Integrated Proactive Admission Control Technique For both UDP And TCP Traffic Flows

Lakshmanan Senthilkumar,
Dept. of CSE, B.S.A.Crescent Engineering College,
Vandalur, Chennai – 600 048, South India.
Email: senthilkumarl@rediffmail.com

Venkadachalam Sankaranarayanan
Director, B.S.A.Crescent Engineering College,
Vandalur, Chennai – 600 048, South India.
Email: sankarammu@yahoo.com

Abstract—**Real time traffic adopting UDP at the transport layer needs some quality of service. It is offered through an admission control scheme. This paper adopts one such scheme which is extended for elastic traffics adopting TCP at the transport layer. The proposed scheme operates on reserving network resources on a proactive manner. It is based on the principle of telephone networks Erlang-B model. The blocking probability measured is used as a flow admission decision parameter. The effectiveness of the proposed admission control algorithm is determined here through simulation. It offers a fair admission rate to both UDP and TCP traffic flows. It also results in a better bottleneck link utilization at a comparatively lower overhead traffic.**

Index Terms—**Admission, QoS, TCP, UDP, and Erlang**

I. INTRODUCTION

Internet in its original form is said to offer "Best Effort" service model. In this the network is considered to be unreliable, doing its best to deliver the packets to the destination without any drop. To incorporate some degree of reliability in the above service an end-to-end quality of service (QoS) scheme in the form of TCP was introduced. Such service had the mechanism to detect packet loss and use retransmission as the solution. It did not focus on reducing the end-to-end transmission delays. The two prominent applications which were deployed initially using TCP were Telnet and FTP. With the growth of Internet and its ubiquitous presence, new applications matching the contemporary demand are being deployed on it. The modern day applications range from live weather updating report to live audio and video telecasting. All these applications are time sensitive and use inter-packet delay and jitter as the QoS parameter. Thus applications run with UDP as the transport layer protocol. The two models proposed exclusively to offer QoS in such IP networks are: IntServ and DiffServ.

IntServ provides strict QoS to the flow of packets by reserving network resources across the path before admitting them. It results in a per flow state maintenance at the core routers. There by making the model non-scalable as it imposes additional burden on routers, whose main activity is to forward the packets at the earliest and also through a shortest possible path.

DiffServ is a solution that has been proposed to overcome the scalability problem of. This service model proposes to provide QoS to flow of packets by first classifying them as different aggregates. An aggregate is usually a user specified entity. It can be based on traffic characteristics, users, organization etc. The DSCP field in IP packet header is marked according to the aggregate to which the packets belong to at the edge router. Such aggregates are then treated based on policies called as PHB (Per Hop Behaviors) defined at each hop across the path. Packet forwarding based on such limited number of aggregate classification does not over burden the core routers.

The mode of QoS offered to the flow of packets in DiffServ can be treated as providing QoS based on "*outside view of network*". It only tries to confine the flow characteristics according to the SLA and not based on the internal state of the network, which we call here as providing QoS based on "*Inside out View*", in which we need to also confine the flow of packets with respect to the current internal state of the network. To provide such QoS, an admission control strategy may be used.

Several admission control strategies have been proposed. They have been mainly studied in the context of UDP traffic. A new UDP traffic flow would get rejected in them when it is found to degrade the quality of the ongoing flow. In this paper we propose an integrated admission control scheme for both UDP and TCP traffic flows. The reason is to ensure that neither traffic flow varieties are affected by their conventional characteristics. UDP traffic flow when uncontrolled would monopolize the link and a TCP flow which is self adapting may drastically change their sending rates which could terminate the session (or) drastically increase the inter-packet delay. It is also said that there is a considerable inefficiencies with TCP flows which use packet loss as the means to probe for their network bandwidth share. Further retransmission of lost packets constitutes a considerable overhead (>10%) whenever flow throughput decreases below a rate of around 6 packets per second [1]. Thereby to maintain a minimal throughput it would be appropriate to use admission control to limit the number of flows currently sharing any bottleneck link.

One such scheme which offers relatively better scalability is the endpoint admission control. On arrival of a flow the scheme first probes the network by sending probe packets on an end-to-end basis. The probe packets are usually sent at a constant rate (*CBR*) equivalent to the peak rate of the flow to be admitted for a fixed duration. The receiver host then measures either the packet drop rate (or) the inter-packet delay as the algorithm specifies and sends it to the sender host as an acknowledgment packet. Based on this the decision of admitting (or) rejecting the flow is done. The above two schemes of endpoint admission control need to always probe the network before admitting flows thereby introduces an initial wait interval before admission [3]. In this paper, we propose an admission control algorithm which would reduce this wait interval and also reduce the overhead traffic because of no need to probe on every flow arrival. The proposed admission control scheme is based on the Erlang-B loss model which measures the flow blocking probability of a link (considered here as the network resource) as done in telephone network. It uses the number of current active flows, the arrival and departure patterns.

The proposed scheme when compared with other variants admission control such as PCP-DV the delay based admission decision scheme and the conventional packet loss it results in comparatively higher bottleneck link utilization. The overhead traffic generated in the scheme is also very less. A higher flow admission rate is also achieved.

The paper is organized as follows: Section II elaborates the various works carried out so far in the related field. In section III the proposed admission control algorithm based on Erlang-B loss model of the telephone network using the call arrival and departure patterns is described. An analytical performance evaluation of the proposed algorithm is illustrated in section IV. Section V describes the simulation environment. Section VI highlights the results comparing the performance of the proposed admission control algorithm with PCP-DV (*Phantom Circuit Protocol-Delay Variation*) [4] an inter-packet delay based algorithm and the conventional packet drop based algorithm for admission control. Section VII provides a summary of the paper.

II. RELATED WORK

In the literature several such QoS models and admission control strategies have been proposed. Bianchi et al [13], introduces an admission control scheme which determines admission decision through probing rate. Viktoria Elek et al [7], investigate an admission control scheme with FEC. Coskun Centinkaya et al [12], propose an admission control technique involving only the egress routers for QoS provisioning. Más Ivars et al [11], present a PBAC scheme for controlled load service. In Christain Vogt [10], a detailed overview of different QoS schemes is provided. Lee Breslau et al [3], provide a detailed review on the performance of the various MBAC. Ming Li et al [5], describe a model which integrates an admission control algorithm with DiffServ service. In this the admission decision is based on the calculation of mean explicit rate and queue length for each class of service.

One such scheme which offers relatively better scalability is the endpoint admission control technique introduced in Lee Breslau et al [3]. This scheme first probes the network by sending probe packets on an end-to-end basis. The flow admission decision is then based on detection of probing packet losses. If the probe packet drop rate is beyond a threshold, the flow is not admitted. The probe packets are usually sent at a constant rate (*CBR*) equivalent to the peak rate of the flow to be admitted. This scheme does not take into consideration of the end-to-end delay and jitter experienced by the probing packets at the receiver for the admission decision.

Jianming Qiu et al [15], suggest that at times of heavy congestion, packet drop based admission criteria is enough. However, when the network is lightly congested or not congested, packet drop ratio does not reflect the correct internal state of the network.

N.Benameur et al. [1], proposes an integrated admission control for both streaming UDP and elastic TCP flows. It works on the principle of measuring the current available bandwidth using a “Phantom” TCP connection.

In a later work Bianchi et al [4], discuss a similar approach of providing admission control based on inter-packet delay called PCP-DV (*Phantom Circuit Protocol-Delay Variation*). IP telephony voice traffic based on Brady two states (ON/OFF) model is the source traffic used in their simulation. Their algorithm fixes an upper threshold and a lower threshold based on the initial inter-packet delay observed to admit a flow.

Lluis Fabrega et al [2], elaborates an admission control scheme exclusively for TCP flows. In their approach the admission control phase occurs within data transfer phase. The first data packets are used for probing the available throughput in the path and after that admission decision is made.

In 2005 we proposed a teletraffic engineering based admission control scheme [9] operating on Erlang-B model. It uses the current measured load principle to calculate the flow blocking probability.

As we present in the following sections, our proposed admission control algorithm in this paper also operates on the same Erlang-B loss model. The fundamental change here is how the flow blocking probability at a link is calculated. It uses the current flow arrival and departure patterns for it instead of the current measured load principle. The proposed algorithm is found to provide a fair admission rate among both UDP and TCP data flows and better link utilization in comparison to the other models, PCP-DV and also with respect to the conventional packet drop based admission control algorithm.

III. ERLANG-B BASED PROACTIVE ADMISSION CONTROL ALGORITHM

This section provides a detailed description of admission control algorithm based on the measurement of blocking probability using the Erlang-B loss model adapted to Internet traffic.

Initially, for the first 1 sec., before admitting the data flow in to the network on arrival, a probe packet flow is initiated. Subsequently the initiation of the probe flow depends on the previous success of admitting the flow after probing and the next probe time (NxPrTime). When current probing results in a successful flow admission then for the next 1sec. (the time interval used in the algorithm for the measurement of the flow blocking probability) data flows are admitted directly into the network without probing. When the probing result is unsuccessful the arrived data flow is rejected and the next probe time interval is reduced by half. Irrespective of success (or) failure of the probing in admitting a data flow, if the next probe time is reached, at which the previous measurement of blocking probability is considered outdated, probing is initiated on arrival of all data flows till we get an acknowledgement to know the current network blocking status. This is how the next probe time interval is changed, shown in the algorithm (Fig. 1). Probing probe rate is set equal to the peak rate of the flow to be admitted. Probe packets on reaching a designated node of the network it copies the current measured value of the blocking probability of its out going link connecting the next hop node. This on reaching the receiver host is sent back to the sender as an acknowledgement (ACK) packet. This measured blocking probability value is used to make the admission decision at the sender host is as shown in Fig. 1. When the measured blocking probability is less than the threshold P_T fixed at 0.05 the flow is admitted else rejected. This threshold is used as a QoS measure to ensure that a maximum of 5% data flows only gets rejected on arrival. A stricter blocking probability would only mean a larger provisioning of the available bandwidth to satisfy more number of data flows. The blocking probability calculation of a link is described in the first sub-section A of this section as shown in Fig. 2 (Pseudo code).

```
Initialize (done once)
    NxPrTime  (Next probe time)              = 1sec;
    P_flag (Boolean, flow admission )        = 0; (successful)
    $T_{ack}$ (Time of ACK packet arrival)   = 0.0
    $P_T$ (Blocking probability threshold )  = 0.05;
Initialization complete
On arrival of a flow at sender
    If ( CT < 1.0 )  { Initiate Probe flow }
    If ( CT >=  ($T_{ack}$ + NxPrTime) )  {  Initiate probe flow }
        else {
            if (P_flag == 0 ) {Admit flow directly without probing ;
                               set NxPrTime = 1;
                              }
        else  {  Drop flow on arrival ;
                 set NxPrTime = 0.5 sec ;
              }
On arrival of ACK at time $T_{ack}$ at sender from receiver
    If (BP < $P_T$ )  { Admit flow;  }  else  { Reject flow; }
where
    CT = Current Time;
    BP = Measured flow blocking probability
```

Figure 1. Pseudo-Code of the sender for the proposed admission control algorithm.

The probing scheme employed is similar to one used in endpoint admission control scheme. The probe packets are generated only to get the current blocking probability of the link. Further, the probe packets generated on a flow arrival is set to a lower priority than the data flow packets. This ensures that the probe packets do not thrash the network rather only starve to enter the network. When a source does not get an acknowledgement within a fixed time interval, probe life time (PLT) the flow admission gets rejected.

A. *Blocking Probability Measure*

The measurement of the flow blocking probability at the link is done at every 1sec (next_mea_interval). During this interval, the number of active data flows, number of new data flows arrived and number of data flows departure are all measured. A data flow is said to be active flow if it has sent at least one packet in the measurement interval. A flow is said to be a new flow if no packets with that particular flow-id has been forwarded through the link during the previous measurement interval. A counter (N_{ar}) of such flow-id is maintained. A counter (N_{dep}) of departed number of flows is also maintained. A flow is considered as departed flow if has sent at least 1 packet in the previous measurement interval but no packet in the current measurement interval. Using these measured values the flow blocking probability at the particular link is calculated using the Erlang-B formula E(a, m) as shown in the pseudo-code (Fig. 2).

The first parameter 'a' in E(a, m) corresponds to the current load + the future load expected within the next blocking probability measurement interval. It is given in terms of number of flows. The second parameter 'm' corresponds to the number of flow channels possible given the link bandwidth during the measurement interval. It is calculated using the average flow rate computed from the current link utilization (load) and the number of active data flows.

```
Initialization (done once)
    next_mea_interval = 1 sec;
    $N_{cac}$ (Number of current active data flows) = 0;
    $N_{ar}$ (Number of new data flow arrivals) = 0;
    $N_{dep}$ (Number of flow departure) = 0;
    BP (Blocking probability) = 0.0;
Initialization complete
On arrival of a packet at ingress edge of a diffserv
If (CT < next_mea_interval ) { Count $N_{cac}$ , $N_{ar}$ and $N_{dep}$ ;
                               Copy BP to every packet ; }
    else {
        L = Current Link Load;   // measured
        a = $N_{cac}$ + ( $N_{ar}$ - $N_{dep}$ ) ;
        r = L / $N_{cac}$ ;
        m = C / r ;
        BP = E ( a , m );
        next_mea_interval = next_mea_interval + 1.0 ;
    }
where E ( a , m ) = $\frac{a^m}{m!} / \sum_{i \le m} \frac{a^i}{i!}$
      C = Link_bandwidth_capacity
      r = Avg_flow_rate
```

Figure 2. Pseudo-Code for the blocking probability measurement based on Erlang-B model of telecommunication network.

IV. ANALYTICAL EVALUATION OF THE METHOD

For evaluation purpose we consider a single bottleneck link having a bandwidth capacity of C. Suppose a source sends a periodic stream of packets to receiver at a rate R_0 , starting at an arbitrary time instant. If the packet size is L bytes then packets are sent with a period of $T = L / R_0$ time units. The one-way delay D^k (OWD) from source to receiver of packet k will be as given in (1)

$$D^k = \frac{L}{C} + \frac{q^k}{C} \qquad (1)$$

where q^k is the queue size at the bottleneck link upon the arrival of packet k, q^k /C is the queueing delay of packet k.

The arrival rate of both UDP and TCP are modeled as poisson process with intensity λ_u and λ_t respectively. Let the UDP flows have a constant bit rate of b_S with mean duration of $1/\mu_d$ and the mean size of TCP elastic flow be $1/\mu_t$. Thus the link load induced by UDP flow will be $\rho_u = \lambda_u /C^* \mu_d$ and by TCP flows will be $\rho_t = \lambda_t * b_S / C^* \mu_t$.. Let d be the admission decision threshold already fixed. Thus a new flow will be admitted only when total link load will be less than the fixed admission threshold d as shown in (2)

$$C - (\rho_u + \rho_t) \le d \qquad (2)$$

V. SIMULATION SETUP

To understand the behavior and to evaluate the performance of the proposed admission control algorithm, a simple dumbbell network with a single bottleneck link of 10Mbps network, is considered for simulation. All other links connected to this core link are also considered to be of 10Mbps bandwidth. All links are assumed to have the same propagation delay of 20msec. Buffer space of up to 250 packets has been provided in the core link. It has been fixed based on a rule-of-thumb attributed to the router buffer sizes [14]. To determine the effect of this large buffer size, we even carried out our simulation with the bottleneck link having a buffer size of only 10 packets.

A multilink topology when considered for simulation would result in a sequence of N links forming the network path between the source and receiver. The two metrics associated with the throughput will be based on end-to-end capacity C computed as in (3) and the available free bandwidth ‘A’.

$$C = \min_{i=1..N} C_i \qquad (3)$$

The A in the path is determined by the link having the least free bandwidth among the N links connecting the source and receiver. (i.e.) $A = \min_{i=1..N} A_i$. Thus in our simulation only a single bottleneck network is chosen as the topology for the simulation.

It is required that the intermediate network router queuing system must be able to differentiate data and probe packets. For this the network is configured as a DiffServ network. The data traffic packets from both streaming UDP and elastic TCP flows are marked with a particular DSCP to make them belong to one aggregate having a higher priority in comparison to the probe packets belonging to another aggregate with a lower priority [3]. It is done so that probe packets do not disturb ongoing data flows. Probe packets are also set to small size, thereby enabling the probe flow to generate a large number of packets [9]. In the proposed the admission control algorithm which extends the Erlang-B loss model, it is assumed that the probe flows and data flows on arrival which fail to admit a data flow do retry. In the simulation a poisson process having an inter-arrival time of 300ms models the flow arrival. The UDP flows admitted have an exponential lifetime with an average of 25sec. The TCP flows used for simulation are categorized as Mice and Elephant flows [1]. Mice flows represent TCP flows with size uniformly distributed between 10 and 19 packets and Elephant flows with size uniformly distributed between 20 and 400 packets. The entire simulation is run for 500sec. It is been proved in [6, 8] that TCP flows in large majority generated in web sessions can be accurately represented by a poisson process. Within a session, a user would be retrieving a certain number of pages, each page possibly requiring a varying number of TCP flows. The number of flows initiated in a session represents a self similarity in the flow arrival process. Thereby, in this paper we make an assumption similar to the realistic situation where the sessions arrive as a poisson process and the flow arrival process within a session being burstier.

On arrival of a flow, probe packets are generated at a constant rate equivalent to the peak rate of flow to the destination host. The probe packets when generated for an incoming flow have a maximum lifetime of 500ms of CBR characteristics. The probe packet size has been fixed at 50bytes. The probe flow packets would either abort or complete. A probe flow gets aborted before PLT only when the decision variable (estimated average delay) exceeds the threshold limit, thus not admitting the flow in to the network. During the entire PLT if the decision variable value is within the threshold limit, the flow would then start and enter the network according to its characteristics as specified.

Table – I and II indicates the different data flow sources used for simulation. Streaming UDP data flow sources are identical to the one used by Breslau et al in their experiments [3] and elastic TCP data flow sources are identical to the one used by Benameur et al in their experiments [1]. EXP and POO represents the Exponential and Pareto ON/OFF sources respectively.

TABLE – I. UDP data flow sources

Source	*Peak Rate (Kbps)*	*On Time (ms)*	*Off time (ms)*	*Av. Rate (Kbps)*	*Shape (β)*
EXP-1	1024	125	875	128	-
EXP-2	512	500	500	256	-

TABLE – II. TCP data flow sources

Source	*Implementation*	*Packet size(bytes)*	*Window Size*	*Rate (Kbps)*
Mice	TCP Reno	1000	5	100
Elephant	TCP Reno	1000	5	100

VI. SIMULATION RESULTS

The simulations have been carried out using network simulator NS2. The various metrics used to determine the effectiveness of the proposed admission control algorithm in comparison to other algorithms involve i) Average utilization of the bottleneck link by the data flow packets. ii) Fairness in allocating the bottleneck link bandwidth resource among both UDP and TCP aggregate flows iii) The amount of overhead traffic, probe packets generated to achieve the effective bottleneck link utilization. The above three performance metrics measures are related to the characteristics of the bottleneck link. When admission decision is based on the measure of flow blocking probability, the probe packets at the designated link must not experience a blocking probability greater than the threshold P_T fixed at 0.05. When PCP-DV is used for comparison two thresholds, the lower threshold (LT) fixed at (D_s–3ms) and upper threshold (UT) fixed at (D_s+3ms) are fixed. Thus for the flow to get admitted, the probe packets must experience a jitter only within this threshold range. The D_s used is the initial inter-packet delay observed by the probe packets at the sender host.

Simulation results of the proposed Erlang-B based proactive resource provisioning is compared with the other variant of call admission control algorithms such as PCP-DV and packet loss based admission algorithm. The comparative results are tabulated in Table – III and Table – IV. The effect of the bottleneck link router buffer size on the admission control algorithm is also highlighted in the tabulation. For the packet loss based admission control algorithm, the decision of aborting the probe flow (or) to allow it to complete at 0.5 sec the PLT is taken based on the packet drop perceived by the receiver. A probe flow is aborted even when one packet is lost (dropped) during the PLT, otherwise continued and completed indicating successful admission of the data flow.

TABLE – III. Performance Evaluation of the 4 different admission control algorithm based on the average utilization of the bottleneck link by the effective flow packets when the link buffer size is set to 250 packets.

Admission decision algorithm	***Avg. bottleneck link utilization by***			
	UDP traffic (%)	*TCP-Elephant (%)*	*TCP-Mice (%)*	*Probe traffic (%)*
Proposed Pro-Active Erlang algorithm	**63.5**	**15**	**1.2**	**2.5**
PCP-DV	63	14	1.3	3
Packet loss	56	16	1.3	3.25

Table – III indicates the performance of the admission control algorithms when the bottleneck link is provided with a buffer space of 250 packets, while Table – IV shows the performances when the buffer space is fixed at 10 packets. An analytical performance evaluation of the proposed algorithm is illustrated in section IV. For different assumptions, Fig. 3 – 4, Fig. 5 – 6 and Fig. 7 – 8 show the bottleneck link utilization by UDP data flows, TCP Elephant and TCP Mice data flows respectively. In all cases, a fair sharing of bottleneck link bandwidth among UDP and TCP variants flows is achieved by using the proposed algorithm for admission control. It is because the proposed algorithm is found to have a lowest flow blocking rate when compared to the other two algorithms.

The packet losses are difficult to be estimated in a short interval [15], resulting in a longer time to notify the receiver of the probing status. This causes a larger probe packet flow when probing is initiated with packet loss as admission decision variable.

TABLE – IV: The average utilization of the bottleneck link by the effective data flow packets and probe packets when the bottleneck router buffer size is fixed equal to 10 packets.

Admission decision algorithm	***Avg. bottleneck link utilization by***			
	UDP traffic (%)	*TCP-Elephant (%)*	*TCP-Mice (%)*	*Probe traffic (%)*
Proposed Pro-Active Erlang algorithm	**66**	**9.5**	**0.9**	**1.6**
PCP-DV	59	11	1.0	2.4
Packet loss	43	13.5	1.3	3.1

From the performance table it can be seen that the proposed algorithm provides effective bottleneck link utilization by data flows than other admission decision algorithm. To achieve this utilization the percentage of probe packets generated is also comparatively very minimal. Irrespective of the buffer size provisioning at the bottleneck link also the proposed erlang model based admission decision generates minimal probe traffic. In other words, the overhead traffic generated to achieve the necessary effective link utilization is comparatively less. It is seen that buffer size variations hardly impact the probe flows because of their lower priority in nature but impact the data flow packets.

The reason for low link utilization in the packet loss based admission decision is because they admit a considerably large number of probe flow packets which would increase the offered load at a given instance of time.

Fig. 9 – 10 and Fig. 11 – 12 shows the inter-packet delay experienced by the admitted UDP and TCP Elephant flows respectively with different bottleneck link buffer sizes. It is observed that the average delay experienced by UDP data flows when computed is uniformly at around 5.5msec for the proposed admission control algorithm irrespective of the bottleneck link buffer. The same is the case for both TCP Elephant and TCP Mice variety of traffic flows. TCP Elephant flows experience an average inter-packet delay of around 28ms while TCP Mice flows experience an average inter-packet delay of 33ms. The simulation shows that the bottleneck link buffer has no greater influence on controlling the average inter-packet delay. However they do control the peak inter-packet delay. Large buffer size especially result in lower inter-packet delay for TCP variety traffic flows than smaller buffer size. This not the case with UDP data flows.

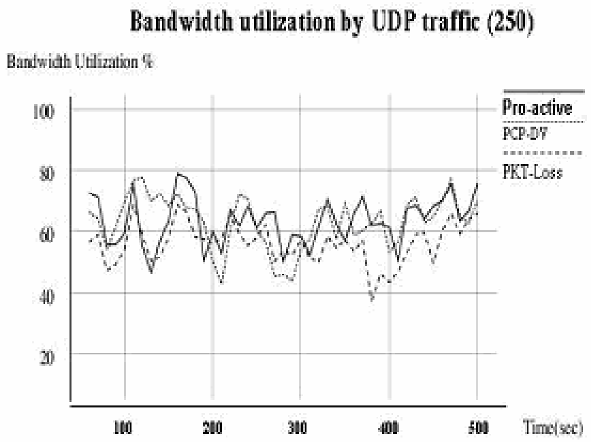

Figure 3: UDP data flow utilizing the bottleneck link by different admission control schemes when buffer space is fixed at 250 packets.

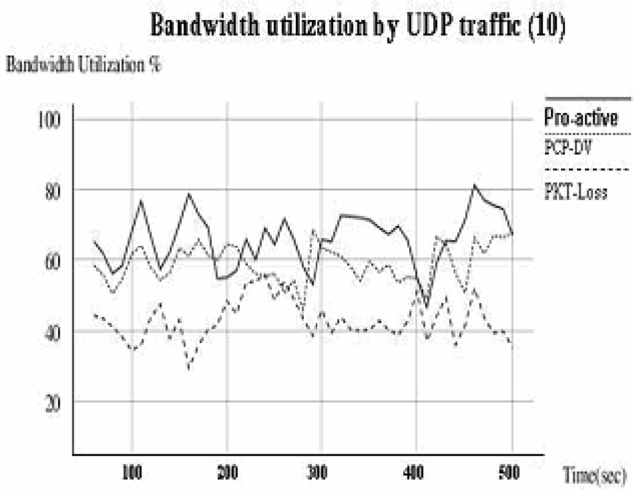

Figure 4: Bottleneck utilization by the admitted UDP data flows when the link is provided with 10 packets of buffer space.

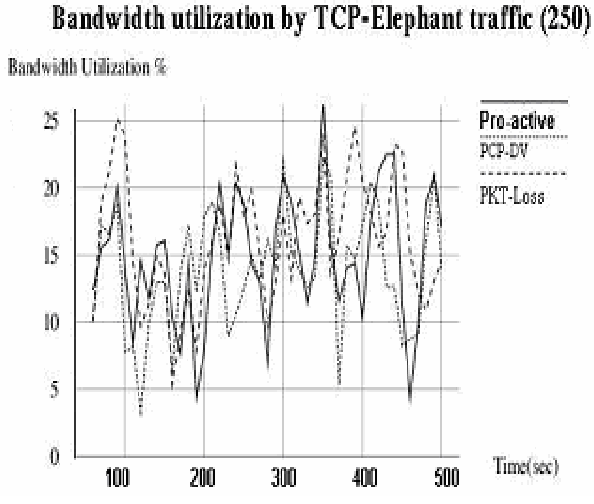

Figure 5: Bottleneck utilization by the admitted TCP Elephant data flows under different admission decision algorithms with bottleneck link provided with 250 packets of buffer space.

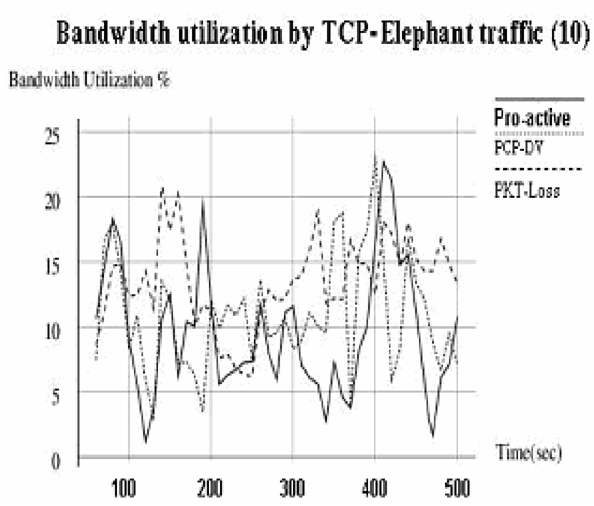

Figure 6: Bottleneck utilization by the admitted TCP Elephant data flows under different admission decision algorithms with bottleneck link provided with 10 packets of buffer space.

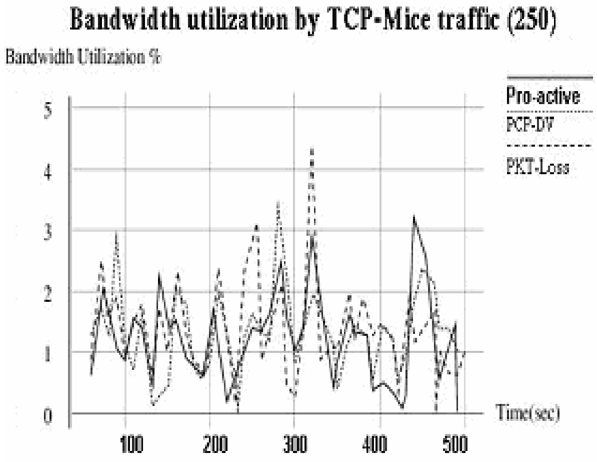

Figure 7: Bottleneck utilization by the admitted TCP Mice data flows under different admission decision algorithms with bottleneck link provided with 250 packets of buffer space.

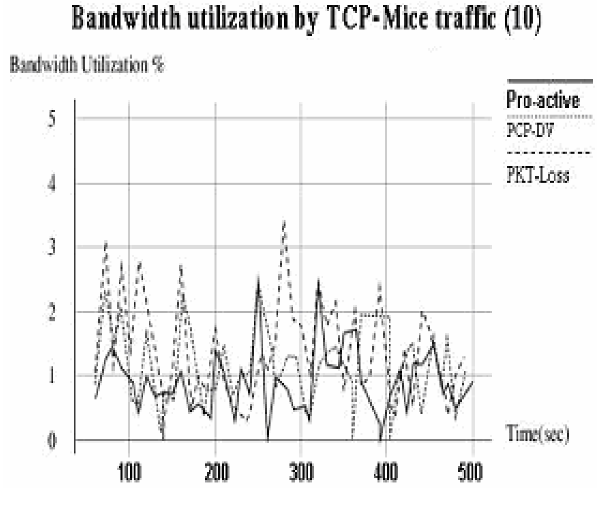

Figure 8: Bottleneck utilization by the admitted TCP Mice data flows under different admission decision algorithms with bottleneck link provided with 10 packets of buffer space.

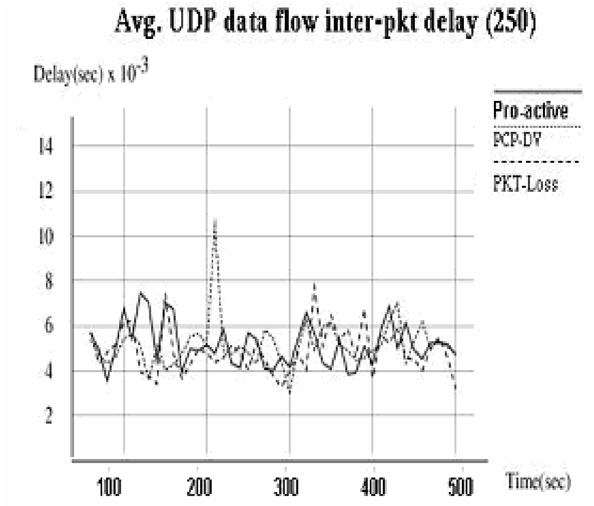

Figure. 9 Average inter-packet delay experienced by UDP data flows under various admission control decisions with bottleneck buffer of 250 packets.

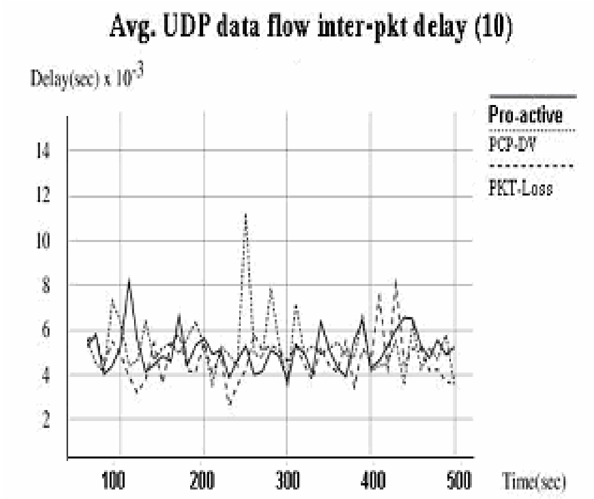

Figure 10 Average inter-packet delay experienced by UDP data flows under various admission control decisions with bottleneck buffer of 10 packets.

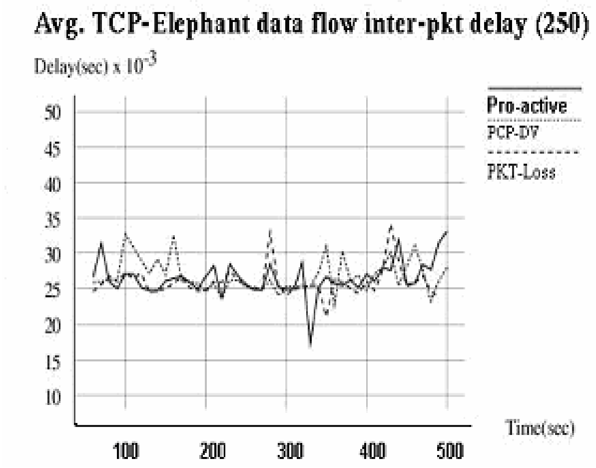

Figure 11 Average inter-packet delay experienced by TCP Elephant data flows under various admission control decisions with bottleneck buffer of 250 packets.

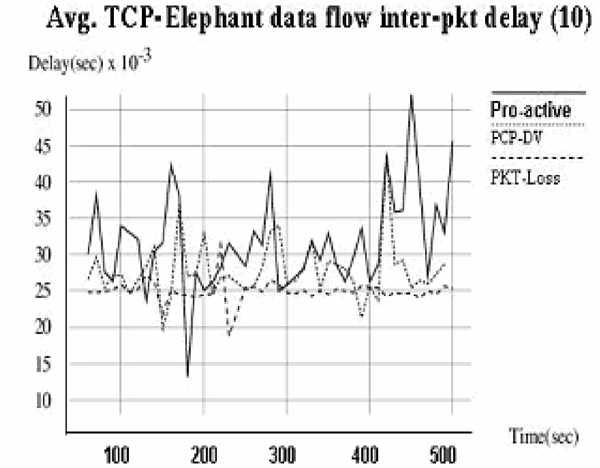

Figure 12 Average inter-packet delay experienced by TCP Elephant data flows under various admission control decisions with bottleneck buffer of 10 packets.

VII. SUMMARY

The proposed admission control algorithm based on proactive resource provisioning uses Erlang-B model of the telecommunication network. It provides a fair share of bottleneck bandwidth between both UDP and TCP data flows. It also results in effective bottleneck link utilization with a reduced overhead traffic of probe flows. The probing mechanism used in the algorithm is of out-of-band. It is because when such probing result in non-admission of the flow the bandwidth consumed for transferring the initial part is considered waste. Such incomplete transfers only constitute a significant source of inefficiency. Thus an admission control scheme for both UDP(streaming) and TCP(elastic) flows have been proposed resulting in a fair share of bottleneck bandwidth among them.

REFERENCES

[1] N.Benameur, S.Fredj, F.Delcoigne, S.Oueslati—Boulahia and J.W.Roberts, “Integrated admission control for streaming and elastic traffic”, *In QofLS, pp 669-81, 2nd COST 263 International workshop, Coimbra, Portugal, Springer, Sep 2001.*

[2] Lluis Fabrega, Teodor Jove, Yezid Donoso Meisel, “Throughput Guarantees for Elastic flows using end-to-end admission control”, *In proceedings of ACM LANC 2003, Oct. 2003*

[3] Lee Breslau et.al., “Endpoint Admission Control: Architectural Issues and Performance”, *In Proceedings of ACM Sigcomm 2000, Stockholm, Sweden*, September 2000.

[4] G.Bainchi et.al., “Endpoint Admission Control with delay variation measurement for QoS in IP networks”, *ACM Sigcomm computer communication, Vol. 32, No.2*, April 2002.

[5] Ming Li et.al., “ Fair Intelligent Admission control over DiffServ network”, *IEEE international conf. on networks 2003*, 2003, pp501-506

[6] S. Ben Fredj, T. Bonald, A. Proutiµere, G. Regnie, and J.W. Roberts. Statistical Band-width Sharing: A Study of Congestion at Flow Level. In ACM SIGCOMM Computer Communication Review, pages 111-122, 2001.

[7] Viktoria Elek, Gunnar Karlsson and Robert Ronngren., “Admission control based on End- to-End measurements*”, IEEE infocomm*, march 2000.

[8] A Feldman, A C Gilbert, W Willinger, “ Data networks as cascades: Investigating the multifractal nature of Internet WAN traffic SIGCOMM 1998.

[9] L.Senthilkumar and V.Sankaranarayanan, “Teletraffic Engineering based flow admission control for a DiffServ Network”, *in Proc of IEEE Indicon-2005, India*, Dec 2005, pp 68-72.

[10] Christian Vogt, “Admission control and Resource Reservation on the Internet”, *ACM sigsoft, Software Engineering Notes, Vol.27, No. 3*, May 2002.

[11] I. Más Ivars and G. Karlsson, `PBAC: Probe-based admission control,' *in Proc. of QofIS 2001, vol. 2156 of LNCS*, (Coimbra, Portugal), pp. 97-109, Springer, September 2001.

[12] Coskun centinkaya and Edward W. Knightly, “Egress Admission Control*”, IEEE trans. on Multimedia, Vol 3, No. 1* , March 2001.

[13] Giuseppe Bianchi, Antonio Capone, and Chiara Petrioli, “Throughput Analysis of End-to-End measurement-based admission control in IP”, *IEEE infocomm, Tel Aviv*, Mar. 2000.

[14] Guido Appenzeller, “Sizing Router Buffers”, In Proceedings of ACM Sigcomm 2004, Portland, Oregon, USA.

[15] Jianming Qiu et.al., “ An end-to-end probing based admission control scheme for multimedia applications”, IEEE international conference on multimedia & Expo., Aug 2001, Tokyo, Japan.

Lakshmanan Senthilkumar, was born in India in 1975. He received his Bachelor’s degree in Computer Science & Engineering from Annamalai University in 1996 and graduate degree in the same branch from Madras University in 2000.

He is currently working as Assistant Professor in B.S.A.Crescent Engineering College, Chennai, India. From 2003 he is doing his Ph.D. degree in Anna University, Chennai. His research area include quality of service provisioning in Internet, High speed Networks and Mobile Computing.

Venkadachalam Sankaranarayanan, was born in India in 1942. He received his B.E. in Electical Engineering from Annamalai University in 1966, M.Sc. (Engg) in High Voltage from college of Engineering, Guindy, Chennai and Ph.D. in computer applications from IIT(Madras) in 1979.

He worked as professor in Anna University for about 3 decades. He was then deputed as Director Tamil Vitual University, from 2004 - 2006 Chennai. He is currently Director at B.S.A.Crescent Engineering College, Chennai, India.