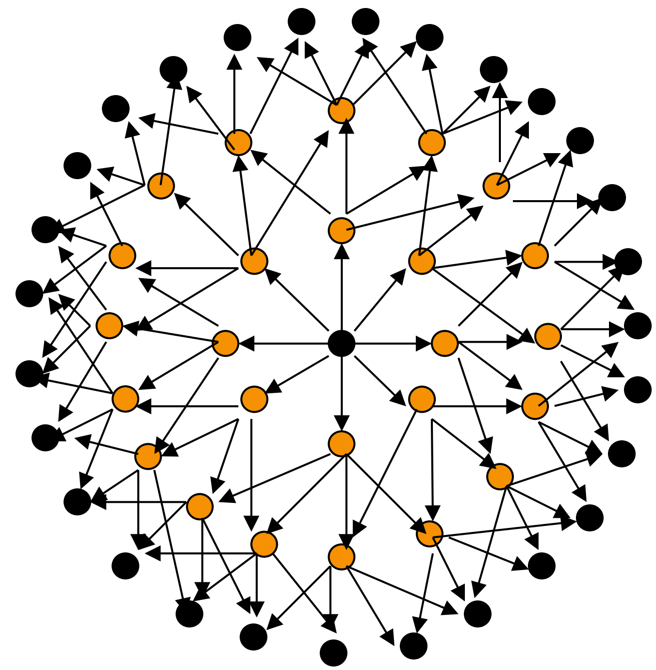
Optimized Link-State Routing OLSR

Some material borrowed from
Qamar Abbas Tarar and
Cholatip Yawut

Link-State Routing

- Each node periodically floods status of its neighboring links
- Each node re-broadcasts link state information received from its neighbour
- Each node keeps track of link state information received from other nodes
- Each node uses above information to determine next hop to each destination



24 retransmissions to diffuse a message up to 3 hops

Retransmission node

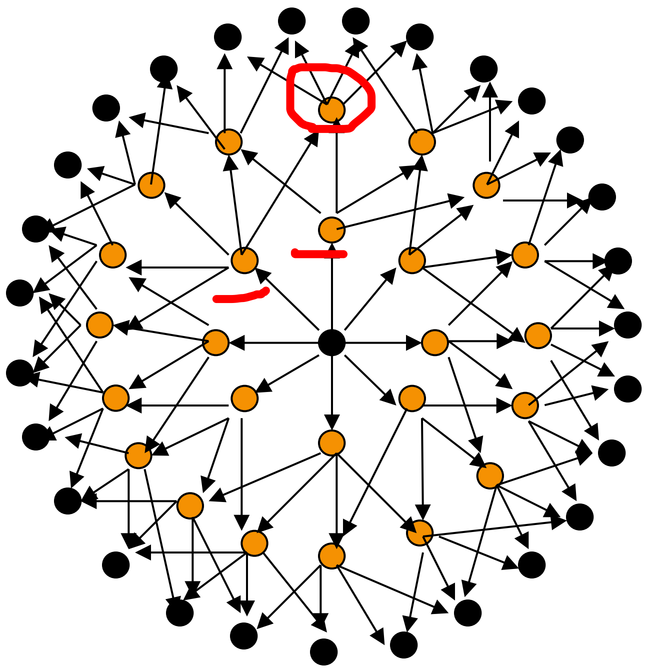
OLSR - Overview

- OLSR
  - Inherits Stability of Link-state protocol
  - Selective Flooding
  - only MPR retransmit control messages:
    - Minimize flooding
  - Suitable for large and dense networks

OLSR – Multipoint relays (MPRs)

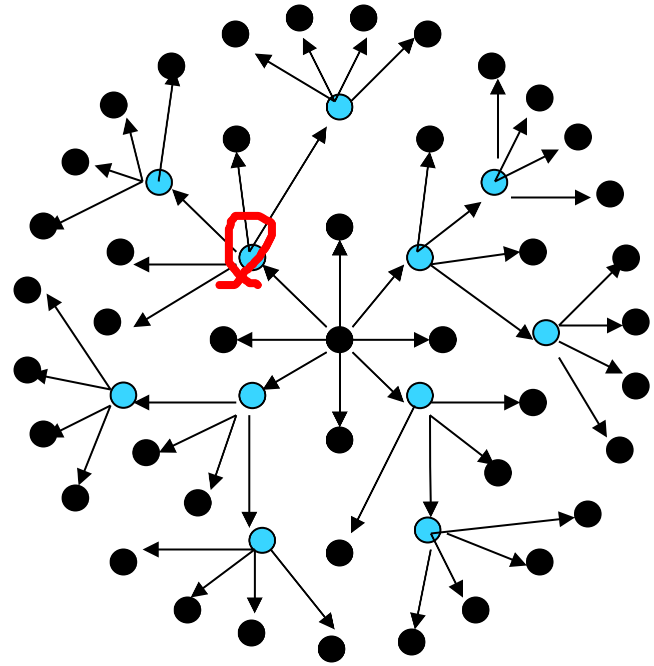
- MPRs = Set of selected neighbor nodes
  - Each node selects its MPRs among its one hop neighbors
  - **The set must cover all the nodes that are two hops away**
  - Used to minimize the flooding of broadcast packets
  - Link between node and it's MPR is **bidirectional**
- The information required to calculate the MPRs:
  - The set of one-hop neighbors and the two-hop neighbors
  - There are many ways you can select MPRs from your neighbors
- MPR Selector (MS) is a neighboring node which has selected me as a MPR

OLSR – Multipoint relays (cont.)



24 retransmissions to diffuse a message up to 3 hops

Retransmission node



11 retransmission to diffuse a message up to 3 hops

Retransmission node

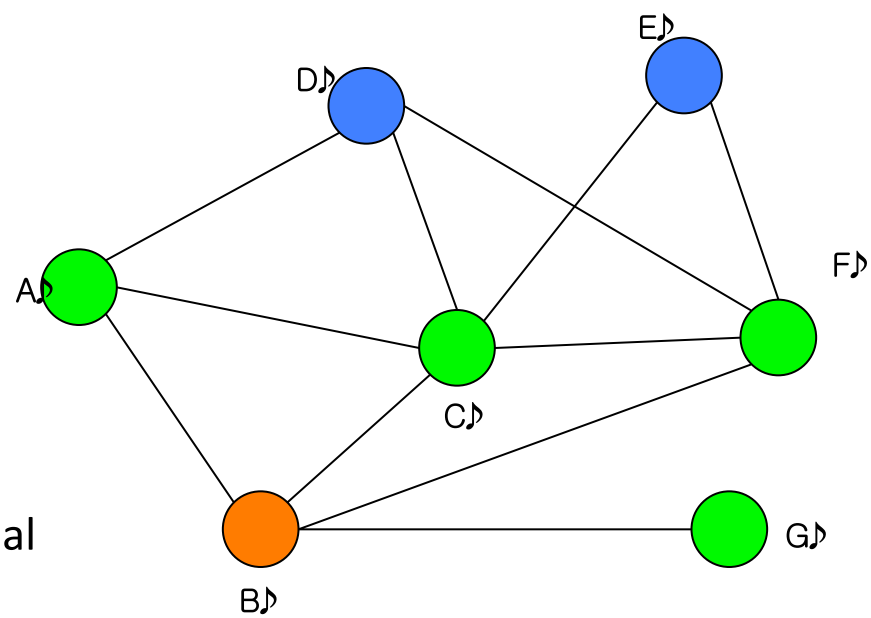
OLSR – Multipoint relays (cont.)

- To obtain the information about one-hop neighbors :
  - Use HELLO message (received by all one-hop neighbors)
- To obtain the information about two-hop neighbors :
  - Each node attaches the list of its own neighbors
- Once a node has its one and two-hop neighbor sets :
  - It can select a MPRs which covers all its two-hop neighbors

OLSR – Multipoint relays (cont.)

Node	1 Hop Neighbors	2 Hop Neighbors	MPR(s)
B	A,C,F,G	D,E	C

Hello Exchange:
A sends Hello to B
B adds A as a **unidirectional** neighbor
B sends hello to A (including A in its list one-directional neighbors)
A adds B as a bidirectional neighbor
A sends Hello to B with B in its bidirectional neighbor list
B adds A as a **bidirectional** neighbor



Neighbor Sensing

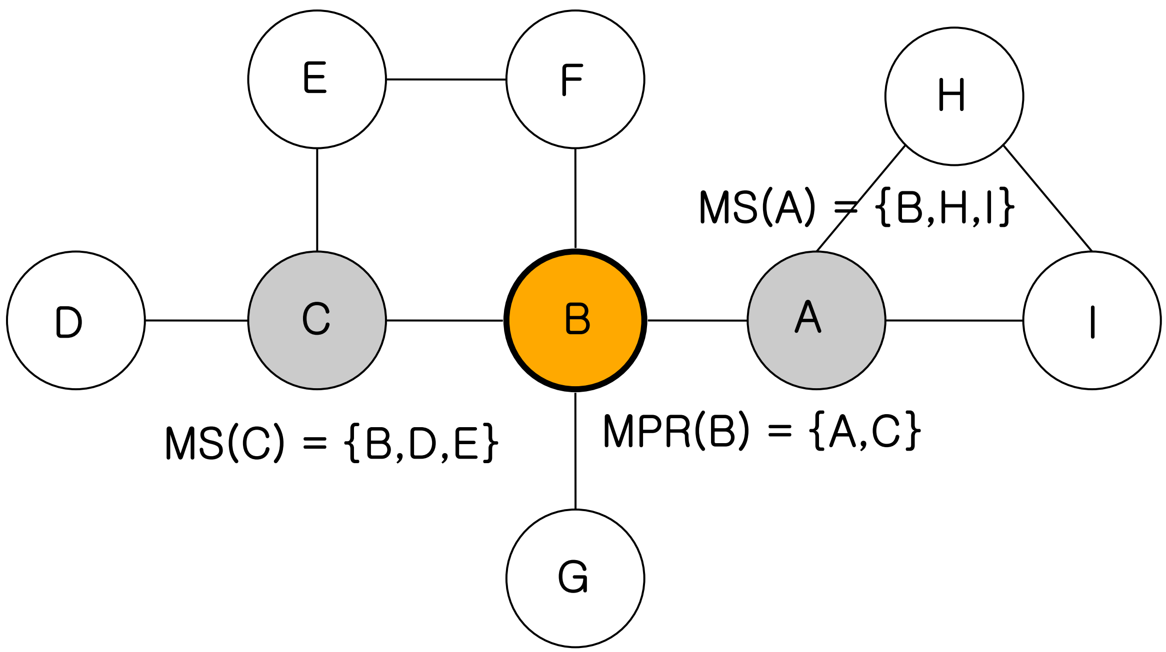
- Each node periodically broadcasts its HELLO messages:
  - Containing the information about its neighbors
  - Hello messages are received by all one-hop neighbors
- HELLO message contains (among other things):
  - List of addresses of the neighbors to which there exists a valid bi-directional link
  - List of addresses of the neighbors which are heard by the node (a HELLO has been received from them)
    - Note: link is not yet validated as bi-directional, could be one-directional!

Protocol functioning – Neighbor sensing (cont.)

- HELLO messages :
  - Serve as Link sensing
  - Permit each node to learn the knowledge of its neighbors up to two-hops (neighbor detection)
  - On the basis of this information, each node performs the selection of its multipoint relays (MPR selection)
  - ***The node's MPR set is included in the HELLO message***

- On the reception of HELLO messages:
  - Each node constructs its MPR Selector (MS) set
    - I.e., it can tell which of its neighbors have chosen it as an MPR

OLSR – Multipoint relays (cont.)



MPR and MS sets

Neighbor Table

- Each node records the information about its one hop neighbors and a list of two hop neighbors

- Entry in the neighbor table has a holding time
  - Upon expiry of holding time, it is removed

- Contains a sequence number value which specifies the most recent MPR set
  - Every time a node updates its MPR set, this sequence number is incremented
  - This is because you tell your neighbors about your own MPR set (distinguish old from new)

Neighbor Table Example

- Example of neighbor table of B

One-hop neighbors

Neighbor's id	State of Link
A	Bidirectional
G	Unidirectional
C	MPR
…	…

Two-hop neighbors

Neighbor's id	Access though
E	C
D	C
…	…

Multipoint relay selection

- Each node selects own set of multipoint relays
  - Multipoint relays are included in the transmitted HELLO messages

- Multipoint relay set is re-calculated when:
  - A change in the neighborhood (neighbor is failed or add new neighbor)
  - A change in the two-hop neighbor set

- Each node also construct its MPR Selector (MS) set with information obtained from the HELLO message

Finding a path to a destination

- How do you find a path to any node in the network?

- What you learn from HELLO messages is not enough!
  - You only learn about two hops away.

The Link-State Routing Part

- Every node forms a routing table of routes to all known destinations. How?

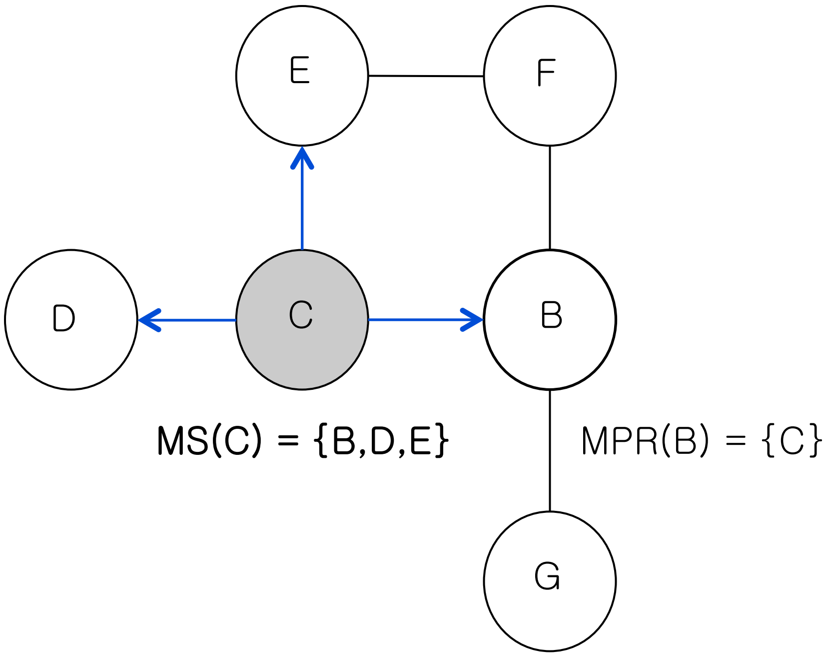
- Every node periodically broadcasts (i.e. a flood in the whole network) its MPR Selector Set (MS)
  - I.e., tell the whole network who has chosen you as an MPR
  - You DON'T mention your whole list of neighbors.

- BTW, if you don't mention your whole list of neighbors, how is it guaranteed that everyone can find a path to you????

Building the Topology

- TC – Topology control message:
  - Nodes with a non-empty MS periodically flood their MS via a TC message
  - Message might not be sent if there are no updates
  - **Contains:**
    - MPR Selector set (MS)
    - Sequence number
- Each node **maintains** a *Topology Table* based on TC messages
  - **Routing Tables** are calculated based on Topology tables

Building the Topology (cont)



Send TC message♪

{B,D,E} update their topology table♪

♪

TC message floods the whole network♪

♪

As it travels, each node updates its topology table ♪

Note: the whole network learns that B, D, and E can be reached via C.♪

TC Table Entry

X selected Y as one of its MPRs, so X is in the MPR Selector set of Y

Destination address	Destination's MPR	MPR Selector sequence number	Holding time
X	Y		

MPR Selector in the received TC message

Originator of TC message

Last-hop node to the destination.

Topology table entry in some node (not necessarily X or Y, just an arbitrary node P

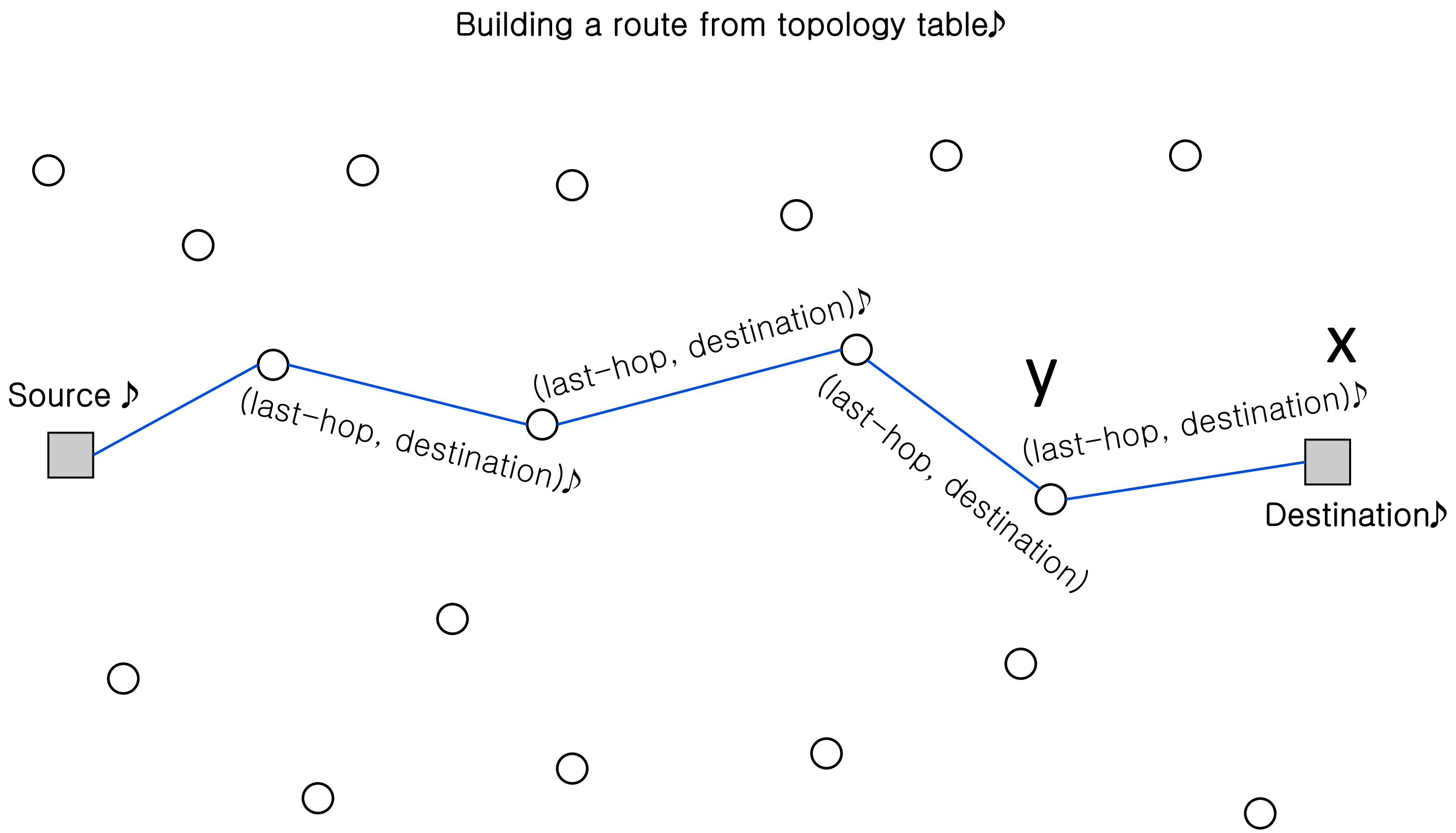
Building the Topology (cont)

- Upon receipt of TC message from Y:
  - If there exist some entry to the same destination X with **higher** Sequence Number, the TC message is ignored
  - If there exist some entry to the same destination X with **lower** Sequence Number, the topology entry is removed and the new one is recorded
  - If the entry (i.e. X, Y) is the **same** as in TC message, the holding time of this entry is refreshed
  - If X is no longer in MS of Y, remove entry (X, Y)
  - If there are **no** corresponding entry – the new entry (X, Y) is recorded

Protocol functioning – Routing table calculation

- Each node maintains a **routing table** (in addition to the TC table) to all known destinations in the network
- After each node TC message is received, you store connected pairs of form (node, last-hop) in the topology table
- Routing table is based on the information contained in the ***neighbor table and the topology table***
- Routing table:
  - Destination address
  - Next Hop address
  - Distance
- Routing Table is recalculated after every change in neighbor table or in topology table

Routing table calculation (cont.)



ALL INTERMEDIATE NODES ARE MPR's OF SOME NODE (y is mpr of x)

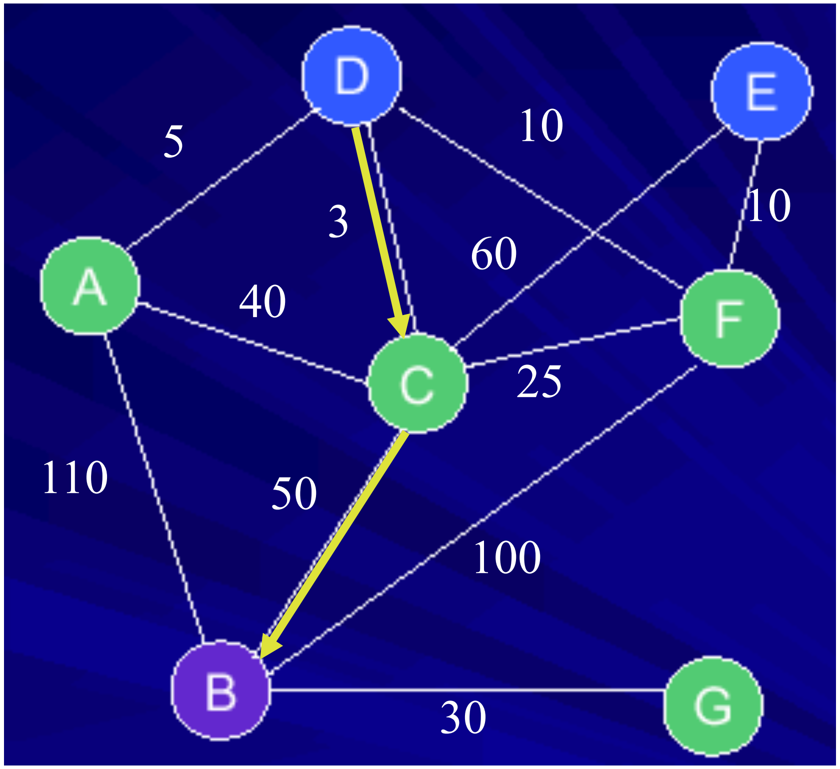
MPR's form a ``backbone'' network so to speak

Think of how a flood reaches from the Destination to the Source

Non-Optimal Paths (link bandwidth metric)

Node	1 Hop Neighbors	2 Hop Neighbors	MPR(s)
B	A,C,F,G	D,E	C

- Traditional OLSR: node B will select C as its MPR because it can reach the largest (actually all) two-hop neighbors
- Thus, all the other nodes know that they can reach B via C

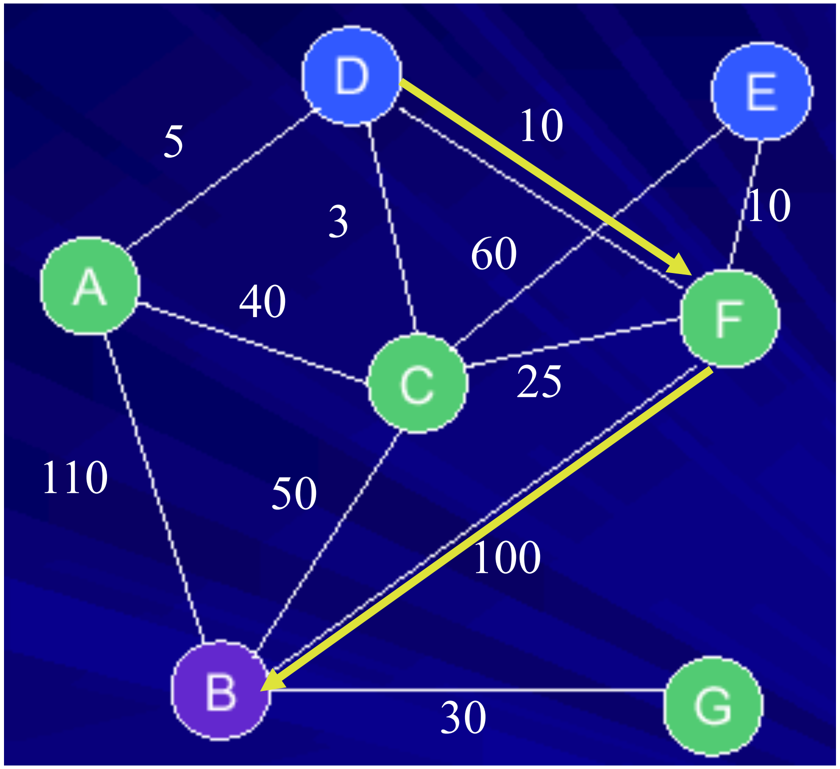


- However, think of bandwidth as the network metric
- D->B route is D-C-B, whose bottleneck BW is 3

Non-Optimal Paths

Node	1 Hop Neighbors	2 Hop Neighbors	MPR(s)
B	A,C,F,G	D,E	C

- Traditional OLSR: node B will select C as its MPR because it can reach the largest (actually all) two-hop neighbors
- Thus, all the other nodes know that they can reach B via C

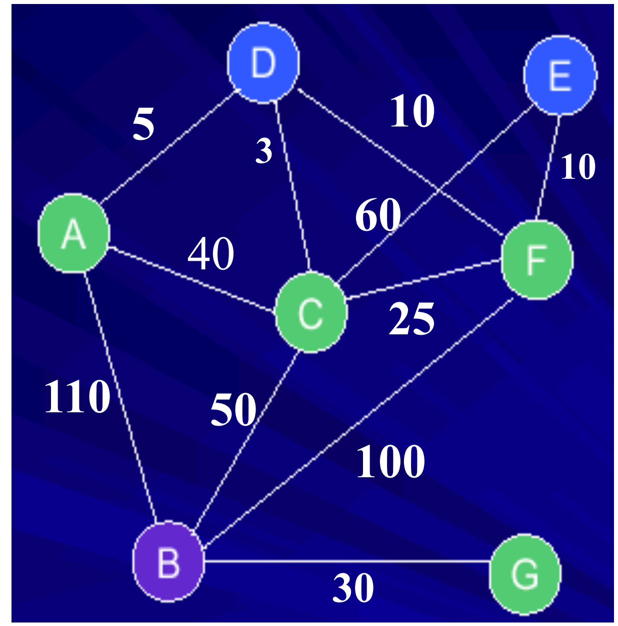


- Optimal route (i.e., path with maximum bottleneck bandwidth: D-F-B (bottleneck bandwidth of 10)

QoS Versions of OLSR

- **OLSR_R1**: similar to OLSR (i.e., choose 1-hop neighbours that cover max. number of 2-hop neighbours), tie-breaker now max BW

Node	1 Hop Neighbors	2 Hop Neighbors	MPR(s)
B	A,C,F,G	D,E	F



- **OLSR_R2**: select the best BW neighbors as MPRs until all the 2-hop neighbors are covered.

Node	1 Hop Neighbors	2 Hop Neighbors	MPR(s)
B	A,C,F,G	D,E	A,F

- **OLSR_R3**: selects the MPRs in a way such that all the 2-hop neighbors have the max. bottleneck BW path through the MPRs to the current node.

Node	1 Hop Neighbors	2 Hop Neighbors	MPR(s)
B	A,C,F,G	D,E	C,F

Conclusion

- OLSR protocol is proactive or table driven in nature
- Advantages
  - Route immediately available
  - Minimize flooding by using MPR
- OLSR protocol is suitable for large and dense networks
- Disadvantages
  - It does not guarantee an optimal path
  - Lot of control overhead if only a few pairs of nodes wish to communicate