Comparison of Endpoint Admission Control Algorithms by probing at Exponential Interval and at Constant Interval

L. Senthilkumar†, Dr. V. Sankaranarayanan††

†Dept. of CSE, Crescent Engineering College, Vandalur, Chennai – 600 048, South India
††Former Director, Tamil Virtual University, Elnet Software city, Taramani, Chennai – 600 113, South India

Summary
An Endpoint admission control adopts probing at a constant rate equivalent to peak rate of the flow to be admitted. This inserts a large burst of packets. The real-time traffic flows by nature do not require such large chunk of bandwidth because of their VBR characteristics. This has prompted to use exponential interval probing to determine the spare bandwidth. With such probing, admission decision based on both packet loss and variants of inter-packet delay estimation are evaluated. It is found that probing at exponential interval and admission decision taken based on the proposed admission algorithm which works on estimation of average inter-packet delay result in a better performance in terms of improved bottleneck link utilization and also a lower inter-packet delay variation being experienced by the admitted data flows.

Keywords:
Admission, Exponential interval, Constant interval, QoS, DiffServ Delay

1. Introduction
In Internet, the two service architectures adopted to provide QoS are Diffserv and IntServ. IntServ architecture is highly suitable to provide strict QoS on a per-flow basis. RSVP is an implementation protocol [1]. It offers strict QoS by reserving resources like bandwidth, buffer at each of the routers forming part of the entire end-to-end path. If sufficient resources are not available at any of such router, then the flow may not get admitted. This mechanism of providing QoS lack scalability. It is because it makes the routers to remember the states of different flows forwarded by it.

Diffserv architecture provides QoS on an aggregate level. A set of flows which form an aggregate is treated based on a policy defined on a per hop behavior (PHB). The different schemes which form the policy include metering, policing, and shaping. This mode of providing QoS solves the scalability issue of IntServ. But it the following problems: i) It does not facilitate the Provision of strict QoS on a per-flow basis. and ii) It offers QoS only by confining the flow characteristics according to their service contract, and not based on the internal state of the network. To overcome the above two shortcomings, a Diffserv network with flow admission control is envisaged in this paper.

In the literature several admission control strategies have been proposed [3][4][6][7][9][10]. One such scheme of interest here because of its simplicity and scalability in implementation is endpoint admission control. In this scheme a flow is admitted into the network only when a sufficiently large bandwidth is available on the entire end-to-end path. To identify the available bandwidth, the network is initially probed. By convention probe packets are always sent at a constant rate equivalent to the peak rate of the flow which needs to get admitted. By default probe packets get into the network only when there is bandwidth available because of its lower priority in comparison to data flow packets. The flow admission decision is then based on the drop rate encountered by the probe packets. If the drop rate is less than a prefixed threshold, the data flow is admitted else rejected.

When the bottleneck link utilization exceeds certain threshold packet loss based admission decision has been found not reflecting the correct internal state of the network. For which we propose to use the following two changes in an endpoint admission control scheme which deviate from the conventional one. i) Sending probe packets at exponential interval rather than sending them at a constant rate (CBR) because the real-time flows do not require a constant bandwidth [2]. ii) Probe packet drop rate at the receiver is not used as an admission decision measure instead the estimation of the average inter-packet delay is used.

This paper organized as follows: In section 2 the proposed admission control algorithm based on the estimation of the average inter-packet delay is described. In section 3 the simulation setup is elaborated. Section 4 highlights the results comparing the performance of the admission control algorithms involving probing at exponential interval and at constant interval along with admission decision based on inert-packet delay as well as on packet drop. In section 5 related works has been reviewed. Section 6 provides the summary with a brief account on the future scope.
2. Admission Control Algorithm

This section provides a detailed description of admission control algorithm based on the estimation of the average inter-packet delay.

A data flow, on arrival, is not admitted directly into the network rather it initiates a probe packet stream. The probe packets thus entering the network would get generated either at constant interval or at exponential interval. It is based on the probing scheme as described in this paper. The approach of generating probe packets in an exponential (consal) can be found in the work of Jianming Qiu [2]. The probe packets thus generated will encounter an inter packet delay at the sender side. Let \( D_s \) be such an inter-packet delay between the first two packets of the probe flow. \( D_s \) is then be used to determine the flow admission threshold limit. The inter-packet delay between probe packets arriving at the receiver is measured. Instead of using the last measured inter-packet delay (jitter) to make admission decision, it makes sense to use an exponential averaging of a series of measured inter packet delays to eliminate random fluctuations. If \( D_s(k) \) represents the jitter measured between \( k^{th} \) and \( (k-1)^{th} \) packet and ‘\( \alpha \)’ is a tuning parameter in the range \([0,1]\) then the estimated average jitter is given by \( D_{jitter}(k) \). This estimated jitter adds a fraction of the new measured jitter \( D_s(k) \) to itself, retaining a fraction \((1-\alpha)\) of past history \( D_{jitter}(k-1) \). Based on this, the admission decision algorithm is as shown in fig.1.

\[
\text{Do } \{ \\
D_{jitter}(k) = (1-\alpha) * D_{jitter}(k-1) + \alpha * D_s(k); \\
\text{if } (D_{jitter}(k) <= D_T) \{ \text{continue} \}; \\
\text{else} \{ \text{Abort probe, (Data flow not admitted)} \} \\
\} \text{ until (PLT)}
\]

Where
- \( D_s \) = Inter-packet delay using the \( k^{th} \) packet
- \( D_{jitter}(k) \) = Estimated average delay
- \( D_T \) = Admission delay threshold = \( D_s + 3\text{msec} \)
- \( D_s \) = Inter-packet delay at the sender end.
- \( \alpha \) = 0.25 (constant)
- PLT = 500msec, Probe Lifetime

![Fig.1 Average inter-packet delay estimation based admission control algorithm](image)

In the admission control algorithm, the receiver fixes the admission decision threshold \( D_T \) at 3msec more of \( D_s \). It is being fixed to ensure a conservative statistical delay bound of 3msec on each hop for every probe packet stream [8]. For real time traffic sources especially voice the end-to-end delay should be less than 150msec. Normally the end-to-end delay comprises of the following fixed components, an encoding delay of 20msec, packetization delay of 30msec, traffic smoothing delay of 40msec at ingress, switching delay of 40msec. The remaining 20msec form the variable queuing delay. If there are approximately 5 hops between the source and receivers, this gives an average queuing delay budget of around 4msec per hop. During the entire probe duration if the estimated delay (based on EWMA - Exponential Weighted Moving Average) is within the threshold, the data flow gets admitted. (i.e.) On successful completion of PLT (probe Life Time) during which the estimated delay experienced by the probe packets has not exceeded the threshold the data flow gets admitted otherwise, at any point of time during PLT if the estimated delay experienced by the probe packets exceeds the threshold the probe flow gets aborted thereby preventing the data flow from entering into the network. This helps to make quick admission decision at the times of threshold exceeding the fixed limit earlier itself. It has been proved that longer probing duration only leads to thrashing [3]. Also larger the threshold, lower will be the quality of service offered to the accepted data flow.

2.1 Validity of delay as admission decision parameter

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. It is because the packet drop rate cannot be estimated correctly over a short interval of time [2]. Also when the bottleneck link utilization is large, the average queuing delay may increase appropriately. For the above reasons it is proposed to make use of the delay as the admission decision criteria, thereby, making it an appropriate measure for more accurately reflecting the internal state of the network.

3. Simulation Setup

To understand the behavior and to evaluate the performance of the above mentioned two probing schemes with estimated average delay as the flow admission decision variable, a simple dumbbell network with a bottleneck link of 10Mbps configured as core link of the DiffServ network, is considered for simulation. All other links connected to the core link through which source traffic enter are also 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 is required that the intermediate network router queuing system must be able to differentiate data and probe packets. For this data traffic packets 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 simulation a Poisson process having an inter-arrival time of 300msec models the data flow arrival from 2 of the sources. The total offered load on the bottleneck link is 1.5 times of its capacity. The flows admitted have an exponential lifetime with an average of 25sec. The entire simulation is run for 500sec.

When the probing scheme employed is of constant interval then on arrival of a data flow, probe packets are generated at a constant rate equivalent to the peak rate of flow to the destination host otherwise for the exponential interval probing scheme probe packets are generated with the following characteristics: rate set equivalent to the peak rate of the data flow, burst time and idle time both set to 50msec. This sets a probing rate equal to half the rate used in the former constant interval probing scheme. The probe packets thus generated for each of the incoming flow have a maximum lifetime of 500msec. The probe packet size has been fixed at 50bytes. The probe packets flow would then either abort or complete. When the flow admission decision variable (estimated average delay) exceeds the threshold limit the probe flow gets aborted before the end of PLT and thereby not admitting the flow in to the network. While 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-1 indicates the different data flow sources used for simulation. They are identical to the one used by Breslau et al in their experiments [3]. The packet size from these sources has been fixed at 125bytes. EXP and POO represents the Exponential and Pareto ON/OFF sources respectively.

<table>
<thead>
<tr>
<th>Source</th>
<th>Burst Rate (Kbps)</th>
<th>On Time (msec)</th>
<th>Off Time (msec)</th>
<th>Shape</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXP-1</td>
<td>256</td>
<td>500</td>
<td>500</td>
<td>-</td>
</tr>
<tr>
<td>EXP-2</td>
<td>1024</td>
<td>125</td>
<td>875</td>
<td>-</td>
</tr>
<tr>
<td>POO-3</td>
<td>256</td>
<td>500</td>
<td>500</td>
<td>1.2</td>
</tr>
<tr>
<td>EXP-3</td>
<td>512</td>
<td>500</td>
<td>500</td>
<td>-</td>
</tr>
</tbody>
</table>

4. Performance Evaluation

The network simulator NS2 is used for our simulation. The various metrics used to determine the effectiveness of the admission control algorithm with different probing techniques involve i) Average admission ratio of the network (ie determining the number of probe flows which were successful in admitting the data flow into the network. ii) Average utilization of the bottleneck link. The measures of the above two performance metrics are related to the characteristics of the bottleneck link.

Results of the simulation are tabulated in Table-2. The performance evaluation of the proposed inter-packet delay based admission control algorithm against PCP-DV (Phantom Circuit Protocol-Delay Variation) [5] a variant of inter-packet delay based algorithm and packet loss for both type of probing scheme is shown.

When PCP-DV is used for comparison two thresholds, the lower threshold (LT) fixed at $(D_s-3msec)$ and upper threshold (UT) fixed at $(D_s+3msec)$ are fixed. Thus for the flow to get admitted, the probe packets must experience an inter-packet delay variation only within this threshold range. For a 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. The probe flow is aborted even when one packet is lost (dropped) in the network during the PLT otherwise probing is continued and completed at PLT indicating successful probe completion, thereby admitting the data flow.

Table – 2: Performance Evaluation of the different admission control algorithm based on average bottleneck link (LU) utilization and the peak inter-packet delay experienced by the data flows at the receiver

<table>
<thead>
<tr>
<th>Admission decision algorithm based on</th>
<th>With CBR probing</th>
<th>With EXP probing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LU (%)</td>
<td>Peak Inter-packet delay (msec)</td>
</tr>
<tr>
<td>Packet loss</td>
<td>72</td>
<td>13</td>
</tr>
<tr>
<td>Proposed Inter-Packet delay</td>
<td>89</td>
<td>8</td>
</tr>
<tr>
<td>PCP-DV</td>
<td>87</td>
<td>8</td>
</tr>
</tbody>
</table>

From the tabulation result it can be seen that admission decision based on probing in an exponential interval result in a better performance in terms of both bottleneck link utilization and peak inter-packet delay. The percentage of increase in the average link utilization...
for packet loss based admission decision with exponential interval probing is 12% higher in comparison to constant interval probing. This increase is very high when comparing the same while using the proposed inter-packet delay algorithm and also PCP-DV, which is of about 5%. Also the peak inter-packet delay experienced by the admitted data flows under the proposed delay based admission decision is comparatively less to that experienced when using packet loss as the admission decision. These small increases in average bottleneck link utilization and also the peak inter-packet delay being experienced by the data flows prove the robustness of choosing delay as admission parameter and also acts as a proof for its selection as an admission decision parameter [2]. It is also found that the proposed inter-packet delay based algorithm perform well in comparison to a similar delay based algorithm namely PC-DV. Fig. 2 – 5 shows the bandwidth utilization of the bottleneck link by data flow along with probe flow where it is significantly large while using the above two probing scheme with both packet loss and the proposed inter packet delay based admission decision algorithm.

Fig. 2 Shows the utilization of the bottleneck link in percentage when probing in an exponential interval and using packet drop as the flow admission decision parameter.

Fig. 3 Shows the utilization of the bottleneck link in percentage when probing in an exponential interval and inter packet delay as the flow admission decision parameter.

Fig. 4 Shows the utilization of the bottleneck link in percentage by both data and probe flow when probing at a constant interval and using packet drop as the flow admission decision parameter.
5. Related Work

In the literature several such QoS models and admission control strategies have been proposed. Bianchi et al [4] introduces an admission control scheme which determines admission decision through probing rate. Coskun Centinkaya et al [7] proposes an admission control technique involving only the egress routers for QoS provisioning. Más Ivars I. et al [9] present a PBAC scheme for controlled load service. In Christain Vogt [10] detailed overview of different QoS schemes is provided. Lee Breslau et al [3] provide a detailed review on the performance of the various MBAC. Key et. al [11] describes an end user probing strategy which relies on ECN feedback to determining the admission decision rather than the packet loss. Ming Li et al [12] describes 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.

6. Conclusion

For probing based admission control strategy it is found that probing the network for free bandwidth can be done by injecting probe packets in an exponential interval (EXP type) rather than at constant interval (CBR type). It is found that probing at exponential interval results in a better performance. Also in either of the probing scheme it is found that flow admission decision based on inter packet delay always results in a higher performance in terms of flow admission and fairly even inter-packet delay for the admitted data flows.

6.1 Future scope

1. Determining the relation between the probe life time and the flow duration.
2. Validation of the admission algorithm in a multi-hop scenario, having very frequent route changes.
3. Validating the same admission algorithm in a multicast scenario.

Reference