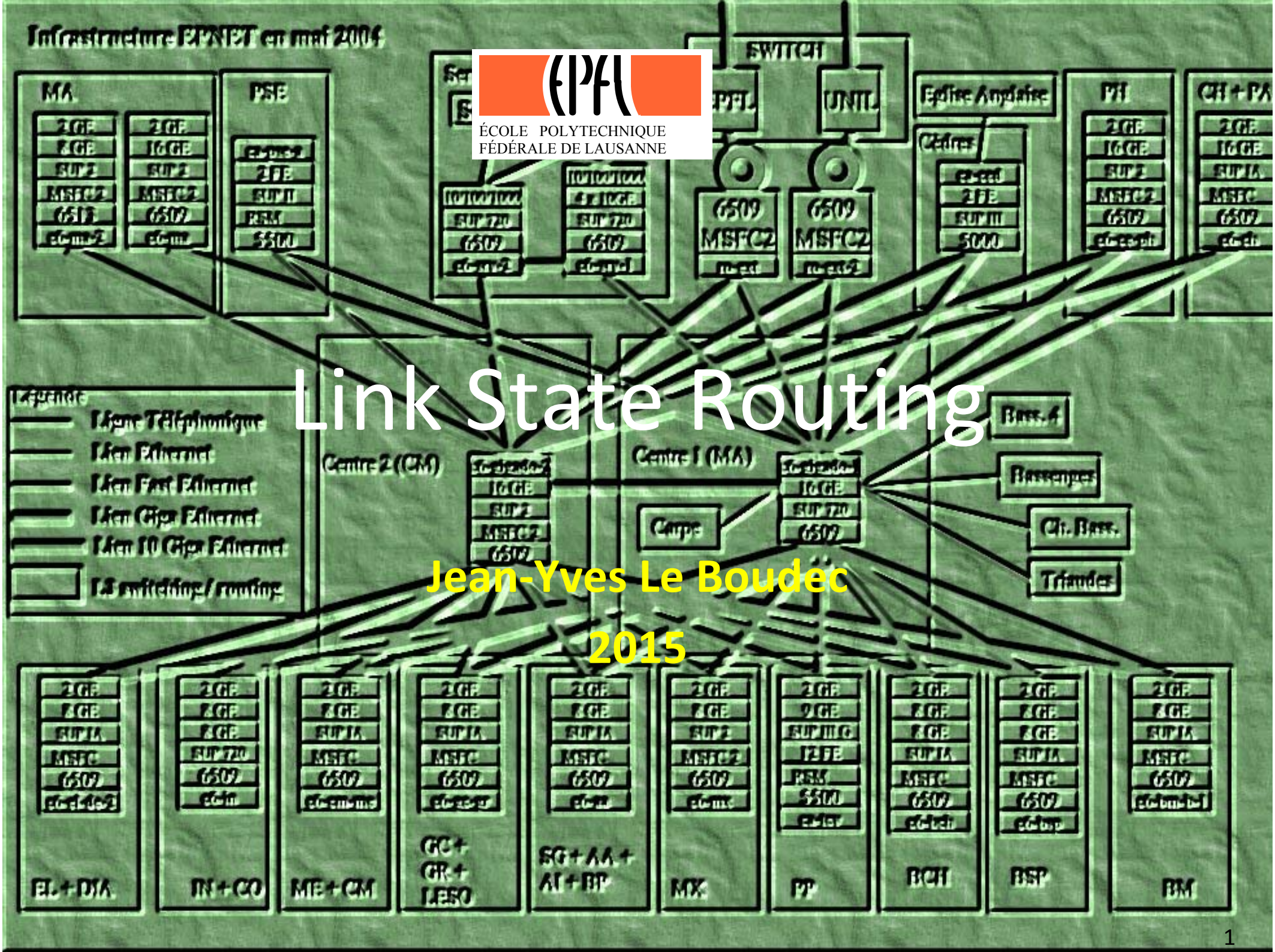


ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Link State Routing

Jean-Yves Le Boudec

2015

Contents

1. Link state
2. OSPF and Hierarchical routing with areas
3. Dynamic metrics and Braess paradox

1. Link State Routing

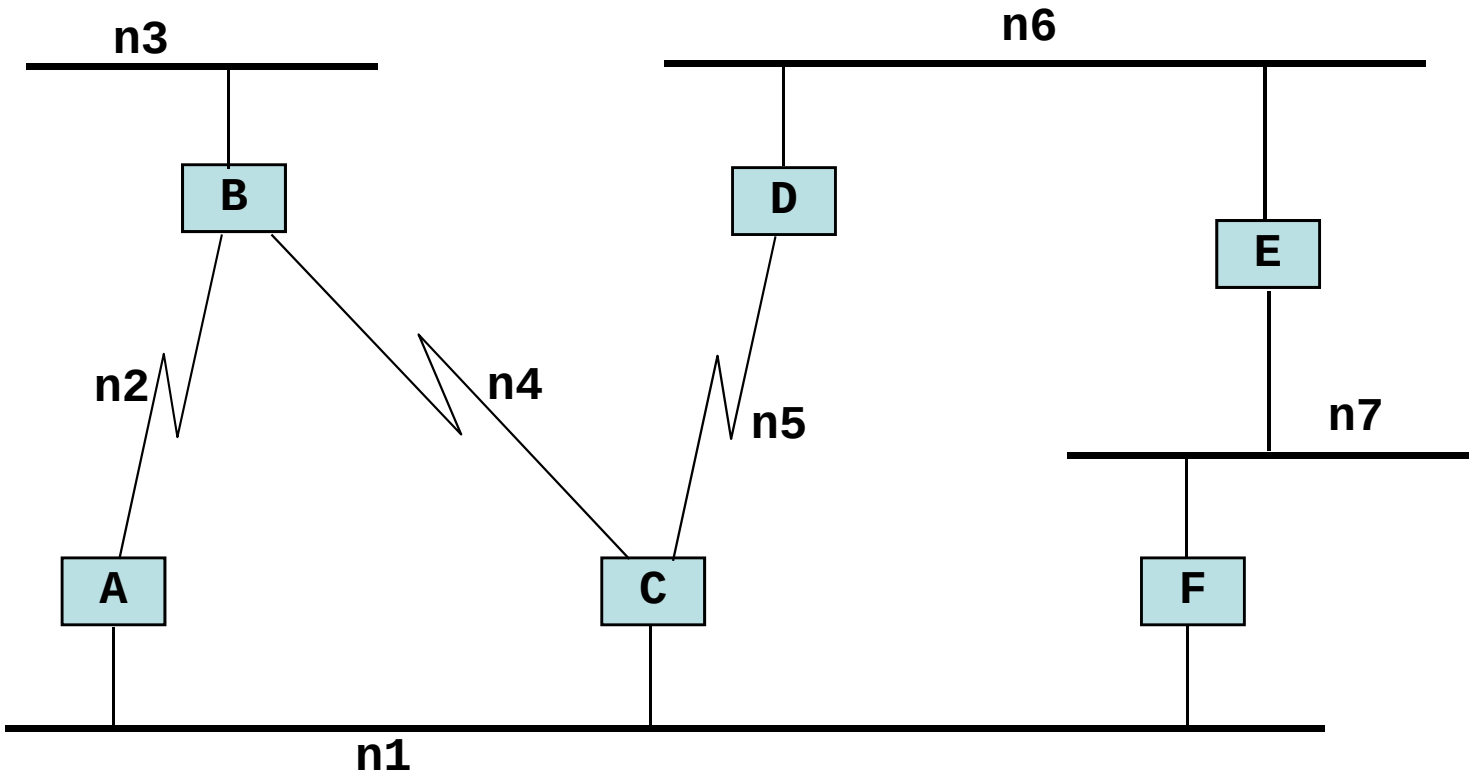
- **Principle** of link state routing
  - each router keeps a topology database of whole network
  - link state updates flooded, or multicast to all network
  - routers compute their routing tables based on topology often uses Dijkstra's shortest path algorithm
- Used in
  - OSPF (Open Shortest Path First, layer 2), IS-IS (similar to OSPF)
  - TRILL (Transparent Interconnection of Lots of Links), SPB (Shortest Path Bridging), layer 2

(a) Topology Database Synchronization

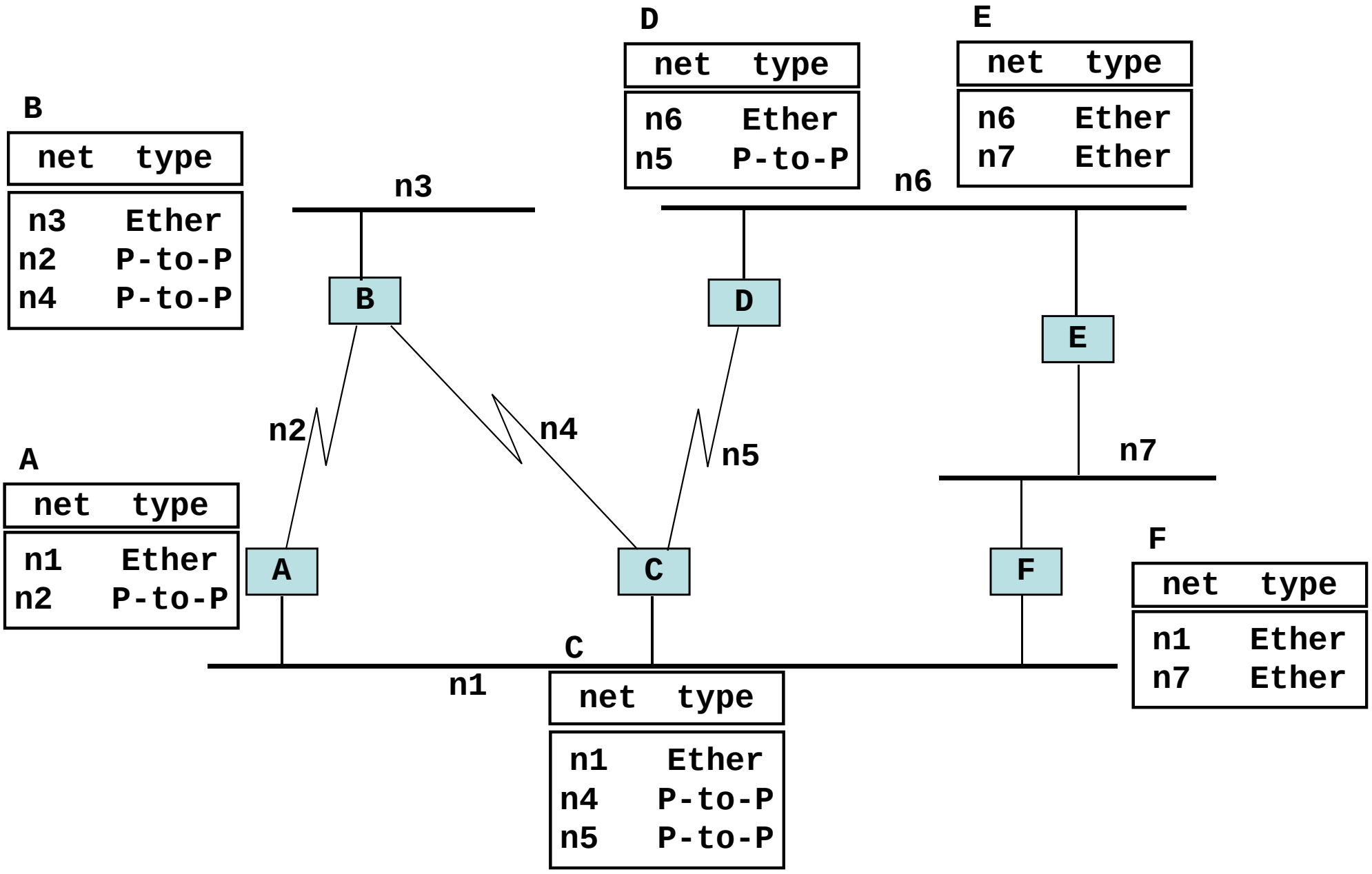
- Neighbouring nodes synchronize before starting any relationship
  - Hello protocol; keep alive
  - initial synchronization of database
  - description of all links (no information yet)
- Once synchronized, a node accepts link state advertisements
  - contain a sequence number, stored with record in the database
  - only messages with new sequence number are accepted
  - accepted messages are flooded to all neighbours
  - sequence number prevents anomalies (loops or blackholes)

Example network

- Each router knows directly connected networks



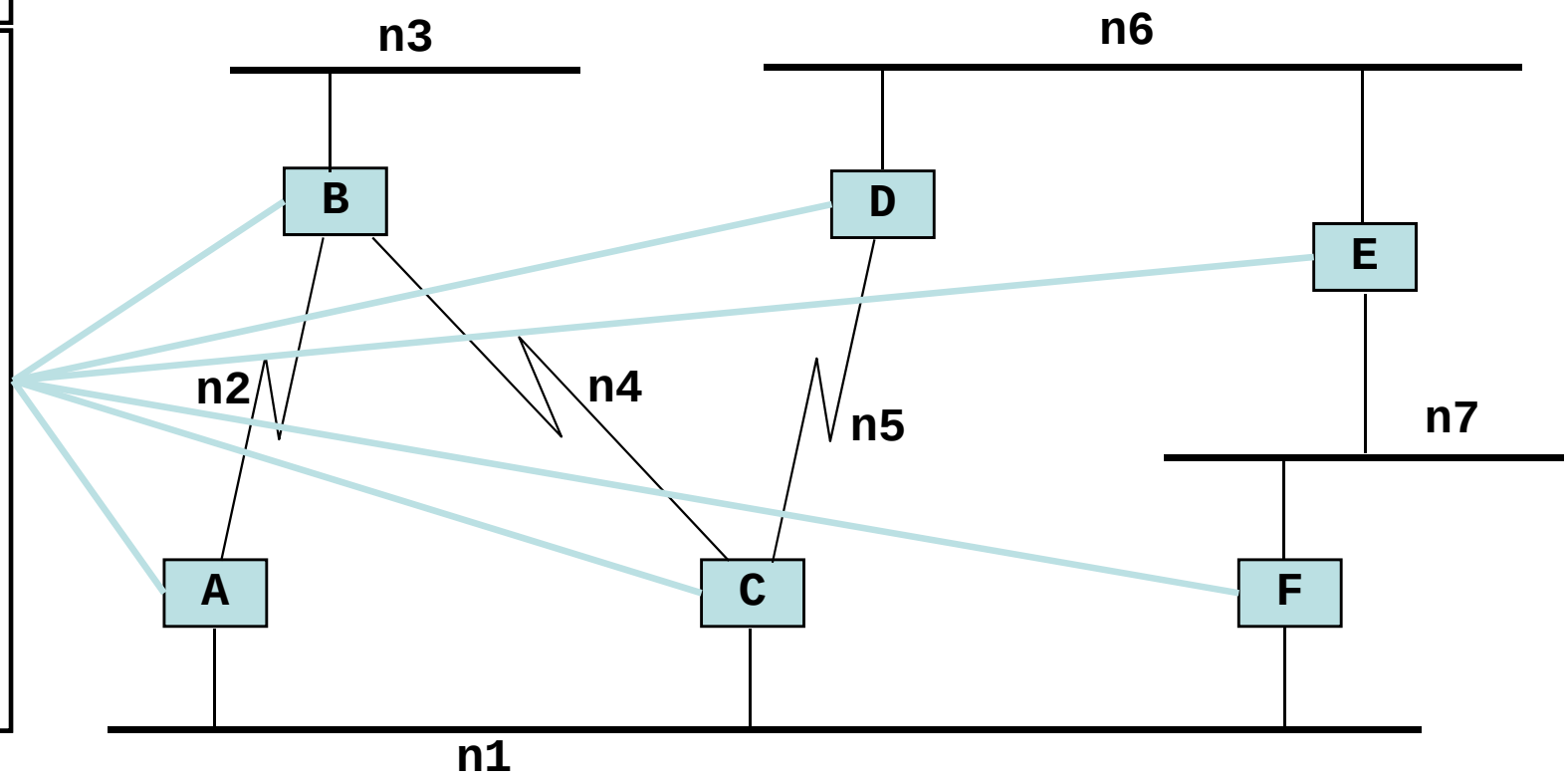
Initial routing tables



After Flooding

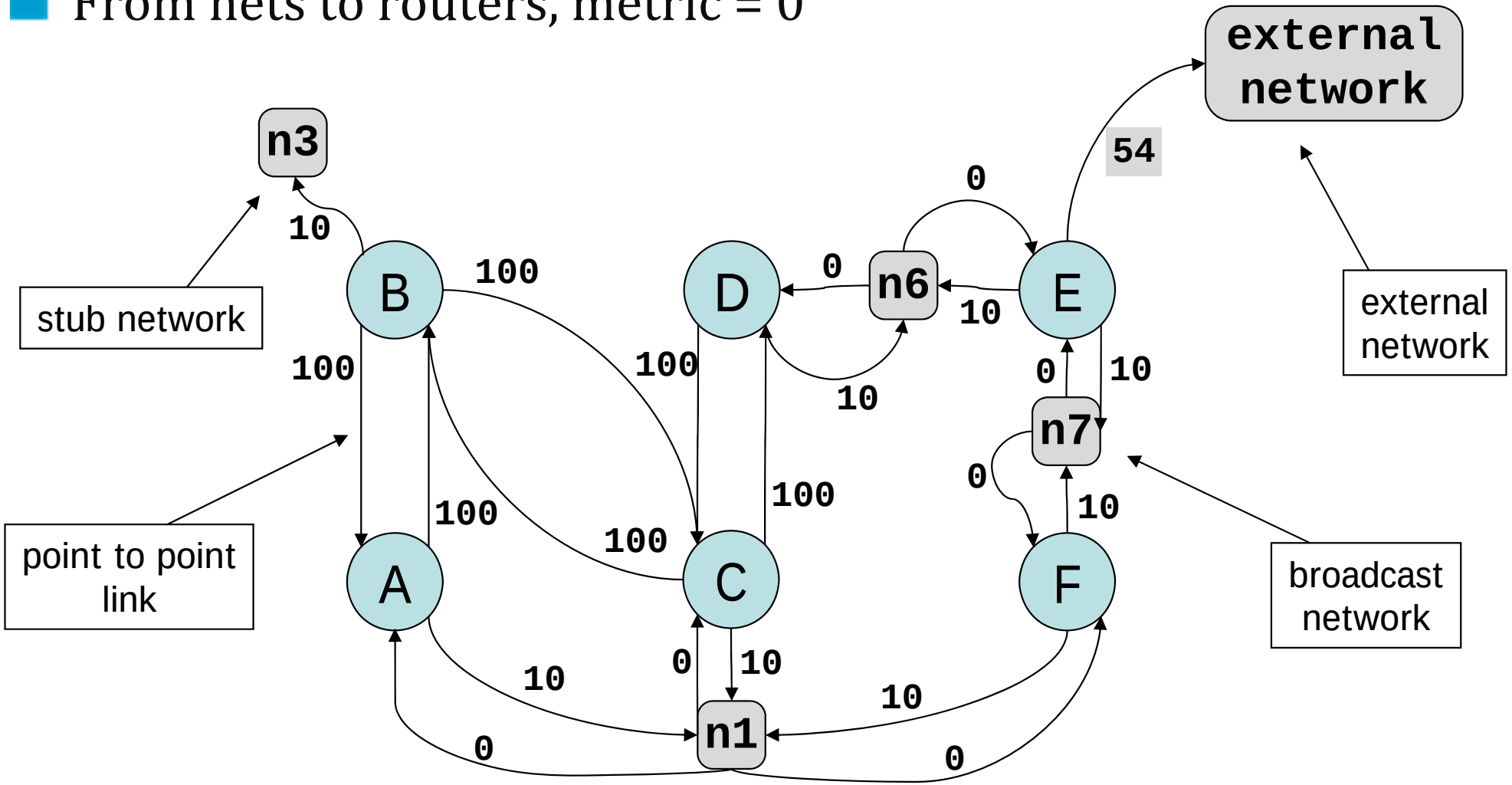
- The local metric information is flooded to all routers
- After convergence, all routers have the same information

rtr	net	cost
A	n1	10
A	n2	100
B	n3	10
B	n2	100
B	n4	100
C	n1	10
C	n4	100
C	n5	100
D	n6	10
D	n5	100
E	n6	10
E	n7	10
F	n1	10
F	n7	10



(b) From Topology Database to Graph

- Arrows routers-to-nets with a given metric
  - except point-to-point, stub, and external networks
- From nets to routers, metric = 0

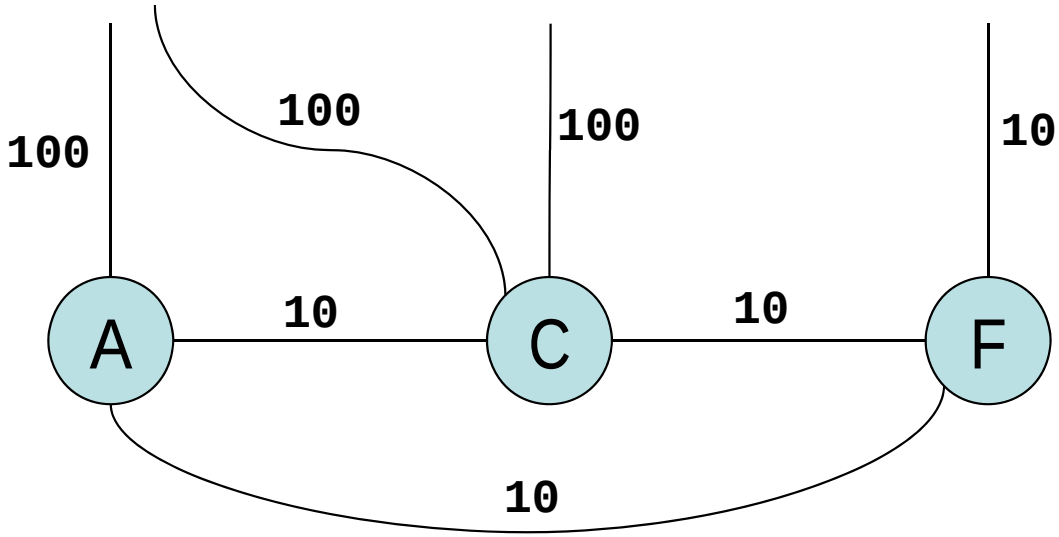


(b) Path Computation

- Performed locally, based on topology database
- Computes one or several best paths to every destination from this node
- Best Path = shortest for OSPF
- OSPF uses Dijkstra's shortest path
  - the best known algorithm for centralized operation
- Paths are computed independently at every node
  - synchronization of databases guarantees absence of persistent loops
  - every node computes a shortest path tree ***rooted at self***

Simplified graph

- Only arrows with metrics between routers
- Every node executes the shortest path computation on the graph – same graph, but different sources

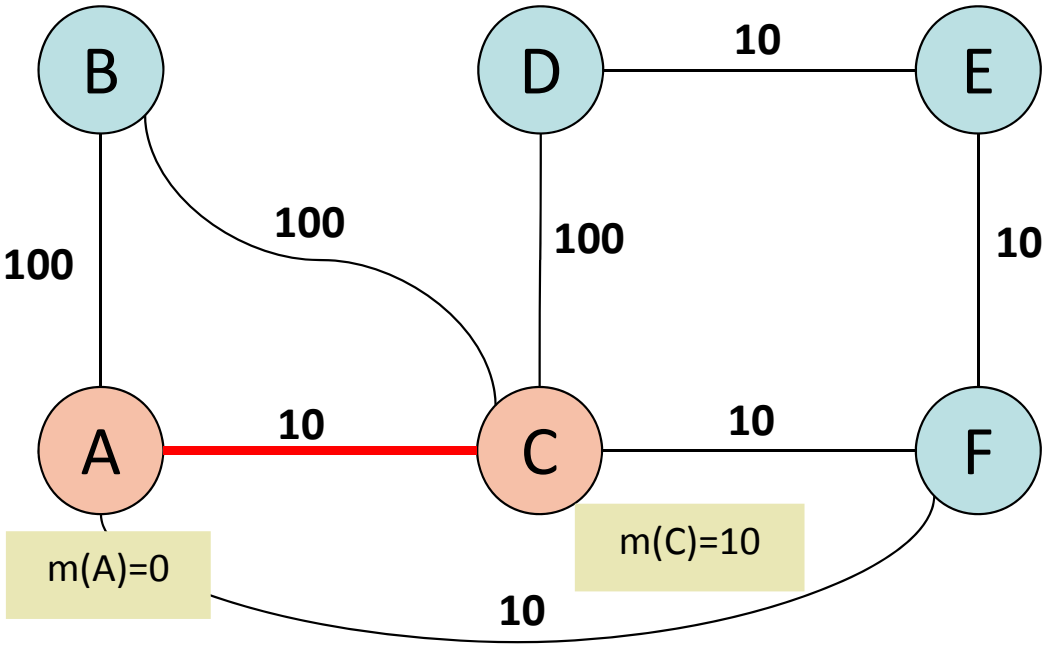


Dijkstra's Shortest Path Algorithm

- The nodes are `0...N` and the algorithm computes best paths from node `0`
- `c(i,j)` is the cost of `(i,j)`,
- `pred(i)` is the predecessor of node `i` on the tree `M` being built
- `m(j)` is the distance from node `0` to node `j`.

```
m(0) = 0; M = {0};
for k=1 to N {
   find (i0, j0) that minimizes m(i) + c(i,j),
                        with i in M, j not in M
   m(j0) = m(i0) + c(i0, j0)
   pred(j0) = i0
   M = M ∪ {j0}
}
```

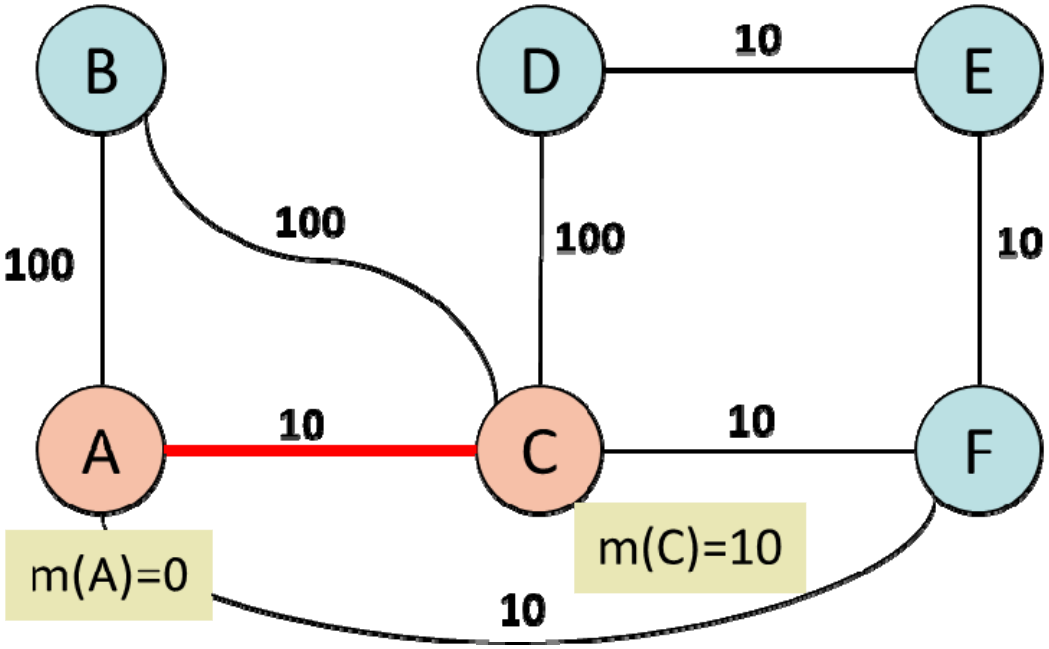
Example: Dijkstra at A



init: M = { A }

step 1:
i0=A
j0=C
m(C)=10
M = {A, C}

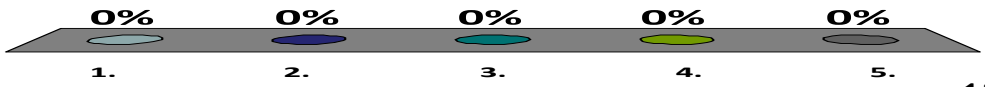
Next, which node is added to M ?



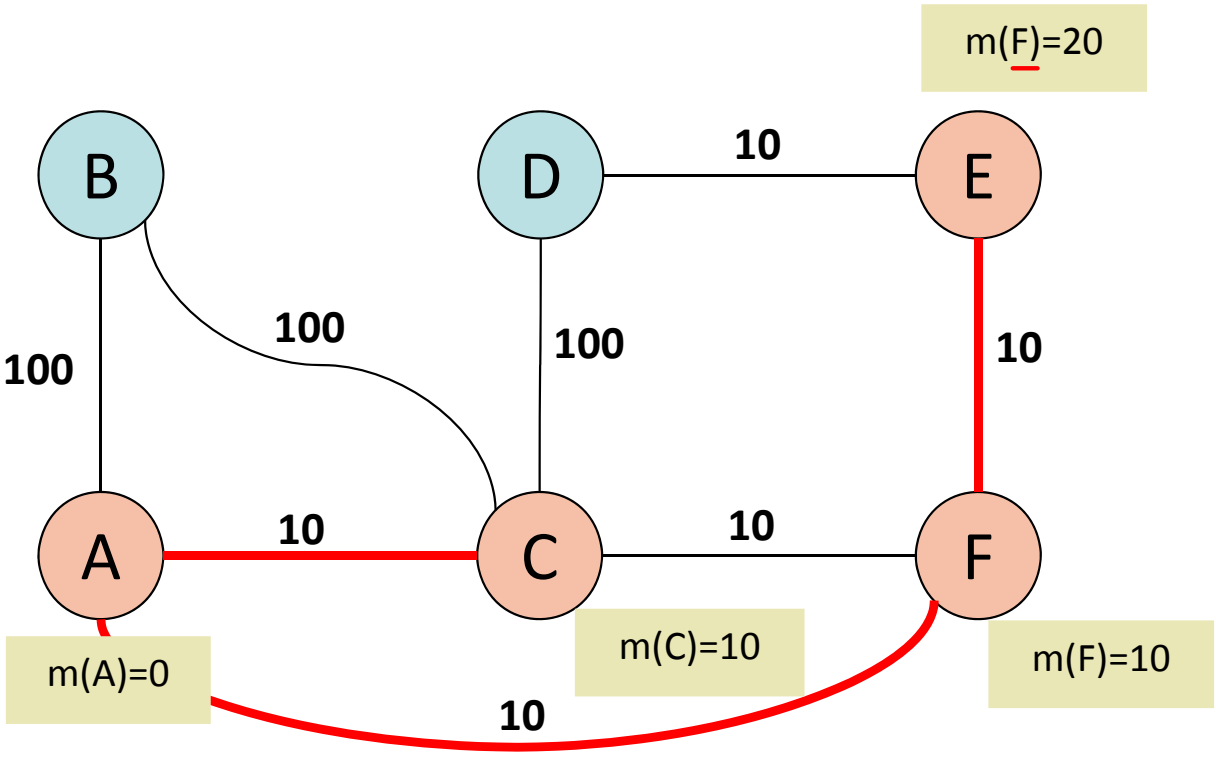
init: M = { A }

step 1:
i0=A
j0=C
m(C)=10
M = {A, C}

1. F
2. E
3. D
4. B
5. I don't know

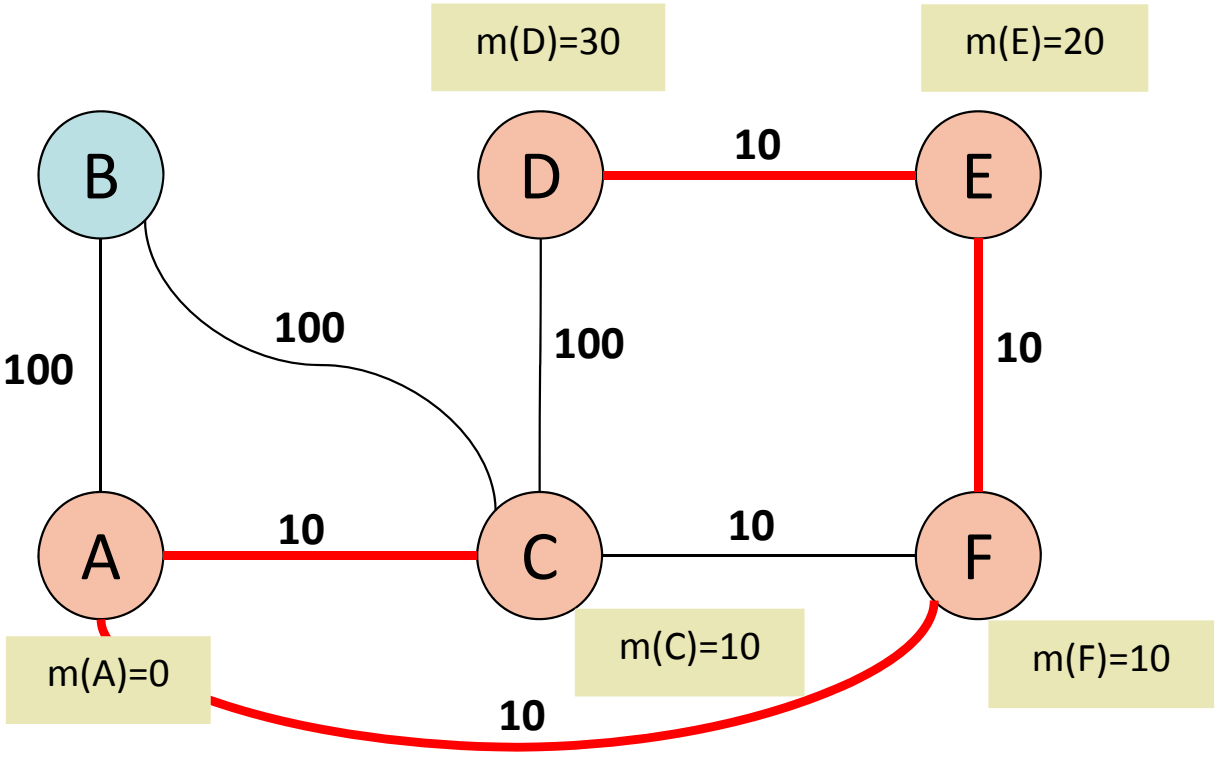


Example: Dijkstra at A



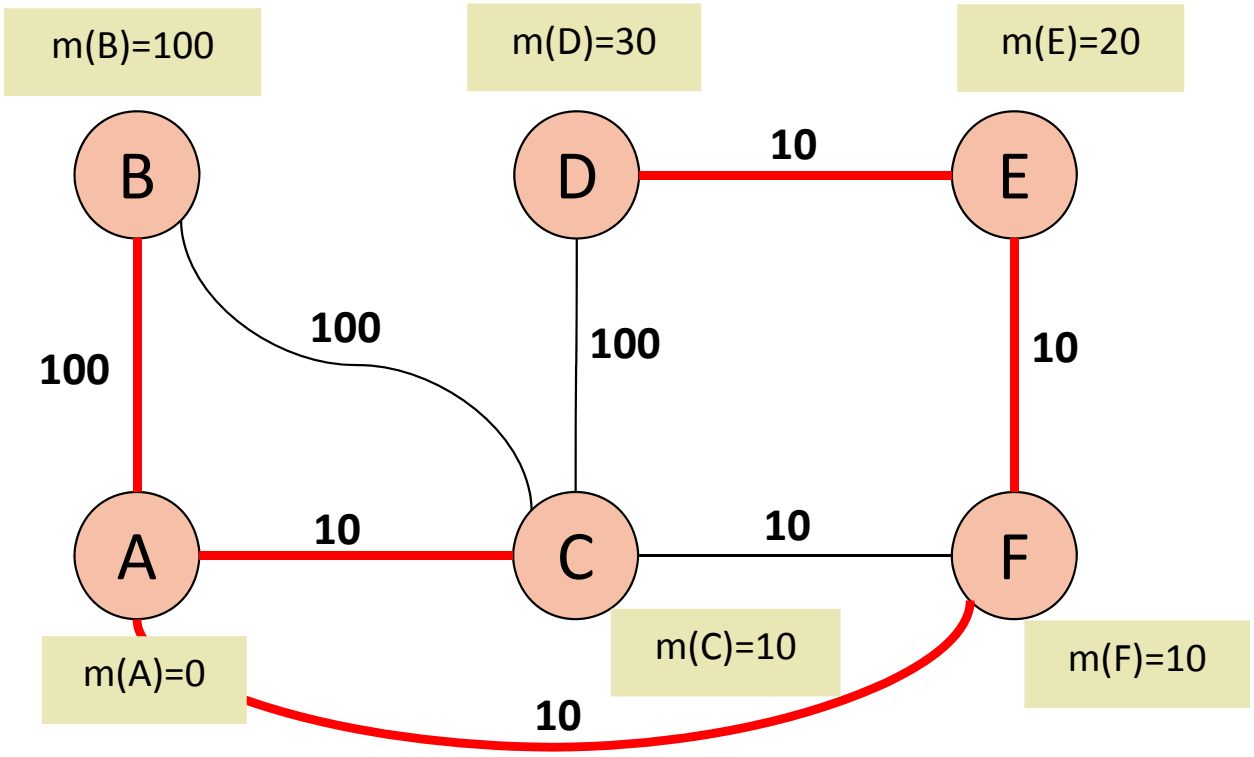
i0=F
j0=E
m(E)=20
M = {A,C,F,E}

Example: Dijkstra at A



i0=E
j0=D
m(D)=40
M = {A,C,F,E,D}

Example: Dijkstra at A



i0=A

j0=B

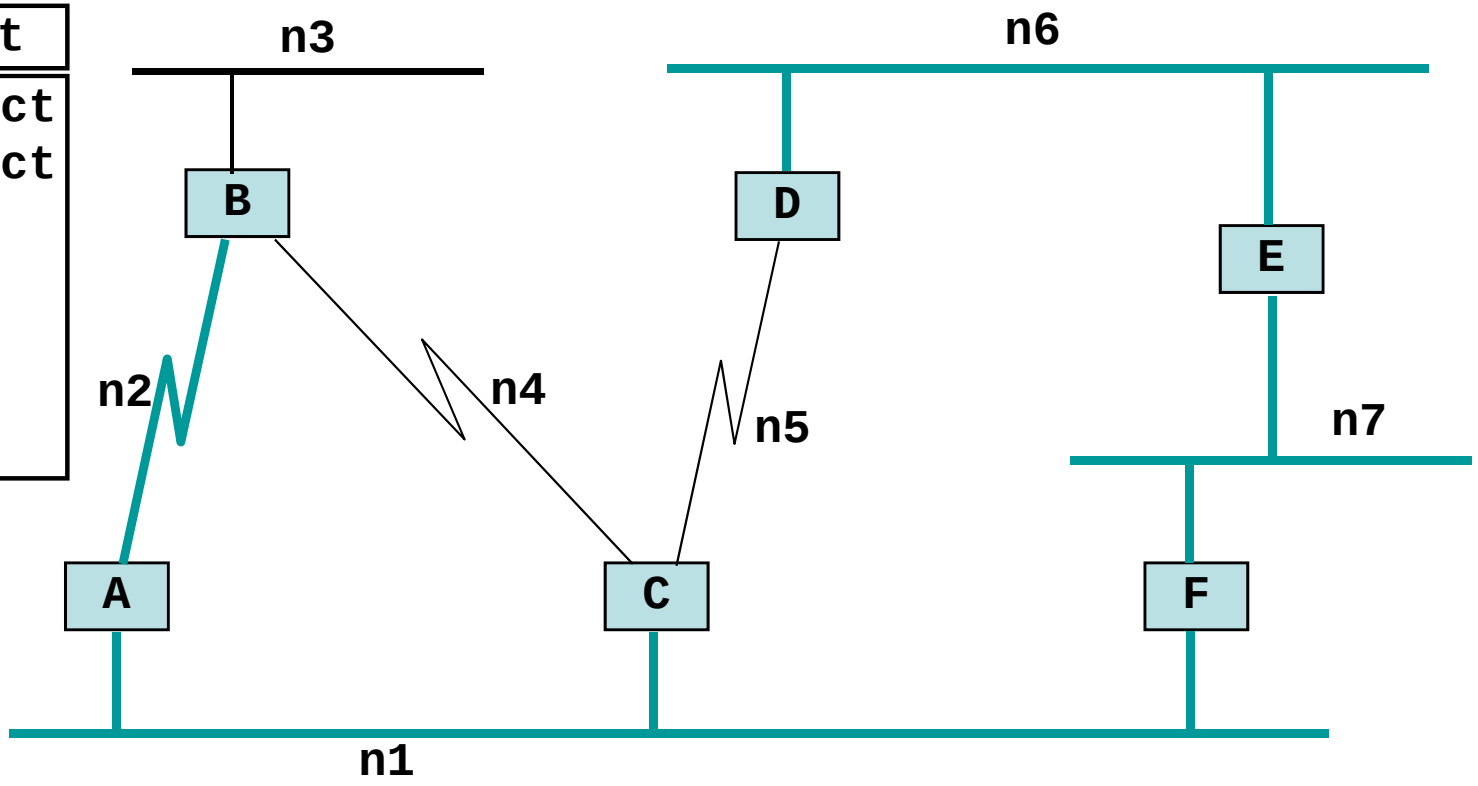
m(B)=100

M = {A,C,F,E,D,B}

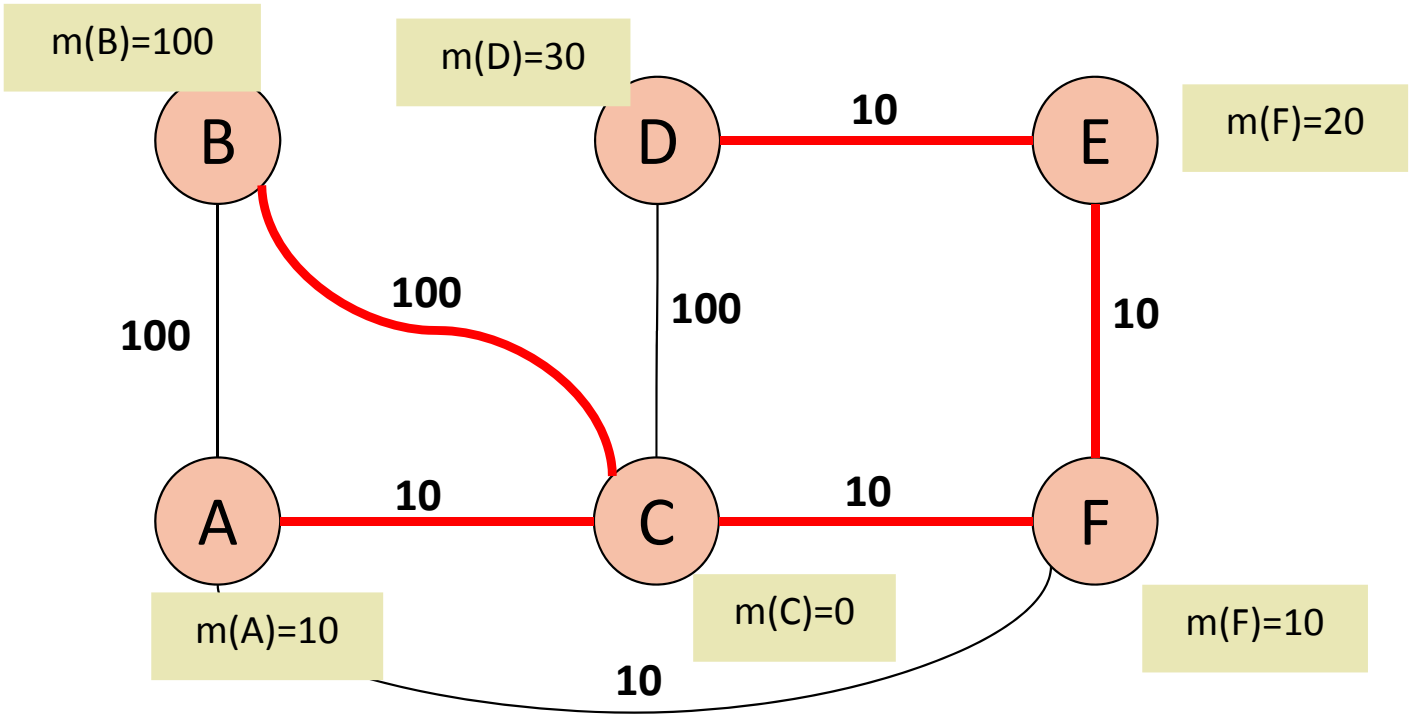
Routing table at A

A

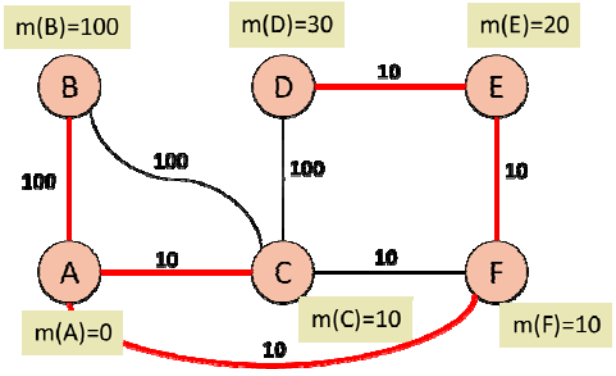
net	next
n1	direct
n2	direct
n3	B
n4	C
n5	C
n6	F
n7	F



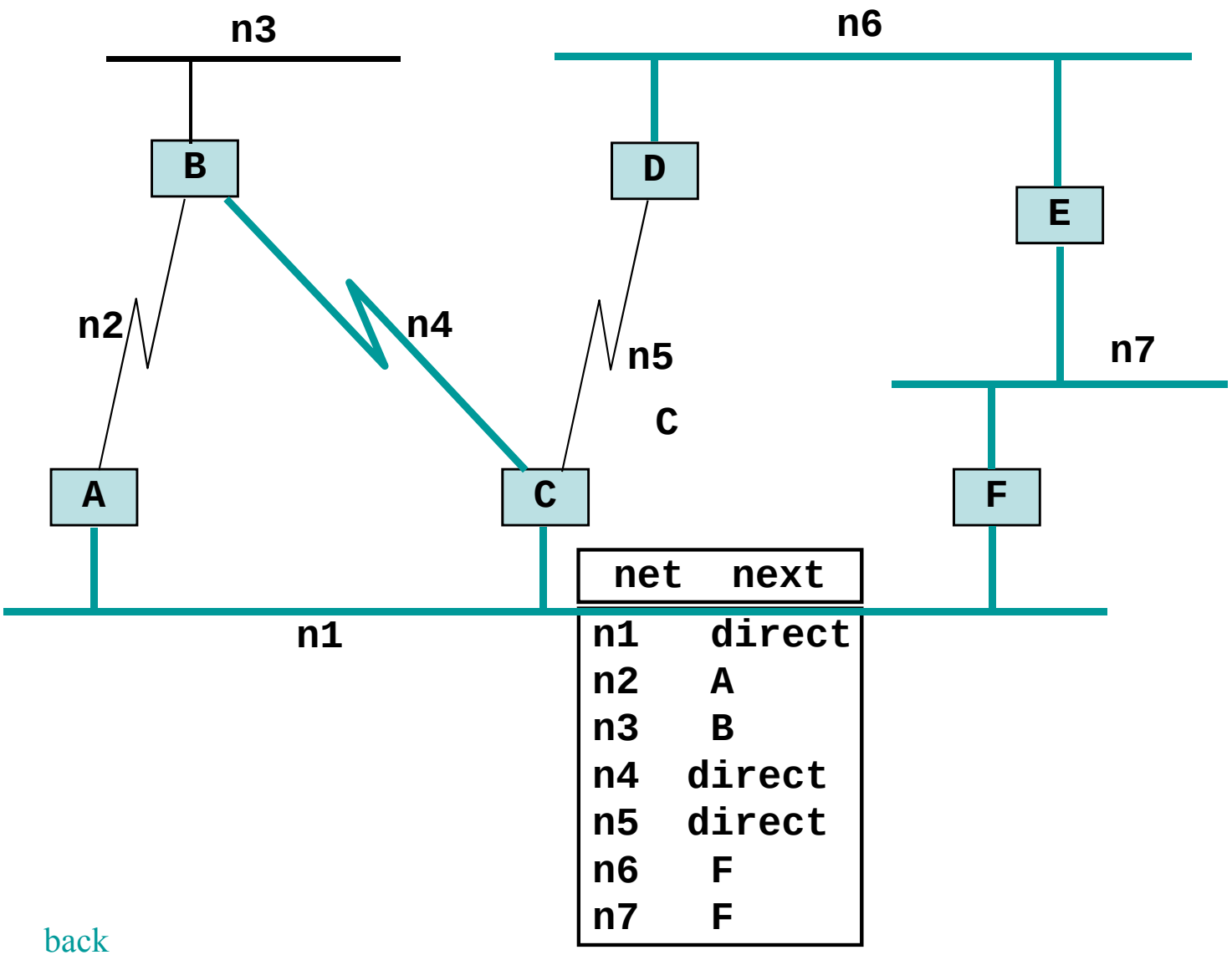
Dijkstra's Algorithm At C



At A



Routing Tables at C



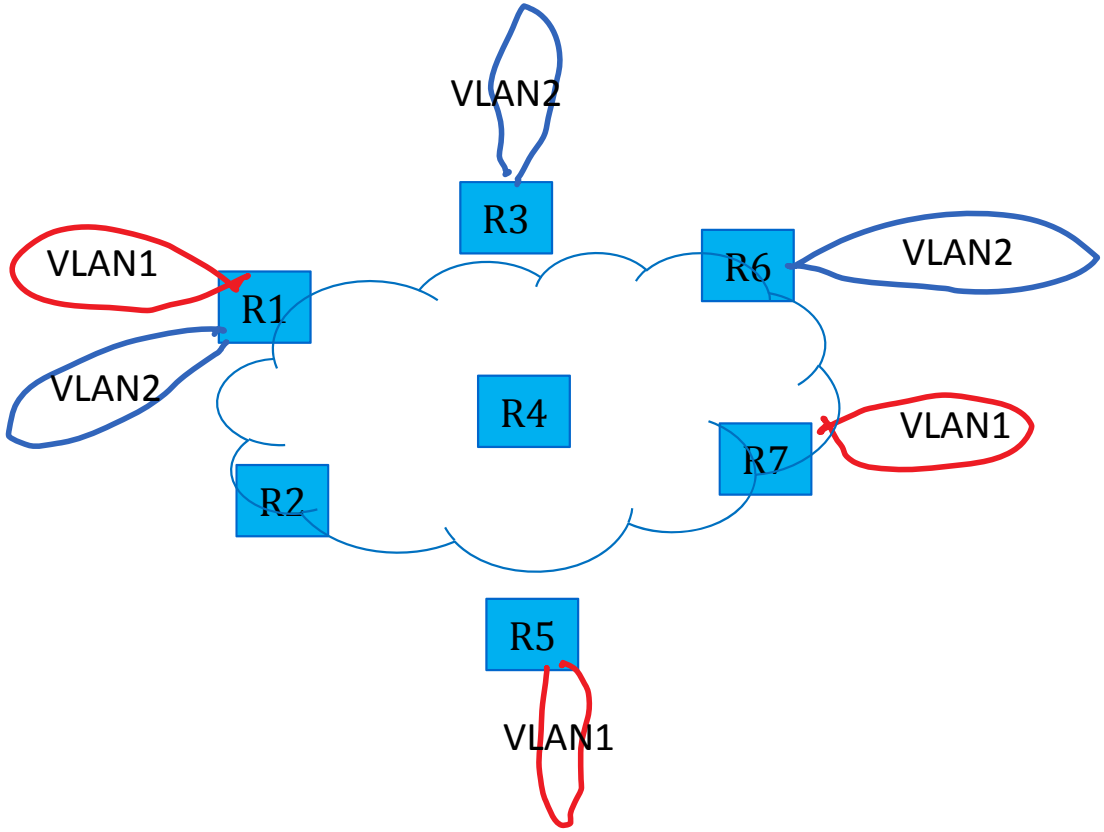
back

Changes to Topology

- Changes to topology (e.g. link failures) cause routers to send new Link State Advertisements
- All routers update their topology database and propagate the change to all their neighbours
- LSA sequence number is used to avoid loops in the propagation
  - One router that has received an LSA already does not propagate it further
- Changes to topology database trigger re-computation of shortest-paths

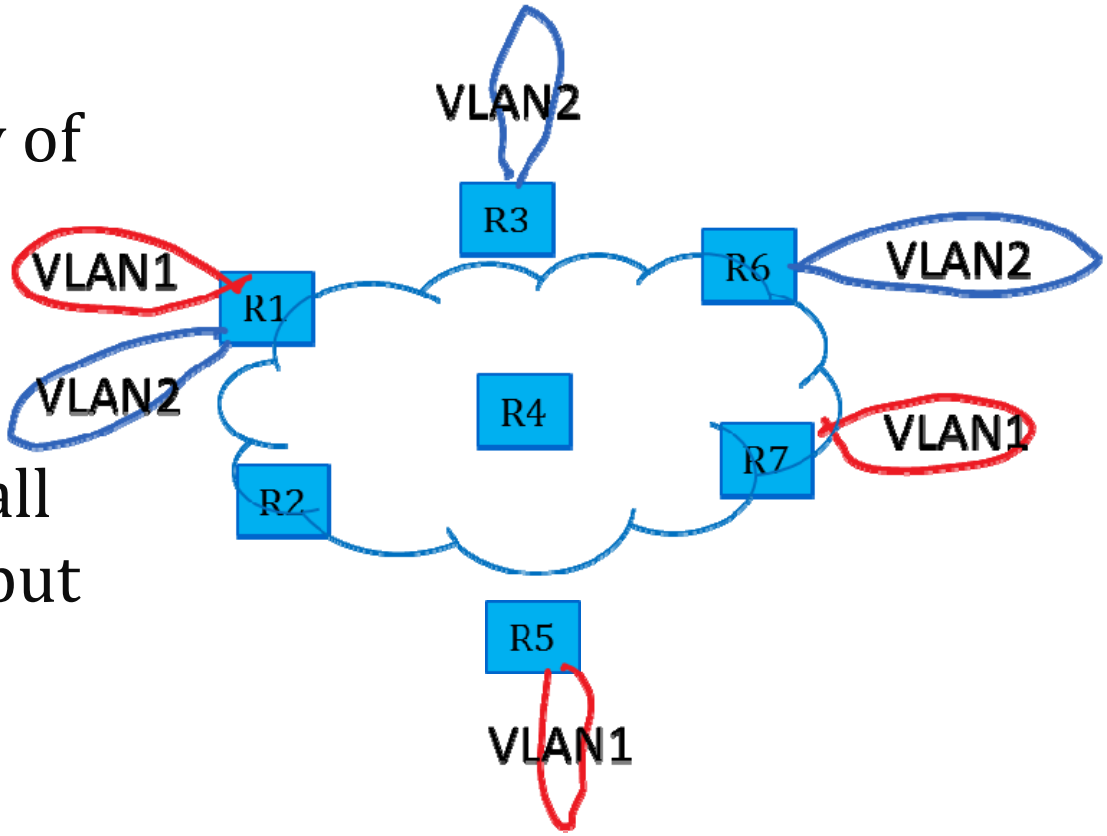
Link State Routing can be used for non-standard operations

- Example: assume you want to bridge VLANs across a campus
- One solution: tunnel MAC packets in IP
  Problem: automatic creation of tunnels



Can you imagine a solution using Link State Routing in R1, R2, … ?

1. Routers R1, R2 … discover which VLAN is active on any of their ports and put this information in the topology database
2. Routers R1, R2 … overhear all MAC source addresses and put the information in the topology database
3. Both of these solutions seem bad to me
4. I don't know



2. The OSPF Protocol and Hierarchical Routing

- OSPF (Open Shortest Path First)
  - IETF standard for internal routing
  - used in large networks (ISPs), in MPLS and in TRILL (Cisco VLAN interconnection)

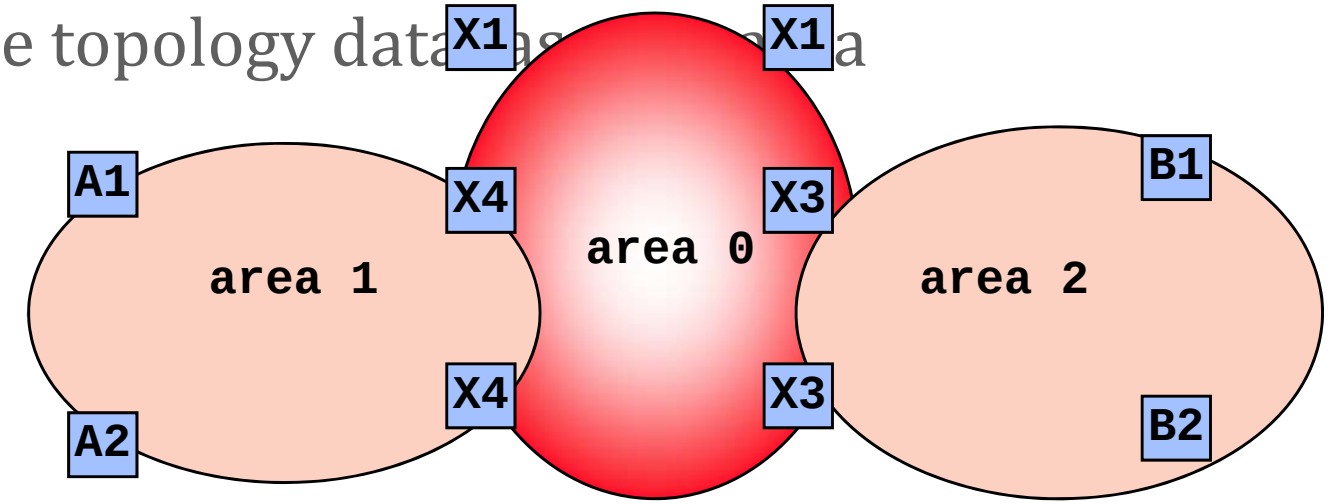
- OSPF uses Link State protocol + Hierarchical

OSPF and hierarchical routing

- Why divide large networks?
- Cost of computing routing tables
  - update when topology changes
  - size of DB, update messages grows with the network size
- Use *hierarchical routing* to limit the scope of updates and computational overhead
  - divide the network into several areas
  - independent route computing in each area
  - inject aggregated information on routes into other areas
- We explain hierarchical routing the OSPF way
  - IS-IS does things a bit differently

Hierarchical Routing

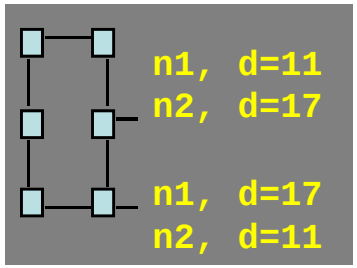
- An OSPF domain is configured in *areas*
  - one *backbone area* (area 0)
  - plus zero or several non backbone areas (areas numbered other than 0)
- All inter-area traffic goes through area 0
  - strict hierarchy
- Inside one area: link state routing as seen earlier
  - one topology data[illegible]a



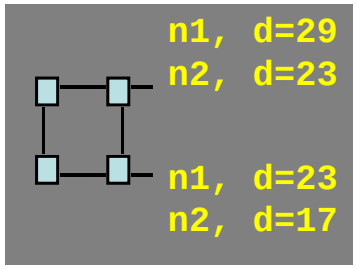
Principles

- Routing method used in the higher level:
  - *distance vector*
  - no problem with loops - one backbone area
- Mapping of higher level nodes to lower level nodes
  - area border routers (inter-area routers) belong to both areas
- Inter-level routing information
  - summary link state advertisements (LSA) from other areas are injected into the local topology databases

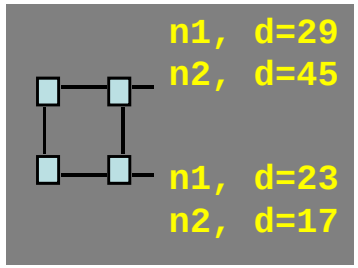
Assume networks n1 and n2 become visible at time 0. Which are the topology databases at A1 ?



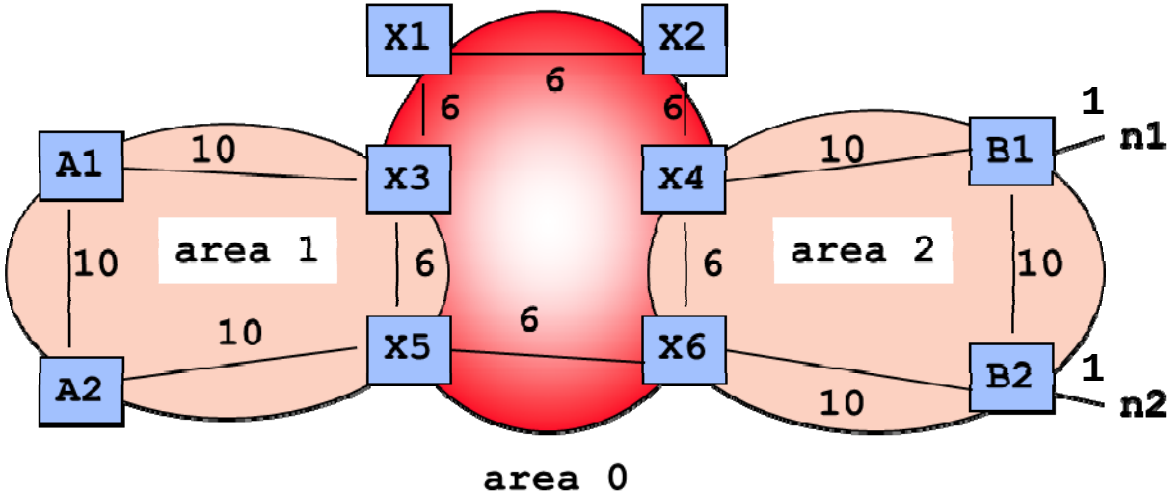
D1



D2



D3



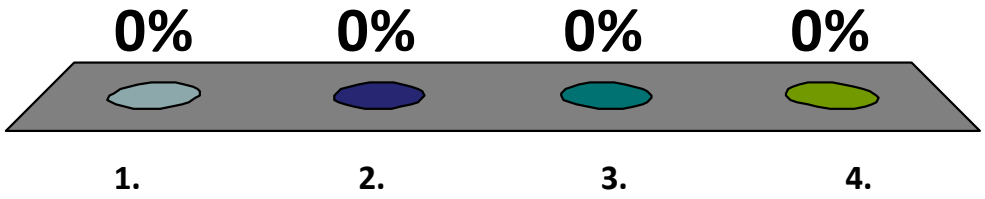
1. D1
2. D2
3. D3
4. I don't know

Comments

- Distance vector computation causes none of the RIP problems
  - strict hierarchy: no loop between areas
- External and summary LSA for all reachable networks are present in all topology databases of all areas
  - most LSAs are external
  - can be avoided in configuring some areas as terminal: use **`default`** entry to the backbone
- Area partitions require specific support
  - partition of non-backbone area is handled by having the area 0 topology database keep a map of all area connected components
  - partition of backbone cannot be repaired; it must be avoided; can be handled by backup virtual area 0 links through non backbone area

3. Dynamic Metrics
Does a routing protocol minimize network utility ?

1. Yes, because it minimizes the cost to destination
2. Yes if TCP is used because it ensures fairness
3. No
4. I don't know

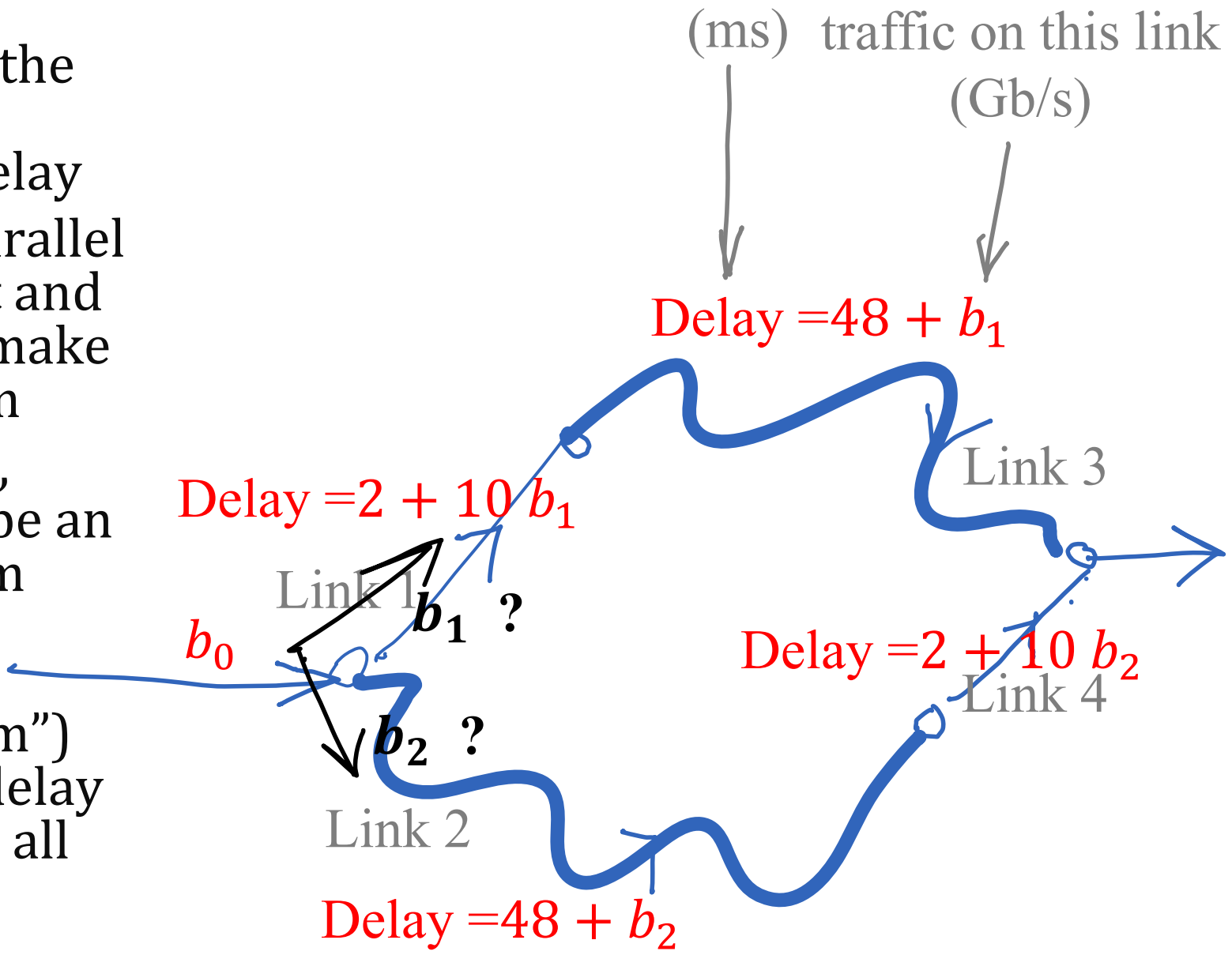


Dynamic Metrics

- Some proposed to use dynamic metrics for improving over shortest path
- high load on a link => high cost => link is less used
  - This is used by EIGRP
- But there may be some issues → Braess paradox

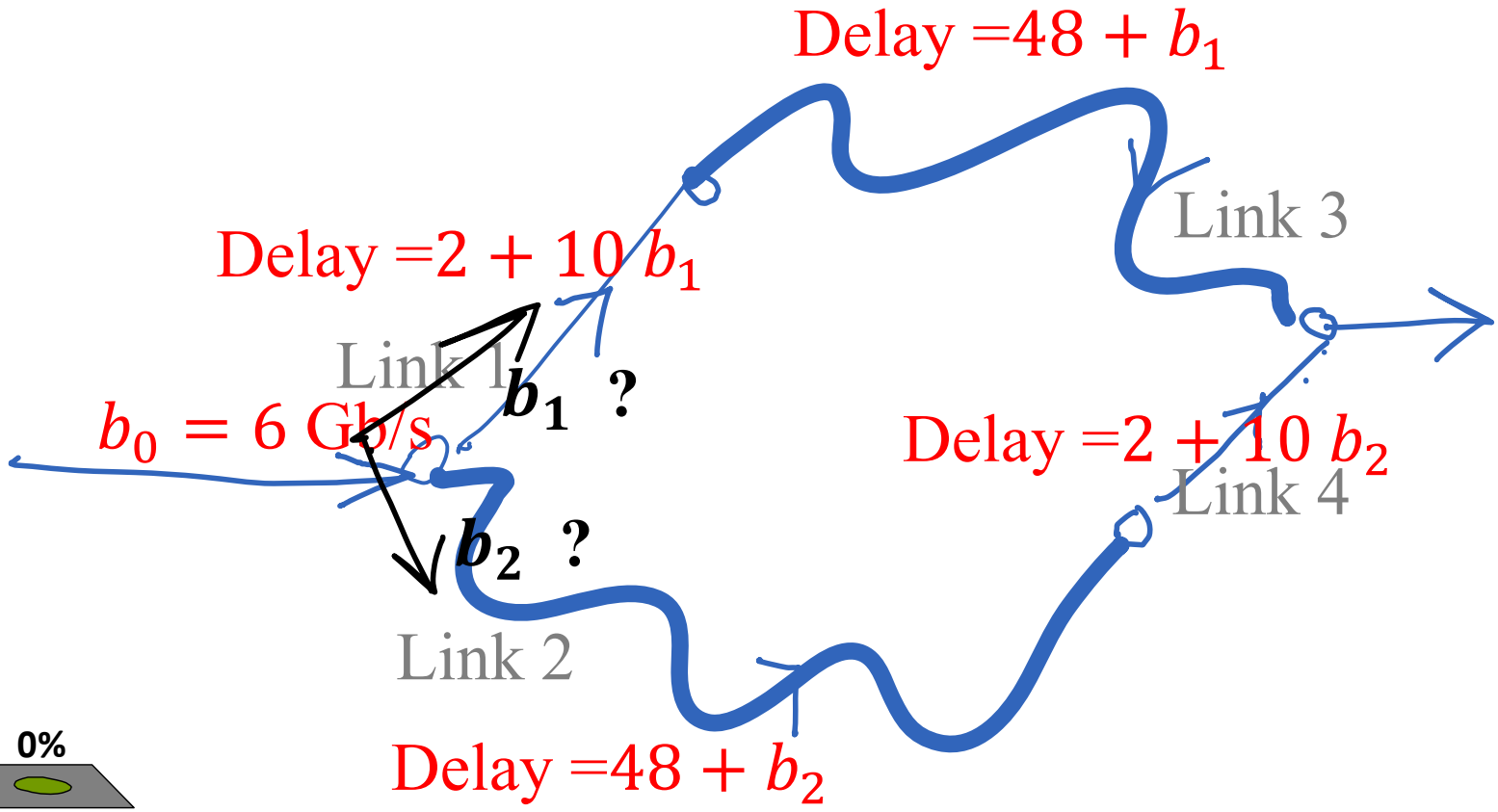
Least Delay Routing and Wardrop Equilibirum

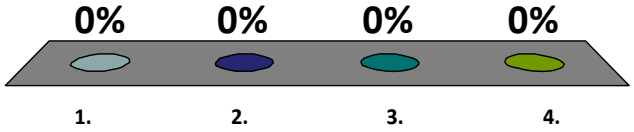
- Assume all flows pick the route with shortest delay
- Assume parallel paths exist and flows can make use of them
- Eventually, there will be an equilibrium (called "Wardrop Equilibrium") such that delay is equal on all competing routes



Which is a Wardrop Equilibrium for this Network ?

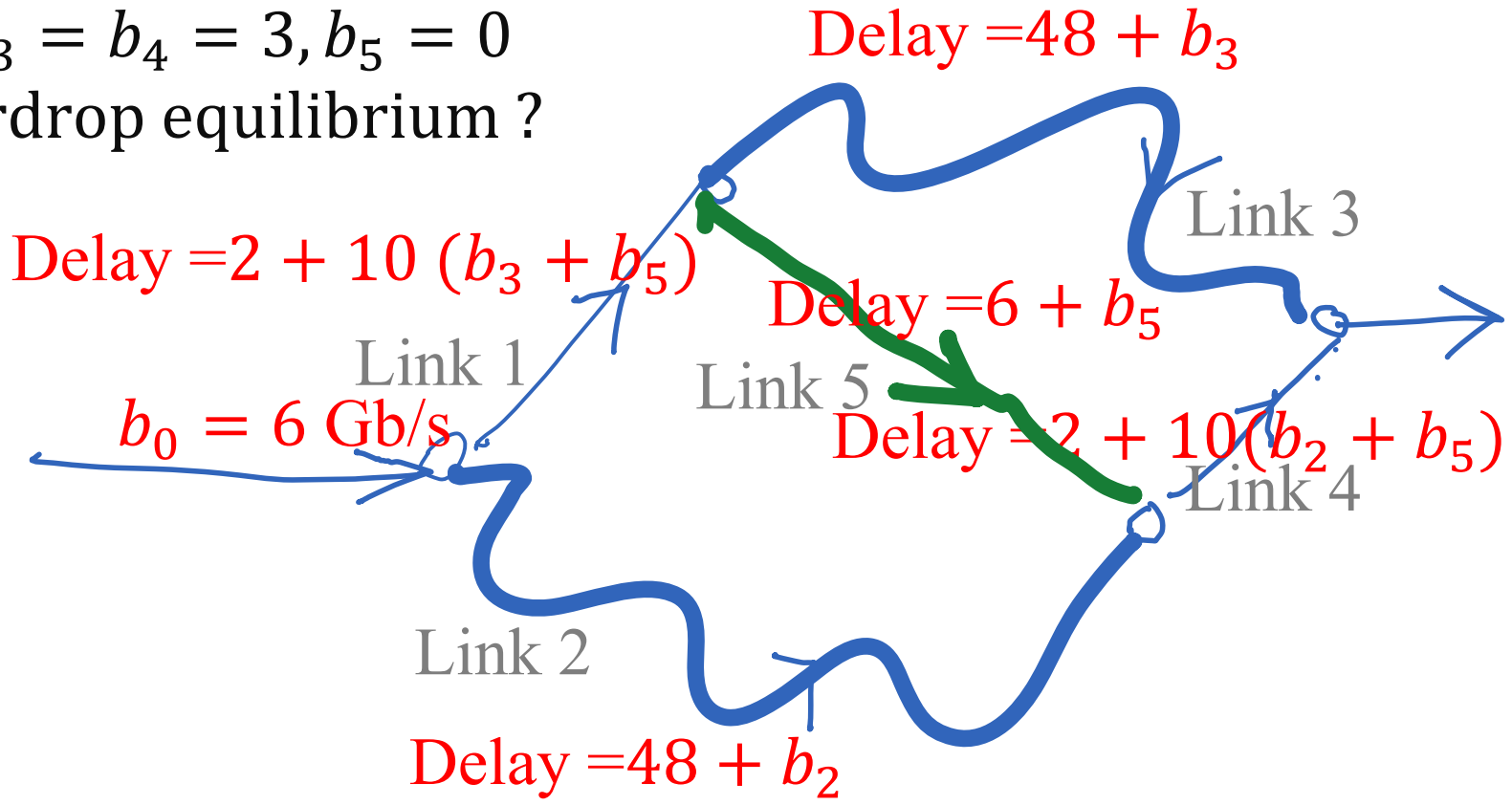
1. $b_1 = 1, b_2 = 5$
2. $b_1 = 3, b_2 = 3$
3. $b_1 = 5, b_2 = 1$
4. None of the above





Now introduce link 5

- Link 5 has delay function $6 + b_5$ i.e. short delay and high capacity
- There are now 3 paths: 13, 154 and 24
- Assume we start from previous equilibrium
  $b_1 = b_2 = b_3 = b_4 = 3, b_5 = 0$
  Is this a Wardrop equilibrium ?

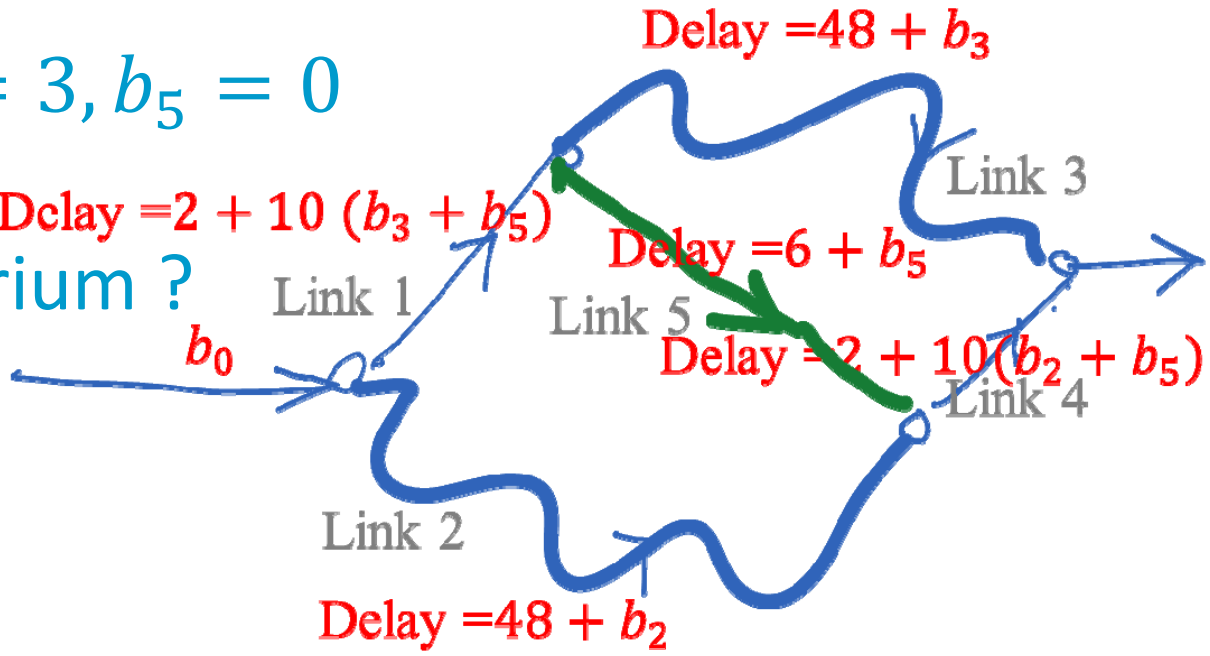


Is

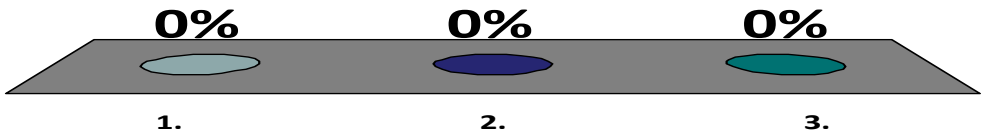
$b_1 = b_2 = b_3 = b_4 = 3, b_5 = 0$

a

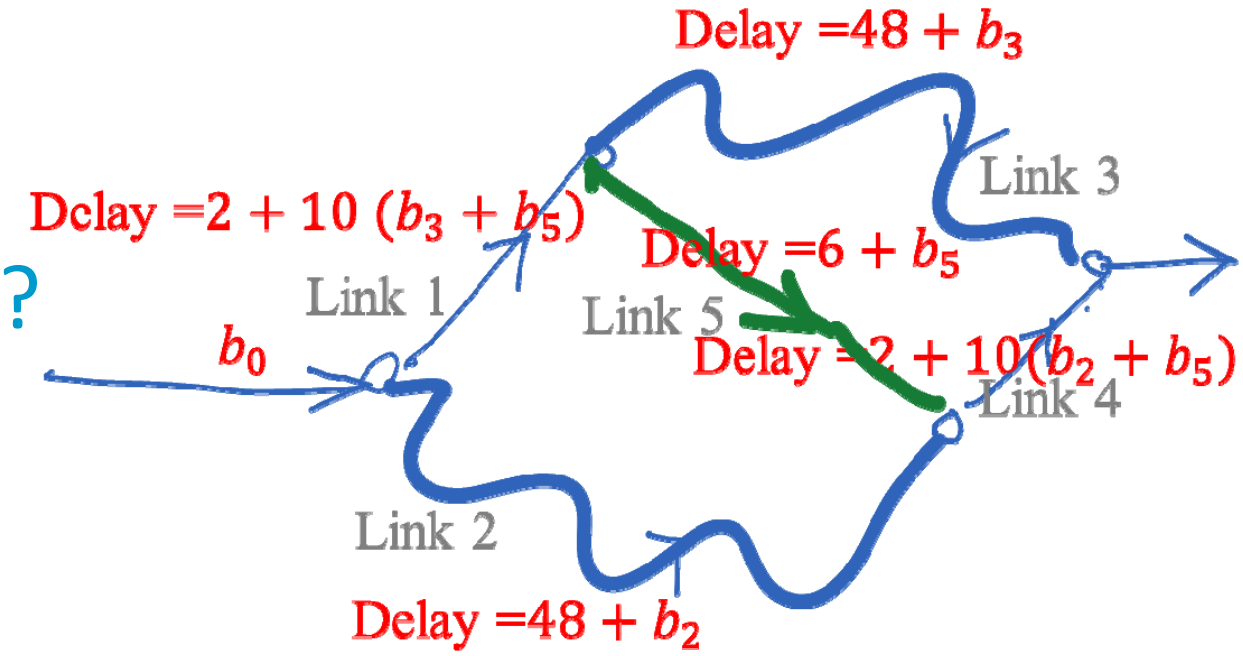
Wardrop Equilibrium ?



1. Yes
2. No
3. I don't know



What is the Wardrop Equilibrium now ?



- delay equations

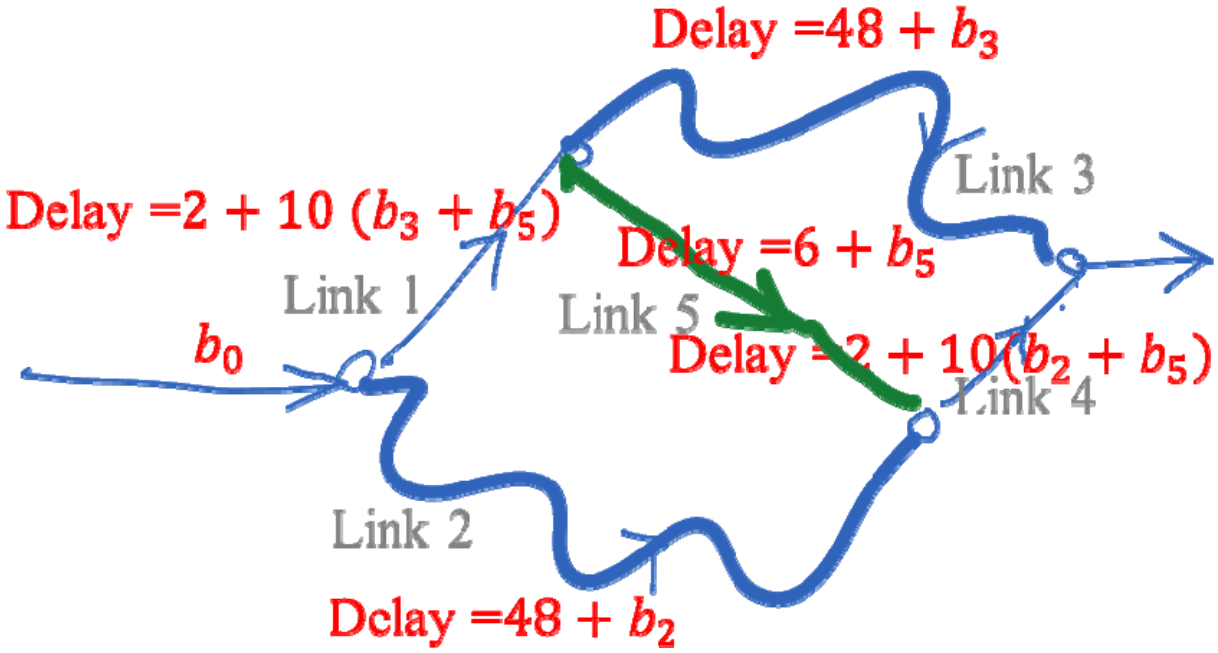
$$50 + 11b_3 + 10b_5$$
$$= 50 + 11b_2 + 10b_5$$
$$= 10 + 10b_3 + 10\,b_2 + 21\,b_5$$

total flow

$$b_1 + b_2 + b_3 = b_0 = 6$$

- Solution : $b_3 = b_5 = b_2 = 2$ Gb/s
- Delay now is 92 ms on all routes

Braess Paradox



- With shortest delay routing:
  disable link 5: delay = 83 ms
  enable link 5 : delay = 92 ms
- Adding capacity made things worse
- This is called Braess paradox
- Shortest delay routing is not optimal

Optimal Routing

- One can change the objective of routing: instead of computing shortest paths, one could solve a global optimization problem maximizing a utility function:
  - minimize total delay subject to flow constraints
  - this is a well posed optimization problem
  - the optimal solution depends on all flows
  - but it can be implemented in a distributed algorithm similar to TCP congestion control ; see [BertsekasGallager92]
- This can be solved using an offline optimization procedure that computes optimal paths for all traffic flows and downloads the routes into all routers

  Can be done with SDN

Conclusion

- Link State Routing is an alternative to distance vector
  - more complex
  - allows more control over the chosen paths
- Shortest path routing may not be globally optimal and may need to be complemented with offline optimization methods