A Fuzzy System for Adaptive Network Routing

A. Pasupuleti[*], *A.V. Mathew*[*], *N. Shenoy*[**] *and S. A. Dianat*[*]
Rochester Institute of Technology
Rochester, NY 14623, USA
E-mail: axp1014@rit.edu

Abstract

In this paper we propose an adaptive routing algorithm in which the link cost are dynamically assigned using a fuzzy system. The traffic in the network is re-routed to nodes, which are less congested, or have spare capacity. Based on a set of fuzzy rules, link cost is dynamically assigned depending upon the present condition of the network. Distance vector algorithm, which is one of the shortest path routing algorithms is used to build the routing tables at each node in the network. The proposed fuzzy system determines the goodness of a link given the present congestion situation measured via the delays experienced in the network and the offered load on the network. Delay in the links, was estimated by the time taken for the test packets to travel from the node to its neighbors. The information collected by the test packets and the number of packets waiting in the queue, are the two inputs to the fuzzy system. The output of the fuzzy system is cost of the link for a certain interval. This algorithm was applied on a simulated NSFNET, the USA backbone as well as to another test network with a different topology. Robustness and optimality of the algorithm was tested by simulating various types of load patterns and by comparing with some of the existing algorithms. The proposed fuzzy based algorithm always gave optimal performance under varying load conditions and topologies.

Keywords: Network routing, link cost, shortest path routing, Fuzzy logic

1. Introduction

With an ever-increasing demand for good communication network services, network control techniques play a vital role in providing reliable communication systems and to use the network resources efficiently. Routing is one of the major components of network control techniques that determine the overall network performance in terms of quality and quantity of the delivered service. The act of moving information from the source to the destination node is called routing. This process involves the distributed activity of building and using routing tables, which tell the incoming data packet which outgoing link to use to continue their travel towards the destination node. A good routing algorithm ensures the coordination between all the nodes, and should adapt to link and node failures and redirects traffic over the less congested routes of the networks. The two main components of routing are path determination and transportation of information from the source to destination. The effect of good routing is to increase throughput for the same value of average delay per packet under high offered load conditions and to decrease average delay per packet under low and moderate load condition. One of the most popular and widely used routing algorithms in today's communication networks is the shortest path routing algorithm. These algorithms are classified under quasi-static algorithms where the link cost remains constant for a short period of time. In this algorithm, each link is assigned a cost based on the particular routing metric used (e.g. delay, queue length, hops and bandwidth), which may be different in each direction. In such algorithms, each node attempts to route packets to their destination over paths with minimum link costs and updates the link costs periodically to adapt to traffic and topological changes. Care should be taken while the link costs are dynamically assigned because, a strong feedback effect could be introduced between the routing policy and the

---

[*] Department of Electrical Engineering
[**] Department of Information Technology

traffic pattern that could result in undesirable oscillations [1]. The shortest path algorithms can be broadly classified into distance-vector and link-state categories, based on the amount information stored at the decision making place.

1.1 Distance-vector Algorithm

The distance vector otherwise called Bellman-Ford algorithm, is based on the principles of dynamic programming and works in an iterative, distributed and asynchronous way. The amount of information stored at a node is comparatively lesser than that of link-state algorithm. The routing table at every node in the network in distance vector algorithm consists of the destination node, estimated distance also known as the cost and the next hop node. The shortest path calculation using Bellman-Ford algorithm has the form of

$$D_i = \min_{j} [\, d_{i\,j} + D_j \,] \tag{1}$$

where, D_i is the minimum cost of the node j of the node i to the destination and $d_{i\,j}$ is the cost of the link (i ,j). Each node i executes periodically this iteration with minimum taken over all the neighbors [2].
In our simulations we have used Bellman-Ford algorithm for route computation where the link costs are updated every 1 sec.

1.2 Routing Metrics

As described above, in shortest path routing algorithms link costs can be assigned dynamically and statically. The simplest of all the metrics is the hop count. Delay, bandwidth of the link, Queue size and propagation delay are some of the other metrics The selection of the routing metrics substantially affects the performance parameters of the routing algorithms, namely throughput and average packet delay. The delay obtained through time-stamps in the packets gives a fairly good estimate of the congestion in the network but it is very difficult to distinguish between the delays caused because of selection of longer routes by the packets or because of the congestion in the network. Queue size defines the buffer capacity at the node and it refers to the number of packets waiting to get processed. In our simulations we have allocated buffer capacity for each link. Also, the length of the output queue is one of the many factors that affect the packet's delay. Transmission capacity of the link is also a major routing metric. The selection of this parameter often depends upon the need and application of the end user. For faster transmission of data a satellite link could be preferred to a terrestrial link.
The first ARPANET routing algorithm used "estimated delay" alone as the routing metric in SPF algorithm, where each node exchanged current delay estimates to the destination node with its neighbors [3]. The affects of delay based routing were discussed in [4]. Under low and moderate load conditions, the usage of Queue size metric is shown [5] to perform well. It was able to divert the traffic from a congested link to a link that had spare capacity efficiently.

Optimal routing algorithms have a network wide perspective and their objective is to optimize a function of all individual link flows. It assumes that the main statistical characteristics of the traffic are known and not time varying [6]. One such algorithm is the Daemon routing algorithm, which is an adaptive optimal routing algorithm whose link cost is a composite metric consisting of weighted average of queue length and delay. Depending upon the load conditions, there is a set of optimal weights associated with queue length and delay, which set the empirical, bound on the achievable performance [7].
In this work, we propose to develop a fuzzy system based routing algorithm, which is valid under all load conditions. This scheme achieves performance close or equal to that achievable by using a set of optimally weighted metrics for a particular load conditions. By doing we maximize the throughput of the network and achieve high utilization.

2. Fuzzy system

Knowledge-based systems such as, Fuzzy logic have been successfully implemented in multifarious applications where the human expertise and dealing with uncertainty play a vital role in decision making process [8]. Fuzzy logic avoids arbitrary rigid boundaries by taking into account the continuous character of imprecise information. A fuzzy system is characterized by the inference system that contains the rule base for the system, input membership functions that are used for the fuzzification of the input variables and de-fuzzification of the output variables.

Fuzzification is a process where crisp input values are transformed into membership values of the fuzzy sets. After the process of fuzzification, the inference engine calculates the fuzzy output using fuzzy rules which are linguistic in the form of if then rules. De-fuzzification is a mathematical process used to convert the fuzzy output to a crisp value. A good fuzzy system is obtained when the rules and the membership functions are tuned to the application.

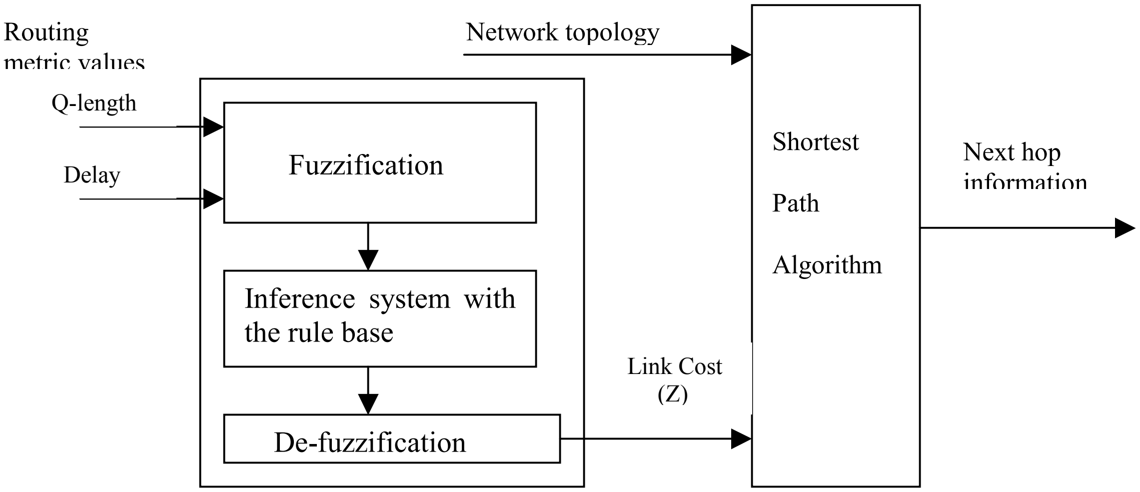

Figure (1) Fuzzy link cost in network routing

The above figure depicts the block diagram of the proposed fuzzy system. The routing metrics, Q-length and the delay are measured at each node of the network for each link and these values are given as input parameters to the fuzzy system. Using the link costs computed by the fuzzy system, the shortest path algorithm updates the routing tables at all the nodes in the network.

In the proposed fuzzy system, Mamdani minimum inference method [9] was used as the fuzzy inference method, where the 'and' operation was set to minimum and de-fuzzification was carried out using centroid defuzzifer. Mamdani's inference system can be mathematically written as,

$$\max(\min(\mu, \mu_{\overline{W}}(Z))) \quad \text{For all z} \tag{2}$$

where, $\mu_{\overline{W}}(Z)$ is the output membership function and μ is the combined membership in the rule antecedent.

2.1 Membership functions

Triangular membership functions were used for the linguistic variables that represent Q-length, delay and link cost. The triangular membership function is specified by a, b and c as shown in figure 2.

$$triangle(x,a,b,c) = \left\{ \frac{(x-a)}{(b-a)}, \frac{(c-x)}{(c-b)} \right\} \tag{3}$$

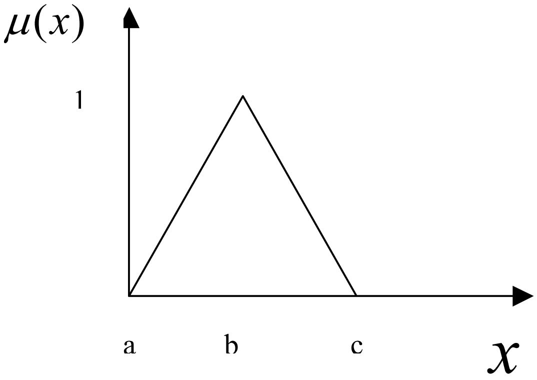

Figure 2: A fuzzy membership function

Q-size membership function. The range of the membership function is 1.0 to 2.2.
The various fuzzy sets used are Very Small: triangle (x,[0.7 1.02 1.033]);Small: triangle (x,[1.02 1.033 1.05]); Medium: triangle (x, [1.033 1.05 1.1]); Large: triangle (x, [1.05 1.1 2]); Very large: triangle (x, [1.1 2 2.19])

Delay membership function: The range of the membership function is 0 to .24. The various fuzzy sets used are
Very Small: triangle (x, [-0.058 -4.337e-019 0.05]); Small: triangle (x, [-4.337e-019 0.05 0.1]); Medium: triangle (x, [0.05 0.1 0.15]); Large: triangle (x, [0.15 0.2 0.24]); Very large: triangle (x, [0.2 0.24 0.29])

Link cost membership function: The range of the membership function is 1to 1.5. The various fuzzy sets used are
Very Small: triangle (x, [0.8 1 1.05]); Small: triangle (x, [1 1.05 1.25]]); Medium: triangle (x, [1.1 1.25 1.35]);
Large: triangle (x, [1.25 1.35 1.5]); Very large: triangle (x, [1.4 1.5 1.625])

Table 1 illustrates the rule base used in the fuzzy inference system. Expert knowledge was used in developing the rule base. While the delay and Q-size are represented in the X and Y-axis's respectively, the link cost fills the table.

Delay / Q-size	Very Small	Small	Medium	Large	Very large
Very Small	Very Small	Very Small	Small	Small	Medium
Small	Very Small	Small	Small	Medium	Medium
Medium	Small	Small	Medium	Medium	Large
Large	Small	Medium	Medium	Large	Large
Very large	Very large	Very large	Very large	Very large	Very large

Table 1: Rule Base for fuzzy based approach for adaptive traffic routing

3. System Setup

In this paper we have used two network topologies as described in [7] and [4] to compare the performance of the proposed fuzzy metric to the existing routing metrics. The focus is on connectionless datagram networks operating on irregular topology. In connectionless services, there are no provisions for flow control and admission control mechanisms. In such systems, as there is no handshaking between the source node and the destination node, the source node does not know whether the destination node received the packet. Examples for such systems are Internet phone, videoconference and streaming multimedia.

The two network topologies are shown in figure 3 and figure 4. Figure 3 consists of 14 nodes that act as forwarding as well as processing nodes and has 21 bi-directional links, which is the NSFNET, the USA backbone. Figure 4 consists of 8 nodes and 12 bi-directional links. The bandwidth of each link in both the networks was set at 1.5 MBPS and their corresponding propagation delays were set to 150 ms. Each and every link in both the network had an associated buffer space otherwise called queue length at the nodes that it connects. The maximum queue size was set to 50 packets.

Distributed, iterative Bellman-Ford algorithm was used to compute the routes between the source destination pairs, where the update interval time was set to 1s. The time length of simulations was set to 30s.

The two routing parameters used for calculating present link cost are Queue size and delay. Queue based parameter is calculated by using the formula

$$\frac{1}{Q_c - Q_{size}} \tag{4}$$

where, Q_c represents the maximum buffer capacity at the node and Q_{size} represents the current queue size at the measured time interval. Small test packets were used to compute the delay experienced by the data packets due to congestion in the network. Each node in the network sent out these test packets to their neighboring nodes at regular intervals of time and the delay experienced by these packets is averaged over the update interval time. The test packets had the same priority as the data packets; hence they experience the same wait in the queue as well as the transmission delay as experienced by the data packets. The typical size of test packet was 100 bytes and the time interval between the launch of two successive test packets was 0.05s. All the simulations were carried out on Network Simulator [10].

4. Results and Analysis

The proposed metric was tested on various load conditions as well as on different topologies. As stated before, the two performance measures used to evaluate the proposed technique were throughput and average packet delay. Some of the existing metrics used for comparison are the hop count, delay, Q-length and various combinations of Q-length and delay.

In the simulations, data packets were generated at a constant bit rate with fixed interval time of 0.005s and varying packet size for each simulation. The load on the network increases as number of Source-Destination pair's increase. The load on the network refers to the maximum amount of data in Mega Bits that can travel in the network between any source destination pairs for the entire simulation. The numbers of Source's were increased from 5 to 8 in network 1 and from 3 to 5 in network 2. As the time taken for a larger packet to travel across a link is more than that of smaller packet, in our simulations for every set of Source - Destination pair, the packet sizes were increased form 800 bytes to 1000 bytes so as to study the affect of packet delay on routing.

Assuming that the links are not broken, hop count remains a static metric. The routing tables, which were created using hop count as a metric, were used for the entire simulation irrespective of the changes in the load conditions. Hence the throughput obtained was very low under moderate and high load conditions. Delay based routing was also not very encouraging. Under low loads, typically 25%, delay based performed very poorly when compared to the other routing metrics, as illustrated in table 2.

Generalizing from the simulation results obtained, the throughput of the network increased when delay was used as a metric to Queue size as a metric. Under low and moderate load conditions Queue size was able to divert the traffic from a congested link to a link, which had spare capacity efficiently. But with the increasing load on the network, we observed that the throughput obtained because of a composite metric of queue size and delay, was better. Also, for a given number of source-destination pairs, with increasing packet size, the composite metric's throughput was higher. This is because the composite metric takes the global as well as the local estimation of load and congestion into effect before making a decision. The weights for the composite metrics were obtained by trying out various combinations. Among all the combinations we have tried 0.5, 0.5 and 0.8, 0.2 for queue size and delay respectively, gave better performance.

Our simulations revealed a great improvement in performance measures with the proposed fuzzy system. Under low load conditions, the fuzzy metric gave close to maximum throughput. This can be illustrated from the simulation results as shown in table 2. Tables 3 and 4 illustrate the throughput obtained for different metrics for network 1 and 2 respectively. It can be very easily observed that throughput obtained using fuzzy metric matched the best composite metric under moderate and high loads.

Tables 5 and 6 illustrate the average packet delay for different metrics for network 1 and 2 respectively. The average packet delay obtained using fuzzy metric is lesser than the average packet delay obtained using the composite metric under low load applications. Under moderate and high loads the average delay experienced by a packet using fuzzy routing almost equals the delay experienced by a packet using composite routing. Thus, the above observations indicate that the proposed fuzzy system provides optimal results under varying load conditions and is independent of the topology chosen.

5. Conclusions and future work

Routing in communication networks using a fuzzy system was addressed in this paper. The proposed fuzzy system showed better system performance and utilization of the network resources when compared to other metrics under various load conditions. As the load on the network increased the weights associated with the composite metric had to be changed so as to attain better performance whereas the fuzzy system was able to match the best performance of the composite metric under different loads and topologies. Robustness and its topology independent high performance are the two main characteristics of the fuzzy system, which give it an edge over the existing routing metrics. Research is in progress in fine-tuning the fuzzy system. Also the network resources are best utilized when multiple paths are used to send the data form the source to destination. A study is being carried out in developing a fuzzy based multi-path routing algorithm.

6. References

[1] Zheng Wang and Jon crowcroft. "Analysis of shortest-path routing algorithms in a dynamic network environment." Computer communication review, ACM SIGCOMM.
[2] Christian Huitema. Routing in the Internet, Prentice Hall, 2000.
[3] John M. Mcquillan and Ira Richer. "The new routing algorithm for the ARPANET." IEEE transactions on communications, VOL. COM-28, NO.5, May 1980.
[4] David W. Glazer and Carl Tropper. "A new metric for dynamic routing algorithm. " IEEE transactions on communications, VOL. 38, NO. 3, March 1990
[5] Witold Pedrycz and Athanasios Vasilakos. Computational intelligence in telecommunication networks, CRC press, 2001
[6] Bertsekas and Gallager. Data networks, page 367, Prentice Hall, 1992.
[7] Gianni Di Caro and Marco Dorigo. "Antnet: Distributed Stigmergetic control for communication networks." Journal of artificial intelligence, 1998.
[8] Ronal R. Yager and Dimitar P. Filev. Essentials of fuzzy modeling and control, A Wiley – Interscience publication, 1994
[9] Hao Ying. Fuzzy control and modeling – Analytical Foundations and applications, IEEE press, 2000
[10] The Network Simulator - ns-2. http://www.isi.edu/nsnam/ns

1) Number of S-D pairs: 6
2) Packet size: 800 bytes
3) Interval: .005 Sec
4) Input: $6x(25-3.5) \times (800x(0.005)^{-1}x8x10^{-6}) = 165.12$Mbits
5) S-D pairs: (1-6), (9-5), (4-14), (6-12), (7-14), (3-2)
6) Load: 24.38 %

Metric	Input (Mbits)	Received (Mbits)	Percentage
Hops	165.12	142.63	86.3796
Delay	165.12	154.61	93.63493
Q-size	165.12	159.77	96.75993
0.8Q +0.2D	165.12	153.33	92.85974
0.2Q +0.8D	165.12	151.91	91.99976
0.5Q +0.5D	165.12	152.65	92.44792
Fuzzy	165.12	161.93	98.06

Table 2: Obtained throughput for different metrics for 6 S-D pairs for network 1

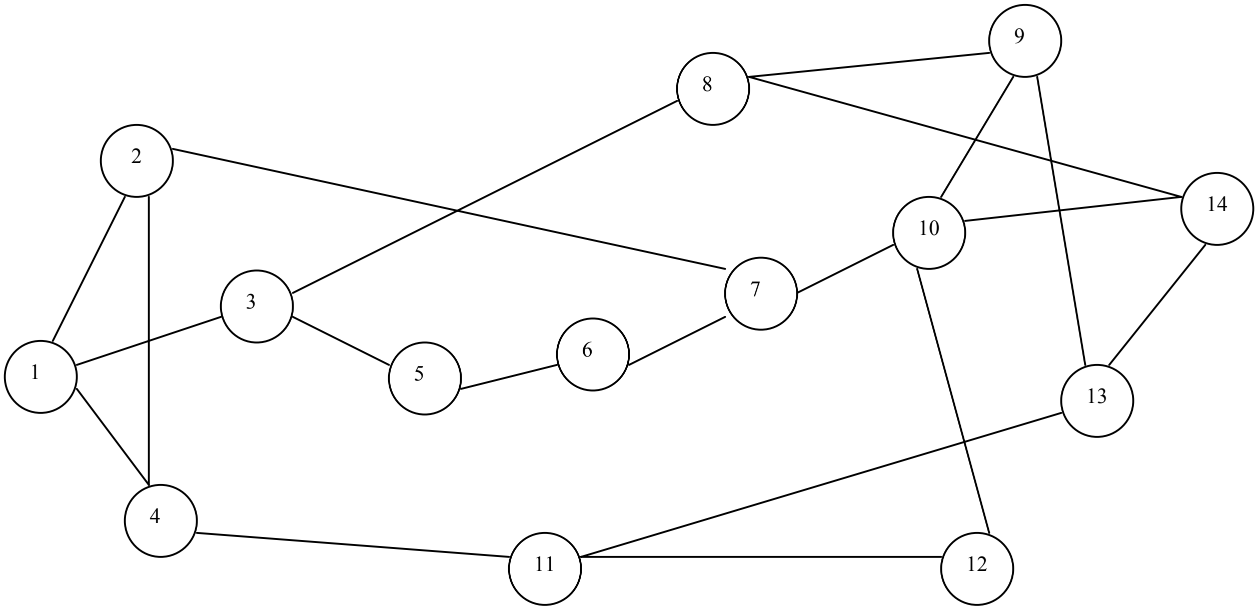

Figure3: NSFNET. Numbers within the circles are node identifiers. Each edge represents a bi-directional link.

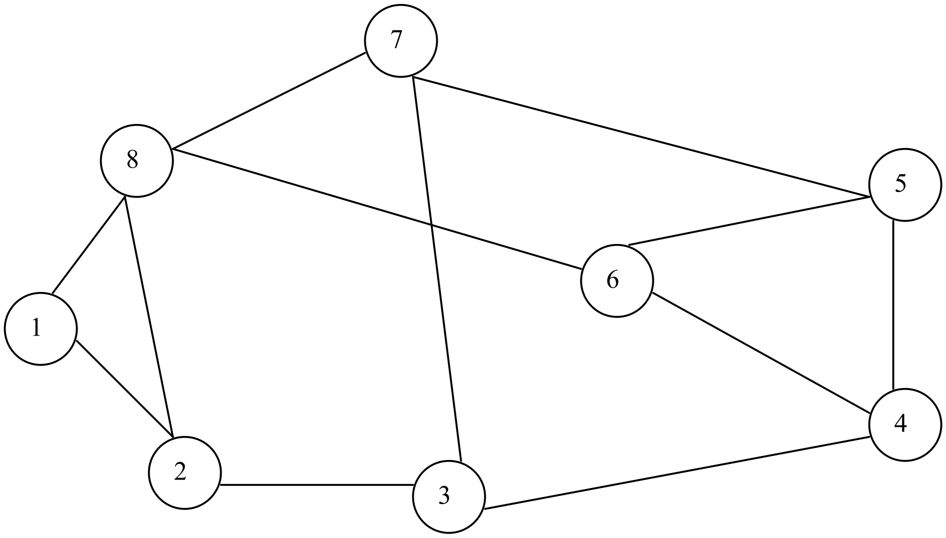

Figure 4: Network 2. Numbers within the circles are node identifiers. Each edge represents a bi-directional link.

Sno	Packet Size (Bytes)	S-D pairs	Load (%)	Q-size Throughput (%)	0.8Q +0.2D Throughput (%)	0.5Q +0.5D Throughput (%)	Fuzzy Throughput (%)
1	800	6	24.38	96.75	92.85	92.44	98.06
2	900	8	36.57	78.18	81.95	80.87	81.28
3	930	8	37.78	79.63	79.55	81.69	81.44

Table 3: Obtained throughput for different metrics for Network 1

Sno	Packet Size (Bytes)	S-D pairs	Load (%)	Q-size Throughput (%)	0.8Q +0.2D Throughput (%)	0.5Q +0.5D Throughput (%)	Fuzzy Throughput (%)
1	900	3	25.00	94.05	89.97	92.63	99.89
2	900	4	32.00	77.17	82.63	80.91	83.41
3	930	5	40.00	84.05	82.52	84.21	84.10

Table 4: Obtained throughput for different metrics for Network 2

Sno	Packet Size (Bytes)	S-D pairs	Load (%)	Q-size Average delay (sec)	0.8Q +0.2D Average delay (sec)	0.5Q+ 0.5D Average delay (sec)	Fuzzy Average delay (sec)
1	800	6	24.38	0.4309	0.4569	0.4540	0.4260
2	900	8	36.57	0.6249	0.6112	0.6019	0.6068
3	930	8	37.78	0.6589	0.6491	0.6507	0.6458

Table 5: Average delay of packet for different metrics for Network 1

Sno	Packet Size (Bytes)	S-D pairs	Load (%)	Q-size Average delay (sec)	0.8Q +0.2D Average delay (sec)	0.5Q+ 0.5D Average delay (sec)	Fuzzy Average delay (sec)
1	800	3	25.00	0.4480	0.4635	0.4663	0.4102
2	900	4	32.00	0.4996	0.4917	0.4855	0.5000
3	930	5	40.00	0.5515	0.5560	0.5331	0.5413

Table 6: Average delay of packet for different metrics for Network 2