Market-Based Resource Allocation for Packet-Switched Networks

Martin Karsten[1*] and Jens Schmitt[2]

1: School of Computer Science
University of Waterloo
Waterloo, ON, N2L 3G1, Canada
www.cs.uwaterloo.ca
mkarsten@bbcr.uwaterloo.ca

2: Multimedia Communications Lab
Darmstadt University of Technology
64287 Darmstadt, Germany
www.kom.e-technik.tu-darmstadt.de
schmitt@kom.tu-darmstadt.de

Abstract

Market forces are the most effective mechanism to fairly and efficiently allocate resources among competing service requests. However, for distributed resources, the implementation of a coherent market mechanism can be complex and costly. In this paper, we present the design and prototype implementation of a distributed resource allocation system that allows to apply flow-based market mechanisms to a network domain. The system design guarantees a constant execution complexity and an extremely simple layout of internal nodes. All relevant intelligence is located in the edge systems. We explain how the system can be used to realize various types of market mechanisms and show the potential for efficient implementation by means of lab experiments using the system prototype. This work is based on earlier conceptual proposals and theoretical analysis. It is focused on the system design and implementation aspects, as well as on questions of detail, which are usually ignored by existing theory work.

1 Introduction

The current Internet consists of a large number of forwarding resources (links and routers) and offers a best-effort transmission service to end systems. Despite many research proposals, there is only limited explicit resource allocation deployed inside the networks forming the Internet. Basically, the networks rely on cooperation between end systems to fairly share resources by using TCP for data transmission and consequently employing the inherent congestion control algorithms implemented in TCP. However, there are several limitations in this basic model that currently restrict the usage of packet-switched networks for different applications, such as business-critical, real-time or multimedia applications that impose stringent performance requirements on the underlying infrastructure. First, the basic TCP-based model cannot offer quantifiable a-priori guarantees of traffic performance and second, it does not allow to control the resource allocation and differentiate between service requests beyond treating every flow as equal.

The usual solution to this problem is given by QoS technologies, which employ some kind of differentiated, *proactive* resource allocation. Numerous proposals have been published throughout recent years and some of the more prominent examples are briefly presented and discussed in Section 2. In this work, we investigate a solution that uses a radically different principle, much more in line with the original TCP-based mechanisms, to provide for predictable traffic performance and service differentiation. The system essentially allocates transmission resources by means of a distributed resource market. Based on earlier conceptual work in this direction, we study the detailed challenges for the overall system design by means of an experimental software prototype. Thereby, this work also complements existing theoretical analysis and simulation results that already indicate many interesting characteristics of such a system.

The paper is organized as follows. In the next section, we review previous and related work. Based on this, the details of the problem statement are explained in Section 3 and solutions are provided in Section 4 by presenting the design and implementation of an effective and resource-efficient market-based resource allocation system. Section 5 contains an overview about the technical experiments that have been carried out with this prototype, so far and Section 6 summarizes the paper and discusses topics for future research.

2 Related Work

The discussion of related work is structured according to the different aspects of resource allocation in packet-switched networks. We begin by discussing prominent QoS proposals and then specifically look at those employing edge-based admission control. Afterwards, some background information on

[*] The work reported here has been carried out while the author worked at the Multimedia Communications Lab at Darmstadt University of Technology. The work is partially sponsored by the EU 5th Framework, IST Programme, Project 11429 “M3I”

distributed resource markets is given before the specific topic of pricing and charging for network services is discussed.

2.1 QoS Proposals for the Internet

The last fifteen years have seen a tremendous amount of research on how to provide network QoS for large-scale internetworks as the Internet (see [1] for a recent overview). In particular, this research also manifested itself in standardization efforts, particularly within the IETF (Internet Engineering Task Force). The first comprehensive architecture being proposed was the so-called IntServ (Integrated Services) architecture [2]. IntServ is built on a rather traditional style of proactively reserving resources per session, thus basing upon the call paradigm known from connection-oriented telecommunication networks. As a signalling protocol, the Resource reSerVation Protocol (RSVP) [3] was proposed to allocate resources according to the IntServ service model [4, 5] on a per-flow and per-hop basis, in its first version. Due to doubts about the scalability of the IntServ architecture because of the per-flow operation, another architecture called DiffServ (Differentiated Services) was put forward [6]. DiffServ explicitly excluded per-flow treatment within the core of DiffServ domains, but operates on a small number of behaviour aggregates by giving them differentiated forwarding behaviour in interior network nodes and controls entrance to DiffServ domains by appropriately conditioning traffic at its ingress. While the scalability characteristics of DiffServ are certainly better than for classical IntServ, it needs to be mentioned that the power of a DiffServ domain to give strict QoS guarantees heavily depends on complex strategies to overprovision the network correctly [7].

Both IntServ and DiffServ have seen numerous enhancements from their basic architecture to alleviate some of their respective problems (see again [1] for a very up to date overview). However, both of them can be considered proactive approaches where the current network state is not taken into account, at least on small (say round-trip time) timescales. However, as is discussed and to some degree shown practically in this paper integrating feedback and interpreting it as economic signal may be a more efficient, yet working alternative. In particular, the open issues with respect to integrating charging mechanisms into technical proposals like IntServ and DiffServ could also be circumvented.

2.2 Edge-based Admission Control

Our architectural choice is for edge-based admission control, that is, 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 technical 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 [8], Kelly et al. describe a system similar in concept to what they propose in [9]. The difference between these two is that in [8] an admission control gateway does the probing for the end systems whereas in [9] 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 [8] 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 insight from [8]. From a technical point of view, our system is simpler, because it does not require explicit probing.

A DiffServ framework for edge-based admission control is described in [10]. 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 [10].

In [11], 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 minimal load feedback as provided in our system proposal, the admission control procedure can be made much more simple and robust.

All related proposals presented here are similar to our work, in that they carry out admission control at system edges. Nevertheless, as discussed, there are a number of differences in the design details. Only the proposal(s) in [8,9] consider the network domain as a distributed resource market and none of the proposals combines technical admission control with appropriate dynamic pricing.

2.3 Distributed Resource Markets

In a seminal paper, [12] brought the economic elegance for congested resources to the attention of the network research community, in their so-called "Smart Market" approach. However, being a centralized solution to the inherently distributed problem of allocating resources to users in the Internet, it could only be viewed as a conceptual proposal to show the theoretically achievable benefits of such a scheme if it could be implemented in the Internet. In particular, the NP-hard nature of multi-dimensional and combinatorial auctions [13] was neglected in the original proposal. In fact, our proposal can be viewed as a distributed approximate solution to this problem with a very low complexity.

2.4 Pricing and Charging for Network Services

Earlier work has studied the problem of charging for network services in the context of traditional, proactive network QoS systems. For example, [14] presents a system to enhance RSVP signalling for intra-domain cost allocation and inter-domain charging signalling. On the other hand, the work in [15] reports about building an auction model for signalled resource reservation and studies the transactional problems associated with multiple resources along a path. In [16], the concept of a *Guaranteed Stream Provider* (GSP) is introduced, which forms the basic theoretical blueprint for the system presented in this paper.

3 Problem Statement and Overview

As illustrated in the previous section, existing work indicates that it is well feasible to consider a system of network forwarding resources as distributed market and achieve stable and predictable rate allocations by means of market forces. The major goal of this work is to investigate how these basic findings can be used for actual system design and to provide solutions for problems of detail. In this context, we focus on long-lived flows with rather stable transmission rate requirements. There are mainly two types of applications that we have in mind. On one hand, multimedia communication services often have only limited capabilities to adapt to changing transmission performance. On the other hand, aggregated traffic, for example in a VPN environment, often has strict minimum transmission requirements, but benefits from additional transmission rate, if available. Similar to earlier proposals, the system uses binary packet marks to encode an aggregated load signal into the packet stream, which is then interpreted at system edges.

For several reasons, it seems impractical to realize a global distributed resource market across the Internet. First, this would require a global, strictly uniform understanding of the meaning and implications of packet marking algorithms. This level of algorithmic homogeneity between networks is very unlikely in the Internet. Second, faulty end systems might corrupt the stability of the resource allocation and compromise the robustness of the system. In this case, it is very hard to handle the resulting liability questions. Third, only the receiver of a packet stream encounters marked packets while only the sender is eventually capable of adjusting the sending rate. This creates a severe trust and security problem, because either receivers must only receive packets from trusted senders, or the sender must be held responsible for the marking rate observed at the other end of the transmission path. For these reasons, the system presented here operates on the scope of a network domain and provides a traditional signalling interface to clients and adjacent networks.

Any kind of proper market mechanism achieves stability by converging towards an equilibrium of supply and demand. Nevertheless, additional regulation is likely necessary to ensure system stability in times of excessive or unstable demand and to fulfil regulatory requirements for general access to basic services. Again, this problem can be solved by carrying out technical admission control and traffic regulation at system edges. However, the location and particular details of such components then have to be considered.

To fully utilize available resources, it might be useful to offer load-adaptive traffic regulation, instead of always regulating incoming traffic according to a fixed sending rate. Nevertheless, the resulting excess traffic must not prohibit resource allocation to new flows. Any kind of reactive resource allocation system can only observe the load that is caused by actual traffic. If senders do not fully utilize their rate allocation, there is a potential for overbooking, since unused but already allocated rate is not accounted for when accepting new requests.

A signalling protocol is needed to communicate with clients of the system, as well as between edge gateways. It is highly beneficial to align signalling procedures to existing signalling protocols to simplify inter-operation with end-to-end signalling and to enable the reuse of available technology.

Finally, there are subtle problems associated with setting an appropriate price. Assuming a fixed charge per marked packet introduces two practical problems. First, when each marked packet is associated with a fixed price, the total price for any given transmission rate is finite, because forwarding capacity is finite. If the price elasticity of users exceeds this finite price, the network faces uncontrolled and excessive demand. Second, depending on the details of the packet marking scheme in place, it is not always clear whether the overall marking probability reflects the packet length and whether that is desirable (which in turn depends on the routers' processing cost compared to the links' bandwidth cost). In any case, a fixed price per mark might open the possibility for arbitrage by choosing appropriate packet lengths.

4 System Design and Implementation

Many aspects of the implementation design, for example the choice of the basic signalling protocol, are not governed by fundamental requirements, but rather chosen according to their practicality for implementation, experimental investigation, and later deployment. The system requires two bits in the IP packet header, such as the ECN bits [17]. The terminology of ECN is adopted for the presentation 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 [18].

4.1 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 and mark packets depending on the current load 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*. The price setting and distribution functions are logically decoupled from the rest of the network control system. They can be implemented as separate components or collocated with load-control gateways. A price distribution protocol and a price setting framework have been specified and implemented [19], but are not discussed here in detail. Figure 1 presents the different roles of system components along the transmission path. Note that the admission control decision can be carried out at either the ingress or the egress gateways. In both cases it is necessary to exchange signalling information between the egress and ingress gateway.

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. 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 also can be employed in a multi-path routing scenario by considering the endpoints of each routing path as virtual peers.

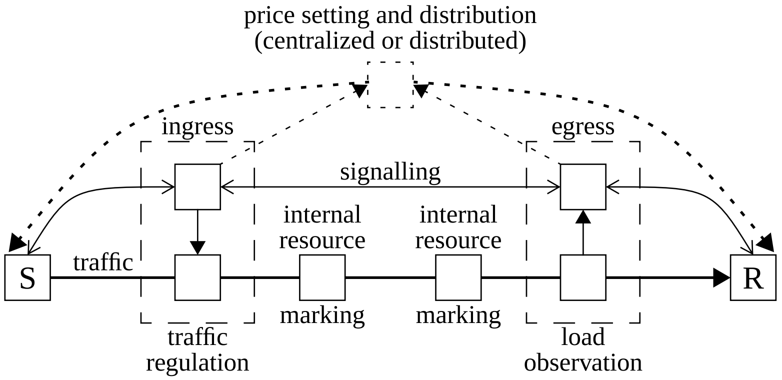

Figure 1: System Overview

4.2 Packet Marking

We distinguish between two basic types of marking algorithms according to the different load signal that is encoded in the packet stream. A threshold-based marking (TBM) algorithm marks all packets when the forwarding rate at a resource exceeds a certain threshold. Otherwise, no packets are marked. For edge systems, the packet stream carries a binary signal, which is set to one when any resource along the path is loaded beyond its local threshold. Note that on the flow time-scale that is being considered for this work, marking algorithms such as AVQ [20] and VQ [8] essentially operate like threshold-based marking. This has been experimentally verified in [21]. The other type of algorithm is load-based marking (LBM) [22], which marks packets with a probability that depends on the relative forwarding load. Since only a continuous marking signal provides enough information to derive load-based prices, the system uses linear LBM for this purpose.

Depending on the overall scenario, packet marking may need to be combined with a differentiated dropping algorithm to discriminate between packets from registered and unregistered flows. This is described and investigated in [21], but not described here in further detail.

4.3 Admission Control

The egress gateway observes the relevant information for load estimation. 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. This information 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 new request can be 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

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

with $\beta \in [0,1]$ determining the relative influence of booked but unused capacity. A small value for β denotes an optimistic system configuration in which the potential overbooking is largely ignored and a large value for β 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 β 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).

4.4 Pricing and Charging

When employing load observations and admission control per path of the network, a dedicated time-based price can be calculated from the aggregated load observation per path. A number of pricing schemes can be built using very little information. As presented in the previous section, the number of packets, marks, bytes and the measurement duration is available. Some possible pricing schemes are presented below, followed by a discussion about service charging.

4.4.1 Static Mark-based Pricing

When setting a fixed price per mark α, the load-dependent price per transmission rate can be calculated as

$$P(r) = \frac{\alpha \times m \times r}{l \times t} \qquad (2)$$

with r being the transmission rate, l being the average packet length, and m being the number of marks during the last measurement period t. This however requires the notion of an average packet length, which is not always obvious and further, because of the fixed price per mark, might not be sufficient to always protect the network from excessive demand.

4.4.2 Dynamic Mark-based Pricing

To handle increasing demand in (2), the price per mark can be adjusted iteratively by increasing it whenever the demand reaches a critical threshold. Alternatively, a load-based price per mark α' can be calculated as

$$\alpha' = \frac{\alpha \times p}{p - m} \text{ for } p > m \tag{3}$$

with m and α as above, and p being the number of packets during the last measurement period. Thereby, in theory the price per mark continuously goes to infinity with increasing network load, while in reality it is discrete because of the discrete number of packets p and number of marks m. In reality, it is also bounded by the fixed transmission capacity of the network. Of course, if all packets would be marked, that is, if $p = m$, (3) cannot be used any more. However, this situation is effectively avoided by the technical admission control described in Section 4.3.

4.4.3 Rate-based Pricing

It is also possible to set the price corresponding to both the packet rate r_p and the byte rate r_b of a request. The pricing function then looks like

$$P(r_p, r_b) = \frac{\gamma' \times r_b \times m}{b} + \frac{\alpha' \times r_p \times m}{p} \tag{4}$$

with m, p and α' as above, γ' being analogous to α', and b being the number of bytes during the last measurement period. Since this price has components for the packet rate and the byte rate of service requests, it can effectively prohibit the potential arbitrage discussed in Section 3. Also, since the pricing function goes to infinity when the network load increases towards a critical value, it can be used to protect the network from overload, except if a faulty client ignores the price.

Since the other two pricing functions can be realized by counting packets respectively marks, they can be implemented without load control gateways. The price function (4), however, also takes into account the transmission rate in bytes per time unit. This information is available at egress gateways without extra execution cost.

4.4.4 Service Charging

The most recent price information is distributed via the separate price distribution protocol [19]. The system is built on the assumption that an arriving service request indicates the client’s consent with the recently published price. A service auction can be implemented, for example by starting with a very high price and reducing it over time. The higher each client’s perceived value from completing the service, the earlier it accepts the published price. However, separate price and service request channels introduce a problem. Since the price distribution protocol can hardly be made fully reliable, it would be necessary to associate an incoming service request with the most recent price by means of the request signalling protocol. This problem is postponed to future work.

4.5 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

$$t_{new} = \min(t_{old} + \tau \times r \times s, d) \text{ with } s = \frac{\Lambda - \varepsilon}{l} \tag{5}$$

for admission control threshold Λ, path load l, and a small $\varepsilon > 0$.

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 requests, the admission control threshold Λ minus a small safety margin ε is divided through the current relative load estimation 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 operation of the linear LBM 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. In other words, the maximum amount of marks that a flow is responsible for, is implicitly bounded by the initial service request, which conforms to the goals of the pricing schemes presented in Section 4.4. Fairness between multiple flows is obviously given, because each flow's service rate allocation is proportional to its requested rate.

The scaling factor *s* is never set below 1, such that any flow can always exploit its negotiated rate. Therefore, it is not necessary to use the adapted estimated load from (1) for adaptive traffic regulation, because booked traffic automatically displaces excess traffic by increasing the relative path load.

4.6 Signalling

RSVP [23] is an IETF QoS signalling protocol that is well-suited to incorporate appropriate extensions to implement the control path of the system for admission control and market-based resource allocation. RSVP is a receiver-initiated setup protocol for simplex flows and each RSVP instance on a router administers the respective outgoing link in the direction of the flow. In the context of this market-based resource allocation system, the egress gateway thus reports load information towards the ingress and the ingress carries out admission control. 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 (1) and, more importantly, to relate the load situation to the actual usage rate for pricing function (4) introduced in Section 4.4.3. 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 hop count is needed to compute the estimated average load, which is not discussed here in detail.

All information included in the load report, except the hop count, is gathered from a kernel-level 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 adaptive traffic regulation as described in Section 4.5. 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.

4.7 Prototype Implementation

The signalling, admission control and load reporting functionality is implemented in the framework of the publicly available KOM RSVP implementation [24]. The data path modules are implemented in the ALTQ software framework, which is publicly available, as well [25]. The price setting and communication modules are implemented in Java. Load control gateways communicate with the price setting components through the COPS protocol [26]. End systems run an integrated QoS manager for price and QoS signalling, which is also implemented in Java. The system has been developed and tested in a mixed configuration of FreeBSD and Linux. The extensions presented here will also be published as open-source software. To our knowledge, no such comprehensive market-based resource allocation system exists as a real system prototype.

5 Evaluation

The system is evaluated in two ways. The first challenge is to assess the economic efficiency of the market mechanisms. This question can be answered only by mathematical modelling using quite strict assumptions or, more realistically, by large-scale trials taking into account the real behaviour of real clients. Therefore, we only discuss the system's properties with respect to offering dynamic pricing for guaranteed transmission rates. The technical challenges, however, can be assessed to a large extent by lab experiments and/or simulations. This includes the reaction times of marking and edge nodes to a dynamic network load situation and the resulting convergence behaviour of the whole system, as well as a verification of the signalling procedures to communicate information between the various entities of the system. Therefore, we present quantitative experimental results to show the feedback and convergence behaviour of the system.

5.1 Conceptual Evaluation

The proposed system allows for both market-based resource allocation and technical admission control. Thereby, it is possible to efficiently allocate resources to the most demanding requests while the system is not fully loaded. If for any reason the demand suddenly increases, the performance assurances of accepted flows are protected through the technical admission control function until the market mechanisms can adapt to the changing demand. For example, it might not be suitable to change the pricing structure of already accepted flows. Therefore, the delay between an increase in demand and the effectiveness of price adaptation might cover a significant time period.

Depending on the marking algorithm used, the system can offer different types of fairness among competing requests. In the context of long-lived, fixed-rate flows, threshold-based marking results in fairness with regard to bottleneck resources, because those dominate the path marking rate. Load-based marking takes into account all resources traversed by a flow, whether being the bottleneck or not. In any case, respective market mechanisms like auctions can be built based on the underlying fairness criterion.

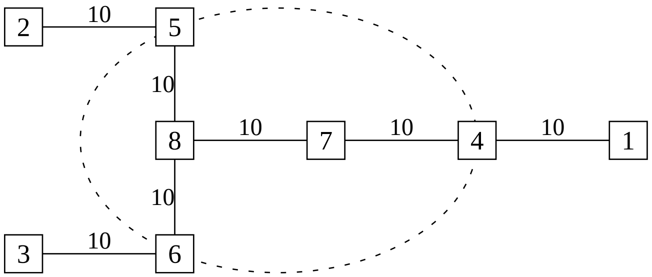

Figure 2: Experiment Topology

While load-based marking can be regarded to produce a more detailed estimation of an individual flow's resource usage in relation to the system state, there is one important drawback associated with it. Whenever flows are subject to load-based marking at multiple nodes, the resulting marking rate is higher than each node's marking rate, because of the combinatorial properties of the random experiment. Therefore, plain load-based marking might lead to a sub-optimal utilization of network resources. To this end, we are studying to use exponential load-based marking in combination with suitable admission control rules or a combination of load-based and threshold-based marking to overcome this restriction [21].

5.2 Technical Evaluation

All experiments are carried out in the topology shown in Figure 2. 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 [27]. 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. We show the results of two experiments to illustrate the feedback behaviour of the system.

In the first experiment, VoIP-like sessions from node 2 to 1 and from node 3 to 1 are generated with a deterministic inter-arrival time of 0.5 seconds and a duration of 50 seconds. In total, the demand exceeds the available transmission capacity. The behaviour of the system is illustrated in Figure 3.

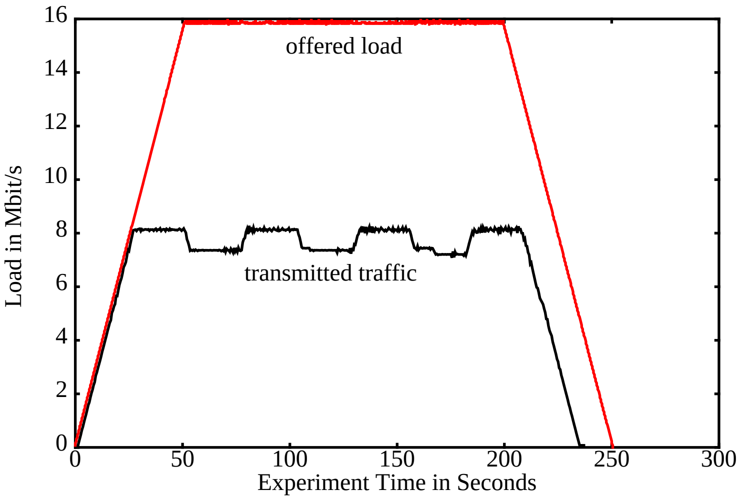

Figure 3: Admission Control

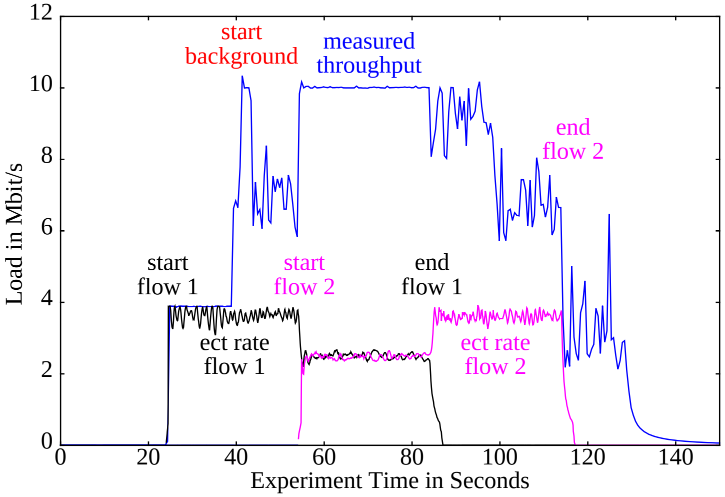

Figure 4: Load-Adaptive Traffic Regulation

The periodicity of session acceptance is due to the session arrival process and the reactive nature of the system becomes apparent by observing this periodicity. Essentially, the slope and length of the increase and decrease segments of the accepted load curve represent the feedback delay in the system. The experiment clearly shows that the system is capable to effectively carry out admission control and that the reaction delay is in the order of a few seconds. It thereby also shows as a proof of concept that the proposed system can indeed be realized. Additional technical experiments studying more aggressive and randomized demand, as well as the influence of background traffic on the system, are reported in [21]. Those results also support the conclusion that the system is capable to carry out precise admission control by reacting quickly to changes of the network load. Consequently, it can be expected that the inherent price adaptation that results from an increased marking rate is propagated fast enough to enable efficient resource allocation.

To verify the operation of load-adaptive traffic regulation, another experiment has been carried out with a small number of bigger flows to study the system's behaviour. Two reserved flows and some background traffic are started with a certain time interval in between to observe the reaction of the system to the changes in demand. Flows 1 and 2 are the reserved flows, but inject more traffic into the system than signalled. The result is depicted in Figure 4 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 currently 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.

Many further experiments have been carried out to verify the system's operation and admission control behaviour in combination with other marking algorithms. A number of those experiments are reported in [21].

6 Summary and Future Work

We presented the system design and a prototype implementation of an admission control system to efficiently allocate the forwarding resources of a network domain by means of market mechanisms. Based on earlier proposals, this work is focused on the design details of packet marking, admission control, pricing and adaptive traffic regulation. In particular, the signalling extensions to implement such a system in the context of RSVP signalling are specified and verified. The proposed system differs from previous work in terms of its design details, which are partially influenced by real-world requirements, such as interaction with an existing signalling protocol. To our knowledge, no such comprehensive prototype system has been built, so far. The overall system design and the underlying concepts are technically evaluated by means of lab experiments. Thereby, the validity of earlier theoretical proposals is backed up.

Clearly, future work is required to further study the properties of market-based reactive resource allocation. While our results can be regarded as promising indication of the real-world feasibility, additional details need to be considered and potentially require further improvements of the system design. For example, the adaptive traffic regulation part of the system can likely be improved by appropriately adjusting internal system parameters. Additionally, the interaction of traffic from end systems using flow-control, such as TCP, with both the admission control and the traffic regulation part of the proposed system is an important research area to investigate whether multiple types of traffic flows can accurately be supported by a single-class forwarding system or whether multiple forwarding are necessary, for example by means of DiffServ-based differentiated scheduling.

Acknowledgements

The authors would like to acknowledge the basic concepts and valuable input contributed by all colleagues from the M3I project team. Please see the project's web page for further details at

http://www.m3i.org

References

[1] 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.

[2] R. Braden, D. Clark, and S. Shenker. Integrated Services in the Internet Architecture: an Overview. Informational RFC 1633, June 1994.

[3] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource Reservation Protocol (RSVP) - Version 1 Functional Specification. Proposed Standard RFC 2205, September 1997.

[4] S. Shenker, C. Partridge, and R. Guerin. Specification of Guaranteed Quality of Service. Proposed Standard RFC 2212, September 1997.

[5] J. Wroclawski. Specification of the Controlled Load Network Element Service. Proposed Standard RFC 2211, September 1997.

[6] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. RFC 2475 - An Architecture for Differentiated Services. Experimental RFC, December 1998.

[7] A. Charny and J. Y. L. Boudec. Delay Bounds in a Network with Aggregate Scheduling. In *Proceedings of Quality of future Internet Services Workshop (QofIS 2000), Berlin, Germany*, pages 105–116. Springer LNCS, September 2000. ISBN 3-540-41076-7.

[8] 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.

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

[10] 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.

[11] 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.

[12] J. K. MacKie-Mason and H. R. Varian. Pricing the Internet. In *"Public Access to the Internet"*. JFK School of Government, Harvard University, May 1993. available from http://www.spp.umich.edu/spp/papers/jmm/Pricing_the_Internet.ps.Z.

[13] M. H. Rothkopf, A. Pekec, and R. M. Harstad. Computationally Manageable Combinational Auctions. *Management Science*, 44(8), August 1998.

[14] M. Karsten, J. Schmitt, L. Wolf, and R. Steinmetz. An Embedded

Charging Approach for RSVP. In *Proceedings of the Sixth IEEE/IFIP International Workshop on Quality of Service (IWQoS'98), Napa, USA*, pages 91–100. IEEE/IFIP, May 1998. ISBN 0-7803-4482-0.

[15] P. Reichl, G. Fankhauser, and B. Stiller. Auction Models for Multi-Provider Internet Connections. In *Proceedings of 10. GI/ITG Fachtagung MMB'99, Trier, Germany*. VDE-Verlag, September 1999.

[16] R. Andreassen, editor. *M3I Deliverable 1 - Requirements Specification Reference Model*. EU 5th Framework, Program IST, Project 11429 (M3I), June 2000. Available from http://www.m3i.org/results/m3idel01v7_1.pdf.

[17] K. Ramakrishnan and S. Floyd. RFC 2481 - A Proposal to add Explicit Congestion Notification (ECN) to IP. Experimental RFC, January 1999.

[18] K. Nichols, S. Blake, F. Baker, and D. Black. Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. Proposed Standard RFC 2474, December 1998.

[19] O. Heckmann, V. Darlagiannis, M. Karsten, and R. Steinmetz. A Price Communication Protocol for a Multi-Service Internet. In *Informatik 2001 - Wirtschaft und Wissenschaft in der Network Economy - Visionen und Wirklichkeit (GI/OCG 2001)*, pages 112–119, September 2001.

[20] 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.

[21] M. Karsten and J. Schmitt. Admission Control based on Packet Marking and Feedback Signalling - Mechanisms, Implementation and Experiments. Technical Report TR-KOM-2002-03, KOM, TU Darmstadt, May 2002. Available at http://www.kom.e-technik.tu-darmstadt.de/publications/abstracts/KS02-1.html. Currently under submission.

[22] 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.

[23] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. RFC 2205 - Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification. Standards Track RFC, September 1997.

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

[25] 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.

[26] D. Durham, J. Boyle, R. Cohen, S. Herzog, R. Rajan, and A. Sastry. RFC 2748 - The COPS (Common Open Policy Service) Protocol. Standards Track RFC, January 2000.

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