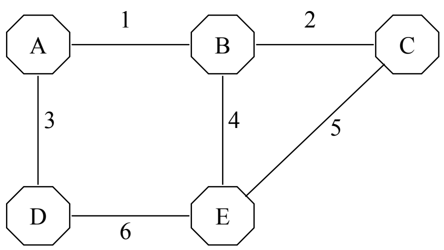
Link State Routing Principles

Link state routing

- The goal is to avoid the routing loops typical of DV routing and to scale to bigger networks and to varying topologies.
- A link state protocol maintains the topology map (link state DB) of the network.
  - Same map in every node
  - When the topology changes, the maps are updated quickly
- OSPF (Open Shortest Path First) is the IETF specified link state protocol for Internet.
  - OSPF is recommended as the follower of RIP
  - More complex than RIP

The map is the full list of all links

- Example network



- One node is responsible for a particular entry
- Link directions are separate entries
- Same map in every node
  - No loops

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

The routing table is generated from the link state database

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

Link state database

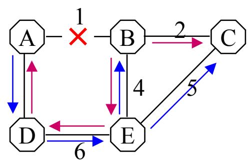
Dijkstra's shortest-path-first algorithm

From	To	Link	Cost
A	A	local	0
A	B	1	1
A	C	1	2
A	D	3	1
A	E	3	2

A's routing table

Flooding protocol distributes information about topology changes

- The updates are distributed to the whole network



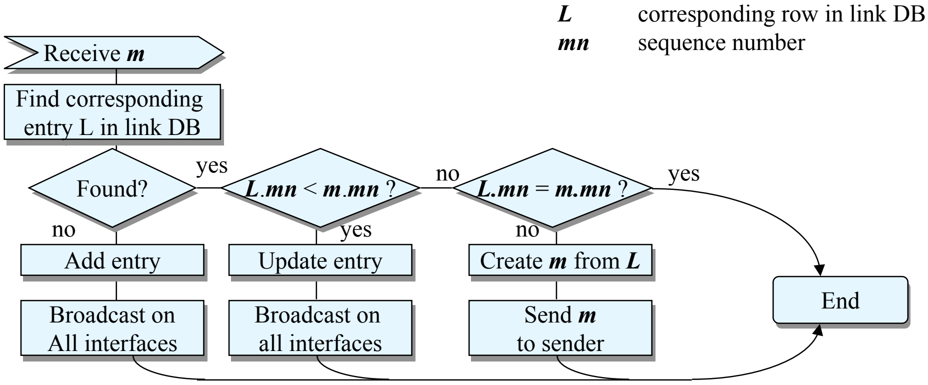
From	To	Link	Cost	Seq.num
A	B	1	inf	2

From	To	Link	Cost	Seq.num
B	A	1	inf	2

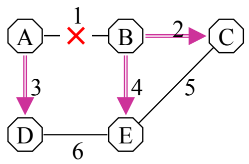
Flooding protocol distributes information about topology changes

Flooding algorithm:

m	received entry
L	corresponding row in link DB
mn	sequence number



Link database after distribution of failure of link AB



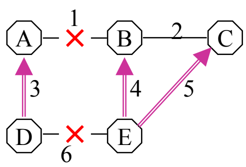
From	To	Link	Cost	Seq.num
A	B	1	inf	2
A	D	3	1	1
B	A	1	inf	2
B	C	2	1	1
B	E	4	1	1
C	B	2	1	1
C	E	5	1	1
D	A	3	1	1
D	E	6	1	1
E	B	4	1	1
E	C	5	1	1
E	D	6	1	1

- Message numbering starts from 1 on node restart.
- Modulo arithmetic is used to determine what is “a little bigger than”

⇒ message numbering can overflow without problems.

⇒ 4294967295 + 1 = 0

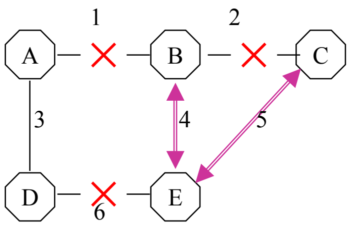
If network splits into islands, databases in islands may diverge



From	To	Link	Cost	Seq.num
A	B	1	inf	2
A	D	3	1	1
B	A	1	inf	2
B	C	2	1	1
B	E	4	1	1
C	B	2	1	1
C	E	5	1	1
D	A	3	1	1
D	E	6	inf	2
E	B	4	1	1
E	C	5	1	1
E	D	6	1	1

From	To	Link	Cost	Seq.num
A	B	1	inf	2
A	D	3	1	1
B	A	1	inf	2
B	C	2	1	1
B	E	4	1	1
C	B	2	1	1
C	E	5	1	1
D	A	3	1	1
D	E	6	1	1
E	B	4	1	1
E	C	5	1	1
E	D	6	inf	2

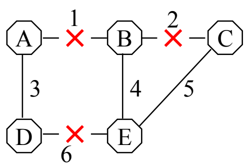
Link 2 fails ⇒ databases diverge even more



DBs in B, C and E:

From	To	Link	Cost	Seq.num.
A	B	1	inf	2
A	D	3	1	1
B	A	1	inf	2
B	C	2	inf	2
B	E	4	1	1
C	B	2	inf	2
C	E	5	1	1
D	A	3	1	1
D	E	6	1	1
E	B	4	1	1
E	C	5	1	1
E	D	6	inf	2

Databases diverge

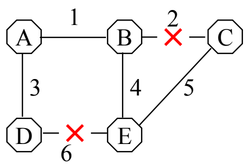


There is no immediate problem, but if link 1 goes up again ...

From	To	Link	Cost	Seq.num
A	B	1	inf	2
A	D	3	1	1
B	A	1	inf	2
B	C	2	1	1
B	E	4	1	1
C	B	2	1	1
C	E	5	1	1
D	A	3	1	1
D	E	6	inf	2
E	B	4	1	1
E	C	5	1	1
E	D	6	1	1

From	To	Link	Cost	Seq.num
A	B	1	inf	2
A	D	3	1	1
B	A	1	inf	2
B	C	2	inf	2
B	E	4	1	1
C	B	2	inf	2
C	E	5	1	1
D	A	3	1	1
D	E	6	1	1
E	B	4	1	1
E	C	5	1	1
E	D	6	inf	2

Link 1 goes up

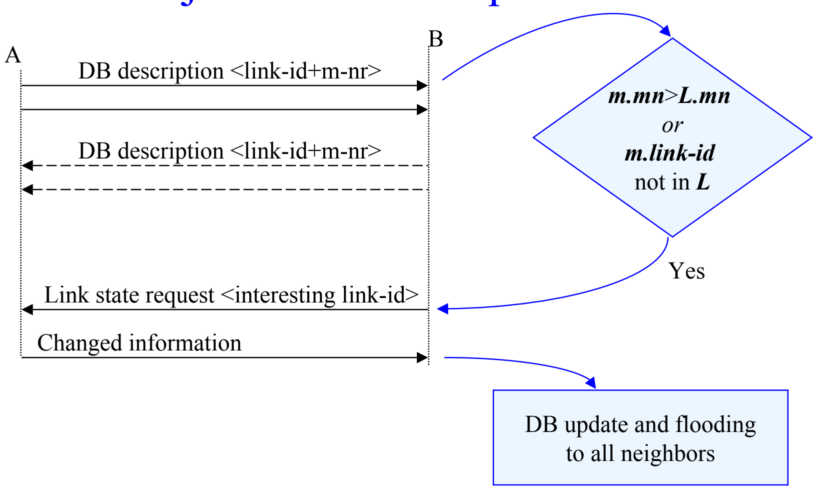


From	To	Link	Cost	Seq.num
A	B	1	1	3
A	D	3	1	1
B	A	1	1	3
B	C	2	1	1
B	E	4	1	1
C	B	2	1	1
C	E	5	1	1
D	A	3	1	1
D	E	6	inf	2
E	B	4	1	1
E	C	5	1	1
E	D	6	1	1

From	To	Link	Cost	Seq.num
A	B	1	1	3
A	D	3	1	1
B	A	1	1	3
B	C	2	inf	2
B	E	4	1	1
C	B	2	inf	2
C	E	5	1	1
D	A	3	1	1
D	E	6	1	1
E	B	4	1	1
E	C	5	1	1
E	D	6	inf	2

From	To	Link	Cost	Seq.num
A	B	1	1	3
A	D	3	1	1
B	A	1	1	3
B	C	2	inf	2
B	E	4	1	1
C	B	2	inf	2
C	E	5	1	1
D	A	3	1	1
D	E	6	inf	2
E	B	4	1	1
E	C	5	1	1
E	D	6	inf	2

After reconnection of the islands “bringing up adjacencies” is required



What happens if router C is restarted?

- There may be LSAs in the network that were distributed by C before the restart
  - After the restart, the router numbers the LSA:s with the initial sequence number
  - The neighbor replies that is has newer information.
  - If C wants to keep it own LSAs alive, it increases the sequence number by 1 and redistributes.
  - If the information of the neighbor is no longer valid, C removes it by distributing the same entry with the age = MaxAge.

Integrity of the link DB must be secured

Protection:

- Flooding messages are acknowledged link by link.
- DB description messages are acknowledged.
- Each DB entry is protected by obsolescence timer. If an update does not arrive in time, the entry is removed.
- Each entry is protected by a checksum.
- Messages also carry authentication info.

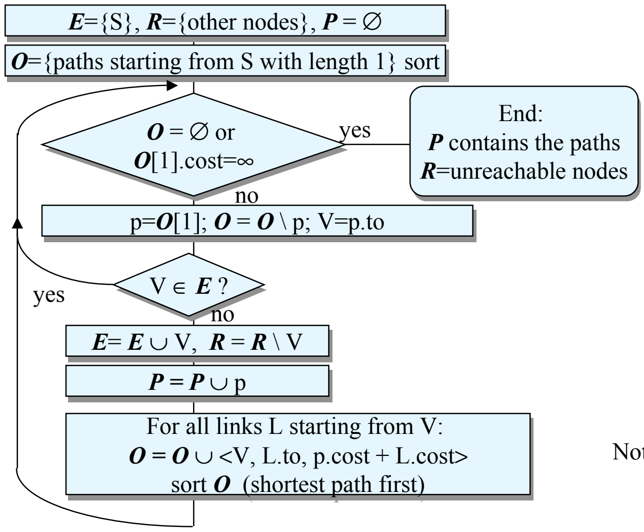
But: while update is in progress, some nodes receive info earlier than others ⇒ routing mistakes happen.

Dijkstra's shortest-path-first algorithm

Dijkstra's shortest-path-first algorithm

- OSPF is based on Dijkstra's shortest-path-first (SPF) algorithm.
- The purpose is to create a routing table.
- The shortest-path-first algorithm computes the shortest path from source node ***S*** to all other nodes.
- Dijkstra's algorithm converges faster than Bellman-Ford.
  - $O(M \log M) < O(N \cdot M)$
  - M is number of links, N is number of nodes, both the of same order of magnitude
- Nodes are divided to evaluated ***E***, the paths from which are known, and other nodes ***R***.
- In addition an ordered list of paths ***O*** is needed.

Dijkstra's shortest-path-first algorithm



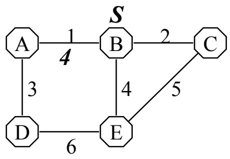
S	start node
E	evaluated nodes
R	other nodes
O	ordered list of paths <from, to, cost>
P	paths (result)
L	link state database <from, to, cost>

Note: the loops are removed here

Dijkstra's shortest-path-first algorithm

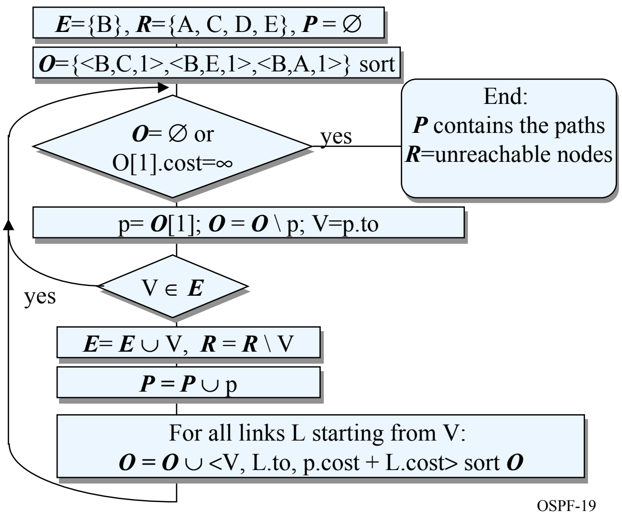
1. E={S}, R={N-S}, O={all one-hop paths starting from S}
2. If O is empty or is the first path in O has infinite length:
   - Mark all the remaining nodes in R as unreachable
   - Stop
3. P is the shortest path in O. Remove P from O. V is the last node of P.
4. Is V is in E:
   - Go to step 2
5. Create a set of paths by adding to P all links starting from V. The path is the previous path + the cost of the link. Add these paths to O in length order.
6. Go to step 2

Dijkstra's shortest-path-first algorithm – example

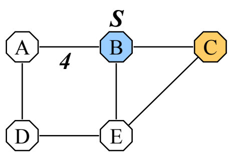


L

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



Dijkstra's shortest-path-first algorithm – example



E

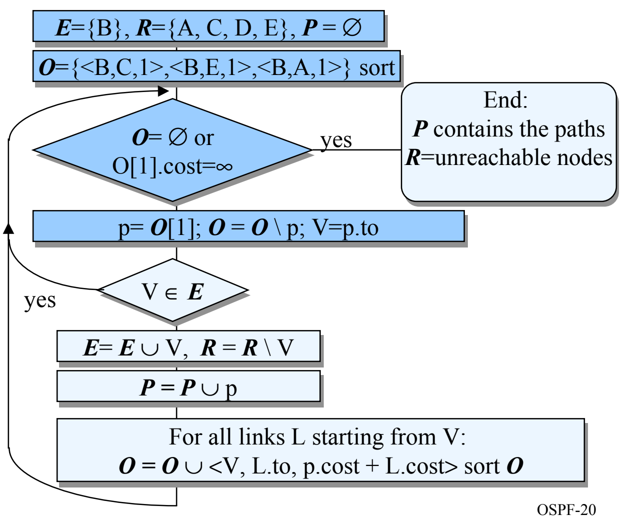
B

R

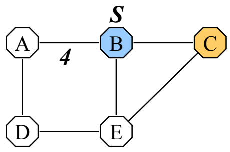
A, C, D, E

O

<B,C,1>
<B,E,1>
<B,A,4>



Dijkstra’s shortest-path-first algorithm – example



E

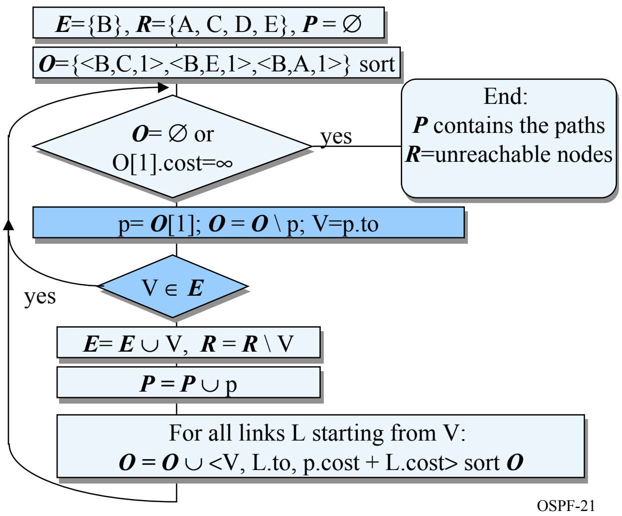
B

R

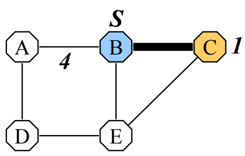
A, C, D, E

O

<B,E,1>
<B,A,4>



Dijkstra’s shortest-path-first algorithm – example



E

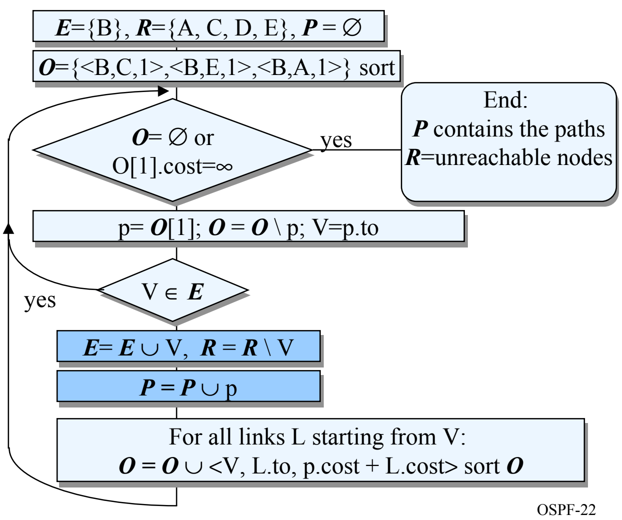
B, C

R

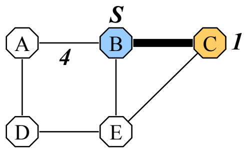
A, D, E

O

<B,E,1>
<B,A,4>



Dijkstra’s shortest-path-first algorithm – example



E

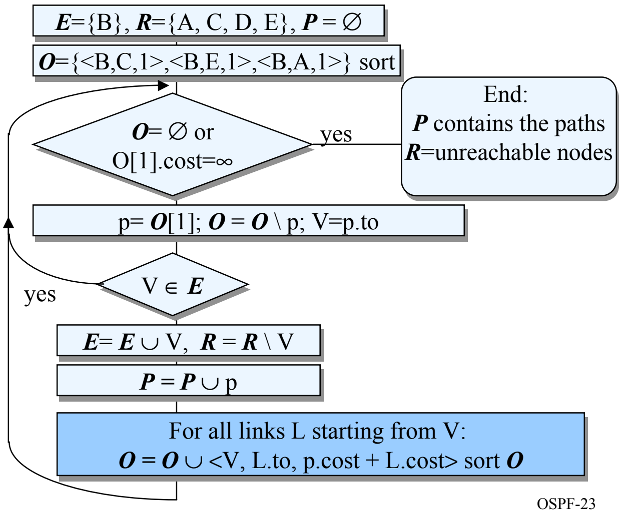
B, C

R

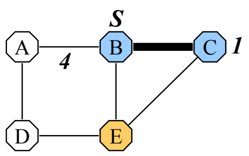
A, D, E

O

<B,E,1>
<C,E,2>
<B,A,4>



Dijkstra’s shortest-path-first algorithm – example



E

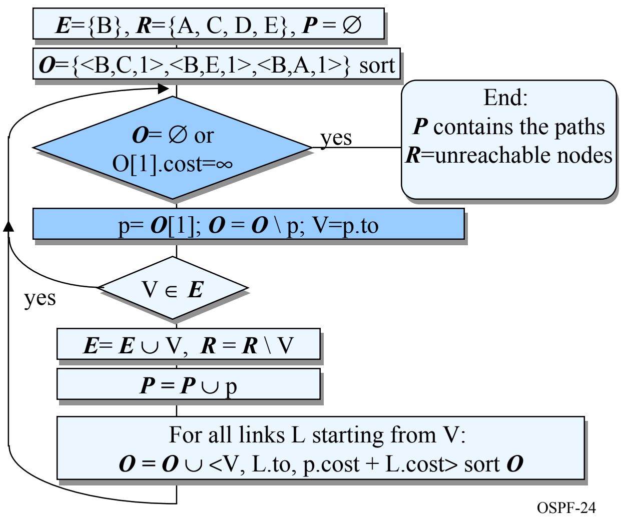
B, C

R

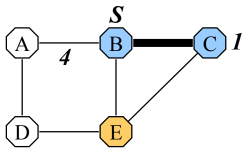
A, D, E

O

<B,E,1>
<C,E,2>
<B,A,4>



Dijkstra’s shortest-path-first algorithm – example



E

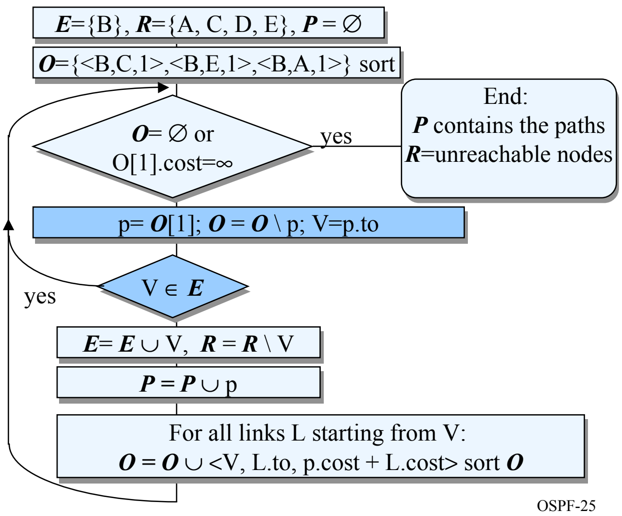
B, C

R

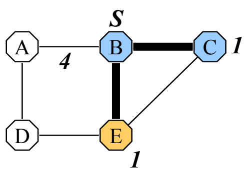
A, D, E

O

<C,E,2>
<B,A,4>



Dijkstra’s shortest-path-first algorithm – example



E

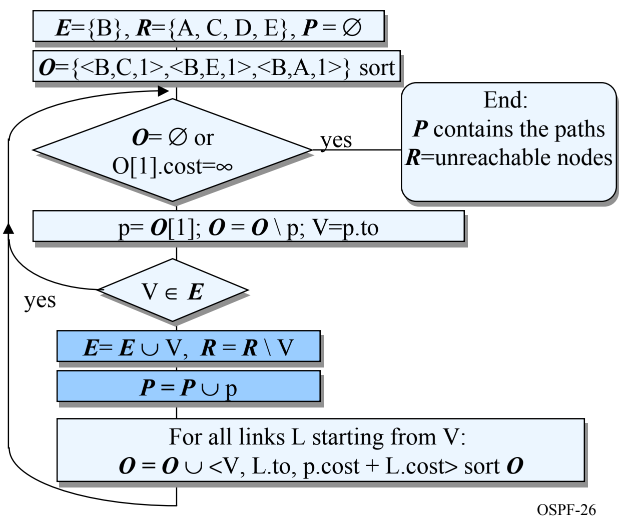
B, C, E

R

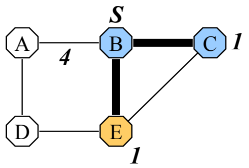
A, D

O

<C,E,2>
<B,A,4>



Dijkstra's shortest-path-first algorithm – example



E

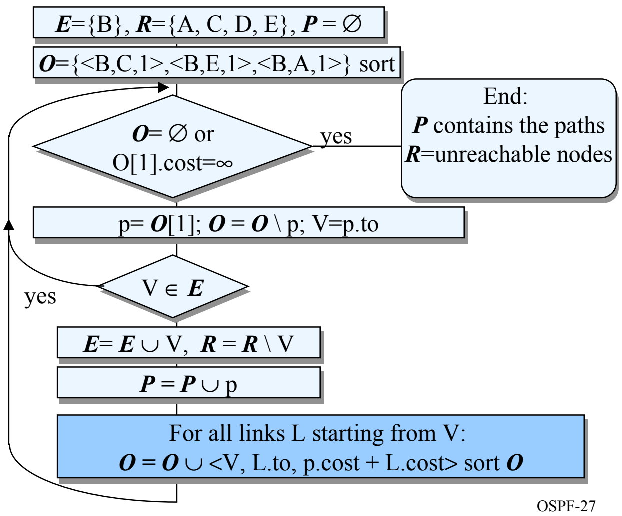
B, C, E

R

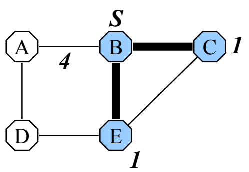
A, D

O

<C,E,2>
<E,D,2>
<B,A,4>



Dijkstra's shortest-path-first algorithm – example



E

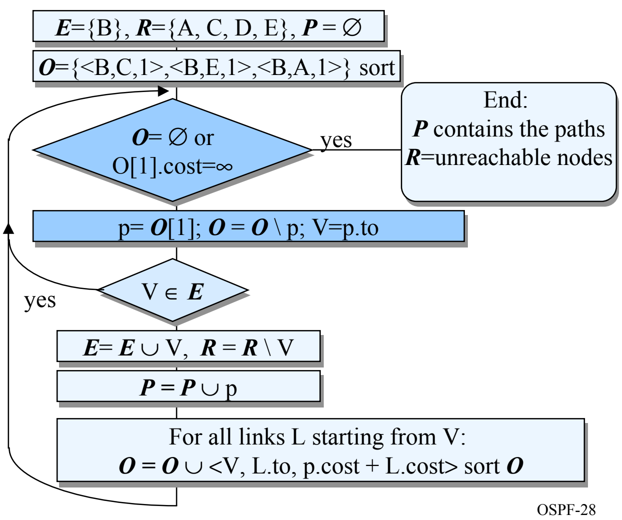
B, C, E

R

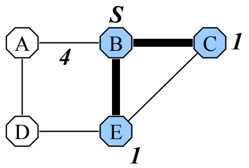
A, D

O

<C,E,2>
<E,D,2>
<B,A,4>



Dijkstra’s shortest-path-first algorithm – example



E

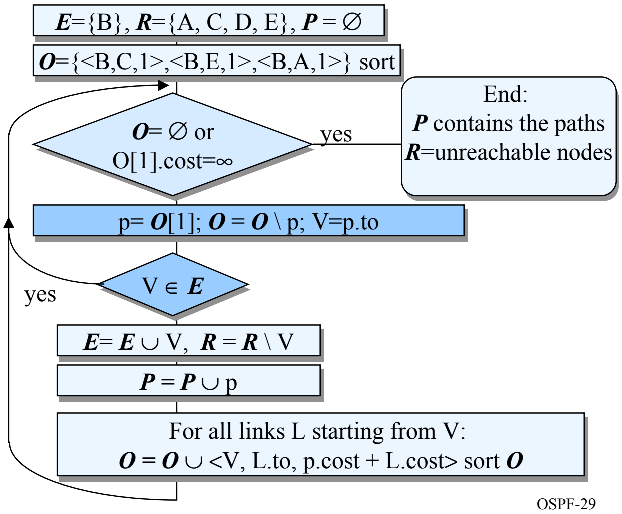
B, C, E

R

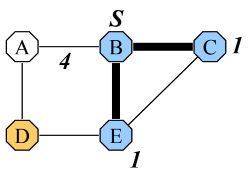
A, D

O

<E,D,2>
<B,A,4>



Dijkstra’s shortest-path-first algorithm – example



E

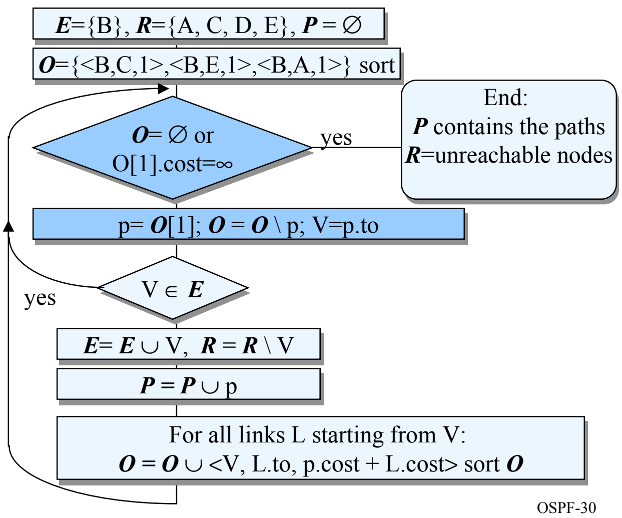
B, C, E

R

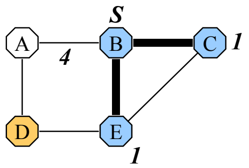
A, D

O

<E,D,2>
<B,A,4>



Dijkstra's shortest-path-first algorithm – example



E

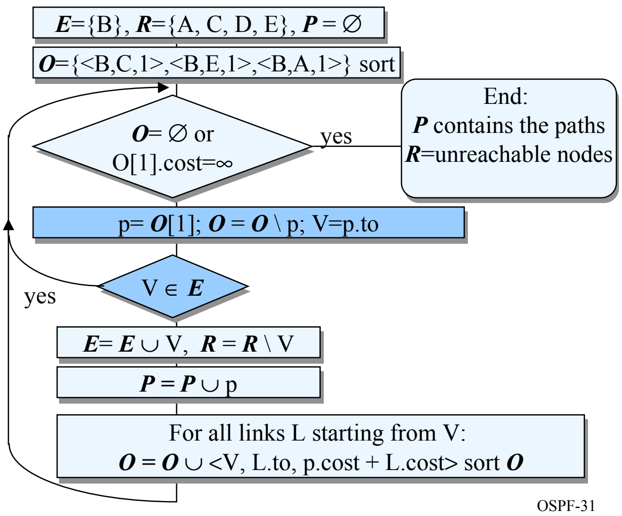
B, C, E

R

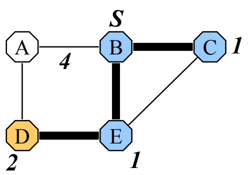
A, D

O

<B,A,4>



Dijkstra's shortest-path-first algorithm – example



E

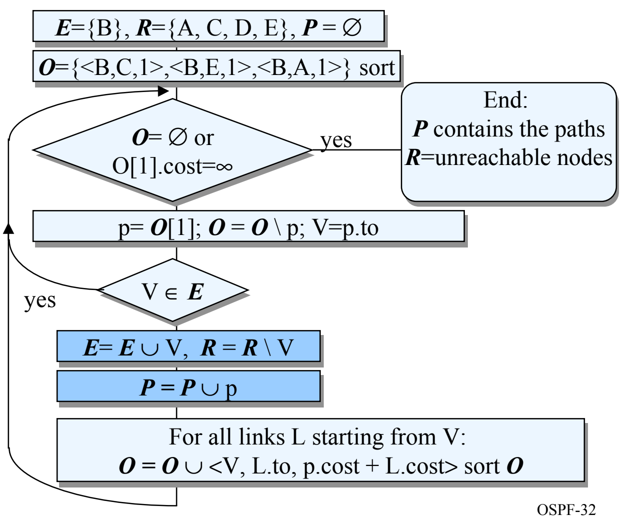
B, C, E, D

R

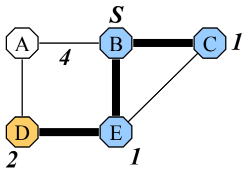
A

O

<B,A,4>



Dijkstra's shortest-path-first algorithm – example



E

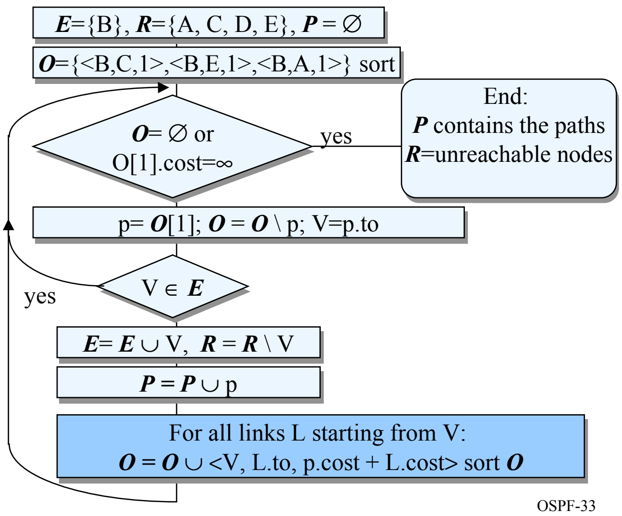
B, C, E, D

R

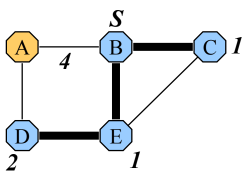
A

O

<D,A,3>
<B,A,4>



Dijkstra's shortest-path-first algorithm – example



E

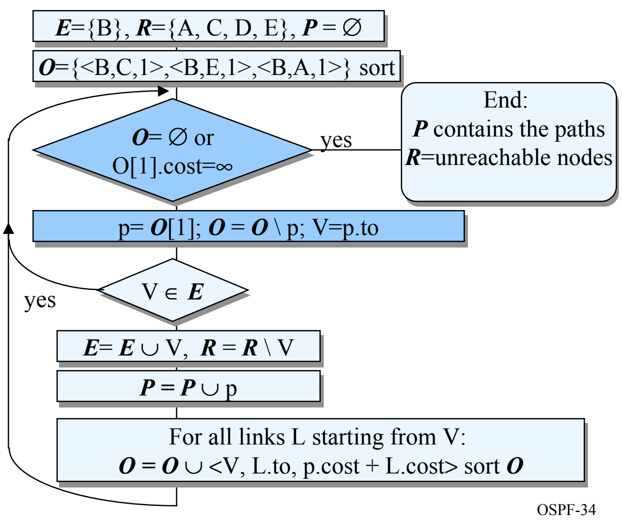
B, C, E, D

R

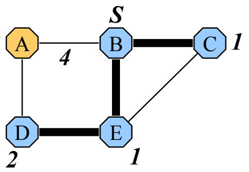
A

O

<D,A,3>
<B,A,4>



Dijkstra's shortest-path-first algorithm – example



E

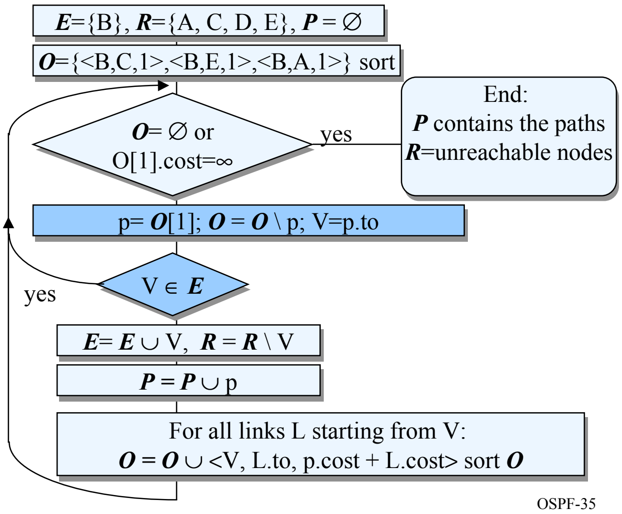
B, C, E, D

R

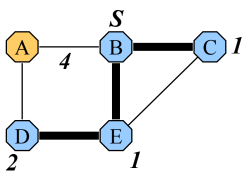
A

O

<B,A,4>



Dijkstra's shortest-path-first algorithm – example



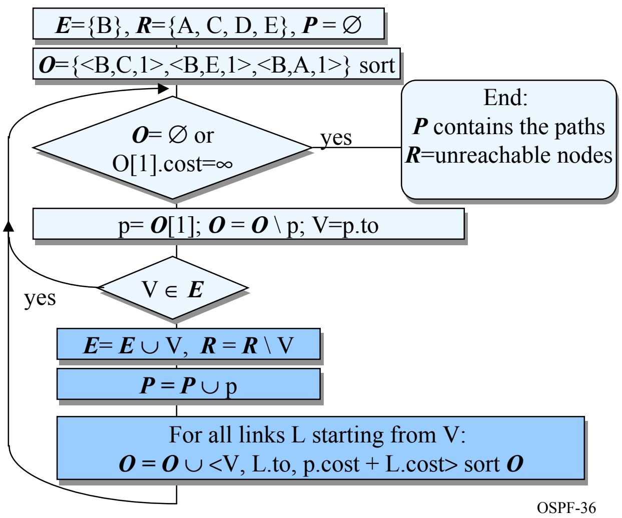
E

B, C, E, D, A

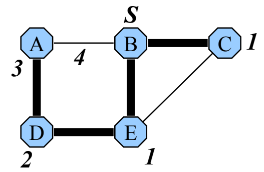
R

O

<B,A,4>



Dijkstra’s shortest-path-first algorithm – example



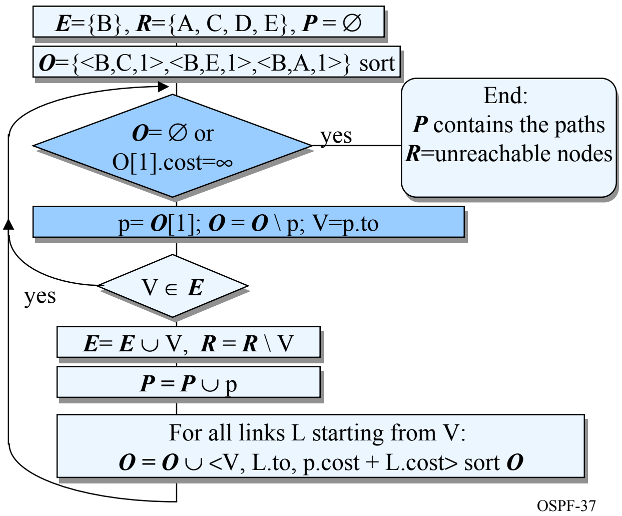
E

B, C, E, D, A

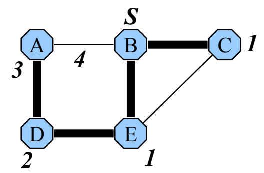
R

O

<B,A,4>



Dijkstra’s shortest-path-first algorithm – example

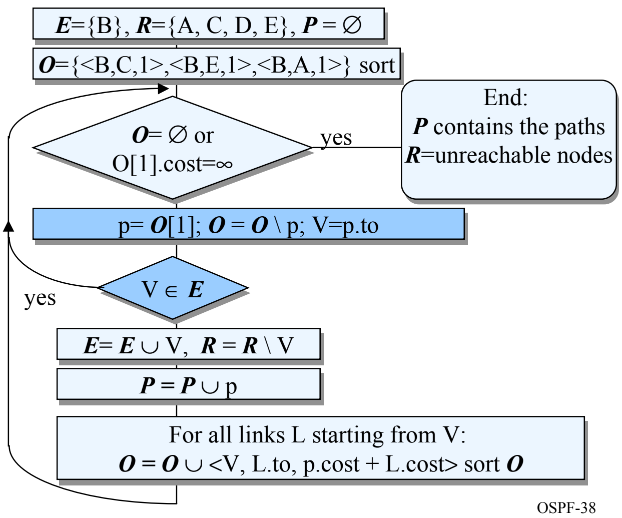


E

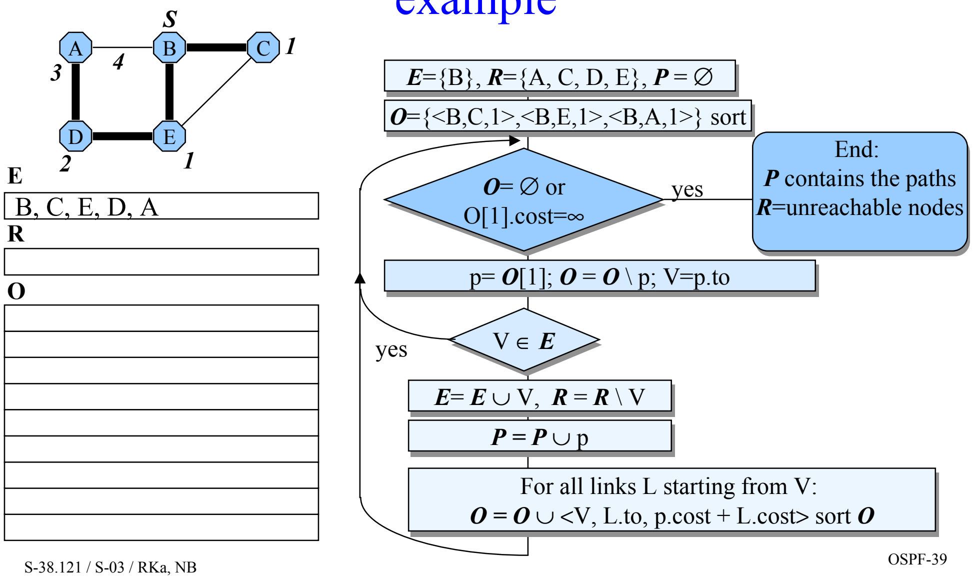
B, C, E, D, A

R

O



Dijkstra's shortest-path-first algorithm – example



Advantages of Link State Protocols

- Link State DBs converge quickly, no loops are formed
  - $O(M \log M)$ *M = number of links*
- Metrics can be quite accurate.
  - In DV-protocols, counting-to-infinity limits (inf=16)
- One protocol can easily support several metrics:
  - A routing table for each metric: capacity, delay, cost, reliability.
- Can maintain several routes to a destination.
  - Load sharing
- Exterior routes can have their own representation.

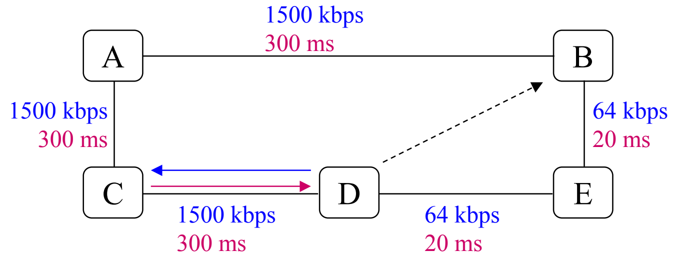
Using several metrics (1)

Using several metrics requires:

- Metrics must be stored for each link (***L.cost1***, ***L.cost2*** ...)
- The protocol must carry all metrics
- Computing separate routing tables for each metric (***P***(***cost1***), ***P***(***cost2***) ...)
- User packets must be marked with the required metric.

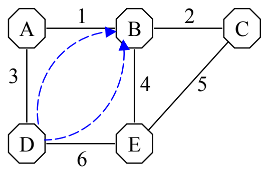
Using several metrics (2)

A routing loop is possible if different nodes use different metrics for one user packet



$\Rightarrow$ User packets must be marked with the required metric

Spreading load to alternative equidistant paths improves network efficiency

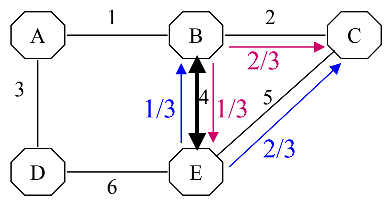


- ⊕ Queues in nodes become shorter
- ⊕ Average delay is decreased
- ⊕ End-to-end jitter decreases
- ⊕ Less traffic to reroute under failure conditions

- ⊖ May change packet order because paths may have different delay (queue lengths in nodes)
- ⊖ Existing traffic can not be pinned down to primary path so that only overload would take the alternative path ⇒ stability is a problem
- ? When are paths equidistant enough?

When are paths equidistant enough?

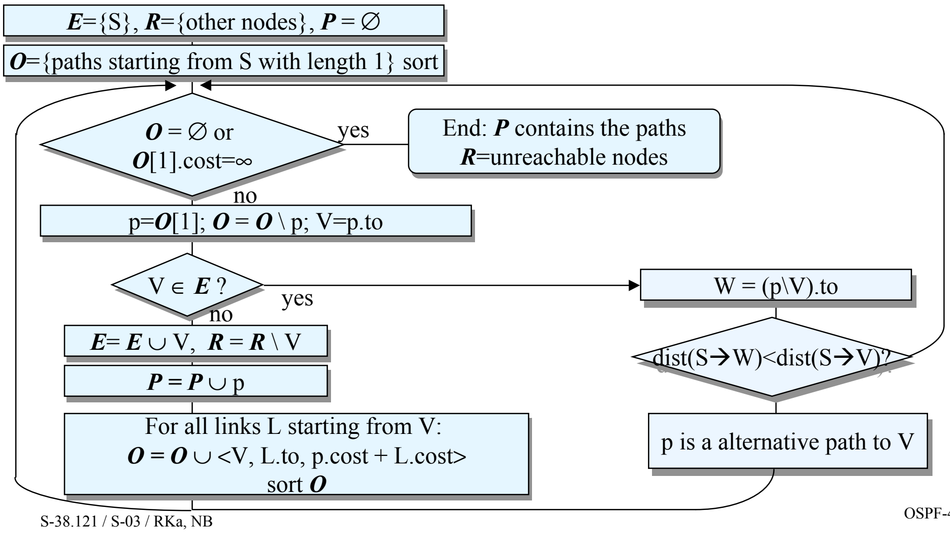
- What happens if the traffic to C is divided between two alternative paths?



⇒ The packet to X can be sent through Y is Y is closer to the destination than the local node

- Rule A→Y...→X, if distance(Y→X) < distance(A→X) accepts only monotonic alternative routes

Dijkstra's shortest-path-first algorithm that finds alternative paths



Dijkstra's shortest-path-first algorithm that finds alternative paths

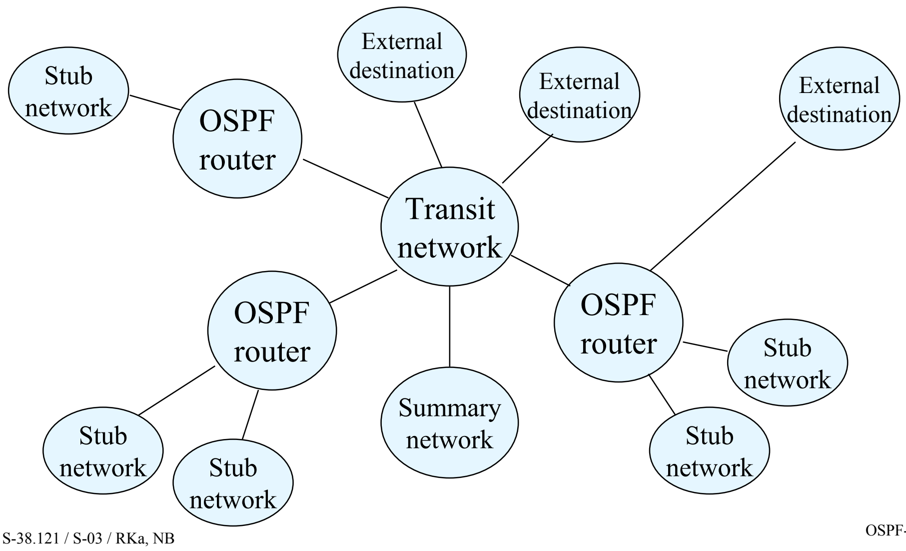
1. E={S}, R={N-S}, O={all one-hop paths starting from S}
2. If O is empty or is the first path in O has infinite length:
   - Mark all the remaining nodes in R as unreachable
   - Stop
3. P is the shortest path in O. Remove P from O. V is the last node of P.
4. Is V is in E:
   - Go to step 6
5. Create a set of paths by adding to P all links starting from V. The path is the previous path + the cost of the link. Add these paths to O in length order. Go to step 2.
6. If the distance of path P from S to V is the same as previously calculated distance from S to V
   - Add the alternative path to V.
7. Go to step 2

Link state protocol can describe several external routes with accurate metrics

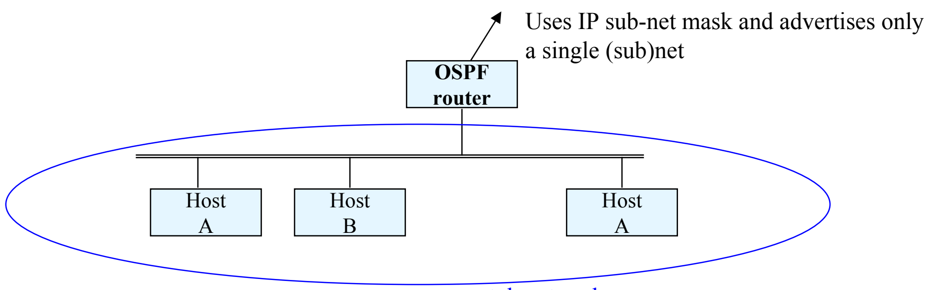
- DV-protocol capability to describe external routes is limited due to counting to infinity problem and due to complexity of Bellman-Ford algorithm
  - Inf=16 ⇒ maximum distance limited
  - Bellman-Ford complexity is O(N**2)
- Link state protocol is free from those limitations.
  - SPF route computation is O(N.logN)
    where N = number of external routes
- E.g. about 30 000 external routes ⇒ $9 \cdot 10^8$ vs. 450 000

The OSPF protocol

OSPF sees the network as a graph



OSPF makes a difference between a router and a host

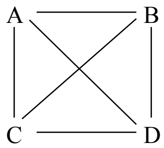


This creates two link state records:
+ router
+ stub network

OSPF supports broadcast networks (1)

In a broadcast network

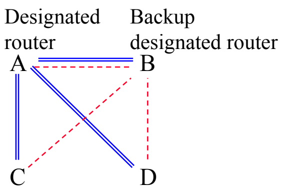
- Each device can send to each other
- One can send to all or to a sub-set of connected devices
- If it has N routers, they have $N*(N-1)/2$ adjacencies and
- Each router would advertise N-1 routes to other routers + one stub network $\Rightarrow N^2$

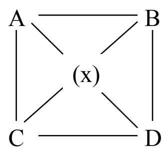


$N*(N-1)/2$ adjacencies (known neighbors)

E.g. Ethernet, Token ring, FDDI

OSPF supports broadcast networks (2)

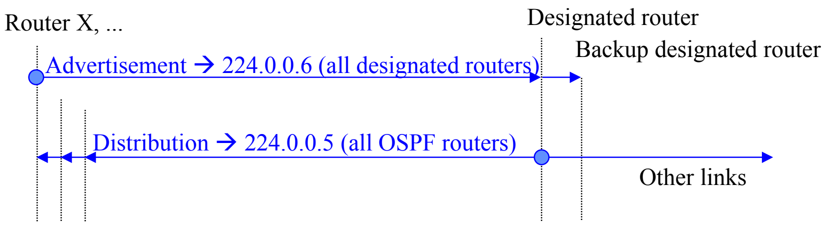




- Adjacencies are formed only with the ***designated router*** (A)
  - ⇒ Must be selected using the Hello protocol
  - ⇒ Synchronization of link DBs becomes simpler
- ***Backup designated router*** (B) is selected together with the designated.

- The broadcast network is modeled using a ”***virtual router***”
- The links from the virtual router are ***network links***
  - Advertised by the designated router
  - Cost = 0
- The links from the routers to the virtual router
  - Advertised by the routers

OSPF flooding protocol in a broadcast network

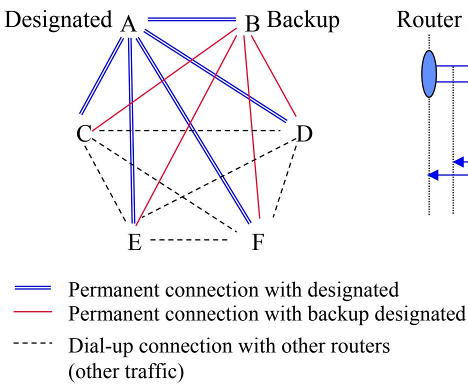


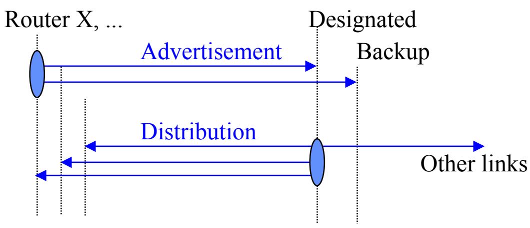
⇒ No need to process acks from all other routers in the sub-net

Backup designated stays as silent as possible

OSPF flooding protocol in a non-broadcast network

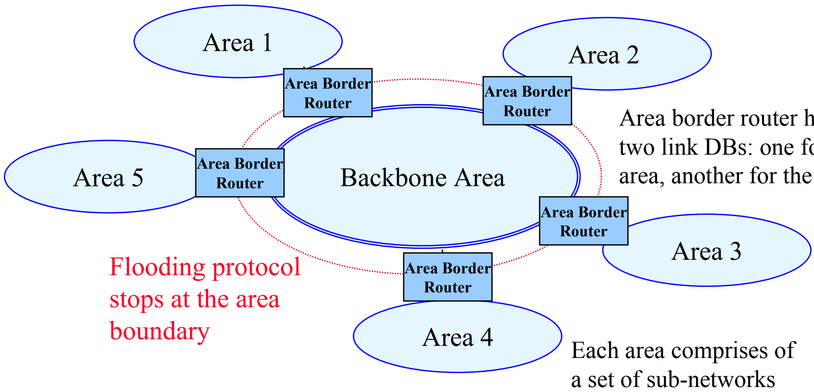
- In non-broadcast networks (e.g. X.25, ATM, frame relay), OSPF works in the same way except that broadcasts are replaced by point-to-point messages





NB: it makes sense to minimize permanent connections due to their cost

By breaking down a large network into areas, OSPF eases flooding and reduces the size of link DBs



Area border router has two link DBs: one for the area, another for the BB

Flooding protocol stops at the area boundary

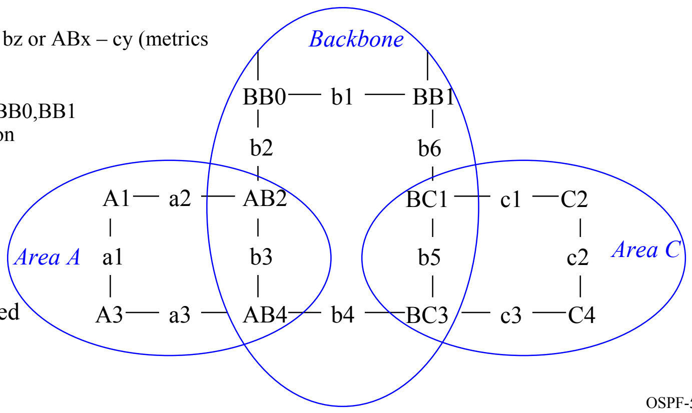
Each area comprises of a set of sub-networks

All OSPF routers on an area have identical link databases

(Sub)networks of other areas are described in summary records – the metric is computed in “RIP-style”

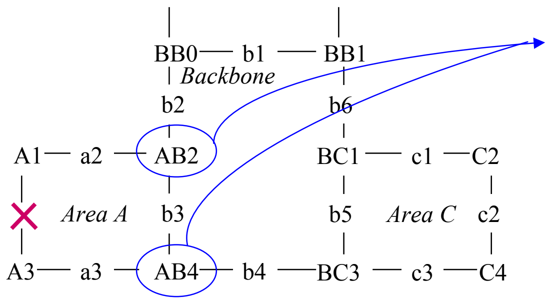
Link DB for Area A:

- a1, a2, a3
- Sub-net records of backbone and Area C (*summary records*)
  - ← AB2,AB4
  - Distance ABx – bz or ABx – cy (metrics are summed).
- *External records*
  - ←AB2,AB4 ←BB0,BB1
  - Same information in all areas



Hierarchy:
areas are only connected through the backbone
⇒ No loops

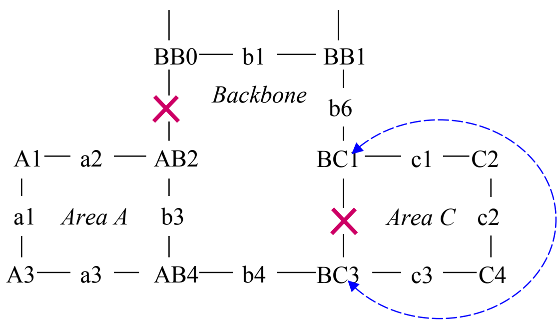
OSPF easily recovers from failures in areas



AB2 and AB4 advertise only those sub-nets which they can reach: AB2: a2 and AB4: a3.

Backbone does not know exact structure of area A - but it knows identities of all reachable sub-nets.

A virtual link can help if backbone splits into isolated segments due to a failure



Virtual link through Area C: distance=c1+c2+c3

In a *Stub Area*, all external routes are summed to the default route

- If an OSPF area has only one area border router, all traffic to and from the Internet goes through this ABR. It is of no use to advertise all Internet routes separately towards such an area.
- There can even be several ABRs, but it is not possible to select the best of them based on destination prefix (leading bits (<32) of IP address)
- A "*Not So Stubby Area*" (NSSA) is an area, on which all external routes have been summed into the default route except for some.

OSPF link state records

Link State Advertisement (LSA) types in OSPF

- LS Type = 1 **Router LSA**
  - describes set of active interfaces and neighbors
- LS Type = 2 **Network LSA**
  - describes a network segment (BC or NBMA) along with the IDs of currently attached routers
- LS Type = 3 **Summary LSA for IP Network**
- LS Type = 4 **Summary LSA for Border Router**
- LS Type = 5 **External LSA**
  - describes external routes
- LS Type = 6 Group Membership LSA
  - used in MOSPF for multicast routing
- LS Type = 7 Not So Stubby Area LSA
  - to import limited external info
- LS Type = 8 (proposed) external attributes LSA
  - in lieu of Internal BGP

Hierarchical Routing

BC = Broadcast, e.g. Ethernet
NBMA = Non-Broadcast Multiple Access, e.g. ATM

Common header of Link State Advertisement (LSA)

LS age	**options**	**LS type**
Link state ID		
Advertising router		
LS sequence number		
LS checksum	**length**	

32 bits

Key

LS age:
- Seconds from advertisement

Options:
- E - external links
- T - type of service
  - when many metrics are in use

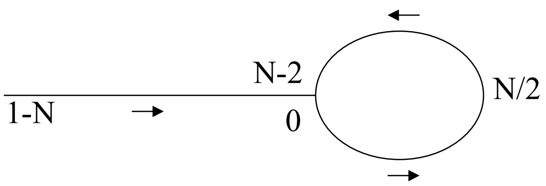
LS checksum:
- Protects header and content

Length:
- Total length of the record

Link state ID:
- Depends on LS type

LSA Sequence Numbers

- ”Lollipop sequence space”



$N = 2^{31}$

- If one of the number is < 0
  - The higher number is newer
- If both numbers are ≥ 0
  - If (b-a) < (N-1)/2 then b is newer

LSA Sequence Numbers

S_{Max} = 0x7fffffff

S_0 = 0x80000001
Initial Seq Nr

- To roll the space over, first delete record with S_{Max}
- A router may update a self originated record only once in 5 sec.
- In absense of errors rolling the space over takes at least 600 years.
- LS Age is updated during flooding at each step. Records with max Age are discarded. This breaks inf loops.

Router LSA (type 1)

RouterType	0	Number of links
Link ID		
Link data		
Type	# TOS	TOS 0 metric
TOS=x	0	TOS x metric
TOS=y	0	TOS y metric
. . .		
TOS=z	0	TOS z metric

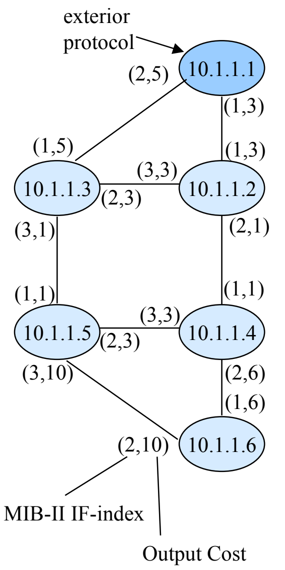
for each link

Router type
- E-bit (External)
  - This router is an area-border router
- B-bit (Border)
  - This router is a border router

TOS 0 metric when no TOS is used

Type
1. Link is a point-to-point link to another router
   - Link ID = router's OSPF ID, link data = router's interface address
2. Link connects to a transit network
   - Link ID = IP address of designated router's interface, link data = router's interface address
3. Link connects to a stub network
   - Link ID = Network/subnet number, link data = network/subnet mask

Example of the Router LSA



Router 10.1.1.1's router-LSA:

	LS Age = 0 seconds	Options	LS type=1	E-bit; 1=Router LSA
	Link State ID = 10.1.1.1			
	Advertising Router = 10.1.1.1			
	LS Sequence Number = 0x80000006			
	Checksum= 0x9b47		Length = 60 bytes	
0=ordinary	RouterType=0	0	Nrof links = 3	
	Link ID = 10.1.1.2 (neighb)			
	Link Data = IF-index 1 (unnum)			
1=pt-t-pt	Type=1	#TOS=0	Metric=3	
	Link ID = 10.1.1.3 (neighb)			
	Link Data = IF-index 2 (unnum)			
1=pt-t-pt	Type=1	#TOS=0	Metric=5	
	Link ID = 10.1.1.1			
	Link Data = 255.255.255.255			
3=stub network	Type=3	#TOS=0	Metric=0	

Length = 24 + 3 * 12 = 60 bytes

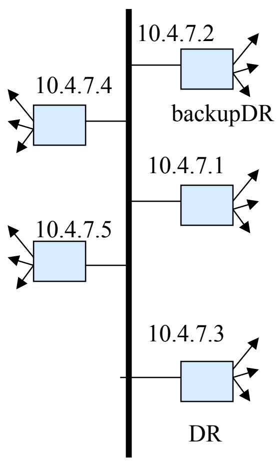
Router with 100 interfaces:
- Length = 24 + 100 * 12 = 1224 bytes

Network LSA (type 2)

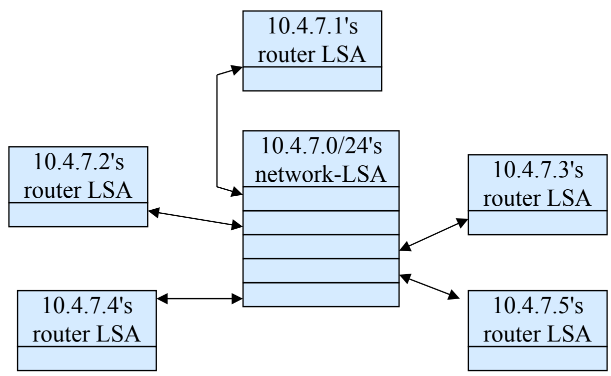
Network mask
Attached router
Attached router
. . .
Atttached router

- Advertised by designated routers for transit networks
- Link state ID = IP interface ID
- Attached router = OSPF identifier of the attached router

Network LSA reduces LinkDB for BC networks



DR - designated router



Network-LSA is generated by DR
Network-LSA reflects DB sync status with DR both ways
This reduces nrof link records from O(n*(n-1)) to 2*n.
Particularly important if Network is ATM or FR with a lot of routers attached!

Summary Link LSA (type 3,4)

Network mask		
0	0	TOS 0 metric
TOS=x	0	TOS x metric
TOS=y	0	TOS y metric
. . .		
TOS=z	0	TOS z metric

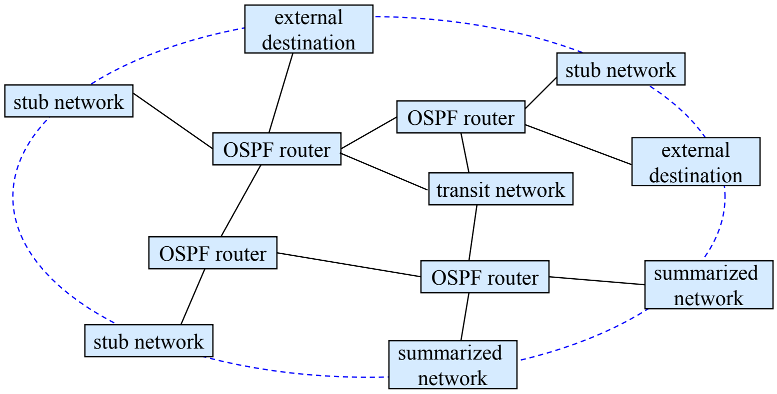
- For IP networks
  - LS type = 3
  - Network mask of network/subnet
  - Link state ID = IP network/subnet number
- For border routers
  - LS type = 4
  - Network mask = FFFFFFFF
  - Link state ID = IP address of border router
- One separate advertisement for each destination

External Link LSA (type 5)

Network mask		
E,TOS=0	0	TOS 0 metric
External route tag (0)		
E,TOS=x	0	TOS x metric
External route tag (x)		
. . .		
E,TOS=z	0	TOS z metric
External route tag (z)		

- Advertised by border routers
  - Information from external gateway protocols (BGP-4)
- One destination per record
- Link state ID = IP network/subnet of destination
- Network mask = network/subnet mask
- E-bit indicates that distance is not comparable to internal metrics
  - Larger than any internal metric
- Route tag is only used by border routers (not used by OSPF)

Computation of routes

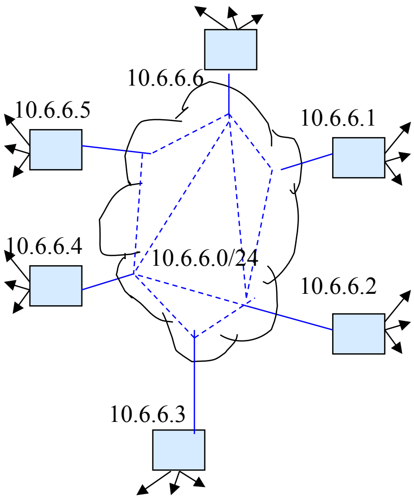


- Separate routes for each TOS and for TOS 0
  - Possibly unreachable destinations for some TOS if not all routers support TOS ⇒ routed with TOS 0 (no loops because same decision in all nodes)

Nonbroadcast multiaccess (NBMA) subnets support many routers communicating directly but do not have broadcast capability

- Examples are ATM, Frame Relay, X.25
- IP routing requires more manual configuration
- Designated router and backup DR concept reduce the number of adjacencies
- The model is prone to failures that may be hard to track

Point-to-multipoint subnet is more robust but less efficient



- There is no DR nor backup DR
- Every OSPF router maintains adjacencies with all neighbors with whom it has direct connectivity
- Alternative is a set of NBMA networks
- Next hop routing protocol improves scalability

The OSPF protocol

OSPF packets – the protocol itself

- OSPF works directly on top of IP.
  - OSPF protocol number is 89.
- For most packets TTL = 1, except for hierarchical routing
- Destination IP address =
  - Neighbors IP address or
  - AllOSPFRouters (224.0.0.5) or
  - AllDesignatedRouters (224.0.0.6)
- OSPF has 3 sub-protocols:
  - Hello protocol
  - Exchange protocol
  - Flooding protocol

The common OSPF message header

Version	Type	Packet length
Router ID		
Area ID		
Checksum		Authentic. type
Authentication		
Authentication		

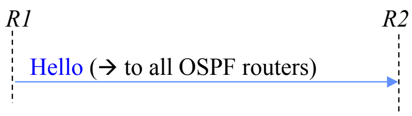
Type differentiates messages

- Type 1: Hello
- Type 2: Database Description
- Type 3: Link State Request
- Type 4: Link State Update
- Type 5: Link State Ack

- Current version of OSPF is 2
- Area ID
  - Usually a (sub)network number
  - 0 = Backbone
- Authentication type
  - 0 = No authentication
  - 1 = Password
    - Limited protection
  - 2 = Cryptographic authentication
    - MD5

0	Key ID	Length
Cryptographic sequence number		

Hello protocol ensures that links are working and selects designated router and backup DR



OSPF packet header type = 1		
Network mask		
Hello interval	Options	Priority
Dead interval		
Designated router		
Backup designated router		
Neighbor		
- - -		
Neighbor		

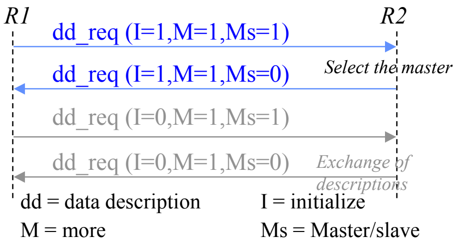
- Neighbors – a list of neighbors that have sent a hello packet during last dead interval seconds.
- Hello interval tells how often in seconds hello packets are sent.
- Priority tells about eligibility for the role of designated router.
- A hello packet must be sent in both directions before a link is considered operational

- Options
  - E = external route capability.
  - T = TOS routing capability.
- DR and Backup DR = 0 if not known

The Hello protocol selects the DR and the backup DR

1. Eligibility is achieved after one dead interval provided two-way reachability is OK.
2. From the routers that announced eligibility, the one with highest priority is elected to Backup Designated. Tie is broken by electing the one with highest ID.
3. If no neighbor proposed itself to backup DR, the neighbor with the highest priority is selected. Tie is broken by selecting the one with highest ID.
4. If one or several neighbors proposed themselves as designated router, the one with the highest priority is selected. Tie is broken by selecting the one with highest ID.
5. If none proposed itself to DR, the backup DR is promoted. Actions 2 and 3 are repeated to re-select the backup DR.
6. A high priority former DR postpones its proposal to retake the position of DR after recovery to minimize changes. Actions 2....5 are continuous.

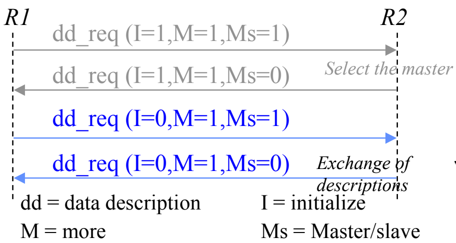
Exchange protocol initially synchronizes link DB with the designated router (1)



OSPF packet header type = 2 (dd)			
0	0	Options	0 IMMs
dd sequence number			

- Exchange protocol uses database description packets
- First the master and slave are selected
- If both want to be masters, the highest address wins
- Retransmission if the packet is lost
- The same sequence number in the replies

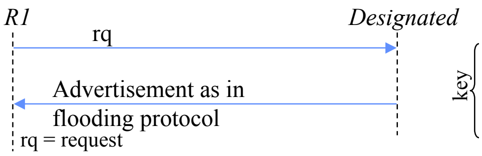
Exchange protocol initially synchronizes link DB with the designated router (2)



OSPF packet header type = 2 (dd)			
0	0	Options	0 IMMs
dd sequence number			
Link state type (key)			
Link state ID (key)			
Advertising router (key)			
Link state sequence number			
Link state checksum		Link state age	
- - -			

- Master sends its Link DB description in sequence numbered packets
- Slave acks by sending its corresponding description packets.
- Exchange continues until all descriptions are sent and acknowledged. (M=0)
- Differences are recorded on the list of “records-to-request”.

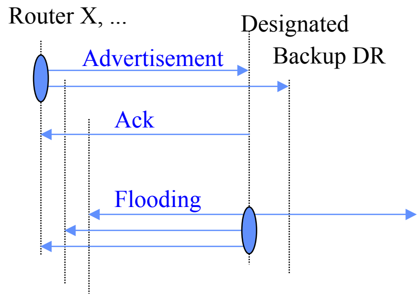
Request packets are used to get record contents. Requests are acknowledged by flooding protocol packets



OSPF packet header type = 3 (rq)
Link state type
Link state ID
Advertising router
- - -

- Router waits for ack for resend interval. If no response, the request is repeated.
- The records to request may be split into many requests, there are too many.
- If something goes wrong, the typical remedy is to restart role negotiation.
- The first request can be sent immediately when the first differing record has been detected. Then dd-packet exchange and rq packet exchange take place in parallel.

The flooding protocol continuously maintains the area's Link DB integrity



OSPF packet header type = 4 (upd.)
Number of advertisements
Link State Advertisements (*see LSA format*)
- - -

OSPF packet header type = 5 (ack.)
LSA headers
- - -

- Original LSA is always sent by the router responsible for that link.
- Advertisement is distributed according to flooding rules to the area (age=age+1).
- Ack of a new record by DR can be replaced in BC network by update message.
- One ack packet can acknowledge may LSAs.
- By delaying, several acks are collected to a single packet

Summary of OSPF subprotocols

	Hello (1)	DD (2)	LS rq (3)	LS upd (4)	LS ack (5)
Hello protocol	X				
Database exchange		X	X	X	X
Flooding protocol				X	X

Server Cache Synchronization Protocol (SCSP) is OSPF without Dijkstra's algorithm and with more generic data objects.

Link records have an age, old/dead ones are removed from Link DB (1)

- Old information must be removed from DB
- Every node must use the same information
  ⇒ The removals must be synchronized
- The LSAs of OSPF have an age
  - Age = 0 when the advertisement is created
  - Age = number of hops through which the advertisement has traveled + seconds from reception
- Max age is 1 hour
  - Not used in the calculation of routes
  - Must be removed
- Every entry must be advertised at a 30 min interval.
  - The new advertisement zeroes the age and increments the sequence number.

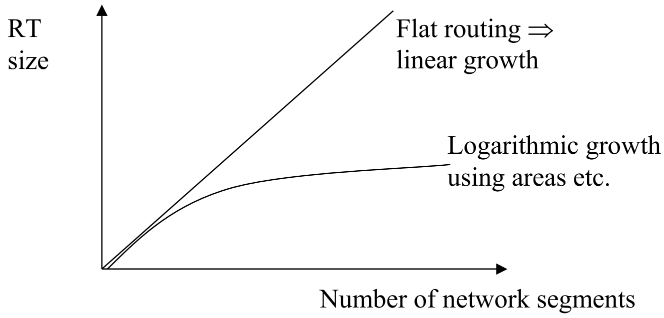
Link records have an age, old/dead ones are removed from Link DB (2)

- When the age reaches MaxAge (= 1 h) the entry is removed
  - The router must send an advertisement to the neighbors when the aged entry is removed
- The flooding algorithm examines the age of the received advertisement
  1. MaxAge advertisement is accepted and flooded – this removes obsolete info.
  2. If the age difference of the advertisement to the DB is small, the advertisement is not flooded to avoid overloading the network with multiple copies of the same info. This is due to normal routing when the entry is received on different paths.
  3. If the age difference is large (>MaxAgeDiff), the newest advertisement is accepted and distributed. In this case, the router has probably been restarted.
  4. If a MaxAge record is not found, advertisement has not impact. The router most likely has already removed the dead LSA.

OSPF timeouts – LS Age field

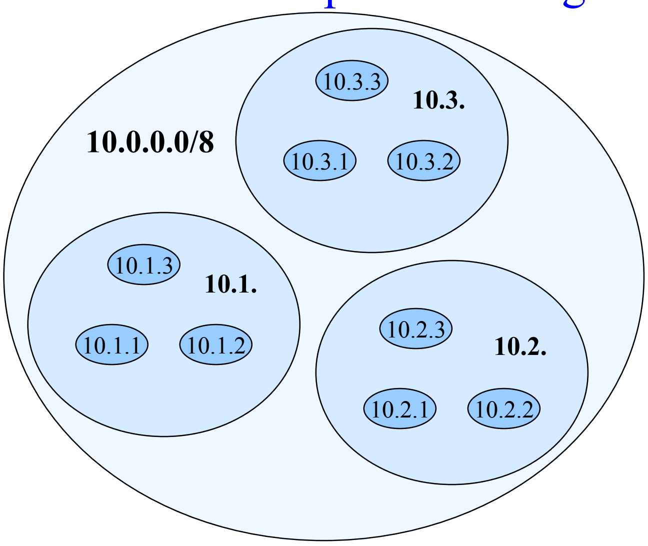
Constant	Value	Action of OSPF router
MinLSArrival	1 second	Max rate at which a router will accept updates of any LSA via flooding
MinLSInterval	5 seconds	Max rate at which a router can update an LSA
CheckAge	5 min	Rate to verify an LSA Checksum in DB
MaxAgeDiff	15 min	When Ages differ more than 15 min, they are considered separate. Smaller LS age - newer!
LSRefreshTime	30 min	A Router must refresh any self-originated LSA whose age has reached 30 min.
MaxAge	1 hour	LSA is removed from DB.

The purpose of hierarchical routing in OSPF is to reduce routing table growth



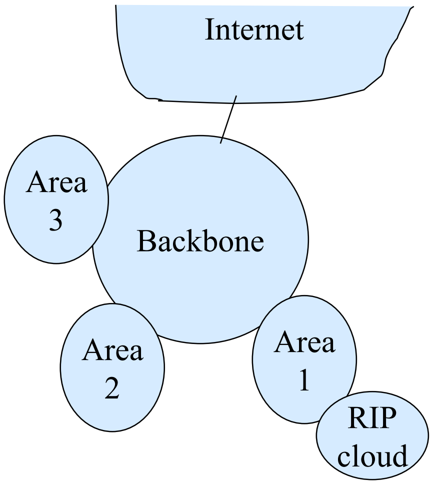
The cost is: sometimes suboptimal routes.

Example of routing hierarchy



Example:
- 16 segments in each lowest level network
- flat routing:
  RTsize= 16*9=144
- areas 10.1.1:
  16 local routes +
  10.1.2/24
  10.1.3/24
  10.2/16
  10.3/16
  == 20 RT entries!

OSPF supports 4 level routing hierarchy



Level	Description
1	Intra-area routing
2	Inter-area routing
3	External Type 1 metrics
4	External Type 2 metrics

- Type 1 metrics are of the same order as OSPF metrics, e.g. hop count (for RIP and OSPF)
- Type 2 metrics are always more significant than OSPF internal metrics

Why is it difficult to route packets around network congestion?

- BBN ARPANET link state metric varied with the length of the output queue of the link $\Rightarrow$ lead to route trashing.
- The problem is there is no route pin-down for existing traffic.
- By limiting the range of the metric changes, an equilibrium could be reached. Nevertheless routing instability is the problem.

When QoS or Class of Service a'la DiffServ is introduced this problem again becomes important.

OSPF development history