OSPF
Open Shortest Path Protocol

- Link state routing protocol developed by the IETF for use in the internet (RFCs 1583, 2178, 2328)
  - “Distributed Map” Concept
  - Flooding protocol for the dissemination of information
- Advantages over Distance Vector Routing Protocols
  - Fast, loopless convergence
  - Precise metrics, and if needed, multiple metrics per link
  - Supports multiple paths to a destination, that can be used simultaneously

OSPF Features

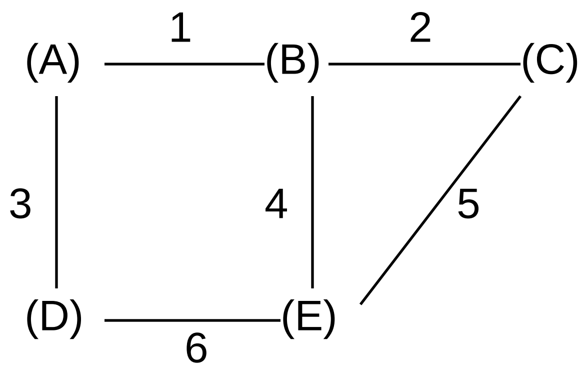
- Features :
  - type of service routing
  - load balancing (multiple routes to a destination)
  - network partitioning (areas made independent of each others)
  - authentication of exchanges between routers
  - supports host-specific routes, network-specific routes, and subnet routes
  - reduction of the routing traffic on broadcast networks by means of a designated router
  - supports exchange of information learned from other (external) sites

Basic Idea

- Each router has a complete map of the network topology.
- The map is built by “flooding”:
  - Each router advertises the state of all its interfaces (their costs, and where it connects to).
  - These link state advertisements are flooded through the network; upon reception, the other routers repeat them on all their interfaces.
  - Advertisements have sequence numbers.
- Given the map, each router uses Dijkstra’s algorithm to compute the shortest path tree from itself to all other routers.

The Distributed Map

- "Distributed Map" concept :



network

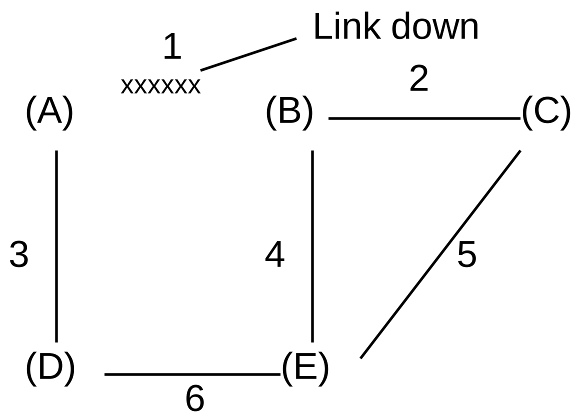
From	To	Link	Distance
A	B	1	1
A	D	3	1
B	A	1	1
B	C	2	1
B	E	4	1
C	B	2	1
C	E	5	1
D	A	3	1
D	E	6	1
E	B	4	1
E	C	5	1
E	D	6	1

Database

  – Every router has a copy of the distributed map in memory

Updating the Database

- Flooding Protocol
- Database is updated after each change of link state



From	To	Link	Distanc	Number
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

The database after flooding

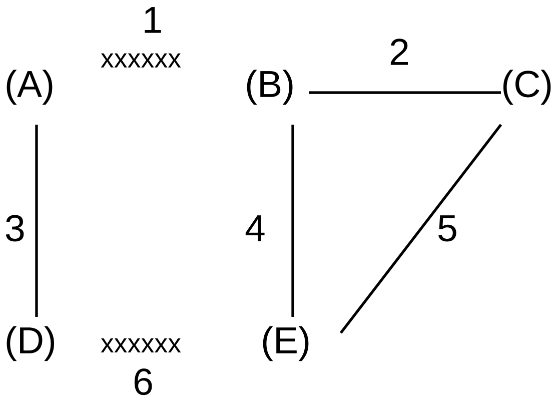
Message < From A, to B, link 1, distance = infinite >
Need : Timestamp or message number

Flooding algorithm

- 1. Receive the message. Look for the record in the database
- 2. If record is not present, add it to the database and broadcast the message
- 3. Else, if the number in the database is lower than the number in the message, replace record with new value, and broadcast the message
- 4. Else if the number in the database is greater than the number in the message, transmit the database value in a new message through the incoming interface
- 5. Else, if both numbers are equal, do nothing

Map Inconsistency

- Possibility of inconsistency in maps



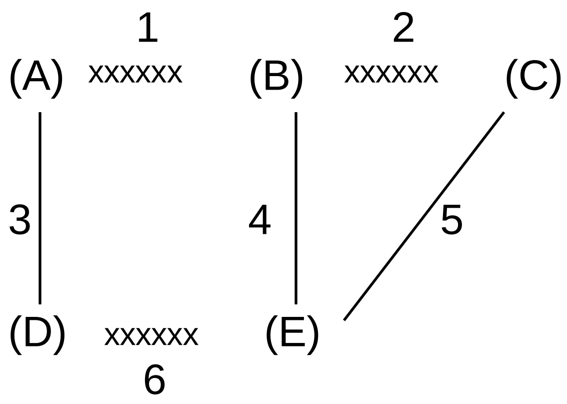
From	To	Link	Distanc	Number
A	B	1	inf	2
A	D	3	1	1
B	A	1	inf	2
B	C	2	1	1
B	E	4	1	1
C	B	2	1	1
C	E	5	1	1
D	A	3	1	1
D	E	6	inf	2
E	B	4	1	1
E	C	5	1	1
E	D	6	1	1

The database in nodes A and D

From	To	Link	Distance	Number
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

The database in nodes B,C and E

Inconsistency (cont.)



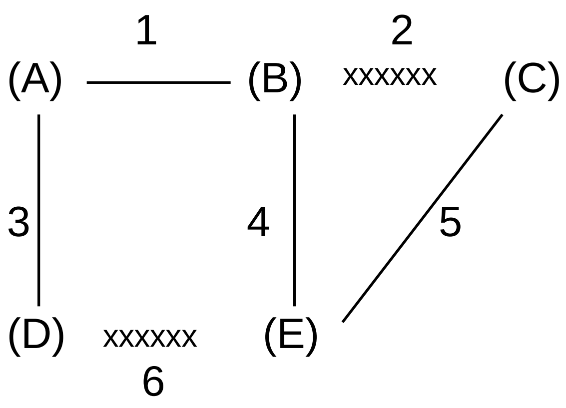
From	To	Link	Distanc	Number
A	B	1	inf	2
A	D	3	1	1
B	A	1	inf	2
B	C	2	1	1
B	E	4	1	1
C	B	2	1	1
C	E	5	1	1
D	A	3	1	1
D	E	6	inf	2
E	B	4	1	1
E	C	5	1	1
E	D	6	1	1

The database in nodes A and D

From	To	Link	Distance	Number
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

The database in nodes B,C and E

Inconsistency (cont.)



From	To	Link	Distanc	Number
A	B	1	1	1
A	D	3	1	1
B	A	1	1	1
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

The database in nodes A and D

From	To	Link	Distance	Number
A	B	1	1	1
A	D	3	1	1
B	A	1	1	1
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

The database in nodes B,C and E

Synchronizing Databases

- Neighboring routers need to “bring up the adjacency” (synchronize their databases).
- Made easy by the existence of link identifiers and version numbers
  - Links are identified by the network IP address.
- Exchanging complete copies of databases is inefficient
- OSPF defines “database description” packets
  - link identifiers and version numbers only
- Neighboring routers will synchronize their databases :
  - Phase 1 - routers will send complete description of their databases - compile list of interesting records (ones that are newer than their local records)
  - Phase 2 - each router polls its neighbor for a full copy of interesting records by means of “link state request” packets

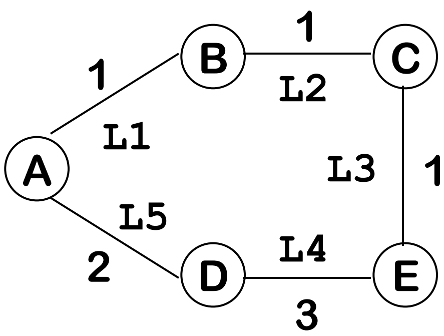
Securing the Map Updates

- Coherency of routing is fully dependent on maintaining synchronized copies of databases in all nodes
- Each router is only required to be synchronized with its neighbors
- Measures introduced in OSPF
  – a) flooding procedure include hop-by-hop acknowledgment
  – b) Database description packets are transmitted in a secure fashion
  – c) each link state record is protected by a timer and is removed from the database if not refreshed in due time
  – d) all records are protected by a checksum
  – e) messages can be authenticated, e.g. by passwords, or encrypted

OSPF Algorithm (Dijkstra's)

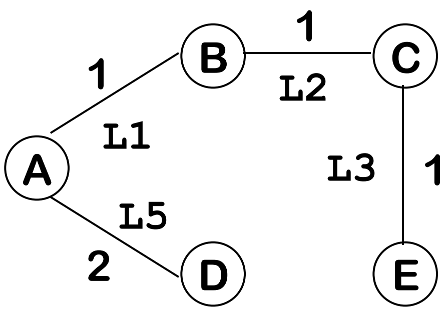
- 1. Initialize the set ***E*** to contain only the source node S and the set ***R*** to contain all other nodes. Initialize the list of paths ***O*** to contain all the one hop paths starting from S. Each of these paths has a cost equal to the corresponding link's metric. Sort list ***O*** by increasing metrics.
- 2. If list ***O*** is empty, or if the first path in ***O*** has an infinite metric, mark all nodes left in ***R*** as unreachable. The computation is finished.
- 3. First examine P, the shortest path in list ***O*** . Remove P from ***O*** . Let V be the last node in P. If V is already in set ***E***, go back to step 2. Otherwise, P is the shortest path to V. Move V from ***R*** to ***E*** .
- 4. Build a set of new candidate paths by concatenating P and each of the links starting from V. The cost of these paths is the sum of the cost of P and the metric of the link appended to P. Insert the new links in the ordered list ***O*** , each at the rank corresponding to its cost. Go to step 2.

Example



Tree from A

Iteration	Set E	Set R	List O	Distance	Path P	Node V
1	A	B, C, D, E	A-L1-B	1	A-L1-B	B
			A-L5-D	2		
2	A, B	C, D, E	A-L5-D	2	A-L5-D	D
			A-L1-B-L2-C	2		
3	A, B, D	C, E	A-L1-B-L2-C	2	A-L1-B-L2-C	C
			A-L5-D-L4-E	5		
4	A, B, D, C	E	A-L1-B-L2-C-L3-E	3	A-L1-B-L2-C-L3-E	E
			A-L5-D-L4-E	5		
5	A, B, D, C, E	empty	A-L5-D-L4-E	5		



Shortest Path Tree

Advantages of OSPF

- Why is a link State Protocol Better?
  - 1. Fast, loopless convergence
  - 2. Support of precise metrics and, if needed, multiple metrics
  - 3. Support of multiple paths to destination

Fast, Loopless Convergence

- 1) Fast, Loopless Convergence
- Fast:
  - Distance vector protocol execute a distributed computation using the Bellman-Ford algorithm. The number of steps required is proportional to the number of nodes in the network.
  - Link state scenario, on the contrary, consists of two phases:
    - A rapid transmission of the new information through the flooding protocol.
    - A local computation
- Loopless:
  - Immediately after the flooding and the computation, all routes in the network are sane - no intermediate loops, no counting to infinity. Given the disruptive consequences of routing loops, this property alone is enough to make OSPF preferable to RIP.

Support for Multiple Metrics

- 2) Support of Multiple Metrics (1)
  - The shortest-path computation is executed with a full knowledge of the topology, one can use arbitrarily precise metrics without slowing the convergence.
  - Convergence speed is not a function of the metrics.
  - The precision of the computation makes it possible to support several metrics in parallel.

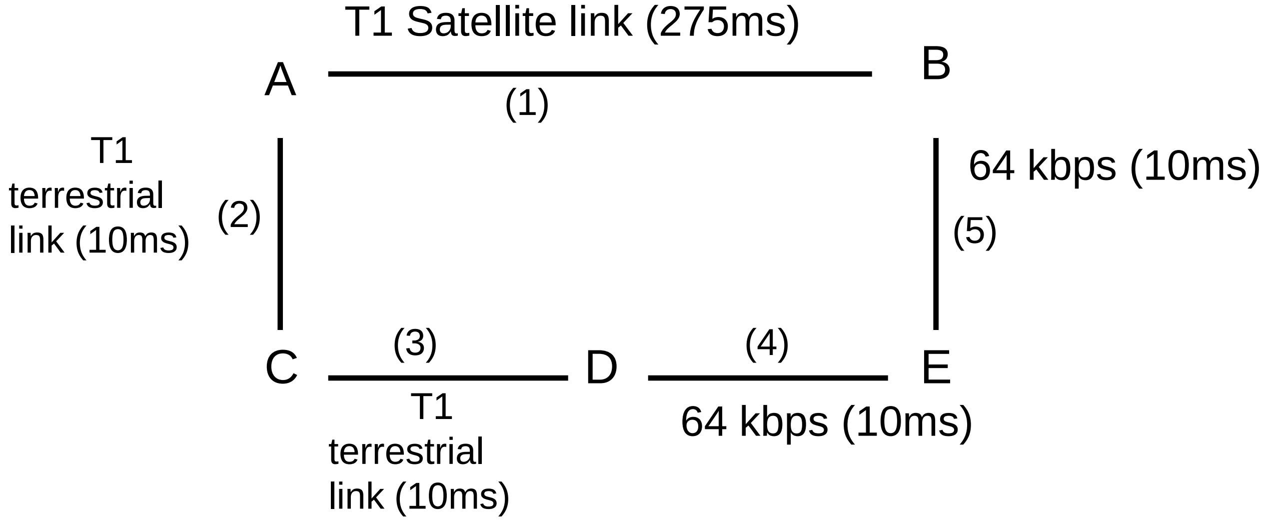
Multiple Metrics

- Possible metrics:
  - Largest throughput
  - Lowest delay
  - Lowest cost
  - Best reliability

Need to
- Document several metrics for each link
- Compute different routing table
- Present the selected metric in packet

Multiple Metrics Example

Support of Multiple Metrics



DCAB - 1.5 Mbps delay = 295ms
DEB - 64 kbps delay = 20ms

- Must make consistent decision in all nodes

Support for Multiple Paths

- 3) Multiple Paths
  - In complex networks, there are usually several "almost equivalent" routes toward a destination.
  - Mathematical analysis has proven that splitting the traffic over multiple paths is more efficient. This will lead to out-of-order delivery of some packets, but the average delay will be lower in the split-traffic case. The variations of the delay will also be lower due to the reduction in the correlation between packet arrivals on any single path.
  - Spreading the traffic also alleviates the effect of the disconnection in one single path. Without spreading the traffic, if the path becomes unavailable, all of a sudden the traffic will be routed through the alternate path, possibly leading to congestion of this path.

Algorithm for Multiple Paths

- Modified OSPF Algorithm

1. Initialize the sets E and R, and the list O, as in the standard SPF algorithm.
2. If O is empty, the algorithm is finished.
3. First examine P, the shortest path in the list O. Remove P from O. Let V be the last node in P. If V is already in the set E, continue at step 4. Otherwise, P is the shortest path to V. Move V from R to E. Continue at step 5.
4. Look at W, the node preceding V in the path P. If the distance from S to W is lower than the distance from S to V, then note P as an acceptable alternate path to V. In all cases, continue at step 2.
5. Build the new set of candidate paths, add them to O, as in the step 4 of the standard algorithm. Continue at step 2.

Issues

- Design of OSPF
  - Separating hosts and routers
  - Broadcast networks (Ethernet, FDDI …)
  - Non-broadcast networks (X.25, ATM)
  - Splitting very large networks into areas