A Modular Architecture for Measurement Based Admission Control

Thomas Bohnert, Edmundo Monteiro

University of Coimbra
Laboratory of Communications and Telematics
CISUC/DEI, Po II, Pinhal de Marrocos, 3030-290 Coimbra Portugal
Phone: +351239790000 Fax: +351239701266
WWW home page: `http://cisuc.dei.uc.pt`
{`tbohnert,edmundo`}`@dei.uc.pt`

Abstract. In the Laboratory for Communications and Telematics (LCT) of the Centre of Informatics and Systems of the University of Coimbra (CISUC), research is ongoing towards a Measurement Based Admission Control (MBAC) framework for DiffServ [1] based networks. Till now, many research in this area was and still is undertaken with an encyclopedic literature about this subject as one achievement. While some articles present complete MBAC algorithms, others focus on specific details. In fact, we show that the MBAC problem is a compound of different subproblems and in this paper we present a modular architecture for an MBAC algorithm. In our proposal, we dismantle the MBAC problem in *four* different modules, which in turn leads to seamless integration in DiffServ edge router design. Further, and in contrast to current approaches, we argue that the logical separation between the operation of an MBAC algorithm and a queueing discipline is disadvantageous and propose a cooperative framework where the parameters of the queueing discipline are adjusted according to resource consumption of ongoing and arriving resource consumers, both determined by the MBAC algorithm.

Keywords: Measurement Based Admission Control, QoS

1 Introduction and Background

Independent from any Quality of Service (QoS) enabled network architecture and generally valid for any network in which, customers compete for finite and shared resources, Admission Control (AC) is a fundamental premise for service guarantee provisioning. In short, an admission controller is an entity, which decides, as the term implies, about admittance of service requesting customers to a network. Thus, AC can be considered as a pure decider, whose decision is based on arguments and an admission policy. Generally, a new resource consumer (packet, flow, or session) is admitted if, and only if, QoS requirements of all active, previously admitted resource consumers as well as the demands of the requesting consumer can be guaranteed, either strictly or statistically.

As disclosed by the seminal work of Leland et al., statistical features of IP traffic are fundamentally different form that of the POTS[1] [2]. As a consequence, mathematical models from traditional teletraffic theory are *invalid* for IP networks [3]. One result of that disclosure was the evolution of MBAC. Short, the rationale of MBAC is estimation of network state based on measured quantities like for instance current loss rate, as substitution for theoretical source models, the foundations of traditional, parameter based AC algorithms. This complies to a shift from static models to dynamic load estimation algorithms.

Instantaneously, it can be stated that an MBAC algorithm consists of at least two *logical* modules, measurement and estimation. However, a complete MBAC algorithm can be further decomposed.

Our MBAC framework is a compound of four modules, namely a Selector, Sampler, Estimator and Policy module, see Fig. 1. As we will explain more detailed in following sections, and as depicted in Fig. 2, this allows seamless integration in DiffServ edge router [1] design. In fact, our framework is designed to be deployed on Tier-II level DiffServ edge routers of an Interne t typical ISP topology [4], exemplarily illustrated in Fig. 3.

Furthermore, based on this design, we propose a cooperation between the MBAC algorithm and the queueing discipline, where the key idea is to use flow arrival information as well as current and future resource demand, estimated by the MBAC to adjust the queueing discipline dynamically.

In the residual paper, we discuss each module in detail from Sec. 2 to Sec. 5 and in Sec. 6, we introduce our proposed cooperative, adjustable resource management. Finally, Sec. 7 we close with a conclusion and pointers to further directions.

[1] Plain Old Telephone System

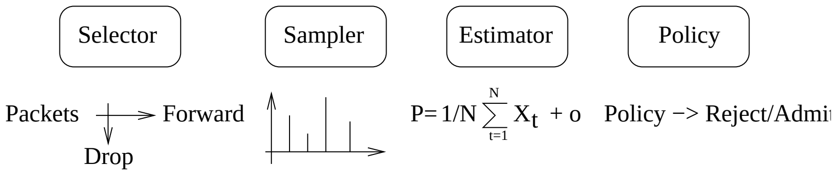

Fig. 1. Four Module MBAC Architecture. The Selector module for flow management, a Sampler module for packet sampling and measuring, an Estimator module for estimation and prediction of QoS parameters and a Decider module for admission policy enforcement

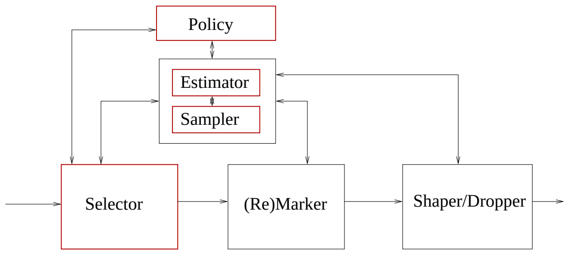

Fig. 2. DiffServ edge router with MBAC enhancement.

Fig. 3. Representative two layer (Tier-I and II) ISP topology of the German Research Network (DFN - G-WIN). Large points belong to the Tier-I (core) layer, small points are Tier-II layer access routers.

2 Selector Module

A fundamental premise of our concept is DiffServ compatibility and in compliance to, the Selector module can be considered as an enhanced Classifier [1]. Indeed, it substitutes the latter in our framework as depicted in Fig. 2. Contrary to being a plain Classifier, the Selector module additionally incorporates complete flow management functionalities.

The Selector detects new flows by computing a hash value (ID) over predefined IP header content and a successive list query. If the ID is not in any list, i.e. it belongs to a new flow, the Selector triggers an admission process for. Therefore, the Selector maintains a Black- and Whitelist with the hash values for either rejected or admitted flows. Besides of flow detection, this technique additionally allows the enforcement of admission policies and protects the controller from denial-of-service vulnerableness, by malfunctioning hosts or intentionally insisting admission applicants. Admitted flows are added to the Whitelist, while a rejected flow is temporarily added to the Blacklist. After a specific timeout it is permitted for reapplication. During this time, arriving packets are dropped and a tear down notification is sent. Keeping a Whitelist also allows us to detect inactive flows by deleting entries after a predefined timeout.

As we apply implicit AC, that means we refrain from explicit signalling, a source may not be aware of its rejection and continues sending packets. To prevent this waste of resources, and to avoid disappointed users, the network is not working for no obvious reason, the Selector module sents a tear down notification to rejected flows by triggering an existing Internet Control Protocol Message (ICMP). This is in contrast to [5] where only non TCP sources are notified and the latter are stopped by intercepting initial SYN packets. We favour our approach cause of extensive TCP backoff periods.

3 Packet Sampling Module

In many frameworks, sampling, measurement and estimation is treated as one monotolitic block, as in the DiffServ architecture where these concepts are integrated in the Meter. However, a closer look discloses, that these areas clearly pose different problems and for each single one, research has been undertaken independently. Again, our goal is an optimal architecture based on DiffServ, and therefore we split the meter in a Sampler and an Estimator, with the Sampler integrating packet sampling and measurement.

The motivation for Sampling is to select a representative subset of packets that allow accurate estimates of properties of the unsampled traffic to be formed. Sampling is targeted at the selection of a representative *subset* of packets. The subset is used to infer knowledge about the whole set of observed packets without processing them all [6].

In the scope of our initial work, however, all MBAC algorithms under investigation left sampling techniques unconsidered. In fact, the choice of a sampling technique is determined by the QoS metric, e.g. local packet delay. Considered MBAC algorithms for our framework, however, calculate solely with *arrival rates*, i.e. bits per second, thus, each packet needs to be sampled in a given period. This implies k-of-k sampling within fixed time frames. However, MBAC algorithms can also operate on alternative metrics and thus, different sampling techniques could be advantageous or even strictly required, see the references in [6] for some examples. Henceforth, four our framework we locate sampling techniques in an extra module for fast and easy integration of different sampling methods, e.g. n-of-k, random etc.

The Sampler module also hosts a group of measurement procedures. Contrary to sampling methods, a comprehensive set of measurement techniques have been introduced in the past. On a first level, one have to differentiate between active probing and local measurements. Active probing techniques are known in the scope of Probe Based Admission Control (PBAC). The measurement rationale is sending probe packets to destinations, and measure for example jitter. From this measurements, available (path) bandwidth is estimated [7] [8].

For various reasons, bandwidth stealing, timing etc., we favour local measurements. In general, the mathematical roots of the estimation algorithm determine the measurement procedure applied to. For example, the MBAC approach in [9] requires average arrival rates over a fixed time scale. In [10] maximum values of instant arrival rates are recorded with the so-called Time-Window measurement, also over a fixed time scale. A more advanced measurement procedure is used in [11] where maximum average rates, called maximum evelope rates, are measured over different time scales.

In a multi-service network like the Internet, statistical features of traffic classes are heterogeneous. To deal with, integrating different measurement procedures, allows us to evaluate, identify and apply most accurate estimation techniques for traffic aggregates. This is opposite to current approaches where one MBAC technique is applied to control all kinds of traffic.

4 Estimation and Prediction Module

The challenge for MBAC algorithms is accurate estimation of statistical QoS parameters, e.g. buffer overflow probability based on measurements, as these parameters are finally used as decision arguments.

Intuitively, one might assume that, for instance a mean value computed using a Moving Average filter, provides a close to optimal estimation. This, however, is a *misleading* assumption, only valid for Markovian, i.e. memory-less arrival processes, a premise *invalid* for IP networks. Arrival processes for IP networks exhibit (asymptotic second-order) Self-Similarity (SS) and Long-range-dependence (LRD) [2] [3] [12]. In short, burstiness of packet arrival processes (due to protocol features [13], user behaviour, heavy-tailed object sizes [14] or others [15]) persist over large timescales. While for a Markovian process the mean value (λ) unambiguously describes the process, for long-memory processes, second-order parameters as autocovariance and autocorrelation are dominant.

Consequently, the Estimator module, must apply an appropriate estimation technique for QoS parameters. We favour the technique in [11] and its improved version [16] based on [17] for our proposed architecture. Both techniques are so-called maximum variance techniques which derive a probabilistic model for the traffic aggregates worst case peak rate, while accounting for multiplexing gains. Based on this model remaining capacity in terms of bandwidth is calculated by subtracting the worst case peak rate from the available bandwidth.

An optimal MBAC algorithm admits a new flow only if the QoS demands for it, as well for all ongoing flows can be granted for each individual lifetime. Implicitly, an MBAC algorithm has to predict future traffic development. While SS and LRD effects are damaging for resource management, regarding prediction, strong correlations (LRD) are rather supportive. However, recent work draw the impact of LRD correlation structure beyond a certain time scale, called the Engineering Time Scale (ETS) in question [18]. Furthermore, the authors of [19] show that prediction of traffic with correlation structure of higher magnitude (LRD) compared with Short-Range-Dependend traffic does not significantly improve.

Regarding our proposed framework, currently we do not support explicit prediction techniques, which account for LRD but use the proposed, partly implicit, techniques integrated in the selected MBAC algorithms.

5 Admission Policy Module

Until now, packets are sampled, parameters measured and based on this current and future QoS parameters are computed. Finally, if predefined thresholds can be guaranteed, a waiting flow could be admitted.

Admitting whenever resources are available is certainly the simplest admission policy but leads to non trivial unfairness. Actually, fairness is still an unsolved issue in MBAC for multi-service networks [20]. Supported by the Internet typical traffic composition, small volume low rate traffic dominates the number of arriving flows, over a small number of high rate medium or large volume traffic [21] [22]. As a consequence, on a saturated link, mainly small quantities of resources are freed by the departure of small rate flows, which are high likely to be again occupied by small rate flows, cause of their higher fluctuation rate. High resource demanding flows, assuming static resource assignment, could therefore suffer persistent rejections.

One solution is to monitor the average arrival rate of both kinds of flows and according to, admit selectively regarding fairness. Another way, is to replace departured flows only by flows of the same class.

Besides of admission in accordance to resource availability, other factors like for instance economical (accounting) aspects could be taken into account for admission policies. Therefore, for our framework, we locate the final admission finding logic in a separate module called Policy module. This allows fast integration of different policies with minimal effort.

The Policy module interacts with the Selector module. Whenever the latter detects a new flow, it triggers the Policy module to initiate an admission process. The Policy module in turn triggers the Estimator module, which computes an estimation based on constantly recorded measurement of a limited, sliding time frame of the past. The estimates are back reported, and finally the Policy module admits, applying the defined admission policy.

6 Cooperative Queueing Module

In multi-services architecture like the Internet, different traffic classes have different arrival patterns over time. While, for example typical web traffic exhibits daily patterns [21] [22] (busy hour concept [23]), peer-to-peer applications might not, as they are rather non-interactive background traffic. For these, and other reasons aforementioned, resource consumption of IP traffic is subject to fluctuations of high magnitude. This makes static resource assignment difficult and inefficient. Furthermore, if clients are supposed to pay for streaming services, each rejection, due to insufficient resources, means lost revenue for an ISP. In the worst case, a client (flow) would be rejected although resources are currently unused, but (statically) assigned to an other class of traffic.

For our architecture, we propose henceforth, a cooperation between the queueing discipline and the MBAC algorithm, with streaming sources enjoying strict priority. Whenever a streaming flow, would be rejected, the Policy module additionally triggers a bandwidth consumption estimate for all other classes. In case of, disposable resources are dynamically redistributed to support the admission of the streaming flow. Assuming Wheighted Fair Queueing (WFQ), this is done be readjusting the weights. Furthermore, to guarantee the required loss probability and delay thresholds, buffer space is also subject to redistribution.

To integrate this feature, the Policy module and the queueing discipline communicate over a defined interface. This is further required, since the adaptive redistribution of resources could lead to a unfairness condition where a arrival burst of streaming sources occupy the majority of resources. To avoid this, the Policy module upper limits the number of admitted streaming sources due to resource distribution. To do so, it monitors the bandwidth distribution of the queueing system. If it falls below a minimal threshold, the Policy module temporarily suspends admission priority for streaming classes and prioritizes other classes. This leads to a redistribution towards the default configuration. Finally, it should be noted that streaming classes currently pose only ∼10 percent of total number of flows [21] [22]. Thus, the amount of redistributed resources is inherently limited.

Finally, it is worth to notice, that except of computing the weigths, there is nearly no extra computational effort needed. Dynamic adjustment, as performed on *call level time scale*, simply exploits existing estimations of the Estimator module for AC. With the integration of this scheme, we extend from to pure implicity AC to implicit AC with implicit and conditional resource reservation.

7 Conclusion and Further Directions

In this article we presented our *proposal* for a modular MBAC architecture for DiffServ edge routers. As we showed, the MBAC problem can be decomposed in a four module architecture, a Selector for packet and flow management, a Sampler for sampling and measurement, an Estimator for estimation and future prediction of QoS parameters and resource availability and finally a Policy module to apply various admission policies. Further, we argued that the logical separation of MBAC operation and service differentiation can be disadvantageous and propose a cooperation of both entities, where the resources are dynamically distributed according to expected demand computed by the MBAC estimator.

For each module, due to space limitations, we presented the main functionalities and the logical location in DiffServ edge router design.

Based on this concept, we are currently implementing the framework in the NS-2[2] network simulator for extensive and rigorous evaluation under various conditions. Based on the modularity, we aim to test the framwork in various configurations, by exchanging modules, to identify issues and their, close to optimal, solutions for given conditions and demands.

Acknowledgement This work was supported by the Portuguese Ministry of Science and High Education (MCES), and by European Union FEDER under program POSI (project QoSMap).

References

1. S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss, An Architecture for Differentiated Services, Request For Comments 2475, December 1998. http://www. ietf.org/rfc/rfc2475.txt?number=2475
2. W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, On the Self-Similar Nature Of Ethemet Traffic (extended version), IEEE/ACM Transactions on Networking, vol. 2, no. 1, pp. 1-15, 1994.
3. W. Willinger and V. Paxson, Where Mathematics Meets the Internet, Notices of the AMS, Vol 45, No.8, P961-970, 1998.
4. A. Riedl, Routing Optimization and Capacity Assignement in Multi-Service IP Networks, PhD-Thesis, University of Munich, 2003.
5. R. Mortier, I. Pratt, C. Clark, S. Crosby, Implicit Admission Control, IEEE Journal on selected areas in communications, vol. 18, no. 12, Dec. 2000.
6. M. Molina, N. Duffield, S. Niccolini, F. Raspall, T. Zseby, Sampling and Filtering Techniques for IP Packet Selection, Internet Draft, Feb. 2005. http://www.ietf.org/internet-drafts/draft-ietf-psamp-sample-tech-06.txt
7. F. Agharebparast and V. C. M. Leung, A New Traffic Rate Estimation and Monitoring Algorithm for The QoS-Enabled Internet, in Proc. IEEE Globecom03, San Francisco, CA, Dec. 2003.
8. G. Bianchi, F. Borgonovo, A. Capone, L. Fratta, C. Petrioli, Endpoint admission control with delay variation measurements for QoS in IP networks SIGCOMM Computer Communication Review archive Volume 32, Issue 2, P. 61-69, Apr. 2002.
9. S. Floyd, Comments on measurement-based admissions control for controlled-load services. Technical report, Lawrence Berkeley Laboratory, Jul. 1996.
10. S. Jamin and S. J. Shenker, Measurement-based Admission Control Algorithms for Controlled-Load Service: A Structural Examination. Technical Report CSE-TR-333-97, University of Michigan, (pp 58, 59, 108, 109, 160, 167, 171, 176, 186, 206), Apr. 1997.
11. E. Knightly and J. Qiu, Measurement-based admission control with aggregate traffic envelopes, Proceedings of IEEE ITWDC '98, Sep. 1998.
12. V. Paxson and S. Floyd, Wide-area Traffic: The Failure of Poisson Modeling, IEEE/ACM Transactions on Networking, pp.226-244, Jun. 1995.
13. P. Owezarski, N. Larrieu, Measurement tools and techniques for traffic and QoS management, International Conference on Integrated Modeling and Analysis in Applied Control and Automation (IMAACA'2004), pp.278-286, Bergeggi (Italy), 28-30 Oct. 2004.
14. K. Park, W. Willinger, Eds., Self-Similar Network Traffic and Performance Evaluation, John Wiley and Sons, Inc., 2000.
15. W. Willinger, M. Taqqu, R. Sherman, and D. Wilson. Self-similarity through highvariability: statistical analysis of Ethernet LAN traffic at the source level. In ACM Sigcomm '95, pages 100–113, 1995.
16. K. S. Seo, B. G. Li, Measurement-based admission control using maximum burstiness, IEEE Commun. Lett., vol.6, no.9, pp.403-405, Sep. 2002.
17. S. Valaee, A recursive estimator of worst-case burstiness, in IEEE/ACM Trans. on Networking, vol. 9, no. 2, pp.211-222, Apr. 2001.
18. G. Mao, Finite Timescale Range of Interest for Self-Similar Traffic Measurements, Modelling and Performance Analysis, IEEE International Conference on Networks, ICON'03, September 2003, pp.7-12, Sydney, 2003.
19. S. Ostring and H.Sirisena, The Influence of Long-range Dependence on Traffic Prediction, Proceedings of ICC'01, Helsinki, Jun. 2001.
20. L. Breslau, S. Jamin and S. Shenker, Measurement-based admission control: what is the research agenda?, in Proc. IWQOS, pp. 3-5, London, UK, May 1999.
21. K. Thompson, G. J. Miller, and R. Wilder, Wide-Area Internet Traffic Patterns and Characteristics, IEEE Network, Nov. 1997.
22. M. Fomenkov, K. Keys, D. Moore, and K. Claffy, Longitudinal study of Internet traffic from 1998-2003, Winter International Symposium on Information and Communication Technologies (WISICT), Cancun Mexico , 2004.

[2] http://www.isi.edu/nsnam/ns/

23. V. B. Iversen, Teletraffic Engineering Handbook, COM Center, Technical University of Denmark, ITC in cooperation with ITU-D SG2, 2005.