Journal of Theoretical and Applied Information Technology
20th October 2013. Vol. 56 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

MITIGATION OF SUBSEQUENT REQUEST PROBLEM IN PROBE BASED ADMISSION CONTROL FOR MULTICAST

[1]I. SATHIK ALI, [2]P. SHEIK ABDUL KHADER

[1]Assoc. Prof., Dept. of Computer Applications, BSA university, Vandalur, Chennai-48, India
[2]Prof. & Head, Dept. of Computer Applications, BSA university, Vandalur, Chennai-48, India
E-mail: [1]isathikali@yahoo.co.in, [2]psakhader@bsauniv.ac.in

ABSTRACT

Real-time applications such as multimedia streaming and video conferencing have quite stringent Quality of Service (QoS) requirements from the network, because they are more sensitive to available bandwidth and loss rate than non real-time traffic. To provide scalable and simple Quality of Service (QoS) mechanism for multicast services, Probe-Based Multicast Admission Control (PBMAC) scheme was proposed. PBMAC encounters subsequent request problem which degrades system performance significantly when the network traffic is heavily loaded. In this paper, this problem is investigated and Improved Probe-Based Multicast Admission Control (IPBMAC) scheme is then proposed to overcome this problem. The simulation shows that this improved multicast admission control results in reduction of the bandwidth requirement for probe flows and increase of the available bandwidth.

Keywords: *Multicast, Quality Of Service(Qos), Admission Control, Intserv, Diffserv.*

1. INTRODUCTION

During the last years there has been an increasing deployment of multicast applications in the Internet, most of them oriented towards multimedia. Many of these applications demand quite stringent quality of service to provide smooth play-out at the receiver. Such requirements are not possible to meet with the current best-effort Internet. Recently, there have been many research efforts to provide quality of service in distributed manner. These efforts share the common idea of endpoint admission control: a host sends probe packets before starting a new session and decides about the session admission based on the statistics of probe packet loss [1], [2], delay or delay variations [3], [4].

One advantage of the traditional best-effort Internet service model is its simplicity. Another one is its efficiency, since a high degree of sharing is achieved. The disadvantage is that best-effort is a service with no absolute guarantee. Therefore the high variability of the provided QoS might not meet the requirements of some applications. The need for improvement to the basic best-effort infrastructure has resulted in various QoS Models and Services. The main focus of this research work is to improve PBMAC which is simple and scalable QoS mechanism for multicast services.

The accuracy of the proposed scheme depends on the number probe packets used for probing process. Further, this scheme requires a high level of multiplexing on the links to make sure that load variations are less compared to average load. These limitations are as similar as in PBMAC.

1.1 QoS Models

The QoS models for the Internet are open standards defined by the Internet Engineering Task Force (IETF). There are two Internet QoS models standardized by IETF: integrated services and differentiated services. These two Internet QoS models augment the traditional best-effort service model described in RFC1812.

Integrated services: Integrated services **(**IntServ) model is a dynamic resource reservation model for the Internet described in RFC 1633 [5]. IETF defines two services for IP Networks which are collectively known as IntServ: controlled load service and guaranteed rate service [5], [14], [15], [16]. Controlled lad service defines a service that approximates the behavior of best effort service under lightly loaded networks. Guaranteed rate service, which we refer to as IntServ in this paper guarantees end-to-end QoS. In IntServ, hosts use a signaling protocol called Resource ReSerVation Protocol (RSVP) to dynamically request a specific

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

quality of service from the network. An important characteristic of IntServ is that this signaling is done for each traffic flow and reservations are done at each hop along the route. Although this model is well suited for meeting the dynamically changing needs of application, there exist some significant scaling issues which imply that it cannot be deployed in the network in which single router handles many simultaneous flows. The strength of IntServ model is that it provides an absolute service guarantee [5], [6], [12].

Differentiated services: Differentiated services (DiffServ) model removes the per-flow and per-hop scalability issues, replacing them with a simplified mechanism of classifying packets [7]. Rather than a dynamic signaling approach, DiffServ uses bits in the IP Type of Service (TOS) byte to separate packets into classes. DiffServ is the current trend in the Internet community for the development of scalable Internet architecture [8]. A drawback of the DiffServ schemes is that it does not contain admission control [12]. In an effort to combine DiffServ's superior scalability with IntServ's superior QoS, several papers have proposed the quite novel approach of using endpoint admission control [4], [9], [12], [13].

1.2 Admission Control

Some applications, such as video conferencing or streaming audio, require a guaranteed level of Quality-of-Service (QoS) to work properly. These QoS requirements may be in terms of a minimum bandwidth, bounded end-to-end delays, or maximum packet loss rates suffered by a flow. Network routers that support such flows must be able to allocate and maintain their finite network resources to uphold their guarantees. Thus, these routers may also have to reject new traffic flows that would cause the router to violate its promises. The process of deciding to accept or reject a new flow is called admission control. If the sum of the bandwidth usage of the current flows and a new flow is greater than network total bandwidth, the flow is rejected. These QoS guarantees, which have no tolerance for violations, are called 'hard" guarantees, and some flows demand this guaranteed service [6]. Other flows, however, may accept some amount of QoS guarantee violation, usually bounded by some probability values. This is called predictive service, and such statistical or "soft" guarantees provide more flexibility for the admission control algorithm, leading to increased network utilization. There are several call admission control mechanisms that assure end-to-end QoS.

Endpoint admission control: As an alternative to the IntServ algorithm, endpoint admission control has been introduced [9]. The IntServ achieves individual QoS in IP Network on per-flow basis by using a RSVP as means to reserve resources in the network from source to destination. However, it has a scalability problem, since routers need to retain state information and reserve resources along the way. Meanwhile, endpoint admission control algorithm does not depend on the routers for the admission control. Therefore, routers do not need to keep per-flow state or process reservation request and routers can drop or mark packets for some other QoS related-purposes. Previous efforts to provide a soft real-time service, such as Controlled Load specified in the IETF, have met with limited success as they were built on a signaling protocol (e.g., RSVP) and router support for per-flow admission control and scheduling was needed . The scalability and deployability of these mechanisms are hindered by the need for routers to process signaling messages and make admission decisions for each flow, as well as to maintain per-flow state. Endpoint admission control investigates whether such services could be provided with minimal support from network routers. With endpoint admission control, end hosts make their own admission control decisions by probing the network for available bandwidth and admitting or rejecting themselves based on the results of these probes [9]. End point admission control mainly targets unicast end-to-end connections. In this paper, our focus is on probe based admission control for multicast and mitigation of subsequent request problem.

Admission control mechanism for multicast: I Mas extended probe-based multicast admission control (PBMAC) to support multicast applications [10]. PBMAC borrows the idea from probe-based unicast admission control, which received many research efforts recently [11]. In probe-based schemes, hosts probe available network bandwidth before joining a new session and receiving data. The probe traffic may have a lower priority than data traffic, thus the probing process will not affect QoS perceived by existing multicast sessions. Without keeping per-flow states in the routers, the probe-based scheme achieves high scalability and is easy to deploy. In this paper, we will focus on PBMAC proposed by I Mas. Although PBMAC inherits the merits of probe-based unicast admission control on scalability and simplicity, there is a

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

problem related to PBMAC, called subsequent request problem. The problem is that probe traffic of a later request for a multicast session is not aware of the co-existing data traffic for the same multicast session. Thus, over some nearly overloaded links, existing data traffic may prevent the later arrived requests joining the same multicast group. It will obviously degrade the performance of PBMAC in bandwidth utilization and scalability [11]. Le Chunhui proposed EPBMAC scheme in which complementary probing was devised to solve this subsequent request problem. The idea is to utilize the existing data traffic and reduce the probe traffic over congested links. A part of the routers may be required to implement the task and change the priority of data traffic to the priority for probe traffic. But still, complementary probe traffic in the shared link is an additional load in the congested link / router [13].

In this paper, the subsequent request problem found in PBMAC, which degrades system performance significantly when the network traffic is heavily loaded, is investigated. This subsequent request problem prevents new receivers from joining the same multicast group to receive multicast data even though admission of new receivers will not cost any extra resources in the network. This is a serious problem which degrades network performance significantly when the network is nearly overloaded. Here, we aim at overcoming this serious issue. Based on the investigation of this issue, an improved PBMAC is then proposed, in which when a subsequent request arrives, the request is accepted without probing the link further instead of complementary probing devised to solve this problem in EPBMAC. Like other probe-based admission control schemes, the improved PBMAC (IPBMAC) fits well for Controlled-Load Services (CLS) as well as Differentiated Services (DiffServ).

The rest of the paper is organized as follows: Section 2 describes the general probing procedure of PBMAC, the subsequent request problem and improved PBMAC . Section 3 speaks about the implementation and results. Section 4 gives the conclusion.

2. PROCEDURAL DESCRIPTION

Real-time applications such as multimedia streaming and video conferencing demand quite stringent Quality of Service from the network, because they are more sensitive to available bandwidth and loss rate than non real-time traffic. To provide scalable and simple Quality of Service mechanism for multicast services, Probe-Based Multicast Admission Control (PBMAC) scheme was proposed. The subsequent request problem found in PBMAC degrades system performance significantly when the network traffic is heavily loaded. This is a serious problem which prevents new receivers from joining the same multicast group to receive multicast data even though admission of new receivers will not require any extra resources in the network. Here, we aim at overcoming this serious issue.

2.1 The General Probing Procedure of PBMAC

Admission control scheme used here is an Endpoint admission control. Here there is neither reservation of flows in the routers all along the path nor any significant field in the header to indicate the service priorities. Hence the implementation environment should be able to have enough features to make the admission control effective. In order to adapt the admission control for multicast, I Mas proposed to create two multicast groups: one for probe process and one for data session itself. Senders probe the path until the root node of the multicast tree, and start to send data if accepted by this node. The probe from the sender is continuously sent to the root node, and it will be forwarded along the multicast tree whenever receivers have joined the probe group.

Every receiver trying to join needs to know the addresses of both multicast groups. It first needs to join probe group to be admitted into data group. Once a receiver has performed the acceptance decision, it leaves the probe group and joins the data group. The root node of the multicast tree must perform an admission decision for new senders, but the rest of the routers only need to have the priority based queuing system to differentiate probes from data. All that the probing procedure assumes is a shared tree multicast routing protocol with a root node (rendezvous point). Multicast receivers perform an admission decision for each one of the flows from different senders independently and there is no need to perform an admission decision for senders, as the root node is the sender itself.

When a host wishes to set up a new flow, it starts by sending a constant bit rate probe till root node of the multicast tree at the maximum rate that the data session will require. The probing time is chosen by the sender from a range of values defined in the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

service contract. This range forces new flows to probe for a sufficient time to obtain an accurate enough measurement, while prohibiting unnecessary long probes. The probe packet size should be small enough so that we get sufficient number of packets in our probing period to perform the acceptance decision. The acceptance threshold is fixed for the service class and is the same for all sessions.

The root host starts counting the number of received packets and the number of lost packets (by checking the sequence number of the packets it receives). When the probing period finishes, it compares the probe loss ratio measured (P_{lods}) with the threshold loss ratio (P_{target}) and sends a reply packet indicating the decision. If P_{target} is greater than $P_{loss,}$ that flow is accepted and root sends its acceptance decision to the sender. If not, sender is rejected**.** Reply packet is sent at high priority to minimize the risk of loss.

Finally, when the sending host receives the acceptance decision, it starts sending data to the root. The probe from the sender is continuously sent to the root and is forwarded along the multicast tree, whenever the receiver joins the probe group. It means that the root joins the data group after the admission decision is positive and also it joins the probe group to receive the probe continuously. Now the root node becomes the root for the probe group and data group.

The receiver first joins the probe group to receive the probes from the root. Now the root node sends probe to the receiver at the maximum rate that the data session will require for certain period (probing time). The acceptance threshold is fixed for the service class. The receiver receives the probe packets from the root for certain period of time and measures the probe packet loss and makes the decision for admission. Longer the admission period gives a higher accuracy of the probe packet loss. Once the receiver has compared the Packet loss with Target loss, it makes the decision. If the decision is positive, the receiver immediately joins the data group, while in the case of a negative decision, it needs to back off for a period of time before trying to join again.

2.2 Subsequent Request Problem

If the multicast data traffic is being delivered over link L due to the successful admission of request A when probing process for request B starts, we call request B a subsequent request over link L, and L the shared link of request A and B. It is clear that the admission of a subsequent request over the link will not cost any extra resources on L. However, when the traffic on the bottleneck link is close to its admissible level, the blocking probability of the subsequent requests may be extremely high. We call this problem subsequent request problem. The cause of subsequent request problem is the co-existence of the probe traffic and the data traffic on the bottleneck link. In PBMAC, when a subsequent joining request arrives, probe traffic is sent to the receiver through the bottleneck link where data traffic exists, which requires much more extra bandwidth. If the available bandwidth is not sufficient for the probe traffic, probes will experience a high loss, which results in high request blocking probability. As subsequent requests problem restricts the number of receivers, the scalability of PBMAC and bandwidth utilization of networks are significantly debased.[12, 13]. Based on the analysis of the subsequent request problem, Le Chunhui proposed an enhanced probe-based multicast admission control (**EPBMAC)** scheme to solve this subsequent request problem which is explained below.

Essence of EPBMAC: EPBMAC inherits the basic idea of the conventional PBMAC. A multicast source creates a multicast data group and a probe group, and traffic of probe group is marked to a lower priority than that of data group traffic. Traffic at the peak data rate with a lower priority is used in PBMAC to probe the new multicast branch. However, in EPBMAC, complementary probe traffic is used on the shared links, and remarking operation is executed on the node at the graft point of the multicast tree for the new receiver. In EPBMAC, the traffic used to probe the newly grafted multicast branch is composed of two parts:basic probe traffic F_{pe} and additional probe traffic F_{pd}.. F_{pe} is generated by the multicast source and sent to the probe group. It is complementary to the data traffic, i.e., the source sends the probe traffic at rate $R_{pe}(t)$ at time t:

$$R_{pe}(t)=R_{pk}-R_d(t) \quad (1)$$

where R_{pk} is the peak rate of the data group and $R_d(t)$ is the data rate at time t. F_{pd} is actually the traffic of data group, but it is remarked to the same priority as the probe at the graft point of the new branch. Hence, F_{pe} is also complementary to F_{pd}. By using complementary probe mechanism and remarking operation, EPBMAC achieves following targets:

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

(1) Peak rate probing: As F_{pe} is complementary to F_{pd} , the sum of two parts of probe traffic has the constant rate R_{pk} .
(2) Admitted session protection: Since the data traffic is remarked to the same priority as F_{pe} before it is conveyed on the un-probed branch, the early admitted sessions will not be impacted by the probing process.
(3) Subsequent request problem avoidance: On shared links, F_{pe} is complementary to the data traffic, and the rate of the total traffic will not exceed R_{pk} , *hence* subsequent request problem could be well solved.

But still complementary probe traffic in the shared link is an additional load in the congested link / router [17]. Our improved algorithm given in the next sub section aims at removing complementary probe traffic from the bottleneck link or shared link so that the bandwidth occupied by the complementary probe traffic can be used by other traffics.

2.3 Improved PBMAC

Based on the analysis of the PBMAC and EPBMAC, an improved PBMAC is proposed in this paper. In the improved scheme, there are two multicast groups: one for probe process and one for data session itself as in PBMAC and EPBMAC. We make an assumption that from representing node to multicast receivers enough bandwidth is available. In this scheme, root-based or center-based multicast approach is used. In IPBMAC, apart from multicast root, some more nodes are identified as representing nodes for receivers. For a set of receivers, a representing node is a node which is closer to receivers on the multicast path from the multicast source to multicast receivers and also the node is an enhanced router or source. We also make an assumption that from representing node to multicast receivers enough bandwidth is available.

The sender's procedure to send data flow is very much similar to the sender procedure of PBMAC. When a sender wishes to set up a new flow, it sends a constant bit rate probe till root node of the multicast tree at the maximum rate that the data session will require. The probing time is selected by the sender from a range of values defined in the service contract. The probe packet size should be small enough so that sufficient number of packets is generated in our probing period to perform the acceptance decision. The acceptance threshold is fixed for the service class and is the same for all sessions.

The root host starts counting the number of received packets and the number of lost packets. When the probing period finishes, it compares the probe loss ratio measured (P_{lods}) with the threshold loss ratio (P_{target}) and sends a reply packet indicating the decision. If P_{target} is greater than $P_{loss,}$ that flow is accepted and root sends its acceptance decision to the sender. If not, sender is rejected**.** Reply packet is sent at high priority to minimize the risk of loss. When the sending host receives the acceptance decision, it starts sending data to the root. It means that the root joins the data group after the admission decision is positive and the root node becomes the root for the data group.

The probe generating procedure for multicast receiver is shown in Figure 1. When a request from a receiver to join the probe group for receiving a data flow comes in, if the required data (not probe) is delivered through *shared link L* in the multicast path from the root to the receiver, the last router in *shared link L* generates probe packets at the peak rate of data flow, else the root node keeps generating the probe packets. Here, *Shared link L* is a path which connects the nodes (except *the representing node)* through which the required data flow is being forwarded and could be shared among different nodes.

When a request to join probe group for receiving multicast data comes in :

If the required data is delivered through *shared link L*
The last router in *L* generates probe packets at the peak rate of data;
Else
The root node generates the probe packets;
Endif

Figure 1: Probe Generating Procedure For Receiver

A node which wants to join the data group to receive multicast data follows the procedure shown in Figure 2. Whenever a multicast receiver wants to receive multicast data from the multicast source, first it checks whether the representing node is receiving the required data. If multicast data is being delivered to other nodes through the representing node of the receiver, the receiver will not join the probe group but it will immediately join the data group. This procedure is shown in *Case 1* of Figure 2.

Journal of Theoretical and Applied Information Technology
20th October 2013. Vol. 56 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

//*Case 1*: (*Representing node* already delivering the required data to other nodes)

If *representing node* delivers the required data to other nodes
No probing is needed;
The receiver joins the data group immediately;
Endif

//*Case 2*: (*Representing node* not delivering the needed data to other nodes)

If *representing node* does not deliver the required data to other nodes
The receiver sends join-request for joining in the probe group;
The receiver joins the probe group;
Probe packet loss at the receiver is computed;
If probe packet loss is less than packet loss threshold
The receiver joins data group;
Else
The receiver will back off;
Endif
Endif

Figure 2: Receiver's Procedure

If representing node of a receiver has not been forwarding the required data to other node through it, the receiver sends join-request for joining in the probe group. The join–request message is forwarded using unicast routing toward the root node until it either arrives at the root or any other router that already belongs to the multicast tree of the data group and also generates or forwards the probe packets. Now, the receiver joins the probe group either at the root of the multicast tree or at

the last router, on the path from the root to receiver node, which generates or forwards probe packets in the multicast tree. Then, probe packet loss at the receiver is computed. If probe packet loss is less than packet loss threshold, the receiver joins data group otherwise the receiver will back off. This procedure is shown in *Case 2* of Figure 2.

3. IMPLEMENTATION

The topology used for the simulation is shown in Figure 3. and it consists of sender nodes (5 , 6), receiver nodes (7,8,9,10,11, 12), representing nodes (2,3,4), root node (0), bottleneck link (between node 0 and node 1)with capacity of 2Mb and other links as shown in Figure 3. We also make an assumption that from representing node to multicast receivers enough bandwidth is available. The network is made multicast enabled. P_{target} is fixed as 0.1 and probing period is fixed as 1 second. Simulation period is 101 seconds and link delay is 0.01 millisecond or 10 microseconds for each link. The link delay / propagation delay can be ignored in the computation since it is very small.

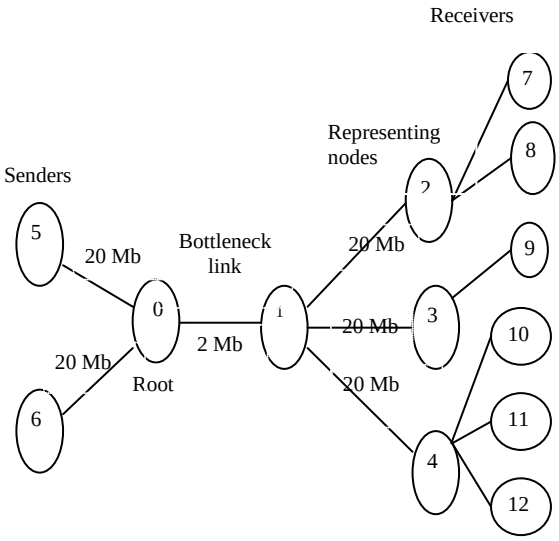

Figure 3: Simulation Topology

The UDP protocol is used in the simulation predominantly. We use three different on-off traffic sources. Two of them are having exponential on-off times. The third on-off traffic source has Pareto on and off times (described by a shape parameter, β). Table 1 contains the parameter values for these on-off sources. The Exponential and Pareto on-off sources are denoted with the labels EXP and POO, respectively.

Table 1: Source Parameters

Source	Peak Rate (Kbps)	On Time (ms)	Off Time (ms)	Average Rate (Kbps)
EXP-1	512	500	500	256
EXP-2	256	500	500	128
POO-1 (β=1.2)	256	500	500	128

Journal of Theoretical and Applied Information Technology
20th October 2013. Vol. 56 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The experiment is carried out for PBMAC, EPBMAC and IPBMAC. In this simulation, three senders (EXP-1, EXP-2, POO-1) probe the network with peak rate of the data flow till the root node 0, from 0 sec to 1 sec. EXP-1 and EXP-2 are attached to node 5 and POO-1 is attached to node 6. The senders start transmitting data from 1 sec, till the root after the successful probing. In our experiment on PBMAC and EPBMAC, initially multicast receivers who want to receive multicast data sent by the three senders, join the probe group at 1sec. The probe ends at 2 sec. Based on the successful probe decision, data flows are sent to the receivers. Receivers can leave the data group randomly, but send the join-request to join the probe group only at 1, 2,3,4,5,6,…seconds (i.e., at equal interval of 1 sec.). Average bandwidth occupied in the bottleneck link by the probe flows and data flows during the simulation are tabulated in Table 2. In our simulation on EPBMAC, 3rd second onwards, probe traffic is reduced to half in the bottleneck link due to complementary probing.

Whereas on IPBMAC, due to the modified procedure for receivers, if at least any single receiver receives multicast data through the bottleneck link, the probe traffic for that multicast data will not be sent through bottleneck link. This leads to reduction of probe traffic in the bottleneck link as shown in Table 2.

In Table 3, comparison of our simulation for PBMAC, EPBMAC and IPBMAC is given. The comparison shows that the available bandwidth percentage in the bottleneck link is increased to 74.75% in our improved PBMAC as against 25.25% in PBMAC and 50.0 % in EPBMAC. The simulation also shows that the average bandwidth utilization percentage by the probes in the bottleneck link is decreased to 0.5% in our improved PBMAC as against 50 % in PBMAC and 25.25 % in EPBMAC.

Table 2: Average Bandwidth Utilization In Bottleneck Link.

Time (Sec)	PBMAC		EPBMAC		IPBMAC	
	Probe (Kbps)	Data (Kbps)	Probe (Kbps)	Data (Kbps)	Probe (Kbps)	Data (Kbps)
0	Probes are sent till root					
1	Data flows are sent till root after successful probing					
2	1024	-	1024	-	1024	-
3	1024	512	512	512	-	512
4	1024	512	512	512	-	512
5	1024	512	512	512	-	512
6	1024	512	512	512	-	512
7	1024	512	512	512	-	512
8	1024	512	512	512	-	512
9	1024	512	512	512	-	512
10	1024	512	512	512	-	512
11	1024	512	512	512	-	512
.	-					
.	-					
.	-					
101	1024	512	512	512	-	512

Whereas on IPBMAC, due to the modified procedure for receivers, if at least any single receiver receives multicast data through the bottleneck link, the probe traffic for that multicast data will not be sent through bottleneck link. This leads to reduction of probe traffic in the bottleneck link as shown in Table 2.

Table 3: Comparison Of PBMAC, EPBMAC AND IPBMAC

Algorithm	In the Bottleneck link		
	Available Bandwidth %	Average Bandwidth utilization % by Probe Flows	Average Bandwidth utilization % by Data Flows
PBMAC	25.25 %	50.0 %	24.75%
EPBMAC	50.0 %	25.25 %	24.75%
IPBMAC	**74.75 %**	**0.5 %**	**24.75%**

In Table 3, comparison of our simulation for PBMAC, EPBMAC and IPBMAC is given. The comparison shows that the available bandwidth percentage in the bottleneck link is increased to 74.75% in our improved PBMAC as against 25.25% in PBMAC and 50.0 % in EPBMAC. The simulation also shows that the average bandwidth utilization percentage by the probes in the bottleneck link is decreased to 0.5% in our improved PBMAC as against 50 % in PBMAC and 25.25 % in EPBMAC.

Figure 4 illustrates the comparison of available bandwidth percentage in the bottleneck link of our simulation for PBMAC, EPBMAC and IPBMAC and Figure 5 reveals the average bandwidth utilization percentage by the probes in the bottleneck link during the simulation for PBMAC, EPBMAC and IPBMAC. From the Figure 4, it is learnt that the available bandwidth percentage in

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the bottleneck link of our simulation for IPBMAC is increased when compared to PBMAC and EPBMAC. Figure 5 demonstrates that the average bandwidth utilization percentage by the probes in the bottleneck link during the simulation for IPBMAC is decreased largely when compared to PBMAC and EPBMAC.

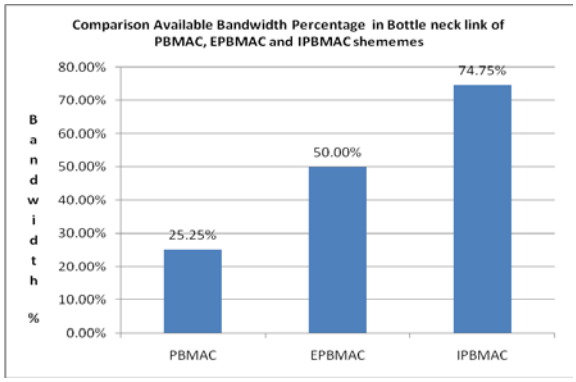

Figure 4: Comparison Of Available Bandwidth Percentage In Bottleneck Link Of PBMAC, EPBMAC And IPBMAC

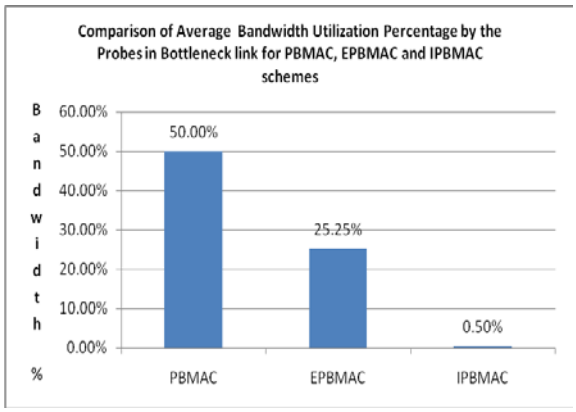

Figure 5: Comparison Of Average Bandwidth Utilization By Probes In Bottleneck Link Of PBMAC, EPBMAC And IPBMAC

4. CONCLUSION

The Endpoint admission control scheme does not need the routers to keep the state of various flows. Further the decision of whether or not to admit the flow is based on the calculated loss percentage of the probe packets. By having uniform threshold for all the admitted flows the Quality of Service is ensured equally for all the flows. The improved scheme called IPBMAC proposed in this paper handles the subsequent request problem found in probe based multicast effectively. Consequently, this leads to reduction of the bandwidth requirement for probe flows and increase of the available bandwidth in the bottleneck link. The simulation further shows that IPBMAC leads to stable link utilization and also this improved scheme enables the admitted flows to have a limited loss. The packet loss ratio in the probe stream provides a reliable and efficient solution for QoS provisioning for loss sensitive applications, without extensive support in the routers.

As future work, the admission threshold can be decided based on parameters such as delay and jitter along with packet loss there by making the admission decision more realistic and robust. And also this work may be extended for admission control for mobile nodes.

REFERENCES

[1] V. Elek, G. Karlsson, and R. Ronngren, "Admission control based on end-to-end measurements," in *Proc. of IEEE INFOCOM 2000* , Tel Aviv, Isralel, March 2000, pp. 623-630.

[2] G. Karlsson, "Providing quality for internet video services," in *Proc. of CNIT/IEEEE ITWoDC 98*, Ischia, Italy, September 1998, pp. 133-146,

[3] G. Bianchi, A. Capone, and C. Petrioli, "Packet management techniques for measurement based end-to-end admission control in IP networks," Journal of *Communications* and *Networks*, June 2000, vol. *2*, pp. 147-156.

[4] *G.* Bianchi, A. Capone, and C. Petrioli, "Throughput analysis of end-to-end measurement-based admission control in IP Networks", in Proc. *of IEEE INFOCOM 2000* , March 2000, pp. 1461-1470.

[5] R. Braden, D.Clarke and S.Shenker," Integrated Services in Internet Architecture: an Overview ", IETF RFC1633, June 1994.

[6] S. Shenker, C. Patrigde and R.Guerin, "Specification of Guaranteed Quality of Service", IETF RFC 2212, September 1997.

[7] S. Blake et al., "An Architecture for Differentiated Services", IETF RFC 2475, December 1998.

[8] Giuseppe et. al., "Quality of service aware multicasting over diffserv and overlay networks", IEEE Network, January/February 2003.

[9] Lee Breslau et al., "EndPoint Admission Control: Architectural Issues and Performance", SIGCOMM'00, ACM, August 2000.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[10] Ignacio Mas et al., "Probe Based Admission Control for Multicast"Proc of the 10th IEEE/IFIP IWQoS, May 2002.
[11] Le Chunhui et al.,"A study on Probe based Admission Control and Enhancement", Journal of Electronics(China), January 2006
[12] Ignacio Mas , Gunnar Karlsson, "Probe-based admission control for a differentiated-services internet", Computer Networks, April 2007
[13] L. Senthilkumar and V. Sankaranarayanan, "Provisioning Erlang-B Model Based Flow Admission Control for Packet Networks", Journal of Information Science and Engineering, 2008
[14] J. Wroclawski, "Specification of the controlled-load network element service", RFC 2211, September 1997
[15] P.P. White, "RSVP and Integrated Services in the Internet: A Tutorial", IEEE Communications Mag.,vol. 35,May 1997
[16] Jinoo Joung et. al., "Flow-based QoS management architectures for the next generation network", ETRI journal, vol.30, April 2008.
[17] I. Sathik Ali et. al., "Study of probe based admission control and further enhancement", IEEE-ICCCET , March 2011.