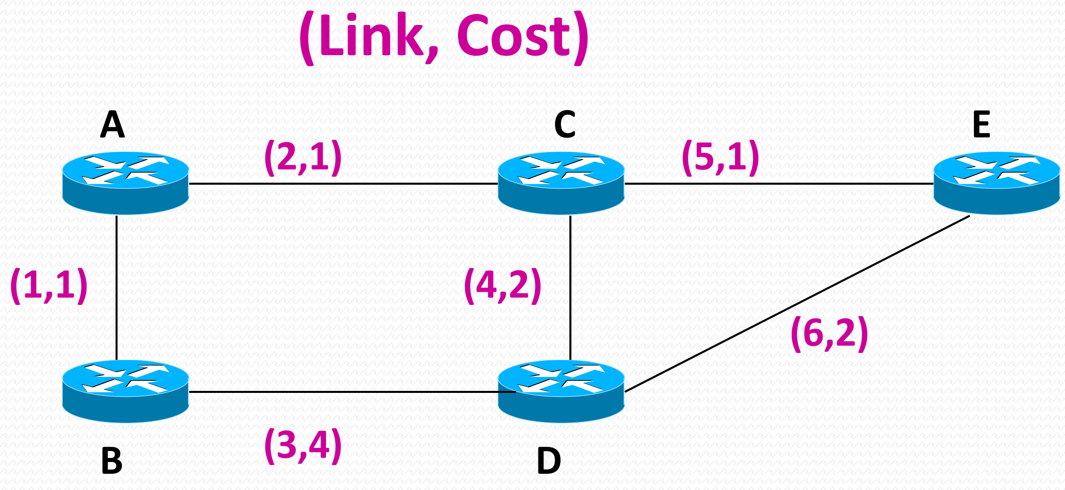
A(3) = {2,4,5} → A(3)∩U={4}

F(4) = min [F(4), F(3)+D(4,3)] = min [6,3+2] = 5

j=4 → A(4) = {3,5} ; U=U-{4}={6} → A(3)∩U=ϕ

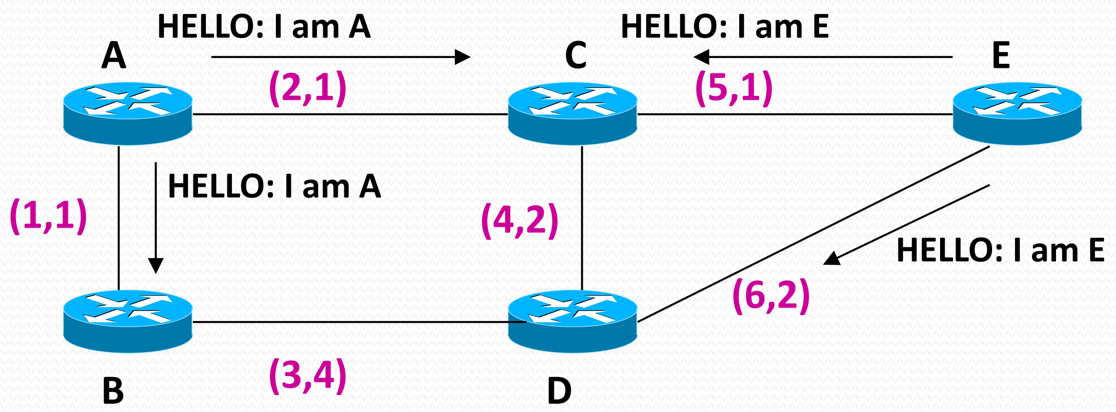


- **Link state general issues:**



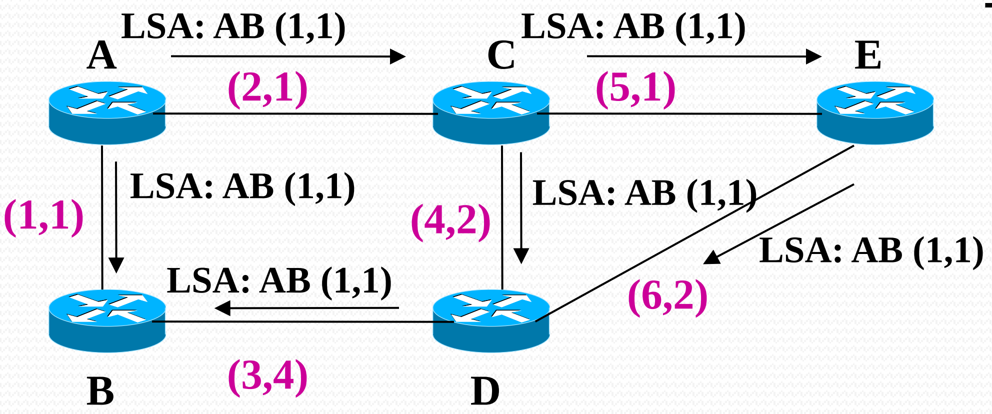
- Do not compute routes in a distributed manner.
- Create a common network topological data base
- Locally each router computes its routing table

- **Link state general issues:**
  - Discovering neighbours



First, each router has to learn who are they neighbors → use a **HELLO protocol**

- **Link state general issues:**
  - Drawing the Topological Network Database



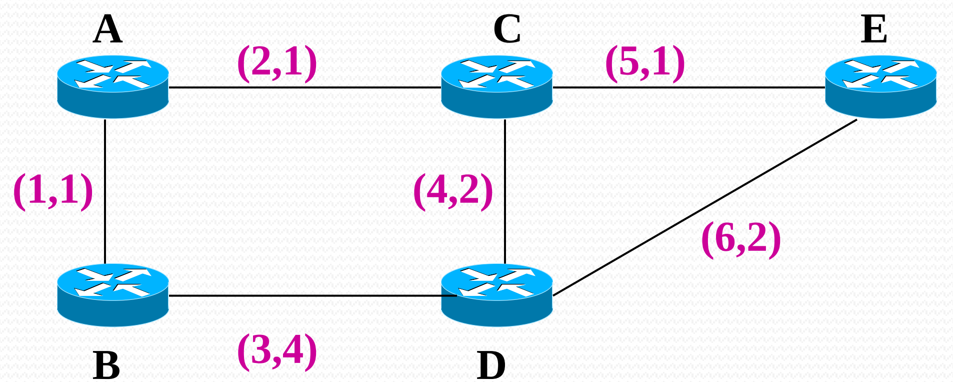
Database (DB) at each router

From	To	Link	Cost	Age
A	B	1	1	1
A	C	2	1	1
B	A	1	1	1
B	D	3	4	1
C	A	2	1	1
C	D	4	2	1
C	E	5	1	1
D	B	3	4	1
D	C	4	2	1
D	E	6	2	1
E	C	5	1	1
E	D	6	2	1

First, each router has to learn who are they neighbors → use a **HELLO protocol**

Second→ Use a **flooding protocol** to send LSA (Link State Advertisements) in order all router gets the same topological data base

- **Link state general issues:**
  - Use **Dijkstra** at each router to get the routing table



Routing table at router A

To	Link	Cost
B	1	1
C	2	1
D	2	3
E	2	2

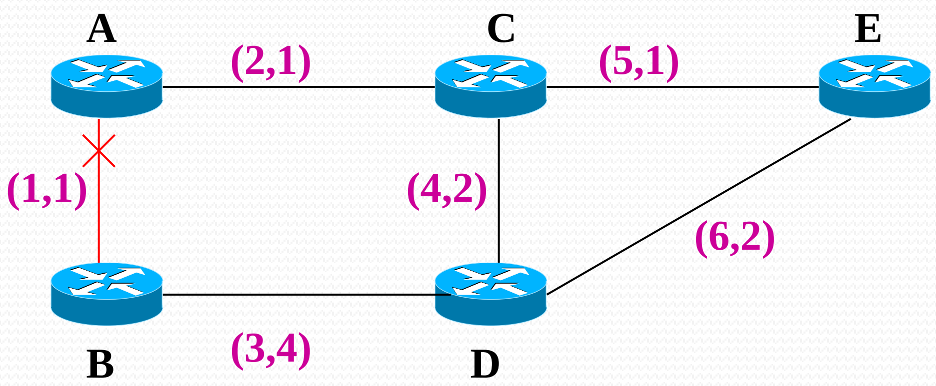
Routing table at router B

To	Link	Cost
A	1	1
C	1	2
D	3	4
E	1	3

Routing table at router D

To	Link	Cost
A	4	3
B	3	4
C	4	2
E	6	2

- **Link state general issues:**
  - Maintain the routing information



Routing table at router A

To	Link	Cost
B	2	7
C	2	1
D	2	3
E	2	2

If a link state change is detected, the topological data base has to be updated. The router/s that detect the link state change has to inform to the rest of routers. For that the routers send LSA's (Link State Advertisement) using **Flooding protocols** to disseminate information.

A and B detect a link state change, modify their DBs and transmit the change to update the neighbor DB → **From A to B, link 1, cost = ∞**

All routers re-compute and refresh their routing table.

- **OSPF (Open Shortest Path First):**
  - Link State protocol
  - Routing messages are encapsulated in IPv4 packets and identified from transport protocols using number 89 (remember that TCP=6, UDP=17)
  - OSPF may be used in:
    - **BMA** (Broadcast Multi-Access) topologies: LANs
    - **Point-to-point** topologies: dedicated lines (e.g.; E1)
    - **NBMA** (Non-Broadcast Multi-Access) topologies: ATM or Frame Relay

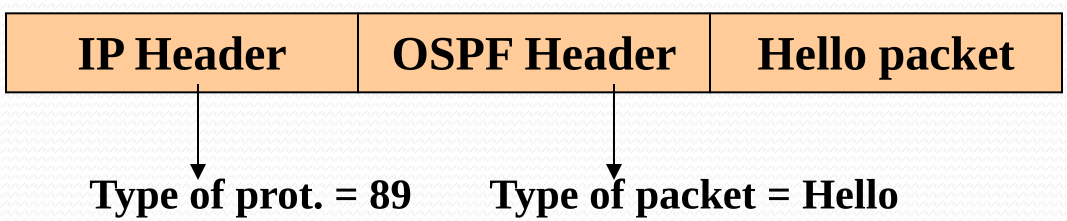
OSPF general issues:

- Each router **draws a map** with the whole network
- Each router sends information to ALL network routers when a link state change is detected
- From this information each router re-calculates the routing table using **Dijkstra algorithm (minimum cost algorithm)**

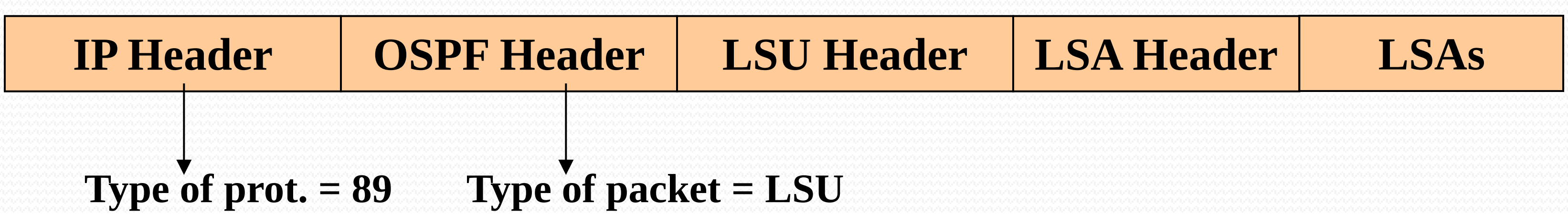
- Thus, OPSF is based on:
  - Discovering neighbors using a protocol called HELLO
  - Send LSAs (Link State Advertisements) with all the changes detected
  - Maintain a data base with the network topology (Link State Database) at each router
  - A minimum cost algorithm (Dijkstra) that calculates the best next hop using the data base

OSPF packet format

- Use IP encapsulation with transport protocol 89
- Different OSPF packets (**HELLO, UPDATE, REQUEST, …**) defined inside the OSPF header and identified in the “type of packet” field.
- Then further encapsulation for the different ospf packets
  - **HELLO packets**



  - **DBD, LSU, LSR and LSACK packets**



- # OSPF Header:

```
      1               2               3               4
0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Version      |     Type      |        Packet Length          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Router ID                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Area ID                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Checksum            | Authentication Type           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Authentication information                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Data (variable)                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- **OSPF Header:**

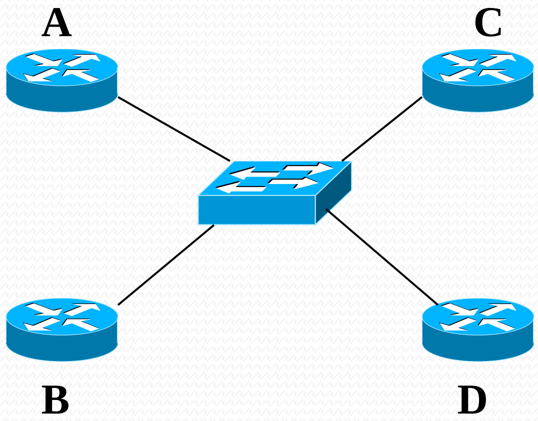
  - **Version**: OSPF version ID
  - **Type**: OSPF packet type. Five OSPF packets:
    - HELLO packets
    - Database Description (DBD) packets
    - Link-State Request (LSR) packets
    - Link-State Update (LSU) packets
    - Link-State ACK (LSAck) packets
  - **Packet Length**: including OSPF header

OSPF Header:

- **Router ID (RID)**: identifies the OSPF packet owner (each router chooses RID as the largest @IP among the active router @IP and some choose the loopback if active)
- **Area ID**: identifies the routing OSPF area
- **Checksum**
- **Authentication type**:
  - Type 0: no authentication
  - Type 1: clear-text password or simple authentication
  - Type 2: cryptographic or MD5 authentication
- **Authentication information**: contains the authentication info
- **Data**: encapsulates routing information

- **OSPF packet types (TYPE field in OSPF header)**
    - **HELLO packets (Type=1)**: establish and maintain neighbor relationships
    - **Database Description (DBD) packets (Type=2):** describe DB (DataBase) contents
    - **Link-State Request (LSR) packets (Type=3):** request portions of the DB
    - **Link-State Update (LSU) packets (Type=4):** answer with the DB portions
    - **Link-State ACK (LSAck) packets (Type=5):** acknowldege link-state updates
- **LSAs (Link-State Advertisements):** Data unit that describes the network or router local state. For each router, this includes router interface states and adjacencies
    - LSA are encapsulated in DBD, LSU, LSR o LSAck packets, never in Hello packets

- **DR (Designated Router) and BDR (Backup-DR):**
  - The main objective of the DR is to minimize the amount of flooding (forwarding) and the DBs synchronization mechanism centralizing the exchange of information
    - Routers do not exchange link state information among them since the amount of packets would be very high → exchange link state only with DR and BDR
    - DR maintains DBs (DataBases) of all routers that is responsible synchronized
    - BDR do nothing while there is an active DR, only acts if the DR fails
    - **Adjacency**: relationship established among a router and its DR and BDR



In a LAN with N links there are N*(N-1)/2 adjacencies. Each router has to announce these adjacencies. In order to optimize the number of adjacencies announced to the N-1 routers a **designated router (DR)** is selected to receive information from its neighbors. DR will have to announce the DB so all routers are synchronized.

The **HELLO protocol** is used to elect the DR and BDR router

HELLO packets:

- **Test** that the line with a neighbor is operative and thus may interchange packets
  - If a HELLO packet is not received in 4*DeadInterval, then the neighbour is lost (Link State down)
- **Choose** a designated router (DR or Designated Router) and the backup router (DBR or Designated Backup Router)
- **Sent** periodically (HelloInterval=10 seconds) using the ***multicast address All-OSPF-routers 224.0.0.5***

- # HELLO Header:

```
      1                   2                   3                   4
0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         OSPF packet header with Type= 1 (HELLO packet)        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Network Mask                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          HelloInterval          |    Options    |    Rtr Pri  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     RouterDeadInterval                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Designated Router                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Backup Designated Router                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Neighbor                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             ...                               |
```

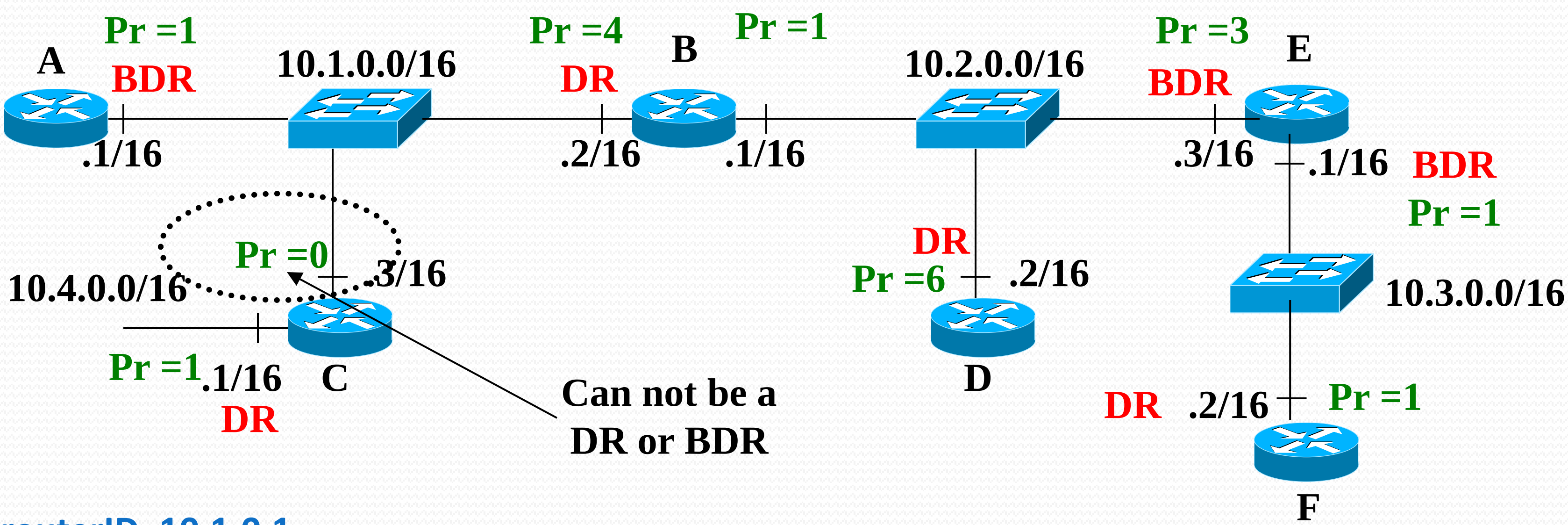
HELLO Header:

- **Network mask**: mask associated with that interface
- **Hello Interval**: interval in which se HELLO’s packets are sent (10 seconds)
- **Options**: documented in the OSPF RFC
- **Router Priority**: priority (default =1)
- **Router-Dead-Interval**: time that a router waits to consider that a neighbor is or it is not active (4*HelloInterval)
- **DR y BDR**: both IP addresses (0.0.0.0 if unknown)
- **Neighbors**: neighbor RouteID list listened in the last Router-Dead-Interval seconds

DR and BDR election process:

- **Each BMA network segment has a DR and a BDR**, so, a router connected to multiple networks may act as DR in a BMA segment and act as a normal router in other segment. So DR and BDR are set at INTERFACE level
- The router with <u>highest priority level</u> is chosen as DR while the second is chosen as BDR
- Default router priority is 1, <u>highest RID</u> is used to break ties (where the router ID use to be the highest active @IP in the router)
- Routers with **priority=0** cannot be elected as DRs or BDRs
- If a router with higher priority than a DR or BDR joins the network, **the DR and BDR does not change**. These only change when the DR fails → the BDR becomes DR, and a new BDR is elected
- A BDR detects that a DR fails because during certain Tout does not listen LSAs (Link State Advertisements)

- DR and BDR election process:



R_A- routerID=10.1.0.1

R_B- routerID=10.2.0.1

R_C- routerID=10.4.0.1

R_D- routerID=10.2.0.2

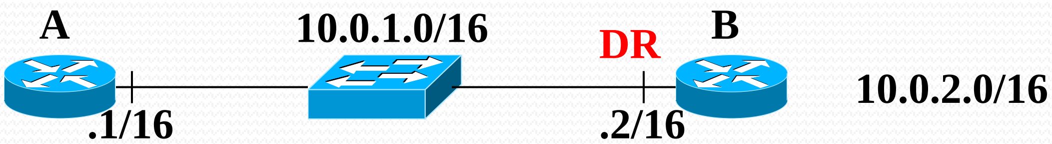
R_E- routerID=10.3.0.1

R_F- routerID=10.3.0.2

Each router has to find an adjacency in each network segment

- **Building the routing table:**
  - Once the DR and BDR are elected, routers has to learn network routes → ***Exchange Protocol***
    - DR and DBR form an adjacency with each router of the BMA network **(Exstart State)**
      - At each adjacency, one of the routers acts as "master" (usually the DR) and the other as "slave"
    - The master router sends a DB summary to the slave and this one acknowledges this packet and viceversa **(Exchange State)**
    - The slave looks at its database and request for those routes to which it haven't enough information **(Loading State)**
    - **Builds the routing table (Full State)**

- **Building the routing table:**



Exstart state

Hello: I will start exchange since I have routerID=10.0.1.1

No I will start exchange since I have a higher routerID=10.0.1.2 :Hello

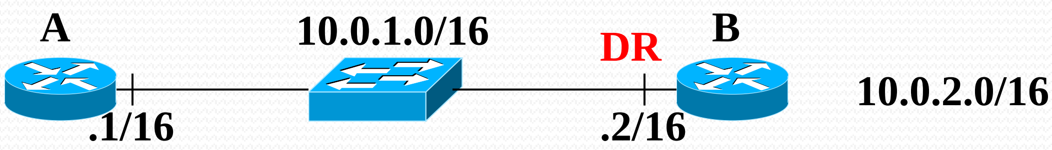
Exchange state

Here is a summary of my link-state DataBase :DBD

DBD: Here is a summary of my link-state DataBase

LSAck: Thanks for the info LSAck:

- **Building the routing table:**



Loading state

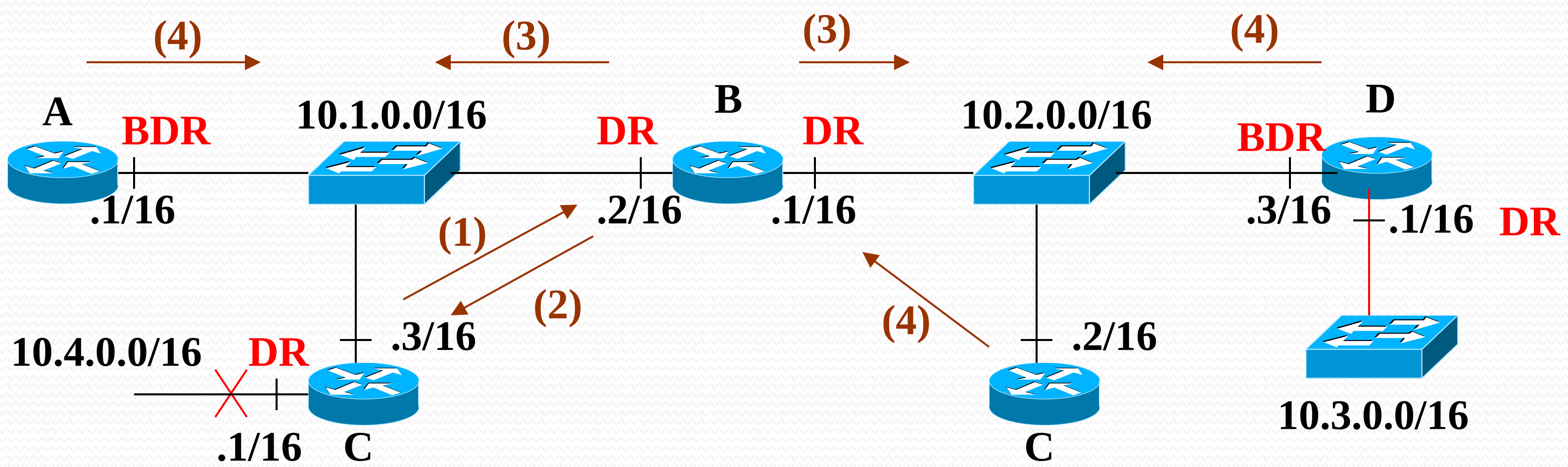
LSR: I need info about net 10.0.2.0/16

Here is the entry of net 10.0.2.0/16 :LSU

LSAck: Thanks, all is OK LSAck:

Full state

Routing Maintenance:



All routers must have the same DB

(1) Router C sends an LSU to its DR using multicast address 224.0.0.6 (All-DR-DBR-routers)

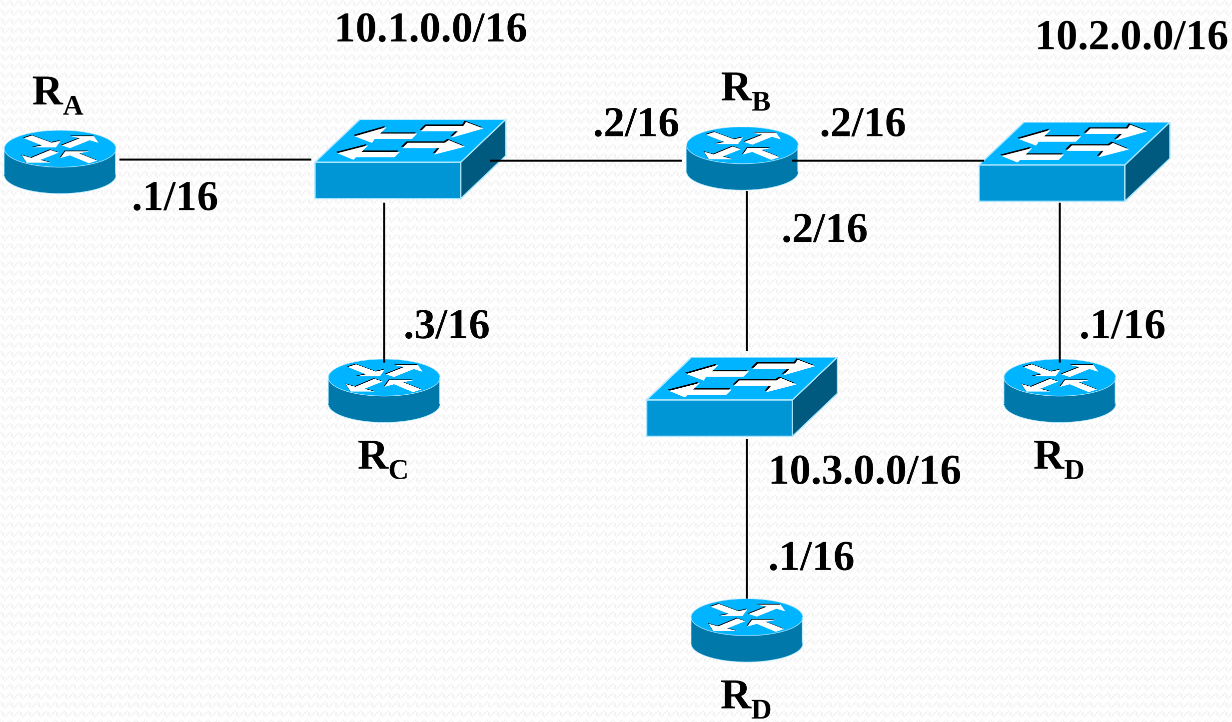
(2) DR ACKs the LSU with a LSAck

(3) DR floods the LSU using multicast address 224.0.0.5 (All-OSPF-routers)

(4) All routers ACK that LSU

(5) All routers recalculate their routing table

- **OSPF configuration in a single area:**



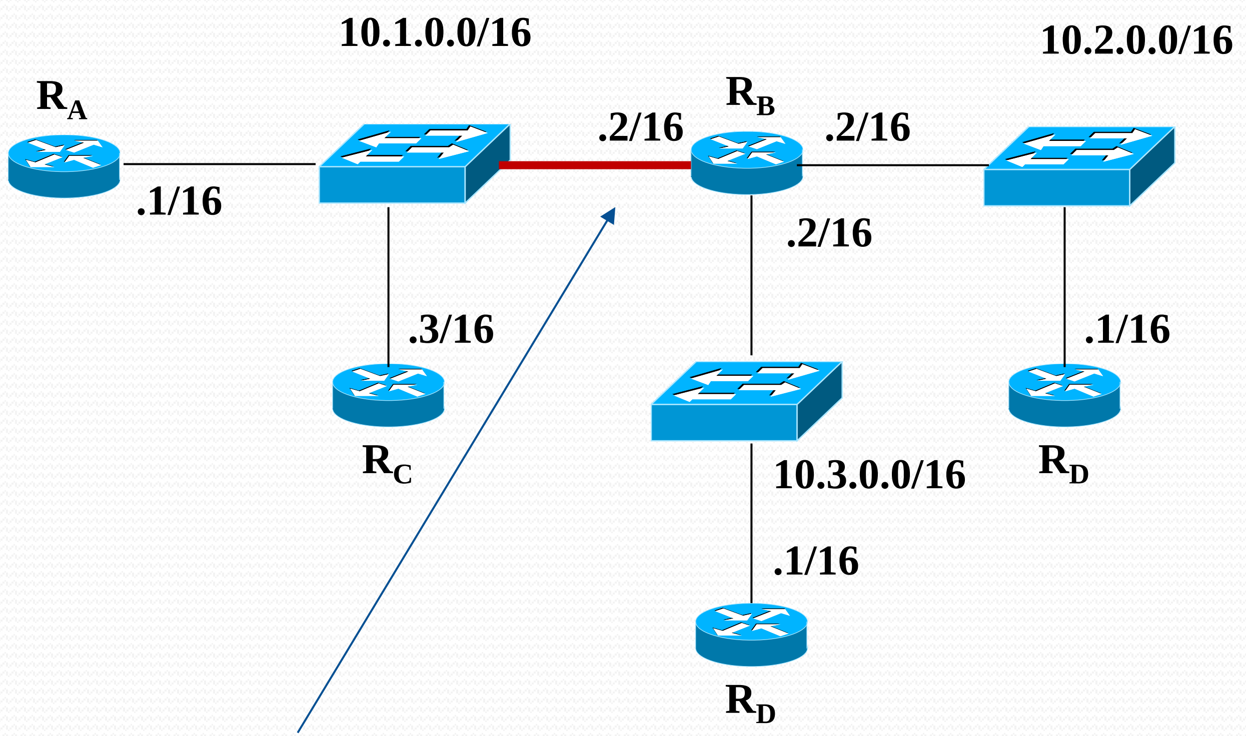
- **OSPF configuration in a single area: CISCO IoS**

!!!! Configure Router RB

RB(conf)# **router** ospf 1

RB(conf-r)# **network** 10.1.0.0 255.255.0.0 **area** 0

- **OSPF configuration in a single area:**



Interface 10.1.0.2 is OSPF active in area 0.

The interface understands multicast address 224.0.0.5 (all-ospf-routers)

The router sends HELLOS on this interface and announces route 10.1.0.0/16 on any active OSPF interface

- **OSPF configuration in a single area : CISCO IoS**

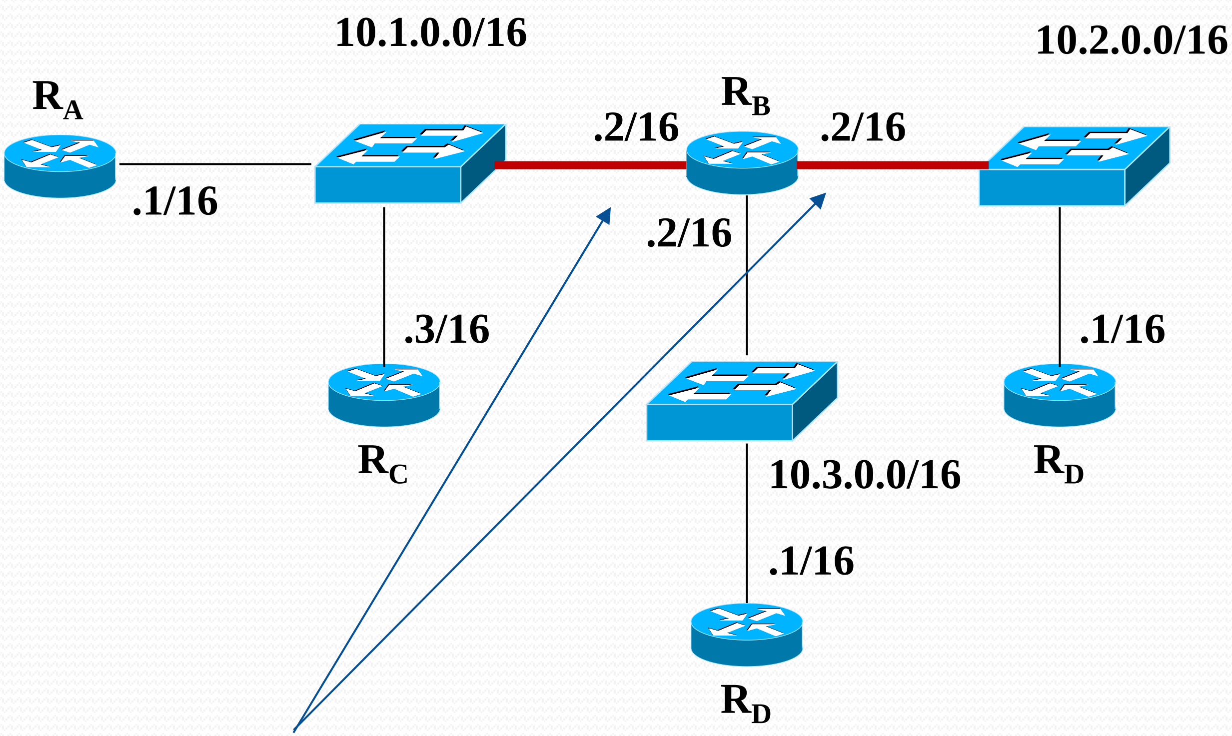
  !!!! Configure Router RB

  RB(conf)# **router** ospf 1

  RB(conf-r)# **network** 10.1.0.0 255.255.0.0 **area** 0

  RB(conf-r)# **network** 10.2.0.0 255.255.0.0 **area** 0

- **OSPF configuration in a single area:**



Interfaces 10.1.0.2 and 10.2.0.2 are OSPF active in area 0.

The 2 interfaces understand multicast address 224.0.0.5 (all-ospf-routers)

The router sends HELLOS on these interfaces and announces route 10.1.0.0/16 and route 10.2.0.0/16 on any active OSPF interface

- **OSPF configuration in a single area : CISCO IoS**

  !!!! Configure Router RB

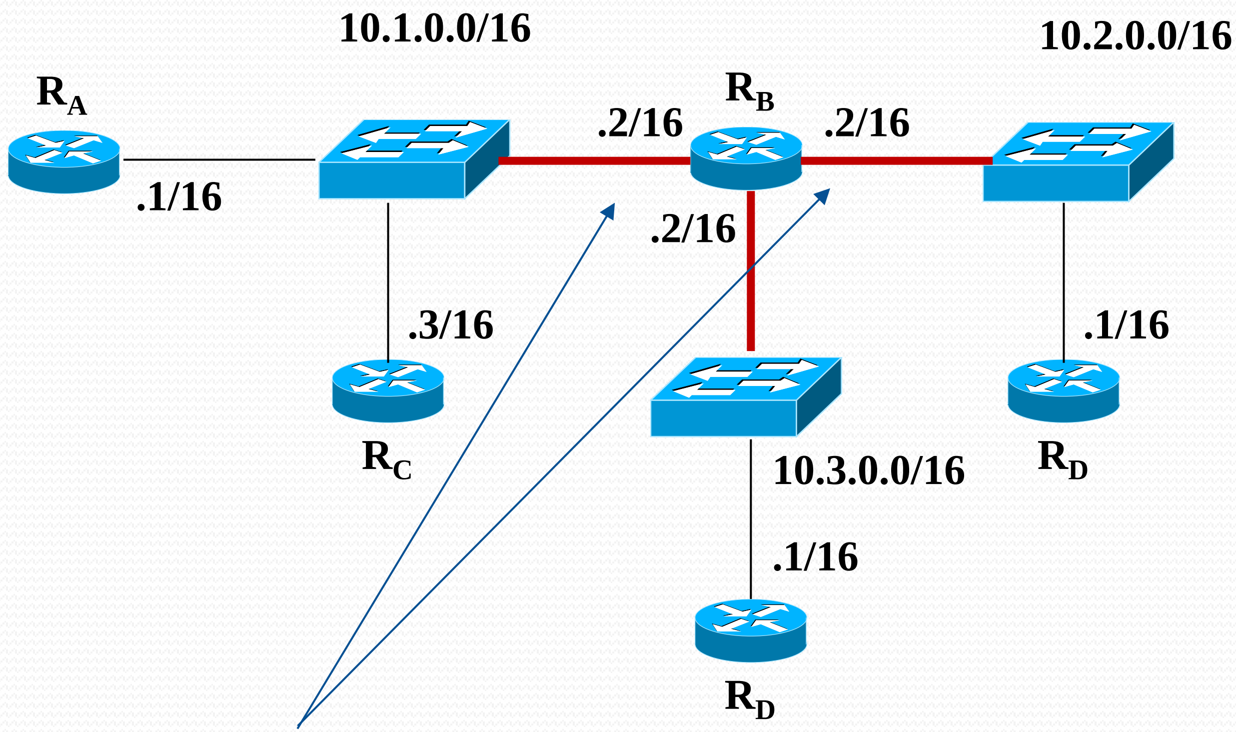
  RB(conf)# **router** ospf 1

  RB(conf-r)# **network** 10.1.0.0 255.255.0.0 **area** 0

  RB(conf-r)# **network** 10.2.0.0 255.255.0.0 **area** 0

  RB(conf-r)# **network** 10.3.0.0 255.255.0.0 **area** 0

- **OSPF configuration in a single area:**

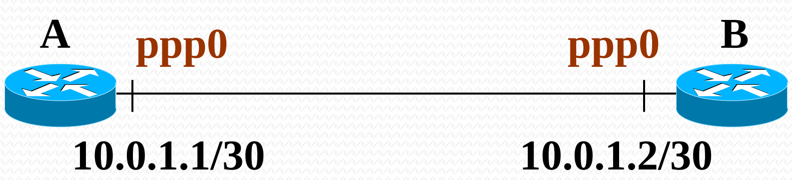


Interfaces 10.1.0.2, 10.2.0.2 and 10.3.0.2 are OSPF active in area 0.

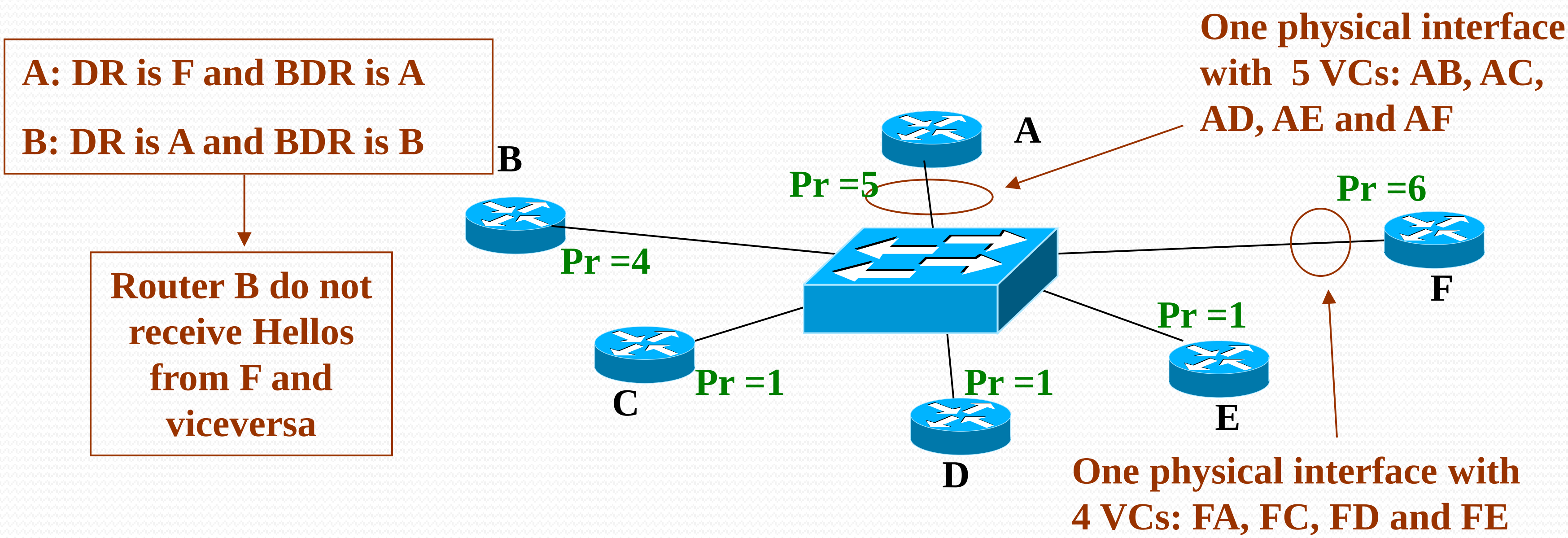
The 3 interfaces understand multicast address 224.0.0.5 (all-ospf-routers)

The router sends HELLOS on these interfaces and announces routes 10.1.0.0/16, 10.2.0.0/16 and 10.3.0.0/16 on any active OSPF interface

- **Point-to-Point topologies** (e.g.; dedicated lines E1 or T1)
  - Each router is adjacent by definition since there only are two routers. One is DR and the other BDR.
  - Both use HELLO packets to discover each other (using multicast address 224.0.0.5 “All-OSPF-routers”)



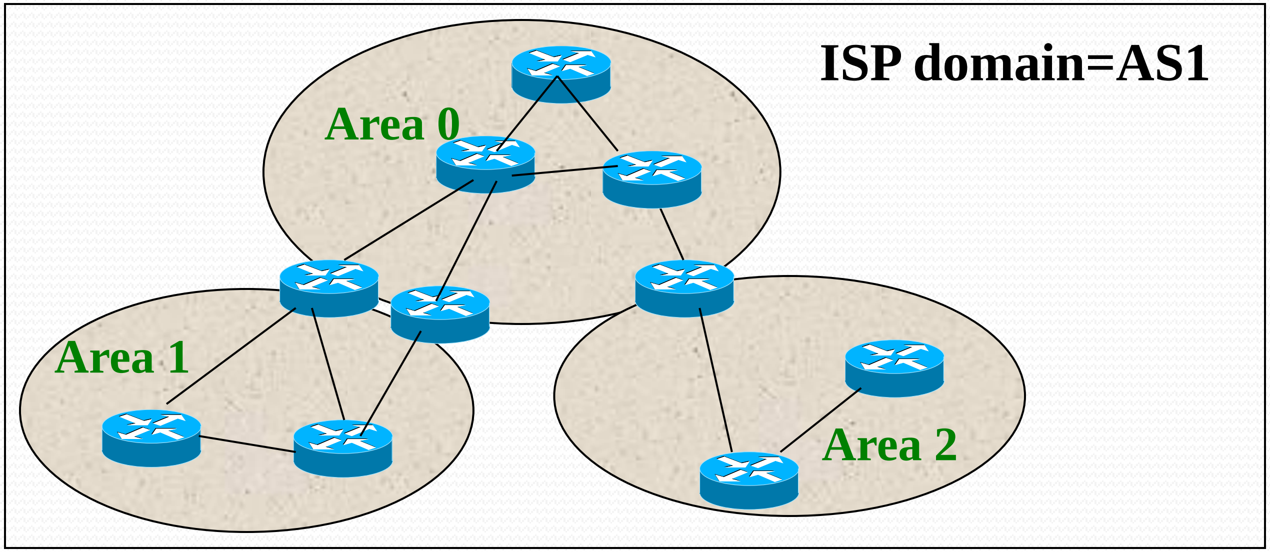
- **NBMA (Non-broadcast Multi-Access) topologies:**
  - NBMA topologies are those ones in which a non broadcast WAN communicates the routers (e.g.; Frame Relay or ATM)
  - Problem: difficulty in the DR/BDR election
    - E.g.; WAN as Frame Relay or ATM (Virtual Circuit switching)



- **NBMA (Non-broadcast Multi-Access) topologies:**
  - Solutions (non-standard, manufacturer dependent):
    1. **NMBA**: emulate OSPF in a MBA network → needs a full meshed network (N-1)*N/ VC and sends a Hello per VC
       - CPU consuming !!! and bandwidth intensive !!!
       - Useful in small networks (does not scale well)
    2. **Point-to-MultiPoint**: each link creates a point-to-point connection per VC (partial meshed networks)

- **OSPF Scalability:**
  - Assume big networks (e.g., 400 routers !!!)
    - Routers has to send large amount of LSUs (bandwidth intensive)
    - Routing table re-computation each time there is a change in the network (Dijkstra consumes a lot of CPU)
    - The DataBase may be quite large (store all links and costs for the whole network)
    - Routing table can be very large
  - Solution:
    - Hierarchical routing using the concept of "**areas**"
    - **Area**: set of routers and networks in the same AS (Autonomous System) using single area OSPF and connected to other OSPF areas

OSPF Scalability:



	Minimum	Mean	Maximum
Routers in a Domain	20	510	1000
Routers in a single Area	20	160	350
Areas per Domain	1	23	60

Standard Area: single area as we have previously seen

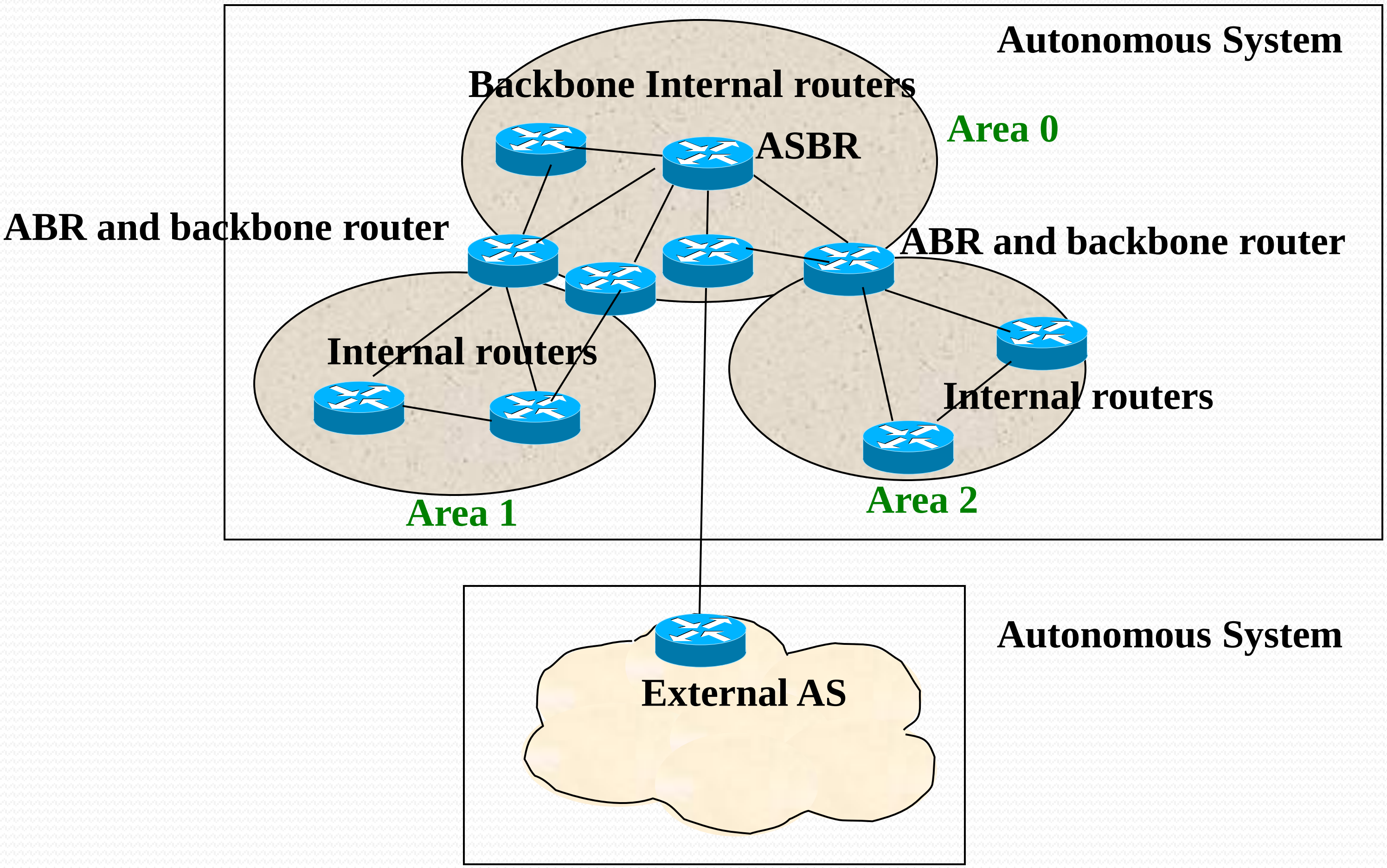
Backbone Area (transit area): area (area 0) that interconnects other areas in a multi-area system.

Stub area: area that is connected to area 0 with only one link → use default routers (0.0.0.0)

OSPF Scalability:

- Router types in a multi-area system:
  - **Internal Router:** router all interfaces inside the same area
    - maintain only one DB (identical to other routers in the same area)
  - **Backbone or transit Router :** routers all their interfaces in Area 0
    - Area 0 serves as backbone (transit) with other areas
  - **Area Border Router (ABR):** routers with interfaces in multiple areas
    - Maintain one BD per area to which they are connected including the backbone (consume memory and CPU)
    - Summarize area information and distribute them at each area using the backbone area
    - When an ABR receives information from one area, calculates paths towards that area
  - **Autonomous System Boundary Router (ASBR):** routers with one interface towards other AS

- **OSPF Scalability:**

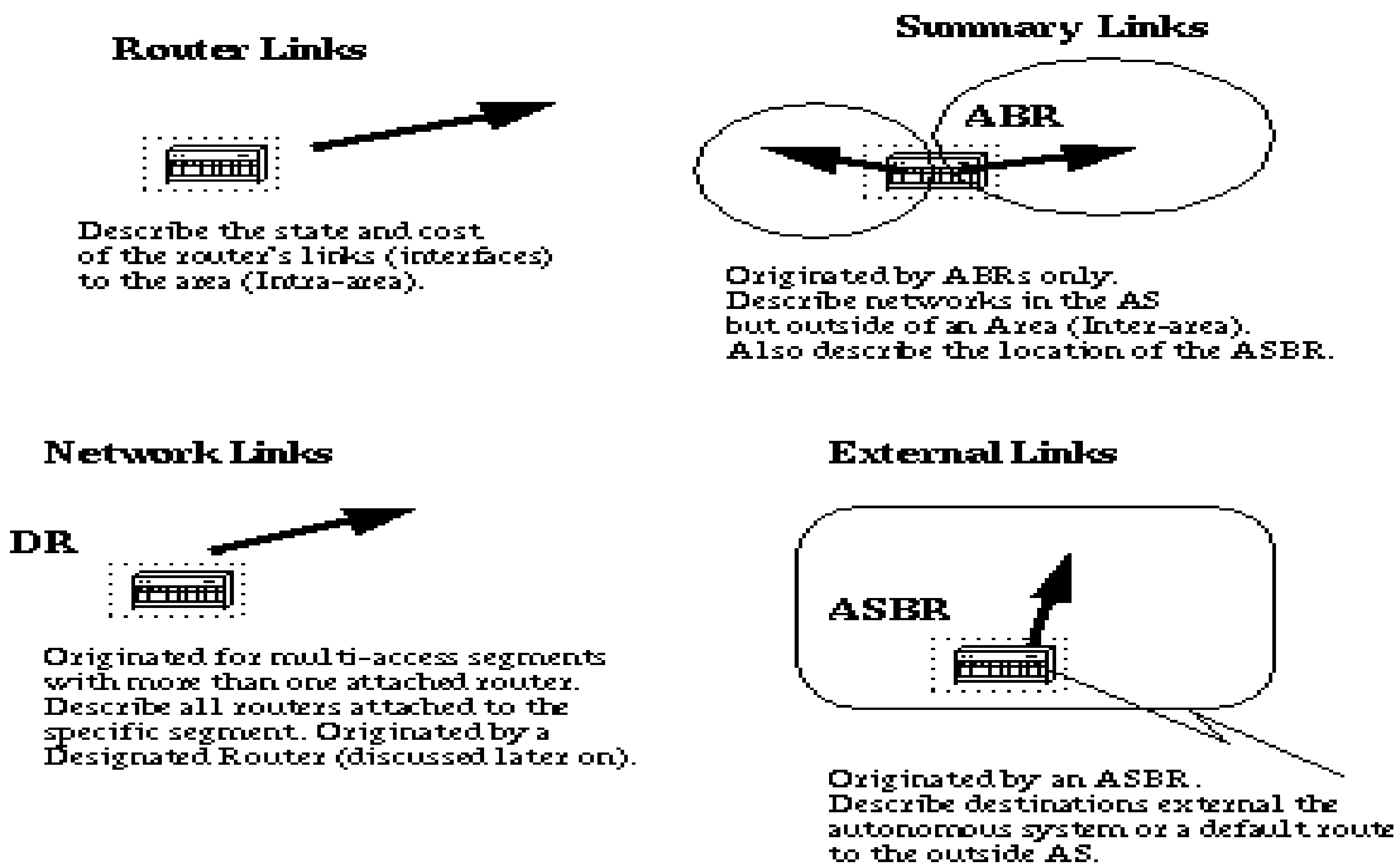


- **OSPF Scalability:**
  - Data packet transmission:
    - **Intra-area routing:** packets directed to internal network
    - **Inter-area routing:** packets directed to external network
      - The packet goes to ABR routers (intra-area routing)
      - The ABR router sends the packet to other ABR connected to the destination area (backbone routing)
      - This ABR router sends the packet inside the destination area (intra-area routing)

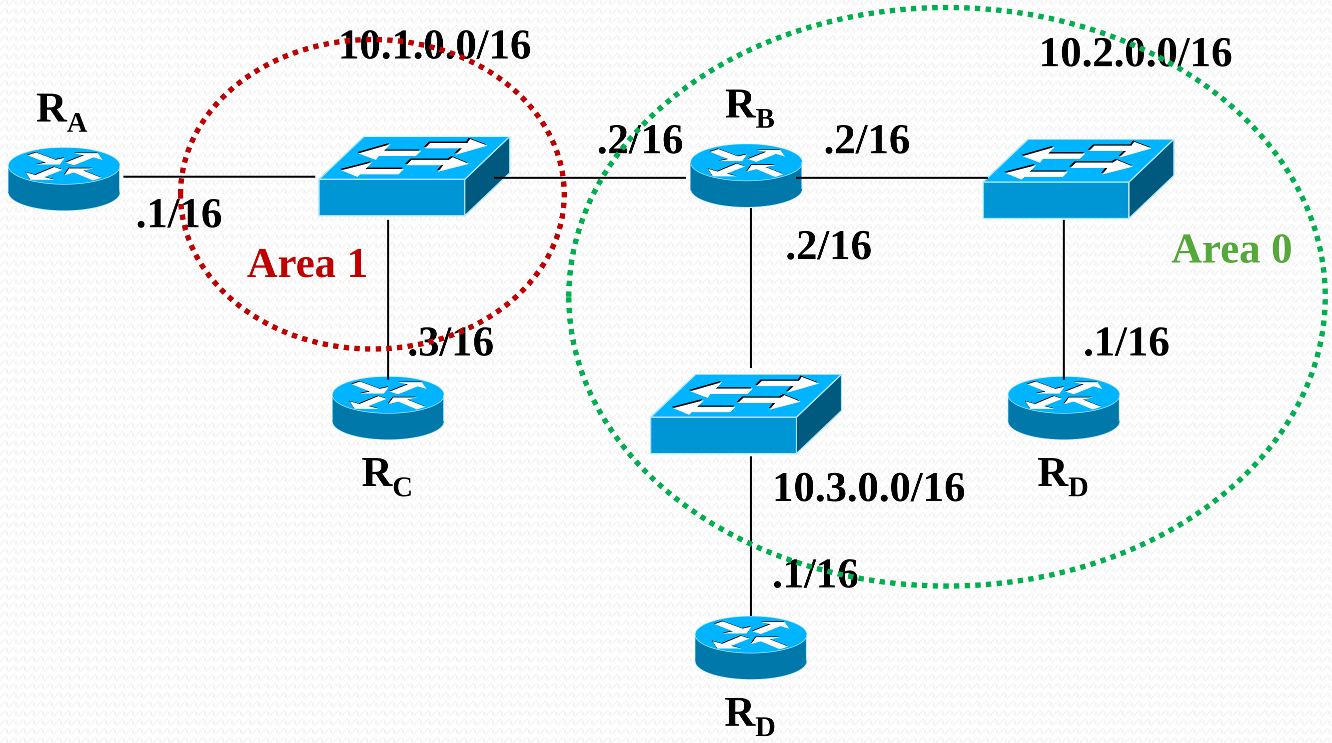
OSPF Scalability:

- LSAs types in OSPF multi-area network
  - **Type 1: Router LSA**:
    - generated by each router inside its area towards each internal router
    - Describes each link state and cost for each internal router
  - **Type 2: Network LSA**:
    - generated by each DR (in a BMA network), describes the set of routers connected to that BMA network (only sent inside the area)
  - **Type 3: Summary LSA (IP network)**:
    - generated by each ABR router, describe external routes (got from other area ABR's)
    - There is an entry per each subnetwork (summarized)
  - **Type 4: ASBR summary LSA**:
    - Generated by each ABR router, describe routes towards ASBR's (to get out the AS)
  - **Type 5: AS external LSA**:
    - generated by ASBR routers, describe external routes belonging to other AS (e.g., default network to get out the AS)

- OSPF Scalability and ISP design



- **OSPF configuration in a multiple area:**



- **OSPF configuration in a single area: CISCO IoS**

```
!!!! Configure Router RB
RB(conf)# router ospf 1
RB(conf-r)# network 10.2.0.0   255.255.0.0    area 0
RB(conf-r)# network 10.30.0   255.255.0.0    area 0
RB(conf-r)# network 10.1.0.0   255.255.0.0    area 1
```

- ISP design: Intra-domain routing