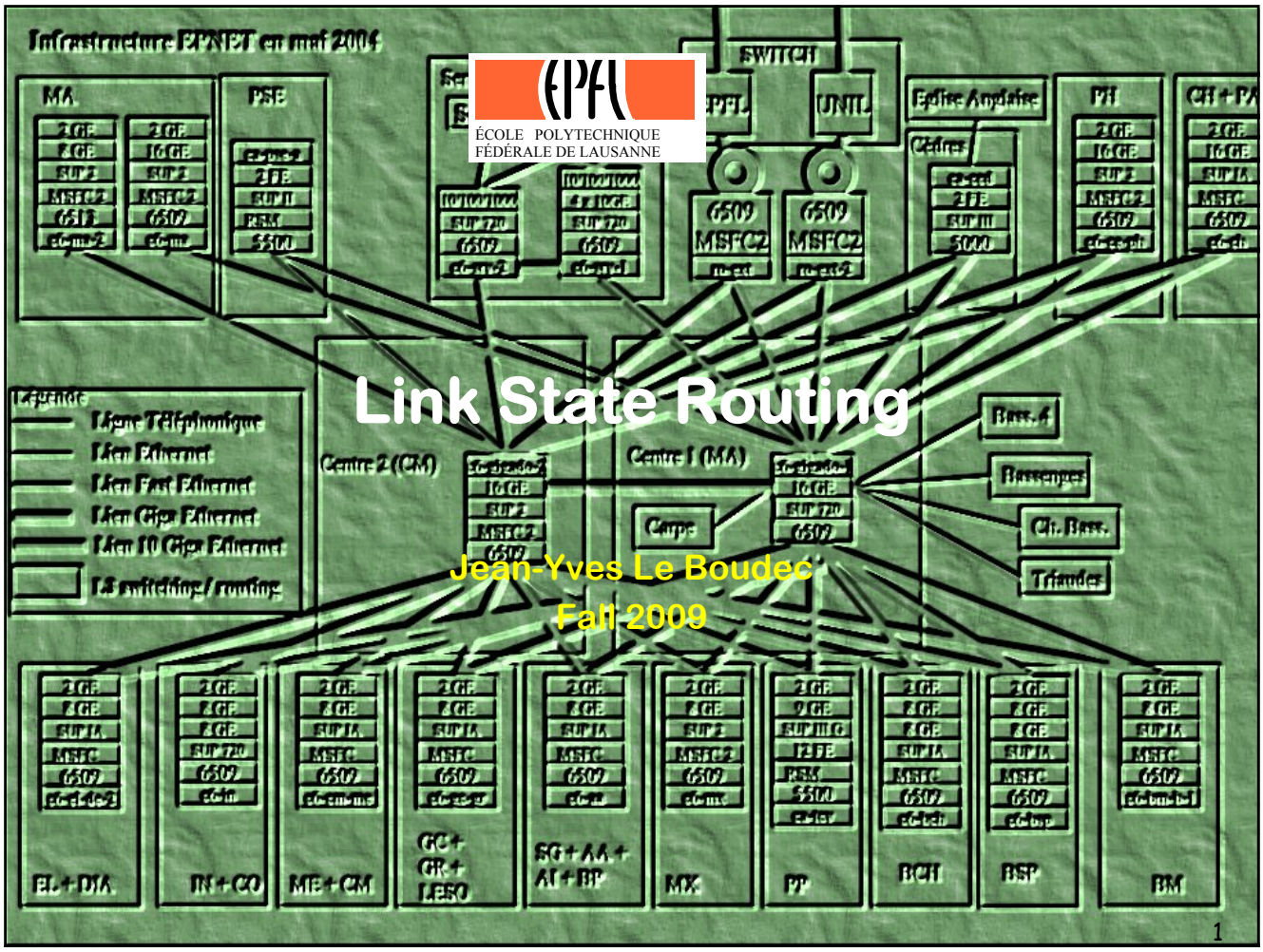


Link State Routing

Jean-Yves Le Boudec
Fall 2009

Contents

1. Link State Routing

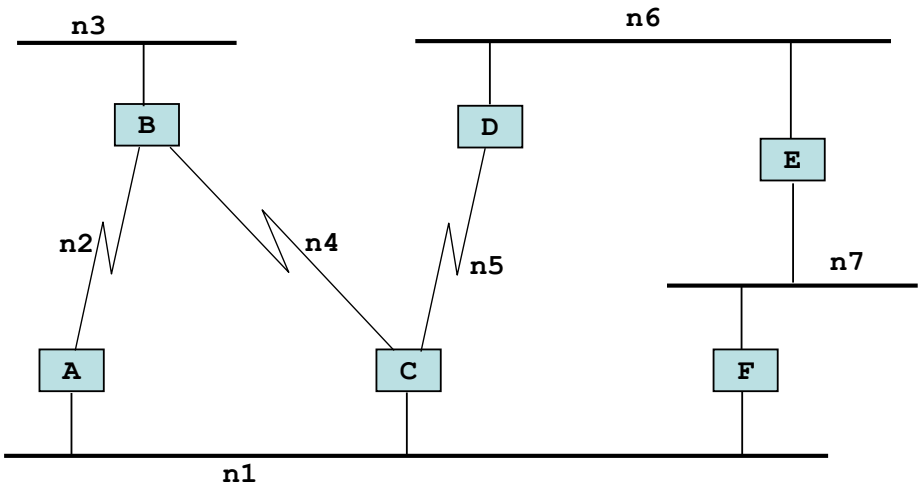
- **Principle** of link state routing
  - each router keeps a topology database of whole network
  - link state updates flooded, or multicast to all network
  - routers compute their routing tables based on topology
    often uses Dijkstra's shortest path algorithm
- Used in OSPF (Open Shortest Path First), IS-IS (similar to OSPF)and PNNI (ATM routing protocol)

(a) Topology Database Synchronization

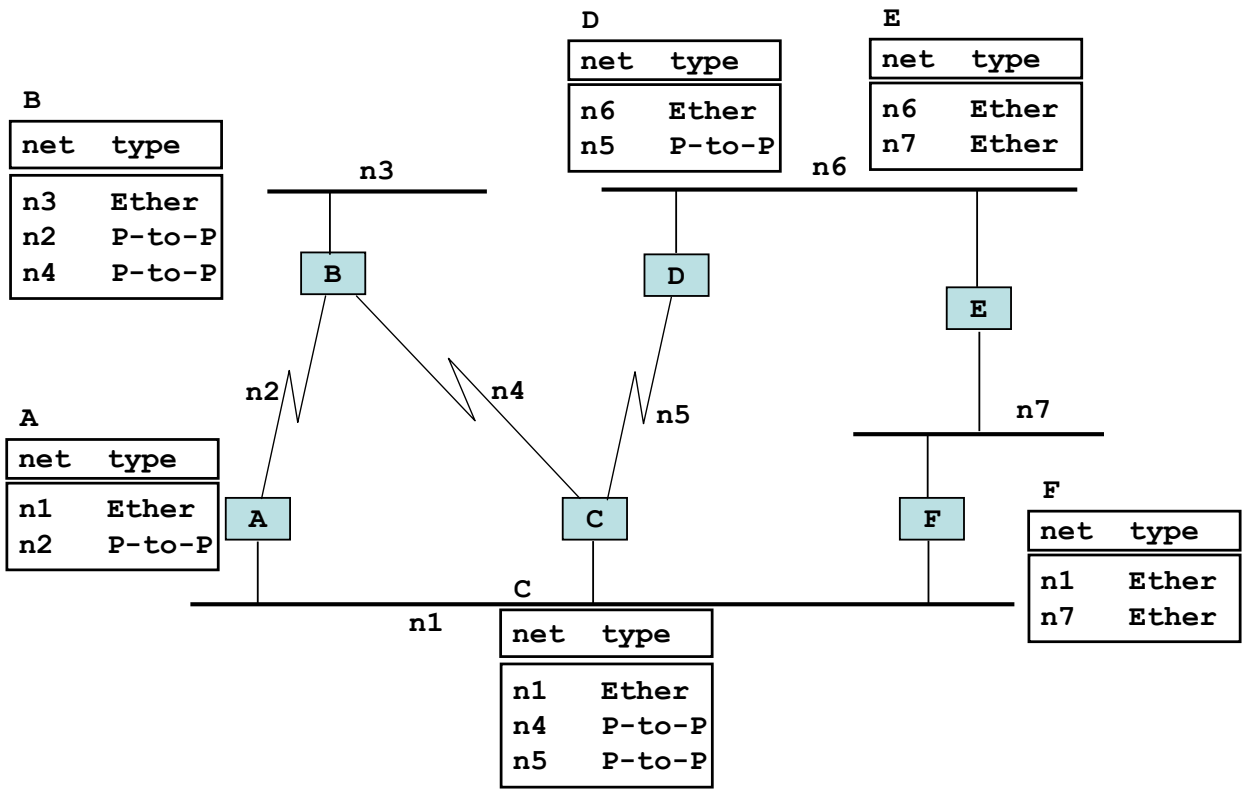
- Neighbouring nodes synchronize before starting any relationship
  - Hello protocol; keep alive
  - initial synchronization of database
  - description of all links (no information yet)
- Once synchronized, a node accepts link state advertisements
  - contain a sequence number, stored with record in 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



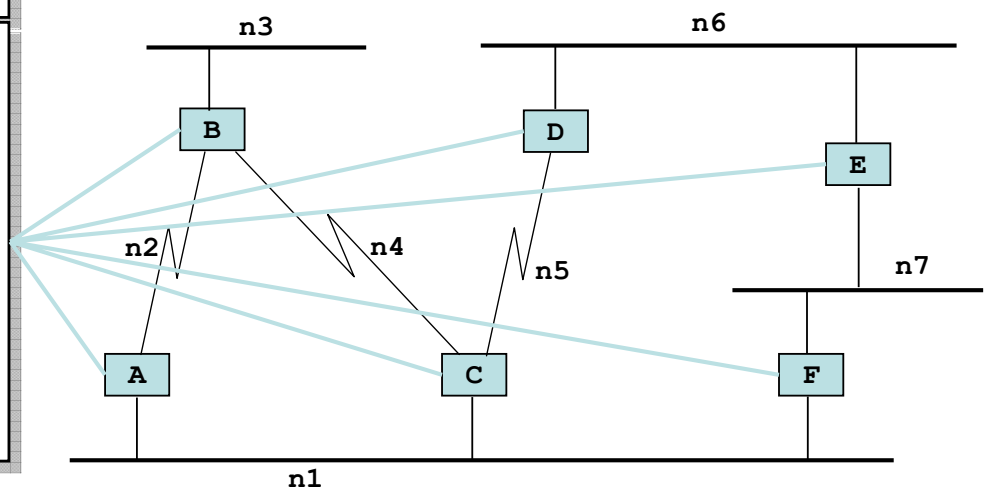
Initial routing tables



After Flooding

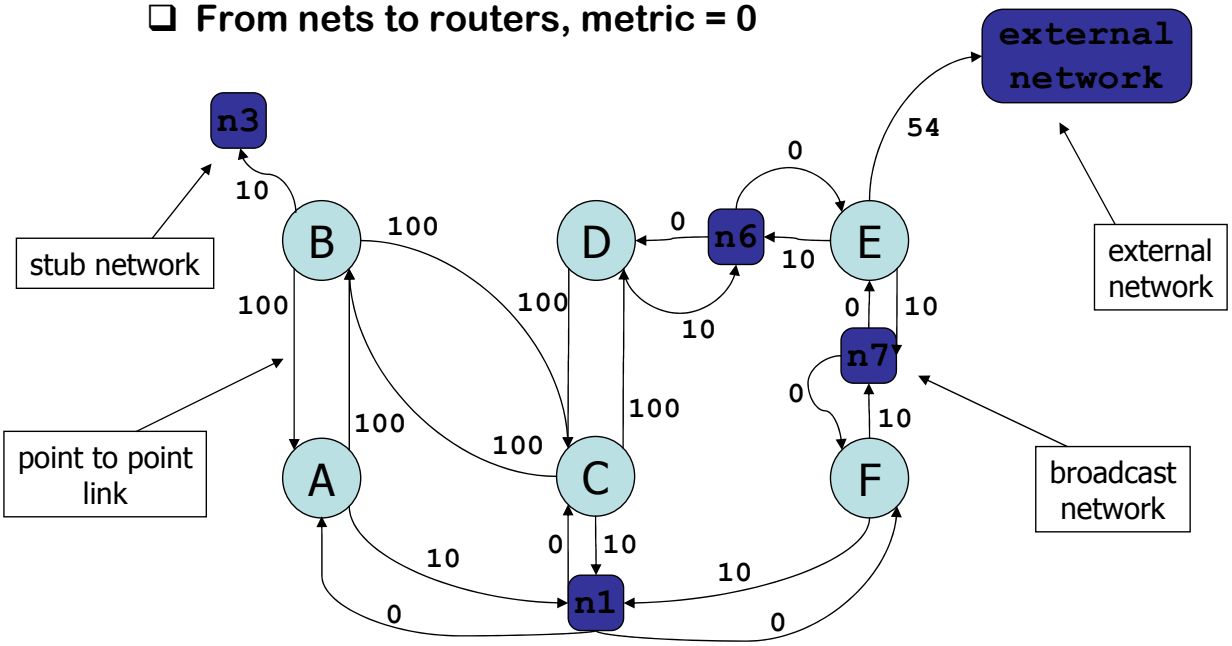
- The local metric information is flooded to all routers
- After convergence, all routers have the same information

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) Topology graph

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

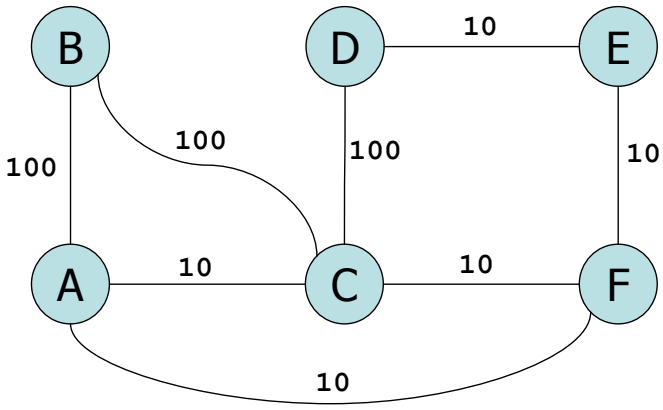


(b) Path Computation

- 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



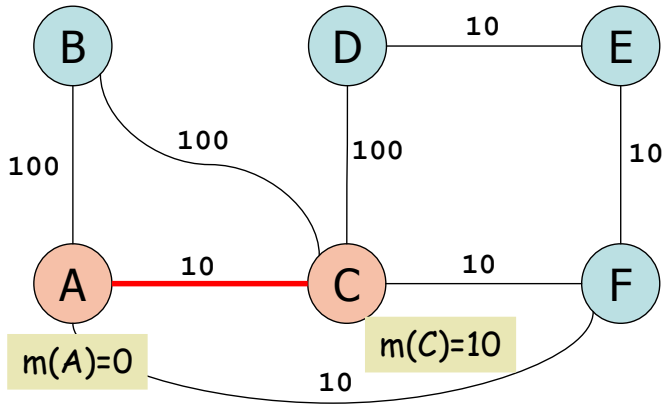
Dijkstra's Shortest Path Algorithm

- 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}
}
```

- as Bellman-Ford, works for any min-plus algebra

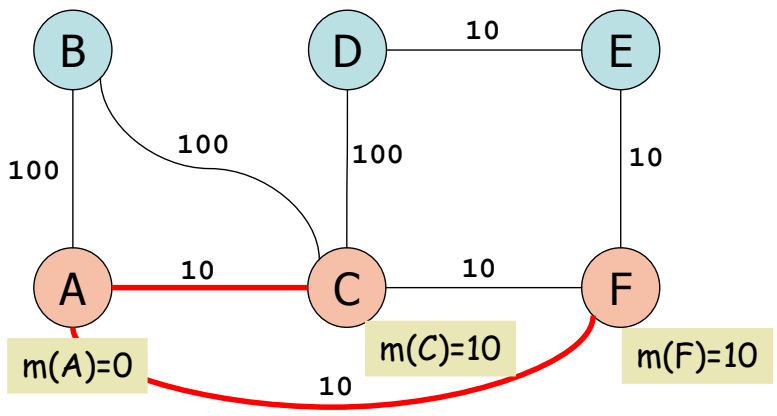
Example: Dijkstra at A



init: M = { A }

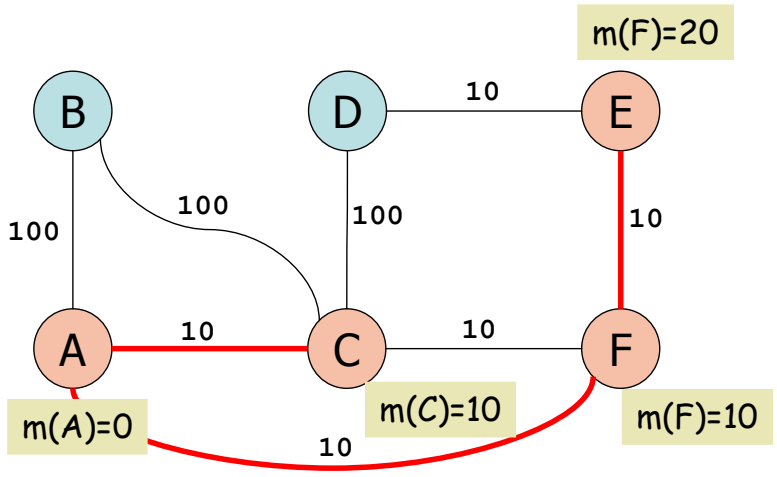
step 1:
i0=A
j0=C
m(C)=10
M = {A, C}

Example: Dijkstra at A



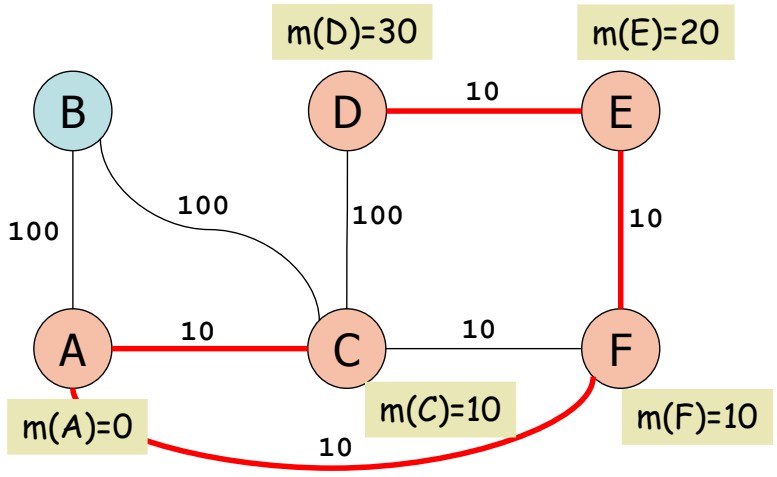
i0=A
j0=F
m(F)=10
M = {A,C,F}

Example: Dijkstra at A



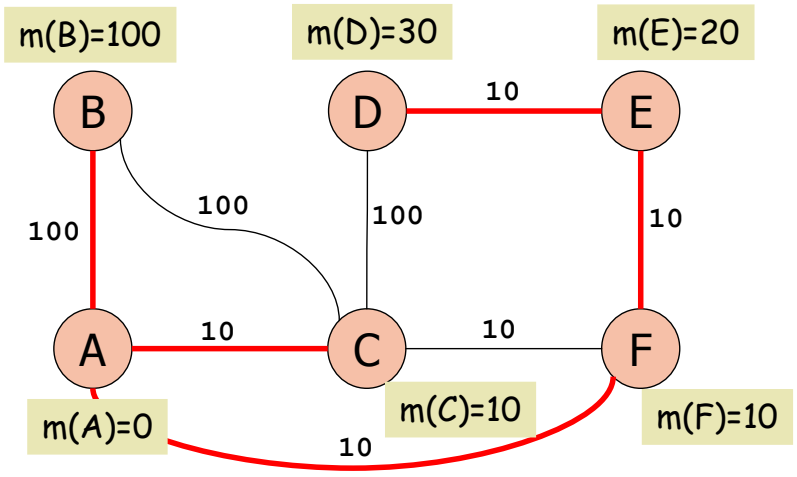
i0=F
j0=E
m(E)=20
M = {A,C,F,E}

Example: Dijkstra at A



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

Example: Dijkstra at A

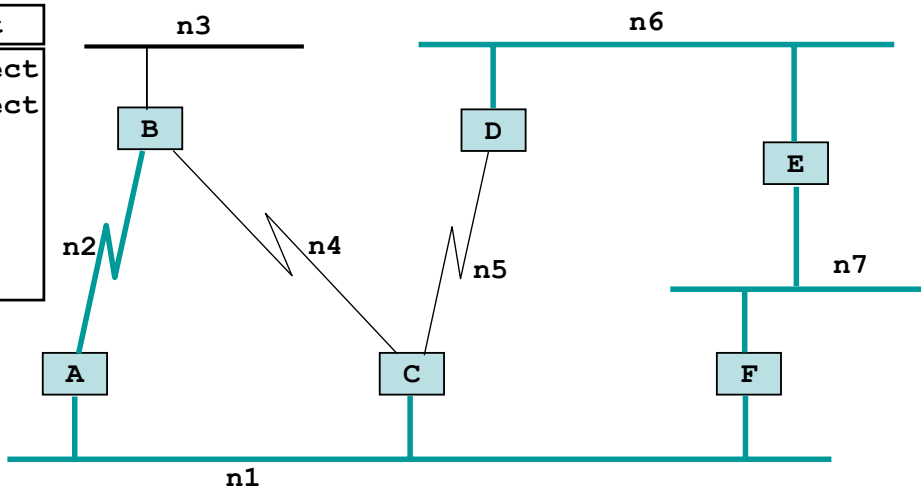


i0=A
j0=B
m(B)=100
M = {A,C,F,E,D,B}

Routing table of A

A

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



Test Your Understanding

- Q1: Run Dijkstra at C
- Q2: What are the routing tables at C

solution

LS: Summary

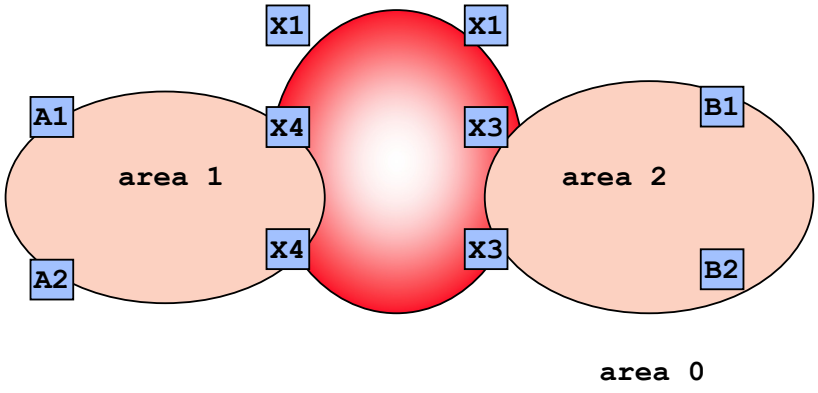
- All nodes compute their own topology database
  - represents the whole network
  - strongly synchronized
- All nodes compute their best path tree to all destinations
- Routing tables are built from the tree
  - used for next hop routing only
- LS versus DV
  - LS avoids convergence problems of DV
  - supports flexible cost definitions; can be used for routing ATM connections
  - LS is much more complex

2. Divide large networks

- Why divide large networks?
- Cost of computing routing tables
  - update when topology changes
  - SPF algorithm
    - *n* routers, *k* links
    - complexity O(*n***k*)
  - 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

- A large OSPF domain can be configured into *areas*
  - one *backbone area* (area 0)
  - 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 database per area



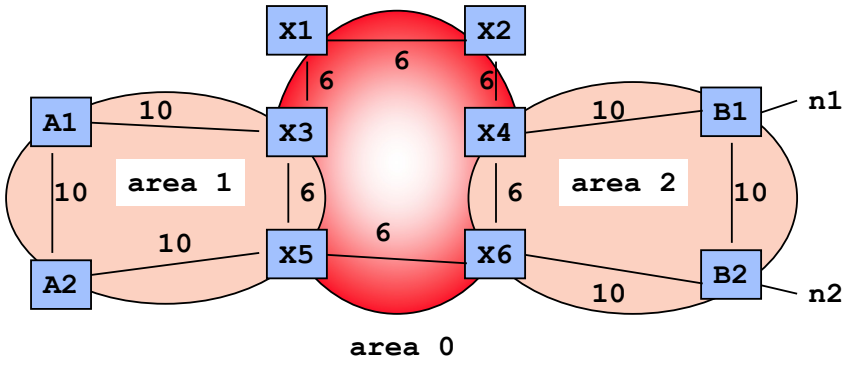
Principles

- 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

Example

- Assume networks `n1` and `n2` become visible at time 0. Show the topology databases at all routers

solution



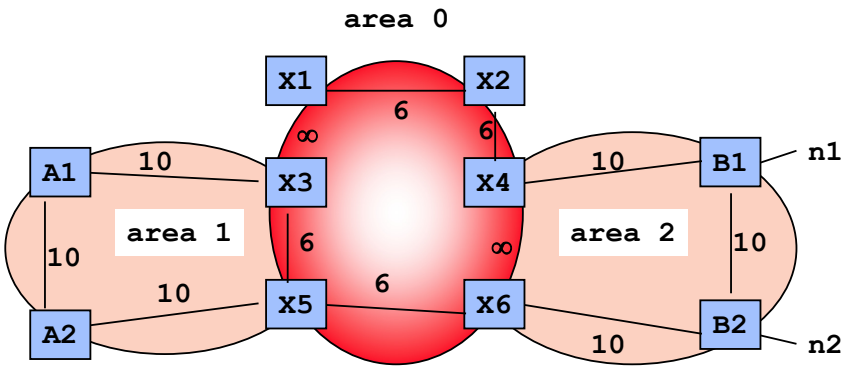
Hints

- All routers in area 2 propagate the existence of n1 and n2, directly attached to B1 (resp. B2). Draw the topology database in area 2.
- Area border routers X4 and X6 belong to area 2, thus they can compute their distances to n1 and n2
- Area border routers X4 and X6 inject their distances to n1 and n2 into the area 0 topology database (item 3 of the principle). The corresponding summary link state record is propagated to all routers of area 0. Draw now the topology database in area 0.
- All routers in area 0 can now compute their distance to n1 and n2, using their distances to X4 and X6, and using the principle of distance vector (item 1 of the principle). Do the computation for X3 and X5.
- Area border routers X3 and X5 inject their distances to n1 and n2 into the area 1 topology database (item 3 of the principle). Draw now the topology database in area 1.

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

*Example of issue : partitioned backbone



- No connectivity between areas via backbone
- There is a route through Area 2
- Virtual link
  - X4 and X6 configure a virtual link through Area 2
  - virtual link entered into the database, metric = sum of links

3. The OSPF Protocol

- OSPF (Open Shortest Path First)
  - IETF standard for internal routing
  - used in large networks (ISPs)

- Link State protocol + Hierarchical

*OSPF Components

- On top of IP (protocol type = 89)
- Multicast
  - `224.0.0.5` - all routers of a link
  - `224.0.0.6` - all designated and backup routers
- Sub-protocols
  - **Hello** to identify neighbors, elect a designated and a backup router
  - **Database description** to diffuse the topology between adjacent routers
  - **Link State** to request, update, and ack the information on a link (LSA - Link State Advertisement)

*TOS and metric

- TOS
  - mapping of 4 IP TOS bits to a decimal integer
  - 0 - normal service
  - 2 - minimize monetary cost
  - 4 - maximize reliability
  - 8 - maximize throughput
  - 16 - minimize delay
- Metric
  - time to send 100 Mb over the interface
  - C = 10^8/bandwidth
  - 1 if greater than 100 Mb/s
  - can be configured by administrator

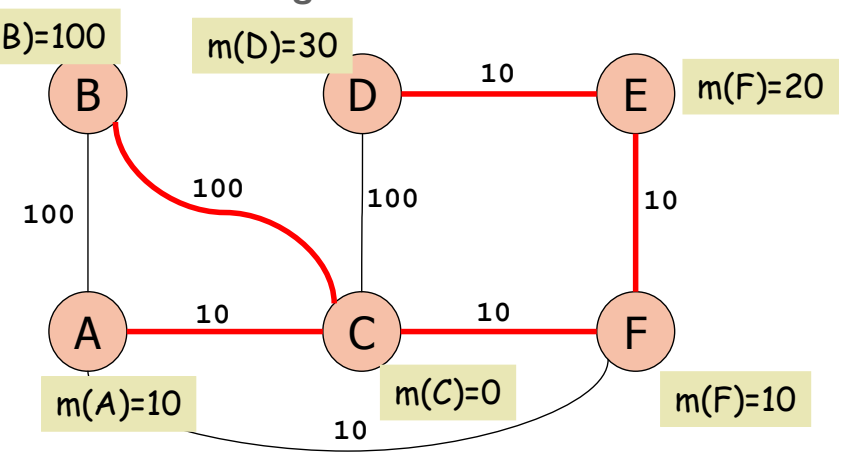
*OSPF - summary

- OSPF vs. RIP
  - much more complex, but presents many advantages
    - no count to infinity
    - no limit on the number of hops (OSPF topologies limited by Network and Router LSA size (max 64KB) to O(5000) links)
    - less signaling traffic (LS Update every 30 min)
    - advanced metric
    - large networks - hierarchical routing
  - most of the traffic when change in topology
    - but periodic Hello messages
    - in RIP: periodic routing information traffic
  - drawback
    - difficult to configure

Solutions

Test Your Understanding

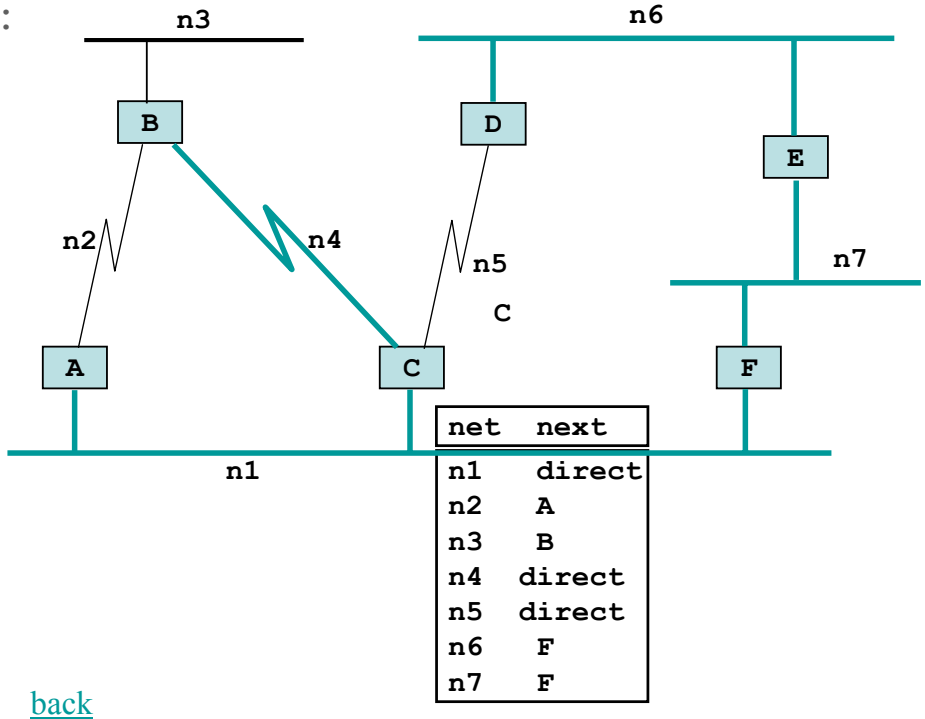
- Q1: Run Dijkstra at C
  A: (final step)
- Q2: What are the routing tables at C



Test Your Understanding

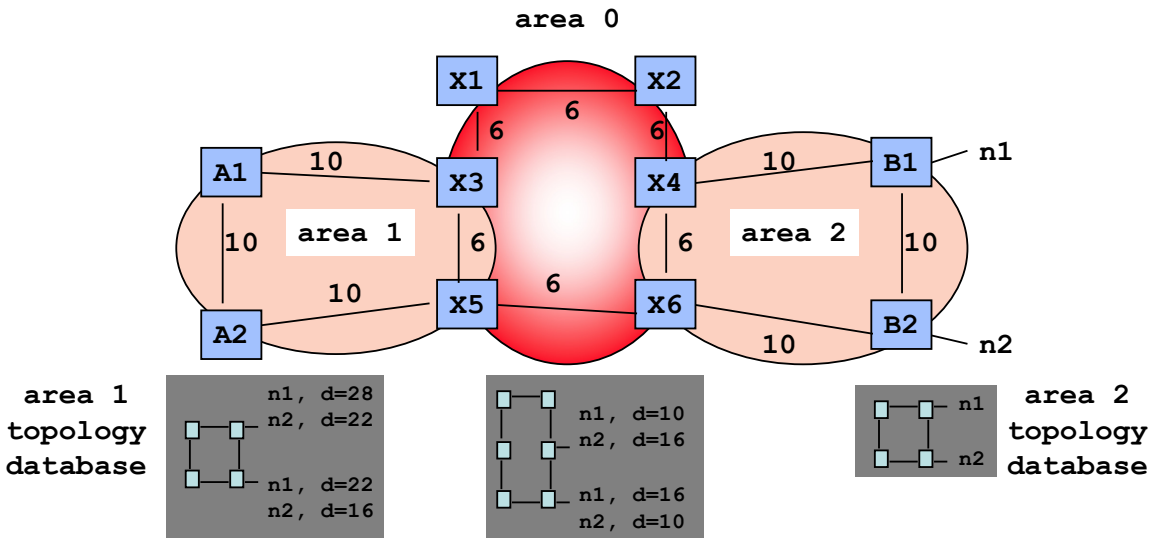
- Q2: What are the routing tables at C

A:



back

Solution



back