Unicast Routing Protocols for Wireless Ad hoc Networks

CS: 647
Advanced Topics in Wireless Networks

Dr. Baruch Awerbuch & Dr. Amitabh Mishra
Department of Computer Science
Johns Hopkins

Reading

- Chapter 3 – Ad Hoc Networking, C. Perkins, Addison Wesley, 2001
- Chapter 5 - C. K. Toh, Ad Hoc Mobile Wireless Networks, Overview of Ad Hoc Routing Protocols", Prentice Hall, 2002
- Section 9.3, Jochen Schiller, *Mobile Communications*, Addison-Wesley, 2000
- Sections 13.5-13.6, D. P. Agrawal, Wireless and Mobile Systems, Thompson, 2003

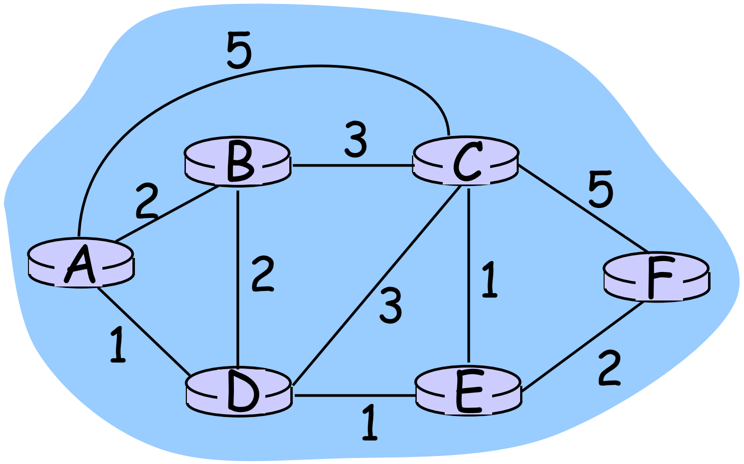
Outline

- Routing in the Internet
  - Distance Vector
- Why do we need routing protocols for Ad hoc networks?
- Unicast Routing in MANETS
  - Goals of routing protocols
  - Classification
    - Proactive or Table Driven
    - Reactive or On Demand
- Current approaches
  - DSDV (Distance Sequence Distance Vector)
  - OLSR

Routing

Routing protocol

Goal: determine "good" path (sequence of routers) thru network from source to dest.



Graph abstraction for routing algorithms:

- graph nodes are routers
- graph edges are physical links
  - link cost: delay, $ cost, or congestion level

- "good" path:
  - typically means minimum cost path
  - other def's possible

Internet - Routing Algorithm classification

Global or decentralized information?

Global:

- all routers have complete topology, link cost info
- "link state" algorithms

Decentralized:

- router knows physically-connected neighbors, link costs to neighbors
- iterative process of computation, exchange of info with neighbors
- "distance vector" algorithms

Static or dynamic?

Static:

- routes change slowly over time

Dynamic:

- routes change more quickly
  - periodic update
  - in response to link cost changes

A Link-State Routing Algorithm

Dijkstra's algorithm

- net topology, link costs known to all nodes
  - accomplished via "link state broadcast"
  - all nodes have same info
- computes least cost paths from one node ('source") to all other nodes
  - gives forwarding table for that node
- iterative: after k iterations, know least cost path to k dest.'s

Notation:

- c(x,y): link cost from node x to y; = ∞ if not direct neighbors
- D(v): current value of cost of path from source to dest. v
- p(v): predecessor node along path from source to v
- N': set of nodes whose least cost path definitively known

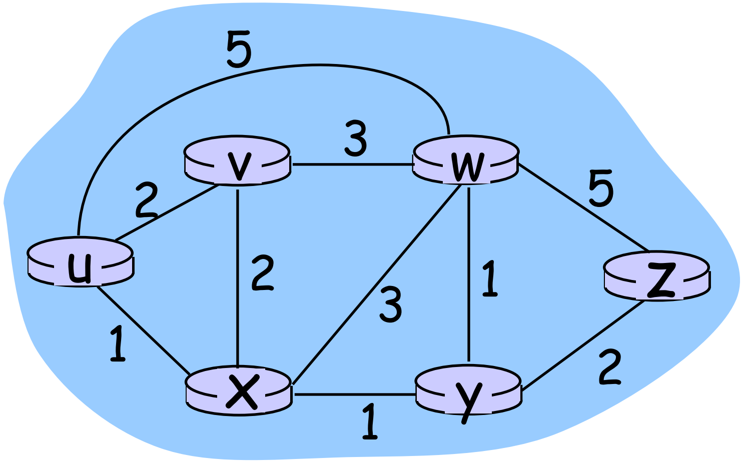
Dijsktra's Algorithm

```
Initialization:
  N' = {u}
  for all nodes v
    if v adjacent to u
        then D(v) = c(u,v)
    else D(v) = ∞

 Loop
   find w not in N' such that D(w) is a minimum
   add w to N'
   update D(v) for all v adjacent to w and not in N' :
      D(v) = min( D(v), D(w) + c(w,v) )
   /* new cost to v is either old cost to v or known
    shortest path cost to w plus cost from w to v */
 until all nodes in N'
```

Dijkstra's algorithm: example

Step	N'	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(z)
0	u	2,u	5,u	1,u	∞	∞
1	ux	2,u	4,x		2,x	∞
2	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					



Distance Vector Routing Algorithm

iterative:

- continues until no nodes exchange info.
- *self-terminating*: no "signal" to stop

asynchronous:

- nodes need *not* exchange info/iterate in lock step!

distributed:

- each node communicates *only* with directly-attached neighbors

Distance Table data structure

- each node has its own
- row for each possible destination
- column for each directly-attached neighbor to node
- example: in node X, for dest. Y via neighbor Z:

$$D^X(Y,Z) = \text{distance } \textit{from } X \textit{ to } Y, \textit{ via } Z \text{ as next hop}$$

$$= c(X,Z) + \min_w \{D^Z(Y,w)\}$$

Distance Vector Routing: overview

Iterative, asynchronous:
each local iteration caused by:

- local link cost change
- message from neighbor: its least cost path change from neighbor

Distributed:

- each node notifies neighbors *only* when its least cost path to any destination changes
  - neighbors then notify their neighbors if necessary

Each node:

wait for (change in local link cost of msg from neighbor)

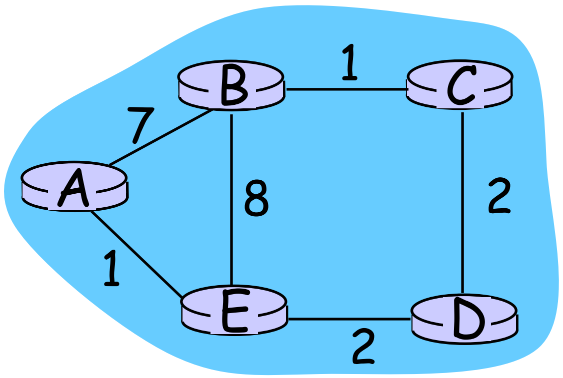
↓

recompute distance table

↓

if least cost path to any dest has changed, *notify* neighbors

Distance Table: example



$$D^E(C,D) = c(E,D) + \min_w\{D^D(C,w)\}$$
$$= 2+2 = 4$$

$$D^E(A,D) = c(E,D) + \min_w\{D^D(A,w)\}$$
$$= 2+3 = 5$$ loop!

$$D^E(A,B) = c(E,B) + \min_w\{D^B(A,w)\}$$
$$= 8+6 = 14$$

cost to destination via

$D^E()$	A	B	D
A	1	14	5
B	7	8	5
C	6	9	4
D	4	11	2

destination

Distance table gives routing table

cost to destination via

$D^E()$	A	B	D
A	1	14	5
B	7	8	5
C	6	9	4
D	4	11	2

destination

	Outgoing link to use, cost
A	A,1
B	D,5
C	D,4
D	D,2

destination

Distance table → Routing table

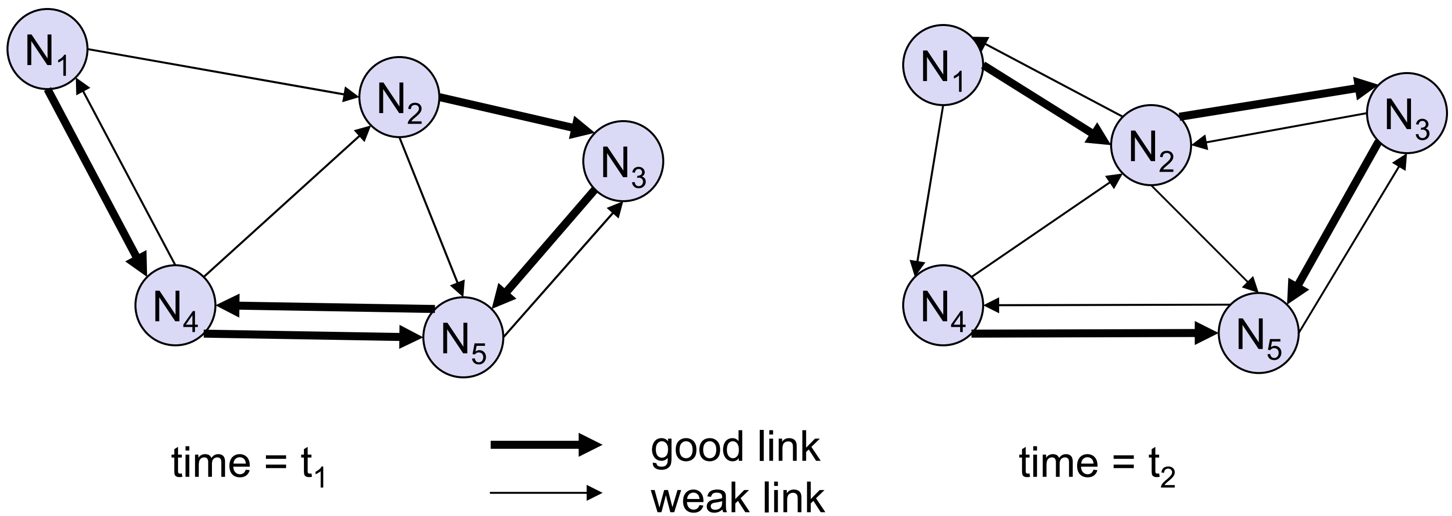
MANETs vs. Infrastructure Networks

- Infrastructure-based wireless network …
  - Access points or base stations define cells or service areas
  - Routing is relatively simple -- there is just a single hop from the access point to the wireless node
- Ad hoc wireless network …
  - There is no pre-defined or static network structure imposed by infrastructure
  - Wireless nodes are not necessarily all adjacent, so a node may need to forward data for other nodes (i.e., to participate in routing)

MANETs vs. Wireline Networks

- Wireline networks …
  - Symmetric links, usually with respect to both capacity and quality
  - Limited planned redundancy for reliability and load sharing
  - Planned links, typically of uniformly high quality, in a fixed topology
- Ad hoc wireless networks …
  - Asymmetric links
  - High degree of random redundancy in connectivity between wireless nodes
  - Unplanned, dynamic links with quality that may vary greatly due to interference, signal, etc.

Routing examples for an ad-hoc network



Problems of traditional routing algorithms

- Dynamic of the topology
  - frequent changes of connections, connection quality, participants
- Limited performance of mobile systems
  - periodic updates of routing tables need energy without contributing to the transmission of user data, sleep modes difficult to realize
  - limited bandwidth of the system is reduced even more due to the exchange of routing information
  - links can be asymmetric, i.e., they can have a direction dependent transmission quality
- Problem
  - protocols have been designed for fixed networks with infrequent changes and typically assume symmetric links

MANET versus Traditional Routing (1)

- Traditional routing algorithms are likely to be …
  - Inefficient due to slow convergence times (e.g., using distance vector algorithms as in RIP)
  - Non-functional due to large amounts of data or inability to deal with asymmetric links (e.g., link state algorithms as in OSPF)
- MANET routing must rely on data link information, not just network layer updates
  - Link layer determines connectivity and quality of links

MANET versus Traditional Routing (2)

- Centralized approaches are too slow and not robust enough for MANET
- All (or almost all) nodes in a MANET may be routers
- Long-lived circuits cannot be used in MANETs
- Path length (hop count) may not be the best metric for routing in MANETs

Outline

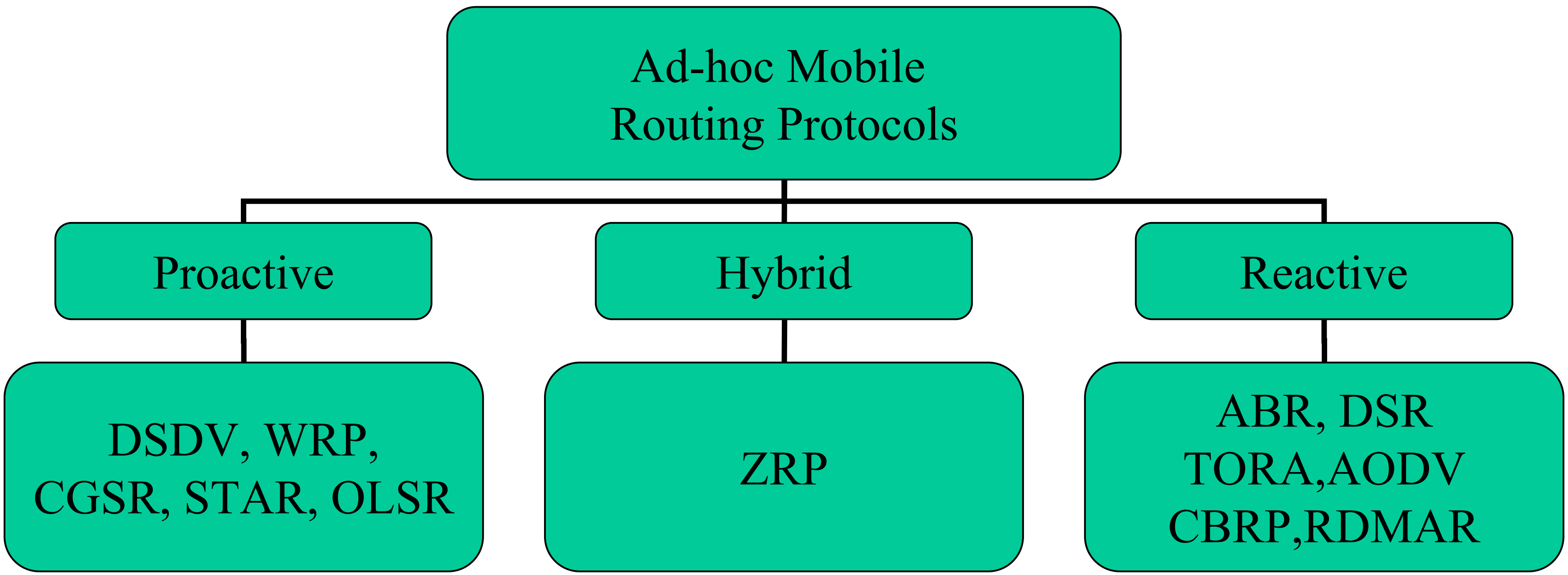
- Routing in the Internet
  - Distance Vector
- Why do we need routing protocols for Ad hoc networks?
- Unicast Routing in MANETS
  - Goals of routing protocols
  - Classification
    - Proactive or Table Driven
    - Reactive or On Demand
- Current approaches
  - DSDV (Distance Sequence Distance Vector)

Unicast Routing Protocols

Goals of a good routing protocol:

- Minimal control overhead
- Minimal processing overhead
- Multi-hop path routing capability
- Dynamic topology maintenance
- No loops
- Self-starting

Current Approaches



Primary Approaches (1)

- Proactive or Table Driven
  - Based on traditional distance-vector and link-state protocols
  - Each node maintains route to each other network edge
  - Periodic and/or event triggered routing update exchange
  - Higher overhead in most scenarios because of continuous updates
    - Mobility results in significant updates – some of the updates may never be used, if the ad hoc network moves faster than the route requests
  - Low latency of packet forwarding as the route is known (compared to reactive approach)
  - Longer route convergence time
  - Examples: DSDV, WRP, TBRPF, OLSR

Primary Approaches (2)

- Reactive (on Demand)
  - Source builds routing on demand by "flooding"
  - Maintain only active routes
  - Route Discovery cycle
  - Typically, less control overhead, better scaling properties
  - Drawback: route acquisition latency or long delay in finding the route
    - May not be suitable for real-time traffic
  - Example: AODV, DSR

Destination-Sequenced Distance-Vector (DSDV) Routing

Destination-Sequenced Distance-Vector (DSDV) Routing

- Proactive - based on Bellman – Ford
- Packets transmitted according to the routing table
- Each node maintains routing table with entry for each node in the network

  <dest_addr, dest_seqn#, next-hop, hop_count, install_time>
- Each node maintains its own sequence number
  - Updates at each change in neighborhood information
  - Used for freedom from loops
  - To distinguish stale routes from new ones

DSDV: Routing Update

- Each node periodically transmits updates to keep table consistency
  - Includes its own sequence number #, route table updates
    <dest_addr, dest_seq#, hop-count>
- Nodes also send routing table updates for important link changes (i.e. link breaks)
- When two routes to a destination received from two different neighbors
  - Choose the one with the greatest destination sequence number
  - If equal, choose the smallest hop-count

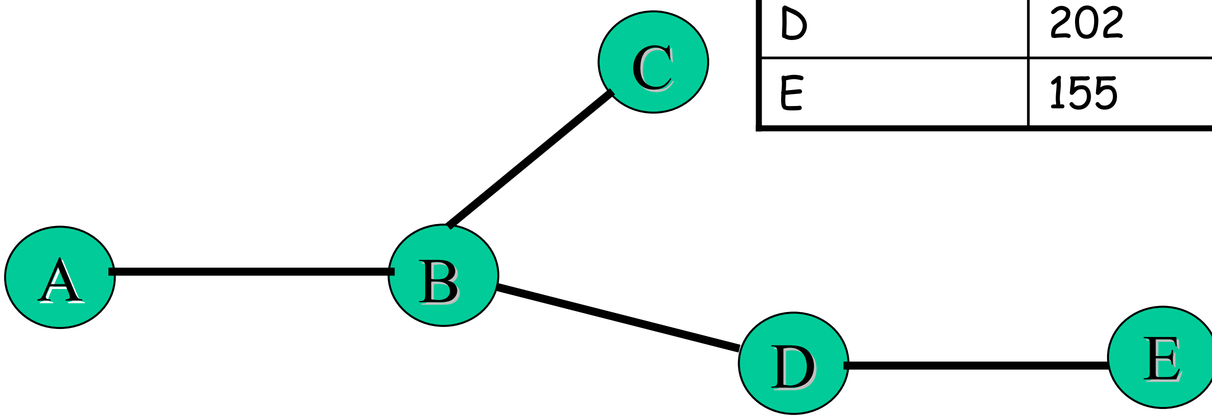
DSDV:Full Dump/Incremental Updates

- Routing table updates create lots of control traffic
- DSDV addresses this problem by using two types of routing update packets
  1. Full Dumps
     - Carry all routing table information (Several NPDUs)
     - Transmitted relatively infrequently
  2. Incremental Updates
     - Carry only information changed since last full dump
     - Fits within one network protocol data unit (NPDU)
     - When updates can no longer fit in one NPDU, send full dump

DSDV: Link Additions

B's Route Table:

Destination Address	Destination Seq #	Next_hop	Hop_count
B	132	B	0
C	144	C	1
D	202	D	1
E	155	D	2

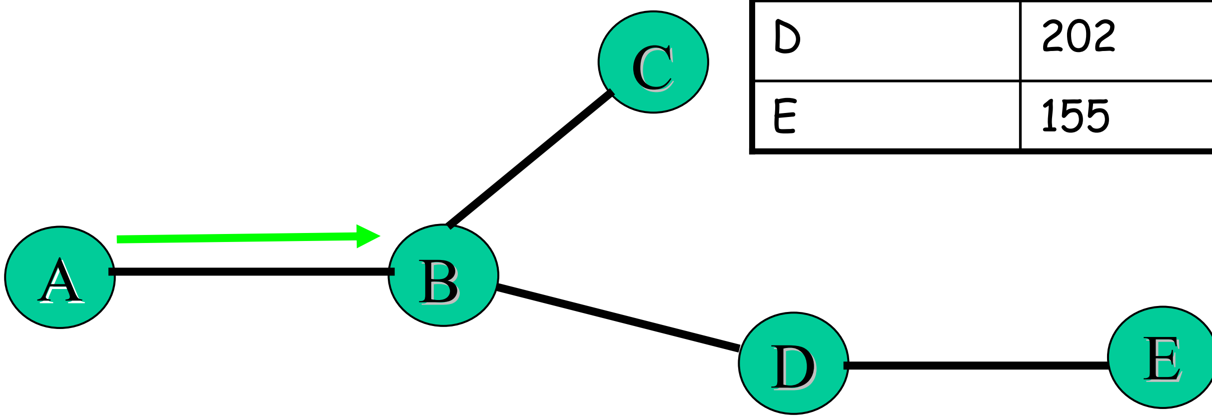


1. Node A joins network

DSDV: Link Additions

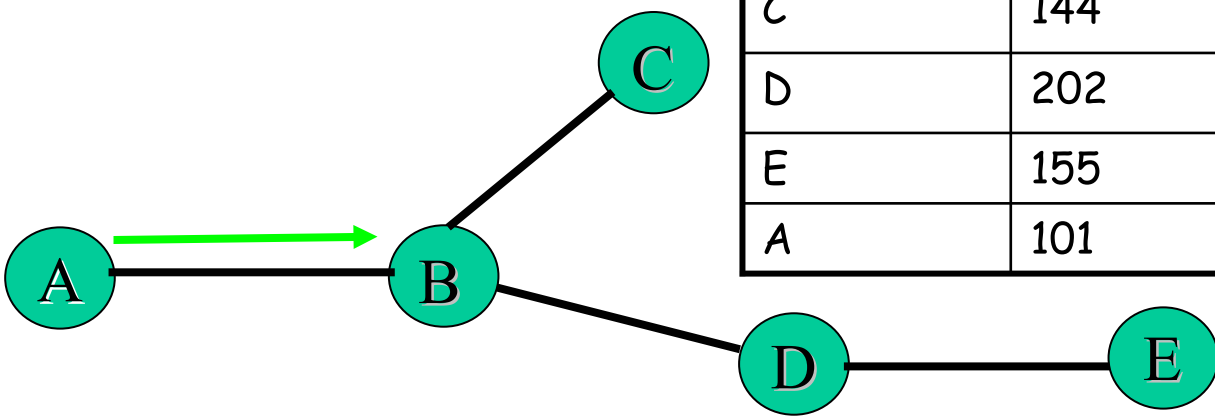
B's Route Table:

Destination Address	Destination Seq #	Next_hop	Hop_count
B	132	B	0
C	144	C	1
D	202	D	1
E	155	D	2



1. Node A joins network
2. Node A transmits routing table update
   <A, 101, 0>

DSDV: Link Additions

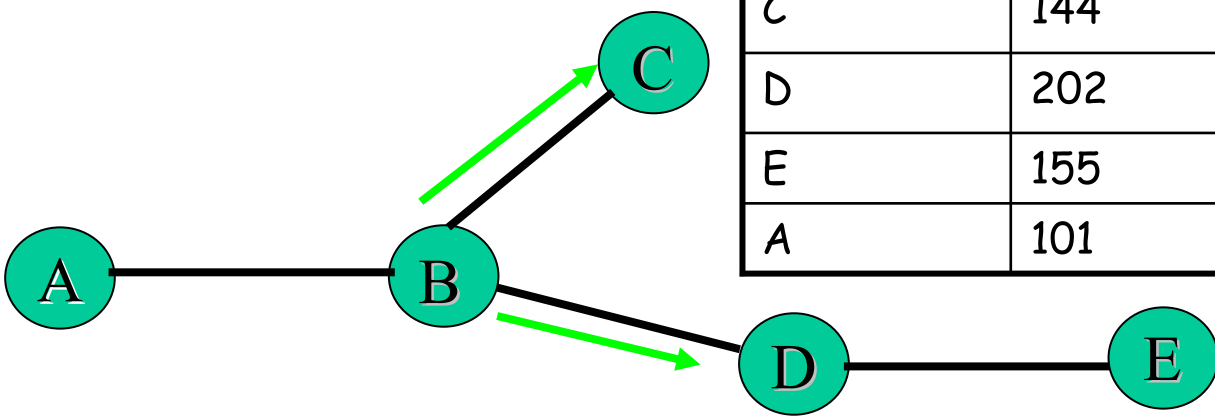


B's Route Table:

Destination Address	Destination Seq #	Next_hop	Hop_count
B	132	B	0
C	144	C	1
D	202	D	1
E	155	D	2
A	101	A	1

1. Node A joins network
2. Node A transmits routing table: <A, 101, 0>
3. Node B receives transmission, inserts <A, 101, 1> into route table

DSDV: Link Additions



B's Route Table:

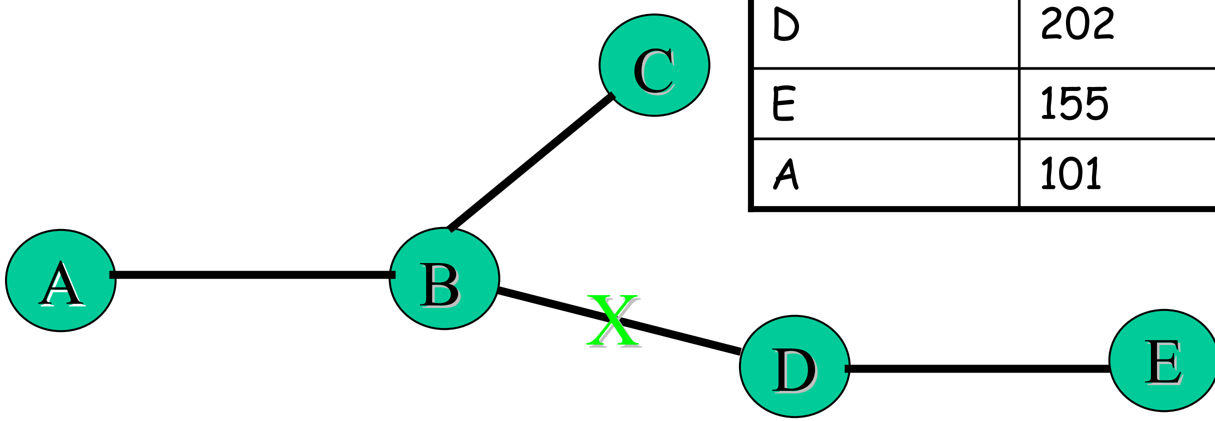
Destination Address	Destination Seq #	Next_hop	Hop_count
B	132	B	0
C	144	C	1
D	202	D	1
E	155	D	2
A	101	A	1

4. Node B propagates new route information to neighbors: <A, 101, 1>
5. Neighbors update their routing tables e.g. nodes C & D, <A, 101, B, 2>, and continue propagation information

DSDV: Link Breaks

B's Route Table:

Destination Address	Destination Seq #	Next_hop	Hop_count
B	132	B	0
C	144	C	1
D	202	D	1
E	155	D	2
A	101	A	1

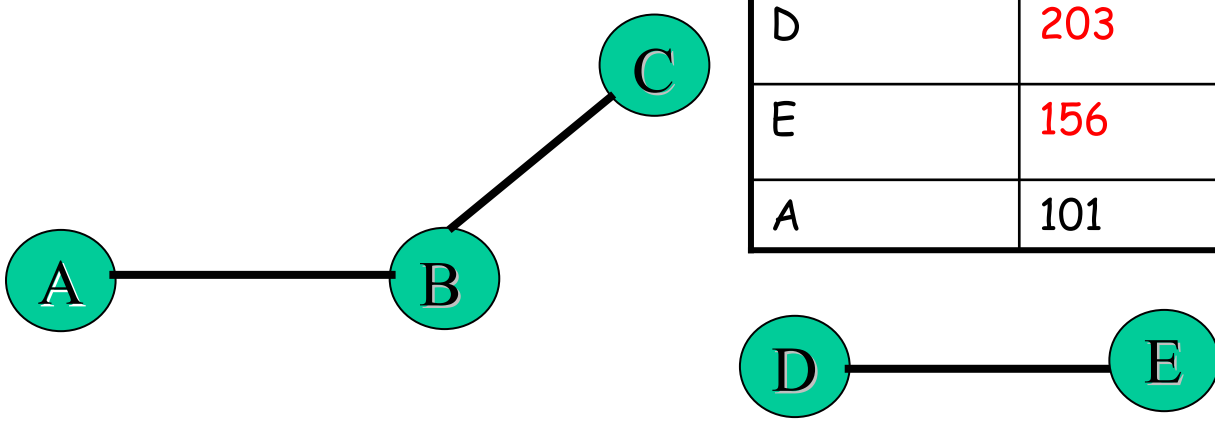


1. Link between B and D breaks

DSDV: Link Breaks

B's Route Table:

Destination Address	Destination Seq #	Next_hop	Hop_count
B	132	B	0
C	144	C	1
D	203	D	∞
E	156	D	∞
A	101	A	1

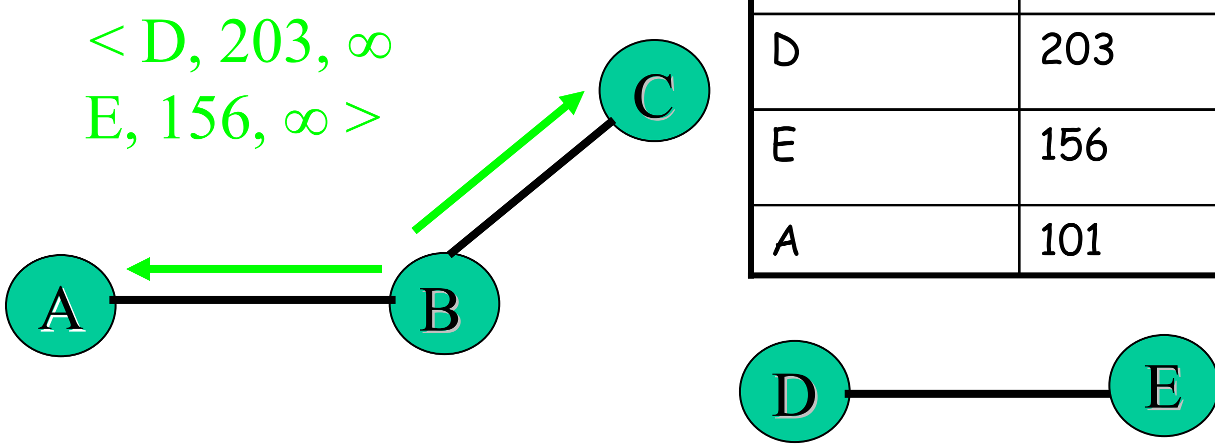


1. Link between B and D breaks
2. Node B notices break
    1. Updates hopcount for D & E to infinity
    2. Increments seq# for D & E

DSDV: Link Breaks

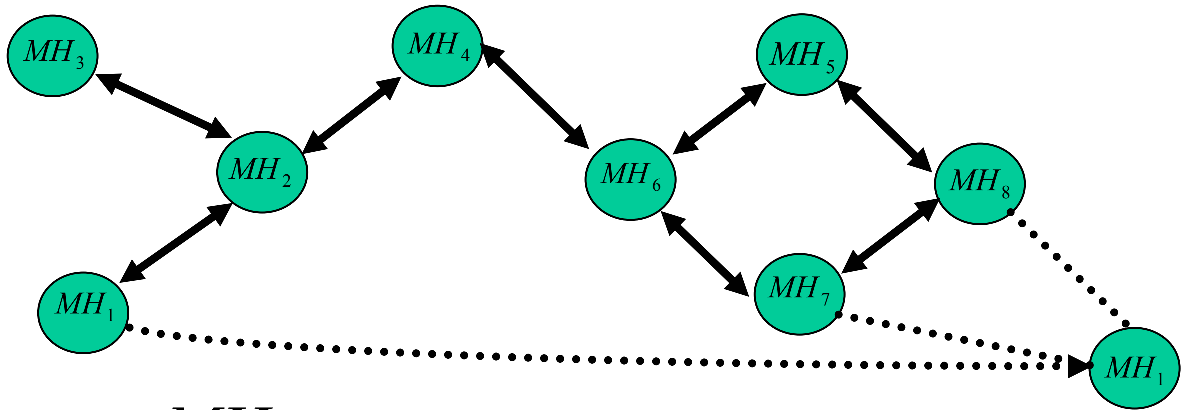
B's Route Table:

Destination Address	Destination Seq #	Next_hop	Hop_count
B	132	B	0
C	144	C	1
D	203	D	∞
E	156	D	∞
A	101	A	1



1. Link between B and D breaks
2. Node B notices break
   1. Updates hopcount for D & E to infinity
   2. Increments seq# for D & E
3. Node B sends update with new route information

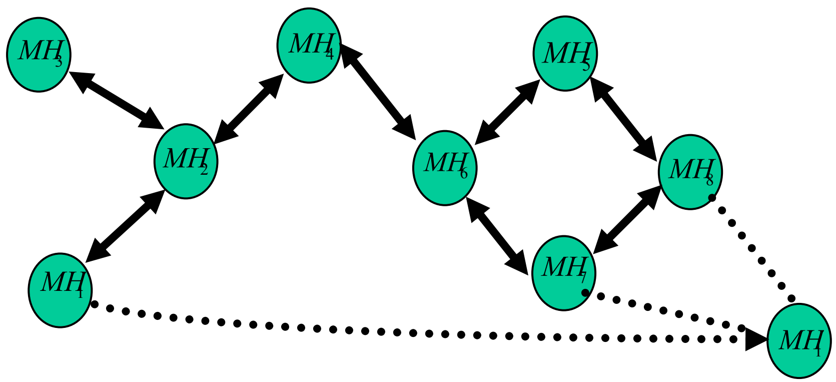
Movement in Ad Hoc Networks - Example



MH_4 Forwarding Table Before MH1 Moves Away

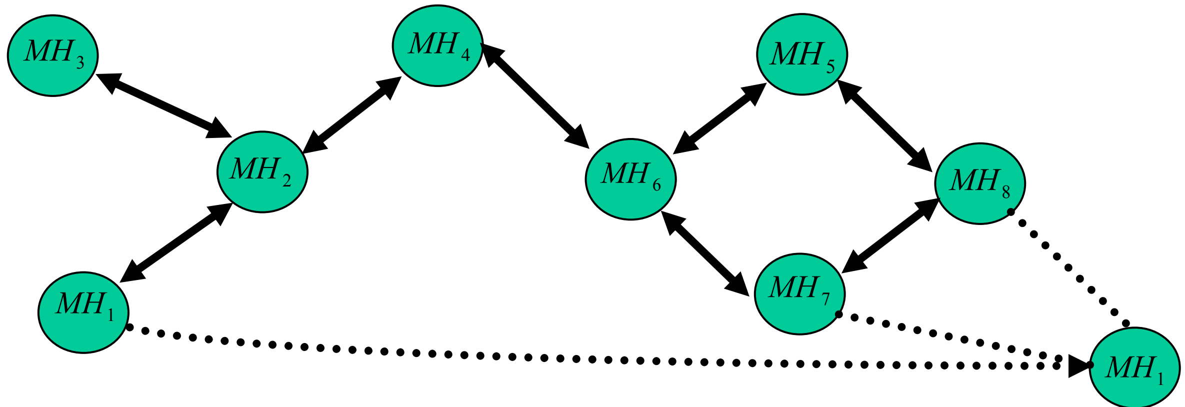
Destination	Next Hop	Metric	Sequence Number
MH_1	MH_2	2	S406_MH_1
MH_2	MH_2	1	S128_MH_2
MH_3	MH_2	2	S564_MH_3
MH_4	MH_4	0	S710_MH_4
MH_5	MH_6	2	S392_MH_5
MH_6	MH_6	1	S076_MH_6
MH_7	MH_6	2	S128_MH_7
MH_8	MH_6	3	S050_MH_8

Movements in Ad Hoc Networks - MH4 Advertized Route Table



Destination	Metric	Sequence Number
MH1	2	S406_MH1
MH2	1	S128_MH2
MH3	2	S564_MH3
MH4	0	S710_MH4
MH5	2	S392_MH5
MH6	1	S076_MH6
MH7	2	S128_MH7
MH8	3	S050_MH8

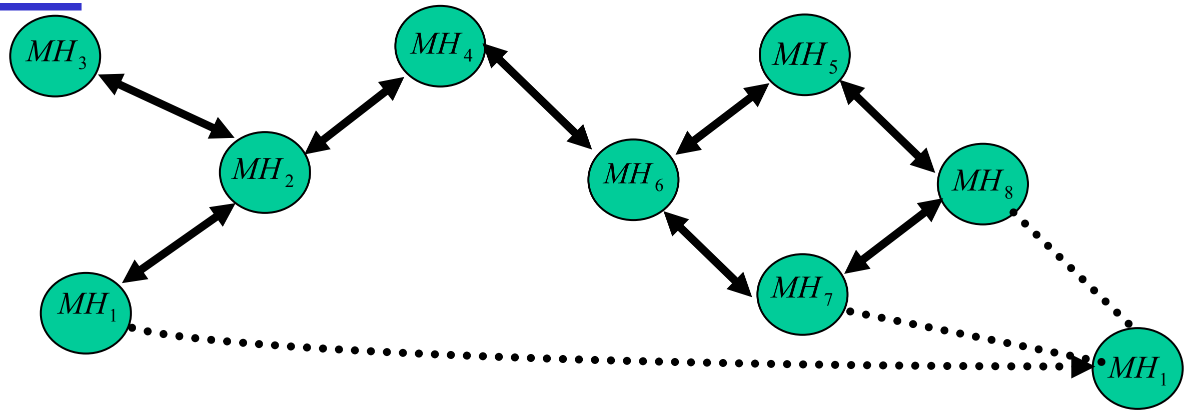
Movement in Ad Hoc Networks - MH1 Moves



MH_4 Forwarding Table (Updated)

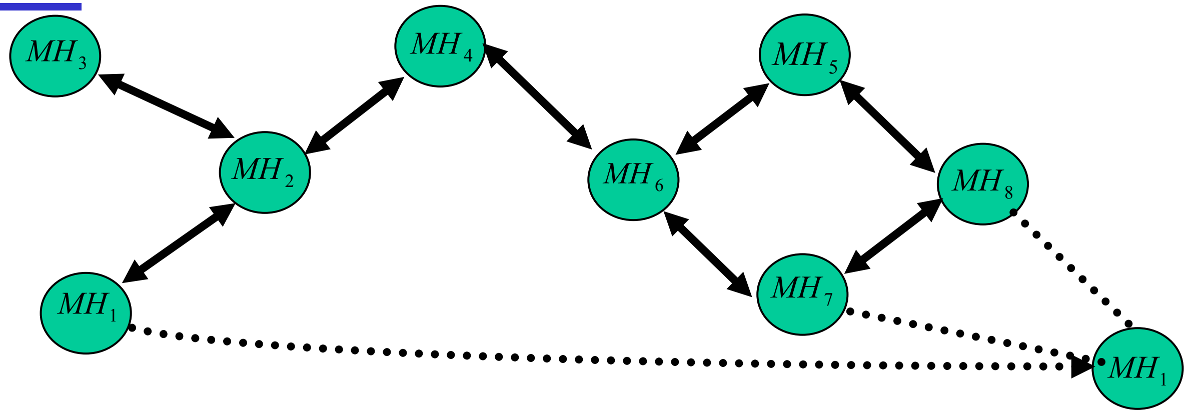
Destination	Next Hop	Metric	Sequence Number	Install	Stable_Data
MH_1	MH_6	3	S516_MH1	T810_MH_4	Ptr1_MH_1
MH_2	MH_2	1	S238_MH_2	T001_MH_4	Ptr1_MH_2
MH_3	MH_2	2	S674_MH_3	T001_MH_4	Ptr1_MH_3
MH_4	MH_4	0	S820_MH_4	T001_MH_4	Ptr1_MH_4
MH_5	MH_6	2	S502_MH_5	T002_MH_4	Ptr1_MH_5
MH_6	MH_6	1	S186_MH_6	T001_MH_4	Ptr1_MH_6
MH_7	MH_6	2	S238_MH_7	T002_MH_4	Ptr1_MH_7
MH_8	MH_6	3	S160_MH_8	T002_MH_4	Ptr1_MH_8

Movement in Ad Hoc Networks - MH1 Moves



- MH1 moves away from MH2 and is in the vicinity of MH8 and MH7
- The updated forwarding Table at MH4 are shown in the previous slide
- Only the entry for MH1 shows the new metric, but in the intervening time many new sequence number entries have been received. (Sequence number changed for each destination).

Movement in Ad Hoc Networks - MH1 Moves



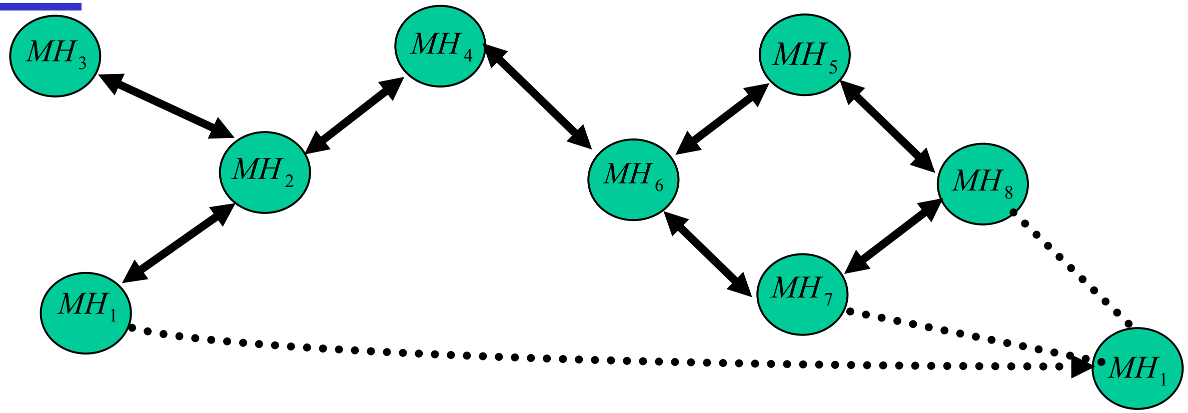
- The first entry is advertised in subsequent incremental routing information updates until the next full dump

- When MH1 moves in the vicinity of MH8 and MH7 away from MH2, it triggers an immediate incremental routing information updates

- MH7 broadcasts this information to MH6

- MH6 determines that it received new routing information regarding MH1, and it triggers an immediate update

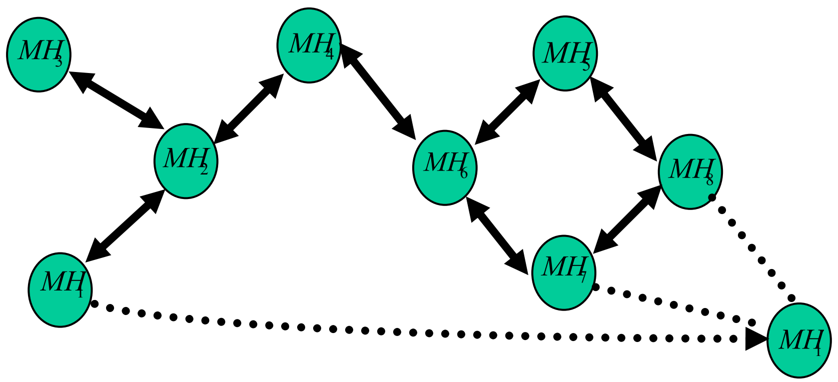
Movement in Ad Hoc Networks - MH1 Moves



- MH6's update carries the new routing information for MH1

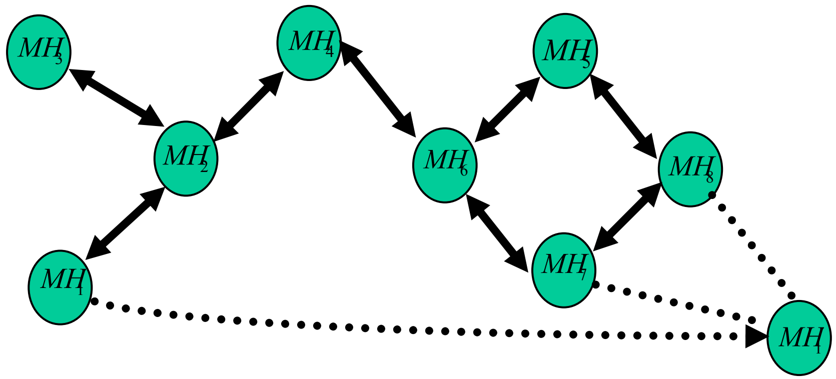
- MH4 receives update and it starts to broadcast it at every interval until the next full routing information dump

Movements in Ad Hoc Networks - MH4 Advertized Route Table (Updated)



Destination	Metric	Sequence Number
MH4	0	S820_MH4
MH1	3	S516_MH1
MH2	1	S238_MH2
MH3	2	S674_MH3
MH5	2	S502_MH5
MH6	1	S186_MH6
MH7	2	S238_MH7
MH8	3	S160_MH8

Comments on MH4 Updated Advertised Route Table



- In the advertisement the information regarding MH4 comes first since it is doing the advertisement
- The information regarding MH1 comes next because it's the only one that has significant route change affecting it
- As a general rule -> Routes with changed metrics are first included in each incremental packet
- The remaining packet space is filled by those routes whose sequence numbers are changed

General Comments

- In this example one node changed its location, so it has to change its routing information
- All nodes have recently transmitted new sequence numbers
- If there are too many updated sequence numbers to fit in a single packet – only the ones that fit would be transmitted, selected fairly for transmission according to their turn
- Others will go in subsequent incremental updates
- No required format for the full routing information packets – number of packets as needed
- All available information is transmitted
- Frequency of transmitting full updates is reduced if the volume of data begins to consume significant bandwidth

DSDV: Summary

- Proactive
- Routes maintained through periodic and event triggered routing table exchanges
- Incremental dumps and settling time used to reduce control overhead

Proactive protocols tend to perform best in networks with low to moderate mobility, fewer nodes, and many data sessions

Proactive Routing Approaches - TBRPF

- Topology Broadcast based on Reverse Path Forwarding Protocol
  - Considers broadcasting topology information (including link costs and up/down status) to all MHs
  - Each link-state update is sent on every link of the network through flooding
  - Communication cost of broadcasting topology can be reduced if updates are sent along spanning trees

Proactive Routing Approaches - TBRPF

- Topology Broadcast based on Reverse Path Forwarding Protocol
  - Messages are broadcast in the reverse direction along the directed spanning tree formed by the shortest paths from all nodes to source
  - Messages generated by a given source are broadcast in the reverse direction along the directed spanning tree formed by the shortest paths from all MHs (nodes) to the source

Proactive Routing Approaches

- The Optimized Link State Routing Protocol (OLSR)
  - Based on the link state algorithm
  - All links with neighboring MHs are declared and are flooded in the entire network
  - Minimizes flooding of this control traffic by using only the selected MHs, called multipoint relays
  - Only normal periodic control messages sent
  - Beneficial for the traffic patterns with a large subset of MHs are communicating with each other
  - Good for large and dense networks
  - An in-order delivery of its messages is not needed as each control message contains a sequence number

Proactive Routing Approaches

- Multipoint Relays
  - Minimize the flooding of broadcast packets in the network by reducing duplicate retransmissions in the same region
  - Each MH selects a set of neighboring MHs, to retransmit its packets and is called the multipoint relays (MPRs)
  - This set can change over time and is indicated by the selector nodes in their hello messages
  - Each node selects MPR among its one hop bi-directional link neighbors to all other nodes that are two hops away

Illustration of Multipoint Relays