Computer Network Routing Using Neural Networks

Dr. Abdulkareem Y. Abdalla, Prof. Dr. Turki Y. Abdalla and Khloud Ahmad Nasar
Computer Science Dept. Computer Engineering Dept. Computer Science Dept.
University of Basrah, Basrah, Iraq

Abstract

The neural networks present widely conducing to solve the routing problem in the computer networks. In this paper, two methods are proposed to solve this problem. A feedforward neural network is included at each node to make local decision, or in central node to determine the complete path between pair of nodes (source, destination). For central routing decision, monitor is included as part of routing system to provide the necessary information to the neural network router. The proposed methods are applied for typical examples of computer networks. The neural networks are trained. Results of the testing proof on their good performance.

1. Introduction

In a communication network information is transferred from one node to another as data packets. Packet routing is a process of sending a packet from its source node (s) to its destination node (d). On its way, the packet spends some time waiting in the queues of intermediate nodes while they are busy processing the packets that came earlier. Thus the delivery time of the packet, defined as the time it takes for the packet to reach its destination, depends mainly on the total time it has to spend in the queues of the intermediate nodes. Normally, there are multiple routes that packet could take, which means that the choice of the route is crucial to the delivery time of the packet for any (source, destination) pair (S. Kumar and R. Miikkulainen 1998) .

Successful operation of data communication network is critically dependent on the provision of an adequate routing algorithm. Routing algorithms are methods for finding the best way from a node s to another node d. This may be via a large number of other nodes or it may be in the next subnetwork. On a small, simple network the problem is almost trivial, statically allocating routes and defining them by hand, but when dealing with a huge internetwork such as the Internet this is not possible. It heavily interconnected network has many routes from one node to another, and these routes span many different types of link with different bandwidth and latency characteristics. Calculating the best route through such a complex system is computationally intractable and impossible to do by hand (D. Davies, D. Barber, W. Price and C. Solomonides 1979, J. Malrand 1991, W. Newton 2001) .

If part of network becomes over-filled with packets it can become impossible for packets to move. The queues into which they should be accepted are always full. This is called congestion. Routing algorithms strongly interact with congestion (D. Davies, D. Barber, W. Price and C. Solomonides 1979, G. Caro and M. Dorigo 1997).

2. Routing Using Neural Networks

The routing problem defined by determining the optimal route for a packet from source node to destination node. The routing decision must be made under the network current conditions such that traffic load, congestion and node or link failures.

Using the conventional algorithms and particular mathematical programming methods to solve this problem is not recommended for practical purposes. Because, their calculations may need long time, that which is caused to slowly response to very frequently changing of status of the networks.

The parallel, distributed processing structure of the neural networks and their ability to learn are justified to use it as available structure for solving the routing problem. As a result, neural networks are considered to solve such kind of optimization problem. Optimal route is obtained by routing decision which is based on observed delay function. Also, for local decision routing, neural network at each node of computer network use just local information to decide to which neighbor node should be send the packet in order to reach its destination quickly (W. Newton 2002, C. Tseng and M. Garzon 1996, J. Menke 1999).

Considerable research has been devoted toward using neural networks to solve the routing problem in the computer networks. Paper of C.Tseng and M.Garzon (1996) describes a hybrid distributed adaptive neural router. By combining a simple neural network (perceptron) for the routing decision, a rule based network manager for updating and learning process, and a queuing module. Paper of J. Menke (1999) demonstrates the ability of a simple single-layer neural network to find the shortest paths, when applied in a distributed fashion.

S.Pierre, H.Said and W.Probst (2001) present a new approach based on the Hopfield model. The proposed method is based on a network representation enabling to match each network configuration with a Hopfield neural network in order to find the best path between any node pair by minimizing an energy function. V. krishnamoorthy, Y. Pan and Y. Zhang (1997) introduce study for a qualitative and quantitative analysis of multistage interconnection network routing using Hopfield neural network.

3. Proposed Methods

This paper presents proposed approaches based on using neural network to solve the routing problem in a context of computer networks. The neural network of these approaches may be built as a routing system, or as a part of integral routing system. It is responsible on making the routing decision.

The computer networks which are considered in this paper modeled as graphs. Two examples, the first is a 9-node mesh computer network (CN1) shown in Figure (1). While the second is a random computer network (CN2) shown in Figure (2). The imposing values of cost of the links (packet delay) and queue length of the nodes for these two computer networks are shown in Tables (1) and (2), respectively. The values of the queue lengths in these Tables are located under changing continuously, when the nodes send or receive the packets.

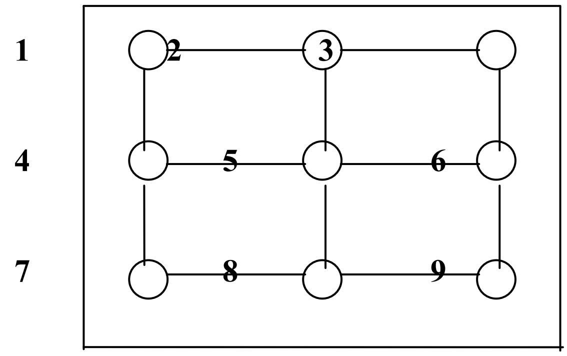

Figure (1) computer network (CN1)

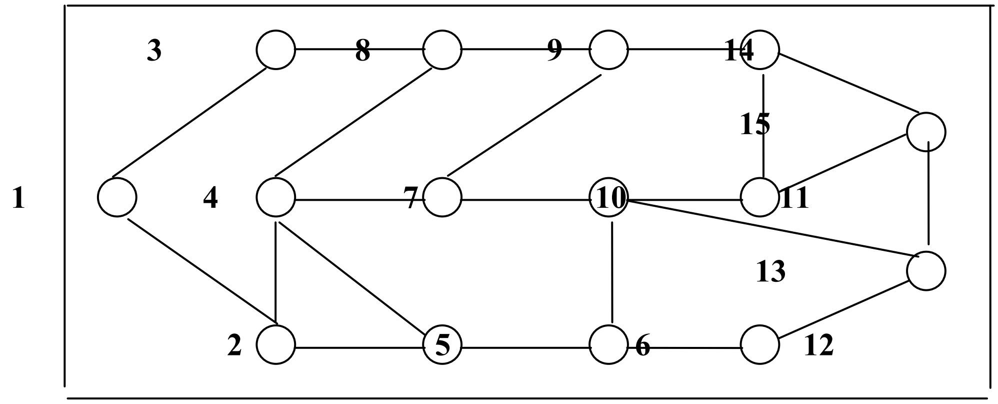

Figure (2) computer network (CN2)

Table (1) link costs and queue lengths of the computer network (CN1)

Links		**Nodes**	
link	**cost (packet delay) (in second)**	**node**	**queue length (in packet)**
1—2	10	1	6
1—4	2	2	5
2—3	3	3	8
2— 5	15	4	4
3— 6	8	5	7
4— 5	2	6	7
4— 7	1.6	7	4
5— 6	1.2	8	6
5— 8	2.1	9	6
6— 9	1.5		
7— 8	3		
8— 9	1		

Table (2) link costs and queue lengths of the computer network (CN2)

Links		Nodes	
link	cost (packet delay) (in second)	node	queue length (in packet)
1—2	6.312	1	5
1—3	6.312	2	5
2—4	1.544	3	7
2— 5	6.312	4	7
3— 8	6.312	5	4
4— 5	3.352	6	8
4—7	6.312	7	8
4— 8	3.352	8	5
5— 6	12.624	9	6
6— 10	3.352	10	5
6— 12	6.312	11	4
7— 9	3.152	12	6
7— 10	3.152	13	3
8—9	6.312	14	4
9— 14	12.624	15	6
10— 11	3.152		
10— 13	3.152		
11— 14	3.152		
11— 15	3.152		
12— 13	6.312		
13— 15	6.312		
14— 15	6.312		

3.1. Local Routing

To solve the routing problem locally, a feedforward neural network (NN1) is developed. his neural network has 2n + 1 inputs, they are destination, n of link time costs and n of queue time costs, where n is a number that represent the maximum number of neighbor nodes in the computer network, and one output which is the best neighbor node, with n units in one hidden layer, as shown in Figure (3).

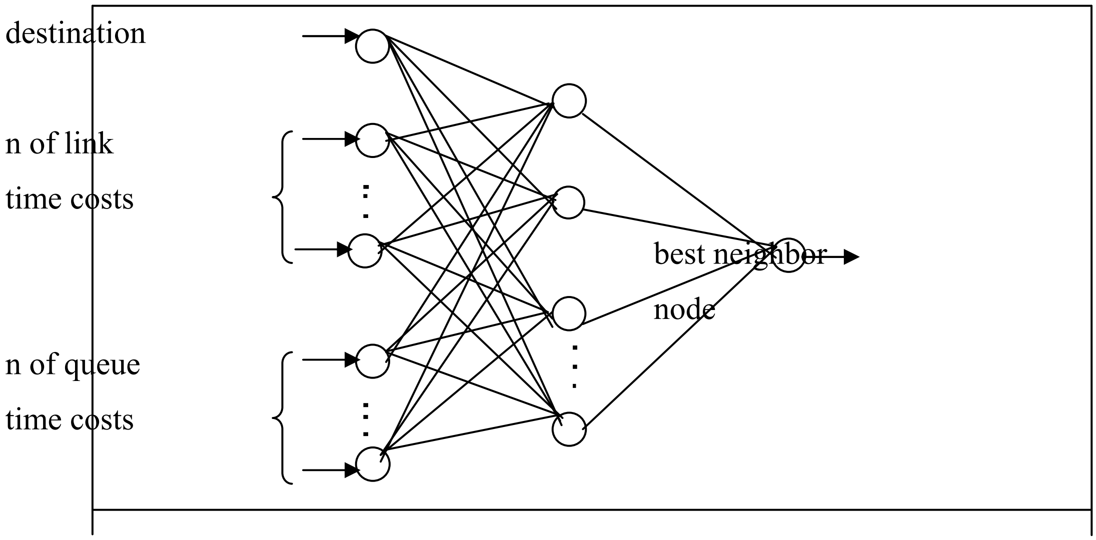

Figure (3) the feedforward neural network (NN1)

The computer network includes such neural network at each node. The node has information on time costs of links to send packet to its neighbor nodes. And also, it receives information from these neighbor nodes about time costs of waiting the packet at queue. The time delay of packet at queue of node can be obtained by Little's law of queueing theory, as follows.

$$L_q = P * T_q \qquad (1)$$

where L_q is average length of queue (in packet), p is average number of arriving packets (in packet / second) of the node, and T_q is average time of packet in queue (in second). When the actual number of neighbor nodes is less than n, the neural network receives very large values for link time costs and queue time costs for that excessive neighbor nodes. As a result the neural network determines a best neighbor node, for that, the packet at source node can be arrived to destination node on shortest path with shortest waiting time at queues. For example, in the computer network (CN2), if the node 8 has packet to be sent to destination node 12, then the neural network (NN1) at node 8 receive destination 12, time costs of links 8─ 3, 8─ 4, and 8─ 9 and queue time costs of neighbor nodes (3, 4, 9). It determines the best neighbor node to reach node 12 quickly. The same process is repeated at each node successively till the destination node.

3.2. Central Routing

In this method a feedforward neural network (NN2) is used as a part of routing system which also has monitor part. This neural network has 2n + 2 inputs, they are source, destination, n of link time costs and n of queue lengths, where n is a number that represent the maximum number of neighbor nodes in the computer network , and one output which is the best neighbor node. With n units in one hidden layer, Figure (4) view this neural network router.

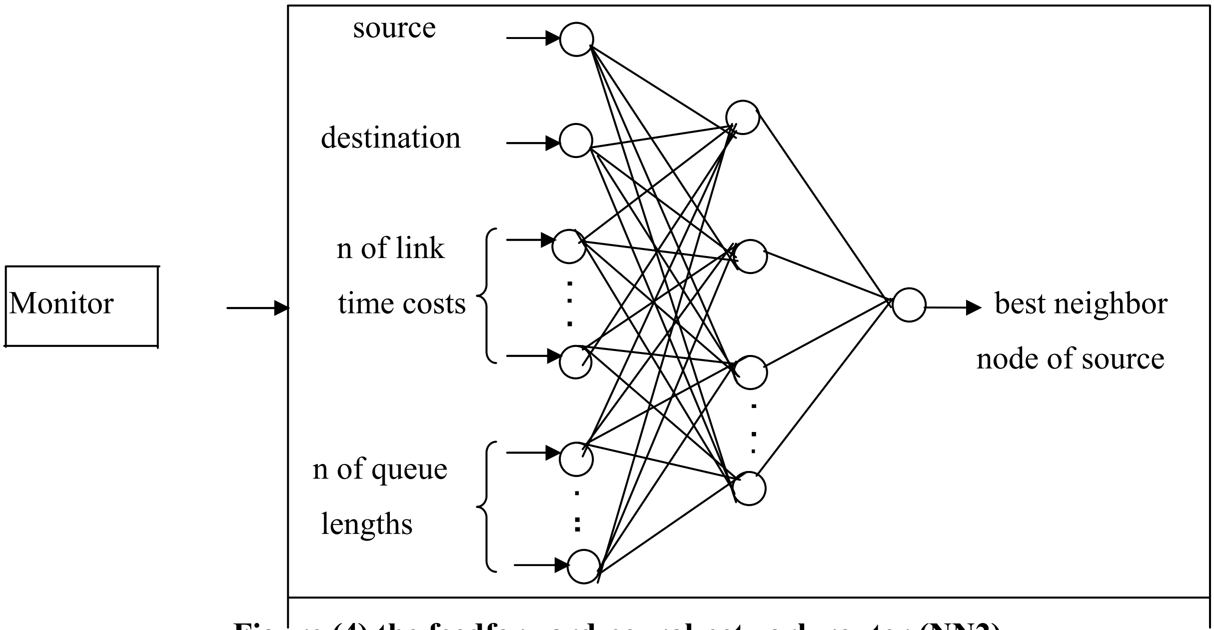

Figure (4) the feedforward neural network router (NN2)

Such routing system must be included at a central node of the computer network. Monitor part takes readings on time costs of all links and queue lengths of all nodes to get necessary information for the neural network router. The neural network receives its inputs from the monitor, and decides the best neighbor node of the source node. When the actual number of neighbor nodes is less than n, it receives very large values for link time costs and queue lengths for that excessive neighbor nodes. For that, the routing system can make central decision of determining best path to send packet form source node to destination node on many steps. At beginning, the best neighbor node of the source node is determined. Then, this neighbor node becomes a source node and its neighbor node is determined, and so on, until the destination node is reached. This routing system finds shortest path with shortest lengths of queues. For example, in the computer network (CN1), for determining the best path to send packet from node 7 to node 3, the neural network router receives source 7, destination 3, and link time costs and queue lengths of neighbor nodes (4, 8) of node 7 from the monitor, then decides node 4 is the best neighbor node . After that, it receives source 4, destination 3, and link time costs and queue lengths of neighbor nodes (1, 5) of node 4 from the monitor and decides node 1 is the best neighbor node. And then, it decide node 2 as the best neighbor node of source 1, and node 3 (the destination) is the best neighbor node of source 2.

3.3. Simulation Result

In order to test the proposed methods, they are applied for two examples of computer networks (CN1, CN2) shown in Figures (1) and (2). The implementation of the simulation has been realized using C++ programming language.

The activation function used of these feedforward neural networks is a sigmoid function as in equation:

$$F(x) = 1 / (1 + e^{-\lambda x}) \qquad (2)$$

where λ is a size of step, $\lambda \geq 0$ (is taken between 0 and 1). Where real numbers are used to represent their inputs and outputs.

Table (3) describe the number of inputs, outputs and hidden units of these neural networks. At this Table n = 8 is assumed as the maximum number of neighbor nodes for the computer networks (CN1, CN2).

For two feedforward neural networks, the training sets and the test sets are prepared. The training set consists of the inputs of the neural network, for example source node, destination node, link costs and queue lengths, and the desired outputs, such as best neighbor node. In training stage, random values between – 0.5 and 0.5 are used as initial weights of connections of the neural networks. The neural networks are trained on 30 training sets (are taken from Tables (1) and (2)) with the backpropagation algorithm for each computer networks (CN1, CN2). For any set, the training is continued until the mean squared error (MSE) becomes acceptable. The neural networks are trained on all possible routing decisions. In this stage, values of learning rate (η) and momentum rate (α) are selected by trial and error, which are given in Table (4). The obtained results are shown in Figures (5)-(8). These Figures view mean squared error versus number of epochs, where this error is decreased when number of epochs is increased.

Then, testing the performance of these neural networks is performed on the trained sets, and on other test sets which are unseen before. Tables (5)-(8) exposes some results of the testing of computer networks (CN1, CN2).

Through the testing of the proposed feedforward neural networks, the results show their good performance, as observed in Tables (9) and (10) for computer networks (CN1, CN2), respectively. These Tables describe the success rates to test trained sets and test sets.

Table (3) the number of inputs, outputs and hidden unit of the proposed feedforward neural networks

Neural network	Number of inputs	Number of outputs **output**	Number of hidden units
NN1	2n + 1	1	n
NN2	2n + 2	1	n

Table (4) learning rates and momentum rates of the proposed feedforward neural networks

Neural network	Learning rate (η)	Momentum rate (α) ((**output**
NN1	0.5	0.5
NN2	0.5	0.5

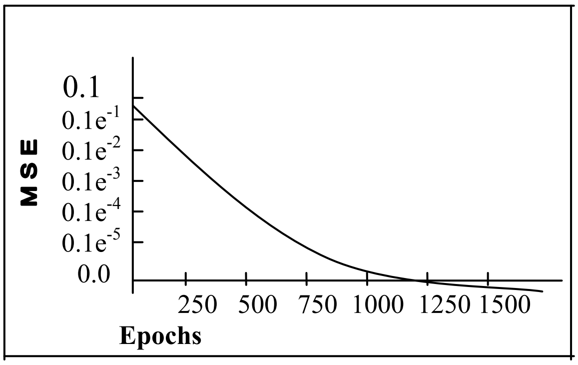

Figure (5) error versus number of epochs of the feedforward neural network (NN1) for the computer network (CN1)

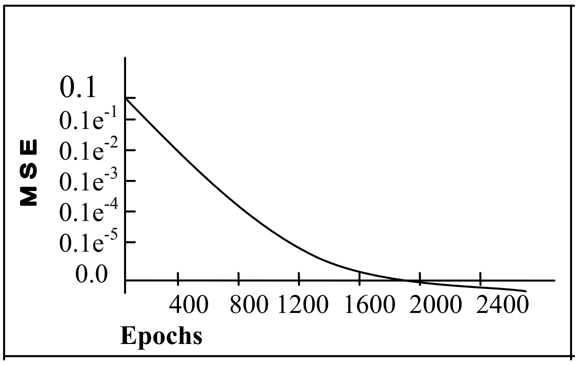

Figure (6) error versus number of epochs of the feedforward neural network (NN1) for the computer network (CN2)

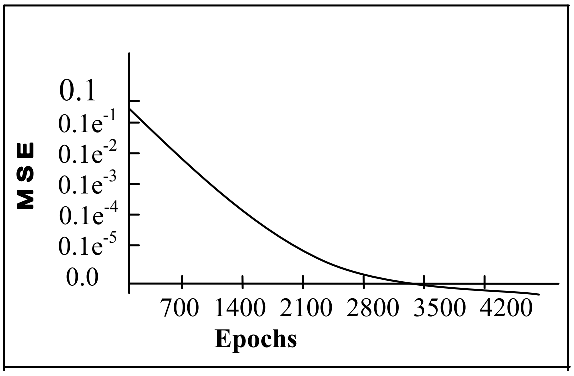

Figure (7) error versus number of epochs of the feedforward neural network (NN2) for the computer network (CN1)

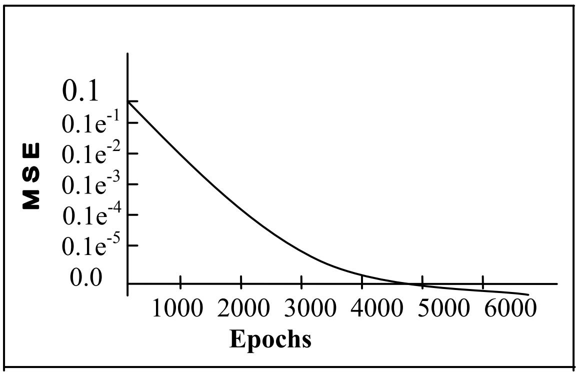

Figure (8) error versus number of epochs of the feedforward neural network (NN2) for the computer network (CN2)

Table (5) some of the test results of the feedforward neural network (NN1) for the computer network (CN1)

Source	Destination	Best path	Outputs of the local neural networks (NN1)
1	3	(1 2 3)	(1 2 3)
2	9	(2 3 6 9)	(2 3 6 9)
3	7	(3 2 1 4 7)	(3 2 1 4 7)
7	3	(7 4 1 2 3)	(7 4 1 2 3)
9	7	(9 8 7)	(9 8 7)
6	1	(6 5 4 1)	(6 5 8 7 4 1)

Table (6) some of the test results of the feedforward neural network (NN1) for the computer network (CN2)

Source	Destination	Best path	Outputs of the local neural networks (NN1)
9	10	(9 7 10)	(9 7 10)
11	4	(11 14 9 7 4)	(11 14 9 7 4)
1	8	(1 2 4 8)	(1 2 4 8)
10	9	(10 7 9)	(10 7 9)
15	8	(15 11 10 7 9 8)	(15 11 10 7 9 8)
12	14	(12 13 10 11 14)	(12 13 10 13 15 14)

Table (7) some of the test results of the feedforward neural network (NN2) for the computer network (CN1)

Source	Destination	Best path	Outputs of the monitor-NN2 system
6	4	(6 5 4)	(5 4)
2	9	(2 3 6 9)	(3 6 9)
3	7	(3 2 1 4 7)	(2 1 4 7)
4	6	(4 5 6)	(5 6)
1	8	(1 4 7 8)	(4 7 8)
1	6	(1 4 5 6)	(4 7 8 9 6)

Table (8) some of the test results of the feedforward neural network (NN2) for the computer network (CN2)

Source	Destination	Best path	Outputs of the monitor_ NN2 system
3	10	(3 8 4 7 10)	(8 4 7 10)
11	4	(11 14 9 7 4)	(14 9 7 4)
1	8	(1 2 4 8)	(2 4 8)
8	1	(8 4 2 1)	(4 2 1)
6	8	(6 10 7 9 8)	(10 7 9 8)
5	13	(5 4 7 10 13)	(4 8 9 7 10 13)

Table (9) the success rates of testing the proposed feedforward neural networks for the computer network (CN1)

Neural network	Success rate of test on trained sets	Success rate of Test on other sets
NN1	% 100	% 95
NN2	% 100	% 95

Table (10) the success rates of testing the proposed feedforward neural networks for the computer network (CN2)

Neural network	Success rate of test on trained sets	Success rate of Test on other sets
NN1	% 100	% 95
NN2	% 100	% 95

4. Conclusions

This paper introduces the description of two proposed methods of using neural networks for solving the routing problem in computer networks. Both, local or central routing decision can be made, when the feedforward neural networks are used. The neural network router require inputs which may be accounted to maximum number of the neighbor nodes. The avoiding of congestion node by node is made efficiently. The number of inputs of the proposed feedforward neural networks which determine the best neighbor node has no related with size of the computer network. The results obtained during training and testing stages of the proposed neural networks show their good performance, in making routing decision, when they are applied for two different examples of computer network structures.

References

C. Tseng and M. Garzon, " Hybrid Distributed Adaptive Neural Router ", Texas University, USA, 1996, E-mail: tseng@cse.uta.edu.

D. Davies, D. Barber, W. Price and C. Solomonides," Computer Networks and Their Protocols "، J. Wiley and Sons Ltd, New York، 1979.

G. Caro and M. Dorigo, " AntNet: A Mobile Agents Approach to Adaptive Routing ", Technical Report, Iridia University, No. 12, 1997.

J. Malrand, " Communication Networks: A First Course ", R. D. Irwin and Akson Associates, Inc, 1991.

J. Menke, " Distributed Neural Network Routing ", Proceeding of the ANNIE, 1999.

S. Kumar and R. Miikkulainen, " Confidence-Based Q-Routing: An On-Line Adaptive Network Routing Algorithm ", Proceedings of the Artificial Neural Networks in Engineering Conference, Texas University, Austin, 1998.

S. Pierre, H. Said and W. Probst, " An Artificial Neural Network Approach for Routing in Distributed Computer Networks", Elsevier Science Ltd, Engineering Applications of Artificial Intelligence, No. 14, 2001.

V. Krishnamoorthy, Y. Pan and Y. Zhang,"A Neural Network Approach for Multistage Interconnection Network Routing ", Georgia State University, USA, 1997, E-mail: pan@cs.gsu.edu.

W. Newton, " A Neural Network Algorithm for Internetwork Routing ", Report in Software Engineering, for Degree of Bachelor, 2002.