Destination Sequenced Distance Vector Routing (DSDV)

Introduction

- Destination Sequenced Distance Vector Routing protocol is a modified version of Bellman Ford Algorithm and is based upon the concepts of Distance Vector Routing.
- In Distance Vector Routing(DVR), each node broadcasts a table containing its distance from nodes which are directly connected and based upon this, other nodes broadcasts the updated routing. Those nodes which are unreachable directly are labelled as "infinite".
- But, this updation of routing tables keeps on happening and an infinite loop is generated which is commonly known as Count-To-Infinity problem.
- To overcome this problem of count to infinity by generating sequence number in the routing table, every time the routing table is updated. The process of DSDV is same as that of Distance Vector Routing but an extra attribute of sequence number is added.

Destination Sequenced Distance Vector Routing : Concept

- DSDV protocol uses and maintains a single table only, for every node individually. The table contains the following attributes.
    - Routing Table : It contains the distance of a node from all the neighboring nodes along with the sequence number(SEQ No means the time at which table is updated).

Format :

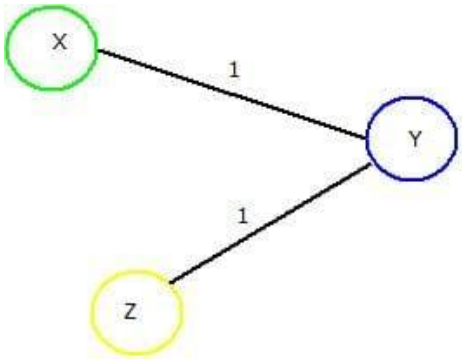
Node	Destination	Next Hop	Distance	SEQ No

Destination Sequenced Distance Vector Routing : Format

- This table is updated on every step and ensures that each node broadcast as well as receives correct information about all the nodes including their distance and sequence number.

Destination Sequenced Distance Vector Routing Protocol : Working

- In DSDV, nodes broadcasts their routing tables to immediate neighbors with the sequence number. Every time any broadcasting occurs, the sequence number is also updated along with distances of nodes.
- Consider a network of 3 nodes having distances of "1" on each of the edges respectively. Below mentioned steps will let you know how DSDV works and routing tables are updated.



Destination Sequenced Distance Vector Routing : Sample Network

- Step-1: Draw separate tables for all the nodes “X”, “Y” & “ Z” along with the distance and sequence number.

For X:

Source	Destination	Next Hop	Cost	SEQ No
X	X	X	0	100-X
X	Y	Y	1	200-Y
X	Z	Y	2	300-Z

For Y:

Source	Destination	Next Hop	Cost	SEQ No
Y	X	X	1	100-X
Y	Y	Y	0	200-Y
Y	Z	Y	1	300-Z

For Z:

Source	Destination	Next Hop	Cost	SEQ No
Z	X	Y	2	100-X
Z	Y	Y	1	200-Y
Z	Z	Z	0	300-Z

- If “Y” wants to broadcast the routing table. Then updated routing tables of all the nodes in the network will look like as depicted in the below tables where red marked cell denotes the change in sequence number.

For X:

Source	Destination	Next Hop	Cost	SEQ No
X	X	X	0	100-X
X	Y	Y	1	210-Y
X	Z	Y	2	300-Z

For Y:

Source	Destination	Next Hop	Cost	SEQ No
Y	X	X	1	100-X
Y	Y	Y	0	210-Y
Y	Z	Z	1	300-Z

For Z:

Source	Destination	Next Hop	Cost	SEQ No
Z	X	Y	2	100-X
Z	Y	Y	1	210-Y
Z	Z	Z	0	300-Z

Advantages :

- Can’t be implemented commercially or on larger scale.
- Efficient results will be produced if applied on small networks.

Disadvantages :

- Slower protocol processing time.
- Less bandwidth.
- Not suitable for large number of networks which are dynamic in nature.

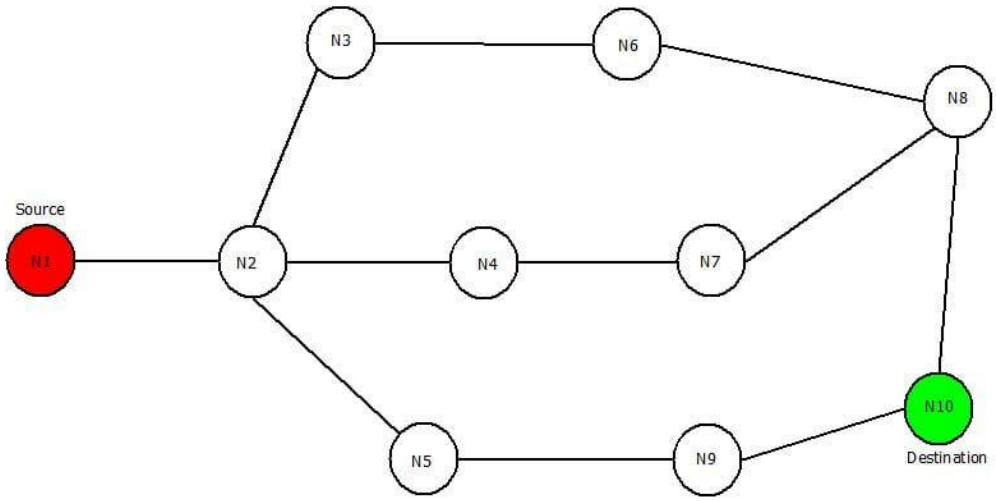
Dynamic Source Routing

Introduction

- Dynamic Source Routing (DSR) comes under the reactive routing protocol category, as it is capable of discovering the route from source to destination only when required and needed.
- Dynamic Source Routing protocol uses a process called "**Route Discovery Mechanism**" that is capable of discovering the route for data packets from source node to destination nodes using intermediate nodes.
- As like proactive routing protocols such as Global State Routing an Dynamic Sequence Distance Vector Routing no separate table is maintained.
- The major change in DSR as compare to GSR and DSDV is, in DSDV after asking a requirement of route from source to destination, path via intermediate nodes is checked for its length. Then a "Re-Request" packet is sent back from destination to source via the smallest route possible in the whole network. The "Re-Request" packet does contains its unique ID also.
- This process of separately sending a "Re-Request" packet from destination to source makes it easier for the sender to send the data packets on fixed path rather than sending it on multiple paths to check for total distance.

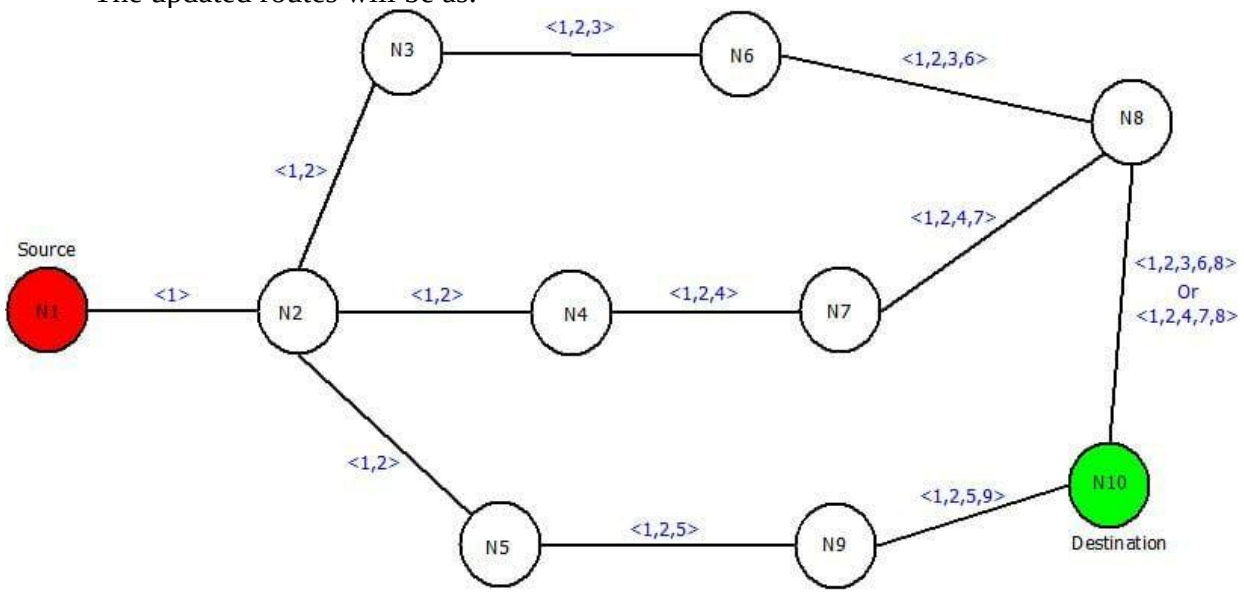
Dynamic Source Routing Protocol : Working

- Dynamic Source Routing does broadcast the route to its neighbors but does not floods the information. It only trace the route by calculating the total distance or by calculating the number of nodes present in between source and destination nodes.
- Consider a network containing 10 nodes where node N1 being the source and node N10 being the destination nodes. Below mentioned steps will let you know how DSR protocol works and how Re-Request packet is transmitted through the network.



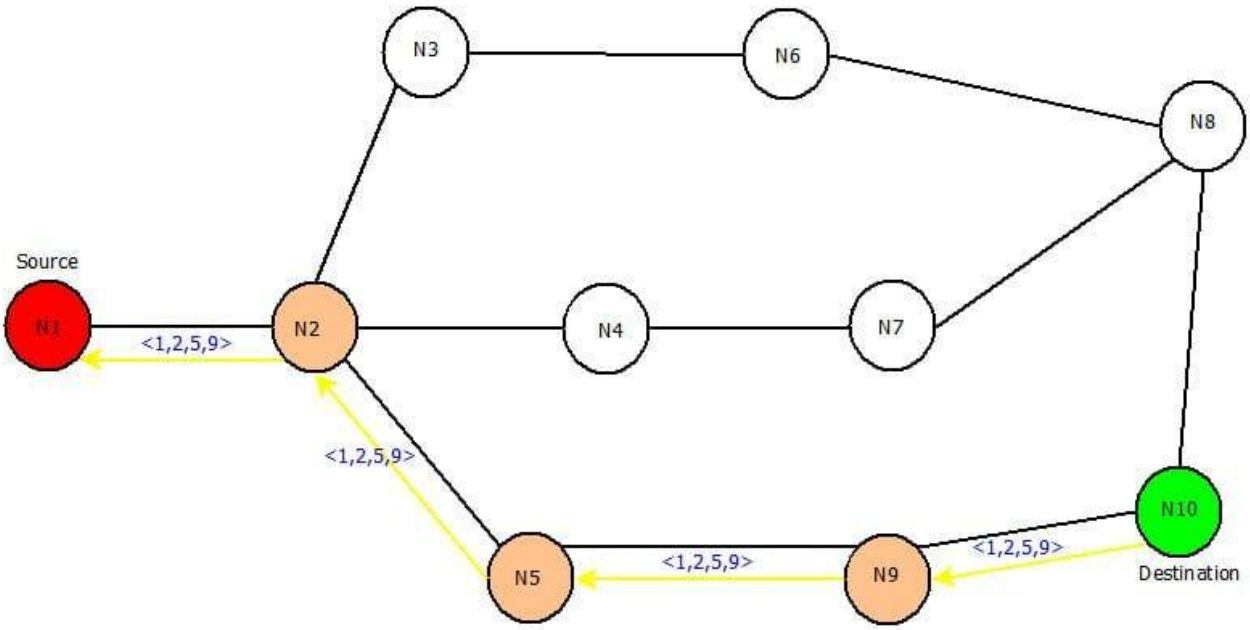
Dynamic Source Routing : Network

- Step 1: Start from source node N1 and broadcast the information about it to its neighbors i.e. in this case the route information is “<1>”, because of its one-to-one link between node N1 and N2.
- Step 2: Broadcast previous route information to neighbors of node N2 i.e. to node N3, N4, N5. The new route will remain same “<1,2>” in all the cases.
- Step 3: Take node N3 and broadcast previous route(<1,2>) to next neighboring nodes i.e. node N6. New route till node N6 will be “<1,2,3>” and same process can be done for other nodes i.e. Node N4 and N5.
- Step 4 : Further, broadcast the new routes i.e. <1,2,3,6> , <1,2,4> , <1,2,5> to nodes N8, N7 & N9 respectively.
- Step 5: Repeat the above steps until destination node is reached via all the routes.
- The updated routes will be as:



Dynamic Source Routing : Updated Network

- After this, “Re-Request” packet will be sent in backward direction i.e. from destination node “N10” to source node “N1”. It will trace the shortest route by counting the number of nodes from route discovered in previous steps.
- The three possible routes are :
  - Route 1: <1,2,3,6,8>
  - Route 2: <1,2,4,7,8>
  - Route 3: <1,2,5,9>
- Route 3 i.e. "<1,2,5,9>” will be chosen as it contains the least number of nodes and hence it will definitely be the shortest path and then data can be transferred accordingly.
- The Re-Request Packet route can be located as:



Dynamic Source Routing

Advantages :

- A perfect route is discovered always.
- Highly efficient.
- Low bandwidth Consumption.

Disadvantages :

- If the route gets broke, data transmission cannot happen.
- Time taking algorithm-Slow.
- If network is large , then it is impossible for the data packets header to hold whole information of the routes.

Ad-Hoc On Demand Distance Vector Routing Protocol(AODV)

Introduction

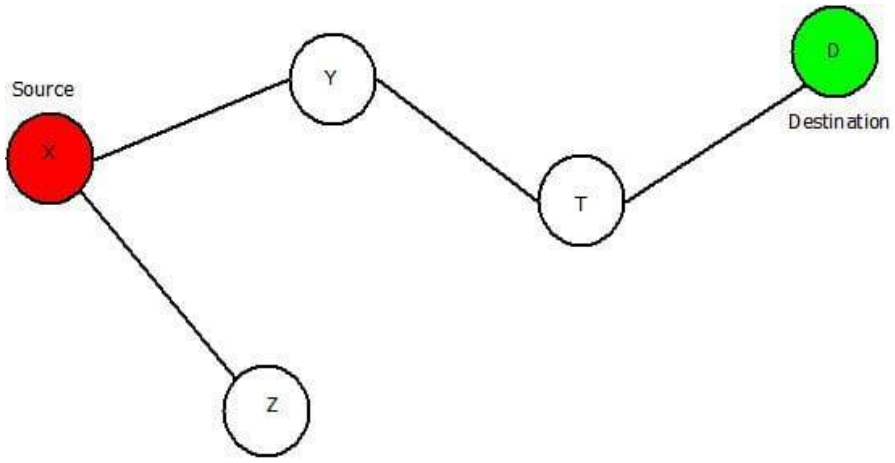
- Another type of reactive routing protocol which does not maintain routes but build the routes as per requirements is Ad-Hoc On Demand Distance Vector Routing Protocol.
- AODV is used to overcome the drawbacks of Dynamic Source Routing Protocol and Distance Vector Routing Protocol i.e. Dynamic Source Routing is capable of maintaining information of the routes between source and destination which makes it slow. If the network is very large containing a number of routes from source to destination, it is difficult for the data packets header to hold whole information of the routes.
- In case of Dynamic Source Routing, multiple routes are present for sending a packet from source to destination but AODV overcomes this disadvantage too.
- In AODV, along with routing tables of every node, two counters including Sequence Number(SEQ NO) and broadcast ID are maintained also.
- The destination IP is already known to which data is to be transferred from source. Thus, the destination Sequence Number(SEQ NO) helps to determine an updated path from source to destination.
- Along with these counters, Route Request(RREQ) and Route Response(RRESP) packets are used in which RREQ is responsible for discovering of route from source to destination and RRESP sends back the route information response to its source.

Ad-Hoc On Demand Distance Vector Routing Protocol : Working

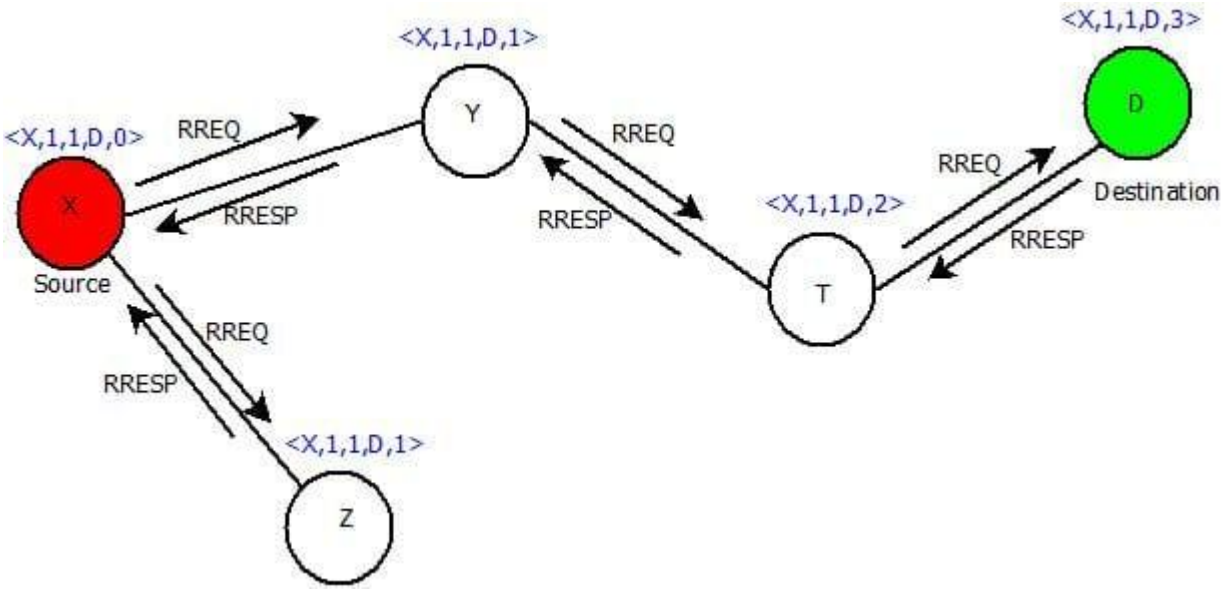
- In Ad-Hoc On Demand Distance Vector Routing, the source node and destination nodes IP addresses are already known.
- The goal is to identify, discover and maintain the optimal route between source and destination node in order to send/receive data packets and informative.
- Each node comprises of a routing table along with below mentioned format of Route Request(RREQ) packet.

RREQ { Destination IP, Destination Sequence Number, Source IP, Source Sequence Number, Hop Count}.

- Consider a network containing 5 nodes that are “X”, “Y”, “Z”,”T”,”D” present at unit distance from each other, where “X” being the source node and “D” being the destination node.



- The IP addresses of source node “X” and destination node “D” is already known. Below mentioned steps will let you know how AODV works and concept of Route Request(REREQ) and Route Response(RRESP) is used.
    - Step 1: Source node “X” will send Route Request i.e. RREQ packet to its neighbours “Y” and “Z”.
    - Step 2: Node “Y” & “Z” will check for route and will respond using RRESP packet back to source “X”. Here in this case “Z” is the last node but the destination. It will send the RREQ packet to “X” stating “Route Not Found”. But node “Y” will send RRESP packet stating “Route Found” and it will further broadcast the RRESP to node “T”.
    - Step 3: Now the field of net hop in the RREQ format will be updated, Node “T” will send back the “Route Found” message to Node “Y” and will update the next hop field further.
    - Step 4: Then Node “T” will broadcast and RREQ packet to Node “D”, which is the destination and the next hop field is further updated. Then it will send RRES packet to “T” which will further be sent back to the source node “X” via node “Y” and Node “T” resulting in generation of an optimal path. The updated network would be:



Advantages : Ad-Hoc On Demand Distance Vector Routing Protocol

- Dynamic networks can be handled easily.
- No loop generation.

Disadvantages : Ad-Hoc On Demand Distance Vector Routing Protocol

- A delayed protocol because of its route discovery process.
- High bandwidth requirement.

Cluster-head Gateway Switch Routing (CGSR)

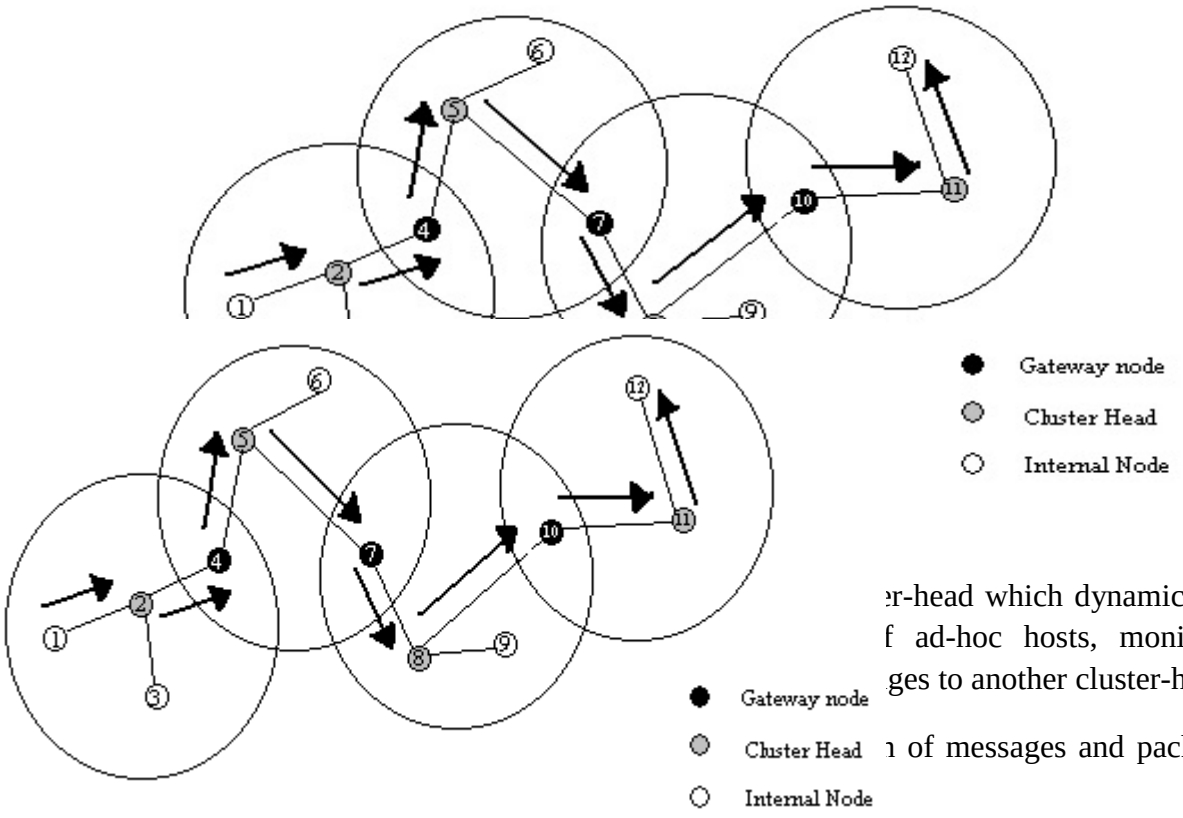
The cluster-head gateway switch routing (CGSR) is a hierarchical routing protocol. It is a proactive protocol.

When a source routes the packets to destination, the routing tables are already available at the nodes. A cluster higher in hierarchy sends the packets to the cluster lower in hierarchy. Each cluster can have several daughters I and forms a tree-like structure in CGSR.

CGSR forms a cluster structure. The nodes aggregate into clusters using an appropriate algorithm. The algorithm defines a cluster-head, the node used for connection to other clusters. It also defines a gateway node which provides switching (communication) between two or more cluster-heads.

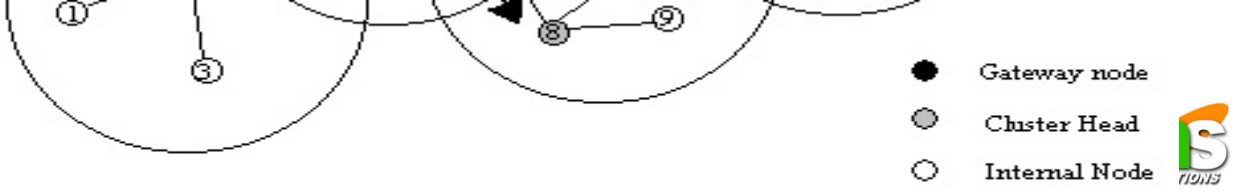
There will thus be three types of nodes—

(i) internal nodes in a cluster which transmit and receive the messages and packets through a cluster-head



er-head which dynamically
f ad-hoc hosts, monitors
ges to another cluster-head

n of messages and packets



The cluster structure leads to a higher performance of the routing protocol as compared to other protocols because it provides gateway switch-type traffic redirections and clusters provide an effective membership of nodes for connectivity.

CGSR works as follow:

- Periodically, every nodes sends a hello message containing its ID and a monotonically increasing sequence number
- Using these messages, every cluster-head maintains a table containing the IDs of nodes belonging to it and their most recent sequence numbers.
- Cluster-heads exchange these tables with each other through gateways; eventually, each node will have an entry in the affiliation table of each cluster-head. This entry shows the node's ID & cluster-head of that node.
- Each cluster-head and each gateway maintains a routing table with an entry for every cluster-head that shows the next gateway on the shortest path to that cluster head.

Disadvantages:

- The same disadvantage common to all hierarchal algorithms related to cluster formation and maintenance.