Darmstadt University of Technology

Admission Control based on Packet Marking and Feedback Signalling – Mechanisms, Implementation and Experiments

Martin Karsten and Jens Schmitt
Multimedia Communications Lab, Darmstadt University of Technology
Merckstr. 25 • 64283 Darmstadt • Germany
http://www.kom.e-technik.tu-darmstadt.de
Email: {Martin.Karsten,Jens.Schmitt}@KOM.tu-darmstadt.de

KOM Technical Report 03/2002
May 2002

Multimedia Communications (KOM)

Department of Electrical Engineering & Information Technology
Merckstraße 25 • D-64283 Darmstadt • Germany

Phone: +49 6151 166150
Fax: +49 6151 166152
Email: info@KOM.tu-darmstadt.de
URL: http://www.kom.e-technik.tu-darmstadt.de/

Admission Control based on Packet Marking and Feedback Signalling – Mechanisms, Implementation and Experiments

Martin Karsten and Jens Schmitt*
Multimedia Communications Lab, Darmstadt University of Technology
Merckstr. 25 • 64283 Darmstadt • Germany
http://www.kom.e-technik.tu-darmstadt.de
Email: {Martin.Karsten,Jens.Schmitt}@KOM.tu-darmstadt.de

Abstract The abstract system design and a prototype implementation of an admission control system is described. The system exploits the load signal generated by low complexity active queue management schemes at internal nodes to carry out flow admission control at edge gateways. The location of functional components is studied and appropriate signalling extensions to exchange load information are presented. The suitability of such signalling extensions for admission control and traffic regulation are discussed in the context of RSVP signalling. A number of questions of detail which are usually ignored by existing theory and simulation work, are examined and solutions are presented. Certain modifications to traffic control algorithms at both internal packet marking nodes and edge gateways are proposed and discussed. The functionality and correct system operation are demonstrated by experiments using the software prototype. Further, a variety of marking algorithms is compared experimentally to assess the suitability of their respective load signal for admission control of inelastic traffic and load-adaptive traffic regulation.

1 Introduction

Traditional network Quality-of-Service (QoS) systems have been designed and built based on *proactive* mechanisms like resource reservation that is enforced by some kind of prioritized scheduling of eligible packets [1,2]. Additionally, such systems are protected from excessive demand by flow admission control. In the absence of other effective and reliable backpressure schemes, this protection is crucial to ensure system stability even under extreme demand conditions. The traditional resource allocation model of the Internet, however, is better described as a *reactive* system, where intelligent end systems deduce information about the network's load respectively congestion situation by monitoring packet loss and/or packet marking [3] and react accordingly by throttling their output rate [4,5]. The combination of *flow control* elements at end or edge systems with *active queue management* (AQM) schemes at forwarding nodes may be regarded as QoS system, if convergence, stability and control goals are met. As a necessary condition for stability and convergence goals, traffic demand must be elastic on the scale it is controlled, such that it is amenable to output throttling in the first place. Further, cooperation is required, such that end systems indeed react to direct or indirect load signals. Besides the need to satisfy basic convergence and stability conditions, the real-world employment of such a reactive QoS system also requires that system stability results in service stability as experienced by users. Further, system convergence must be fast enough to mimic a transactional service response to clients of the system. In the absence of explicit network control, the system must allocate a fair share of capacity to each traffic flow. Otherwise the system should allow for controlled discrimination between flows, for example based on service classes.

The primary goal of this work is to design, implement and evaluate an admission control system that can efficiently offer a reliable guaranteed rate service for inelastic traffic, but without the need for per-flow state, per-flow signalling or any per-flow processing at internal nodes. This is achieved by combining admission control at edge gateways with AQM-based packet marking at internal nodes and feedback signalling between edge gateways. A secondary goal is to provide for load-adaptive resource allocation to flows, which is approached by aggregated flow control at edge gateways. Finally, another important goal is to study the behaviour of the system by means of prototype experiments in order to complement the signifi-

---

* This work is partially sponsored by the EU 5th Framework, IST Programme, Project 11429 “M3I”

cant amount of existing work that has already studied individual system components by means of mathematical modelling and simulation experiments. The main contributions of this work are as follows:

- A complete system design is presented, which allows to exploit AQM-based packet marking for admission control at system edges, without any per-flow state or per-flow signalling at internal nodes.
- The detailed design and location of admission control, load observation and load-adaptive traffic regulation is discussed and appropriate signalling extensions are specified.
- Certain AQM schemes would not perform well on ineligible traffic that is subject to packet discard. This problem is solved through *differentiated queue management* (DQM).
- It is experimentally shown that the general system design can offer stable rate guarantees with a variety of packet marking algorithms, albeit with also varying resource efficiency.
- It is experimentally shown that when choosing an appropriate packet marking algorithm, the system can effectively discriminate traffic without the need for differentiated scheduling in the core.

In the next section, related work is reviewed and the key differences to our approach are identified. Afterwards, the abstract system design is presented in Section 2. In Section 4, the actual implementation design is described along with implementation details. Section 5 reports the experimental investigation and its results. The paper is concluded in Section 6 with a summary and a discussion of our results and open issues.

2 Background and Related Work

There has been a vast amount of work on providing performance assurances to network traffic. The discussion of related work is structured according to the origin and direction of the respective contributions from the areas of flow control and AQM, measurement-based admission control, distributed flow admission control, edge-based admission control, as well as adaptive traffic regulation. An excellent overview of many further recent research results can be found in [6].

2.1 Flow Control and Active Queue Management

The current control paradigm of the Internet is composed of flow control elements at end systems and active queue management schemes at routers. The goal of this distributed resource allocation system is to share available resources efficiently and fairly. The current flow control concept is dominated by TCP’s congestion control algorithms and its different flavours [4]. This requires sources to be able to cope with different rate allocations (elastic traffic). In particular, the assumption of elastic traffic denotes concave utility functions [7]. Active queue management (AQM) is the notion of how to make decisions on discarding or marking which packets under which conditions. The primary goal of AQM schemes is to identify incipient congestion and to signal this to the flow control algorithms at end systems which then react accordingly. The most prominent example of an AQM scheme is Random Early Discard (RED) [8], which uses an exponential weighted moving average of the queue size and a piecewise-linear probability function over this average queue size to determine the discarding or marking probability for a packet. A more recent proposal called Virtual Queue (VQ) [9] proposes to virtually operate a queuing system at a lower capacity than the real system and to mark packets if the virtual queue overflows. Based on this proposal an adaptive version of this scheme called Adaptive Virtual Queue (AVQ) [10] has been suggested. AVQ adapts the capacity of the virtual queue according to the load situation. Although derived differently, alternative recent proposals like PI [11] and REM [12] arrive at similar solutions. Most AQM schemes base their decisions on a queue threshold which needs to be exceeded for packets to be discarded or marked (often randomly). The only AQM scheme known to us which provides direct load-based feedback is Load-Based Marking (LBM) [13] by calculating marking probabilities from the measured link load. A drawback of LBM, however, is that it is theoretically restricted to a single resource. All schemes use feedback and reaction as the main mechanism to fairly distribute resources in transient times of overload. There is work which is built upon such feedback and explicitly allows for the differentiation of traffic flows. Examples are:

- MulTCP [14] which enables weighted proportional fairness by acting like several TCP sessions.
- Generalized AIMD [15] which generalizes TCP’s AIMD scheme such that TCP is a special instance.
- TCP-WTP [13] which also provides weighted fairness, but based on a willingness-to-pay parameter.

Note that all of these schemes assume elastic traffic (with concave utility curves) such that no a-priori flow admission control is required. The other way round, there also is no way of carrying out reliable admission control as it would be desirable for inelastic traffic. In contrast, our work while similar to the above research focuses on admission control for inelastic traffic, which is an inherently different problem. Nevertheless, the admission control system described here is based on the same or very similar AQM mechanisms, which is considered an advantage over QoS systems requiring a completely different set of mechanisms.

2.2 Measurement-based Admission Control

In general, admission control schemes can be distinguished by how the admission decision is made:

- based on worst-case assumptions and resulting in deterministic guarantees,
- based on statistically relaxed assumptions and resulting in statistically controlled guarantees, or
- based on statistical measurements of flow behaviour and resulting in empirical guarantees.

The last approach is the one most related to our work and is commonly called measurement-based admission control. There has been a large amount of research on measurement-based admission control schemes: simple schemes like the measured sum algorithm described in [16], approaches like [17,18] which are based on the tangent at the peak rate of the equivalent bandwidth curve of a flow, or [19] which is based on large deviation theory. In [20], an extensive comparison of measurement-based admission control schemes results in the conclusion that all schemes perform fairly similar with respect to the utilization they yield.

While our work is similar to measurement-based admission control by taking into account past system behaviour, the admission decision here is based on indirect observations rather than direct measurements. Furthermore, in contrast to traditional approaches for measurement-based admission control, it is not a local decision for a single link but an admission decision for a whole path through a subnet including multiplexing with other paths and corresponding cross traffic effects – a much harder problem.

2.3 Distributed Flow Admission Control

A further criterion to distinguish different admission control schemes is given by the location where the admission control decision is made: at each forwarding node, at edge nodes between domains, at a centralized server, or at the endpoints of communication. Traditionally, admission control is performed at each node and only if all nodes accept a request, it is granted by the network. More recent admission control schemes do not require to involve all nodes on a path. Pioneering work in this direction, has been done by Kelly et al. [21,22]. Their analytical results show the basic stability of distributed admission control based on marking at resources even in the case of feedback delays [23]. Building on these results, there is work to shed more light on influence of delayed system reactions on stability, which presents bounds for the reaction delay [24,25]. In [26], a model for an Internet exclusively managed by end systems is presented and thoroughly analysed with respect to stability. More practical approaches, although similar in concept, are described in [27,28]. A simulative comparison of the basic design options for endpoint admission control is presented in [29]. In particular, [29] reports probing durations on the order of several seconds, whereas a recent simulative work [30] argues for much lower values for the initial probing phase.

Our work builds upon the basic stability results from the above research in that it was motivated by their encouraging theoretical results. However, in contrast to these contributions, it takes a more pragmatic approach by building upon the advantages of edge-based admission control and domain-based QoS systems:

- a controlled environment, in particular for traffic regulation and charging,
- observations instead of active measurements / probing,
- QoS provisioning is inherently a domain-based concept (each provider's choice).

2.4 Edge-based Admission Control

As just discussed, our architectural choice is for edge-based admission control, i.e., we assume independent domains providing QoS for elastic and in particular inelastic traffic flows by using admission control gateways located at the edges of these domains. We are not the first to follow this architectural paradigm, yet the different proposals (including ours) differ very much in their details and in the way they are analysed, whether being based on theoretical, experimental or just conceptual considerations.

In [9], Kelly et al. describe a system similar in concept to what they propose in [26]. The difference between these two is that in [9] an admission control gateway does the probing for the end systems whereas in [26] this functionality is distributed to the end systems. The authors regard the latter step as a refinement, however, these two proposals could also be viewed as independent evolution paths. The analysis of the system of admission control gateways in [9] is based on modelling and simulation and therefore abstracts from many real-world issues. While it is not the only goal and would from our perspective be a restricted view, our work could also be seen as an experimental validation of the theoretical insights from [9].

A DiffServ framework for edge-based admission control is described in [31]. It allows for traditional as well as measurement-based admission control. The measurement-based part is based on packet marking at core routers. In contrast to our work, the feedback is generated per-flow while in our case it is aggregated per path. Furthermore, the proposed marking schemes are not evaluated, neither theoretically nor experimentally, in their interplay with admission control schemes, owing to the purely conceptual nature of [31].

In [32], Knightly et al. present an egress-based admission control architecture based on monitoring traffic characteristics per path at egress nodes. These measurements are based one-way per-packet delay measurements, which is all but trivial. Such measurements then allow to make an admission control decision based on the concept of statistical traffic envelopes. The core network is viewed as a black box and in contrast to our work gives no feedback on the current network load. Yet, with the minimal feedback as it is provided in our system the admission control procedure can be made more simple and robust. The admission control in [32] is claimed to work well for self-similar traffic, which is demonstrated via the use of Pareto-distributed on-off sources with Pareto shape parameter 1.9. However, this (approximately) results in a Hurst parameter of 0.55 for a corresponding aggregate – a very moderate degree of self-similarity.

2.5 Adaptive Traffic Regulation

While our work is mainly concerned with admission control for inelastic traffic flows, it also takes into account the requirements of elastic traffic and of traffic that combines both characteristics by requiring a basic amount of service while benefiting from additional service. The need for such a hybrid service has been realized before and we do not claim to be the first to propose appropriate mechanisms, but want to demonstrate that such a service can be elegantly realized. The key to providing service to such *bounded-elastic traffic* is load-adaptive traffic regulation. An edge-to-edge, per-flow traffic conditioner based on congestion marking of control packets is presented in [33]. The scheme necessitates per-flow rate estimation at edge nodes and provides no admission control. In contrast, our work employs AQM-based packet marking instead of explicit control packets and the system adaptation is based on aggregated feedback instead of per-flow. The Aggregate Flow Control (AFC) scheme [34] can be described as a combination of TCP trunking [35] and a token bucket regulator in the framework of DiffServ's Assured Forwarding (AF) per-hop behaviour. The work is focused on details of relative resource allocation between aggregates and reported improved service for TCP and UDP under certain conditions.

Very close to the adaptive traffic regulation part of our work is [36], which proposes an adaptive token bucket regulator controlled by the costs of a path, which depend upon the load level on each link of the path. However, the computation of the cost function is tied to routing protocols which usually operate on a much coarser timescale than necessary for the distribution of up-to-date load information. Our approach addresses that practical problem. Since the focus of [36] is on traffic regulation and load balancing, no admission control scheme is proposed.

3 Abstract System Design

Since many aspects of the implementation design, for example the choice of the signalling protocol, are not governed by fundamental requirements, but rather chosen according to their practicality for implementation, experimental investigation, and later deployment, an abstract system design is presented in this section. The system requires two bits in the IP packet header, such as the ECN bits [3]. For the presentation the

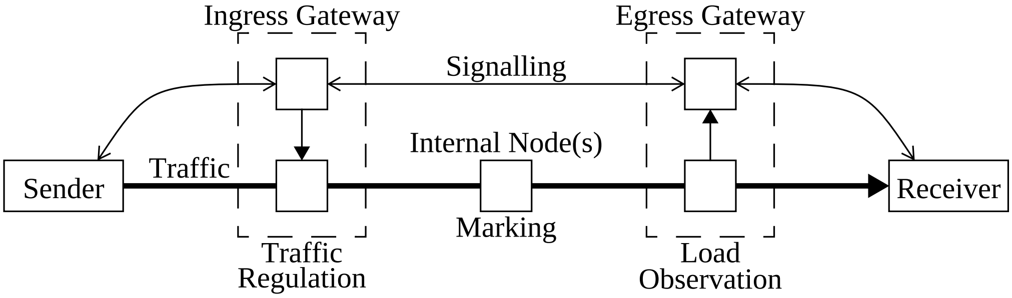

Figure 1: System Overview: Load Control Gateways and Internal Nodes

terminology of ECN is adopted and the prototype actually uses these two bits. However, the abstract system design is of course not bound to using these specific bits. For example, it could as well be implemented using two other bits from the available space of DiffServ code points [2].

3.1 Overview

The system is domain-oriented with *load control gateways* at the edge of the network reacting to signalling requests and carrying out admission control, traffic regulation and path load estimation. *Internal nodes* only perform packet forwarding on a first-in first-out (FIFO) basis. The packet queues of internal nodes are controlled by AQM schemes, which discard or mark packets depending on the current load or queuing situation. Gateways operate in the roles of both *ingress* and *egress* gateways, depending on the direction of traffic flows. An ingress and an egress node connected through a routing path in the network are termed *peers*. Figure 1 presents the different roles of system components along the transmission path.

Ingress gateways control traffic on a per-flow basis through (modified) token bucket regulators and egress gateways collect load information on a per-peer basis by inspecting the packets arriving from the network. No specific precautions are taken to control the delay of packet transmissions other than the overall goal of keeping the queue lengths as short as possible. It is well possible to combine this system with specific scheduling regimes, though. In general, there are multiple scenarios to employ this system. First, the admission control part of the system can be separately applied to a dedicated service class, for example in the framework of DiffServ [2]. Alternatively, the full system might be used to manage resources of a common traffic class and to offer distinguished services to certain traffic flows, using only admission control or a combination of admission and flow control. Further, the system can be employed in a multi-path routing scenario by considering the endpoints of each routing path as virtual peers.

3.2 Load Control Gateways

The load control gateways implement the control path of the system, which employs a request signalling protocol between gateways and clients as well as between peering gateways.

3.2.1 Location of Functionality

Either the ingress or the egress gateway has to calculate an admission control decision. The egress gateway observes the system's relative load along a path through the network, while the ingress gateway is the only place to carry out traffic regulation. Therefore, if the admission control decision is taken at the egress gateway, it has to be reported to the ingress to become effective. Alternatively, the egress gateway can send load reports to the ingress, which then decides about admission and possibly installs the request. Such load reports can be sent periodically or be triggered by a service request. In any case, there is a delay between load observation and the installation of a request at the traffic regulation module. Consequently, the system's significant reaction delay is independent of whether the admission decision is done at egress or ingress.

3.2.2 Packet Handling and Signalling

The ingress gateway is responsible for traffic regulation according to the negotiated service contracts by marking conforming packets with the ECT bit. Non-conforming packets can be discarded to enable strict

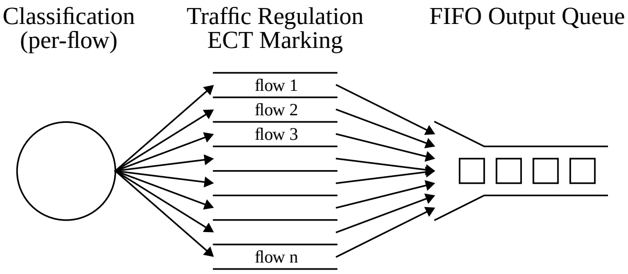

Figure 2: Traffic Regulation at Ingress Gateway

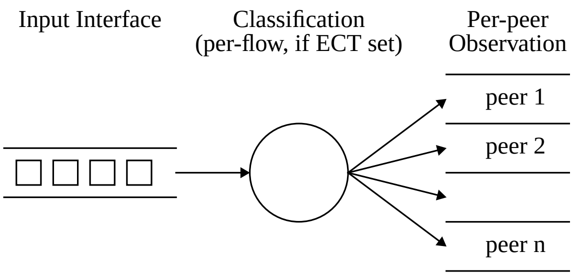

Figure 3: Load Observation at Egress Gateway

policing. As shown in Figure 2, the ingress gateway performs per-flow classification of incoming packets to find the appropriate traffic regulation instance. The egress gateway observes the relevant information for load estimation as depicted in Figure 3. Incoming packets which have the ECT bit set are classified to determine the sending peer (ingress). Then, per-peer statistics containing the number of packets, number of marks and number of bytes, as well as the duration of the observation interval, are updated.

A signalling protocol for simplex traffic flows is suitable, because of the different roles of ingress and egress gateways. There is no fundamental problem with using a duplex protocol, other than increased signalling complexity between ingress and egress for both traffic directions. As well, there is only little preference for a simplex protocol to be sender- or receiver-oriented. In case of a receiver-oriented protocol, the coordination information can be piggybacked onto each signalling request, which is transmitted from the egress to the ingress. In case of a sender-oriented protocol, an additional information exchange is necessary to coordinate egress and ingress to either report load information or to transmit the admission decision.

3.2.3 Admission Control, Reaction Delay, and Overbooking

The information observed by the egress gateway is used by the admission control procedure to estimate the relative load along a transmission path as the fraction of marked packets from total packets received during a recent time period. A request is admitted, if this fraction does not exceed a certain threshold Λ. The nature of the system being reactive requires that a safety margin must be accounted for by this admission test, because there is a feedback delay between the actual load situation in the network and the installation of a new request, and vice versa. Further, there exists a potential problem of unnoticed overbooking. If sources send less traffic than initially negotiated, the observed path load does not account for the unused but booked capacity, which might lead to excessive overbooking. It might be beneficial to allow a controlled amount of overbooking, therefore the admission test includes a parameter to configure the relative amount of overbooking. Let l be the estimated relative load along a transmission path, c the total booked transmission rate and u the actual used rate. The adapted estimated relative load $\hat{l}$ is then calculated as

$$\hat{l} = l \times \left(1 + \alpha\left(\frac{c}{u} - 1\right)\right) \tag{1}$$

with $\alpha \in [0,1]$ determining the relative influence of booked but unused capacity. A small value for α denotes an optimistic system configuration in which the potential overbooking is largely ignored and a large value for α denotes a conservative setting. Note that this calculation implicitly assumes that the overbooking situation at multiple gateways is roughly similar, otherwise a more complicated mechanism is needed. The actual value of α depends on the behaviour of traffic sources and can probably only be determined by long-term observations of an operational system. Such observations could then allow to devise a much more sophisticated and statistically tractable estimator than (1).

3.3 Internal Nodes

Internal nodes run an AQM scheme to mark or discard packets. We have chosen RED, VQ, AVQ and LBM as representatives of the large number of AQM schemes for initial inclusion into the prototype. Special attention is paid to LBM, because being the only load-based AQM scheme, it is able to deliver reliable load information independently of traffic burstiness. This is an interesting characteristic when it comes to ac-

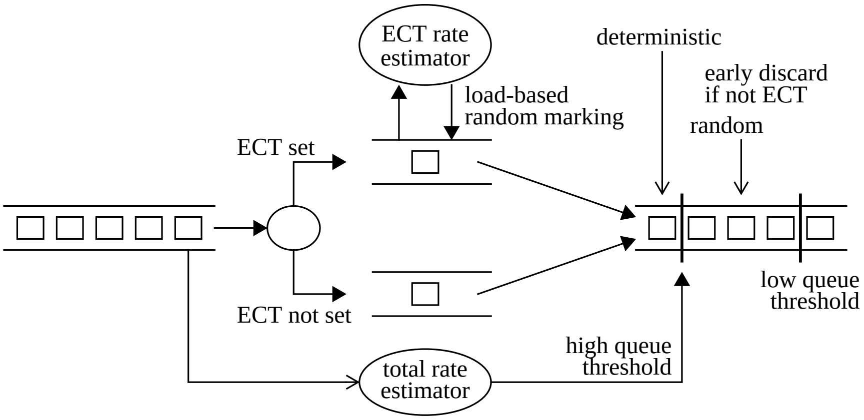

Figure 4: Differentiated Queue Management (DQM) for Load-based Marking (LBM)

commodating inelastic traffic. Because LBM and VQ would lead to quite aggressive discarding of non-ECT packets, both are not well-suited to operate simultaneously on ECT and non-ECT traffic. This problem is approached by *Differentiated Queue Management* (DQM), shown in Figure 4 for the case of LBM. Instead of using a single probability function to decide if a packet is being discarded or marked, DQM first differentiates between ECT and non-ECT packets. Then, two separate algorithms are used to determine whether the packet is marked respectively discarded. Only the rate of ECT packets is used as input for the marking algorithm, since non-ECT packets are regarded as background traffic. Non-ECT packets are subject to a rate-dependent early random discard (ERD) algorithm. The high queue threshold is adjusted inversely linear to the total arrival rate. If the queue length is between the low and high threshold, the packet is discarded randomly. If the queue length exceeds the high threshold, the packet is always dropped. A variant of RED could also be used to control non-ECT traffic, but for simplicity we opted for ERD.

In [13], LBM is proposed as linear marking algorithm for a single resource. The main problem with employing LBM on multiple resources along a path is that a flow traversing multiple resources is subject to each resource's marking probability $m(x_i)$ with x_i being the relative load at resource i, while at the same time, there is only a single marking bit available in the packet header. Thus, the path marking probability $M(p)$ for a path p of resources is given by

$$M(p) = 1 - \prod_{i \in p} (1 - m(x_i)) \tag{2}$$

which might be significantly higher than the highest individual marking rate. To avoid over-estimating the path load, the admission control test is based on an estimate for the average load of a path, calculated as

$$\bar{L}(p) = m^{-1}\left(1 - \sqrt[n_p]{1 - M(p)}\right) \tag{3}$$

with n_p being the number of resources on path p. Note that this assumes an invertible marking function $m(x)$. On the other hand, (3) might lead to an under-estimation, if only a small number of nodes are significantly loaded. To accommodate for this disadvantage, the system uses an exponential marking function at internal nodes to increase the influence of highly-loaded resources. The same weighting principle is inversely used in [12] for rapid and precise congestion detection. As an additional safety margin, the LBM component of DQM can be configured with a load threshold, above which all packets are marked.

In contrast to LBM, all other AQM schemes draw a marking decision which eventually depends on the current queue length or its variation. Therefore, a corresponding discard decision can be expected to interoperate well with TCP-like flow control in end systems (in fact, RED has initially been designed to do just that). Therefore, DQM is not applied to these marking schemes. Further, a queue-oriented load signal is not repeatedly generated at subsequent nodes along a path, if no independent occurrences of bursty traffic arrival exist. Consequently, the notion of an average load value is only necessary for LBM. On the other hand, LBM has another particularly interesting feature. Since it does not operate on the state of a queue, it can be used to signal a node’s processing load to the network edges, rather than information about a link’s queue. This may better reflect the actual bottlenecks in today’s Internet.

3.4 Load-Adaptive Traffic Regulation

One of the system’s goals is to offer a load-adaptive service to clients, that is, clients can request a basic service rate, but might be allowed to exceed this rate, if network capacity is available. A modified version of the token bucket algorithm is used to control the amount of traffic entering the network. A standard token bucket regulator (TBR) is characterized by depth d and rate r and the amount of available tokens t is calculated for each packet transmission as

$$t_{new} = \min(t_{old} + \tau \times r, d)$$

with τ being the time interval between the current and the previous packet. To offer a load-adaptive service, similar to other proposals, we propose an *Adaptive Token Bucket Regulation* (ATBR) algorithm which additionally includes a scaling factor s. The amount of tokens is then calculated as

$$t_{new} = \min(t_{old} + \tau \times r \times s, d) \text{ with } s = \frac{\Lambda - \varepsilon}{\bar{L}} \quad (\Lambda \text{ is the admission threshold and } \bar{L} \text{ is the load}).$$

The scaling factor s is determined by the estimated load along the path and allows to temporarily exceed a request’s basic rate allocation when the network is lightly loaded. It is however necessary to avoid the system to be fully loaded with excess traffic from scaled token buckets, because load control gateways cannot distinguish between regular traffic load and such excess load. If scaling of ATBRs were not limited, the excess traffic could increase the network load above the admission control threshold. Incoming requests would then be rejected, although resources are still available in principle. In order to maintain priority of incoming service request, the admission control threshold Λ minus a small safety margin ε is divided through the current relative load estimation $\bar{L}$ and used as scaling factor for the token buckets.

To explain the basic rationale for the adaptation of the scaling factor s, consider the case of LBM as marking algorithm at internal nodes. Assuming that the sum of basic rate allocations is less than a certain fraction of the capacity (expressed through the admission control threshold), then at the same time, this is true for the sum of all scaled rate allocations, as well (at least in case of an invertible marking function as discussed in Section 3.3). In other words, the maximum amount of marks that a flow is responsible for, is kept constant. Fairness between multiple flows is obviously given, because each flow’s service rate allocation is proportional to its requested rate. Packets that are not conforming to the token bucket are forwarded without the ECT bit set and they are then subject to the differentiated discarding algorithm at internal nodes. Note that in this case, it is assumed that internal nodes do not reorder packets. However, this condition is met by all AQM schemes under consideration. Essentially, the combination of ATBR with load-based packet marking allows to integrate the different needs of inelastic traffic and load-adaptive network service.

3.5 System Complexity

The worst-case complexity of an ingress gateway is comparable to that of a DiffServ ingress node. Per-flow classification is required prior to traffic regulation. Egress gateways have a potentially higher complexity than their DiffServ counterparts, since per-flow classification is needed for packets leaving the domain, as well, in order to collect traffic and marking statistics per peering gateway. Internal nodes have a complexity of O(1) in the number of flows, because all AQM schemes are stateless with respect to flows. Internal nodes do not participate in the signalling protocol, so they indeed operate with constant complexity.

4 Implementation

The signalling, admission control and load reporting functionality is implemented in the framework of the publicly available KOM RSVP implementation [37]. The data path modules are implemented in the ALTQ software framework, which is publicly available, as well [38]. The system has been developed and tested on FreeBSD. The extensions presented here will also be published as open-source software. To our knowledge, no such comprehensive reactive resource allocation system exists as a real system prototype.

4.1 Control Path

As a receiver-oriented simplex signalling protocol, RSVP is very suitable to implement the control path of the abstract design for load control gateways as presented in Section 3.2. The protocol is extended by a few local elements to enable operation as signalling protocol between load control gateways. A new message object is used to transport load information from egress to ingress gateways, specified as:

```
LOAD_REPORT ::= <packet count> <mark count> <byte count>
                <time interval> <hop count>
```

The information contained in this object describes the load situation along a path through the total number of packets and the number of marks received during a recent time interval. The number of transmitted bytes and the length of the observation interval are reported, as well, such that the ingress gateway has precise information about the transmission rate during the observation interval. This information is necessary to carry out the adapted load estimation, presented as Equation (1) in Section 3.2. While it would be possible for the ingress gateway to measure the transmission rate locally, it would then also be necessary to associate the measured transmission rate with the corresponding load situation which is observed at the other end of the network domain. This complexity can be avoided by reporting the full information from the egress gateway, which has to inspect all incoming packets anyway. The egress gateway can determine the number of hops on the path from the ingress by comparing the TTL values in the IP and RSVP headers of a path message. The number of internal nodes is needed to compute the estimated load as specified in Equation (3) in Section 3.3. To accommodate for potential dynamic routing changes, the current hop count is always reported to the ingress gateway as part of the load report.

All information included in the load report, except the hop count, is gathered from a kernel observation module. A `LOAD_REPORT` object is included into each reservation message sent from an egress to the respective ingress gateway. If no reservation message is transmitted for a certain period of time, the egress sends periodic messages to report the current load situation to the ingress. These reports are transmitted as a dedicated message type, termed *Load* message and contain the egress gateway's `RSVP_HOP` information and the current load situation in a `LOAD_REPORT` object. Periodic load reporting is a fallback mechanism for times of little signalling activity and ensures that the ingress gateways always have proper load information to adjust the setting of the ATBR modules as described in Section 3.4. Additionally, all packets carrying signalling messages are marked with the ECT bit and are subject to load measurement and marking at internal nodes. Thus, periodic load reporting generates a small traffic stream, which allows to observe load information, even when no other traffic is present between peers. Thereby, a gateway has at least some load information available at the very beginning of the next busy period. In the prototype, the RSVP daemon is extended to create and process the above protocol elements and to appropriately interact with the gateway kernel-level module, which implements the actual handling of data packets.

4.2 Data Path

The system's data path is implemented as ALTQ output queuing modules. The `LCG` (load control gateway) module implements the functionality of an ingress and egress gateway. On the ingress side, packets are classified and marked with the ECT bit, if they conform to the token bucket specification of a known flow. Packets exceeding the token bucket specification are either dropped or forwarded with a cleared ECT bit, depending on the system's configuration. The egress gateway part of the `LCG` module implements traffic and load observation. First, an incoming packet's ECT bit is checked and if set, the packet is classified to identify the peer it comes from. Afterwards, the length of the packet, its arrival time and its setting of the CE bit are recorded in a fixed-size ring buffer. The total number of packets, marks, bytes and the length of the observation interval are updated before replacing the oldest packet's information. Thereby, the length of the observation interval reflects the time period, during which the n last packets arrived with n being the configurable size of the ring buffer. Additionally, a maximum lifetime of packet information can be specified to avoid keeping stale information during idle times with only few packet arrivals. Since this cleanup functionality is only invoked during such idle periods, it is not considered to decrease performance.

The DQM/LBM functionality is implemented in the `DQM` (differentiated queue management) module. Similar to the `LCG` module, packet information containing a packet's length, arrival time, ECT setting and discard information is kept in a configurable ring buffer of fixed size, as well as in counters that are updated for every arriving packet. If an incoming packet has the ECT bit set, the rate-dependent packet marking decision is drawn from a pre-calculated table with the values of the marking probability function. Otherwise the packet is subject to the ERD algorithm as presented in Section 3.3.

When implementing other AQM schemes, it turns out that some of the schemes introduce rather unrealistic challenges. In the AVQ scheme, the service rate of the virtual queue is adapted according to the current traffic arrival rate. The proposed algorithm in [10] to update the virtual service rate estimates the arrival rate based on just a single packet arrival, which would require a very high clock precision. Even then, an arriving packet train or a significant variation of packet sizes might lead to erroneous estimation of the arrival rate. To implement this scheme on a real platform, it is necessary to estimate the traffic arrival rate by averaging over multiple packets. The VQ algorithm in [9] contains another problem. When the virtual queue exceeds the marking threshold, all packets that are currently queued in the real system are marked. However, a write operation on every queued packet is at least a very costly operation on real router platforms, if not prohibitively expensive. The prototype contains AVQ and VQ modules with appropriate modifications.

4.3 Test Environment

We have created a sophisticated yet inexpensive software-based test environment for the experiments. The software part of the test environment is already available as part of the KOM RSVP implementation [37]. A traffic emulator has been implemented, which is capable of creating an arbitrary number of CBR, (aggregated) Pareto on/off or greedy sources with fixed or exponential arrival patterns. It initiates RSVP signalling and keeps track of the reservation success. The traffic sinks keep extensive statistics about packet delays, goodput rates and packet drops. All nodes' clocks are synchronized by a single GPS clock and packets are time-stamped at the kernel level. Load control gateways and internal nodes provide proper logging information about their actions, as well as status information, such as the estimated load and queue lengths. Because all clocks are synchronized, these values can be associated with each other, despite the observations being distributed over multiple machines.

5 Experimental Investigation

All experiments are carried out in the topology shown in Figure 5. The link between node 8 and 7 is the potential bottleneck link. Nodes 4-6 are load control gateways. All nodes are standard PentiumIII/450MHz PCs running FreeBSD 4.5, enhanced by network driver polling [39]. Links operate full-duplex at 10 MBit/s. The systems' clock rate is set to 1000hz and fast forwarding of IP packets is enabled. The FreeBSD network code is slightly modified to ensure that packets from crucial network services, such as routing or address resolution, are not subject to any traffic control or policing action.

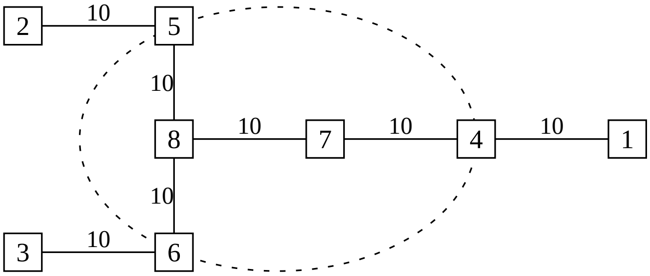

Figure 5: Experiment Topology

5.1 Admission Control and Resource Utilization

In this first series of experiments, RED, AVQ and DQM/LBM are assessed individually and compared with respect to their suitability to carry out admission control at the network edge. At the same time, the overall system design is verified to operate correctly in combination with these marking schemes.

The basic experiments consists of deterministically generated VoIP-like flows with an inter-arrival time of 0.5 seconds and a duration of 50 seconds, which in total exceed the available transmission capacity. Additional background traffic is generated from aggregated Pareto sources with a Hurst value of 0.8. For all marking schemes, the basic experiment is repeated with varying configuration parameters. As mentioned in

Section 3.1, the abstract system can be used in multiple scenarios. To represent two of these different scenarios, experiments are run with and without background traffic. To illustrate the experiment and to calibrate the results, the same experiment has also been carried out in an IntServ-like configuration where all network nodes participate in RSVP signalling and perform local admission control and per-flow HFSC scheduling. The HFSC module is configured to accept a high number of sessions to achieve a high resource utilization. The result is illustrated in Figure 6 and shows that the network discriminates perfectly between signalled and background traffic. The small periodicity in the number of offered and accepted sessions stems from the periodic session generation and removal in combination with the fixed session duration. If a signalling request is rejected, the session is terminated after a short timeout. Therefore, the offered session curve is always near to the accepted session curve and not shown in subsequent figures.

Before starting the actual comparison of AQM schemes, a large number of experiments has been conducted to study the feasible region of averaging ring buffer sizes in both core and edge nodes. The results of these experiments have shown that buffer sizes between 500 packets and 4000 packets are feasible for the experimental prototype. Below 500 packets, the system reacts very nervous which is likely being caused by both measurement and statistical inaccuracies. Above 4000 packets, the reaction delay of the system significantly decreases the overall performance. For the experiments, all buffer sizes are set to 1000 packets.

5.1.1 RED

The usual parameter settings for RED [40] are targeted at providing drop feedback to elastic TCP sources and thus, are likely to be not appropriate for a reactive admission control system to provide reliable service without packet loss. Therefore, the RED experiments use two alternative configurations and test these with varying admission control thresholds. In the ‘default’ setting, th_{min} is set to 5 packets, th_{max} to 15 and p_{max} is set to 0.1. In the ‘aggressive’ setting, the configuration is $th_{min} = 2$, $th_{max} = 20$ and $p_{max} = 1$. In both cases, the queue size is set to 100 packets and the weight parameter set to 512. As expected, it turns out that the default RED configuration cannot support reliable admission control for inelastic flows. Instead, even for a very small admission control threshold, too many sessions are accepted and then subject to massive packet losses. Using the aggressive configuration, Figure 7 and Figure 8 show the results of successful RED experiments with and without background traffic respectively, to illustrate its operation. The reactive nature of the system becomes apparent by observing the periodicity introduced by the session generation process. Essentially, the slope and length of the increase and decrease segments of the accepted load curve represent the feedback delay in the system. The system is able to manage network resources, but its capability to discriminate traffic is clearly limited in the presence of background traffic. In fact, sessions exhibit an accumulated packet loss in the order of thousand packets.

5.1.2 DQM/LBM

The LBM module is configured to mark packets along the full range of forwarding capacity. The ERD components varies the upper discard threshold when the arrival rate is between 50% and 90% of the capacity. The low discard threshold is set to 20KB. One of the results of the LBM experiments is shown in Figure 9. The system is capable to accept a high number of session while at the same time, effectively discriminating between reserved and background traffic, and thus, performs much better than in the RED configuration. This can be attributed to the DQM extension of the basic LBM scheme. Because of this discrimination capability, the system performs equally well with and without background traffic (not shown). Again, the reactive nature of the system becomes apparent by observing the periodicity introduced by the session generation process.

5.1.3 AVQ

Experiments with AVQ and background traffic have revealed that the suitability to control resources with AVQ in the context of such a feedback-based admission control system is clearly limited. AVQ adapts the forwarding speed of the virtual queue depending on the traffic arrival rate. If the arrival rate is calculated from all packets, the unresponsive background traffic leads to a slow speed of the virtual queuing system. This results in aggressive discarding of background packets and marking of ECT packets and consequently, only a relatively small number of sessions is accepted. Therefore, appropriate resource utilization can nei-

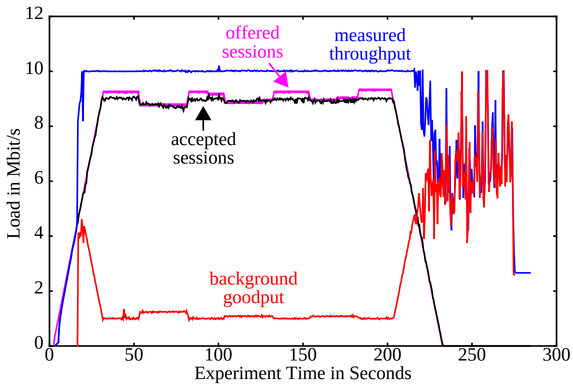

Figure 6: IntServ with Background Traffic

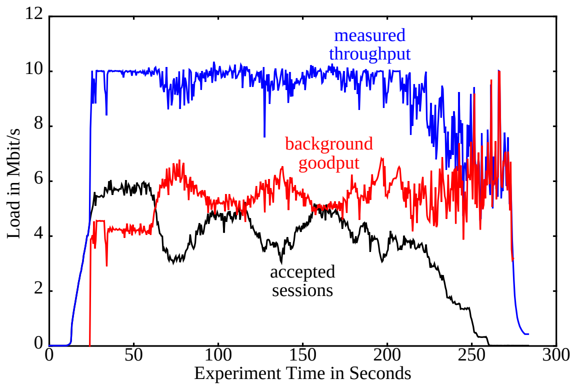

Figure 7: RED with Background Traffic

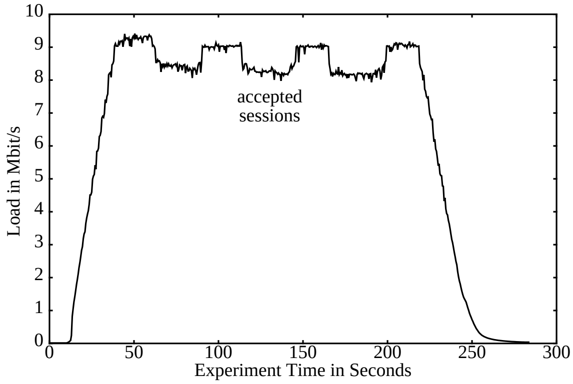

Figure 8: RED without Background Traffic

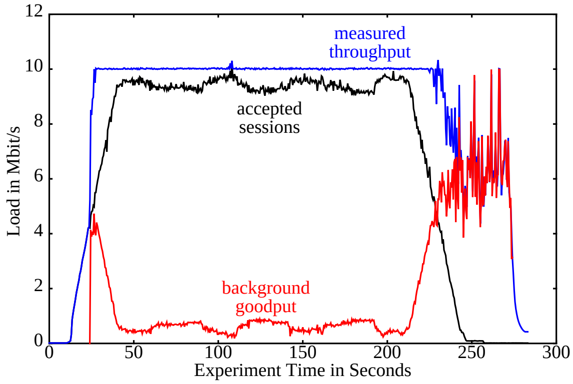

Figure 9: DQM/LBM with Background Traffic

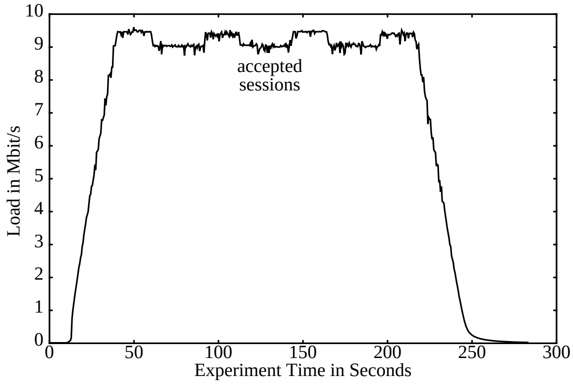

Figure 10: AVQ without Background Traffic

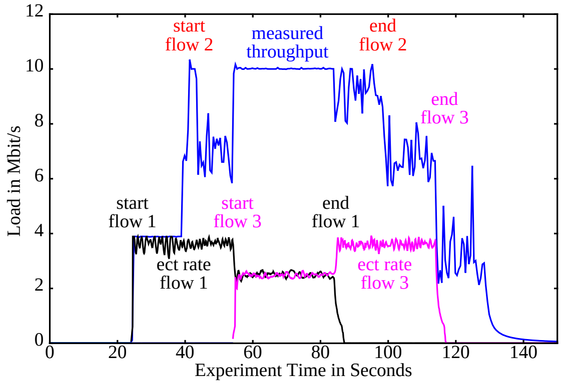

Figure 11: Load-Adaptive Traffic Regulation

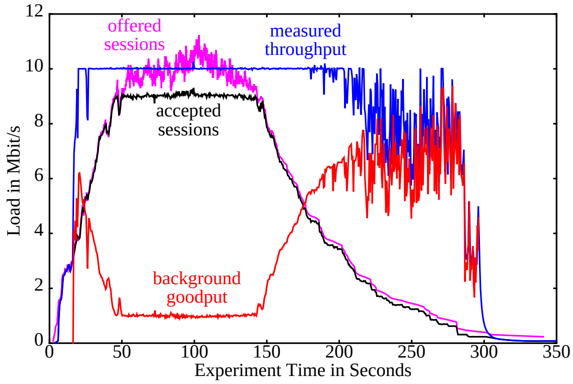

Figure 12: IntServ and Random Session Arrival

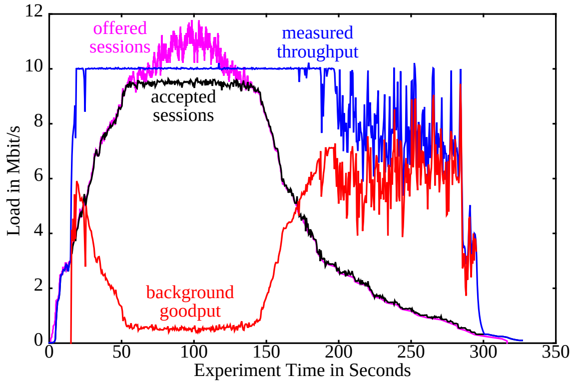

Figure 13: DQM/LBM and Random Session Arrival

ther be achieved with reserved nor background traffic. If, however, the arrival rate is based on ECT traffic only, the virtual queue runs quite fast and the AVQ module does not discard enough background packets, such that both types of packets are subject to later forced drops because of an overflow in the real queue. On the other hand, when controlling dedicated resources with AVQ, the system shows a faster reaction and a more deterministic behaviour than both with LBM (not shown) and RED. This is shown in Figure 10 for a configuration where α is set to 1 and the virtual capacity to 90%. The system is effectively determined by the speed of the virtual queue and the actual queue size has only little influence on the overall performance.

5.2 Load-Adaptive Traffic Regulation

To verify the operation of load-adaptive traffic regulation in the context of LBM packet marking, an experiment has been carried out with a small number of bigger flows to study the system's behaviour. Two reserved and two background flows are started with a certain time interval in between to observe the reaction of the system to the changes in demand. Flows 1 and 3 are the reserved flows, but inject more traffic into the system than signalled. The result is depicted in Figure 11 and shows the ECT marking rate at edge nodes for both signalled flows, as well as the total throughput as measured at an internal node. It is apparent that the system indeed correctly regulates traffic by means of ECT marking at edge nodes according to the current observed load situation. However, the system's reactions are quite nervous and further experiments have shown indications for a certain impact of the averaging buffer sizes and the load report periods on this behaviour. To this end, a detailed study of load-adaptive traffic regulation remains an issue for further research, particularly with respect to its interaction with TCP-like flow control in end systems.

5.3 System Performance

To fully assess the system's performance potential, the DQM/LBM variant is compared to an IntServ configuration under more challenging demand conditions. In contrast to the experiments in Section 5.1, the session arrival is now distributed exponentially with the same average values as before. Sessions are persistent, so in case of rejection, they retry signalling after a short random back-off period. The results of this experiment are shown in Figure 12 and Figure 13 respectively. It is evident that the performance of the reactive admission control system is very similar to that of the IntServ system. However, it must be noted that the reactive system can by itself not provide differentiated delay guarantees, while an IntServ network is capable of doing so.

5.4 Comparison

To further investigate and compare the behaviour of the admission control system employing the different AQM schemes, we have studied the influence of the admission control threshold on utilization and delay. The input for Figure 14 is taken from the experiments without background traffic described in Section 5.1 Each experiment has been carried out 10 times for each admission control threshold and the values shown in Figure 14 are the average results. In these experiments, the total number of accepted sessions is around 480 when the utilization is very high, for example as shown in Figure 6, 8, 9 and 10. Clearly the average delay directly depends on the utilization in this case. The results show that utilization and delay are sensitive to the admission control threshold when employing DQM/LBM and RED. In a scenario where this system is used to control access to a dedicated service class, this allows to achieve certain delay bounds in exchange for lowered utilization. DQM/LBM enables the full range of delay/utilization pairs while RED only covers a part of it. The admission control system employing AVQ is insensitive to the admission control threshold. Instead, utilization and delay depend on the speed of the virtual queuing system expressed as the fraction of the real transmission rate. This is not shown in the figure, but has been established by additional experiments. In theory it can thus support all configurations that can be supported by the DQM/LBM system. In practice however, a change in the system configuration then requires to modify the configuration of core nodes when using AVQ, instead of edge nodes when using DQM/LBM. For all alternatives, the worst-case queue length at internal nodes has been observed during the experiments, as well. It is not shown in Figure 14, but it behaves correspondingly to the average end-to-end delay.

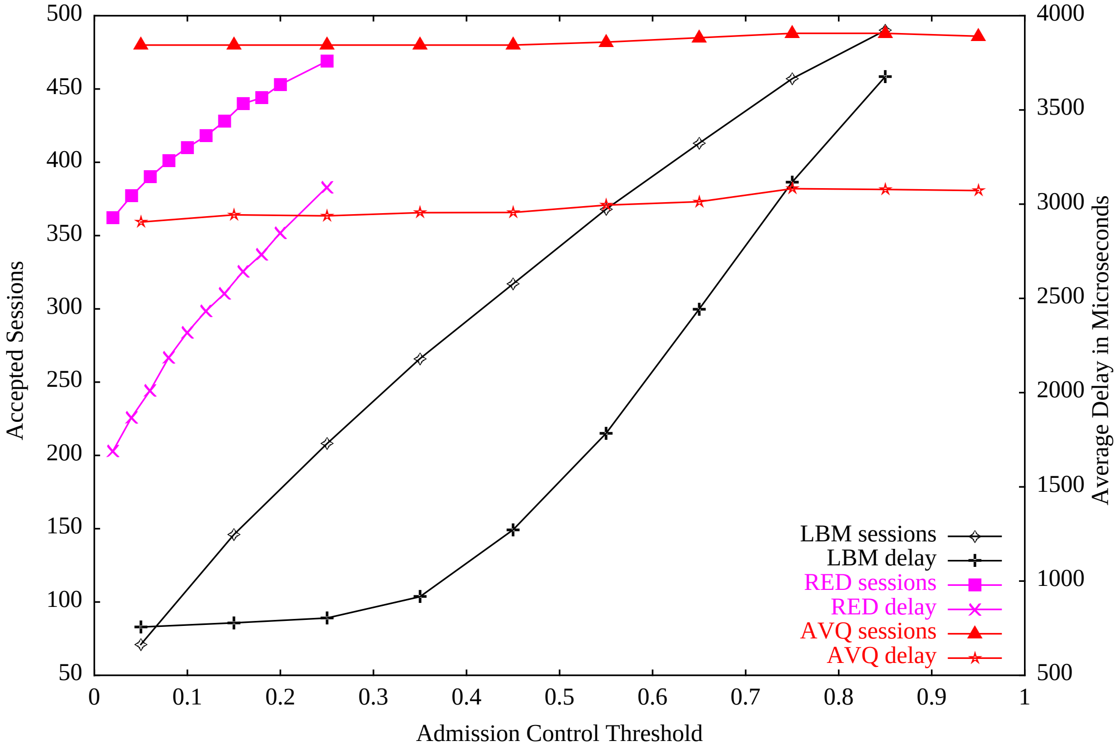

Figure 14: Admission Control with Background Traffic - Summary

6 Summary and Outlook

We have presented the abstract system design and a prototype implementation of a reactive QoS system employing binary AQM-based packet marking at internal nodes and load observation, feedback signalling, traffic regulation and admission control at edge gateways. The system has been studied by means of lab experiments to verify its basic operation and to complement existing theoretical and modelling work on feedback-based resource allocation. In principle, the experiments with the system prototype confirm earlier results about the feasibility of such a design, however, it also becomes evident that life is not always simple and easy. First, the choice of a packet marking scheme influences the capabilities of the overall system with respect to admission control and traffic discrimination at a high resource utilization. While DQM/LBM performs best for those scenarios that have been considered here, it is not clear whether the proposed exponential marking function is sufficient to enable load-based marking across multiple marking nodes. Early experiments which are not reported here in detail, have shown promising results, but it is beyond the scope of this paper to fully investigate this issue. The details of interaction between TCP or TCP/ECN with the admission control system as well as with adaptive traffic regulation poses another set of interesting open questions. A second source of complexity for the overall system design is given by the large number of parameters that have to be configured for an operational system. Particularly, different marking algorithms have a different sensitivity to various configuration parameters. Nevertheless, the results of our work also clearly show that reactive resource allocation is a promising concept to offer stable performance guarantees when combined with admission control. Initially, we assumed that AVQ would clearly outperform schemes like RED and VQ, since it combines both the arrival rate and queuing status into its load signal. However, the experiments showed that this assumption is only partially true.

From a practical and deployment point of view, the system offers many interesting features. It seems feasible to manage resources of a DiffServ-based service class with RED marking, which are both features that should be available in commercial routers already or soon. Multiple instances of the admission control system can be used to control multiple such classes. Further, the system can be employed for individual links or paths of a network, such that there is no need to upgrade a network at once. Thereby, there is a clear deployment path from today's technology to future, more sophisticated configurations of such a system.

Acknowledgements

We would like to acknowledge the cooperation and support of all colleagues of the M3I project team from Athens University of Economics and Business, BTExact Ipswich, ETH Zürich, FTW Vienna, HP Labs Bristol, Telenor Research Oslo and TU Darmstadt.

References

[1] R. Braden, D. Clark, and S. Shenker. RFC 1633 - Integrated Services in the Internet Architecture: An Overview. Informational RFC, June 1994.
[2] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. RFC 2475 - An Architecture for Differentiated Services. Experimental RFC, December 1998.
[3] K. Ramakrishnan and S. Floyd. RFC 2481 - A Proposal to add Explicit Congestion Notification (ECN) to IP. Experimental RFC, January 1999.
[4] M. Allman, V. Paxson, and W. R. Stevens. RFC 2581 - TCP Congestion Control. Standards Track RFC, April 1999.
[5] J. Widmer, M. Mauve, and R. Denda. A Survey on TCP-Friendly Congestion Control. *IEEE Network*, 15(3):28–37, 2000.
[6] V. Firoiu, J.-Y. L. Boudec, D. Towsley, and Z.-L. Zhang. Advances in Internet Quality of Service. Technical Report DSC200149, EPFL-DI-ICA, October 2001.
[7] S. Shenker. Fundamental Design Issues for the Future Internet. *IEEE Journal on Selected Areas in Communications*, 13(7):1176–1188, September 1995.
[8] S. Floyd and V. Jacobson. Random Early Detection Gateways for Congestion Avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, August 1993.
[9] R. Gibbens and F. Kelly. Distributed Connection Acceptance Control for a Connectionless Network. In *Proceedings of 16th International Teletraffic Congress - ITC 16, Edinburgh, Scotland*, 1999.
[10] S. Kunniyur and R. Srikant. Analysis and Design of an Adaptive Virtual Queue Algorithm for Active Queue Management. *ACM Computer Communication Review*, 31(4):123–134, October 2001. Proceedings of SIGCOMM'2001 Conference.
[11] C. Hollot, V. Misra, D. Towsley, and W.-B. Gong. A Control Theoretic Analysis of RED. In *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'2001)*, pages 1510–1519. IEEE, April 2001.
[12] S. Athuraliya, V. Li, S. Low, and Q. Yin. REM: Active Queue Management. In *Proceedings of the 17th International Teletraffic Congress - ITC 17, Salvador da Bahia, Brazil*, 2001.
[13] V. Siris, C. Courcoubetis, and G. Margetis. Service differentiation in ECN networks using weighted window-based congestion control. In *Proceedings of Quality of Future Internet Services Workshop 2001, Coimbra, Portugal*, pages 190–206. Springer LNCS, September 2001.
[14] P. Öchslin and J. Crowcroft. Weighted Proportionally Fair Differentiated Service TCP. *ACM Computer Communication Review*, 28(3):53–69, July 1998.
[15] R. Yang and S. Lam. General AIMD Congestion Control. In *Proceedings 8th International Conference on Network Protocols (ICNP'00)*, pages 187–198. IEEE, November 2000.
[16] S. Jamin, P. Danzig, S. Shenker, and L. Zhang. A Measurement-based Admission Control Algorithm for Integrated Services Packet Networks. *IEEE/ACM Transactions on Networking*, 5(1):56–70, August 1997.
[17] R. Gibbens, F. Kelly, and P. Key. A Decision-Theoretic Approach to Call Admission Control in ATM Networks. *IEEE Journal on Selected Areas in Communications*, 13:1102–1114, 1995.
[18] R. Gibbens and F. Kelly. Measurement-based Connection Admission Control. In *Proceedings of the 15th International Teletraffic Congress - ITC 15, Washington, DC, USA*, pages 879–888, 1997.

[19] S. Crosby, I. Leslie, B. McGurk, J. Lewis, R. Russel, and F. Toomey. Statistical Properties of a Near-Optimal CAC Algorithm. In *Proceedings IEEE ATM'97*, pages 103–112. IEEE, June 1997.

[20] L. Breslau, S. Jamin, and S. Shenker. Comments on the Performance of Measurement-Based Admission Control Algorithms. In *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'2000)*, pages 1233–1242. IEEE, March 2000.

[21] F. Kelly, A. Maulloo, and D. Tan. Rate Control in Communication Networks: Shadow Prices, Proportional Fairness and Stability. *Journal of the Operational Research Society*, 49:237–252, 1998.

[22] R. Gibbens and F. Kelly. Resource Pricing and the Evolution of Congestion Control. *Automatica*, 35:1969–1985, 1999.

[23] F. Kelly. Models for a self-managed Internet. In *Philosophical Transactions of the Royal Society A358*, pages 2335–2348, 2000.

[24] R. Johari and D. Tan. End-to-End Congestion Control for the Internet: Delays and Stability. *IEEE/ACM Transactions on Networking*, 9(6):818–832, December 2001.

[25] L. Massoulie. Stability of Distributed Congestion Control with Heterogeneous Feedback Delays, 2000. Microsoft Research Technical Report, 2000-11.

[26] F. Kelly, P. Key, and S. Zachary. Distributed Admission Control. *IEEE Journal on Selected Areas in Communications*, 18(12):2617–2628, December 2000.

[27] V. Elek, G. Karlsson, and R. Rönngren. Admission Control Based on End-to-End Measurements. In *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'2000)*, pages 623–630. IEEE, March 2000.

[28] G. Bianchi, A. Capone, and C. Petrioli. Throughput Analysis of End-to-End Measurement-Based Admission Control in IP. In *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'2000)*, pages 1461–1470. IEEE, March 2000.

[29] L. Breslau, E. Knightly, S. Shenker, I. Stoica, and H. Zhang. Endpoint Admission Control: Architectural Issues and Performance. *ACM Computer Communication Review*, 30(4):57–69, October 2000. Proceedings of SIGCOMM'2000 Conference.

[30] T. Kelly. An ECN Probe-Based Connection Acceptance Control. *ACM Computer Communication Review*, 31(3):14–25, July 2001.

[31] L. Westberg, M. Jacobsson, G. Karagiannis, S. Oosthoek, D. Partain, V. Rexhepi, and R. Szabo. Resource Management in DiffServ Framework. Internet Draft, February 2002. Work in progress.

[32] C. Cetinkaya and E. Knightly. Egress Admission Control. In *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'2000)*, pages 1471–1480. IEEE, March 2000.

[33] K. Kumar, A. Ananda, and L. Jacob. Using Edge-to-Edge Feedback Control to Make Assured Service More Assured in DiffServ Networks. In *Proceedings of IEEE Conference on Local Computer Networks (LCN 2001)*, pages 160–167, November 2001.

[34] B. Nandy, J. Ethridge, A. Lakas, and A. Chapman. Aggregate Flow Control: Improving Assurances for Differentiated Services Network. In *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'2001)*, pages 1340–1349. IEEE, April 2001.

[35] H. Kung and S. Wang. TCP Trunking: Design, Implementation and Performance. In *Proceedings 7th International Conference on Network Protocols (ICNP'99)*, pages 222–231. IEEE, October 1999.

[36] I. Stoica and H. Zhang. LIRA: An Approach for Service Differentiation in the Internet. In *Proceedings of the 8th Workshop of Network and Operating System Support for Digital Audio and Video (NOSSDAV'98)*, pages 115–128. ACM, July 1998.

[37] M. Karsten. KOM RSVP Engine, May 2002. http://www.kom.e-technik.tu-darmstadt.de/rsvp/.

[38] K. Cho. *The Design and Implementation of the ALTQ Traffic Management System*. PhD thesis, Keio University, Japan, January 2001. Software at http://www.csl.sony.co.jp/person/kjc/programs.html.

[39] L. Rizzo. Device Polling support for FreeBSD, December 2001. http://info.iet.unipi.it/ luigi/polling/.

[40] S. Floyd. RED: Discussions of Setting Parameters, November 1997. http://www.icir.org/floyd/REDparameters.txt.