CH. 3
PART 3: IP ROUTING PROTOCOLS

V1.5

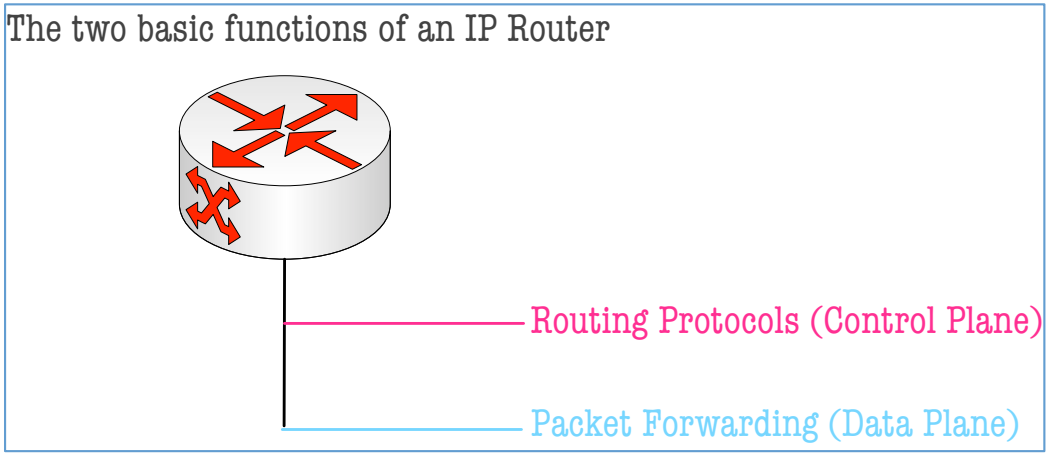
Lecture on how routers communicate over the control plane for sharing routing information

Computer Networks Course, Universidad de León, 2024

What is Routing?

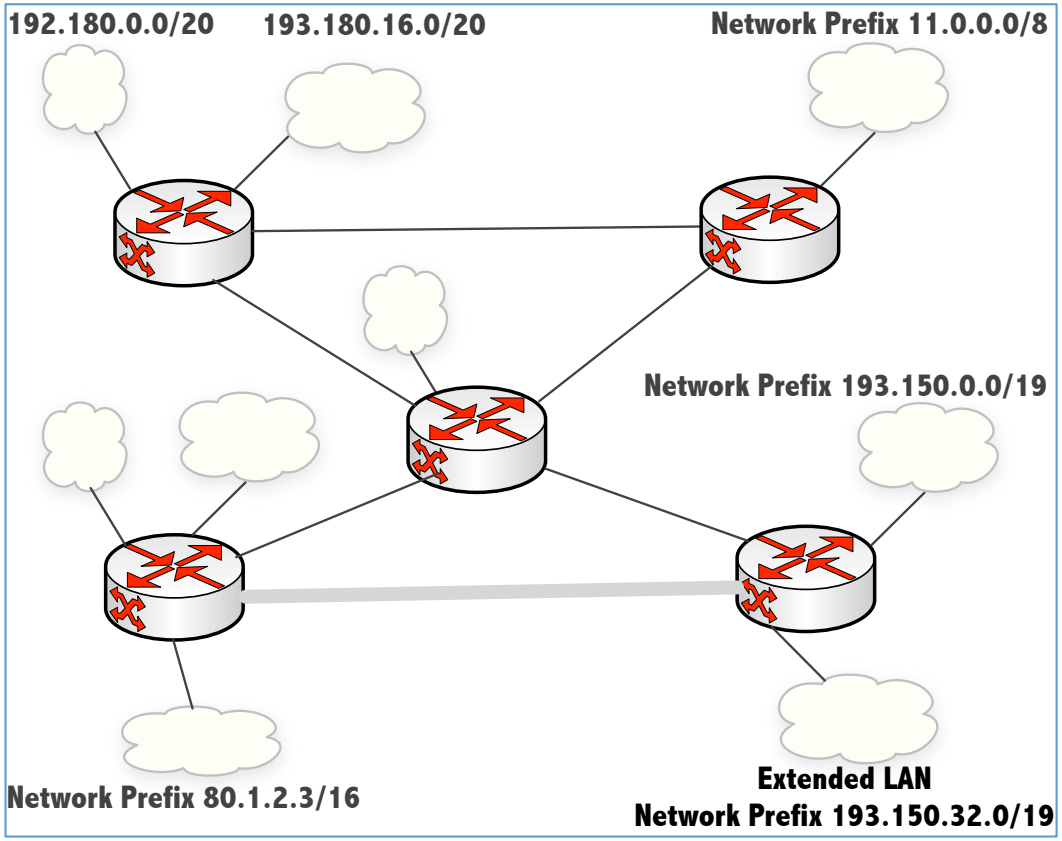
Forwarding vs. Routing

- Forwarding:
  - To select an output port based on looking up each IP packet destination address in the routing table
  - Algorithm: Longest Prefix Matching
- Routing:
  - Process whereby the routing table is built



What is Routing?

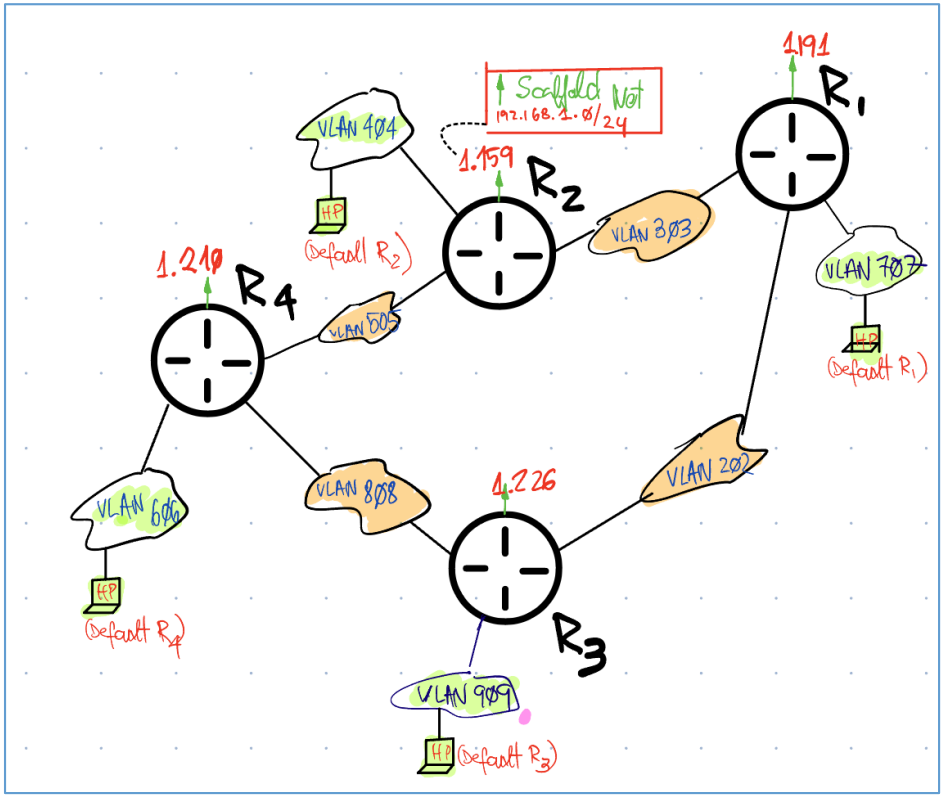
In an internetwork, managing the routing tables of each router is difficult and error prone. How can this problem be solved?



What is Routing?

How can this problem be solved? Not by having an administrator enter the routing tables statically

- We did this in the current practical !
- For appreciating the difficulty of the process



Routing protocols

- The solution comes from running a routing protocol
- The routing protocol shares routing information that allows the each router to build its own forwarding table
- Some of today's Routing Protocols
  - RIP-2 (A distance-vector protocol)
  - OSPF (A link-state protocol; Dijkstra's algorithm)
  - BGP-4 (A Path-vector protocol)

Forwarding *vs.* routing tables

(a)

Prefix/Length	Next Hop
18/8	171.69.245.10

(b)

Prefix/Length	Interface	MAC Address
18/8	if0	8:0:2b:e4:b:1:2

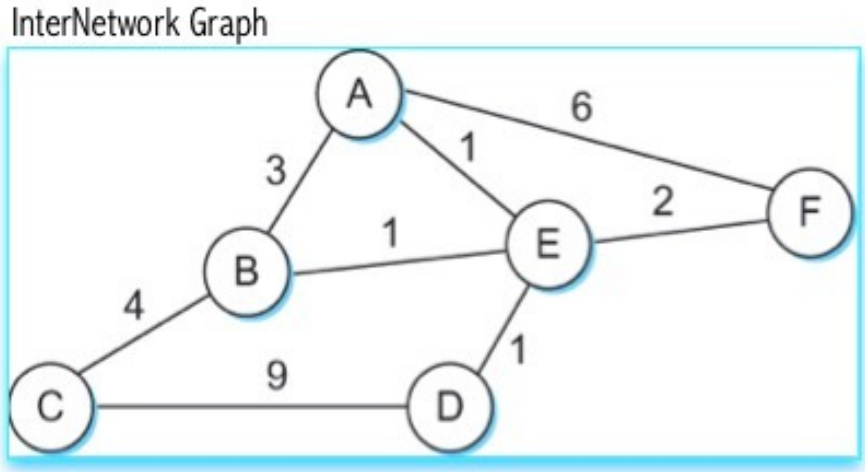
- Forwarding table (b)
  - Used when a packet is being forwarded and so must contain enough information to accomplish the forwarding function
  - A row in the forwarding table contains the mapping from a **network number to an outgoing interface** and some MAC information, such as Ethernet Address of the next hop
- Routing table (a)
  - Built by the routing algorithm as a precursor to build the forwarding table
  - Generally contains mapping from **network numbers to next hops**

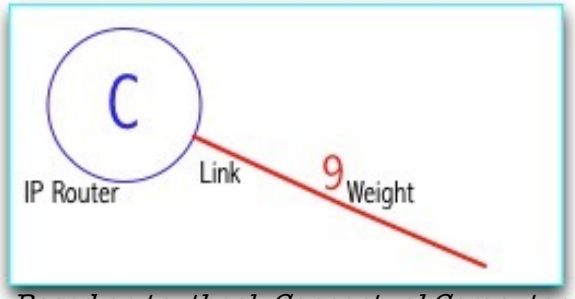
Linux router fwd table


```
root@debian:/home/administrator# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.1.1     0.0.0.0         UG    0      0        0 eno1
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        0 eno1
192.168.2.0     0.0.0.0         255.255.255.128 U     0      0        0 enp1s0.202
192.168.2.128   0.0.0.0         255.255.255.128 U     0      0        0 enp3s0.303
192.168.3.0     192.168.2.130   255.255.255.192 UG    0      0        0 enp3s0.303
192.168.3.128   192.168.2.2     255.255.255.192 UG    0      0        0 enp1s0.202
192.168.3.224   0.0.0.0         255.255.255.240 U     0      0        0 enp1s0.707
192.168.3.240   192.168.2.2     255.255.255.240 UG    0      0        0 enp1s0.202
```

Least cost Routing

- Network (*Internetwork*) as a Graph
- The basic problem of routing is to find the lowest-cost path between any two nodes
  - Where the cost of a path equals the sum of the costs of all the edges that make up the path





Based on textbook Conceptual Computer Networks © 2013-2018 by José María Foces Morán
& José María Foces Vivancos

Routing

- For a simple network, we can calculate all shortest paths and load them into some nonvolatile storage on each node.

- Such a **static** approach has several shortcomings
  - It does not deal with node or link failures
  - It does not consider the addition of new nodes or links
  - It implies that edge costs cannot change

- What is the **solution**?
  - Need a distributed and dynamic protocol
  - Two main classes of protocols
    - Distance Vector (RIP-2)
    - Link State (OSPF)

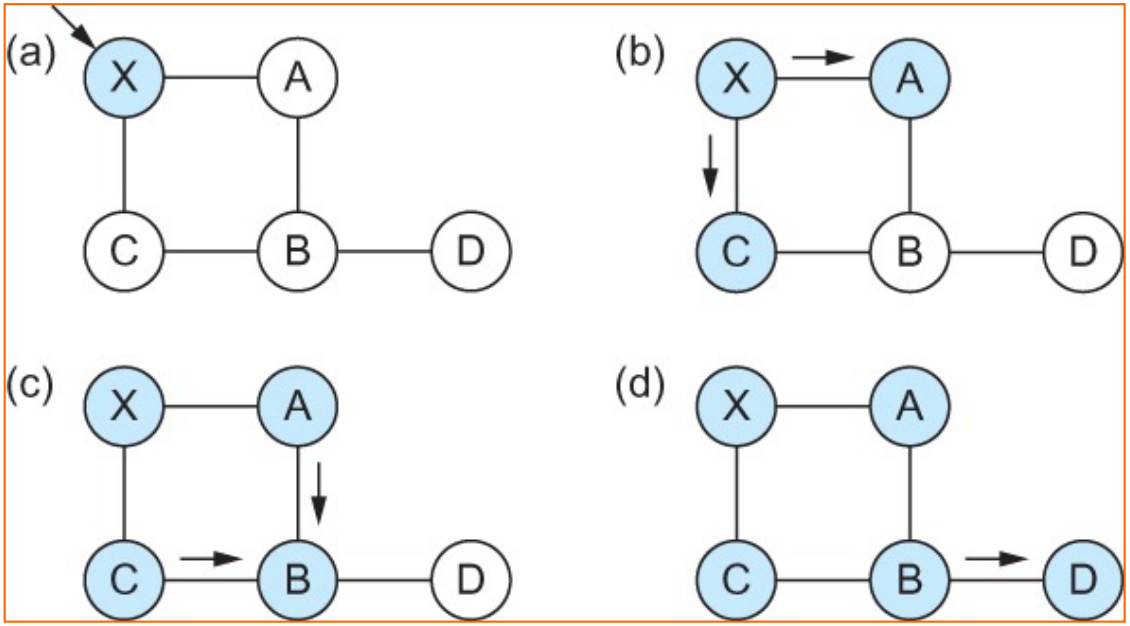
Link State Routing

- **Strategy**: Each node sends the costs of its directly connected links to all the nodes
  - Complementary to DV
- Link State Packet (LSP)
  - id of the node that created the LSP
  - cost of link to each directly connected neighbor
  - sequence number (SEQNO)
  - time-to-live (TTL) for this packet
- Reliable Flooding
  - store most recent LSP from each node
  - forward LSP to all nodes but the one that sent it
  - Start SEQNO at 0; generate new LSP periodically; SEQNO++
  - TTL-- of each stored LSP; discard when TTL=0
  - From hop-to-hop, reliability is provided by acknowledgements and retransmissions

Reliable Flooding

LSP = Link State Packet

a. LSP arrives at node X
b. X floods LSP to A and C
c. A and C flood LSP to B (not X)
d. Flooding is complete



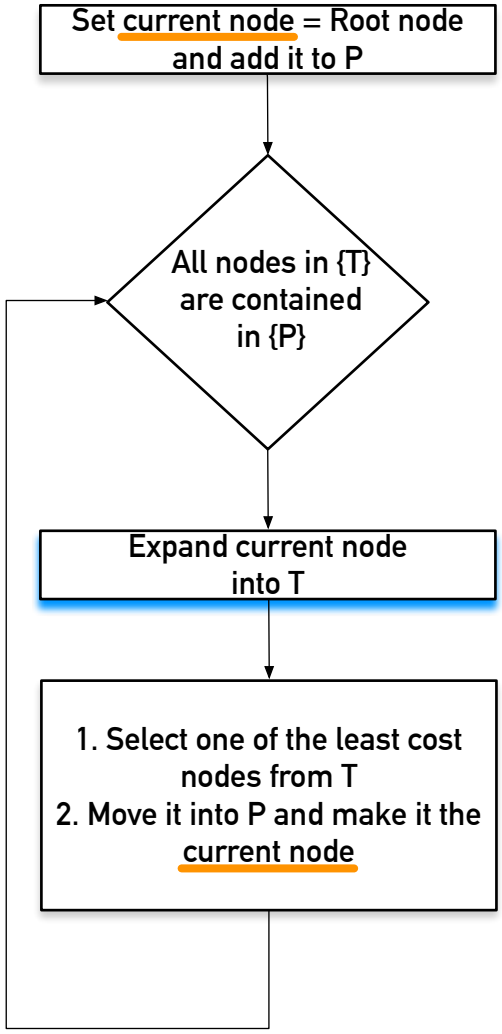
Shortest Path Routing

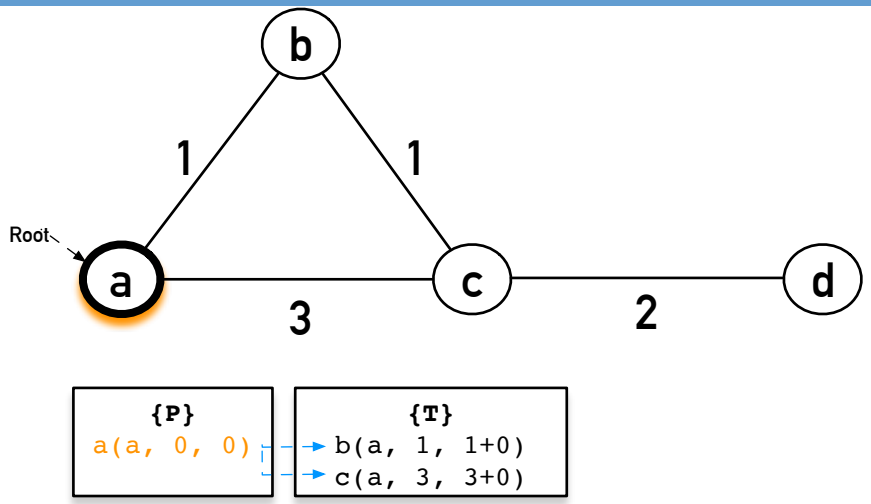
- Each *router* computes its routing table directly from the LSP's it has collected
  - Dijkstra's algorithm Forward Search Algorithm
- Specifically each switch maintains two lists of nodes, known as **Temporary** and **Permanent**
  - **Permanent {P} nodes that do belong to the shortest path from the root**
  - **Temporary {T} nodes: those that have not been added to the shortest path from the root, yet**
- Next node (Current node, partial cost, total cost so far)

  Example: L(K, 3, 17)

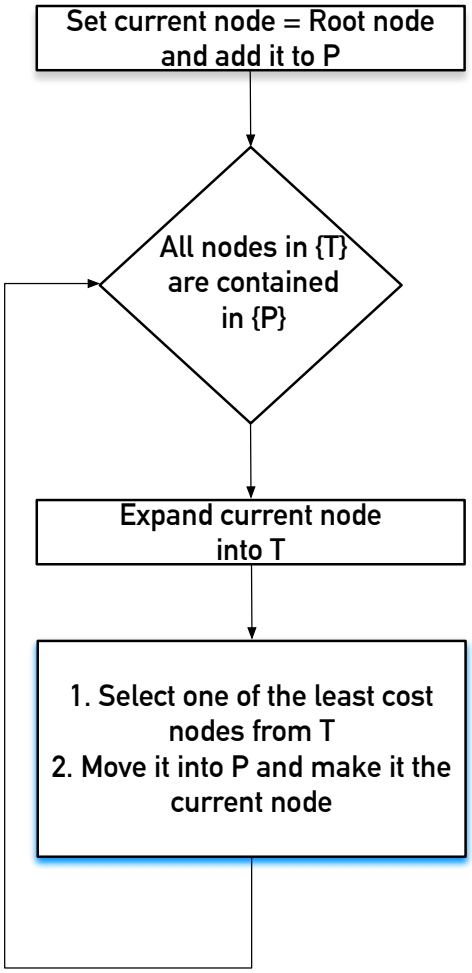
  L can be reached from K at a cost of 3, the total least-cost path, so far is 17 hops
- This is the notation that we are going to use in this course (CN/ADG)
- Beware: it is not the same one used in the textbook by Peterson and Davie

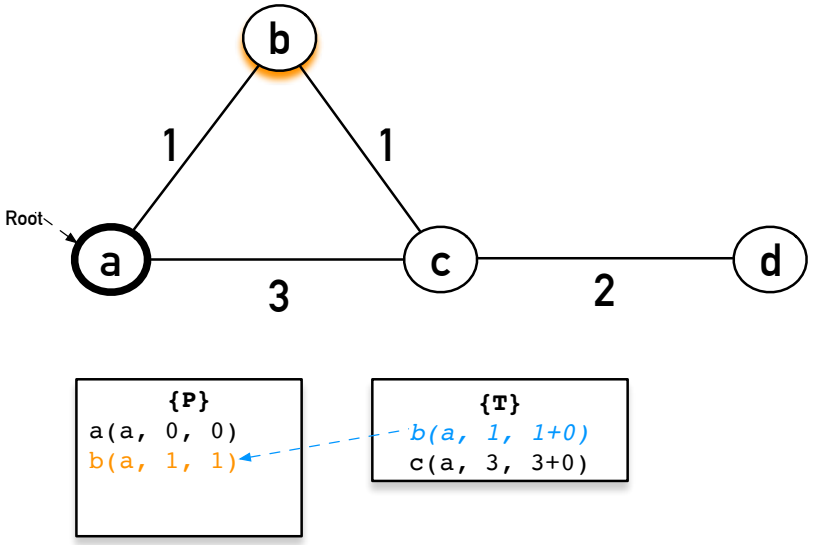
Dijkstra's Algorithm (Fwd Search)





Dijkstra's Algorithm (Fwd Search)

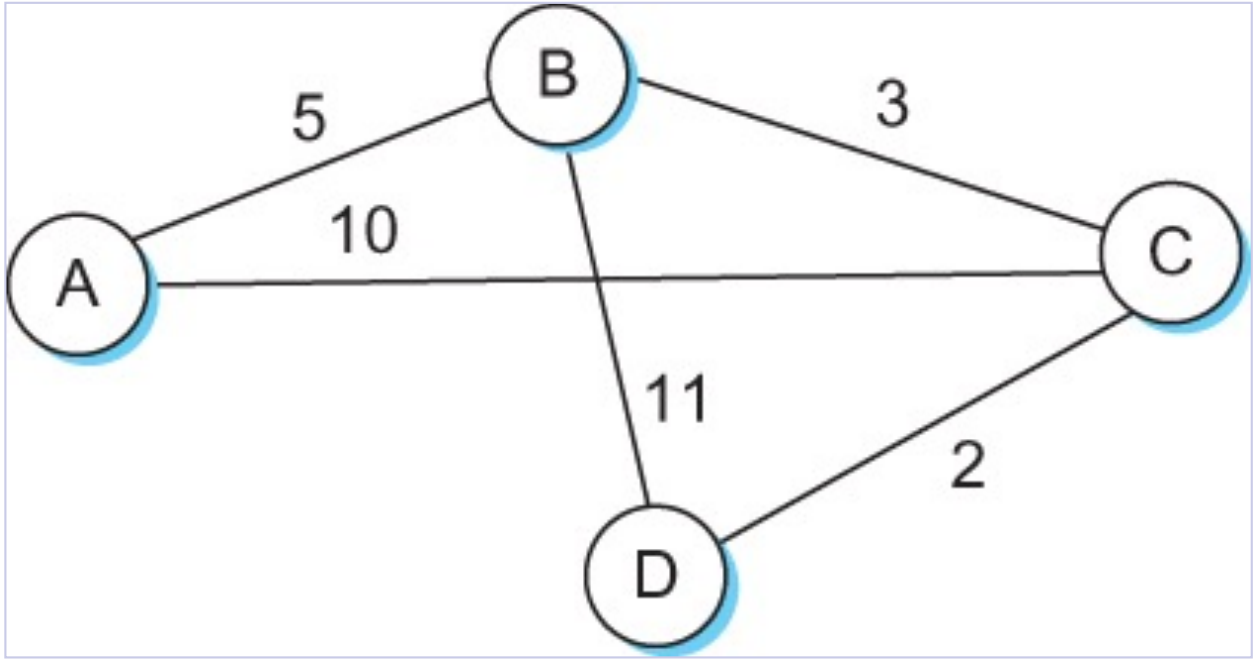




Shortest Path Routing

Example about Link-State routing with the **Forward Search Algorithm (Dijkstra**

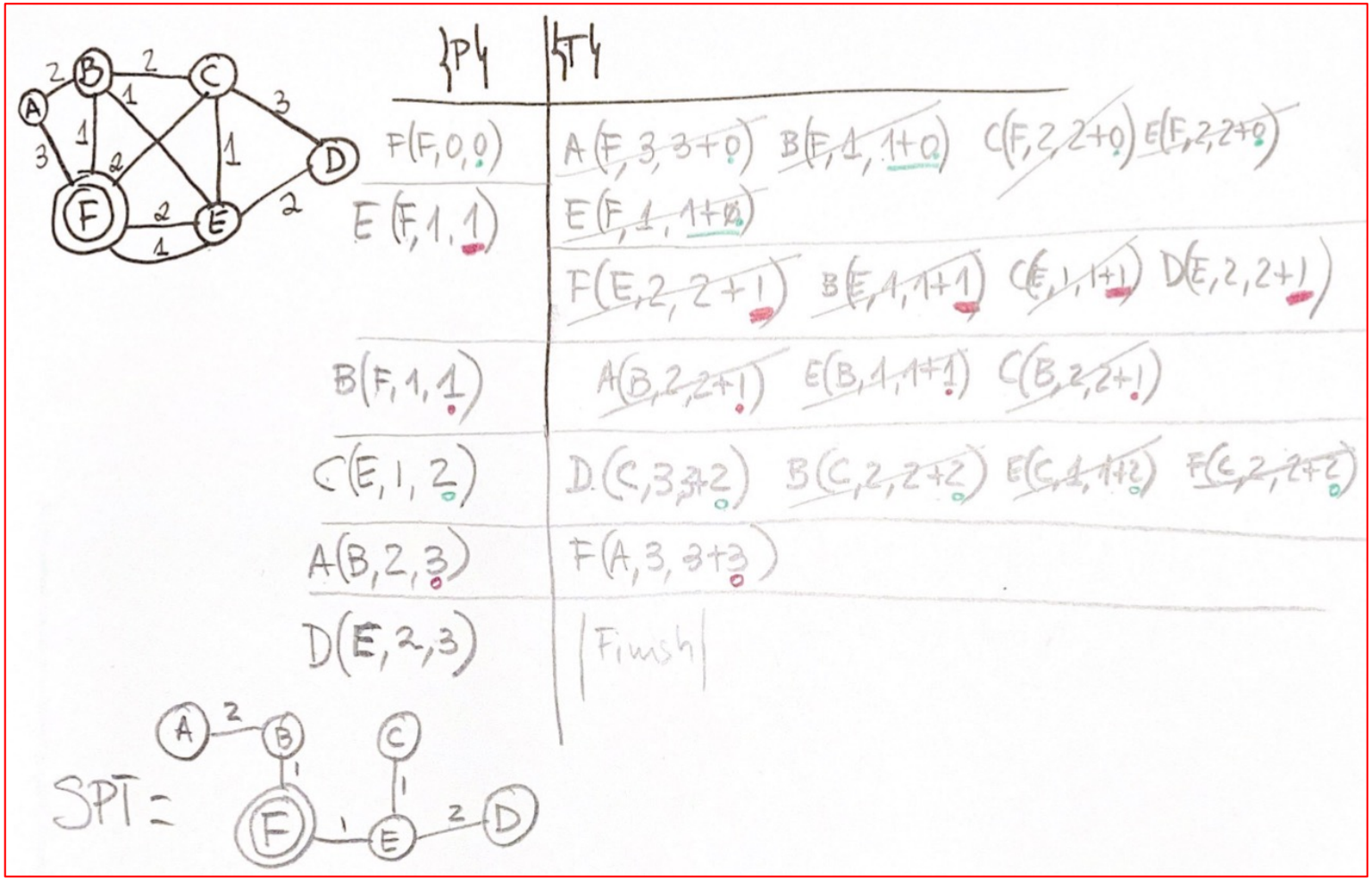
· Calculate the Shortest Path Tree of **node D**



Handwritten exercise

1. Check out the solution

2. Compute the routing table of F



Recommended exercises

- Exams from past terms
- Textbook exercises (Computer Networks, P&D Ch. 3) 46, 48, 49, 62
- Review IP addressing and IP Forwarding Algorithm
- Review the examples and exercises included in this presentation

Summary

- We have looked at some of the issues involved in building scalable and heterogeneous networks by using switches and routers to interconnect links and networks.
- To deal with heterogeneous networks, we have discussed in details the service model of Internetworking Protocol (IP) which forms the basis of today's routers.
- We have discussed in details two major classes of *interior* routing algorithms
  - Distance Vector
  - Link State

The end