www.ijecs.in
International Journal Of Engineering And Computer Science ISSN: 2319-7242
Volume 6 Issue 1 Jan. 2017, Page No. 20007-20010
Index Copernicus Value (2015): 58.10, DOI: 10.18535/ijecs/v6i1.30

Finding the Routing Table by Dynamic Processing Method in Distance Vector Routing

***Jyotsana Pandey*[1], *Ashish*[2]**
[1]Research scholar, Invertis University,
Bareilly, Uttar Pradesh, India
jyotsana20pandey@gmail.com
[2]Research scholar, Invertis University,
Bareilly, Uttar Pradesh, India
a.saxena999@gmail.com

Abstract: ***Distance vector routing is a routing protocol for communication methods used in packet-switched networks. It makes use of distance to decide the best path for forwarding the packets. DVRP (Distance Vector Routing Protocol) uses routing hardware to find distances to all nodes within a network. DVRP method uses mainly two factors – distances and vectors. Distance is the number of count it needs to reach to the destination and vector is the trajectory of the message through the given nodes. DVRP uses the Bellman-Ford equation for calculating the paths and for updating the routing tables. This paper proposes an approach for calculation and updating of routing tables which is based on Dynamic processing.***

Keywords: Dynamic processing, Routing tables, DVRP, Bellman-Ford equation

1. Introduction

Routing protocols determine the best routes to transfer data from one node to another and specify how routers communicate between each other in order to complete this task. There are different classes of routing protocols, two of which are Exterior Gateway Protocol (EGP) and Interior Gateway Routing (IGR). A routing protocol can be dynamic or static, as well as distance-vector or link-state.
Routing protocols define the path of each packet from source to destination. To complete this task, routers use routing tables, which contain information about possible destinations in the network and the metrics (distance, cost, bandwidth, etc.) to these destinations. Routers have information regarding the neighbor routers around them. The degree of a router's network knowledge and awareness depends on the routing protocol it uses. At every change in the network, including link failure and link recovery, routing tables must be updated. The efficiency of these updates determines the efficiency of the routing protocols. There are two main types of routing protocols: static routing and dynamic routing. Static routing assumes that the network is fixed, meaning no nodes are added or removed and routing tables are therefore only manually updated. Dynamic or adaptive routing, more commonly used for internetworking, allows changes in the network topology by using routing tables that update with each network change. In this paper only dynamic routing protocols will be considered. Within the class of dynamic protocols, we can have Interior or Exterior Gateway Protocols.

Distance vector routing is a simple *routing protocol* used in packet-switched networks that utilizes distance to decide the *best packet* forwarding path. Distance is typically represented by the *hop* count. Distance vector routing protocols are efficient for small networks. However, they have poor *convergence* properties.

Distance-vector routing protocols use the Bellman–Ford algorithm, Ford–Fulkerson algorithm, or DUAL FSM to calculate paths. A distance-vector routing protocol requires that a router inform its neighbors of topology changes periodically.DVR also has less computational complexity and message overhead.

Dynamic programming (also known as **dynamic optimization**) is a method for solving a complex problem by breaking it down into a collection of simpler subproblems, solving each of those subproblems just once, and storing their solutions - ideally, using a memory-based data structure. The next time the same subproblem occurs, instead of recomputing its solution, one simply looks up the previously computed solution, thereby saving computation time at the expense of a (hopefully) modest expenditure in storage space. A dynamic programming algorithm will examine the previously solved subproblems and will combine their solutions to give the best solution for the given problem.

2. Literature Survey

Distance-vector protocols are based on calculating the direction and distance to any link in a network. "Direction" usually means the next hop address and the exit interface. "Distance" is a measure of the cost to reach a certain node. Each node maintains a vector (table) of minimum distance to every node. The cost of reaching a destination is calculated using various route metrics. Updates are performed periodically in a distance-vector protocol where all or part of a router's routing table is sent to all its neighbors. . Once a router has this information it is able to configure its own routing table to reflect the changes and then inform its neighbors of the changes.

Bellman-Ford Algorithm

- Define distances at each node *X*

◆ dx(y) = cost of least-cost *path* from *X* to *Y*

- Update distances based on neighbors

◆ dx(y) = min {c(x,v) + dv(y)} over all neighbors *V*

- *c(x,v)* = cost for direct link from *x* to *v*

◆ Node x maintains costs of direct links *c(x,v)*

- *Dx(y)* = estimate of least cost from *x* to *y*

◆ Node *x* maintains distance vector ***D**x = [Dx(y): y є N]*

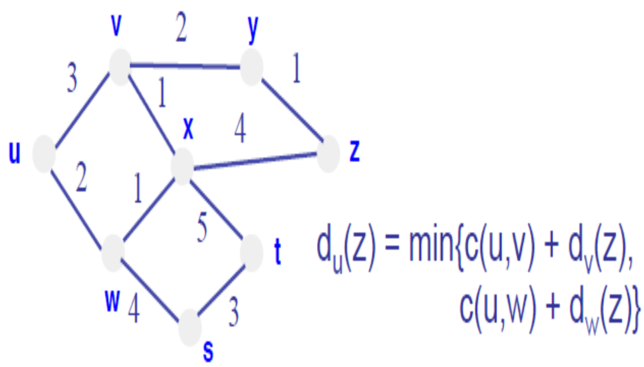
- Node *x* maintains its neighbors distance vectors

◆ For each neighbor *v*, *x* maintains ***D**v = [Dv(y): y є N]*

- Each node *v* periodically sends *Dv* to its neighbors

◆ And neighbors update their own distance vectors

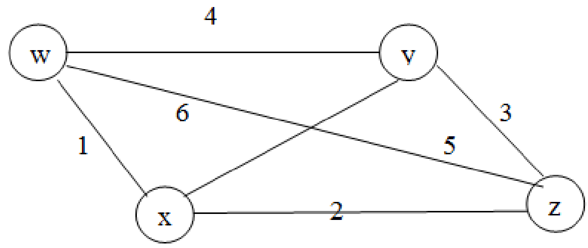
◆ *Dx(y) ← minv{c(x,v) + Dv(y)}* for each node *y* ε *N*



Above explained method is the traditional method to find out the routing table and its optimal solution used in Distance Vector Routing.

3. Proposed methodology

In this paper we are proposing the dynamic processing method for finding out the optimal path in the routing table in Distance Vector Routing.
The dynamic processing method works as follows-
In the first step, for the given network, a matrix form of table is formed depicting the path distances from one node to another.
This matrix is reduced row-vise and column-vise so as to get the reduced matrix for the actual matrix. From that reduced matrix, the value for the lower bound is found.
From the source node, matrix for all the possible paths are formed so as to get the path distance for visiting the next node from that previous node. Again the reduction of the matrix is performed on all the matrices formed for the next nodes.
(The process for reduction of the matrix- First of all the reduction is done row-vise and column-vise and then with previous two, the common value is also added into the lower bound value to get the new L value and the new reduced matrix). This procedure is followed until all the nodes have been visited. The final L value is the minimal possible value for traversing all the nodes in optimal condition.
For keeping the point of the dynamic processing method, the following problem is taken –



The above diagram is a given network of nodes w,x,y and z having path distances as mentioned.
Now we will propose the dynamic programming method for finding out the most optimal path for communication between the nodes.

This network in matrix form can be shown like:

Nodes	W	X	Y	Z
W	∞	1	4	5
X	1	∞	6	2
Y	4	6	∞	3
Z	5	2	3	∞

Now the method says to reduce this above formed matrix row-vise and column-vise so as to find out the lower bound in the matrix.

Nodes	W	X	Y	Z	
W	∞	1	4	5	-1
X	1	∞	6	2	-1
Y	4	6	∞	3	-3
Z	5	2	3	∞	-2

Reduced Matrix-

Nodes	W	X	Y	Z
W	∞	0	3	4
X	0	∞	5	1
Y	1	3	∞	0
Z	3	0	1	∞

Reducing again column-vise

Nodes	W	X	Y	Z
W	∞	0	3	4
X	0	∞	5	1
Y	1	3	∞	0
Z	3	0	1	∞
			-1	

Reduced Matrix-

Nodes	W	X	Y	Z
W	∞	0	2	4
X	0	∞	4	1
Y	1	3	∞	0
Z	3	0	0	∞

Lower Bound, L=1+1+3+2+1=8

L(W,X)

Nodes	W	X	Y	Z	
W	∞	∞	∞	∞	
X	∞	∞	4	1	-1
Y	1	∞	∞	0	
Z	3	∞	0	∞	

Common =0
L=8+1+0=9

Reduced matrix L(W,X)

Nodes	W	X	Y	Z
W	∞	∞	∞	∞
X	∞	∞	3	0
Y	1	∞	∞	0
Z	3	∞	0	∞

L(W,Y)

Nodes	W	X	Y	Z
W	∞	∞	∞	∞
X	0	∞	∞	1
Y	∞	3	∞	0
Z	3	0	∞	∞

Common=2
L=8+2=10

L(W,Z)

Nodes	W	X	Y	Z	
W	∞	∞	∞	∞	
X	0	∞	4	∞	
Y	1	3	∞	∞	-1
Z	∞	0	0	∞	

Common=4
L=8+4+1=13

Reduced Matrix L(W,Z)

Nodes	W	X	Y	Z
W	∞	∞	∞	∞
X	0	∞	4	∞
Y	0	2	∞	∞
Z	∞	0	0	∞

Now, since L value of L(W,X) is lowest ,we will follow the path through X as follows-

L(W,X,Y)

Nodes	W	X	Y	Z	
W	∞	∞	∞	∞	
X	∞	∞	∞	∞	
Y	∞	∞	∞	0	
Z	3	∞	∞	∞	-3

Common=3
L=9=3+3=15

Reduced Matrix L(W,X,Y)

Nodes	W	X	Y	Z
W	∞	∞	∞	∞
X	∞	∞	∞	∞
Y	∞	∞	∞	0
Z	0	∞	∞	∞

L(W,X,Z)

Nodes	W	X	Y	Z	
W	∞	∞	∞	∞	
X	∞	∞	∞	∞	
Y	1	∞	∞	∞	-1
Z	∞	∞	0	∞	

Common=0
L=9+0+1=10

Reduced Matrix L(W,X,Z)

Nodes	W	X	Y	Z
W	∞	∞	∞	∞
X	∞	∞	∞	∞
Y	0	∞	∞	∞
Z	∞	∞	0	∞

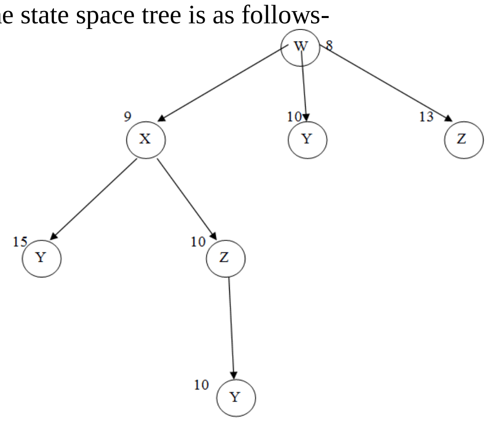
Now, again since L value of L(W,X,Z) < L(W,X,Y), we will procede with Z after W and X as follows :

L(W,X,Z,Y)

Nodes	W	X	Y	Z
W	∞	∞	∞	∞
X	∞	∞	∞	∞
Y	∞	∞	∞	∞
Z	∞	∞	∞	∞

Common=0
L=10+0=10

Visiting of the nodes is completed. Now the next step is to draw the state space tree for the depicting the final solution. The state space tree is as follows-



This state space tree shows the possibilities of the solutions and shows the finally followed optimal path =WXZY for traversing all the nodes covering minimum distance and minimum time.

4. Conclusion

We have seen that the method shown above ie dynamic processing method yields the optimal method for the packets to travel in the network in a very efficient way. This is a method different from the method followed frequently and traditionally for working with the routing table. This method proves to be efficient and optimal and also always finds the correct result for the most efficient path for visiting the nodes. Apart from this method, one can try more processes to find out the same which may prove to be even better than this.

References

[1] Elizabeth M. Royer, Chai-Keong Toh “A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks”, IEEE Personal Communications, April 1999.

[2] Tanenbaum Andrew S.: Computer Networks, Prentice Hall, Inc., 1996, ISBN 7-302-02410-3

[3] C.Jeyalakshmi,Dr.R.Sukumar, “An Experimental Study on Optimized Link State Routing Protocol for Underground Mines”, C Jeyalakshmi et al, Int.J.Computer Technology & Applications, Vol 3 (6), 1886-1893

[4] Royer Elizabeth M. “A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks”, IEEE Personal Communications, April 1999.

[5] Y.C. Hu, D.B. Johnson, and A. Perrig, “SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks”, Ad Hoc Networks Journal, 1 (2003):175-192.

[6] R.V. Boppana And S.P Konduru, "An Adaptive distance vector routing algorithm for mobile, ad hoc networks” in proceedings Of IEEE Ilifocom 2001,Anchorage,Alaska,April 2001,IEEE,vol.3.pp.1753-1762.

Author Profile

Jyotsana Pandey received the B.Tech. degree in Computer Science Engineering from Sri Ram Murti Smarak Women’s College of Engg and Technology in 2015 and is pursuing M.Tech. degree in Computer Science Engineering from Invertis University, respectively. She has published articles and working on networking, mobile computing, genetic algorithms.

Ashish received the B.Tech. degree in Computer Science Engineering from Future Institute of Engineering and Technology in 2014 and is pursuing M.Tech. degree in Computer Science Engineering from Invertis University, respectively. He has published articles and working on mobile computing, genetic algorithms, artificial intelligence, cryptography.