Advanced Computer Networks

Internal Routing - Link State protocols

Prof. Andrzej Duda
duda@imag.fr

http://duda.imag.fr

1

Contents

- Link state
  - flooding topology information
  - finding the shortest paths (Dijkstra)
  - areas - hierarchical routing
- OSPF
  - neighbor discovery - Hello protocol
  - database synchronization
  - link state updates
  - examples

2

Link State Routing

- Principles
  - estimate metrics with neighbors
    - bandwidth, delay, cost (fixed by administrator)
  - build a packet with the metrics of all neighbors
  - flood to all routers
  - compute the shortest path to all destinations (Dijkstra)
  - update if modification of topology
- Used in OSPF (Open Shortest Path First) and PNNI (ATM routing protocol)

3

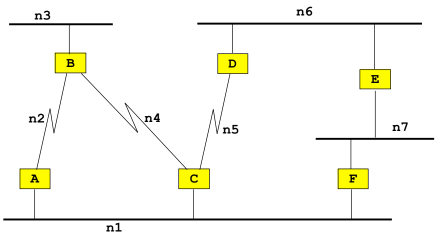
Topology Database Synchronization

- 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)

4

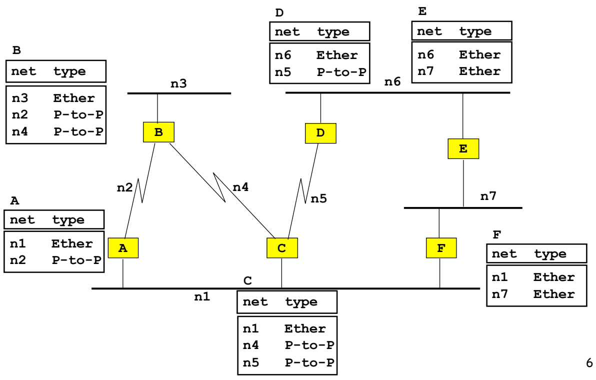
Example network

- Each router knows directly connected networks



5

Initial routing tables



6

Flooding

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

A

net	cost
n1	10
n2	100

B

net	cost
n3	10
n2	100
n4	100

C

net	cost
n1	10
n4	100
n5	100

D

net	cost
n6	10
n5	100

E

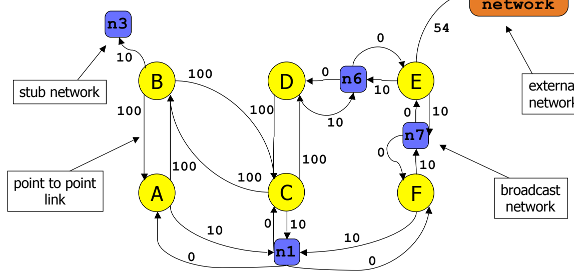
net	cost
n6	10
n7	10

F

net	cost
n1	10
n7	10

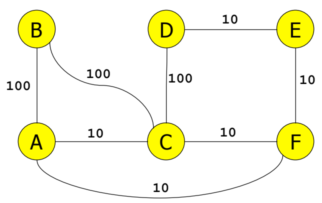
Topology graph

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



Simplified graph

- Only arrows with metrics between routers
- Execute the SPF (Shortest Path First - Dijkstra) algorithm on the graph

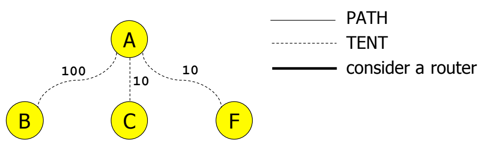


SPF at *A*

1. Initialization
   1. *PATH* variable: router *A* (the best path to destination)
   2. *TENT* variable: *empty* (tentative paths)
2. For each router *N* in *PATH*
   1. for each neighbor *M* of *N*
      1. *c(A, M) = c(A, N) + c(N, M)*
   2. if *M* is not in *PATH* nor in *TENT* with a better cost, insert *M* with direction *N* in *TENT*
3. If *TENT* is empty, end. Otherwise take the entry with the best cost from *TENT*, insert it into *PATH* and go to 2.

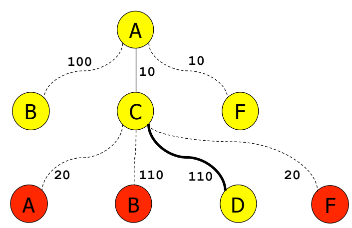
At the end PATH contains the tree of best paths to all destinations

Executing SPF



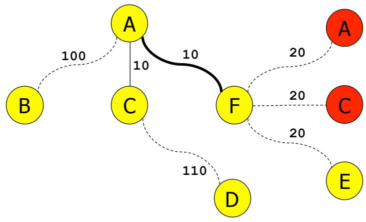

```
Before: TENT: A-B(100), A-C(10), A-F(10)
        PATH: A
After:  TENT: A-B(100), A-F(10)
        PATH: A-C(10)
```


Executing SPF



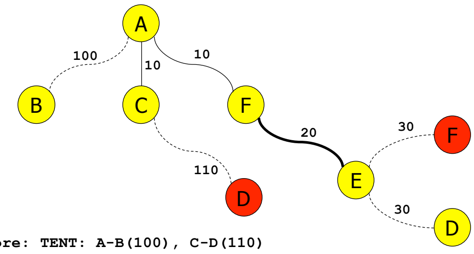

```
Before: TENT: A-B(100), A-F(10), C-D(110)
        PATH: A-C(10)
After:  TENT: A-B(100), C-D(110)
        PATH: A-C(10), A-F(10)
```

Executing SPF



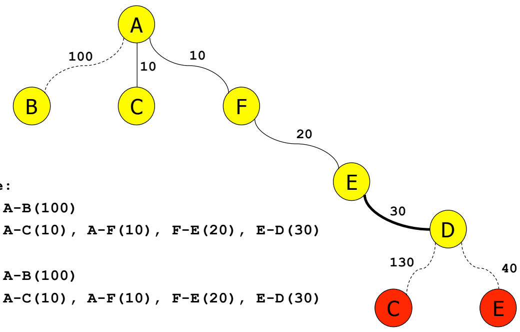

```
Before: TENT: A-B(100), C-D(110), F-E(20)
        PATH: A-C(10), A-F(10)
After:  TENT: A-B(100), C-D(110)
        PATH: A-C(10), A-F(10), F-E(20)
```


Executing SPF



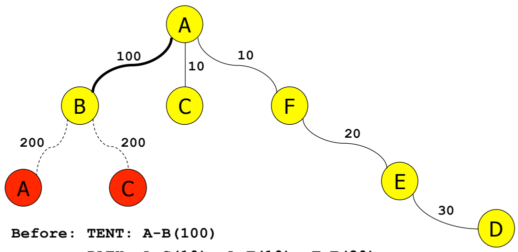

```
Before: TENT: A-B(100), C-D(110)
        TENT: A-B(100), E-D(30)
        PATH: A-C(10), A-F(10), F-E(20)
After:  TENT: A-B(100)
        PATH: A-C(10), A-F(10), F-E(20), E-D(30)
```


Executing SPF



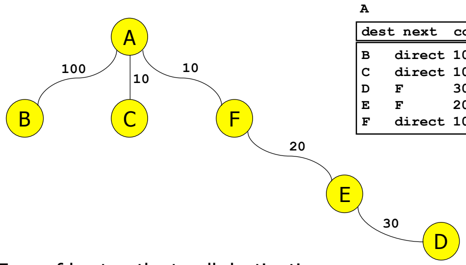

```
Before:
TENT: A-B(100)
PATH: A-C(10), A-F(10), F-E(20), E-D(30)
After:
TENT: A-B(100)
PATH: A-C(10), A-F(10), F-E(20), E-D(30)
```


Executing SPF




```
Before: TENT: A-B(100)
        PATH: A-C(10), A-F(10), F-E(20),
              E-D(30)
After:  TENT:
        PATH: A-C(10), A-F(10), F-E(20),
              E-D(30), A-B(100)
```


Result



A

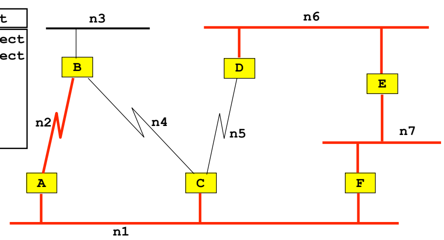
dest	next	cost
B	direct	100
C	direct	100
D	F	30
E	F	20
F	direct	10

- Tree of best paths to all destinations

Routing table of A

A

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



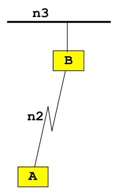
Towards OSPF

- OSPF (Open Shortest Path First)
  - Link State protocol
  - Link State information: LSA (Link State Advertisement)
  - different sub-protocols: Hello, Database Description, Link State flooding
- It allows to
  - separate hosts and routers
  - consider different types of networks
    - broadcast (Ethernet), NBMA (ATM, X.25), point-to-point (PPP)
  - divide large networks into several areas
  - independent route computing in each area

19

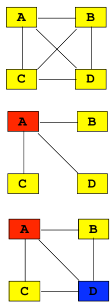
Separate hosts and routers

- Link should be described in the DB
  - link between a router and each host, but LANs in most cases: advertize the link to the "stub network"
  - link of the form of a broadcast network (Ethernet)
    - IP address of the subnetwork (stub network)
    - e.g. **n3** identified by **128.88.38/24**
  - link to a neighbor router
    - IP address of the neighbor router
    - e.g. **n2** identified by **176.44.23.254**
    - no IP address assigned to the interface
      - interface index



20

Designated routers

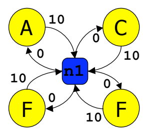


- Number of neighbors
  - if *n* routers, *n*(*n*-1)/2 neighbors
- Election of a designated router on a LAN
  - *n*-1 neighbors
  - flooding
    - advertise to **224.0.0.6** (all designated routers)
    - flooded to **224.0.0.5** (all routers)
  - back-up designated router
    - listens to advertisements, but does not flood
    - failure of the designated router detected by Hello
      - back-up becomes designated router

21

Virtual networks

- LAN represented as a virtual network
  - less entries in the DB
  - real cost to **n1**, zero to routers



22

NBMA networks and P-to-P

- NBMA (Non Broadcast, Multiple Access)
  - several hosts, but no broadcast
- Virtual circuits between all hosts - each link appears in the database
- Managed as broadcast networks
  - designated and back-up router
  - permanent virtual circuits only to them
- Flooding
  - designated router sends a copy of update to all routers

23

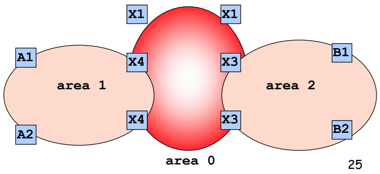
Divide large networks

- 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
- 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

24

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



25

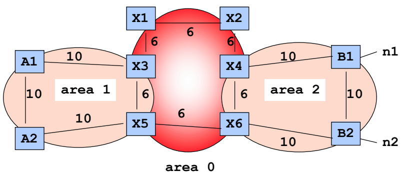
Principles

- 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 two areas
- Inter-level routing information
  - summary link state advertisements (LSA) from other areas are injected into the local topology databases

26

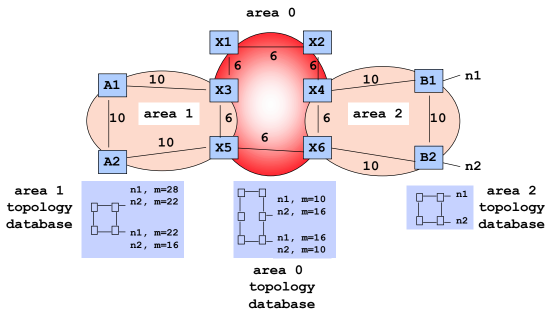
Example

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



27

Solution



28

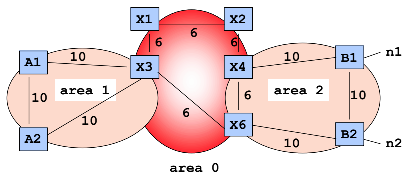
Explanations

- All routers in area 2 propagate the existence of n1 and n2, directly attached to B1 (resp. B2).
- 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 LSA is propagated to all routers of 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).

29

Stub area

- Many networks are connected only via one router
- Stub area
  - all external networks aggregated into **default** route
  - e.g. route to **n1**, **n2** or any other network in Area 0 and 2 goes through X3



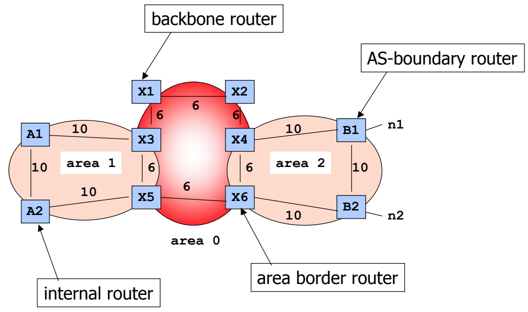
30

Classification of routers

- Internal routers
  - a router with all directly connected networks belonging to the same area
- Area border routers
  - attached to multiple areas
  - condense LSA of their attached areas for distribution to the backbone
- Backbone routers
  - a router that has an interface to the backbone area
- AS boundary routers
  - exchange routing information with routers belonging to other AS

31

Classification of routers



32

OSPF protocol

- 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)
- LSA
  - tagged with the router Id and checksum
  - 5 different types

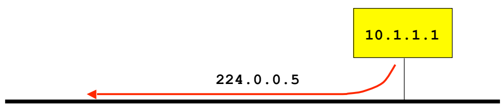
33

OSPF protocol PDUs

- OSPF protocol type = 1
  - Hello
- OSPF protocol type = 2
  - Database description
- OSPF protocol type = 3
  - Link State Request
- OSPF protocol type = 4
  - Link State Update
- OSPF protocol type = 5
  - Link State Ack

34

Example

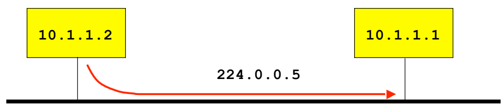



```
OSPFv2-hello 44:
area 0.0.0.1 E mask 255.255.255.0 int 10 pri
  5 dead 40 dr 10.1.1.1 nbrs
```

- `224.0.0.5` - to all routers of a link
- Router `10.1.1.1` with priority 5, prefix `10.1.1.0/24`
- Area 1, not stub area (bit E), interval 10 sec, dead interval 40, it proposes itself as designated router, no neighbors

35

Example

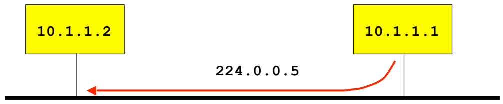



```
OSPFv2-hello 44:
area 0.0.0.1 E mask 255.255.255.0 int 10 pri 4
  dead 40 nbrs
```

- Router `10.1.1.2` with priority 4, prefix `10.1.1.0/24`
- Area 1, not stub area (bit E), interval 10 sec, dead interval 40, no neighbors

36

Example




```
OSPFv2-hello 44:
area 0.0.0.1 E mask 255.255.255.0 int 10 pri 5 dead 40 dr
   10.1.1.1 nbrs 10.1.1.2
```

- Router `10.1.1.1` becomes designated
- Router `10.1.1.2` appears as a neighbor
  - bi-directional connectivity
  - can start synchronizing LS databases

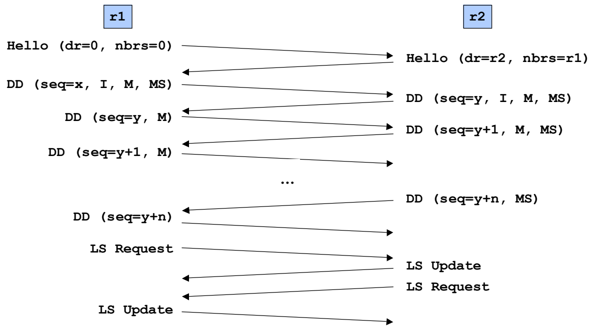
37

Database Description protocol

- Unicast packets between a router and its neighbor
- Master/slave relationship - election of the Master
  - router with larger Id becomes Master
- Master sends packets to slave (polls)
- Slave acknowledges by echoing the sequence number
- If lost packet, master retransmits
- Exchange finished when bit M=0 sent by both routers

38

DD protocol

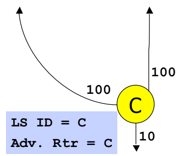


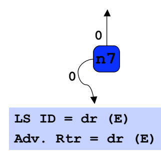
- `r2` is designated router (priority), `r2` has larger router Id than `r1`

39

LSA types

Type	Description
1. **router-LSA**	Originated by all routers. Describes the collected states of the router's interfaces to an area. Flooded throughout a single area only.
2. **network-LSA**	Originated for broadcast and NBMA networks by the Designated Router. Contains the list of routers connected to the network (m=0). Flooded throughout a single area only.





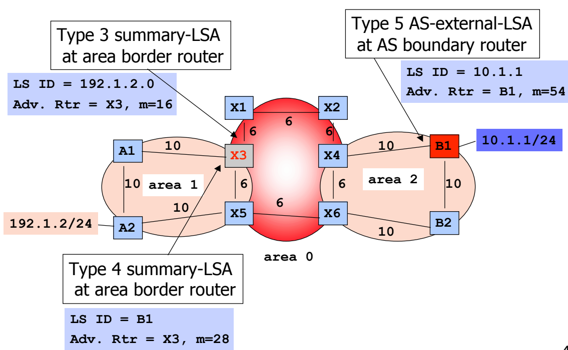
40

LSA types

Type	Description
3. **summary-LSA**	Originated by area border routers. Describes a route to a destination outside the area, yet still inside the AS (i.e., an inter-area route). Type 3 summary-LSAs describe routes to networks. Flooded through out the LSA's associated area.
4. **summary-LSA**	Type 4 summary-LSAs describe routes to AS boundary routers.
5. **AS-external-LSA**	Originated by AS boundary routers. Describes a route to a destination in another Autonomous System. Default routes for the AS can also be described by AS-external-LSAs. Flooded through-out the AS.

41

Types of LSA



42

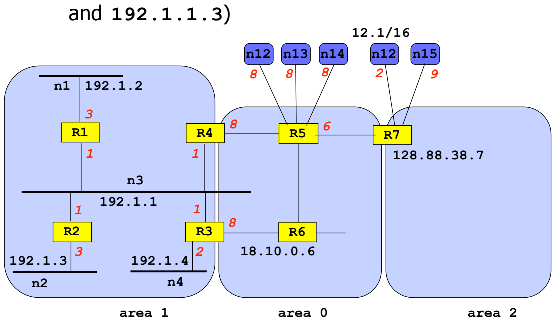
Metric

- 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

43

Example network

- Router address: router number (R3 - `192.1.4.3` and `192.1.1.3`)



44

Router-LSA

```
Router R3 for the area 1
LS age = 0, LS Type = 1
LS Id = 192.1.1.3
Adv. router = 192.1.1.3
bit E = 0, bit B = 1             ; area border router
#links = 2
   Link ID = 192.1.1.4      ;IP address of Desig. Rtr.
   Link Data = 192.1.1.3    ;R3's IP interface to net
   Type = 2                 ;connects to transit network
   # TOS metrics = 0
   metric =   1
   Link ID = 192.1.4.0      ;IP Network number
   Link Data = 0xffffff00   ;Network mask
   Type = 3                 ;connects to stub network
   # TOS metrics = 0
   metric =   2
```

45

Router-LSA

```
Router R3 for the backbone
LS age = 0, LS Type = 1
LS Id = 192.1.1.3
Adv. router = 192.1.1.3
bit E = 0, bit B = 1             ;area border router
#links = 1
   Link ID = 18.10.0.6           ;Neighbor's Router ID
   Link Data = 0.0.0.3           ;interface index (3rd)
   Type = 1                      ;connects to router
   # TOS metrics = 0
   metric =   8
```

46

Network-LSA

```
R4 on behalf of Network n3
   LS age = 0, LS type = 2,
   Link State ID = 192.1.1.4         ;IP address of Desig. Rtr.
   Adv. Router = 192.1.1.4           ;R4's Router ID
   Network Mask = 0xffffff00
          Attached   Router = 192.1.1.4     ;Router ID
          Attached   Router = 192.1.1.1     ;Router ID
          Attached   Router = 192.1.1.2     ;Router ID
          Attached   Router = 192.1.1.3     ;Router ID
```

47

Summary-LSA

```
Summary-LSA for Network n1 by Router R4 into the backbone

   LS age = 0, LS type = 3
   Link State ID = 192.1.2.0        ;n1's IP network number
   Adv. Router = 192.1.1.4          ;R4's ID
   Network Mask = 0xffffff00
   metric = 4

Summary-LSA for AS boundary router R7 by Router R4 into Area 1

   LS age = 0, LS type = 4
   Link State ID = 128.88.38.7       ;R7's ID
   Adv. Router = 192.1.1.4           ;R4's ID
   Network Mask = 0xffffff00
   metric = 14
```

48

AS-external-LSA

```
AS-external-LSA for Network n12 by Router R7
   LS age = 0, LS type = 5
   Link State ID = 12.1.0.0           ;n12's IP network number
   Advertising Router = 128.88.38.7   ;Router R7's ID
   bit E = 1                           ;metric>than internal
   Network Mask = 0xffff0000
   metric = 2
   Forwarding address = 0.0.0.0       ;packets for external
                                      ;destination n12 should
                                      ;be forwarded to Adv.
                                      ;router - R7
```

49

Convergence

- Route timeout after 1 hour
  - LS Update every 30 min.
- Detect a failure
  - 40 sec (dead interval)
- Smallest interval to recompute SPF
  - 30 sec (Dijkstra interval)
- Reconfiguration time
  - 70 sec.
- Proposals
  - Hello each 100 ms
  - SPF immediately

50

Conclusion

- 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

51

52