

# Dynamic QoS Solution for Enterprise Networks Using TSK Fuzzy Interpolation

Jie Li, Longzhi Yang

Department of Computer and Information Sciences  
Faculty of Engineering and Environment  
Northumbria University  
Newcastle upon Tyne, NE1 8XT, U.K.  
Email: {jie.li-2, longzhi.yang}@northumbria.ac.uk

Xin Fu<sup>†</sup>, Fei Chao<sup>‡</sup>

<sup>†</sup>School of Management  
<sup>‡</sup>Department of Cognitive Science  
Xiamen University  
Xiamen, P.R. China  
Email: {xfu, fchao}@xmu.edu.cn

Yanpeng Qu

Information Science and  
Technology College  
Dalian Maritime University  
Dalian, P.R. China  
Email: yanpengqu@dlmu.edu.cn

**Abstract**—The Quality of Services (QoS) is the measure of data transmission quality and service availability of a network, aiming to maintain the data, especially delay-sensitive data such as VoIP, to be transmitted over the network with the required quality. Major network device manufacturers have each developed their own smart dynamic QoS solutions, such as AutoQoS supported by Cisco, CoS (Class of Service) by Netgear devices, and QoS Maps on SROS (Secure Router Operating System) provided by HP, to maintain the service level of network traffic. Such smart QoS solutions usually only work for manufacture qualified devices and otherwise only a pre-defined static policy mapping can be applied. This paper presents a dynamic QoS solution based on the differentiated services (DiffServ) approach for enterprise networks, which is able to modify the priority level of a packet in real time by adjusting the value of Differentiated Services Code Point (DSCP) in Internet Protocol (IP) header of network packets. This is implemented by a 0-order TSK fuzzy model with a sparse rule base which is developed by considering the current network delay, application desired priority level and user current priority group. DSCP values are dynamically generated by the TSK fuzzy model and updated in real time. The proposed system has been evaluated in a real network environment with promising results generated.

## I. INTRODUCTION

A multitude of applications require their data to be transmitted over networks, including real-time high quality voice and video data. The massive cyber traffic is getting busier thanks to the increasingly more connected devices, i.e., the ‘Internet of Things’. The delay-sensitive data, such as real-time voice and video data, need to be transported with a limited delay. The Quality of Services (QoS) is the measure or mechanism to ensure guaranteed high-quality performance regarding application data transmission over the network. It provides preferential delivery service for the applications that need it by ensuring sufficient bandwidth, controlling latency, and reducing data loss [1].

There are generally two principal approaches to implement QoS in modern IP network, Integrated Services (IntServ) and Differentiated Services (DiffServ). IntServ model uses a specially designed protocol to request and reserve the network resources from the network devices, whilst the DiffServ model tailors excess network traffic packets via the queueing strategies based on the value in the DSCP field of an IP header. Regardless of which approach is deployed, the static QoS mappings are usually pre-defined based on the required

services level of different network services. In general, those static mapping policies are not able to be intelligently modified in real time. A number of more efficient QoS mechanisms have been recently developed by employing AI algorithms, such as neural network and fuzzy logic [2]–[5]. These approaches are successful in general, but they are not able to flexibly deal with dynamic priority requirements for different types of clients. To address such limitation, this paper proposes a dynamic DiffServ-based QoS solution for enterprise networks, implemented using a fuzzy inference approach.

Fuzzy inference is a mechanism that uses fuzzy logic and fuzzy set theory to map between inputs and outputs, which has been widely used in various decision making systems. Such conventional fuzzy inference approaches only work with rule bases (or knowledge bases) which cover the entire problem domain; otherwise, no results will be generated. Fuzzy interpolation, initially proposed in [6], not only addresses this issue, but also helps in complexity reduction for complex fuzzy models. By employing fuzzy interpolation, a reasonable result always can be generated even when a given input does not overlap with any rule antecedent. A number of fuzzy interpolation methods have been developed to support inference with either Mamdani-style rule bases or TSK-style rule bases with successful real-world applications [6]–[22].

The dynamic QoS solution proposed in this paper is developed upon a fuzzy interpolation approach with a TSK-style rule base, noted as TSK-interpolation [22]. Briefly, TSK-interpolation extends the original TSK fuzzy inference by allowing it to perform with a sparse rule base. In this work, the TSK-style rule base is extracted from data, and the generated rule base is sparse due to the nature of the problem to be solved herein. The TSK-interpolation estimates the DSCP value in real time for each network packet based on its current attribute values, no matter the case is covered by the rule base or not. Then, the generated DSCP value is used to replace the original value in the DSCP field of the packet’s IP header, thereby adjusting the priority table of the router for the processing of the corresponding packet.

The rest of the paper is structured as follows: Section II introduces the theoretical underpinnings of TSK fuzzy interpolation approach and the quality of services. Section III presents the proposed QoS system utilising TSK-interpolation. Section IV discusses the experimentation to demonstrate the

work of the proposed system. Finally, Section V concludes the paper and points out possible future works.

## II. BACKGROUND

The background techniques, TSK-interpolation and the QoS architecture, especially the DiffServ model, are briefly reviewed in this section.

### A. TSK-interpolation

The original TSK inference system generates a crisp inference result by utilising the weighted average of the sub-consequences based on determined firing strength for all fired rules [23]. Obviously, no rule will be fired if a given input does not overlap with any rule antecedent. As a consequence, the TSK inference cannot be performed. TSK-interpolation was proposed to address such issue which generates a consequence by considering all the rules in the rule base [22]. Suppose that a sparse TSK rule base is comprised of  $n$  rules:

$$\begin{aligned} R_1 : & \mathbf{IF} \ x_1 \text{ is } A_1^1 \text{ and } \cdots x_j \text{ is } A_j^1 \cdots \text{ and } x_m \text{ is } A_m^1 \\ & \mathbf{THEN} \ z = f_1(x_1, \cdots, x_m), \\ & \dots \dots \\ R_n : & \mathbf{IF} \ x_1 \text{ is } A_1^n \text{ and } \cdots x_j \text{ is } A_j^n \cdots \text{ and } x_m \text{ is } A_m^n \\ & \mathbf{THEN} \ z = f_n(x_1, \cdots, x_m), \end{aligned} \quad (1)$$

where  $A_j^i$ , ( $i \in \{1, 2, \dots, n\}$  and  $j \in \{1, 2, \dots, m\}$ ) represents a normal and convex polygonal fuzzy set that can be denoted as  $(a_{j1}^i, a_{j2}^i, \dots, a_{jv}^i)$ ,  $v$  is the number of odd points of the fuzzy set. Given an input  $I = (A_1^*, A_2^*, \dots, A_m^*)$  in the input domain, a crisp inference result can be generated by the following steps:

**Step 1:** Identify the matching degrees between the given input  $(A_1^*, A_2^*, \dots, A_m^*)$  and rule antecedents  $(A_1^i, A_2^i, A_3^i, \dots, A_m^i)$  for each rule  $R_i$  by:

$$S(A_j^i, A_j^*) = \left( 1 - \frac{\sum_{q=1}^v |a_{jq}^i - a_{jq}^*|}{v} \right) \cdot (DF), \quad (2)$$

where  $DF$  is described as *distance factor*, which is a function of the distance between the two concerned fuzzy sets, and is defined as:

$$DF = 1 - \frac{1}{1 + e^{-cd+5}}, \quad (3)$$

where  $c$  is a sensitivity factor, and  $d$  represents the Euclidean distance between the two fuzzy sets. in particular,  $c$  is a positive real number. Smaller value of  $c$  leads to a similarity degree which is more sensitive to the distance of two fuzzy sets, and vice versa.

**Step 2:** Determine the firing degree of each rule by aggregating the matching degrees between the given input and its antecedent terms by:

$$\alpha_i = S(A_1^*, A_1^i) \wedge S(A_2^*, A_2^i) \wedge \cdots \wedge S(A_m^*, A_m^i), \quad (4)$$

where  $\wedge$  is a t-norm operator usually implemented as a minimum operator.

**Step 3:** Generate the final output by integrating the sub-consequences from all rules by:

$$z = \sum_{i=1}^n \alpha_i \cdot f_n(x_1, \cdots, x_m) / \sum_{i=1}^n \alpha_i. \quad (5)$$

### B. Quality of Services

QoS is often implemented in enterprise networks to measure and manage the transmission quality for different type of applications. QoS is usually determined by three factors, including loss rate, delay and jitter. Loss rate is the number of packets that are not received compared to the total number of packets for transmission. The loss is typically a function of network availability. In a highly available network, the loss should be essentially 0. Delay is the finite amount of time it takes a packet to reach the final destination device after being transmitted from the sending endpoint device. Jitter is also called delay variation, which is the difference in the end-to-end delay between packets, or the measuring time difference in packet inter-arrival time. The attribute of delay somehow reflects the other two attributes. In particular, a high amount of delayed packets commonly suggests a congestion in the network, which usually also leads to a high percentage of packet loss and unstable data transmission, and vice versa.

As introduced earlier, there are generally two types of QoS mechanisms in modern IP networks. DiffServ provides QoS by differentiating traffic whereas IntServ provides QoS by building a virtual circuit using the bandwidth reservation technique. IntServ requires the nodes in the network to remember the state information about the flow whilst DiffServ does not as it operates on individual IP packets. Particularly in the DiffServ QoS approach, network devices use the queueing strategy to tailor performance to expectations. In general, manufacturers have their own queueing methods to manage the QoS, such as first-in-first-out (FIFO), weighted fair queueing (WFQ), and priority queueing (PQ).

Regardless of the queueing strategies, a priority value is always assigned to each packet to indicate the different service level requirement for the particular application. The priority value is represented by the first 6 bits in the DSCP field of the IP header, as shown in Figure 1. In theory, 6 bits binary value can define up to 64 possible priority values. However, in the 64 possible values, only 21 of them are commonly used, which are listed in Table I. Other fields of the header each has its purpose, but this is out of the scope of this paper and thus omitted. According to the types of the traffic, such as real time voice data and bulk data, the QoS mechanism is able to allocate suitable priority value from 21 common values upon the current QoS policies to the corresponding traffic packet, thus to enable the routers and switches to manage the quality of services.

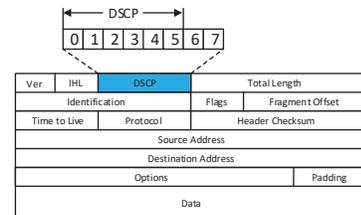


Fig. 1. IPv4 header and its DSCP field

TABLE I. COMMONLY USED DSCP VALUES

No.	DSCP Field Binary	DSCP Field Decimal	No.	DSCP Field Binary	DSCP Field Decimal
1	000 000	0	11	011 010	26
2	001 000	8	12	011 100	28
3	001 010	10	13	011 110	30
4	001 100	12	14	100 000	32
5	001 110	14	15	100 010	34
6	010 000	16	16	100 100	36
7	010 010	18	17	100 110	38
8	010 100	20	18	101 000	40
9	010 110	22	19	101 110	46
10	011 000	24	20	110 000	48
			21	111 000	56

### III. DYNAMIC QoS FOR ENTERPRISE NETWORKS

The proposed dynamic QoS system is essentially a decision making system to determine the value of DSCP in real time. This is achieved by adapting a 0-order TSK fuzzy reasoning system, which considers both the current network congestion index and the application/user desired priority. In particular, given a network packet regarding a certain application, the value of its DSCP field in the IP header is generated by the proposed system. Then, this packet is forwarded to the next hop device, such as a router or a switch, and joins its priority queue. The device takes action based on the values of the DSCP; it either sends this packet out immediately, with a certain delay, or even drops off the packet. The working procedure of the proposed system is illustrated in Fig. 2. Notice that this paper focuses only on the prediction of the DSCP values. The network traffic (including internal and external) are handled based on the determined DSCP values varying from manufacturer to manufacturer, which is beyond the scope of this paper.

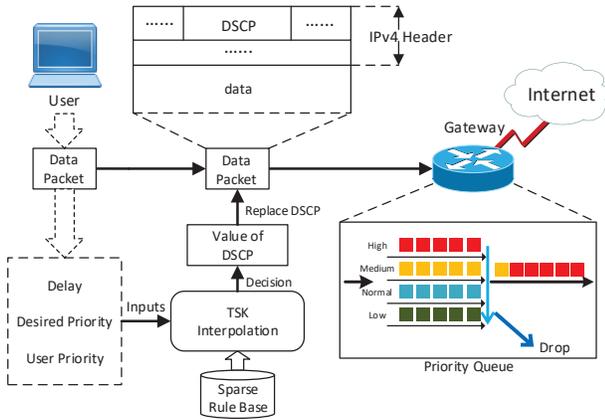


Fig. 2. The overview of the dynamic QoS system

#### A. Rule Base Generation

1) *Rule Antecedent Features*: Within the three factors considered in QoS (i.e., loss rate, delay and jitter), the attribute of delay somehow reflects the other two attributes. Therefore, the attribute of delay is taken as an input feature for TSK-interpolation in implementing the proposed QoS system. In

TABLE II. CISCO QoS APPLICATIONS CLASSIFICATION

Application Class	Description	Examples	Desired Priority Level
IP Routing	Communication between network devices to insure the network connectivity	OSPF, EIGRP	10
Voice	Voice data of VoIP telephony	VoIP traffic	9
Interactive Video	IP video conferencing	Skype video call, WebEx	8
Streaming Video	e-Learning application or multicast company meetings	Cisco DMS, IP/TV	7
Mission-Critical Data	Enterprise own defined critical applications that have the highest priority	N/A	6
Call-Signalling	Signalling traffic for VoIP	Dialing signal traffic from IP phone	5
Transactional Data	Interactive data applications	CRM	4
Network management	Network operations, administration, or management traffic	SSH, SNMP	3
Bulk Data	non-interactive data applications, which means users are not awaiting a response	FTP/SFTP, Online backup	2
Best Effort	Default class non classified applications should belong to	Web browsing	1
Scavenger	Non -business related	Gaming, P2P	0

order to allow the proposed system to deal with different situations in real time to reflect the dynamic priority requirements for various types of applications from a variety of clients, two more attributes are also considered as system inputs: the application desired priority and the user priority group. Consequently, the proposed system takes three input features (i.e., network delay, application desired priority, and user priority group), and it then produces one crisp output indicating the desired DSCP value. For simplify, only normal and convex triangle fuzzy sets are used in this work. The three input features and the fuzzy partition on these input domains are detailed below.

*$x_{delay}$* : The delay value is usually between 0 and 1000 milliseconds. In this work, the problem domain of this variable is partitioned into 4 triangle fuzzy sets as shown in Fig. 3(a) based on the knowledge of network engineers. Any delay value greater than 1000ms is capped as 1000ms in this work, as the influence of very large delay on network transmission is almost the same as 1000ms.

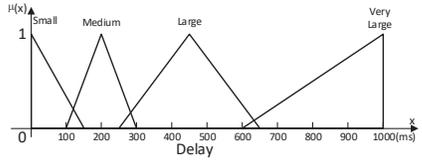
*$x_{DesiredPriority}$* : Every application has a unique traffic pattern and service level requirement, which usually requires a dedicated QoS priority to provide and guarantee data transmission quality over networks. In order to help group multiple applications in the enterprise network, Cisco unifies the QoS baseline and provides standards-based recommendations, which defines 11 classes of applications based on their functionality [24]. The details of the Cisco QoS baseline classification and relevant priorities are listed in Table II. These 11 linguistic classes outlined by Cisco are fuzzified by 11 fuzzy sets, as illustrated in Fig. 3(b). Notice that different group of applications can be identified by network devices based on the source and destination IP addresses, the type of the protocol, and the source and destination port numbers in the IP header, which is beyond the scope of this paper and not detailed here.

*$x_{UserPriorityGroup}$* : This input feature defines the priority of the end user to utilise the network resources. For instance, VoIP devices always require higher service level than that for email servers. Also, the required service levels for different departments may vary. For example, staff in sales department often requires a higher priority to use the internet resources than the shop floor staff, as smooth online payments should always be guaranteed. In this work, four different priorities are assigned to end users by following the work of [25]: low, normal, priority, and critical. The fuzzy representation

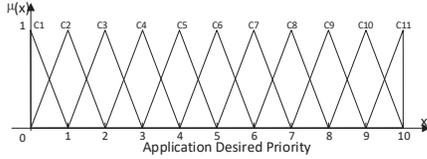
TABLE III. END USERS PRIORITY MAPPING

End Users	Class
VoIP Devices	Critical
Enterprise Servers	Priority
Department Managers	Priority
IT Department	Priority
Accounting Department	Normal
Purchasing Department	Normal
Others	Low

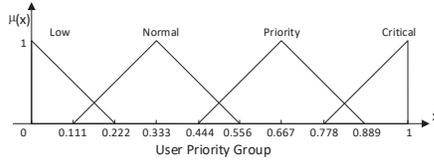
of this input domain partition is shown in Fig. 3(c), and the assignment of priority categories to different departments in a typical company is listed in Table III. Note that flexible priority for different user groups may be required to maximise the efficiency of enterprise networks to support various activities. In this case, manual modification of the priority table is required.



(a) Input - Network delay



(b) Input - Application desired priority



(c) Input - User priority group

Fig. 3. Fuzzy input variables

2) *Rule Consequence Feature*: As discussed in Section II-B, the first 6-bit of DSCP in the differentiated services field of an IP header indicates traffic priority. Currently, 21 DSCP values are commonly used to define 21 different priorities which are listed in Table I. The consequence of each rule then is an integer number between 1 and 21 to map the 21 commonly used priority levels of QoS.

3) *Sparse TSK-style Rule Base*: Based on the problem domain fuzzy partition discussed above, 176 fuzzy rules would be included in the rule base. However, the inclusion of the majority of these rules does not lead to noticeable system improvement due to the generalisation ability of TSK-interpolation. Therefore, the rule base is reduced based on expert knowledge in this work. In particular, all 11 fuzzy rules corresponding to the Cisco recommendation are kept

in the rule base. For instance, Cisco recommends that the VoIP data requires a critical priority bandwidth; network delay should be no more than 150ms, and should be marked with a DSCP value of 46. Also, the network management data should be of a DSCP value of 16, with the requirement of a minimum bandwidth and up to large delay. Based on those recommendations, 2 TSK rules are defined as follows:

$$\begin{aligned}
 R_1 : & \text{IF } x_{delay} \text{ is } S \text{ and } x_{DP} \text{ is } C10 \text{ and } x_{UPG} \text{ is } Critical \\
 & \text{THEN } z = 19, \\
 R_2 : & \text{IF } x_{delay} \text{ is } L \text{ and } x_{DP} \text{ is } C4 \text{ and } x_{UPG} \text{ is } Normal \\
 & \text{THEN } z = 6,
 \end{aligned} \tag{6}$$

where the consequences  $z = 19$ , and  $z = 6$  represent the DSCP value of 46 and 16 (as shown in Table I), respectively.

Noting that only the domain of variable  $x_{UserPriorityGroup}$  is fully covered by the 11 rules based on the Cisco recommendation. In order to have a better coverage for the other two variable domains, more rules have been kept based on expert knowledge of network engineers, which results in a more compact and concise rule base containing 27 rules as listed in Table IV.

TABLE IV. GENERATED RULE BASE

No.	Input			Output Decision	No.	Input			Output Decision
	Delay	ADP	UPG			Delay	ADP	UPG	
1	S	C11	C	20	15	L	C4	C	10
2	S	C10	C	19	16	L	C5	C	11
3	M	C9	P	15	17	L	C6	C	14
4	M	C8	P	14	18	M	C7	C	15
5	M	C7	N	11	19	M	C8	C	18
6	L	C6	N	10	20	M	C9	C	19
7	L	C5	N	7	21	VL	C9	C	20
8	L	C4	N	6	22	VL	C8	C	19
9	L	C3	L	3	23	VL	C6	C	15
10	VL	C2	L	2	24	VL	C5	C	13
11	VL	C1	L	1	25	VL	C4	C	14
12	VL	C1	C	2	26	VL	C3	C	9
13	VL	C2	C	3	27	VL	C2	C	5
14	VL	C3	VL	7					

## B. Decision Making for QoS priority

Once the sparse fuzzy rule base is generated, TSK-interpolation is then naturally applied to make decisions on the priority levels of data packets. Assume that the network delay, desired application priority, and user group priority regarding a given traffic packet are given as  $O = (x_1^*, x_2^*, x_3^*)$ , where  $x_1^*$ ,  $x_2^*$ , and  $x_3^*$  are crisp values. The calculation process of the conclusion using the TSK-interpolation approach is summarised as follows:

**Step 1:** Calculate the similarity degrees between the given input ( $O$ ) and rule antecedents of each rule using Equation 2.

**Step 2:** Determine the firing degree of each rule by following Equation 4.

**Step 3:** Aggregate the final crisp inference result through Equation 5.

**Step 4:** Generate the final priority decision by applying the round function on the aggregated crisp inference result.

Once the crisp inference result has been generated, it will be mapped back to a corresponding DSCP value based on Table I, and this DSCP value is then used to replace the existing one in the corresponding IP header. From this, the network device places this packet into the right position in the priority queue awaiting for further operation.

#### IV. EXPERIMENTATION

The proposed system has been applied to a real local area network (LAN), which is designed to simulate an enterprise network, for validation and evaluation purposes. The experimentation and its results are detailed in this section.

##### A. Experimental Environment

The network environment used in the experiment is shown in Figure 4. In particular, a Cisco 2921 enterprise router with DiffServ model enabled was used as the gateway router to handle different QoS requests. Two client PCs were employed as packet sender and receiver to simulate different types of network traffic. An IP phone was set up with a SIP service to generate VoIP traffic. As the proposed system only dealt with DSCP values, any DiffServ supported router or layer 3 switch could be used in this experimentation to replace the Cisco router. The proposed system was deployed upon the network bridge device between the router and the LAN devices. The network bridge in this experiment had two important functions, including decapsulating network data packets between the router and the terminal devices, and then collecting part of the input values from IP headers of network packets for the proposed priority classification system.

The information in the protocol field of the IP header indicates the type of applications, and thus the desired priority by looking up Table II. The department of clients can be identified by source IP address from the IP header, and the priority category of the corresponding end user group can be mapped using Table III. Finally, the packet delay attribute can be detected by the network bridge itself. Based on these retrieved input values for a given packet, TSK-interpolation approach generates the priority values, which is in turn used to calculate the DSCP value. The calculated DSCP value is then re-encapsulated in the corresponding IP header for further processing. To facilitate network analysis, network monitoring software, such as Wireshark, was also launched on the bridge side to capture and monitor the network traffic. The bandwidth on the router WAN port was limited to 512Kbps in order to simulate a network congestion in the real world.

##### B. Experiment Results

The main purpose of this experimentation is to test whether the proposed TSK-interpolation system can successfully dynamically generate DSCP values according to network requirements for different situations. This experimentation was carried out by following three scenarios.

1) *Scenario 1*: This scenario considers a normal network condition that User 1 and User 2 were browsing web pages only, and the IP phone was making a telephone call via SIP service. At this point, both users were in the low priority group. The input feature values and the generated inference results by the proposed TSK-interpolation system are shown in Table V. According to Table I, the generated decisions 1 and 19 correspond to DSCP values 0 and 46, respectively. This had been proven by the captured data from Wireshark as shown in Figure 5; the DSCP value was 101110 in binary and 46 in decimal.

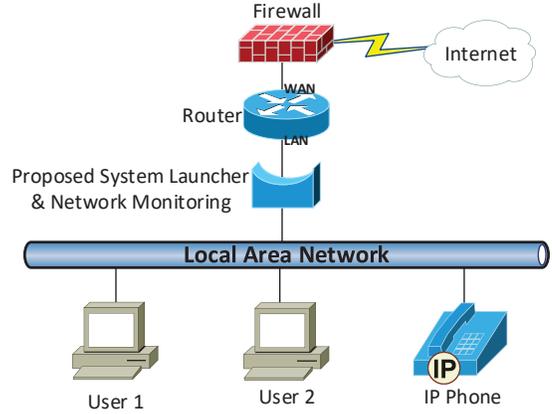


Fig. 4. The experimental environment

TABLE V. DETAILS FOR SCENARIO 1

	Delay	Input		Output
		Application Desired Priority	User Priority Group	Decision / Corresponding DSCP
User 1	Very Small	C2	Low	1 / 0
User 2	Very Small	C2	Low	1 / 0
IP Phone	Very Small	C10	Critical	19 / 46

```
Differentiated Services Field: 0xb8 (DSCP: EF PHB, ECN: Not-ECT)
1011 10.. = Differentiated Services Codepoint: Expedited Forwarding (46)
.... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
Total Length: 200
```

Fig. 5. DSCP value in a VoIP packet for Scenario 1

2) *Scenario 2*: From this point, User 1 enabled a P2P download, which took a significant amount of the available bandwidth and caused a network congestion. As a result, as expected, the network delay was increased which significantly limited the usable network resources and thus seriously affected the performance of the phone call over VoIP. This situation had been successfully detected by the system, and the DSCP values for corresponding packets were accordingly adjusted. The DSCP results for this new situation are shown in Table VI and the IP header structure captured by Wireshark is shown in Figure 6. In particular, the priority value was 110000 in binary, and 48 in decimal. This means that the network resources had been successfully dynamically allocated by the proposed system.

TABLE VI. DETAILS FOR SCENARIO 2

	Delay	Input		Output
		Application Desired Priority	User Priority Group	Decision / Corresponding DSCP
User 1	Very Large	C1	Low	1 / 0
User 2	Very Large	C2	Low	1 / 0
IP Phone	Very Large	C10	Critical	20 / 48

```
Differentiated Services Field: 0xc0 (DSCP: CS6, ECN: Not-ECT)
1100 00.. = Differentiated Services Codepoint: Class Selector 6 (48)
.... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
Total Length: 470
```

Fig. 6. DSCP value in a VoIP packet for Scenario 2

3) *Scenario 3*: In the above situation, the service quality of the web browser for User 2 was not guaranteed, due to the network congestion caused by P2P downloading. In order to get a better service quality for the web browsing, User 2 submitted a request which led to an increase of service level. The system input for different users based on the updated scenario and the generated DSCP results are listed in Table VII. It is clear from this table that the DSCP value of web browsing traffic had been increased to 10 from 0, which distinguished the priorities between User 1 and User 2. In this situation, the DSCP value of VoIP traffic still remained at 48, which means that the quality of a phone call over VoIP was not affected.

TABLE VII. DETAILS FOR SCENARIO 3

	Input			Output
	Delay	Application Desired Priority	User Priority Group	Decision / Corresponding DSCP
User 1	Very Large	C1	Low	1 / 0
User 2	Very Large	C2	Normal	3 / 10
IP Phone	Very Large	C10	Critical	20 / 48

The above experimental results show that the proposed system is able to provide an effective dynamic QoS solution for the enterprise network. The proposed system is highly suitable and flexible for different network environments. The rule base developed herein can be readily revised based on the requirement of different networks. Note that Cisco and other manufactures have their own smart QoS services, such as Cisco's AutoQoS. However these functions only work with certain models of network connected devices, but this is not the case for the proposed system.

## V. CONCLUSION

This paper presented a dynamic QoS solution for enterprise networks, which enhances the traditional QoS mechanisms by considering the data transmission requirements from applications and user groups. The proposed system is implemented by adapting the TSK fuzzy interpolation approach. The generated priority decisions are integrated in the DSCP fields of IP headers to manage the priority of data traffic. The experimental results show that the proposed system is able to dynamically advise data transmission priority for different situations to improve the quality of service. Although promising, the work can be further improved in the following areas. Firstly, the recently proposed data-driven rule base generation approaches [26], [27] may be employed in this work for automatic rule base generation and adaptation. Secondly, given that the proposed system is built upon TSK-interpolation, it is worthwhile to investigate how the proposed system may be developed using Mamdani-style rule bases with others FRI approaches.

## REFERENCES

- [1] Z. Wang. *Internet QoS: architectures and mechanisms for quality of service*. Morgan Kaufmann, 2001.
- [2] M. Beggas, L. Médini, F. Laforest, and M. T. Laskri. Towards an ideal service qos in fuzzy logic-based adaptation planning middleware. *Journal of Systems and Software*, 92:71–81, 2014.
- [3] T. Samak and E. Al-Shaer. Fuzzy conflict analysis for qos policy parameters in diffserv networks. *IEEE Transactions on Network and Service Management*, 9(4):459–472, 2012.
- [4] R. Zhang and J. Ma. Fuzzy qos management in diff-serv networks. In *Systems, Man, and Cybernetics, 2000 IEEE International Conference on*, volume 5, pages 3752–3757. IEEE, 2000.

- [5] S. Garba, B. G. Bajoga, M. B. Mu'azu, D. D. Dajab, and U. F. Abdu-Aguye. Development of an anfs-based qos model for a gsm service provider (mtn nigeria kano region network). In *3rd IEEE International Conference on Adaptive Science and Technology*, pages 165–173, 2011.
- [6] L. Kóczy and K. Hirota. Approximate reasoning by linear rule interpolation and general approximation. *International Journal of Approximate Reasoning*, 9(3):197–225, 1993.
- [7] L. T. Kóczy and K. Hirota. Interpolative reasoning with insufficient evidence in sparse fuzzy rule bases. *Information Sciences*, 71(1):169–201, 1993.
- [8] Z. Huang and Q. Shen. Fuzzy interpolation and extrapolation: A practical approach. *Fuzzy Systems, IEEE Transactions on*, 16(1):13–28, 2008.
- [9] L. Yang and Q. Shen. Towards adaptive interpolative reasoning. In *2009 IEEE International Conference on Fuzzy Systems*, pages 542–549, 2009.
- [10] L. Yang and Q. Shen. Adaptive fuzzy interpolation and extrapolation with multiple-antecedent rules. In *International Conference on Fuzzy Systems*, pages 1–8, 2010.
- [11] L. Yang and Q. Shen. Adaptive fuzzy interpolation. *Fuzzy Systems, IEEE Transactions on*, 19(6):1107–1126, Dec 2011.
- [12] Q. Shen and L. Yang. Generalisation of scale and move transformation-based fuzzy interpolation. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 15(3):288–298, 2011.
- [13] L. Yang and Q. Shen. Adaptive fuzzy interpolation with uncertain observations and rule base. In *2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011)*, pages 471–478, 2011.
- [14] S. Jin, R. Diao, C. Quek, and Q. Shen. Backward fuzzy rule interpolation. *IEEE Transactions on Fuzzy Systems*, 22(6):1682–1698, 2014.
- [15] L. Yang and Q. Shen. Closed form fuzzy interpolation. *Fuzzy Sets and Systems*, 225:1 – 22, 2013. Theme: Fuzzy Systems.
- [16] L. Yang, C. Chen, N. Jin, X. Fu, and Q. Shen. Closed form fuzzy interpolation with interval type-2 fuzzy sets. In *2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 2184–2191, 2014.
- [17] L. Yang and Q. Shen. Adaptive fuzzy interpolation with prioritized component candidates. In *2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011)*, pages 428–435, June 2011.
- [18] S. Chen and W. Hsin. Weighted fuzzy interpolative reasoning based on the slopes of fuzzy sets and particle swarm optimization techniques. *Cybernetics, IEEE Transactions on*, 45(7):1250–1261, 2015.
- [19] J. Li, L. Yang, H. P. H. Shum, G. Sexton, and Y. Tan. Intelligent home heating controller using fuzzy rule interpolation. In *UK Workshop on Computational Intelligence*, 2015.
- [20] S. Chen and Z. Chen. Weighted fuzzy interpolative reasoning for sparse fuzzy rule-based systems based on piecewise fuzzy entropies of fuzzy sets. *Information Sciences*, 329:503 – 523, 2016. Special issue on Discovery Science.
- [21] L. Yang, F. Chao, and Q. Shen. Generalised adaptive fuzzy rule interpolation. *IEEE Transactions on Fuzzy Systems*, (DOI: 10.1109/TFUZZ.2016.2582526), 2016.
- [22] J. Li, Y. Qu, H. P. H. Shum, and L. Yang. *TSK Inference with Sparse Rule Bases*, pages 107–123. Springer International Publishing, Cham, 2017.
- [23] T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to modeling and control. *Systems, Man and Cybernetics, IEEE Transactions on*, SMC-15(1):116–132, 1985.
- [24] T. Szigeti, C. Hattinigh, R. Barton, and K. Briley Jr. *End-to-End QoS Network Design: Quality of Service for Rich-Media & Cloud Networks*. Cisco Press, 2013.
- [25] S. Jeff, M. Charles, and M. Lauren. Broadband applications: Categories, requirements, and future frameworks. *First Monday*, 17(11), 2012.
- [26] J. Li, H. P. H. Shum, X. Fu, G. Sexton, and L. Yang. Experience-based rule base generation and adaptation for fuzzy interpolation. In *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 102–109, July 2016.
- [27] Y. Tan, J. Li, M. Wonders, F. Chao, H. P. H. Shum, and L. Yang. Towards sparse rule base generation for fuzzy rule interpolation. In *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 110–117, July 2016.