B&W Call Admission Control for Multimedia Communication Networks

Lain-Chyr Hwang，Ph.D
Department of Electrical Engineering
I-Shou University, Taiwan
lain@ isu.edu.tw

San-Yuan Wang , Ph.D
Department of Information Engineering
I-Shou University, Taiwan
sywang@ isu.edu.tw(corresponding author)

***Abstract*—**In the multimedia communication networks providing quality of service (QoS), an interface between the signal processing systems and the communication systems is the call admission control (CAC) mechanism. Owing to the heterogeneous traffic produced by diverse signal processing systems in a multimedia communication network, the traditional CAC mechanism that used only one CAC algorithm can no longer satisfy the aim of QoS CAC: Utilize the network resource to the most best and still satisfy the QoS requirements of all connections. For satisfying the aim of QoS CAC in the multimedia communication networks, this study proposed an innovative CAC mechanism called black and white CAC (B&W CAC), which uses two CAC algorithms. One of them is called black CAC controller and is used for the traffic with specifications more uncertain, which is called black traffic here. The other is call white CAC controller and is for the traffic with clearer specifications, which is call white traffic. Because white traffic is simple, an equivalent bandwidth CAC is taken as the white CAC. On the other hand, a neural network CAC (NNCAC) is employed to be the black CAC to overcome the uncertainty of black traffic. Furthermore, owing to more parameters needed in a QoS CAC mechanism, a hierarchical NNCAC is proposed instead of the common used NNCAC. Besides to accommodate more parameters, a hierarchical NNCAC can keep the complexity low. The simulation results show the B&W CAC can obtain higher utilization and still meet the QoS requirements of traffic sources.

Keywords*-Multimedia communication networks, Quality of service, Call admission control (CAC), B&W call admission control (CAC), Neural network;*

*****

I. INTRODUCTION

Different signal processing systems produce different traffic sources for communication networks. For example, the traffic output of an MPEG encoder belongs to the traffic of variable bit rate (VBR). The traffic characteristic of VBR traffic in general cannot be predicted beforehand. On the other hand, for some data sources their characteristics are simple and may be known beforehand. Owing to the diversity of applications in the Internet, the Internet will be a multimedia communication networks that transmit regular data, video data and audio data, etc. in the future. In the multimedia communication networks, the traffic sources are diverse in terms of traffic characteristics and requirements of quality of service (QoS).

In order to provide the differentiated QoS, the IETF has developed two service architectures for the Internet. The first one is the Integrated Services (IntServ) [1] and the other one is the Differentiated Services (DiffServ) [2]. The IntServ architecture is proposed for differentiated QoS in per flow base. However, owing to millions of flows in the Internet, this approach will result in much more status information kept in the router. In this way, routers become the bottlenecks in the Internet. Consequently, when the number of flows increases to some degree, the performance of the Internet will degrade rapidly. That is why it is said the IntServ is lack of scalability. The DiffServ is used to solve this problem. The scope of the DiffServ focuses on the class, instead of the flow in the IntServ, so it will not be influenced by the scale of the Internet. Although the scalability of IntServ is week, it still suitable to the enterprise networks. No matter what architecture is used in a multimedia communication network, the network resource allocation is an important issue to provide the QoS. In a multimedia communication network, the network resource may be dynamically allocated, which is a property of packet switching network. However, a minimum resource is needed for each flow/class to meet the QoS requirement for that flow/class. This specification is also described explicitly in the corresponding RFC. For example, RFC 2597 [3]: Four AF classes are defined, where each AF class is in each DS node allocated a certain amount of forwarding resources (buffer space and bandwidth), and RFC 2598 [4]: The EF PHB is defined as a forwarding treatment for a particular diffserv aggregate where the departure rate of the aggregate's packets from any diffserv node must equal or exceed a configurable rate. That is, there is a guaranteed bandwidth needed for no matter each AF class or the EF class.

For bandwidth allocation, call admission control (CAC) plays an important role in multimedia communication networks. CAC mechanism is the interface between the signal processing system and the communication network. It is also the first outpost of the QoS architecture. This study focused on the CAC mechanism of multimedia communication networks. First, we must understand the diversity of the traffic sources in the multimedia communication networks. Under this situation, the traditional CAC may not do a good job. The aim of this paper is to find a new mechanism of CAC, which is more suitable for the diverse traffic in the multimedia communication networks.

Conventional CACs are based on mathematical analyses of offered traffic characteristics and service qualities [5-9]. In the equivalent bandwidth (EB) scheme, it based on the traffic characteristics from a new call and network performance statistics to calculate the equivalent bandwidth required by the new call. However, it is difficult to acquire complete characteristics of traffic, so it is not easy to determine accurately the equivalent capacity. Because of modeling, approximation and the unpredictable statistical fluctuations of the system, decision errors always occur in these control schemes and results in performance degradation.

The neural-network-based CAC can solve this problem, and it is unnecessary for users to specify the values of traffic descriptors in detail. Neural networks have been applied widely to deal with traffic-control-related problems in communication networks [10-13]. The self-learning capability of the neural network is used to characterize the relationship between input traffic and system performance. In [10], Cheng and Chang proposed a neural-network connection-admission-control (NNCAC) method for traffic control, which uses three preprocessed input parameters to simplify the training process and to improve the controller performance. In [11], Hiramatsu proposed a connection-admission controller, which uses a neural network. This admission controller employed the offered traffic characteristics, QoS requirement and actual network-operation-performance measures to decide whether to accept or reject a call-setup request. The results showed that the neural network learned a complicated boundary for a call-acceptance decision. Hah [12] investigated an efficient NNCAC mechanism for ATM networks with heterogeneous arrivals. NNCAC efficiently and accurately estimates the cell delay and cell loss ratio of each class in real time by means of a pre-trained neural network. In [13], Fontaine and Smith proposed a CAC scheme, which is based on a consideration of fuzzy logic and artificial neural networks (ANNs). The ANN is used in the learning phase to tune the fuzzy system automatically. However, in most of the proposed neural-network approaches, all parameters were selected as inputs dependently. Thus, the neural-network dimensions will increase as the number of parameters grows.

The objective of multimedia communication networks is to provide service to every kind of traffic source. Traffic sources are heterogeneous in multimedia communication networks. We think single kind of CAC algorithm cannot meet all the kinds of input traffic, no matter equivalent bandwidth (EB) algorithms or neural network algorithms, so this study suggest a CAC mechanism that combine both of EB and neural network. The motivation of this combination comes from individual characteristics of EB CAC and neural network CAC (NNCAC). EB CAC is efficient for the traffic source with more clear characteristics. On the other hand, NNCAC is good at the traffic type with less clear characteristics. This new CAC mechanism adopts more than one control algorithm simultaneously to admit new calls with different CAC algorithm for different type of traffic source. It is named Black & White Call Admission Control (B&W CAC), which includes two kinds of admission control algorithm. One is a hierarchical neural network algorithm for "black" traffic type, of which the characteristics are not very clear. Furthermore, the architecture of the neural network is with multiple layers. The purpose of multiple layers is to reduce the dimension of the neural network. With this architecture, more parameters can be adopted to make a good admission decision, while the dimension of neural network is still kept reasonable. The other is the equivalent bandwidth algorithm for the "white" traffic type, of which the characteristics are clearer.

The remaining organization of this paper is as follows. Section 2 describes some preliminaries that should operate in coordination with CAC. The core of this paper, B&W CAC, is specified in section 3, including the hierarchical neural network and a summary of equivalent bandwidth CAC. Section 4 takes a simple example to illustrate the good performance of B&W CAC. Finally, some concluding remarks and future works are given in Section 5.

II. The preliminaries

CAC is an important issue in the field of network resource allocation and QoS provision. When a connection makes a call setup request, CAC controller would decide whether to accept or reject the call according to the traffic parameters and the QoS requirements of the new call and the system performance statistics (e.g. load, queue length). A call is accepted only if the network has enough resource to provide the QoS requirements of the new connection request without affecting the QoS requirements of the connections already established in the network.

CAC cannot finish the admission decision alone without assistance of other mechanisms or protocols. For example, the network meters and the signaling protocols, e.g. RSVP in IntServ, are necessary for a CAC controller to make a decision to still keep the QoS provided by the multimedia communication network. The achievement of QoS cannot be made out of nothing. For providing QoS, not only the network statuses but also the traffic statuses, including traffic parameters and the parameters corresponding to QoS, should be considered in the CAC procedure. Different from traditional CAC procedures, a QoS CAC controller should consider the QoS requirements of the traffic to make an accurate decision. However, how the QoS parameters can be taken into account is still an open question. This issue strongly depends on the signaling protocol. The signaling protocol should carry the relevant information from some other nodes and even from some other network domains, if inter-domain connections are involved, to the CAC controller to assistant the admission decision. The study of signaling protocol is out of the scope of this CAC study, so it is not discussed here.

In this study, we assume all the relevant mechanisms, e.g. the network meters and the signaling protocols, operate properly. The focus of this B&W CAC is put on the processing of the relevant information collected from network and traffic.

III. B&W CAC

A structure of B&W CAC is shown in Figure 1. According to different types of traffic sources, the CAC can be implemented with two different admission control algorithms for different traffic types. The black control algorithm is applied to black traffic source, whose traffic characteristics are uncertain and QoS requirements may be strict or diverse, such as video source and voice source. The transmission rates of traffic sources of this type are more unpredictable. This black CAC mechanism is implemented by a hierarchical neural network. On other hand, the white control algorithm is applied to white traffic source, whose traffic characteristics are clearer and QoS requirements may be looser, such as file transmission.

This white CAC mechanism is implemented by an EB CAC controller.

When a new call is coming, it is classified into either black traffic source or white traffic source by B&W classifier. According to black or white traffic type, the CAC controller adopts different call admission control algorithm. We adopt a hierarchical neural network as black CAC algorithm. The neural network can be trained to fit the uncertainty of the black traffic source because of self-learning capability. This hierarchical neural network will be described in detail below. In the white CAC algorithm, we adopt the equivalent bandwidth method in [9], which is simple and efficient, and has been investigated completely by other researchers. The network status measurer periodically collects the network performance, such as load and loss ratio, and feeds to the CAC controller for the admission decision. In this structure, the black and the white CAC algorithms have their respective buffers.

Owing to B&W CAC uses a structure of neural network different from the common used, we will introduce this hierarchical neural network in detail in the next subsection. Furthermore, owing to the EB CAC controller is well-known, only some summary of equivalent bandwidth is given in the next second subsection.

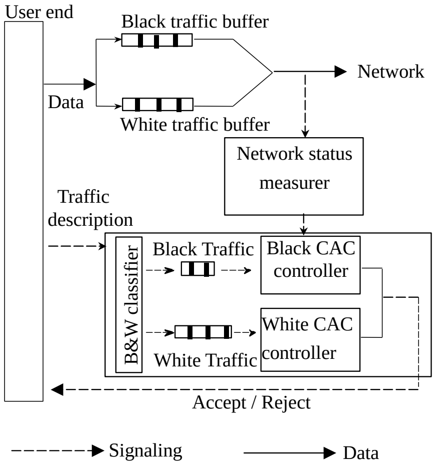

Figure 1 B&W CAC controller structure

A. Hierarchical neural network

For the purpose of obtaining a more accurate admission decision, there are more parameters fed into the CAC controller. Those parameters may belong to different "type." That is, some parameters may be uncorrelated to the other ones. Under this situation, it is more suitable to use multiple hierarchies instead of the common used single hierarchy, while a neural network is used to be the CAC controller. In the first hierarchy, parameters of different types are fed into different neural networks. Their outputs are then fed into the neural network of the second hierarchy. In fact, the advantage of the usage of a hierarchical neural network is twofold. It gains both the accuracy and the efficiency, because it can consider more input parameters, and still keep a low complexity. A simple comparison between the single-hierarchy neural network and the two-hierarchy neural network is given after the description of the hierarchical neural network.

The hierarchical NNCAC has two hierarchies as shown in Figure 2. The first hierarchy consists of two neural networks. The second hierarchy is constituted by one neural network. The whole NNCAC has two types of inputs: one type is corresponding to traffic characteristics and QoS requirements. The other type is corresponding to network status. The parameters of different types are treated individually by the two neural networks of the first hierarchy. Then, the two outputs, called traffic descriptor, *tr*, and network parameter (system performance parameter), *nt*, are fed into the neural network of the second hierarchy. *tr* is the output of the neural network with four inputs of traffic characteristics and QoS requirements, which include peak rate *p*, mean rate *m*, delay requirement *d*, and loss ratio constrain *l* of the new call. *nt* is the output of the neural network also with four inputs of network statuses, which include network utilization (load) ρ, queue length of buffer *q*, queue length variation Δ *q*, and statistic of loss ratio *s*. Here, the queue length variation is included into the input of NNCAC in order to catch the dynamics of the network status.

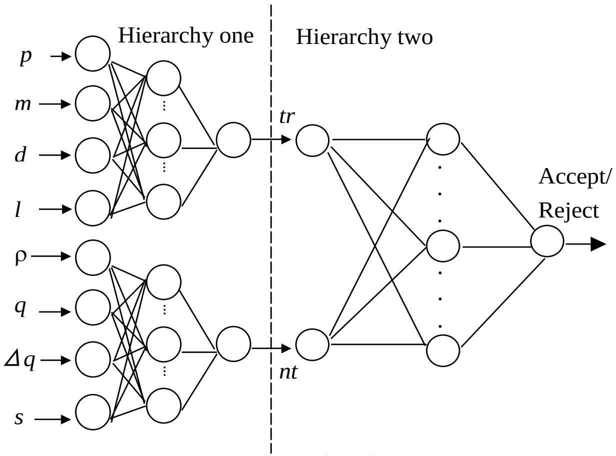

Figure 2 Hierarchical NNCAC

In other words, the NNCAC is a hierarchical neural network, which has two parts of inputs: traffic descriptors (traffic characteristics and QoS requirements declared by the new call) and network parameters (statistics gathering from network). As shown in Figure 2, traffic descriptors (p, m, d, l) and network parameters (, q, Δq, s) are preprocessed, and the outputs tr and nt are fed into another neural network. By way of considering more parameters, we hope to improve the decision accuracy of the CAC, but not to affect the speed of execution. That is, the increase of accuracy does not increase the complexity of computation.

Each neural network module of the NNCAC is a multi-layered feed-forward neural network. In the NNCAC mechanism, we employ the back-propagation learning method [14] for the off-line training of the neural network. The basic idea of this algorithm is a gradient method in which the weight vector is updated by gradient of the error function.The objective of the back-propagation learning algorithm is to minimize the decision error *E* by recursively adjusting its weight in each layer. The general form of decision error *E* is as follows

$$E = \frac{1}{2}\sum_{i}\left|y_i - F(x_i; w)\right|^2 \quad , \qquad (1)$$

where y_i is the desired output when the input (vector) is x_i and the set of weight is w, and F represents the output function of the neural network.

Considering an M-layer feed-forward neural network, each layer has a number of processing elements called neurons, which are interconnected via links with adaptive weights. There are at least two layers in a neural network, i.e. the input layer and the output layer. Between the input and output layers, there may be some hidden layers. A simple back-propagation neural network has three layers: input layer, hidden layer and output layer. The input layer just transmits the input values to the next layer. Neurons in the hidden layer and output layer perform two operations--weighted summation and transformation. The summation of the j-th neuron in the k-th layer, denote by S_j^k, is expressed by

$$S_j^k = \begin{cases} x_j, & k = 1, \\ \sum_i o_i^{k-1} w_{ij}^k, & 2 \le k \le M, \end{cases} \qquad (2)$$

where o_i^{k-1} is the output of the i-th neuron in the layer k-1 and w_{ij}^k is the weight of the link between the i-th neuron in layer k-1 and the j-th neuron in layer k. Furthermore, S_j^k is transformedinto an output o_j^k via an activation function $G(\cdot)$ [10]. That is, the output o_j^k can be given by

$$o_j^k = \begin{cases} S_j^k, & k = 1, \\ G(S_j^k), & 2 \le k \le M, \end{cases} \qquad (3)$$

where $G(x)$ is generally given by

$$G(x) = \frac{1}{1+e^{-x}}, \qquad (4)$$

for hidden layers and

$$G(x) = U(x) = \begin{cases} 1, & x > 0, \\ 0, & x \le 0, \end{cases} \qquad (5)$$

for the output layer of NNCAC. Finally, if the final output of NNCAC is 1, the new call is accepted. On the other hand, the new call is rejected, if the output is 0.

The adjustment of weights is based on a steepest-descent algorithm [14]. It can be expressed as

$$w_{ij}^k(t+1) = w_{ij}^k(t) - c \left. \frac{\partial E}{\partial w_{ij}^k} \right|_{w_{ij}^k = w_{ij}^k(t)}, \qquad (6)$$

where c is the learning constant, which is positive and small. By repeating this weight updating for various example pairs, it is expected that the total error for all input vectors is minimized.

The training of the neural network is an important step when NNCAC is employed. The training procedure is first by off-line training and then by on-line training if necessary. In general, the on-line training is triggered periodically to make the NNCAC controller to capture the dynamics of traffic and network. No matter which training procedures, the first step is to collect the training data into a training data table. Each entry of the training data table includes the input parameters and the desired output corresponding to those inputs. The desired output $\hat{o}$ is 1 if the QoS of the call is met after the call is accepted and is 0 otherwise. It can calculated by

$$\hat{o} = U((d - \hat{d})(l - \hat{l})), \qquad (7)$$

where $\hat{d}$ and $\hat{l}$ are the statistics of the delay and loss ratio respectively of the call. The statistical duration could be the whole period or partial period of the holding time of the call. Note that the $\hat{d}$ needs more specification. It could be the mean, the mean plus some standard deviations, or the maximum delay, depending on what delay performance is required. Equation (7) illustrates that the desired output is 1 if both QoS requirement, i.e. delay requirement and loss requirement, are matched, and is 0 if any QoS requirement is not matched. If more QoS requirements are specified, the argument of unit step function $U(x)$ will be more terms multiplying together. Additionally, in order to get the accurate output, or the optimal weights, by back-propagation, the training data table should be updated from time to time and the entry for each training should be randomly selected. In general, the leaky pattern table selection method [15] is employed.

Owing to the vigorousness of multimedia communications, the consideration of QoS is the trend of the future communication networks. In this way, the QoS parameters should be taken into account when a call is setup. Consequently, the input of a CAC controller contains the traffic description (characteristics and requirements) and the network status. The description of the new call is independent of the current network status; although they will be correlative in the future time if the call is accepted and they will cooperate to influence the CAC decision. Utilizing thisproperty, a hierarchical NNCAC is proposed to decrease the complexity of the neural network and, however, still to be able to contain more parameters input to the controller. In the following, a simple comparison is done to illustrate the reduction of complexity when a hierarchical NNCAC is used instead of a common single-hierarchy NNCAC. In a neural network, the training procedure is to find an optimal set of weights. The more weights result in more complexity. Considering a three-layered neural network with 8 input neurons, 30 hidden neurons and 1 output neuron, it has 270 connecting weights in total. On the other hand, if our proposed hierarchical NNCAC is used, we can evenly divide the 30 hidden neurons between the two neural networks of the first hierarchy and the second hierarchy still uses 30 hidden neurons. Finally, there are only 240 connecting weights in total. It is obvious that single-hierarchy NNCAC uses one-eighth weights more than our hierarchical NNCAC. Note that the connections between the output neurons of hierarchy one and the input neurons of hierarchy two are virtual, so no connecting weights are needed. The output neuron of hierarchy one and the input neuron of hierarchy two may be on a same neuron. The expression on figure 2 is used to illustrate the concept of hierarchy more clearly. Obviously, our NNCAC has fewer connecting weights. It confirms our purpose that to concern more input parameters and not to be more complex.

B. *Equivalent bandwidth call admission control*

The white CAC scheme utilizes an equivalent-bandwidth-based call admission control, which uses statistical bandwidth as the decision criterion. The statistical bandwidth is computed by using diffusion approximations [16,17]. The basic approach is abstracted as follows. When a new call request arrives at, the aggregated traffic characteristics, including the characteristics of the new call and those calls having already existed, are used to find a statistical bandwidth by a certain equation derived from the diffusion model. If the statistical bandwidth is less than or equal to the output link capacity, the new call is accepted. Otherwise, the call is rejected. The mathematical expression of the diffusion equation used in this approach includes the users' loss requirements and traffic characteristics, and the available buffer size of the network. More detailed description can be referred to [16,17].

An implementation of classifier

The input of the classifier is the signaling packet, not general data packet, so the classifier should cooperate with the signaling protocol. This sub-section demonstrates a possible implementation, which involves with a possible signaling protocol. But our purpose is not to design a signaling protocol in detail, because it is out of the scope of this paper. Similar to general signaling protocol, e.g. RSVP [18,19], there should be some messages carrying some specifications about the traffic specifications and the reservation resource specifications. For example, in RSVP, PATH message and RESV message are used to carry the traffic specifications (TSpec) and the reservation specifications (RSpec). RSVP is a destination activating reservation protocol. That is, the reservation is activated by the destination end. It is not necessary for B&W CAC. Hence, a signaling protocol is suitable, if it can carry the following specifications: 1) Application identification, which may be the combination of the identification of transport layer and the port number; 2) Traffic characteristics, e.g. (but not limited to) peak rate and mean rate; 3) QoS parameters, e.g. delay requirement and loss requirement. The specifications may include more than the above. It depends on what kind of CAC algorithm is used and what parameters are needed in the algorithm. The application identification described above is used to identify the type of traffic type. It should contain enough information for classifier to judge the traffic type, so it may contain some information of application layer, if the identification of transport layer and the port number are not enough. The classifier should have the ability to identify to which kind of the traffic belongs by the application identification, and then passes the corresponding traffic characteristics and QoS requirements to the appropriate CAC controller.

IV. PERFORMANCE EVALUATION

In this section, we evaluate the B&W CAC scheme by comparing it with equivalent-bandwidth-based CAC scheme in [16,17] and neural-networks-based CAC scheme. Each neural network in B&W CAC or NNCAC uses a three-layered architecture with 30 hidden nodes and a back-propagation learning algorithm. Owing to there are two neural networks in the first hierarchy of the neural network in B&W CAC controllers, each of the two neural networks has 15 hidden nodes. Besides, both the neural networks of the second hierarchy of the single-hierarchy NNCAC controller have 30 hidden nodes and the neural network in B&W CAC controller.

The link capacity of the network is normalized to be 1. And for simplicity, the packet length is fixed at 1 length unit, or called 1 unit in short. The time unit, called tick, is defined to be the time for the network to transmit 1 packet (equivalently 1 unit). For decreasing the complexity of simulation environment, we avoid to consider the scheduling mechanism between the black traffic and the white traffic. Instead, the link capacity is allocated to the two traffic types evenly, i.e. both of them have bandwidth 0.5. There is a separate finite buffer for each traffic type and the buffer size is 150 units.

As regards the traffic models, we use the Bernoulli process as the arrival process In this section, we evaluate the B&W CAC scheme by comparing it with equivalent-bandwidth-based CAC scheme in [16,17] and neural-networks-based CAC scheme. Each neural network in B&W CAC or NNCAC uses a three-layered architecture with 30 hidden nodes and a back-propagation learning algorithm. Owing to there are two neural networks in the first hierarchy of the neural network in B&W CAC controllers, each of the two neural networks has 15 hidden nodes. Besides, both the neural networks of the second hierarchy of the single-hierarchy NNCAC controller have 30 hidden nodes and the neural network in B&W CAC controller.

The link capacity of the network is normalized to be 1. And for simplicity, the packet length is fixed at 1 length unit, or called 1 unit in short. The time unit, called tick, is defined to be the time for the network to transmit 1 packet (equivalently 1 unit). For decreasing the complexity of simulation environment, we avoid to consider the scheduling mechanism between the black traffic and the white traffic. Instead, the link capacity is allocated to the two traffic types evenly, i.e. both of them have bandwidth 0.5. There is a separate finite buffer for each traffic type and the buffer size is 150 units.

As regards the traffic models, we use the Bernoulli process as the arrival process of the white traffic and the ON-OFF model as the traffic model of the black traffic. The ON-OFF model is used to model the black traffic, because it is more bursty and more uncertain. The traffic generation process of black traffic source follows a discrete-time ON-OFF process. During the ON state, packets are generated at rate *R* packets per tick; while during the OFF state, no packets are generated. The transition rate from OFF to ON, denoted by α, and that from ON to OFF, denoted by β (see figure 3). In other words, if a source is in ON (OFF) state, then it will be in OFF (ON) state with probability β (α) next tick. In order to simplify the number of variables, the transition rate from OFF to ON and that from ON to OFF are selected to sum to 1. That is, $\alpha+\beta=1$, $\alpha=1-\beta$.

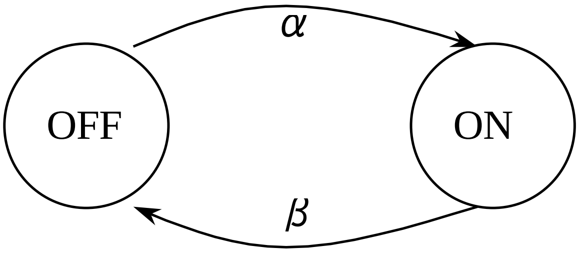

Figure 3 ON-OFF model for black traffic source

The packet generation rate *R* of each black traffic source is a random variable with uniform distribution, among 1/30, 1/40 and 1/50, i.e. *R* is equal to 1/30, 1/40, or 1/50 with equal probability 1/3. Also, the transition rates and the holding time of a call are uniform distribution. The transition rate from ON to OFF (β) is randomly selected among 0.2, 0.4, 0.6 and 0.8. The holding time of a call of black traffic source is randomly selected among 600, 1200, 1800, 2400 and 3000 ticks. Furthermore, the packet generation rate of white traffic source is a random variable with equal probability among 1/60, 1/80, 1/100. The holding time of a call of white traffic source is randomly and uniformly selected among 500, 1000, 1500, 2000 and 2500 ticks. The QoS requirements of a call are loss ratio and delay requirement. Loss ratio is set to 0.01 and delay requirement is set to 50 ticks. The simulation is terminated when the system time is 5,000,000 ticks.

Figure 4 shows the utilization of a network employing the B&W CAC scheme and figure 5 shows the utilizations of a network employing the single-hierarchy NNCAC and the EB CAC. To protect the readability of the figures, the results of NNCAC and the results of B&W CAC are plotted on different figures. It is obvious that the utilizations of NNCAC and B&W CAC are much higher than that of EB CAC. This shows the conservative property of equivalent bandwidth such that the utilization is low. Furthermore, in average, the utilization of B&W CAC is higher than that of NNCAC. Beside the higher utilization, B&W CAC can make the network utilization more stable than NNCAC does. This phenomenon can be illustrated by the violent fluctuation of the curve of NNCAC on figure 5.

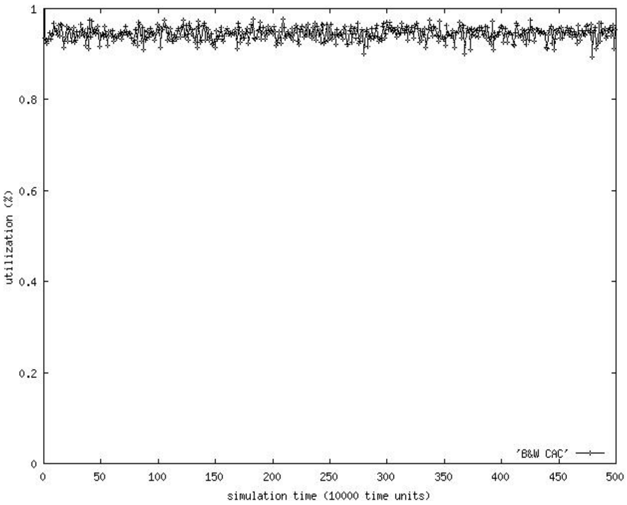

Figure 4 Network utilizations of B&W CAC

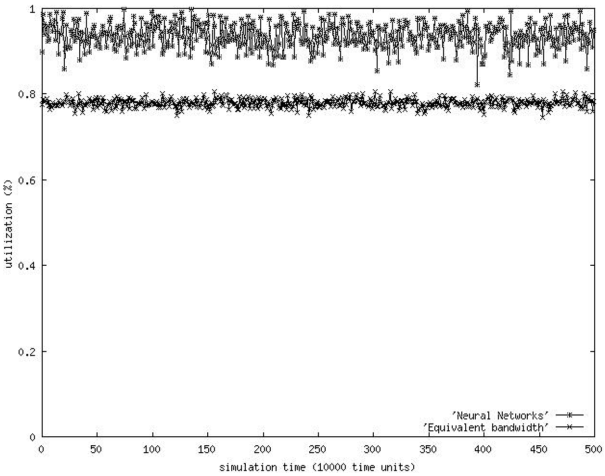

Figure 5 Network utilizations of EB CAC and NNCAC

Figures 6 and 7 show the comparisons of delay times and loss ratios, respectively, among B&W CAC, NNCAC and EB CAC. They are used to specify the influence of CAC on QoS. From figure 6, it is seen that the delay times for NNCAC is worse than the other two CAC schemes, and even not to satisfy the delay requirement. Similarly, from figure 7, the loss performance of NNCAC is the worst, while the loss ratios of the other two schemes are both close to 0. That is because the NNCAC cannot response the network status rapidly and results in the utilization sometimes up and sometimes down (see figure 5). This also results in that there are sometimes too many connections in the network and sometimes too few connections in the networks. This instability finally causes the performance low. When there are too many connections, the packets are dropped seriously and the packets buffered wait a longer time. Even when the number of connections cools down, the mean waiting times and the loss ratios cannot be descended from the very high values effectively. Note that the data on the figures are statistics during every 10000 ticks. We think the cause of the inability of NNCAC partially comes from the heterogeneity of the traffic. That is why this paper separates the traffic into black traffic and white traffic, and treats them individually. With regarding to the performance comparison between the results of EB CAC and B&W CAC, it can be seen that the variation and the mean of the delay time of B&W CAC is higher than those of EB CAC. Although the performance of EB

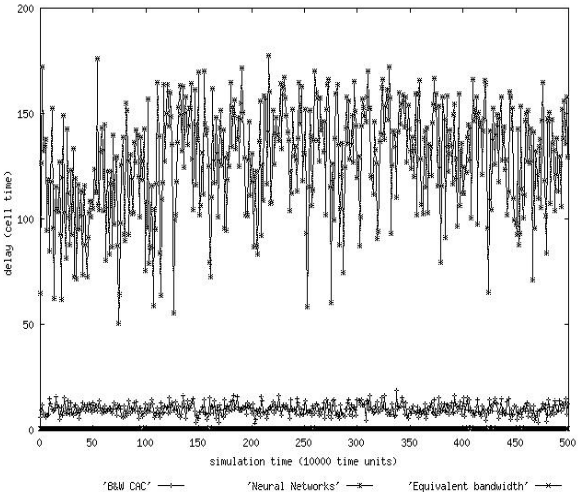

Figure 6 Comparison of delay performance

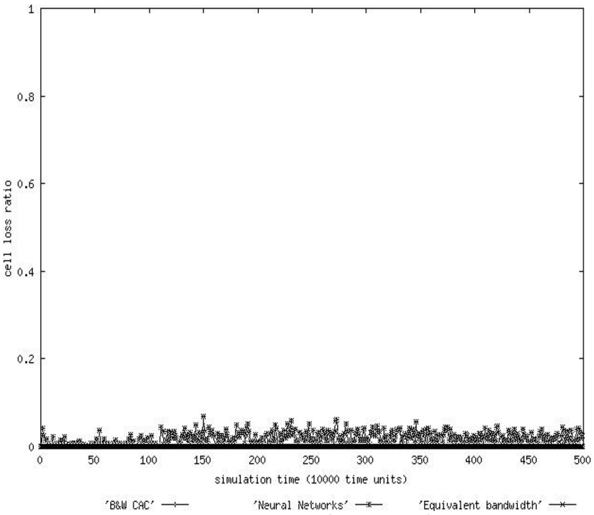

Figure 7 Comparison of loss performance

CAC seems a little better than that of B&W CAC, however, the results of B&W CAC still satisfy the QoS requirements. Recall that B&W CAC gets much higher utilization than EB CAC does. Evaluating wholly, B&W CAC perform much better than the other two CACs do. It tightly conforms to the objective of QoS CAC: Utilize the network bandwidth to the most best under the condition of matching the QoS requirements.

V. CONCLUSION AND FURTHER STUDY

In this paper, we proposed a B&W CAC scheme, which employs two CAC algorithms. Previously, all the studies just implement one CAC algorithm to judge whether to accept or reject a new call for all kinds of traffic sources. However, in the trend of multimedia communication, there are heterogeneous traffic sources. Furthermore, providing differentiated treatment is an important goal in future multimedia communication networks. Therefore, to treat different type of traffic source with different CAC algorithm is a reasonable and innovative approach. Owing to the parameters of traffic may include the characteristic parameters and the QoS parameters, the number of parameters input to the QoS CAC controller will get more in the future. According to different type of traffic sources and different QoS requirements, B&W CAC scheme applies two CAC algorithms, equivalent bandwidth and hierarchical neural network. The simulation results show the B&W CAC can keep the network to meet the QoS requirements of traffic sources and also gets higher utilization. In a word, B&W CAC can fully utilize the system bandwidth and still satisfy the QoS requirements. Although the single-hierarchy NNCAC scheme gets not bad utilization, the delay and the loss requirement may be not satisfied. This also specifies the demand of B&W CAC, because for heterogeneous traffic sources the NNCAC may not do a good job.

This paper has not yet finished the study of all kinds of B&W CAC mechanisms. It just reveals a new direction of CAC in the future communication networks, in which the traffic sources will be very diverse. Which CAC algorithm should be taken as black CAC and which as white CAC can be further studied. Another important issue in B&W CAC is the classification of traffic. This involves the signaling protocol. This paper also demonstrates a possible implementation. Here does not design a complete signaling protocol. The design of signaling protocol is also worthy to investigate.

ACKNOWLEDGEMENT

This work was supported in part by the Ministry of Science and Technology of the Republic of China under Contract MOST108-3116-F-042A-004-CC2.

REFERENCES

[1] R. Braden, D. Clark, and S. Shenker, “Integrated services in the Internet architecture: an overview,” Internet RFC 1633, June 1994.

[2] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, “An architecture for differentiated services,” Internet RFC 2475, Dec. 1998.

[3] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, “Assured forwarding PHB group,” Internet RFC2597, June 1999.

[4] V. Jacobson, K. Nichols, and K. Poduri, “An expedited forwarding PHB,” Internet RFC 2598, June 1999.

[5] R. Guerin, H. Ahmadi, and M. Naghshineh, "Equivalent capacity and its application to bandwidth allocation in high-speed networks," IEEE JSAC vol. 9, no. 7, pp. 968-981, Sep. 1991.

[6] Z. Zhang and A. S. Acampora, "Equivalent bandwidth for heterogeneous sources in ATM networks," Proc. IEEE ICC 94, vol. 2, pp.1025-1031, 1-5 May, 1994.

[7] Z. Dziong, M. Juda, and L. G. Mason, "A framework for bandwidth management in ATM networks-aggregate equivalent bandwidth estimation approach," IEEE/ACM Trans. on Networking, vol.5, pp. 134-147, Feb. 1997.

[8] E. Oki, N. Yamanaka, K. Shiomoto, and S. D. Moritra, “A new multiple QoS control scheme with equivalent-window CAC in ATM networks,” IEICE Trans. Commun., vol. E81-B, no.7, pp.1462-1474, July 1998.

[9] E. Gelenbe, X. Mang, and R. Onvural, “Bandwidth allocation and call admission control in high-speed networks,” IEEE Comm. Mag., pp. 122-129, May 1997.

[10] R. G. Cheng and C. J. Chang, “ Neural network connection-admission control for ATM networks,” IEE Proc. Commun., vol.144, no.2, pp.93-98, April 1997.

[11] A. Hiramatsu, “ ATM communications network control by neural networks,” IEEE Tran. On Neural Networks, vol.1, no.1, pp.122-130, March 1990.

[12] J. M. Hah, P. L. Tien, and M. C. Yuang, “Neural-network-based call admission control in ATM networks with heterogeneous arrivals,” Computer Commun., vol. 20, issue 9, pp.732-740, Sep. 1997.

[13] M. Fontaine and D.G. Smith, “Automatically designed fuzzy system for connection admission control in ATM networks,” IEE Proc. Commun., vol.146, no2, pp.113-119, April 1999.

[14] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, Learning internal representations by error propagation, Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol.1, ch. 8, pp. 318--362, MIT Press, Cambridge, 1986.

[15] A. Hiramatsu, “ATM communication network control by neural networks,” IEEE Trans. Neural Networks, vol. 1, no. 1, pp. 122-130, March 1990.

[16] E. Gelenbe, X. Mang, and R. Onvural, “Bandwidth allocation and call admission control in high-speed networks,” IEEE Comm. Mag., pp. 122-129, May 1997.

[17] E. Gelenbe, X. Mang, and R. Onvural, “Diffusion based statistical call admission control in ATM,” Perf. Eval., vols. 27-28, pp. 411-436, Oct. 1996.

[18] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Resource ReSerVation Protocol (RSVP)--Version 1 Functional Specification," Internet RFC 2205, Sep. 1997.

[19] R. Braden and L. Zhang, "Resource ReSerVation Protocol (RSVP)--Version 1 Message Processing Rules," Internet RFC 2209, Sep. 1997.

[20] D. Kornack and P. Rakic, “Cell Proliferation without Neurogenesis in Adult Primate Neocortex,” Science, vol. 294, Dec. 2001, pp. 2127-2130, doi:10.1126/science.1065467.