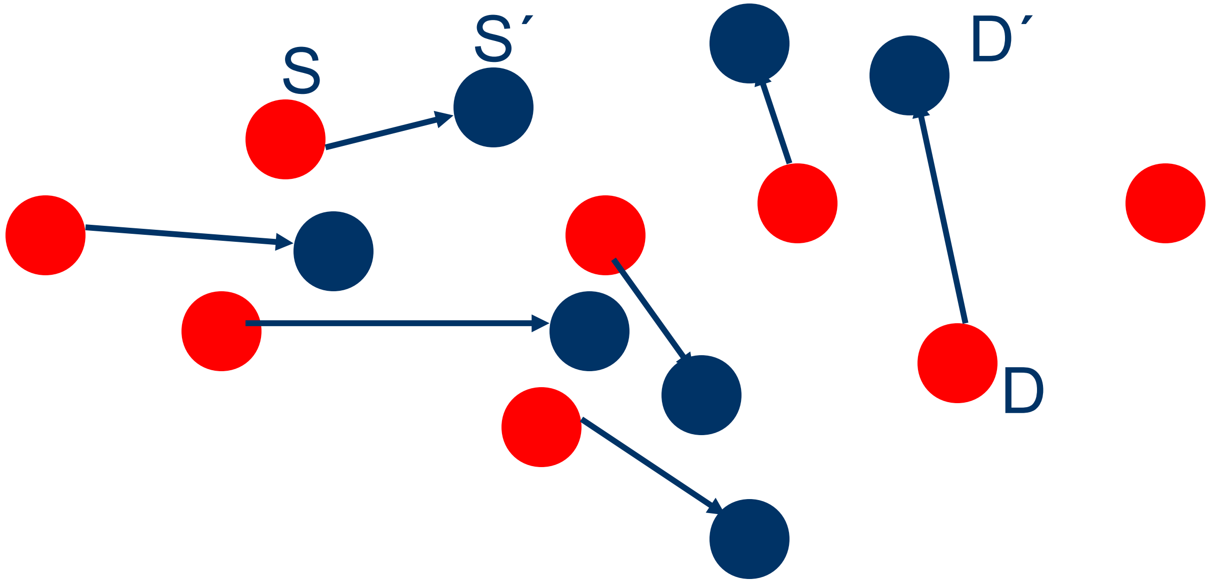
The Destination Sequenced Distance Vector (DSDV) protocol

Diwakar Yagaysen
CSE, BBDNITM, Lucknow

The Routing Problem



- The routing problem is to find a route from S to D when some or all of the nodes are mobile.

The property of ad-hoc networks

- Topology may be quite dynamic
- No administrative host
- Hosts with finite power

The properties of the ad-hoc network routing protocol

- Simple
- Less storage space
- Loop free
- Short control message (Low overhead)
- Less power consumption
- Multiple disjoint routes
- Fast rerouting mechanism

Continued…

- Routing Protocol:
  - Table-driven (proactive)
  - Source-initiated on-demand (reactive)
  - Hybrid

- Routing Algorithm
  - Link-State algorithm:

    Each node maintains a view of the network topology
  - Distance-Vector algorithm:

    Every node maintains the distance of each destination

Proactive Protocols

- Proactive protocols are based on periodic exchange of control messages and maintaining routing tables.

- Each node maintains complete information about the network topology locally.

- This information is collected through proactive exchange of partial routing tables stored at each node.

Continued...

- Since each node knows the complete topology, a node can immediately find the best route to a destination.

- However, a proactive protocol generates large volume of control messages and this may take up a large part of the available bandwidth.

- The control messages may consume almost the entire bandwidth with a large number of nodes and increased mobility.

Reactive Protocols

- In a reactive protocol, a route is discovered only when it is necessary.
- In other words, the protocol tries to discover a route only on-demand, when it is necessary.
- These protocols generate much less control traffic at the cost of latency, i.e., it usually takes more time to find a route compared to a proactive protocol.

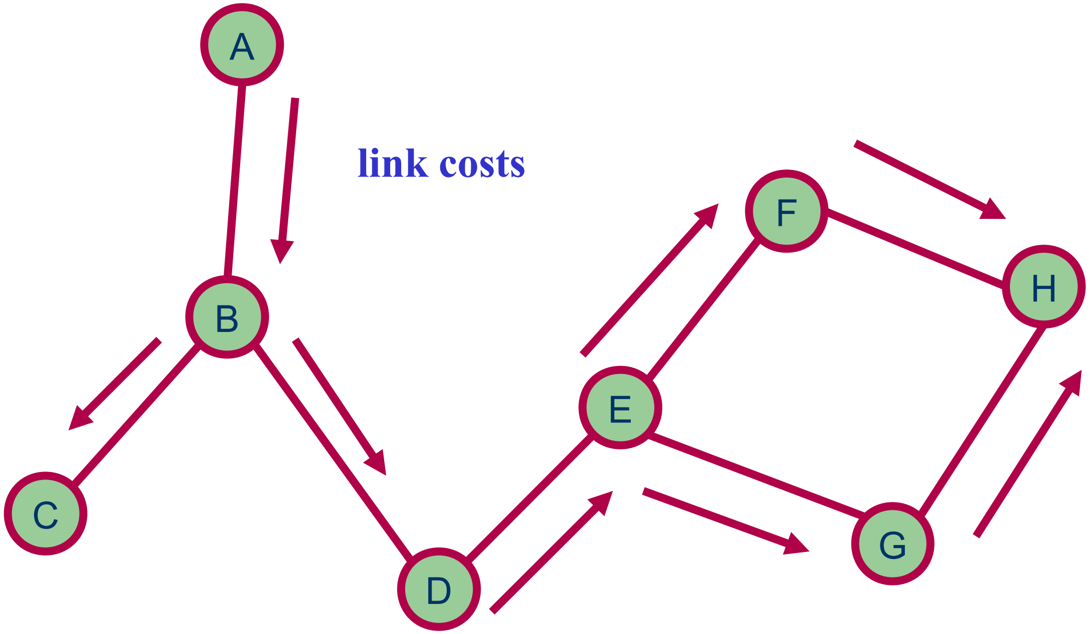
Some example protocols

- Some examples of proactive protocols are :
  - Destination Sequenced Distance Vector (DSDV)
  - STAR
- Some examples of reactive protcols are :
  - Dynamic Source Routing (DSR)
  - Ad hoc On-demand Distance Vector (AODV)
  - Temporally Ordered Routing Algorithm (TORA)

Link-State

- Each node maintains a view of the network topology with a cost for each link
- Periodically broadcast link costs to its outgoing links to all other nodes such as flooding

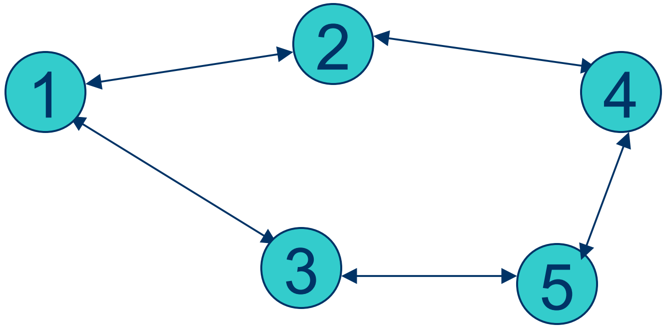
Link-State



Distance-Vector

- also known as Distributed Bellman-Ford or RIP (Routing Information Protocol)
- Every node maintains a routing table
  - all available destinations
  - the next node to reach to destination
  - the number of hops to reach the destination
- Periodically send table to all neighbors to maintain topology

Continued...

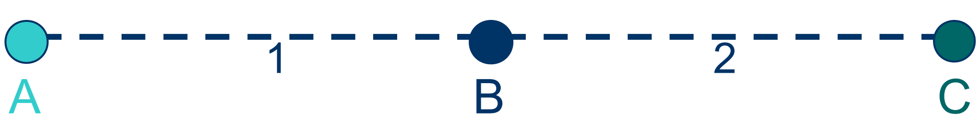


- We consider only the number of hops as the cost for sending a message from a source to a destination.

- Suppose node 1 wants to send a message to node 4.

- Since the shortest path between 1 and 4 passes through 2, 1 will send the message to 2.

Distance Vector (Tables)



A

Dest.	Next	Metric	…
A	A	0	
B	B	1	
C	B	3	

B

Dest.	Next	Metric	…
A	A	1	
B	B	0	
C	C	2	

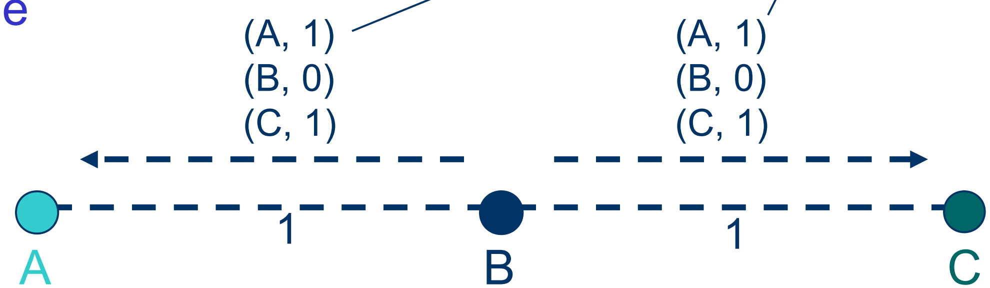
C

Dest.	Next	Metric	…
A	B	3	
B	B	2	
C	C	0	

Distance Vector (Update)

B broadcasts the new routing information to its neighbors

Routing table is updated



A

Dest.	Next	Metric	…
A	A	0	
B	B	1	
C	B	~~3~~ 2	

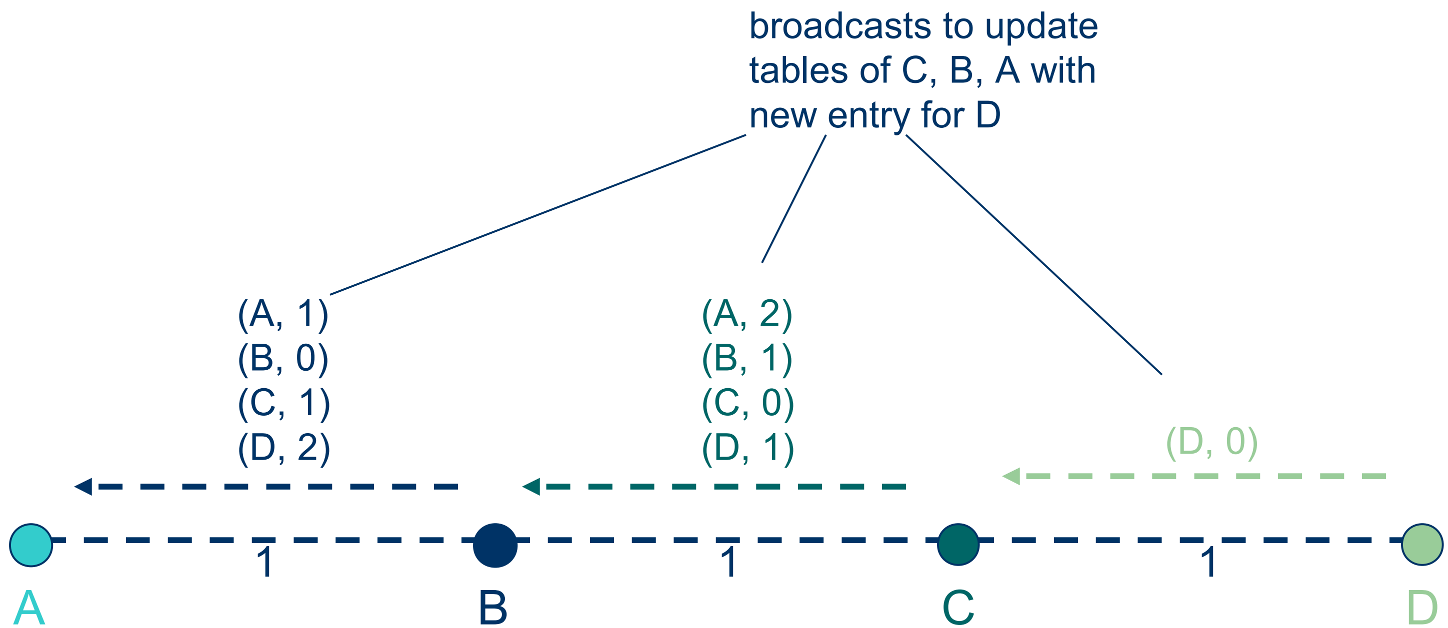
B

Dest.	Next	Metric	…
A	A	1	
B	B	0	
C	C	1	

C

Dest.	Next	Metric	…
A	B	~~3~~ 2	
B	B	1	
C	C	0	

Distance Vector (New Node)



Dest.	Next	Metric	…
A	A	0	
B	B	1	
C	B	2	
D	B	3	

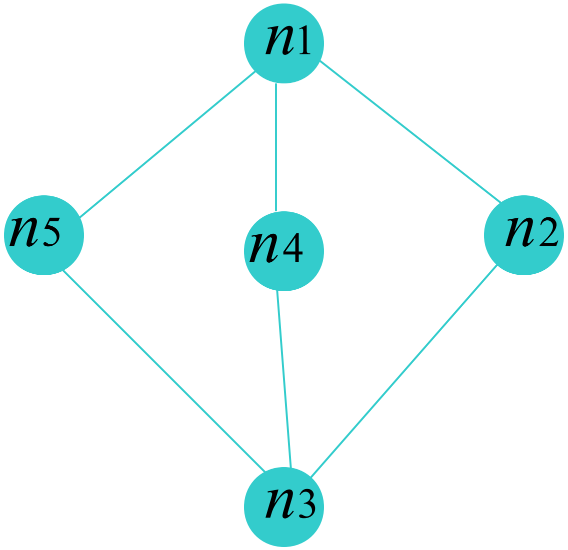
Dest.	Next	Metric	…
A	A	1	
B	B	0	
C	C	1	
D	C	2	

Dest.	Next	Metric	…
A	B	2	
B	B	1	
C	C	0	
D	D	1	

Problems with Distance Vector

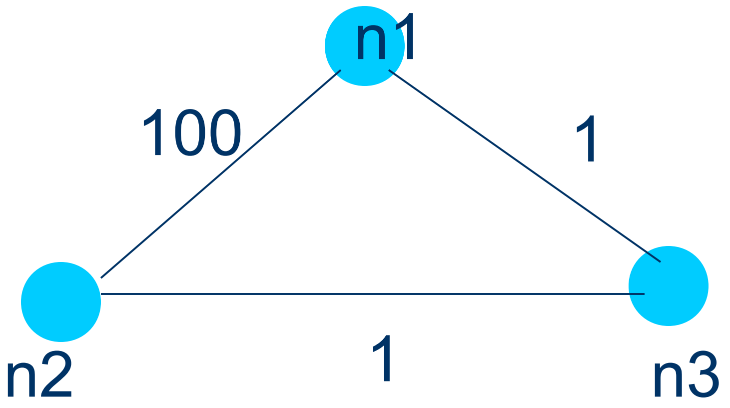
- All routing decisions are taken in a completely distributed fashion. Each node uses its local information for routing messages.
- However, the local information may be old and invalid. Local information may not be updated promptly.
- This gives rise to loops. A message may loop around a cycle for a long time.

Formation of Loops



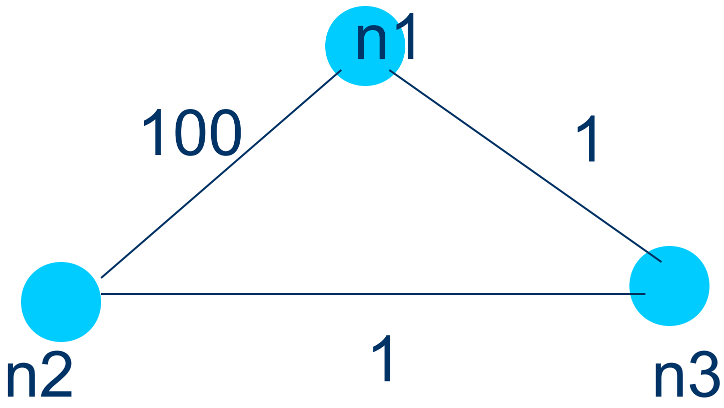
- Suppose n5 is the destination of a message from n1.
- The links between n1,n5 and n3,n5 have failed.
- A loop (n1,n4,n3,n2,n1) forms.

Counting to Infinity



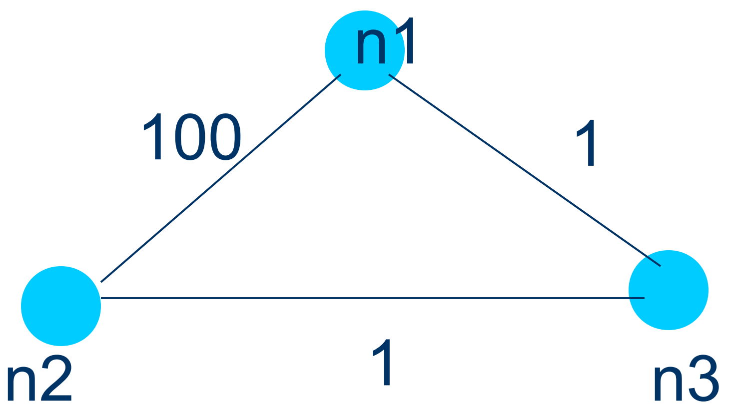
- The preferred neighbour for n2 is n3 and preferred neighbour for n3 is n2.
- Suppose the link n1-n3 fails.

Continued...



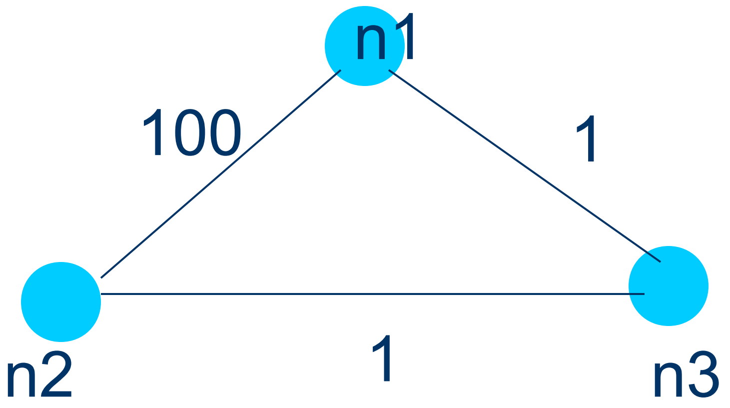
- Suppose n2 wants to send a message to n1. The only way to do this is to use link n2-n1.
- However, n2 chooses n3 as its preferred neighbour.

Continued...



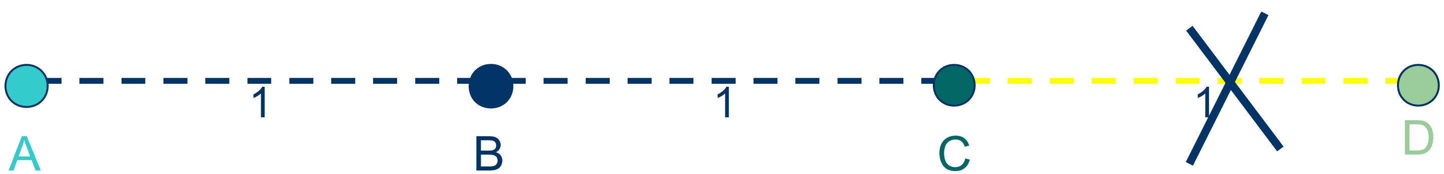
- Also, n2 knows (from old routing table) that its distance to n1 is 2.
- This information is received by n3 and n3 updates its distance to n1 as 3, i.e, 2+1.

Continued...



- Next, n2 updates its distance to n1 as 3+1=4 and so on.
- This process continues until the cost of the link n2-n1 is less than the cost of n2-n3.

Distance Vector (Broken Link): Example with Table Information

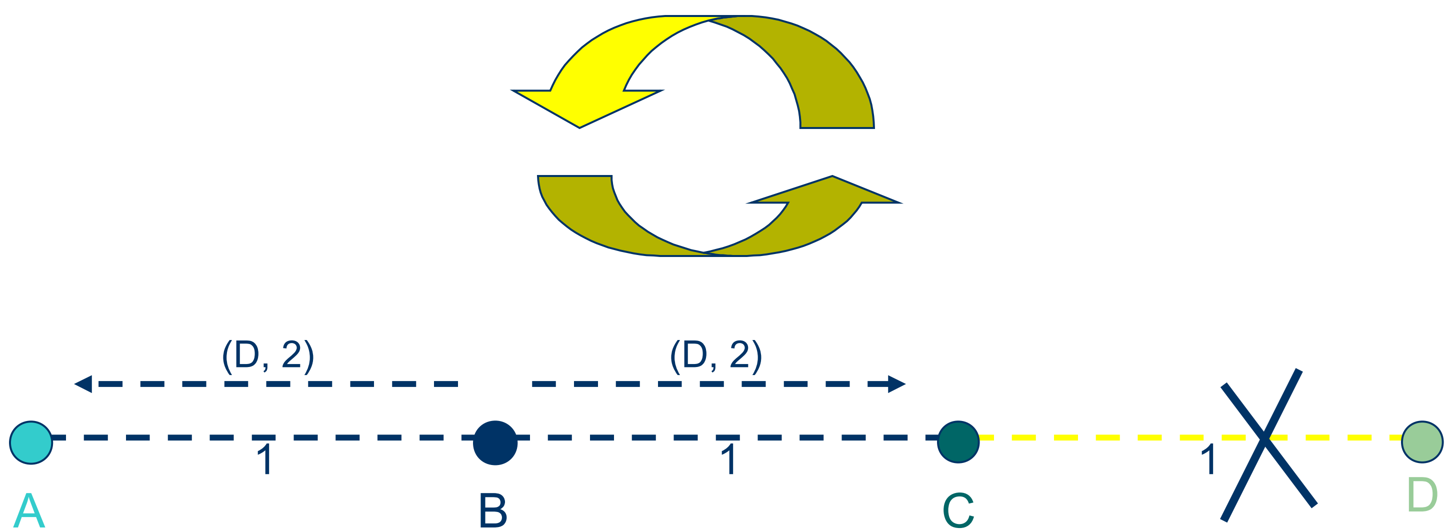


Dest.	Next	Metric	…
…	…	…	
D	B	3	

Dest.c	Next	Metric	…
…	…	…	
D	C	2	

Dest.	Next	Metric	…
…	…	…	
D	D	∞	

Distance Vector (Loops)

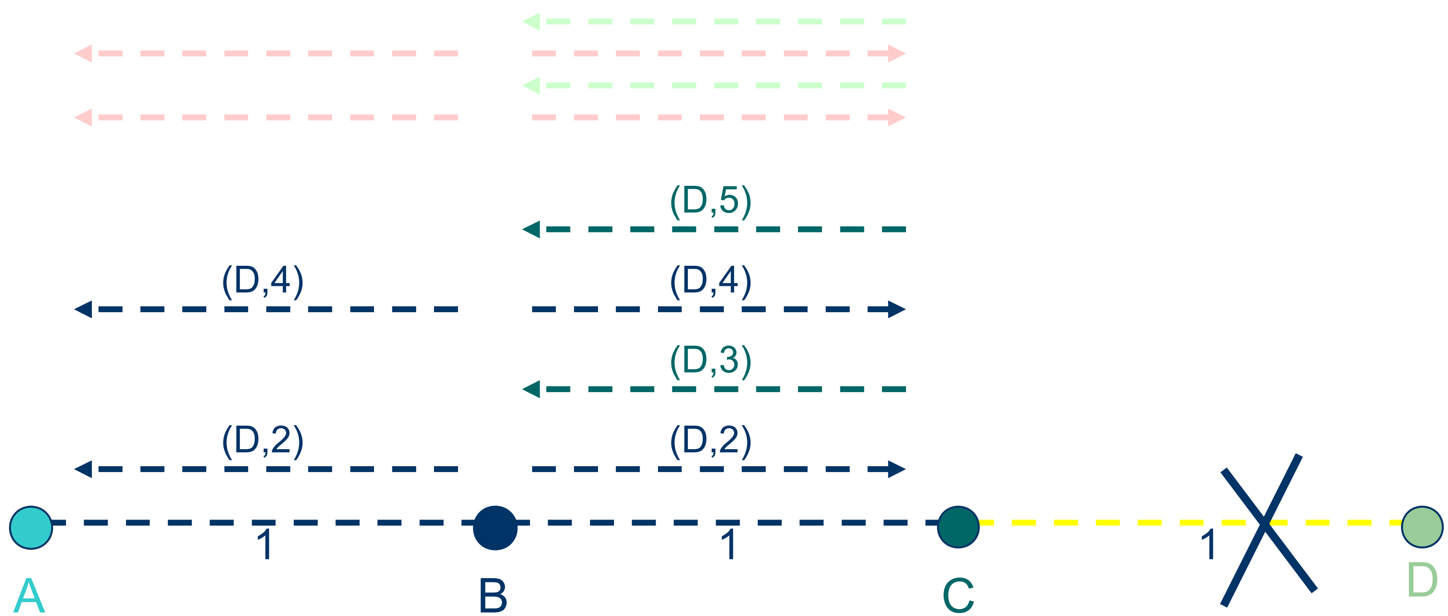


Dest.	Next	Metric	…
…	…	…	
D	B	3	

Dest.	Next	Metric	…
…	…	…	
D	C	2	

Dest.	Next	Metric	…
…	…	…	
D	B	3	

Distance Vector (Count to Infinity)



Dest.	Next	Metric	…
…	…	…	
D	B	3, 5, …	

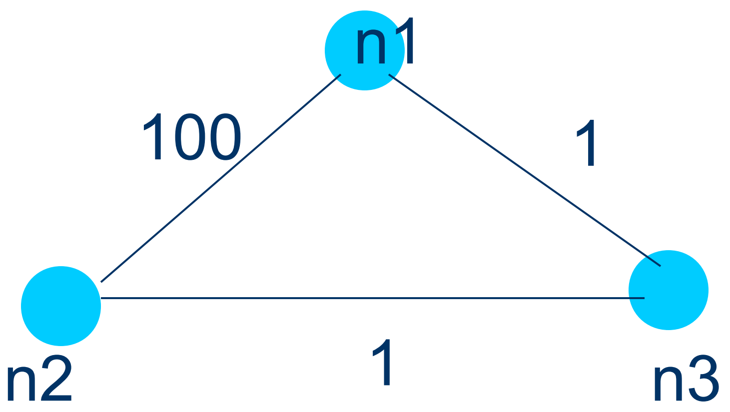
Dest.c	Next	Metric	…
…	…	…	
D	C	2, 4, 6…	

Dest.	Next	Metric	…
…	…	…	
D	B	3, 5, …	

New Versus Old Information

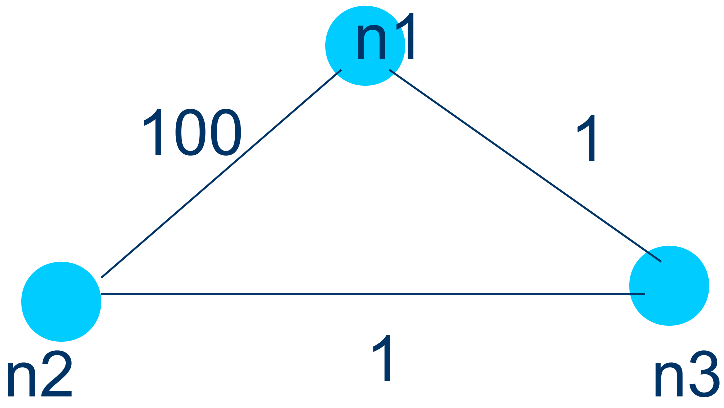
- The formation of loops and the problem of counting to infinity are due to the use of old information about the network.
- Another problem is the use of indirect information.
- If node i is trying to send a message to node x,it is better to consider the view of node x.

How to Use New Information



- Suppose each node broadcasts its routing table stamped with an increasing sequence number.
- Initially, n2 will receive updates from n1 and knows that the distance of n1 is 2.

Continued...



- However, when the link n1-n3 is broken, this will be noted by n1 in its routing table.
- In future, n2 will receive broadcasts from n1 with this information and avoid the path through n3.

Timestamps

- Each time a node like n1 broadcasts its routing table, it adds an increasing sequence number (timestamp) to the broadcast.
- Any node receiving the broadcast rejects old routing information and takes the new information for updating its routing table.
- This avoids looping and counting to infinity.

How to maintain routing tables?

- Routing tables are maintained by periodically broadcasting the tables stored in each node.

- Each node executes an algorithm like Dijkstra's shortest path algorithm to update its table.

- The broadcasts are done through a flooding scheme.

Distance Vector

- Distance Vector is not suited for ad-hoc networks!
  - Loops
  - Count to Infinity

- New Solution -> DSDV Protocol

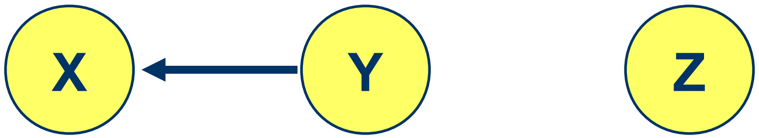
Destination-Sequenced Distance-Vector (DSDV): Summary

- Each node maintains a routing table which stores
  - next hop, cost metric towards each destination
  - a sequence number that is created by the destination itself
- Each node periodically forwards routing table to its neighbors
  - Each node increments and appends its sequence number when sending its local routing table
- Each route is tagged with a sequence number; routes with greater sequence numbers are preferred

- Each node advertises a monotonically increasing even sequence number for itself
- When a node finds that a route is broken, it increments the sequence number of the route and advertises it with infinite metric
- Destination advertises new sequence number

Continued…

- When X receives information from Y about a route to Z
  - Let destination sequence number for Z at X be S(X), S(Y) is sent from Y



  - If S(X) > S(Y), then X ignores the routing information received from Y
  - If S(X) = S(Y), and cost of going through Y is smaller than the route known to X, then X sets Y as the next hop to Z
  - If S(X) < S(Y), then X sets Y as the next hop to Z, and S(X) is updated to equal S(Y)

DSDV Protocol

- We consider a collection of mobile computers, (nodes) which may be far from any base station.
- The computers (nodes) exchange control messages to establish multi-hop paths in the same way as the Distributed Bellman-Ford algorithm.
- These multi-hop paths are used for exchanging messages among the computers (nodes).

Continued…

- Packets are transmitted between the nodes using routing tables stored at each node.
- Each routing table lists all available destinations and the number of hops to each destination.
- For each destination, a node knows which of its neighbours leads to the shortest path to the destination.

Continued…

- Consider a source node S and a destination node D.
- Each route table entry in S is tagged with a sequence number that is originated by the destination node.
- For example, the entry for D is tagged with a sequence number that S received from D (may be through other nodes).

Continued…

- We need to maintain the consistency of the routing tables in a dynamically varying topology.

- Each node periodically transmits updates. This is done by each node when significant new information is available.

- We do not assume any clock synchronization among the mobile nodes.

Continued…

- The route-update messages indicate which nodes are accessible from each node and the number of hops to reach them.

- We consider the hop-count as the distance between two nodes. However, the DSDV protocol can be modified for other metrics as well.

Continued…

- A neighbour in turn checks the best route from its own table and forwards the message to its appropriate neighbour. The routing progresses this way.

- There are two issues in this protocol :
  - How to maintain the local routing tables
  - How to collect enough information for maintaining the local routing tables

Maintaining Local Routing Table

- We will first assume that each node has all the necessary information for maintaining its own routing table.

- This means that each node knows the complete network as a graph. The information needed is the list of nodes, the edges between the nodes and the cost of each edge.

Continued…

- Edge costs may involve : distance (number of hops), data rate, price, congestion or delay.

- We will assume that the edge cost is 1 if two nodes are within the transmission range of each other.

- The DSDV protocol can also be modified for other edge costs.

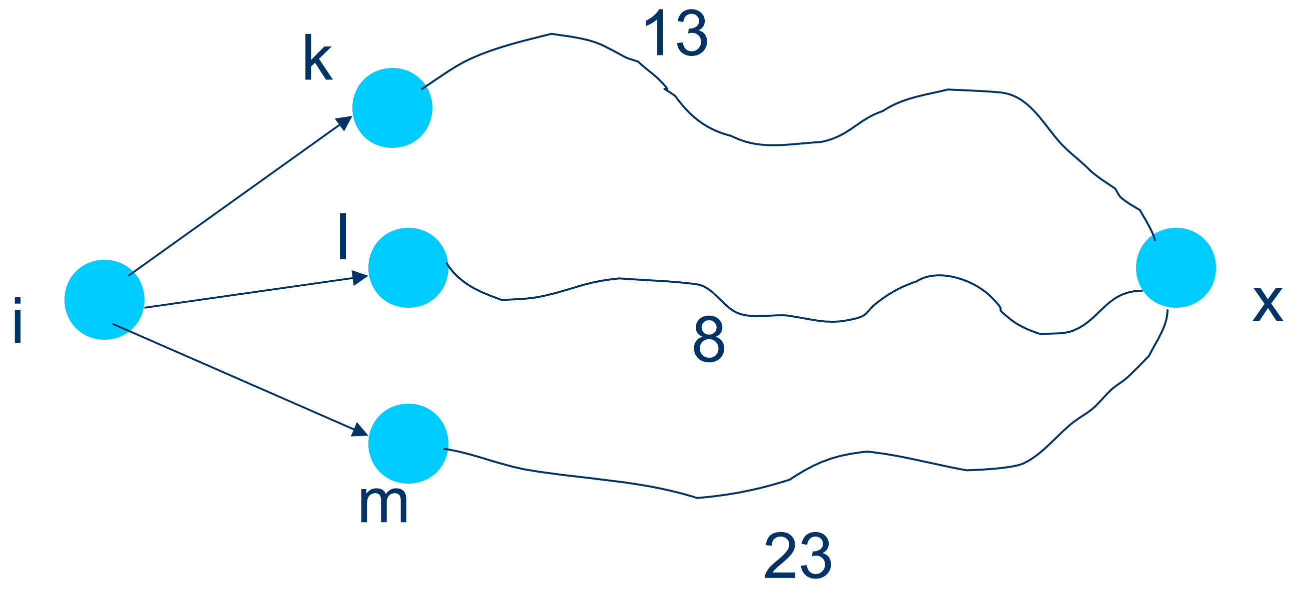
How the Local Routing Table is Used

- Each node maintains its local routing table by running the distributed Bellman-Ford algorithm.

- Each node i maintains, for each destination x , a set of distances $d_{ij}(x)$ for each neighbour j

- Node i treats neighbour k as the next hop for a packet destined for x if $d_{ik}(x)$ equals minimum of all $d_{ij}(x)$

Continued…



The message will be sent from i to l as the cost of the path to x is minimum through l

Collecting Information for Building Local Table

- Each node exchanges information with its neighbors to keep its local routing table updated.

- Whenever a node receives some new information about other nodes, it sends this information to its neighbours.

- Neighbours update their routing tables with this new information.

Route Advertisements

- The DSDV protocol requires each mobile node to advertise its own routing table to all of its current neighbours.
- Since the nodes are mobile, the entries can change dynamically over time.
- The route advertisements should be made whenever there is any change in the neighbourhood or periodically.

Continued…

- Each mobile node agrees to forward route advertising messages from other mobile nodes.

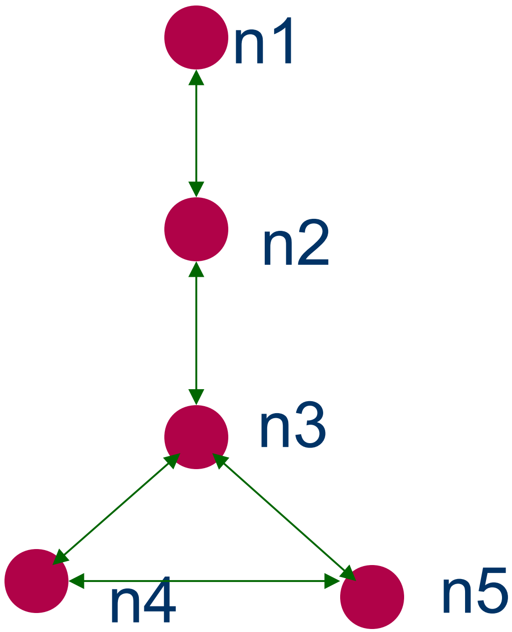
- This forwarding is necessary to send the advertisement messages all over the network.

- In other words, route advertisement messages help mobile nodes to get an overall picture of the topology of the network.

Route Table Entry Structure

- The route advertisement broadcast by each mobile node has the following information for each new route :
  - The destination's address
  - The number of hops to the destination
  - The sequence number of the information received from that destination. This is the original sequence number assigned by the destination.

An Example of Route Update

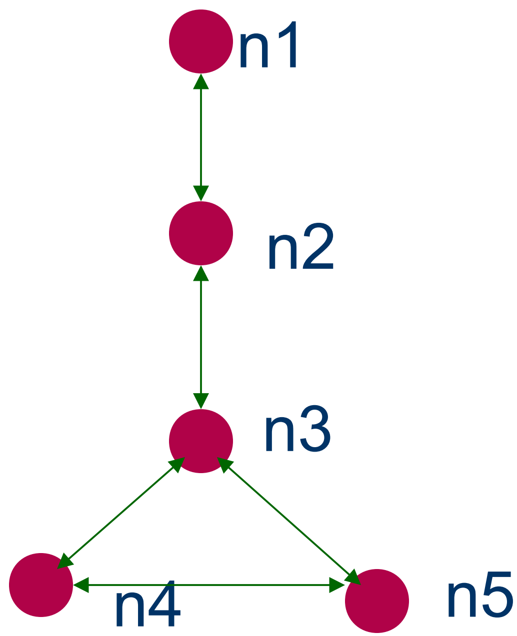


- At the start, each node gets route updates only from its neighbour.
- For n4, the distances to the other nodes are :

  n5=1, n3=1, n2= ∞

  n1 = ∞
- All nodes broadcast with a sequence number 1

Continued…

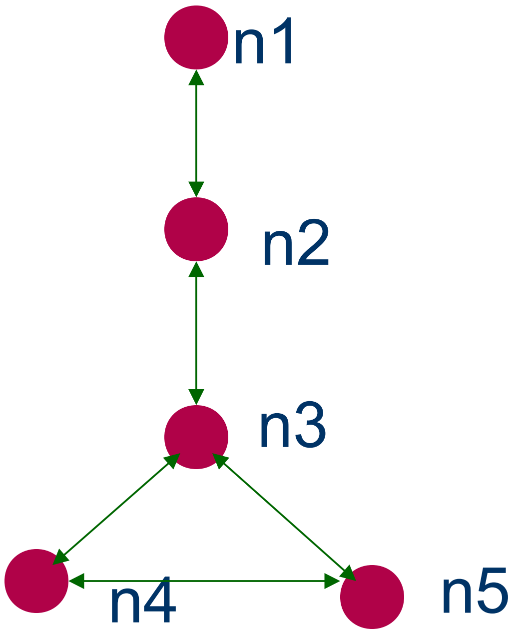


- After this, nodes forward messages that they have received earlier.
- The message that n2 sent to n3 is now forwarded by n3
- For n4, the distances are now :

  n5=1, n3=1, n2=2, n1=∞

All messages have sequence number 1

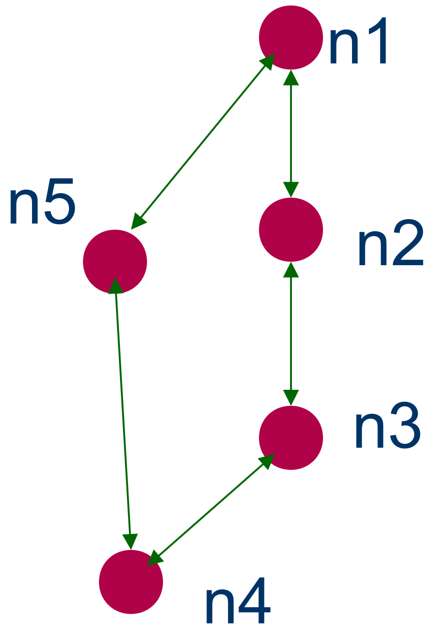
Continued…



- Finally, after second round of forwarding, n4 gets the following distances :

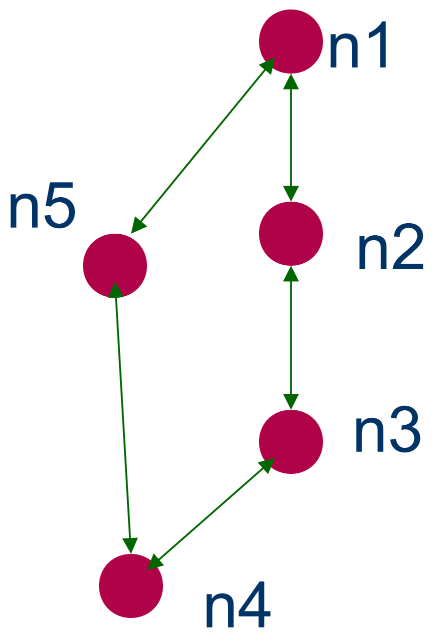
  n5=1, n3=1, n2=2, n1=3

Continued…



- Suppose n5 has moved to its new location.
- Also, n5 receives a new message from n1 with a sequence number 2
- This message is forwarded by n5 to n4
- Two distances to n1 in n4

Continued…



- Distance 3 with seqence number 1, and
- Distance 2 with sequence number 2

- Since the latter message has a more recent sequence number, n4 will update the distance to n1 as 2

Route Table Entry Structure

- For example, a node n may receive two different messages originating from another node m.
- However, node n will forward the most recent message from m to its neighbours.
- Usually n will add one extra hop to the routes in the message received from m as the destination is one more hop away.

Responding to Topology Changes

- Some of the links in a mobile network may be broken when the nodes move.

- A broken link is described by a distance ∞
- When a link to a next hop is broken, any route through that next hop is given a distance ∞

- This is considered as a major change in the routing table and immediately broadcast.

Topology Changes in large mobile networks

- The number of routing updates may be quite high in a large network with high level of mobility.
- It is necessary to avoid excessive control traffic (route update information) in such networks. Otherwise, the bandwidth will be taken up by control traffic.
- The solution is to broadcast two types of updates.

Full Dump and Incremental Dump

- A full dump carries complete routing tables. A node broadcasts a full dump infrequently.
- An incremental dump carries minor changes in the routing table. This information contains changes since the last full dump.
- When the size of an incremental dump becomes too large, a full dump is preferred.

Route Selection Criteria

- When a node i receives incremental dump or full dump from another node j , the following actions are taken :
  - The sequence number of the current dump from j is compared with previous dumps from j
  - If the sequence number is new, the route table at i is updated with this new information.
  - Node i now broadcasts its new route table as an incremental or a full dump.

How frequently should a node broadcast?

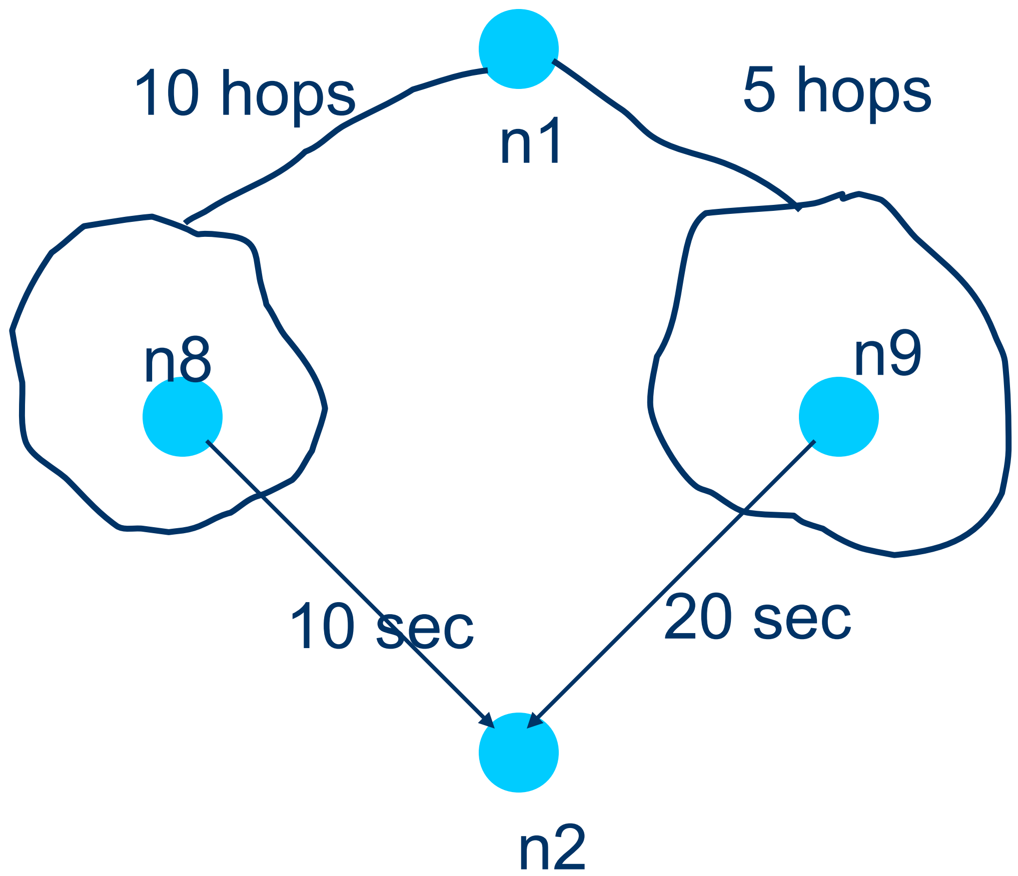
- A node decides on a new route based on one of the two criteria :
  - If a route has a smaller metric (distance) to a destination
  - Or, if an update from the destination with a new sequence number has been received.

- However, it is not desirable that a node broadcasts an update every time it has updated its routing table.

Reducing the number of updates

- A node i may receive the same update message from another node j through several different paths.
- Suppose, one of the updates has a lowest distance to j
- It is better to avoid broadcasting every new update and instead broadcast only the lower metric updates.

An Example

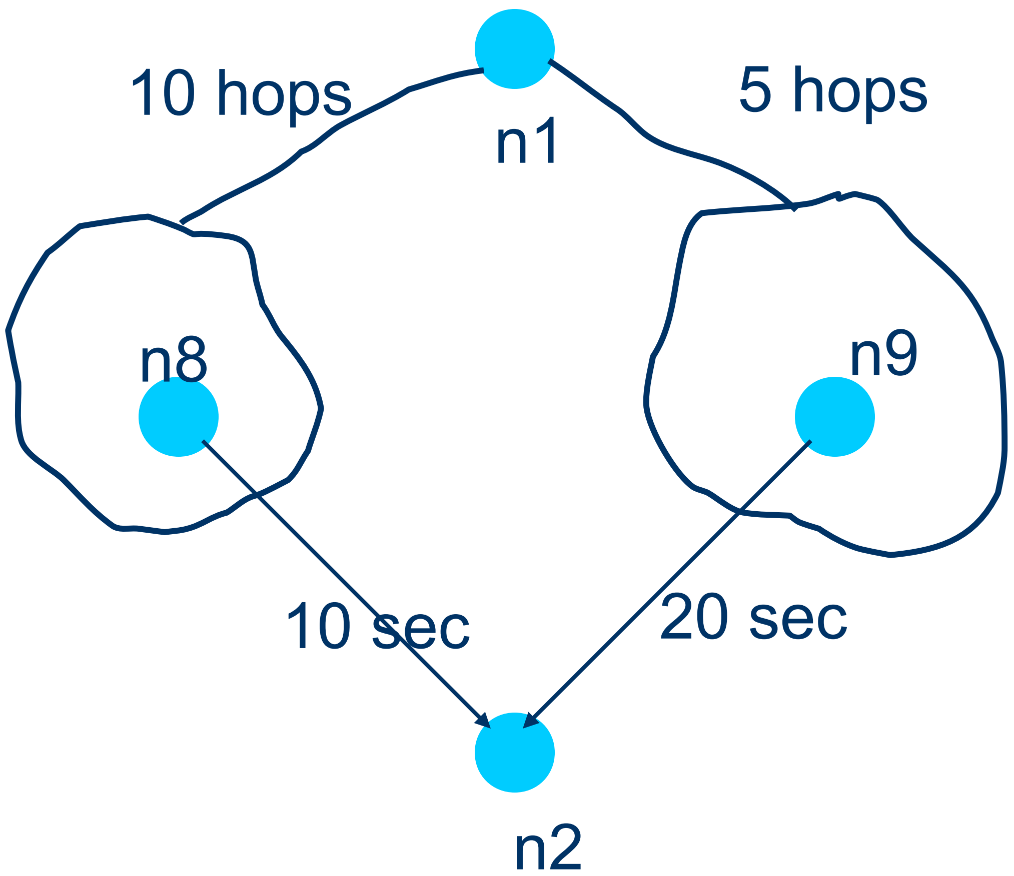


Node n2 should wait longer to get an update from n9

Settling Before Sending an Update Message

- Each node should maintain some statistics about the average settling time of a message from another node.
- In the previous example, n2 receives several messages from n1 with the same sequence number.
- Depending on statistics about last settling time, n2 should wait until it receives all messages from n1.

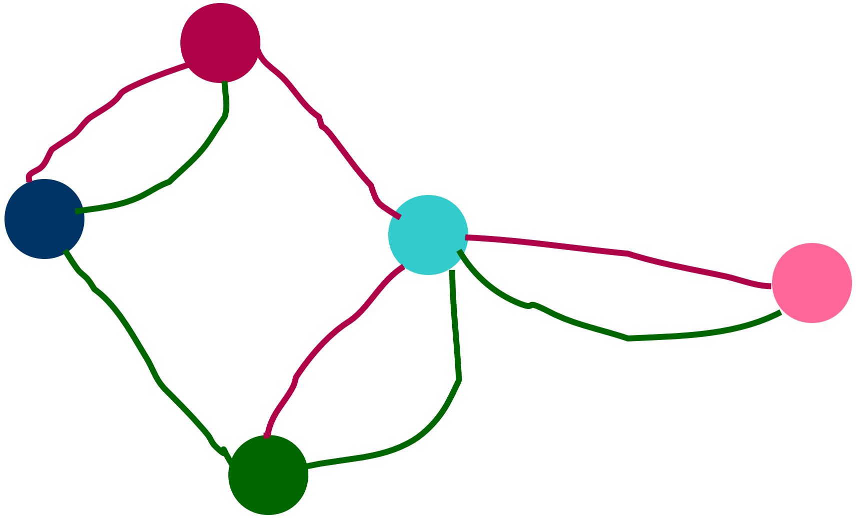
An Example



This reduces the number of updates sent by a node

DSDV Guarantees Loop-Free Paths (Intuitive Proof)

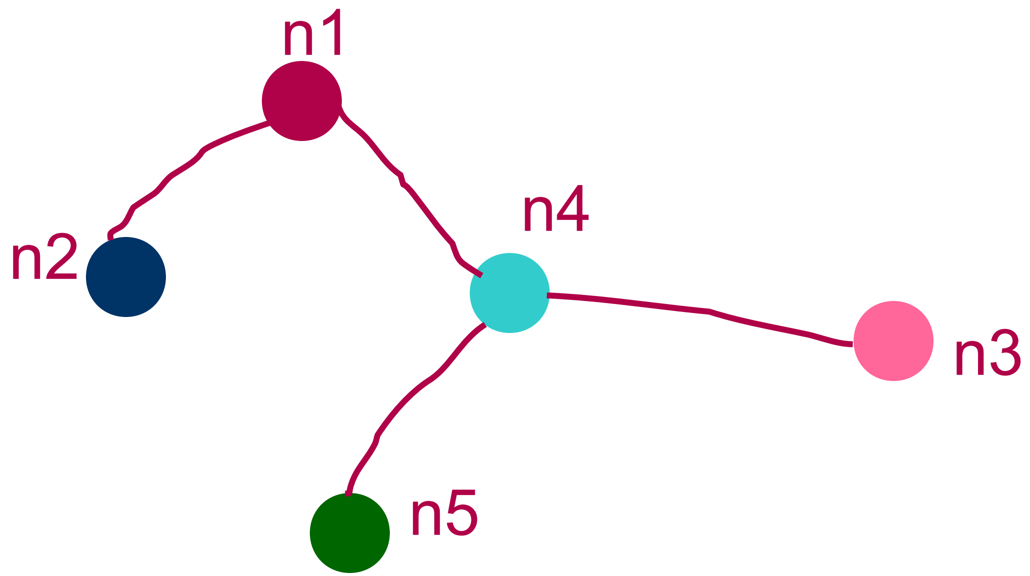
- For an n-node ad hoc network, DSDV maintains n rooted trees, one for each destination.



We have shown two such trees here.

Continued…

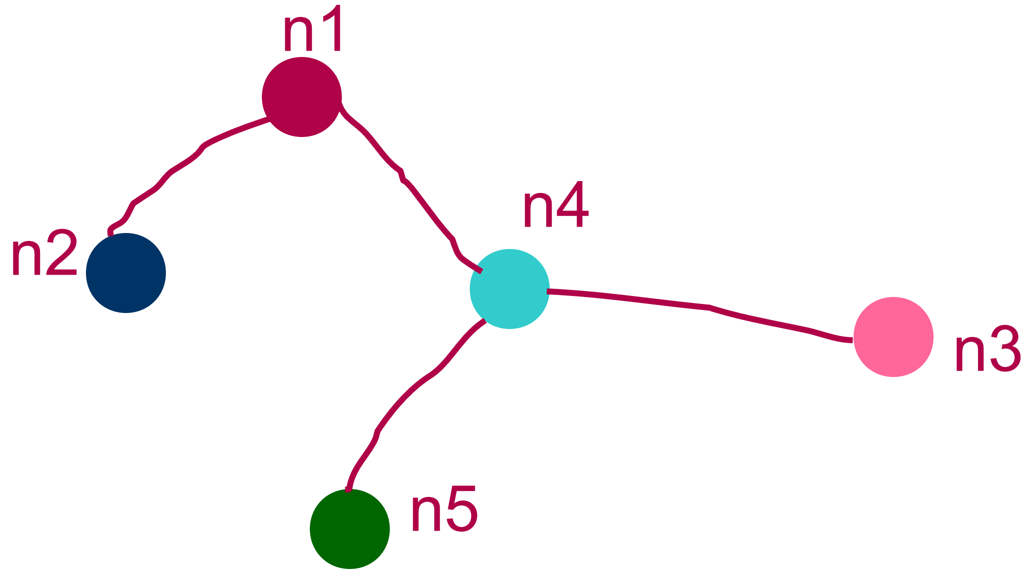
- Consider the tree rooted at n1. Suppose n3 wants to change its link.



- A message from n5 reaches to n3

Continued…

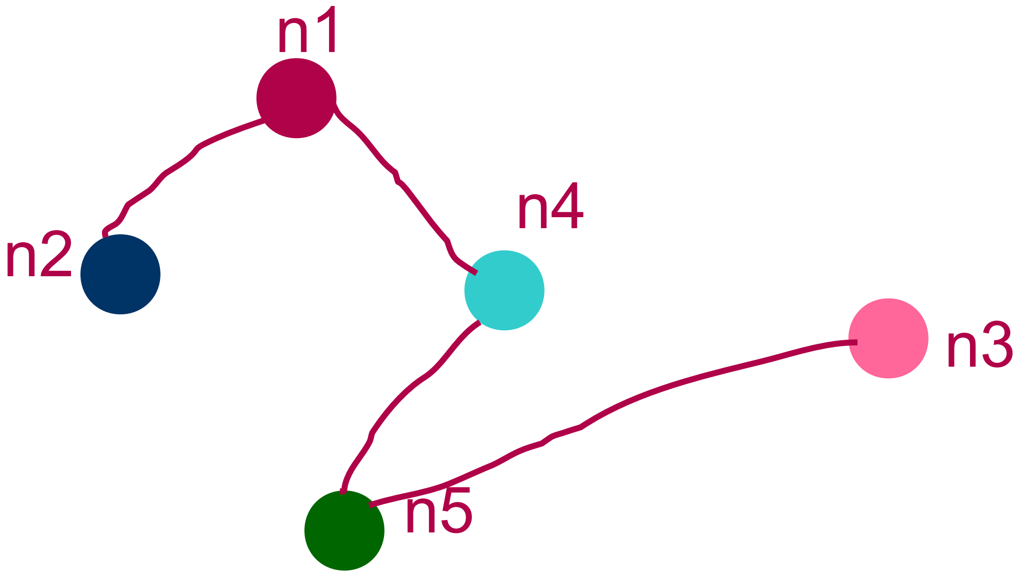
- This message is originally from n1. The message must have either a higher sequence



Number or lower distance to n1

Continued…

- However, n3 removes the old link to n4 before connecting the new link to n5



Advantages of DSDV

- DSDV is an efficient protocol for route discovery. Whenever a route to a new destination is required, it already exists at the source.
- Hence, latency for route discovery is very low.
- DSDV also guarantees loop-free paths.

Disadvantages

- However, DSDV needs to send a lot of control messages. These messages are important for maintaining the network topology at each node.
- This may generate high volume of traffic for high-density and highly mobile networks.
- Special care should be taken to reduce the number of control messages.