A FUZZY ROUTING MECHANISM IN NEXT-GENERATION NETWORKS

Runtong Zhang*
Lab of Computer & Internet Architectures
Swedish Institute of Computer Science
Box 1263, SE-16429 Kista, Sweden
E-mail: runtong@sics.se

Keping Long
ARC Special Research Centre for Ultra Broadband
Information Networks (CUBIN)
The University of Melbourne, VIC 3010, Australia
E-mail: k.long@ee.mu.oz.au

ABSTRACT

QoS (Quality of Service) routing is a key network function for the transmission and distribution of digitized audio/video across next-generation high-speed networks. It has two objectives: finding routes that satisfy the QoS constraints and making efficient use of network resources. The complexity involved in the networks may require the consideration of multiple constraints to make the routing decision. In this paper, we propose a novel approach using fuzzy logic technique to QoS routing that allows multiple constraints to be considered in a simple and intuitive way. Simulation shows that this fuzzy routing algorithm is efficient and promising.

KEY WORDS

fuzzy logic control, QoS, routing, network

1. INTRODUCTION

In the current Internet, data packets of a session may follow different paths to the destinations, and the network resources (e.g., router buffer and link bandwidth) are fairly shared by packets from different sessions. However, this architecture does not meet the QoS requirements of future integrated services networks that will carry heterogeneous data traffic. First, it does not support resource reservation [1], which is vital for the provision of guaranteed end-to-end performance. Second, data packets may experience unpredictable delays and arrive at the destination out of order, which is undesirable for continuous real-time media. Hence, the next generation of high-speed wide area networks is likely to be connection-oriented for real-time traffic. This paper focuses on the routing problem of connection establishment in the unicast case. The problem of QoS routing discussed in this paper is to find an end-to-end network path between two end users that satisfies multiple constraints. Some classic end-to-end routing algorithms can also be refereed to QoS routing, in comparison with the best-effort routing.

QoS routing consists of two basic tasks. The first task is to collect the state information and keep it up to date. The second task is to find a feasible path for a new connection based on the collected information. A routing algorithm generally focuses on the second task, i.e., it assumes that a global state is well detected and the present work falls into this category.

One of the biggest difficulties in QoS routing area is that multiple constraints often make the routing problem intractable. This normally includes things such as node buffer capacities, residual link capacities, and the number of hops on the path (i.e., the number of nodes a packet must pass through on the route). Many common routing algorithms require that these factors be expressed together in a closed, analytical form for evaluation. Oftentimes, combining various factors into such mathematical forms can be extremely complex to derive and difficult to work with. If the complex forms previously mentioned are utilized, they can bead to a tremendous computational burden. Most of the existed QoS routing algorithms are proposed only to some kind of specific constraint. As a consequence, some information that would be helpful to include in the routing algorithm may not be included because there is no closed-form, mathematical way to express the information along with other information. This is one area where fuzzy control may be beneficial.

Many QoS routing algorithms have been proposed with a variety of constraints considered. We will give a brief review in section 2, and a full discussion concerning these algorithms can be found in [2 and 3]. However, as previously discussed, little work has been done in the area where multiple constraints are simultaneously taken into account. In this paper, we propose a novel approach using fuzzy logic technique to solve the QoS routing problem, which allows multiple constraints to be considered in a simple and intuitive way.

Fuzzy control is a control technique based on the principles of fuzzy set theory [4]. Fuzzy control systems are designed to mimic human control better than classical control systems by incorporating expert knowledge and experience in the control process. This is achieved by the use of linguistic variables in the control system. Linguistic variables allow the designer to include control rules that naturally follow human thought.

We introduce fuzzy control to the QoS routing problem based on the following observations: Normally, a good

* This author is also a professor with the Information Systems Institute, Northern Jiaotong University, Beijing, 100044, China.

Internet service requires several criteria simultaneously, and depends on the network situations (e.g., the structure or load), which are generally not available or dynamically changed. Fuzzy control is an intermediate approach between complicated analysis and simple intuition. It can easily handle several non-linear factors and needs no detailed mathematics descriptions for the systems. Some successful examples of implying fuzzy approach to the network optimization can be found in [5-10]. It could also allow a means of expressing complex relationships and dependencies predicted to be evident in future QoS enable communication networks that support various applications. This could have a great impact on the performance of the routing algorithm and consequently, the network performance. In fact, there has been literature written on using fuzzy techniques in communication networks [11 and 12]. A survey of recent advances in fuzzy logic in telecommunications networks can be found in [13].

2. QOS ROUTING

The notion of QoS has been proposed to capture the qualitatively or quantitatively defined performance contract between the service provider and the user applications. The QoS requirement of a connection is given as a set of constraints. A link constraint specifies a restriction on the use of links. A bandwidth constraint of a unicast connection requires, for instance, that the links composing the path must have certain amount of free bandwidth available. A path constraint specifies the end-to-end QoS requirement on a single path.

A feasible path is one that has sufficient residual resources to satisfy the QoS constraints of a connection. The basic function of QoS routing is to find such a feasible path. In addition, most QoS routing algorithms consider the optimization of resource utilization measured by an abstract metric in cost [2]. The cost of a link can be defined in dollars or as a function of the buffer or bandwidth utilization. The cost of a path is the total cost all links on the path. The optimization problem is to find the lowest-cost path among all feasible paths.

The problem of QoS routing is difficult for a number of reasons. First, distributed applications such as Internet phone and distributed games have very diverse QoS constraints on delay, delay jitter, loss ratio, bandwidth, and so on. Multiple constraints often make the routing problem intractable. Second, any future integrated services network is likely to carry both QoS and best-effort traffic, which makes the issue of performance optimization complicated. Third, the network state changes dynamically due to transient load fluctuation, connections in and out, and links up and down. The growing network size makes it increasingly difficult to gather up-to-date state information in a dynamic environment.

The routing problems can also be divided into two major classes: unicast routing and multicast routing. A unicast QoS routing problem is defined as follows: given a source node *a*, a destination node *b*, a set of QoS constraints *C*, and possibly an optimization goal, find the best feasible path from *a* to *b* which satisfies *C*. The multicast routing problem is defined as follows: given a source node *a*, a set *R* of destination nodes, a set of constraints *C* and possible an optimization goal, find the best feasible tree covering *a* and all nodes in *R* which satisfies *C*. The two classes of routing problems are closely related. Multicast routing can be viewed as a generalization of unicast routing in many cases.

QoS routing is different from the traditional best-effort routing. The former is normally connection-oriented to provide the guaranteed service. The latter can be either connection-oriented or connectionless with dynamic performance subject to the current availability of shared resources. Meeting the QoS requirement of each individual connection and reducing the call-blocking rate are important in QoS routing, while fairness, overall throughput, and average response time are the essential issues in traditional routing.

There are three routing strategies: source routing, distributed routing and hierarchical routing. They are classified according to how the state information is maintained and how the search of feasible paths is carried out. This paper considers the case of source routing.

Many routing algorithms in this area are proposed in the literature and we next introduce several typical ones. Generally, most source unicast routing algorithms transform the routing problem to a shortest path problem and then solve it by Djikstra's algorithm [14]. The Djikstra's algorithm is also known as the shortest path routing algorithm, and we will use it as one of the reference frameworks to test our work in section 5. The Ma-steenkiste algorithm [15] provides a routing solution to rate-based networks; The Guerin-Orda algorithm work with imprecision information, and hence is suitable to be used in hierarchical routing; The performance of the Chen-Nahrstedt algorithm [16] is tunable by trading overhead for success probability; The Awerbuch et. al. algorithm [17] takes the connection duration into account, which allows more precise cost-profit comparison. All the above algorithms are executed at the connection arrival time on a per-connection basis. Path precomputation and caching were studied to make a trade-off between processing overhead and routing performance.

3. FUZZY LOGIC CONTROL

A fuzzy control system [4] is a rule-based system in which a set of so-called fuzzy rules represents a control decision mechanism to adjust the effects of certain causes coming from the system. The aim of a fuzzy control system is normally to substitute for or replace a skilled human operator with a fuzzy rule-based system. Specifically, based on the current state of a system, an inference engine equipped with a fuzzy rule base determines an on-line decision to adjust the system behavior in order to guarantee that it is optimal in some certain senses.

The design process of a fuzzy control system consists of a series of steps. The first step in fuzzy control is to define the input variables and the control variables. The input variables may be crisp or fuzzy. Each variable must be quantified; for instance, the variable temperature may be quantified as cold, warm, or hot. Each quantification of the variable is assigned a membership function.

Once these membership functions have been defined for each quantification of the input and control variables, a fuzzy rule base must be design. This rule base determines what control actions take place under what input conditions. The rules are written in an if-then format. Once the rule base is established, an approximate reasoning method must be used to determine the fuzzy control action. The approximate reasoning method provides a means of activating the fuzzy rule base. An implication formula is used to evaluate the individual if-then rules in the rule base. A composition rule is used to aggregate the rule results to yield a fuzzy output set. The implication formula provides a membership function that measures the degree of truth of the implication relation (i.e., the if-then rule) between the input and output variables. One frequently used implication formula is that of Mamdani. Let a fuzzy rule be stated as follows: if x is A, then y is N. The implication formulas of Mamdani is as follows:

$$\mu_{A\to N}(x,y)=\mu_A(x)\wedge\mu_N(y) \quad (1)$$

where $\mu_A(x)$ is the membership of x in A, $\mu_N(y)$ is the membership of y in N, $\mu_{A\to N}(x,y)$ is the membership of the implication relation between x and y, and $\wedge$ is the minimum operator.

A defuzzification method is then applied to the fuzzy control action to produce a crisp control action. One simple and frequently used defuzzification method is the Height method. Let $c^{(k)}$ and f_k be the peak value and height, respectively, of the kth fuzzy set of the fuzzy output. Then by the Height method, the defuzzified crisp output u^* is given

$$u^*=\frac{\sum_{k=1}^{n} c^{(k)} f_k}{\sum_{k=1}^{n} f_k} \quad (2)$$

where n is the total number of the fuzzy sets of the fuzzy output.

There are generally two kinds of fuzzy logic controllers. One is feedback controller, which is not suitable for the high performance communication networks. Another one, which is used in this paper, is shown in Figure 1. This control mechanism is different from the conventional feedback control and considered as an adaptive control.

In principle, to design the fuzzy controller, we need to explore the implicit and explicit relationships within the system and subsequently develop the optimal fuzzy control rules as well as the knowledge base.

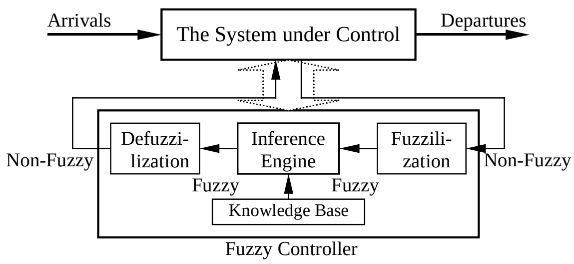

Figure 1: The fuzzy controller

In this paper, most of the membership functions for the fuzzy sets are chosen to be triangular. To describe the fuzzy rules, we use ZO, PS, PM, PB to indicate "zero", "positive small", "positive medium" and "positive big".

We simulate and control queueing systems in C++ language. Mamdani implication is used to represent the meaning of “if-then” rules. This kind of implication is most popular in the fuzzy control field because it is computationally simple and fits various practical applications. The height method of defuzzification is used to transform the fuzzy output into a usable crisp one.

4. THE FUZZY ROUTING ALGORITHM

The network model used for testing the fuzzy QoS routing algorithm is adapted from Balakrishnan et. al. [18], and it is shown in Figure 2.

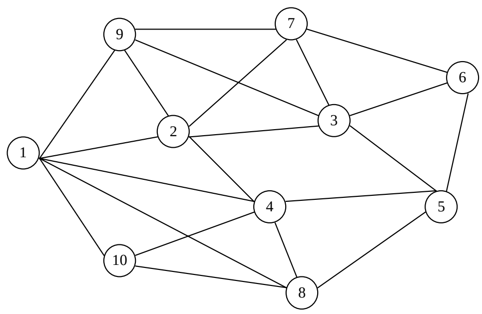

Figure 2: Experimental communication network topology

For the sake of easy illustration, it is assumed that the links between the nodes are with same transmission bandwidth, and their length are all the same. These assumptions are logical because the propagation delay of a traffic flow in the high performance communication is normally very small compared with its queueing delay at the switching nodes. Each node has incoming packet buffer with a maximum capacity of B. Nodes one, five, six seven, eight nine, and ten act as both traffic generating nodes and switching nodes. Nodes two three and four are pure switching nodes. A traffic route should be determined before a traffic flow is going to be sent off at its generating node, and the chosen route will not be changed afterward. The problem is to determine the optimal QoS routing policy for each traffic flow at its generating node based on the state of the system. The optimal criteria are multiple, which are the minimal percentage of connections rejected at the generating nodes, the minimal percentage of connections lost along the routes, and the minimal mean packet delay in the network.

For a given traffic flow at its generating node, the state of each eligible path is described by (s, n_i), where $s \subset \{1,2,3,4\}$ is the number of hops on the path, and $n_i=0,1,2,\ldots,B$, $i=1,2,\ldots,s$, is the number of packets currently in buffer i on the given path. The state of the system changes whenever an arrival or departure at any nodes along the given path occurs. Without loss of generality, the decision epochs are the time instances that a new traffic flow is being generated and sent to the network.

We use fuzzy control technique to solve this QoS routing problem and the algorithm is referred to fuzzy routing algorithm. The algorithm first determines the crisp path ratings for all eligible paths between the source and destination nodes from the viewpoint of fuzzy inference. The path with the highest rating is then chosen to route the traffic flow. The path rate in this paper represents the degree of the path usability in the sense of the multiple criteria required. The connection is only rejected if all of the buffers on the chosen are currently full. Otherwise, the connection traffic is routed over the chosen path for the duration of the connection. Whenever traffic flow is routed to a chosen path, a packet is dropped when it arrives at a full buffer.

We choose as fuzzy inputs: the number of hops on the path s and the path utilization $\bar{\rho}$. The fuzzy output is the path rating r. The fuzzy rule base is shown in Table 1.

Table 1: Fuzzy rule base

r		$\bar{\rho}$			
		ZO	PS	PM	PB
s	ZO	PB	PM	PS	ZO
	PS	PM	PS	ZO	ZO
	PM	PS	ZO	ZO	ZO
	PB	ZO	ZO	ZO	ZO

The path utilization $\bar{\rho}$ is calculated by the following series of steps. First, the utilization of each buffer ρ_i on the path is calculated as in (3). The sum of these utilization measures is taken and used to generate a weighting measure λ_i for each buffer i as in (4) and (5). Finally, the estimated path utilization $\bar{\rho}$ is calculated by multiplying the umber of packets in each buffer by its corresponding weight factor, which is shown in (6).

$$\rho_i = \frac{n_i}{B}, \ i=1 \text{ to } s \tag{3}$$

$$P = \sum_{t=1}^{t} \rho_i \tag{4}$$

$$\lambda_i = \frac{\rho_i}{P}, \ i=1 \text{ to } s \tag{5}$$

$$\bar{\rho} = \sum_{i=1}^{t} \lambda_i \cdot n_i \tag{6}$$

The membership functions for the fuzzy variables s, $\bar{\rho}$ and r are shown in Figures 3 (a), (b) and (a), respectively. The universes of discourse for the fuzzy variables are all chosen [0, 6]. The sojourn time of a packet in the system increases with the total number of packets in the system as the sequence 1,3,6,…, which is given by $t_j=t_{j-1}+j$, $t_0=0$, $j=1,2,\ldots$, thus we choose the fuzzy membership functions for $\bar{\rho}$ as shown in Figure 3(b).

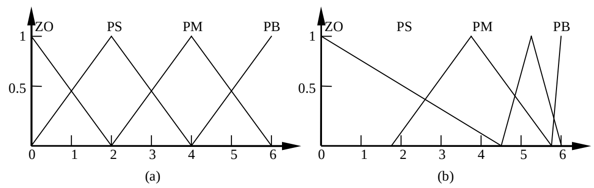

Figure 3: Membership functions

To sum up, the fuzzy QoS routing algorithm is outlined as follows.

(a) All eligible paths between the source and destination nodes and corresponding state information are collected. This work is needed for all source routing related algorithms.
(b) Calculating the values of s and $\bar{\rho}$ for each eligible path by (3-6).
(c) Using the calculated values pair of s and $\bar{\rho}$ as crisp inputs, we determine the crisp path ratings r for each eligible path via fuzzification (based on the membership functions shown in Figure 3), fuzzy inference (based on the rule base shown in Table 1 and the Mamdani implication) and de-fuzzification (based on the High method of de-fuzzification).
(d) The path with highest rating is chosen to route the traffic flow.

It should be stressed that the controller derived in this section is "optimal" in the intuitive sense of fuzzy logic. Due to the absence of a precise analytical model it seems impossible, at least for now, to prove optimality rigorously. Questions pertaining to validity, accuracy, and sensitivity are too hard to be answered in the context of linguistic variables and “if-then” statements. There is, however, a note of optimism. We have run a number of simulations for various systems and the results we obtained for a wide range of parameters were encouraging. In Section 5, we show on such simulation results.

5. SIMULATION RESULTS

We examine a QoS network model shown in Figure 1. It is assumed that the links between the nodes are all 2 km in length, and the bandwidth of the links are all 100 Mbps. Each node has incoming packet buffer with a maximum capacity of 50 packets. The interarrival rate of connection attempts is assumed to be exponential. The mean of this exponential variable varies from 0.5 to 1.0 in increments of 0.05. A connection is rejected only if all of the buffers on the path chosen for routing the call are completely full. We wish to determine the optimal QoS routing policy for each traffic flow at its generating node based on the state of the system. The optimal criteria are multiple, which are the minimal percentage of connections rejected at the generating nodes, the minimal percentage of connections lost along the routes, and the minimal mean packet delay in the network.

The fuzzy routing scheme is tested against three other routing algorithms: a fixed directory routing algorithm [18], a shortest path routing algorithm, and a "crisp" or non-fuzzy version of the fuzzy routing scheme.

The fixed directory routing algorithm is a simplified version of the shortest path problem, and is also based on the number of hops on the path. All of the one two, three, and four hop paths for a given source/destination pair are listed in a directory. The directory gives preference to the minimum hop paths. When a connection is requested, it is made on the first path in the directory that can accommodate the connection.

The shortest path routing algorithm calculates the shortest delay path. Once again, only the one, two, three, and four hop paths for each source/destination node pair are considered. The delay estimate is obtained by determining the time it will take to service all packets currently at each hop on the path and summing these results. The path with the shortest estimated delay is chosen to route the connection. In the case of a tie, the path with the fewest number of hops is chosen. If both paths have the same number of hops, the first path analyzed is chosen to route the connection.

The "crisp" non-fuzzy version of the fuzzy routing algorithm (henceforth referred to as the crisp routing algorithm) utilizes the path utilization calculation presented above in the breakdown of the fuzzy control routing algorithm. This path utilization metric is calculated for all one, two, three, and four hop paths. The path with the lowest value of this metric is chosen to route the call. In the case of tie, the path with fewest number of hops is chosen. If both paths have the same number of hops, the first path analyzed is chosen to route the connection. The connection is rejected if all buffers on the chosen path are full.

The graphs for the percentage of connections rejected, the percentage of packets lost, and the mean packet delay (in second) in the network are shown in Figures 4, 5 and 6. These statistic parameters are plotted in the value axis of the three figures, respectively, while the category axis are all mean call interarrival time.

Figure 4 illustrates that the fuzzy routing algorithm rejects a smaller percentage of connections than the other three routing algorithms. Recall that the only reason for which connections are rejected is if all buffers on the route chosen are full; therefore, the fuzzy routing algorithm appears to outperform the others at dispersing traffic in the network (to avoid extreme congestion on individual paths).

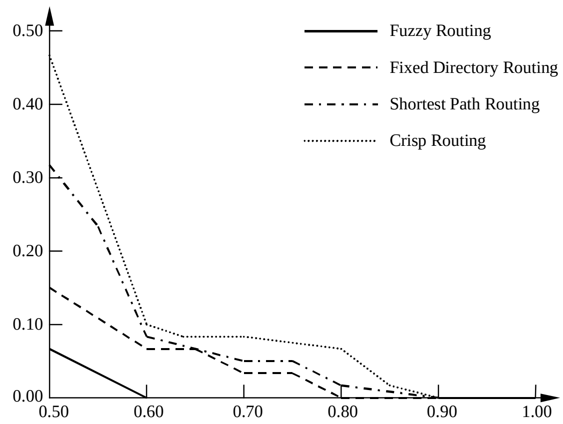

Figure 4: QoS network percentage of connections rejected

Figure 5 reveals that the fuzzy routing algorithm also loses a smaller percentage of packets than the other routing algorithms. This is another illustration of the fuzzy routing algorithm's ability to outperform the other routing algorithms at dispersing traffic in the network. The fact that a fewer percentage of packets are lost under the fuzzy scheme means that not as many packets are approaching full buffers under this scheme.

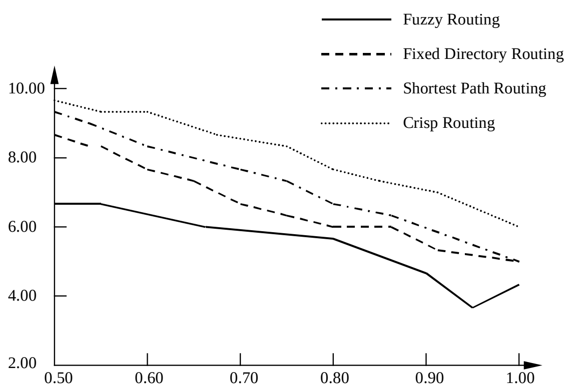

Figure 5: QoS network of packets lost percentage

Figure 6 reveals that the fuzzy routing algorithm results in a smaller mean packet delay in the network than the other routing algorithms. The fuzzy algorithm also does not experience as sharp an increase in mean packet delay as the other algorithms when the call arrival rate increases (i.e., the interarrival rate decreases). This illustrates the ability of the fuzzy routing algorithm to handle an increased traffic load better than the other three algorithms.

Overall, the fuzzy routing algorithm outperforms the other three routing algorithms with regard to all of the measures collected. The results shown in the graphs indicate that the fuzzy routing algorithm does a better job at dispersing traffic in a more uniform manner throughput the network. It also handles an increased traffic load more efficiently. The fact that his fuzzy algorithm can perform so well even while being partly based on an ambiguous measure (the path utilization) clearly highlights the true merit of fuzzy control.

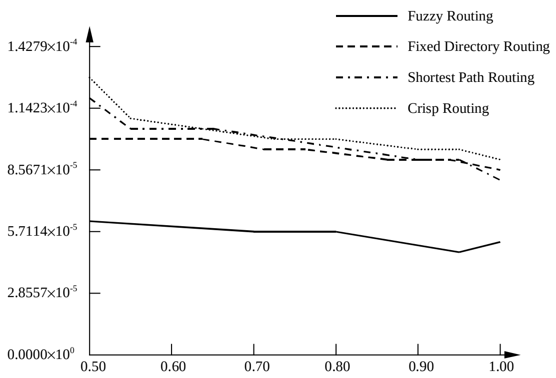

Figure 6: QoS network mean packet delay in the network

6. CONCLUSIONS

We propose a routing algorithm based on fuzzy control for QoS routing. The benefits of such an algorithm include increased flexibility in the constraints that can be considered in the routing decision and the ease in considering multiple constraints. The computational burden of a fuzzy control routing system is not severe enough to rule it out as a viable option. This is heavily due to the simple if-then structure of the rule base.

The design of a simple fuzzy control routing algorithm is presented and tested on an experimental QoS network. The results of this experiment prove favorable for the fuzzy control routing algorithm. The fuzzy algorithm displayed better performance than its "crisp" counterpart, the fixed directory routing algorithm and the classic shortest path routing algorithm. The results of this research indicate a promising future for fuzzy control in the world of communication network routing.

REFERENCES

[1] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, Resource Reservation Protocol (RSVP) Version 1 Functional Specification, *RFC 2205*, September 1997.

[2] S. Chen and K. Nahrstedt, An overview of quality of service routing for next-generation high-speed networks: problems and solutions, *IEEE Network*, 12(6), 1998, 64-79.

[3] R. Guerin and A. Orda, QoS based routing in networks with inaccurate information: theory and algorithms, *Proc. IEEE/ INFOCOM'97*, 1997, Japan.

[4] D. Driankov, H. Hellendoorn and M. Reinfrank, *An introduction to fuzzy control*, Springer-Verlag, Berlin, New York, 1993.

[5] Y. Phillis and R. Zhang, Fuzzy service rate control of queueing systems, *IEEE Trans. Systems, Man, and Cybernetics, Part B*, 29(4), Aug. 1999, 503 - 517.

[6] R. Zhang and Y. Phillis, A fuzzy approach to the flow control problems, *J. Intelligent and Fuzzy Systems*, 6, 1998, 447-458.

[7] R. Zhang and Y. Phillis, Fuzzy control of queueing systems with heterogeneous servers, *IEEE Trans. Fuzzy Systems*, 7(1), 1999, 17-26.

[8] R. Zhang and Y. Phillis, Fuzzy control of arrivals to tandem queues with two stations, *IEEE Trans. Fuzzy Systems*, 7(3), 1999, 161-167.

[9] R. Zhang and Y. Phillis, Fuzzy control of two-station queueing networks with two types of customers, *J. Intelligent and Fuzzy Systems*, 8, 2000, 27-42.

[10] R. Zhang and Y. Phillis, Admission control and scheduling in simple series parallel networks using fuzzy logic, *IEEE Trans. Fuzzy Systems*, 9(2), April 2001, 307-314.

[11] R. Cheng and C. Chang, Design of a fuzzy traffic controller for ATM networks, *IEEE/ACM Trans. Networking*, 4(3), 1996, 460-469.

[12] C. Douligeris and G. Develekos, A fuzzy logic approach to congestion control in ATM networks, *Proc. 1995 IEEE International Conference on Communications*, 1995, 1969-1973.

[13] S. Ghosh, Q. Razouqi, H. J. Schumacher and A. Celmins, A survey of recent advances in fuzzy logic in telecommunications networks and new challenges, *IEEE Trans. Fuzzy Systems*, 1998, 6(3).

[14] E. Dijkstra, A note on two problems in connection with graphs", *Numerische Mathematik*, 1, 1959, 169-271.

[15] Q. Ma and P. Steenkiste, Quality of service routing with performance guarantees, *Proc. 4th IFIP Workshop QoS*, 1997.

[16] S. Chen and K. Nahrstedt, On finding multi-constrained paths, *Proc. IEEE/ICC'98*, 1998.

[17] B. Awerbuch and et. al., Throughput-competitive on-line routing, *Proc. 34th Annual Symp. Foundations of Comp. Sci*, 1993.

[18] K. Balakrishnan, D. Tipper and D. Medhi, Routing strategies for fault recovery in wide area packet networks, *Proc. MILICOM'95*, 1995, 1139-1143.

[19] P. R. Bell and K. Jabbour, Review of point-to-point network routing algorithms, *IEEE Communications Magazine*, 24(1), 1986, 34-38.

[20] C. Courcoubetis, G. Kesidis, A. Ridder, J. Walrand and R. Weber, Admission control and routing in ATM networks using inferences from measured buffer occupancy, *IEEE Trans. Communications*, 43(4), 1995, 1778-1784.