Adhoc Wireless Networks

Routing Protocols

Dr. Ram P Rustagi
Dept of CSE
PES University
rprustagi@pes.edu

AHWN Routing Proto Characteristics

- Fully distributed
- Adaptive to frequent topology changes
- Route computation should involve min nodes
- Avoid global state maintenance
- Loop free and not stale routing
- Quick convergence
- Provide some level of QoS
- Optimal use of resources
  – BW, computing, battery, memory

Classification of Routing Protocol

- Classification categories
  - Routing information update mechanism
  - Use of temporal information for routing
  - Routing topology
  - Utilization of specific resources
- Above classification is not mutually exclusive

Routing Info Update Mechanism

- Proactive or table driven
  - Each node maintains topology in routing tables
  - Routing info is flooded in the whole n/w
  - To reach dst, run path finding algo in the table
- Reactive or on-demand routing
  - Node does not maintaint topology info
  - Node obtains path when required
  - No periodic exchange of routing info
- Hybrid routing (combination above two)
  - Within a zone, use table driven approach
  - Beyond the zone, use on deman routing

Use of Temporal Info for Routing

- Using past temporal info
  - Uses past status of links
    - Availability of wirless links
    - Topology change will break this
- Using future temporal info
  - Future status of wirless links
  - Future status of nodes
    - e.g. Battery status, discharge rate
      - Location prediction

Based on N/W Topology

- Internet topology is hierarchical
  - Reduces size of routing tables
- Adhoc networks are much smaller in size
  - Can use flat or hierarchical topology
- Flat topology routing protocols
  - Flat addressing scheme e.g. IEEE 802.3 LANs
    - Assumes globally unique addresses
- Hierarchical topology routing protocols
  - Logical hierarchy & associated addressing
  - Could be based on hops or geographic info

Utilization of Specific Resources

- Power aware routing
  - Minimize the battery consumption
  - Can consider local power consumption of global power consumption
- Geographical info assisted routing
  - Reduce control overhead using
    - Geographic info
  - Improve routing performance

AHWN Routing Protocols

- Table driven routing
  - **DSDV**, WRP
- On demand routing
  - DSR, AODV, TORA
- Hybrid Routing
  - CEDAR
- Others
  - Efficient flooding mechanisms based
  - Hierarchical routing
  - Power aware routing

Direct Sequence Distance Vector

- DSDV: the first protocol proposed for AHWN
- Based on distributed Bellman-Ford algo
  - Shortest path from this node to every node
- Node maintains info about all destinations
- Tables are exchanged periodically
- Sequence number based table updates achieve
  - Address Count to Infinity problem
  - Prevent loops
  - Faster convergence

Distance Vector Routing (Review)

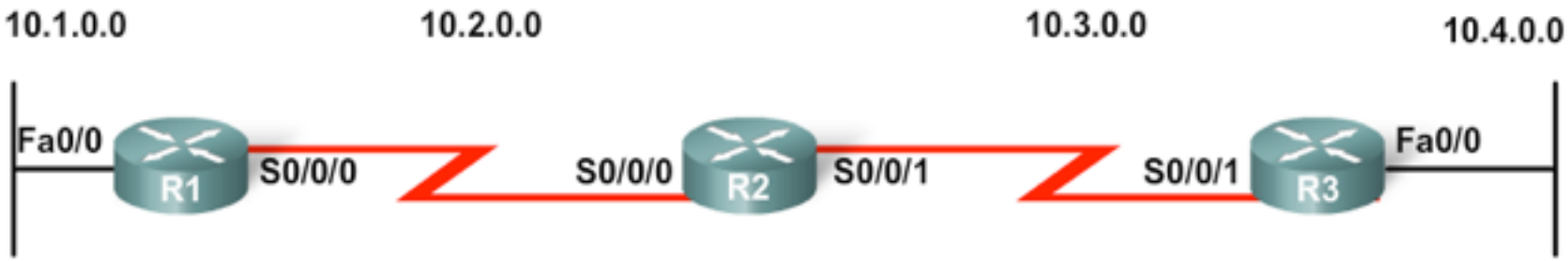
- Assumption
  - node knows cost of its connected link
  - A down (or non-existent) link cost is infinity
- Distributed
  - each node receives info from neighbors
  - computes routing and distributes to neighbors
  - no central computation
- Iterative
  - routing computation stops when no more info
  - it is self terminating - no external control
- Asynchronous
  - each node does not work in sync with others
- Based on Bellman-Ford equation

Distance Vector Routing (Review)

- Router doesn't know exact n/w topology
- Uses router as signposts along the path to dst^n
- Sends periodic updates
- Core of the DV protocol
  - Bellman Ford Algorithm
- Works best in following type of situations
  - network is simple and flat
    - does not require hierarchical design
  - administrators don't have enough knowledge
    - configure/troubleshoot LSP
  - Worst case **convergence** time is not a concern

DV Routing - Example

Initial Network



Network	Interface	Hop
10.1.0.0	Fa0/0	0
10.2.0.0	S0/0/0	0

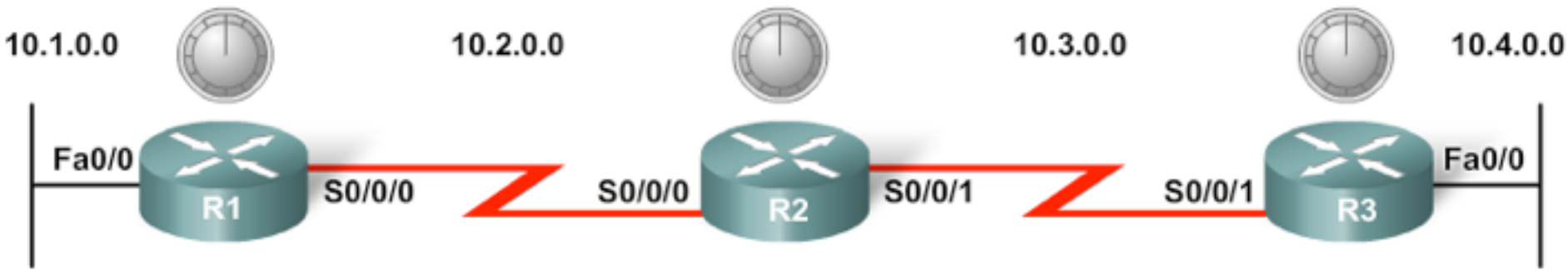
Network	Interface	Hop
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/1	0

Network	Interface	Hop
10.3.0.0	S0/0/0	0
10.4.0.0	Fa0/0	0

Src: CCNA Module 2

DV Routing - Example

- After the exchange of routing packets



Network	Interface	Hop
10.1.0.0	Fa0/0	0
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/0	1
10.4.0.0	S0/0/0	2

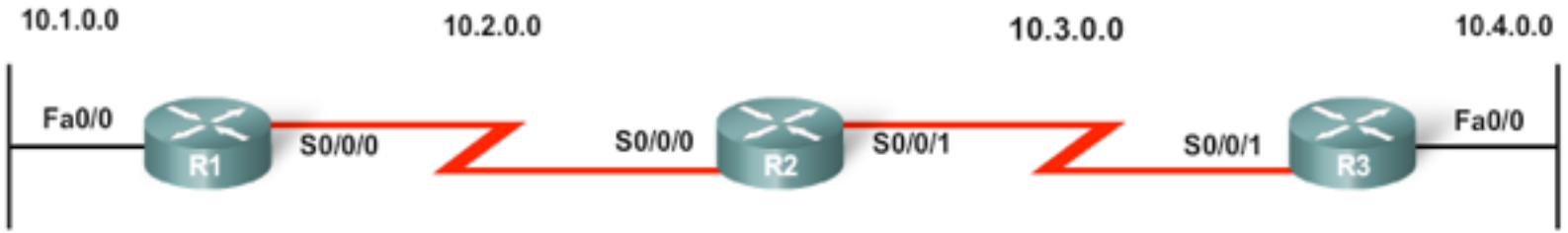
Network	Interface	Hop
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/1	0
10.1.0.0	S0/0/0	1
10.4.0.0	S0/0/1	1

Network	Interface	Hop
10.3.0.0	S0/0/1	0
10.4.0.0	Fa0/0	0
10.2.0.0	S0/0/1	1
10.1.0.0	S0/0/1	2

Src: CCNA Module 2

DV Routing - Loop

- src: CCNA Mod2/Chap4/4.4.1.1 loop animation
  - what happens when Fa0/0 of R3 goes down
  - R2 informs R3 that it can reach 10.4.0.0
  - R3 makes routing entry for 10.4.0.0 to send to R2



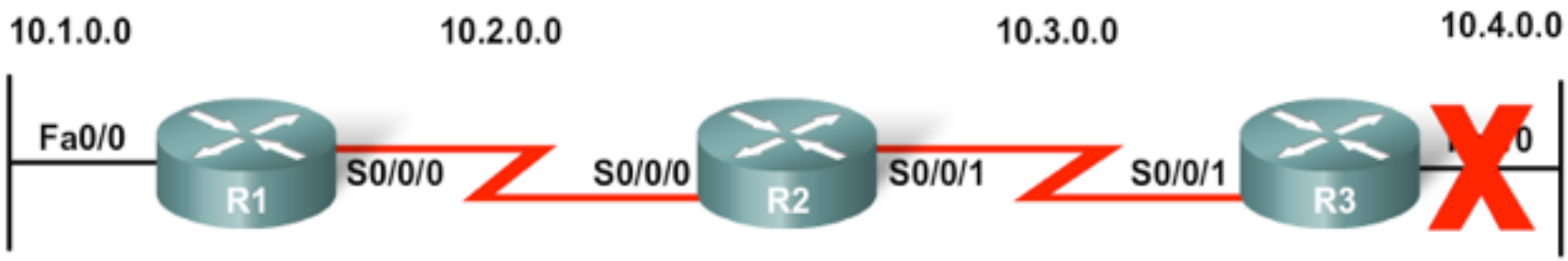
Network	Interface	Hop
10.1.0.0	Fa0/0	0
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/0	1
10.4.0.0	S0/0/0	2

Network	Interface	Hop
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/1	0
10.1.0.0	S0/0/0	1
10.4.0.0	S0/0/1	1

Network	Interface	Hop
10.3.0.0	S0/0/1	0
10.4.0.0	Fa0/0	0
10.2.0.0	S0/0/1	1
10.1.0.0	S0/0/1	2

DV Routing - Count to Infinity

- aec: CCNA Mod2/Chap4/4.4.2.1 count to infinity
  - what happens when Fa0/0 of R3 goes down
  - R2 informs R3 that it can reach 10.4.0.0
  - R3 makes routing entry for 10.4.0.0
  - R3 updates R2, R2 updated its hop count to 10.4.0.0



Network	Interface	Hop
10.1.0.0	Fa0/0	0
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/0	1
10.4.0.0	S0/0/0	2

Network	Interface	Hop
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/1	0
10.1.0.0	S0/0/0	1
10.4.0.0	S0/0/1	3

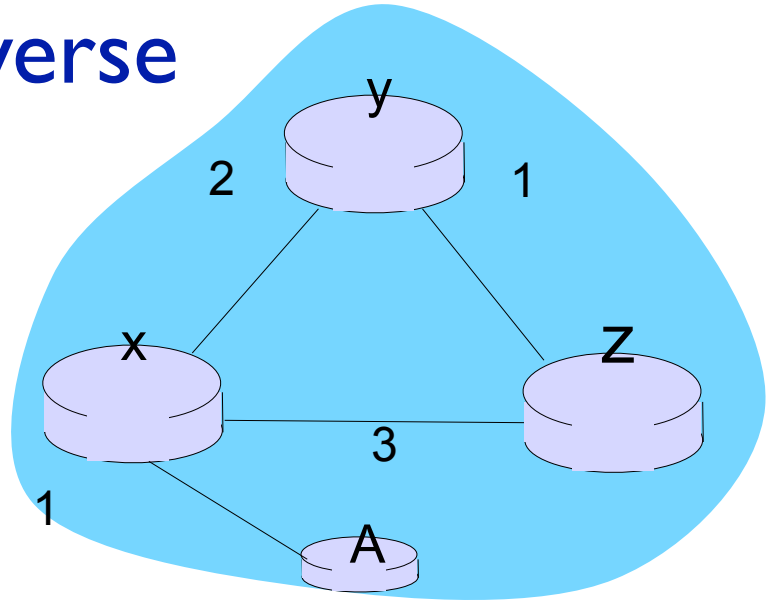
Network	Interface	Hop
10.3.0.0	S0/0/1	0
10.4.0.0	S0/0/1	2
10.2.0.0	S0/0/1	1
10.1.0.0	S0/0/1	2

DV Routing - Split Horizon

- Solution to count to infinity
  - define some fixed value as infinity
- Split horizon Rule
  - prevents routing loops?
  - **"A Router should not advertise a network through the interface from which the update came"**
  - Apply this rule to previous example

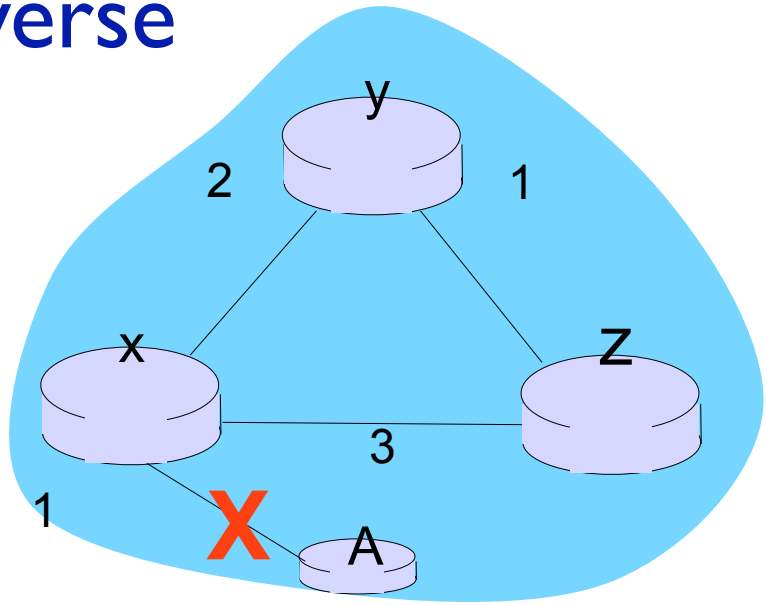
Count to Infinity with poison reverse

- Stable state
  - $d_x(A)=1$, $d_y(A)=3$, $d_z(A)=4$
- Link X to A fails
  - thus A is not reachable at all



Count to Infinity with poison reverse

- Stable state
  - $d_x(A)=1$, $d_y(A)=3$, $d_z(A)=4$
- Link X to A fails
  - Thus A is not reachable at all

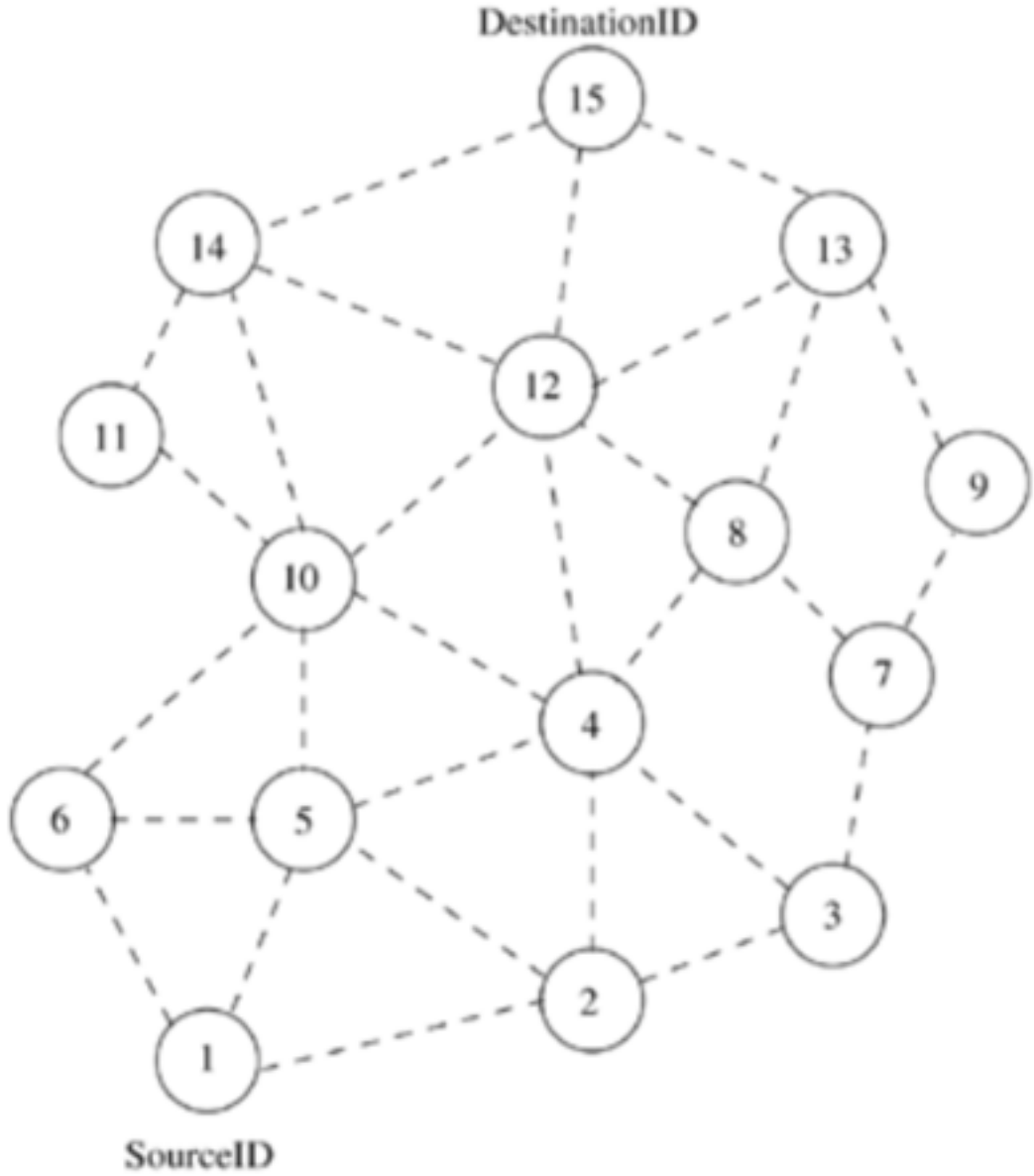


- How does DV work
  - Y learns from Z that it can reach A with cost 4
    - Y updates its cost to A as 5
  - Y advertises to X that it can reach with cost 5
  - X updates it table, says A is reachable with cost of 7
  - X advertises to Z that updated cost to A is 7
  - Z updates its cost to A 10 (7+3) from 4.
  - The process goes on till infinity
- **Solution: Define infinity as some fixed value**

DSDV

- Table updates are of two types
  - Incremental updates
    - Local topo sess minor changes
  - Full dumps update
    - Topology changes significantly
    - Table updates require more than 1 NPDU
- Each update by destination uses new seq num
- One link break propagates across whole n/w
  - Neighbour node uses odd sequence
- Destination node itself used even sequence

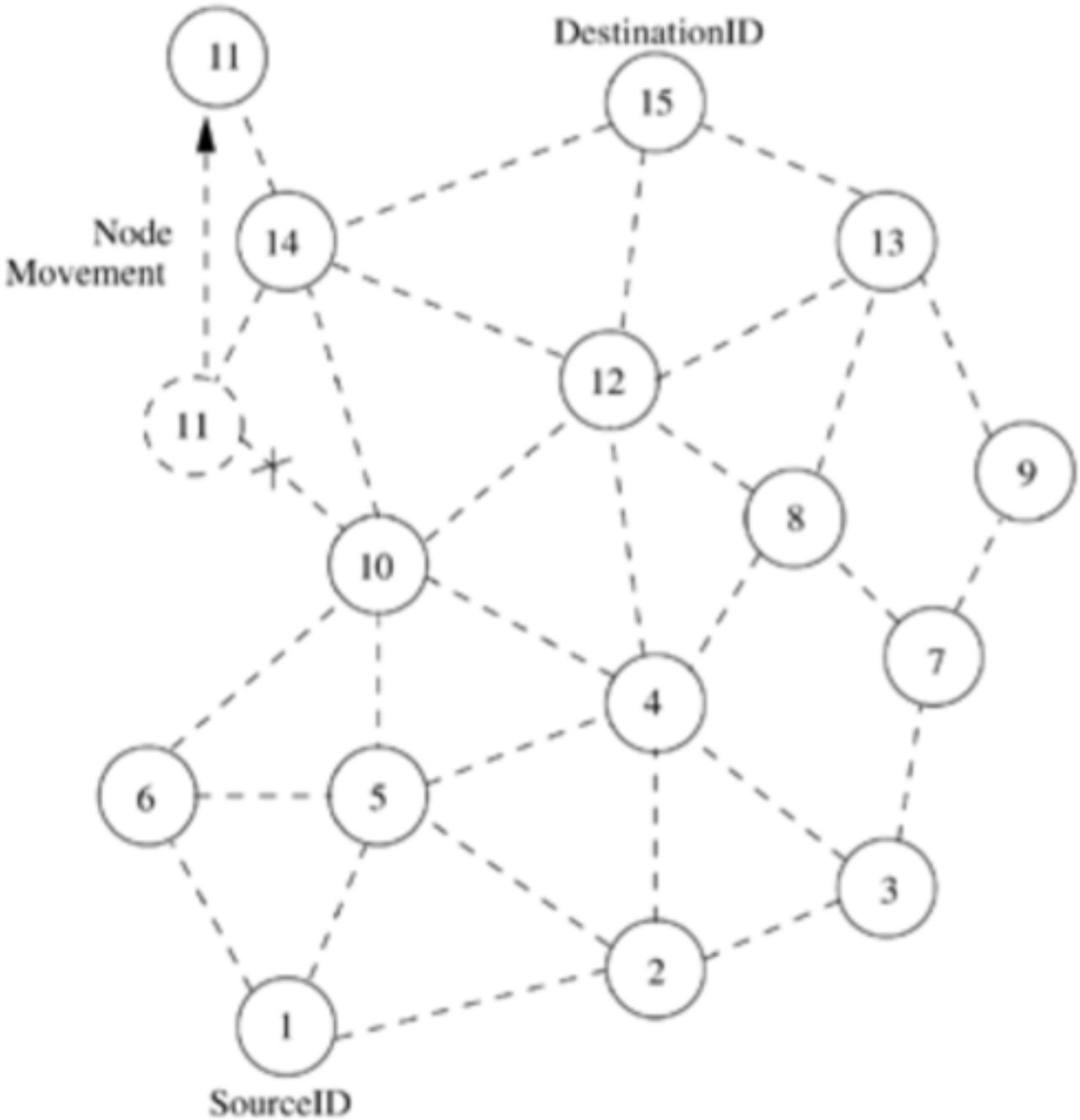
DSDV



(a) Topology graph of the network

Dest	NextNode	Dist	SeqNo
2	2	1	22
3	2	2	26
4	5	2	32
5	5	1	134
6	6	1	144
7	2	3	162
8	5	3	170
9	2	4	186
10	6	2	142
11	6	3	176
12	5	3	190
13	5	4	198
14	6	3	214
15	5	4	256

(b) Routing table for Node 1



Routing Table for Node 1

Dest	NextNode	Dist	SeqNo
2	2	1	22
3	2	2	26
4	5	2	32
5	5	1	134
6	6	1	144
7	2	3	162
8	5	3	170
9	2	4	186
10	6	2	142
11	5	4	180
12	5	3	190
13	5	4	198
14	6	3	214
15	5	4	256

DSDV

- Maintains the simplicity of Distance Vector
- Guarantee loop freeness
  – New entry for seq number in DV table
- Allow fast reaction to topology changes
  – Immediate advertisement on significant changes
  – Wait with advertiszem of unstable routes
- Seq num originates from destination
  – Ensures loop freeness

DSDV

- Routing information in advertisement
  - Dstn addr, seq num, metric (hop count)
- Rules to set seq number information
  - On own advertisement, increase seq num by 2
    - Even seq num
  - If a node is not reachable, increase seq by 1
    - Set metric (hop count) to ∞
- Routing table update
  - Select route with higher dstn seq num
    - Ensures newest info from dstn
  - Same seq num, selcet better metric route

DSDV

Dst	Nxt	Hop	Seq
A	A	0	300
B	B	1	198
C	B	2	100

Dst	Nxt	Hop	Seq
A	A	1	300
B	B	0	198
C	B	1	100

Dst	Nxt	Hop	Seq
A	B	2	300
B	B	1	198
C	C	0	100

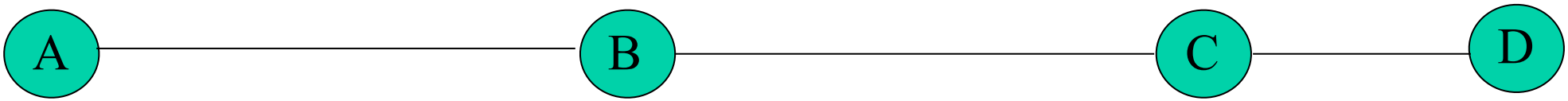
DSDV Route Advertisement

Dst	Nxt	Hop	Seq
A	A	0	300
B	B	1	200
C	B	2	100

Dst	Nxt	Hop	Seq
A	A	1	300
B	B	0	200
C	B	1	100

Dst	Nxt	Hop	Seq
A	B	2	300
B	B	1	200
C	C	0	100

DSDV New Node

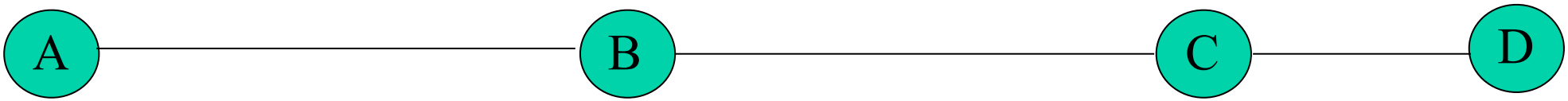


Dst	Nxt	Hop	Seq
A	A	0	300
B	B	1	200
C	B	2	100

Dst	Nxt	Hop	Seq
A	A	1	300
B	B	0	200
C	B	1	100

Dst	Nxt	Hop	Seq
A	B	2	300
B	B	1	200
C	C	0	100
D	D	1	0

DSDV New Node Update

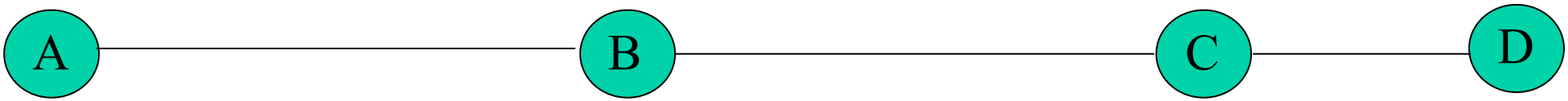


Dst	Nxt	Hop	Seq
A	A	0	300
B	B	1	200
C	B	2	100
D	B	3	0

Dst	Nxt	Hop	Seq
A	A	1	300
B	B	0	200
C	C	1	100
D	C	2	0

Dst	Nxt	Hop	Seq
A	B	2	300
B	B	1	200
C	C	0	100
D	D	1	0

DSDV After Few Updates

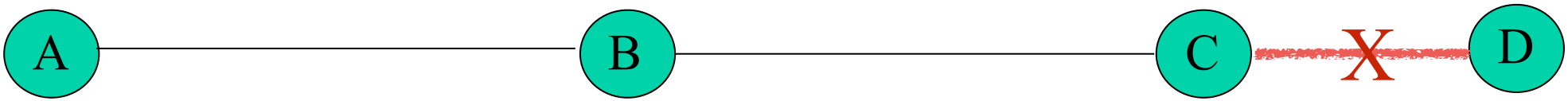


Dst	Nxt	Hop	Seq
A	A	0	400
B	B	1	300
C	B	2	200
D	B	3	100

Dst	Nxt	Hop	Seq
A	A	1	400
B	B	0	300
C	C	1	200
D	C	2	100

Dst	Nxt	Hop	Seq
A	B	2	400
B	B	1	300
C	C	0	200
D	D	1	100

DSDV Link Break

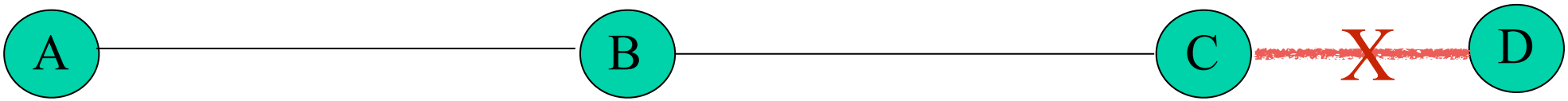


Dst	Nxt	Hop	Seq
A	A	0	400
B	B	1	300
C	B	2	200
D	B	3	100

Dst	Nxt	Hop	Seq
A	A	1	400
B	B	0	300
C	C	1	200
D	C	2	100

Dst	Nxt	Hop	Seq
A	B	2	400
B	B	1	300
C	C	0	200
D	D	∞	101

DSDV Immediate Advt

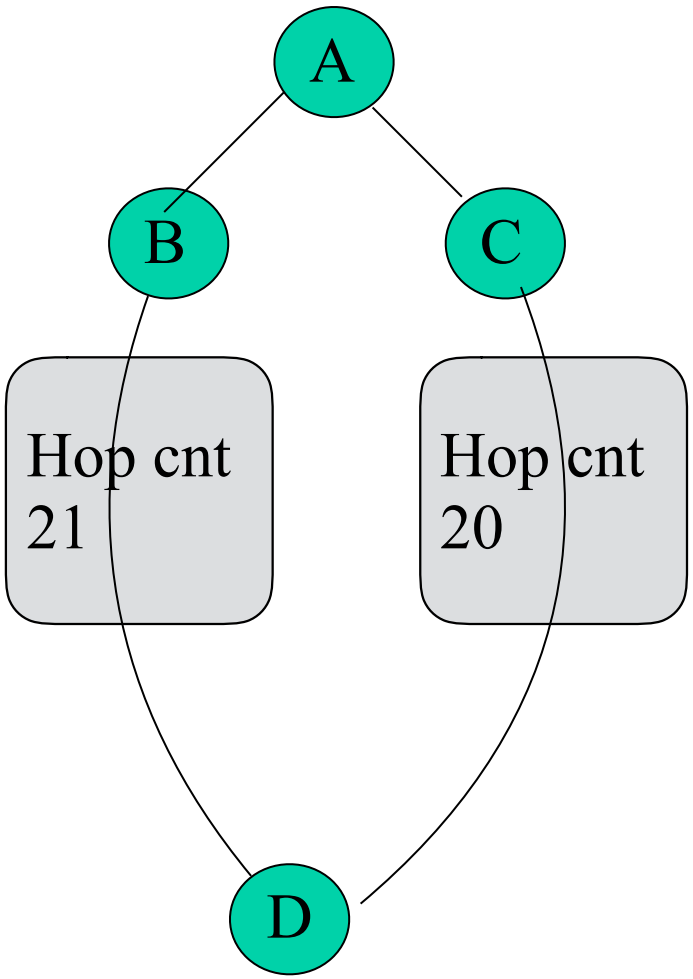


Dst	Nxt	Hop	Seq
A	A	0	400
B	B	1	300
C	B	2	200
D	B	∞	101

Dst	Nxt	Hop	Seq
A	A	1	400
B	B	0	300
C	C	1	200
D	C	∞	101

Dst	Nxt	Hop	Seq
A	B	2	400
B	B	1	300
C	C	0	200
D	D	∞	101

DSDV Fluctuations



- Entry for D in A
  - [D, C, 20, 100]
- D makes Bcast: seq 102
- A receives update from B
  - [D, B, 21, 102]
  - propagates this route
- A receives udpate from C
  - [D, C, 20, 102]
  - propages this route
- Leads to unnecessary route advertisements

DSDV Fluctuations

- How to damp fluctuations
  - Record settling time of every route
  - On first update with new seq num
    - wait for this settling time before advertising it
      - Propose wait time: 2* avg. settling time
  - Results in avoidance of unnecessary advertisement
    - Saving of bandwidth

DSDV Summary

- Advantages
  - Simple, like distance vector
  - Loop free (thru destination seq number)
  - Quick convergene (no latency in route discovery)
- Disadvantages
  - No sleeping nodes
  - When N1 does not communicate with N2
    - Routing table overhead of maintaining this info

AHWN Routing Protocols

- Table driven routing
  - DSDV, **WRP**
- On demand routing
  - DSR, AODV, TORA
- Hybrid Routing
  - CEDAR
- Others
  - Efficient flooding mechanisms based
  - Hierarchical routing
  - Power aware routing

Wireless Routing Protocol

- Similar to DSDV
  - Maintains up to date view of network
  - Each node has route to each dstn node
- Differences from DSDV in following
  - Employs unique method of maintaining info
    - Shortest distance to each dstn node
    - Predecessor (penultimate) node for dstn node
  - Uses multiple tables
    - Maintains more accurate info (than topology)
  - Table update process & maintenance
  - Checks for other neighbour's distances

Wireless Routing Protocol - Tables

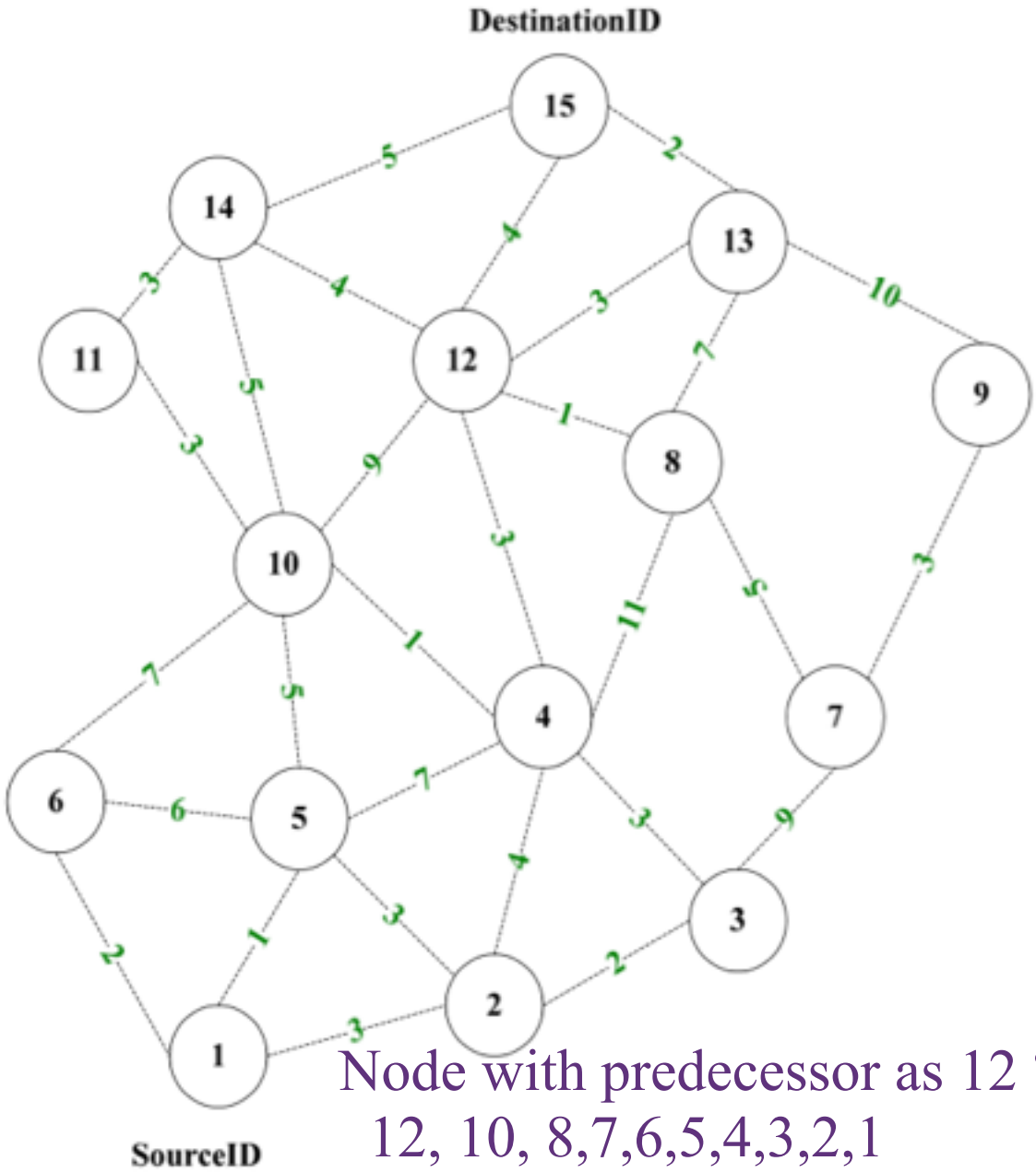
- Distance table (DT) of a node x contains
  - Distance of each dstn y via each neighbour z
  - Downstream neighbour of z via the path to y
- Routing table (RT) of a node x contains
  - Shortest distance to each dstn y,
  - Predecessor (penultimate) node to dstn y
  - Successor node of x on path to y
  - Path status (simple, invalid, loop)

Wireless Routing Protocol - Tables

- Link Cost Table (LCT) of a node x
  - Cost of link to each neighbour
  - Num of timeouts since last successful update
- Message Retransmission List (MRL)
  - Which neighbour node has not ack'ed update msg
  - Update msg to be retransmitted w/ a counter
    - Counter decremented after each retransmission

Wireless Routing Protocol - Tables

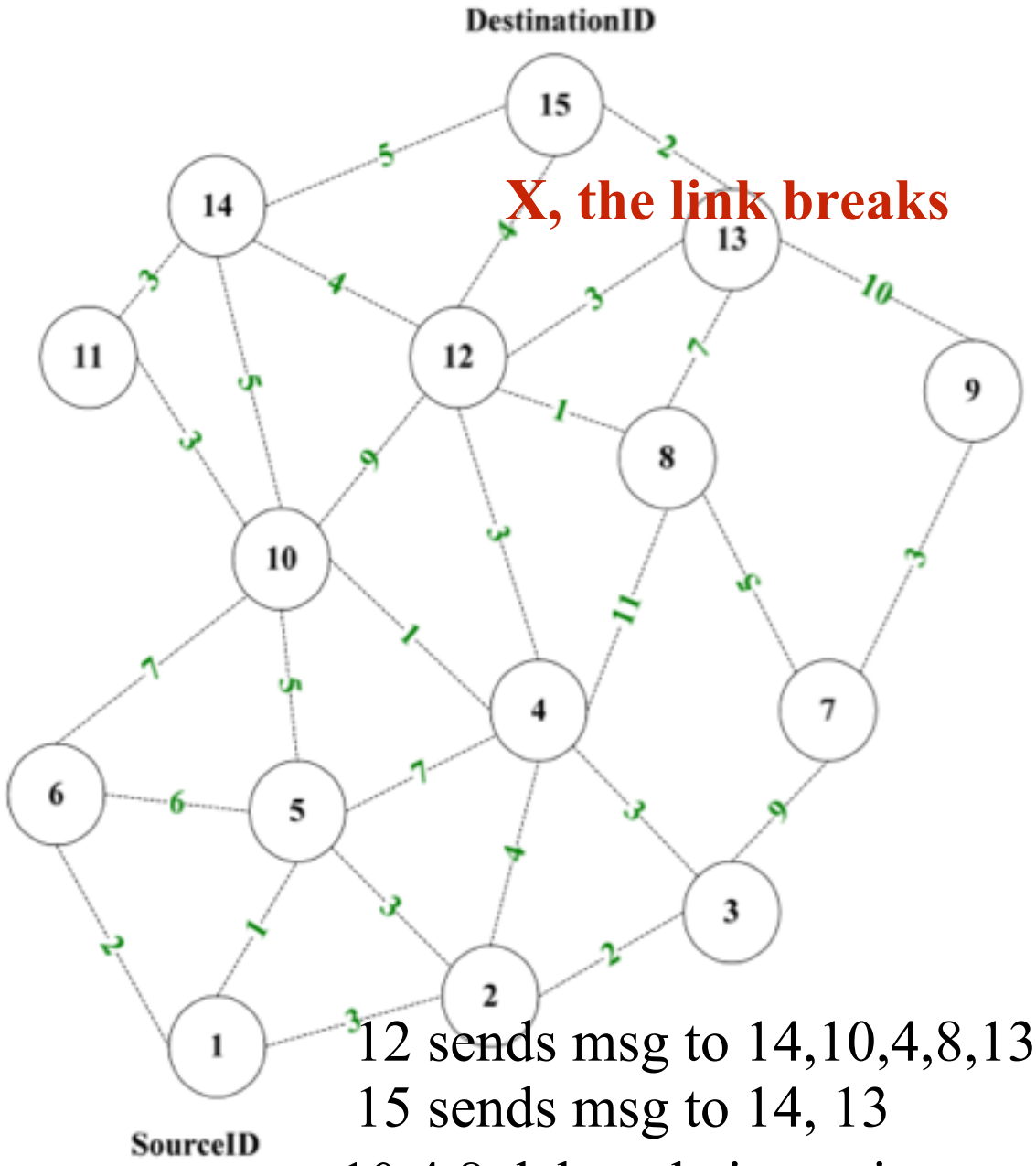
- Mechanism of exchanging info (update msg)
  - Exchange routing tables
  - Link changes info
  - Nodes on MRL need to ack rcpt of update msg
  - If no change in routing table,
    - Node sends Hello msg to ensure connectivity
  - On receiving update msg
    - Node modifies DT, and looks for better path
    - Any new path found is relayed back to orig node
      - Original node can update its tables
  - On receiving Ack, node updates MRL
  - Consistency check of neighbours on link change
    - Faster convergence



Routing Entry at Each Node for DestinationID 15

Node	NextNode	Pred	Cost
15	15	15	0
14	15	14	5
13	15	13	2
12	15	12	4
11	14	14	8
10	4	12	8
9	13	13	12
8	12	12	5
7	8	12	10
6	10	12	15
5	10	12	13
4	12	12	X 7
3	4	12	X 10
2	4	12	11
1	2	12	14

Node with predecessor as 12 ?
12, 10, 8,7,6,5,4,3,2,1



X, the link breaks

Routing Entry at Each Node for DestinationID 15

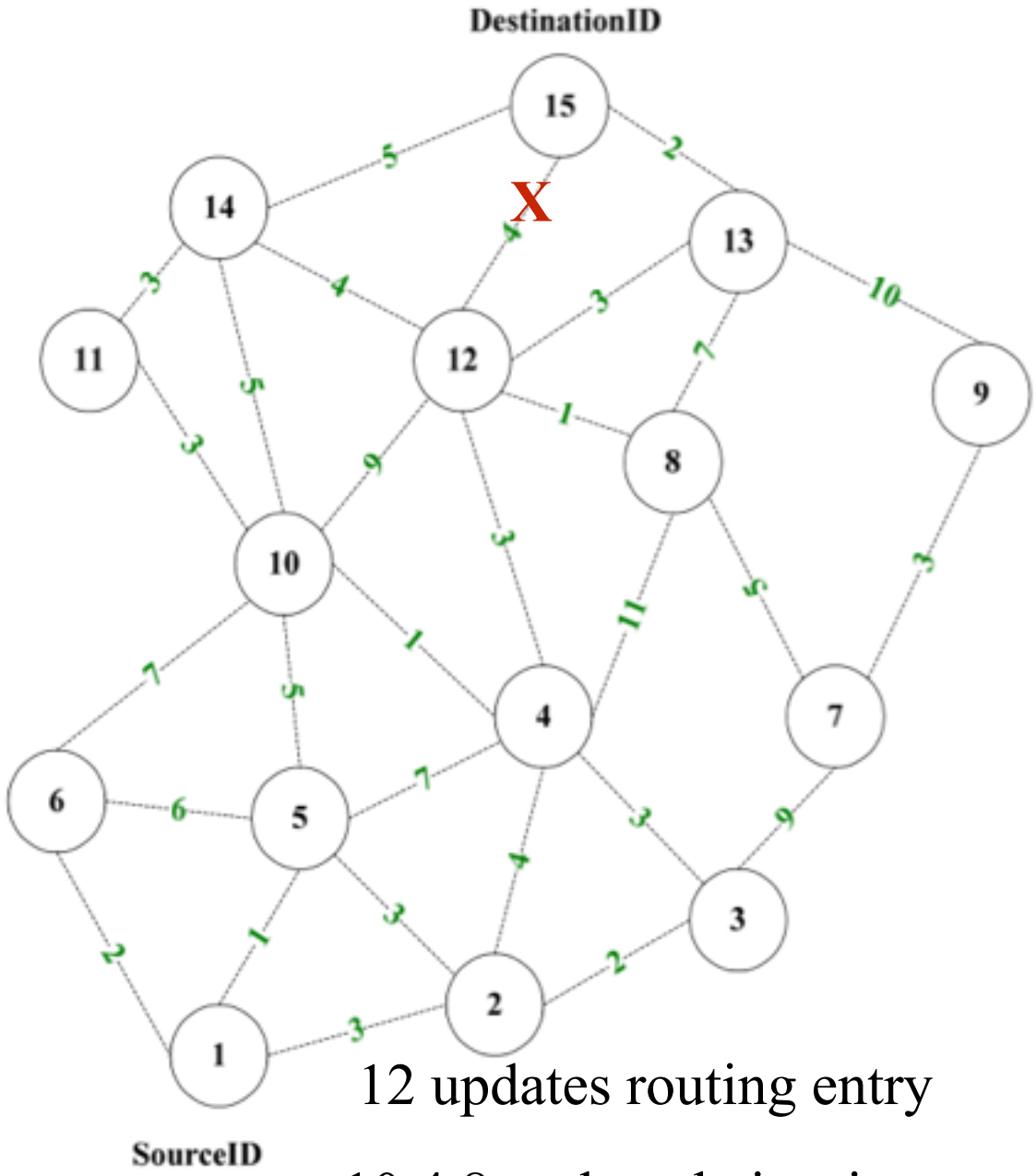
Node	NextNode	Pred	Cost
15	15	15	0
14	15	14	5
13	15	13	2
12	15	12	4
11	14	14	8
10	4	X	8
9	13	13	12
8	12	X	5
7	8	12	10
6	10	12	15
5	10	12	13
4	12	X	X 7
3	4	12	X 10
2	4	12	11
1	2	12	14

12 sends msg to 14,10,4,8,13

15 sends msg to 14, 13

10,4,8 delete their routing entries

AHWN



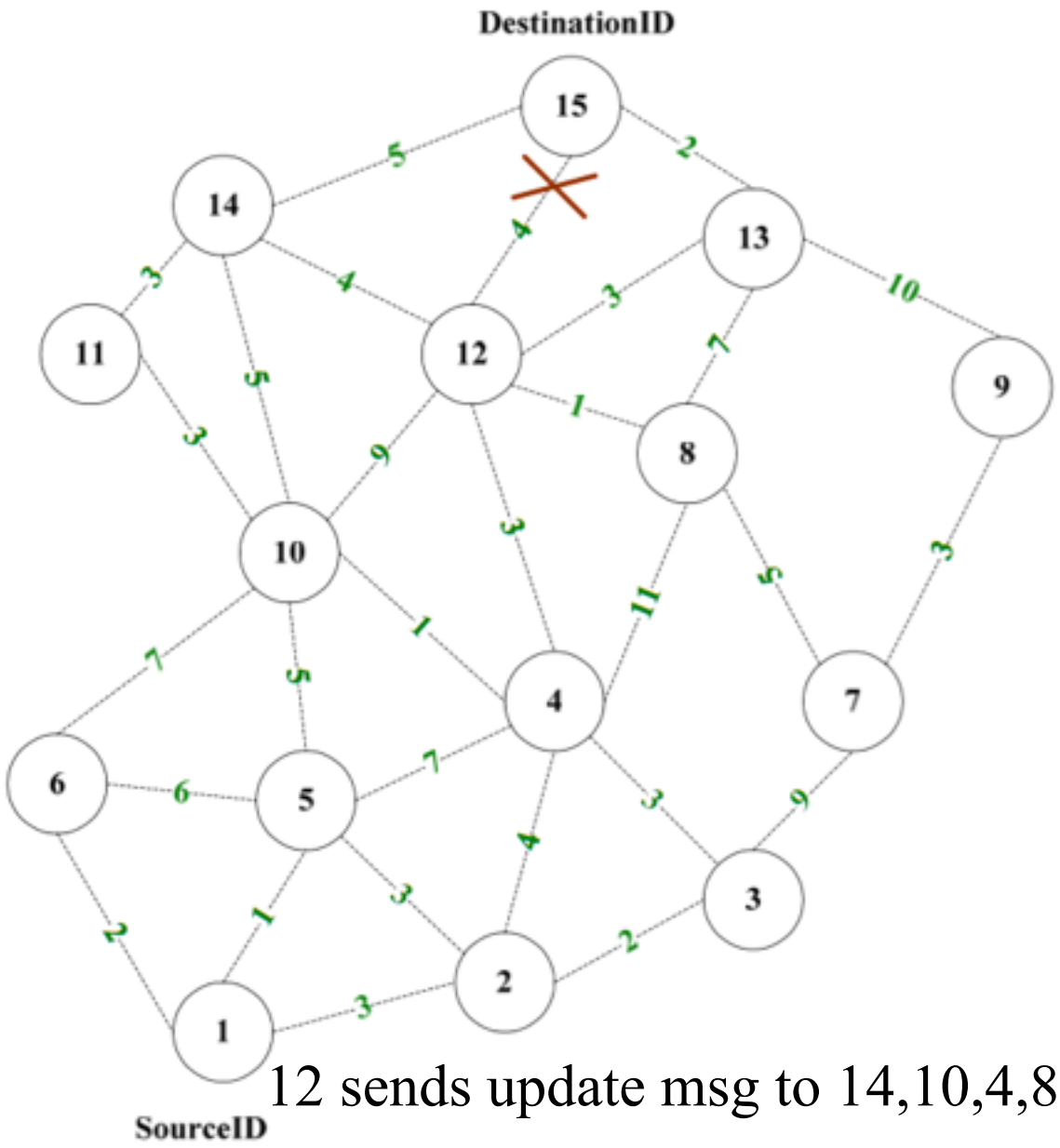
Routing Entry at Each Node for DestinationID 15

Node	NextNode	Pred	Cost
15	15	15	0
14	15	14	5
13	15	13	2
12	X 13	X 13	X 5
11	14	14	8
10	X 14	X 14	X 10
9	13	13	12
8	X 13	X 13	X 9
7	8	12	10
6	10	12	15
5	10	12	13
4	X 10	X 14	X 11
3	4	12	X 10
2	4	12	11
1	2	12	14

12 updates routing entry

10,4,8 update their min cost

AHWN



Routing Entry at Each Node for DestinationID 15

Node	NextNode	Pred	Cost
15	15	15	0
14	15	14	5
13	15	13	2
12	15	13	5
11	14	14	8
10	4	13	9
9	13	13	12
8	12	13	6
7	8	13	11
6	10	13	16
5	10	13	14
4	12	13	8
3	4	13	11
2	4	13	12
1	2	13	15

12 sends update msg to 14,10,4,8,13

10,4 update their routing entries

In turn they update their neighbours

AHWN

WRP: Advantages & Disadvantages

- Advantages:
  - Same as DSDV
  - Faster convergence than DSDV
  - Fewer table updates
- Disadvantages
  - Need large memory and more compute power
    - More number of tables.
  - At high mobility, control overhead for updating table entries same as DSDV
    - Not suitable for highly dynamic topology, AHWN

Assignment -01

- Adhoc Wifi Network