Adaptive Threshold-Based Admission Control

Helena Sandström, Ulf Bodin, Olov Schelén
Department of Computer Science and Electrical Engineering
Luleå University of Technology, SE-971 87 Luleå, Sweden

***Abstract*— In this paper we present an algorithm for performing dynamic per-link admission control. It is designed to be suitable for deployment using existing quality of service and router techniques. The rationale for the algorithm is to find a bit-rate limit for reserved capacity, which enables efficient capacity utilization and statistical multiplexing gain. By using simple traffic meters configured with respect to defined service levels, a network resource controller estimates an admission limit. The estimated limit reflects the ratio between reserved capacity and the aggregate behavior of current flows. Simulations indicate that reliable admission limits can be estimated already at low reservation levels.**

***Index Terms*— QoS, measurement based admission control, bandwidth brokers, resource managers**

I. Introduction

Providing quality-of-service guarantees requires admission control. Strictly guaranteed forwarding services are typically implementing worst case parameter-based admission control, while more relaxed forwarding services often use measurements of current load to guide their admission decisions. An advantage with parameter-based admission control mechanisms is that they can support reservations in advance [1], while measurement-based admission control algorithms (MBACs) do not rely on worst-case bounds, and can thereby achieve a higher level of bandwidth utilization through statistical multiplexing. Specifications for relaxed forwarding services therefore often provide a delay target, rather than a bound.

Admission decisions are based on some type of traffic characterization. In the simple case, applications are assumed to request reservations corresponding to their *peak rates*. This gives predictable forwarding demands for homogeneous flows, but the available forwarding capacity may be poorly utilized for applications having variable sending rates.

Earlier MBAC algorithms based their estimations on statistics of individual flows. More recent proposals are based on the aggregate behavior of flows. Besides being easier to implement, as a consequence of the law of large numbers, the statistics of the aggregate is easier to estimate than the statistics on a per-flow basis [2].

Several MBAC algorithms have been proposed [3][4][5][6][7][8][9][10], ranging from simple ad hoc algorithms with no mathematical underpinning, to complicated detailed analysis based on solid mathematical foundations. A systematic empirical examination of the relative performance of a selection of these algorithms surprisingly showed that they, given the same level of dropping rate, all achieved very similar levels of bandwidth utilization. None of them was capable of accurately meeting the defined loss target [11].

Most architectures for MBAC are designed around the assumption that admission control is performed in packet forwarding devices (e.g., routers) where also traffic measurements are carried out. The success of router-centric QoS architectures depends on the signalling protocols that must be supported by routers to carry admission requests. These signalling protocols have historically had weaknesses in scalability, authentication, authorization and accounting [12].

There are currently two approaches to solve this problem. One approach is to develop next generation signalling protocols for path coupled signalling. The other approach is to establish QoS control functionality in servers and rely on class-based forwarding in network elements. These servers provide a network service interface to applications. The functionality may include authentication, authorization, accounting, and admission control over layer 2 and layer 3 networks. Such servers are known as bandwidth brokers, QoS managers, network resource controllers, etc. In this paper we use the term Network Resource Controller (NRC).

The objective of the paper is to present an MBAC algorithm that is suitable for deployment in network resource controllers. The rationale for the MBAC is to use simple mechanisms available in current routers to measure the actual bit-rate in one traffic class. Measurements are collected from routers by the NRC as a slow feedback process. Given that the aggregate reserved rate is known, measurements can be used to estimate the ratio between aggregate reserved rate and actual traffic load. We show through simulations that this ratio can be accurately estimated already at low levels of reservations/traffic. The ratio is used to maintain a limit for aggregate admitted rates in the admission control algorithm.

The reminder of this paper is organized as follows. In section II we describe the threshold-based admission control algorithm. Section III describes the simulation strategy and present our results. Conclusions are presented in section IV

0-7803-8938-7/05/$20.00 (C) 2005 IEEE

followed by future work in section V.

II. The admission limit adaption mechanism

Taking the peak rate as the only traffic descriptor, without MBAC, reservations can only be made with guarantees until the sum of all reservation peak rates reaches the level of the *forwarding capacity, C*.

The objective of our threshold-based admission control was to find an aggregate bit-rate limit, based on measurements of current load. This can act as a limit for further reservations, considering both immediate reservations and reservations in advance. The admission limit is preferable above the forwarding capacity *C*, but if several flows are heavily exceeding their peak rate, it is theoretically possible to reach a value below the forwarding capacity.

The basic threshold equation (1) includes four parameters.

$$\eta \cdot \Gamma = \mu \cdot \Lambda \tag{1}$$

One *upper monitoring threshold*, μ, and one *lower monitoring threshold*, η, which in reality are represented by two token buckets, used as traffic meters monitoring at different levels. The remaining two parameters are *reserved capacity,* Λ, which corresponds to the sum of *current* reservation peak rates, and the *admission limit*, Γ

The reserved capacity does not always mirror the actual need for capacity (i.e., for flows having variable sending rates, the actual need for capacity can be considerably lower than their peak rate), but the ratio between current load and reserved capacity can be used to estimate a proper bit-rate admission limit, which acting as a future limit for reserved capacity, enables higher network utilization.

In (1), the right-hand side includes the *reserved capacity*, Λ, and the *upper monitoring threshold*, μ. The upper token bucket filter monitors a rate equal to (or somewhat lower than) the forwarding capacity, *C*. The right-hand side thereby represents more stationary parameters, and can be seen as a constant during a period when no flows are added or removed. The left-hand side represents two adjustable values, the *admission limit,* Γ and the *lower monitoring threshold*, η.

Initially the lower token bucket filter is monitoring a rate equal to reserved capacity, η = Λ, and the admission limit is set equal to the forwarding capacity, Γ = *C*. It is then straightforward to see that if the value of the lower monitoring threshold is decreased, the admission limit must increase to balance the equation.

The upper monitoring level in (1) can be used to decide a utilization target for the network. Setting the level somewhat lower than the forwarding capacity will add an additional margin for the guaranteed service, but will have a negative impact on forwarding capacity utilization.

A. The adaption algorithm

For a service created using the admission limit adaptation mechanism, a *target loss-rate*, τ, and a *delay target*, δ, must be defined. The depth of the token bucket filters are measured in bytes (can also be bits). The delay target is considered when setting the depth of the token bucket filters. The depth for the upper and lower filter is then set to μ * δ, and η * δ respectively.

Constant K: The size of the increase step of the limit value are controlled by K. A large value represents a big step and vice versa. With small steps, the time to reach the final limit value increases, although the value reached may be more reliable. With large steps the treshold value reached may be less reliable. A large K also affects the accuracy of the tuning, and an even larger K may cause that (3) never comes to work.

Both token bucket filters are read periodically, and the NRC updates the loss-rate information by calculating *bytes of non conforming packets/bytes of transmitted packets*. After each reading the results are evaluated, This can lead to three different actions depending on the results and the current state: *Increase, Decrease, and Preserve*.

1) Increase: Initially the admission limit is increased linearly according to (2).

$$\Gamma_{new} = \Gamma_{old} + (K \cdot C) \tag{2}$$

After each increase action a new level for the lower monitoring threshold is calculated using (1). After detecting η*-loss* greater than a reference value equal to τ^2 ($\tau < 1$), which corresponds to the breakpoint where (1) and (3) are equal, the increase action switches to (3).

$$\Gamma_{new} = \Gamma_{old} + \left(K \cdot C \cdot \left(\frac{\ln(\eta\text{-}loss)}{\ln(\tau)} - 1\right)\right) \tag{3}$$

Equation (3) is based on (2). In the logarithmic expression, the relative distance between the loss-rate at the lower monitoring threshold and target loss-rate is considered, thereby the increase steps are successively decreased and the algorithm tunes into a proper limit value. The initial increase phase is finished when η*-loss* reaches a level greater than target loss-rate, τ. This will trigger a back off, and the admission limit will be set to the value it had before the last increase step. After leaving the increase phase, it can not be activated again, hence the admission limit will not be increased any further. This lock remains until the number of flows is changed.

The increase process is illustrated in Fig.1. As can be seen, the level of the lower monitoring threshold is initially decreased more rapidly and is successively slowing down, while the admission limit is increased linearly, until it reaches the break point where it switches to (3).

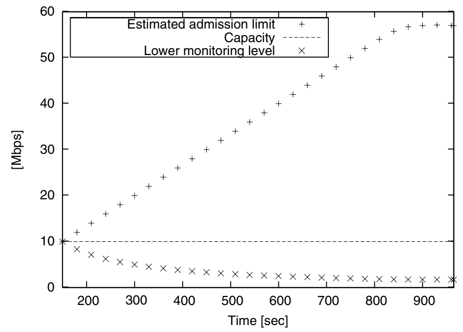

Fig. 1. Increase phase with 57 work trace sources, reading interval 30 secs.

2) Preserve and Decrease: After the increase phase follows a preserve phase. It includes monitoring the upper and lower token bucket filters. The decrease action is activated if η-*loss* exceeds τ and also if μ-*loss* exceeds τ. The new admission limit is calculated according to (4), thereby taking into account the relative distance between measured loss-rate and target loss-rate τ.

$$\Gamma_{new} = \Gamma_{old} * \left(\frac{\ln(loss)}{\ln(\tau)} \right) \tag{4}$$

Provided upper monitoring threshold, μ is set equal to forwarding capacity, when μ-*loss* exceeds τ, capacity is exceeded. The aim is to avoid this situation, but if the forwarding capacity is exceeded, some flows have to be pre-emptied. The reserved capacity, the admission limit, and the lower monitoring level is then recalculated according to the leaving flows corresponding peak rates. The increase phase can now be activated again since the number of flows has changed. The scenario is described by the state diagram in Fig. 2.

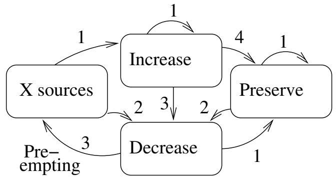

Fig. 2. State diagram for the admission limit estimation process. 1: η-*loss* and μ-*loss* $< \tau$ 2: η-*loss* or μ-*loss* $> \tau$ 3: μ-*loss* $> \tau$ 4: η-*loss* $> \tau$

III. Evaluation

The adaption mechanism was evaluated through simulations in the ns-2 simulator [13]. Since the admission control decisions for the algorithm are made on a per-link basis, a simple topology, in which the adaption mechanism is employed on a single link, was considered sufficient to evaluate the performance aspects of the algorithm.

In order to test our implementation and to gain a deeper understanding of the behavior of the algorithm, the initial simulations was carried out using CBR-sources.

A. Simulation Setup

For all simulations, the link bandwidth was 10 Mbps and sufficient buffer capacity was allocated. For all results illustrated in this paper, except in Fig. 4, max acceptable loss-rate was set to one percent, and the delay target to one ms. Traffic arriving to this link was monitored using two token bucket filters configured according to the adaption mechanism described in section II-A. Data from the startup period was discarded.

1) Source Model: Besides using CBR-sources, traffic is generated from two packet traces taken from two NetMeeting [14] video conference sessions. One sitting conference, and one more bursty including a person frequently moving. Throughout the rest of the paper referred to as *work trace* and *move trace* respectively. When using CBR sources, the sending interval was set to random, and a peak rate was calculated using the shortest time period between two successive packets, with one source sending. In order to avoid synchronization and correlation, sources were started with random time intervals. When using trace files, each source was started from a random place within the trace file.

Data from the three source models are listed in the table. We performed different experiments using one source at a time or mixed. Through the simulations we studied the adaption of the admission value when different numbers of sources were sending, adding one source for each new simulation until the forwarding capacity limit was reached. When studying the increase phase, each simulation was run until the increase phase was finished. The estimated value was then tested by studying loss-rates, while loading the link with the calculated number of flows that would hold according to the estimated admission limit.

Type:	Peak rate: (Mbps)	Avge rate: (Mbps)	Avge interval: (secs)
CBR	0.81917	0.2048	0.005
Work Trace	0.17279	0.0241	0.0421
Move Trace	0.26717	0.10342	0.0400

2) Constant K: In order to find a proper value on constant K where issues discussed in section II-A are balanced, we performed a serie of simulations, with values of K ranging from 0.05 to 0.5. Simulations were run with different source types and different token bucket reading intervals. We found that with K in the range from 0.05 up to around 0.20,

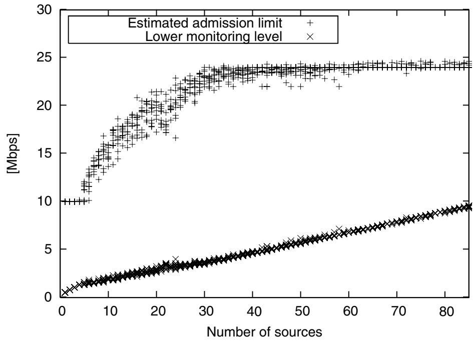

Fig. 3. Estimated admission limit - move trace sources.

simulation results was conforming, and reaching the same admission limit. Based on these results, constant K in (2) and (3) was set to 0.2.

3) Reading Intervals and Sample Periods: We performed a serie of tests were the increase phase was run on different time scales. Our experiments showed that short increase intervals introduces a small risk of overshooting the proper admission limit. This is especially true for bursty sources, since with very short time intervals, there is a risk of missing a bursty period during the increase phase. Our results indicated that the time interval between successive increase actions is recommended to be no shorter than 5 seconds, but could be considerably longer.

The time interval is however selective, and a general time scale can not be defined solely on our simulation results. In reality, token bucket reading intervals will probably be determined with respect to router processing overhead, and by on which time scale loss-rate guarantees are provided. If token bucket reading intervals is shorter than the recommended increase interval, it can easily be adjusted by limiting the increase action to, for example, every third or fourth reading.

B. Simulation Results

Fig. 3 illustrates the process of admission limit estimation for move trace sources. Each simulation was repeated ten times. The admission value approaches its final limit when the link load is around 40%. For work trace sources, the admission value was approaching its final limit when the link was loaded with 50 sources, which corresponds to a link load around 25%. As a result of their different traffic characteristics, the final admission limit was lower for move traces than for work traces.

The estimated admission limit was tested through simulations. The limited number of sources was calculated by *admission limit/sum of peak rate*. For move traces the estimated admission limit was between 24.0 and 24.6 Mbps. An admission limit of 24 Mbps results in 89.83 move sources. Simulations were run with 89, 90 and 91 sending move trace flows. Each simulation was repeated three times measuring the average loss-rate. The results indicated that 90 sources matches the target loss-rate, which should be compared with 89.83 calculated using the estimated admission limit. 91 sources impose a risk of overshooting the target loss-rate. The level of link capacity utilization was around 94, 96, and 98 % respectively. This test was repeated using the other sources models with similar results.

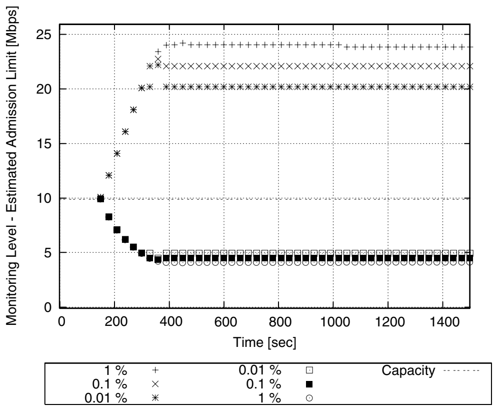

Fig. 4. Initial increase phase - 37 move trace sources. Varying target loss-rate.

We also performed a serie of simulations with different values on the target loss-rate τ. Fig. 4 illustrates the estimated admission limit, and the corresponding lower monitoring level with τ set to 1%, 0.1% and 0.01%. Interesting to investigate in this scenario was how well the estimated admission limit could meet the different performance levels. Results showed that they all came close to the defined target loss-rate, indicating that the estimated admission limit is close to maximum. For the 1% curve a decrease was triggered at time 1050 seconds. The measured loss-rate at that point was 1.04 %.

The estimated admission limit is depending on the characteristics of the current mix of sources. During very low load, corresponding to only a few sources, a traffic mix containing mainly sources with a large span between peak rate and average rate introduces a risk of overshooting the maximum admission value. However, during low load, the lack of multiplexing gain results in a lower estimated admission limit, which thereby reduces this risk. Even though it seems desirable to reach the final admission limit during low load, the conservative pattern during very low load are to be considered positive.

Fig. 5 illustrates the relationship between work and move traces with different distribution. As can be seen, bursty

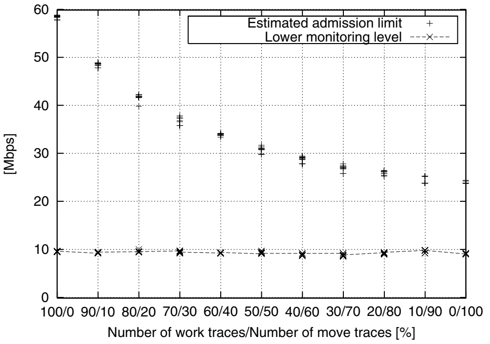

Fig. 5. Reached admission limit with different distribution of work and move traces. Link load close to 100%.

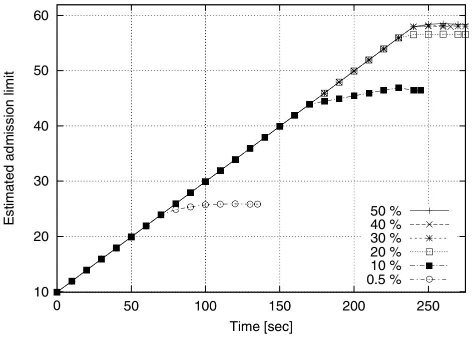

Fig. 6. Increase phase - work trace at different load levels. Reading interval 10 s.

move traces influences the admission value in a conservative way. This support the argument that a somewhat conservative approach during low load is preferable. The conservative influence also indicate that if the admission value initially is to optimistic, when adding new flows, it will quite rapidly adopt to the characteristics of the current traffic mix.

Fig. 6 illustrates the increase phase for work traces, during different load in a time perspective. This is used to gain further insight in how the algorithm perform. Noticeable is that even when the load is as low as 0.5% the estimated value is near half the maximum admission limit.

IV. Conclusion

In this paper, we present a simple adaptive threshold-based admission control algorithm. Based on simple measurements it adapts a dynamic aggregate admission limit for future and immediate reservations.

The algorithm is emphasizing easy deployment in network resource controllers such as bandwidth brokers or QoS managers. By utilizing simple token bucket measurements we are able to avoid heavy router processing overhead. This design aspect also reduces the cost of deployment, and meets a common goal of scalability.

Our simulation results indicate that a trusted admission limit can be estimated already at low reservation levels, prior to busy hours when reservation demands are high. The results also indicate that the algorithm manage to achieve a high level of capacity utilization, and in investigated conditions manages to meet defined performance levels.

V. Future Work

Future research includes systematic analysis of our algorithm, using additional source models and trace files together with more dynamic traffic scenarios. We will also study its relative performance with different parameter settings, including the level of utilization of available forwarding resources, and its ability to meet defined performance levels.

References

[1] Olov Schelén, Andreas Nilsson, Joakim Norrgård, and Stephen Pink. Performance of qos agents for provisioning network resources, June 1999.

[2] Matthias Grossglauser and David N.C.Tse. A time-scale decomposition approach to measurement-based admission control. *IEEE/ACM Transactions on Networking*, 11(4), August 2003.

[3] Claudio Casetti, James F. Kurose, and Donald F. Towsley. A new algorithm for measurement-based admission control in integrated services packet networks. In *Protocols for High-Speed Networks*, pages 13–28, 1996.

[4] S. Crosby. Statistical properties of a near-optimal measurement-based cac algorithm. In *Proceedings of IEEE ATM '97 Workshop*, Lisboa, Portugal, 1997.

[5] R. Gibbens and F. Kelly. Measurement-based connection admission control, 1997.

[6] Matthias Grossglauser and David N. C. Tse. A framework for robust measurement-based admission control. *IEEE/ACM Transactions on Networking*, 7(3):293–309, 1999.

[7] Matthias Grossglauser and David N. C. Tse. A time-scale decomposition approach to measurement-based admission control. In *INFOCOM (3)*, pages 1539–1547, 1999.

[8] Sugih Jamin, Peter B. Danzig, Scott J. Shenker, and Lixia Zhang. A measurement-based admission control algorithm for integrated service packet networks. *IEEE/ACM Transactions on Networking*, 5(1):56–70, 1997.

[9] Sugih Jamin, Scott Shenker, and Peter B. Danzig. Comparison of measurement-based call admission control algorithms for controlled-load service. In *INFOCOM (3)*, pages 973–980, 1997.

[10] Jingyu Qiu and Edward D. Knightly. Measurement-based admission control with aggregate traffic envelopes. *EEE/ACM Transactions on Networking*, 9(2):199–210, 2001.

[11] Lee Breslau, Sugih Jamin, and Scott Shenker. Comments on the performance of measurement-based admission control algorithms. In *INFOCOM (3)*, pages 1233–1242, 2000.

[12] J. Manner, X. Fu, and P. Pan. Analysis of existing quality of service signaling protocols. draft-ietf-nsis-signalling-analysis, IETF Internet draft, Expires: November, 2004.

[13] The network simulator - ns-2. URL:http://www.isi.edu/nsnam/ns/.

[14] Microsoft windows technologies, windows netmeeting. URL: http://www.microsoft.com/windows/NetMeeting.