[ABDHSS03] *Bob Briscoe, Vasilios Darlagiannis, Oliver Heckmann, Oliver-Huw, Vasilios Siris, Burkhard Stiller, and David Songhurst:* **A Market Managed Multi-Service Internet;** Computer Communications, 26(4), March 2003,:S. 405-415.

ELSEVIER

Computer Communications 26 (2003) 404–414

computer communications

www.elsevier.com/locate/comcom

A market managed multi-service Internet (M3I)☆

Bob Briscoe[a,*,1], Vasilios Darlagiannis[b], Oliver Heckman[b], Huw Oliver[c], Vasilios Siris[d], David Songhurst[a], Burkhard Stiller[e]

[a] *BT Research, B54/130, Adastral Park, Ipswich IP5 3RE, UK*
[b] *Darmstadt University of Technology, Darmstadt, Germany*
[c] *HP Labs, Bristol, UK*
[d] *Institute of Computer Science, FORTH, Heraklion, Crete, Greece*
[e] *Computer Engineering and Networks Laboratory TIK, Swiss Federal Institute of Technology, ETH Zurich, Switzerland*

Received 12 May 2002; accepted 14 May 2002

Abstract

In this paper, we describe our approach to managing quality of service (QoS) using pricing. We show how it is possible to synthesise network QoS in the end-systems along the lines of the end to end design principle, as one of many possible business models. We have: (i) developed an architecture for market management; (ii) invented new business models to test and demonstrate its flexibility; (iii) implemented generic mechanisms that not only enable these models but also many others; (iv) modelled selected features of the resulting systems and markets and (v) conducted experiments on users to assess acceptability and the feasibility of the overall approach. Each of these aspects is outlined in brief overview, with numerous references to more detailed work.

Keywords: Data communication; Networks; Internet; Quality of service; Congestion control; Charging; Pricing; Policy; Accounting; Admission control; Architecture

1. Introduction

Lack of flexibility to offer charging schemes beyond flat rate has driven many Internet service providers (ISP) into bankruptcy. The early focus was on speedy technical feasibility of the schemes applied and not on their economic viability. Arguments for a simple technical solution *are* generally valid, but the primary focus for an ISP must be to use business models that maximise net returns. The common misconception that billing accounts for 50% of telephony's costs hasn't helped—true for running costs, but it drops down to 4–6% when depreciation of sunk costs is included. Instead, a proven strategy is to differentiate services, ranging from transport to content services, and offer innovative tariffs to provide an incentive for customers to optimise their choices as the market develops.

In this paper, we describe our approach to managing quality of service (QoS) using pricing. At the same time as managing QoS, our approach allows open innovation both for providers through their tariffs and for customers in their use of the network for new applications in novel and perhaps unpredictable ways. Since before it was first articulated in the early 1980s, adherence to the end to end design principle [1] has fostered Internet innovation by keeping the network dumb and moving intelligence to end systems. Our approach even pushes quality control out of the network into the hands of its customers. However, where providers find this too radical and would rather keep direct control, our approach is broad enough to allow them to grasp back control at the network edge, a decision itself under their own policy control.

This paper presents a broad picture of our achievements. We have: (i) developed an architecture for market management; (ii) invented new business models to test and demonstrate its flexibility; (iii) implemented generic mechanisms that not only enable these models but also many others; (iv) modelled selected features of the resulting systems and markets and (v) conducted experiments on users to assess acceptability and the feasibility of the overall

☆ M3I is co-funded by the Fifth Framework of the European Commission IST Programme, project IST-1999-11429.

* Corresponding author.
E-mail address: bob.briscoe@bt.com (B. Briscoe).
1 http://www.labs.bt.com/people/briscorj/

0140-3664/03/$ - see front matter © 2002 Elsevier Science B.V. All rights reserved
PII: S0140-3664(02)00158-5

approach. Each of these aspects is outlined in the sections that follow.

Our goal was not to promote the business models we invented. They *are* commercially novel, but we must stress they are merely examples to stretch our approach, and demonstrate its viability. The true intention was to open the market to many more business models. But flexibility can be used by fools as well as the wise. So perhaps our main contributions are the guidelines for developing business models that give the correct economic incentives both for raising revenue and for controlling network quality.

To this end, Section 1.1 uses the example of Diffserv to illustrate the problems a business model can have, and how it could be improved. We then use the example of admission control to introduce how QoS technology itself can contain an implicit business model, and we introduce how to break the two apart, but still be able to re-synthesise the traditional admission control business model—but by choice, not design.

1.1. New business models on old QoS technologies

The specification of the differentiated services field [2] defines a QoS technology without any associated business model. On the other hand, the 'native' business model of the technology, termed 'Diffserv' [3,4], defines how to go about sizing these differentiated networks. Note that the term Diffserv implies the whole architecture, and is *not* an abbreviation of the general ability to differentiate services.

Two economic factors are at the heart of QoS: supply of network capacity and demand for it. The Diffserv business model focuses on getting the supply side correct—the sizing of each logical network. Demand is much more volatile, and Diffserv includes nothing new to control short term demand.

Instead, Diffserv uses service level agreements (SLAs) to constrain the unpredictability of customers' demands and simultaneously drive the capacity sizing process. One aim was to avoid costly, per-session charging or policing. However, unless each SLA is between a single pair of addresses it is impossible to avoid occasional congestion events as the unpredictable demands of people and computers coincide at flash points in the network, driven by events in the world at large. The SLA either accepts a certain level of such events as part of the deal, or offers refunds when they occur, both of which fail to meet the legitimate demands of customers. More problematic is that SLAs are only relevant for aggregated demand. For mass market customers, demand is sparse and highly unpredictable, making an SLA impractical for both customer and provider. A final, more subtle problem with SLAs is they constrain customers from doing anything novel. Anyone who invents a new application will be caught in a vicious circle where no-one can use the new application because it breaks SLAs, but SLAs never get changed because demand for a broken application will never actually materialise. Thus the SLA business model is not a general solution to QoS, despite solving some short term problems in the corporate world.

The problem is that the flow of economic information is inadequate. We report in brief below (Section 2.4) a more sophisticated business model (CPS) we have proposed and analysed, which retains the simplicity of SLAs but improves the economic information flow. But, we must emphasise that our purpose is not to recommend any particular business model but to show that M3I technology can be used to transform old 'native' business models into ones with better economic properties.

Focusing on the supply side of the QoS problem, Diffserv does nothing to exploit the huge capacity users and computers have for adapting their *demand*. The core of the QoS problem tackled by our M3I work is to solve the fast control problem—to avoid QoS degradation during short term congestion. If we can solve this problem, and adapt whenever the network size is 'wrong', then 'correct' network sizing becomes a non-problem with respect to short-term QoS.

Traditionally, the demand control problem has been solved by connection-oriented admission control. For a pure connectionless network, the equivalent to dropping a proportion of calls is to drop the same proportion of packets. But, for *some* applications, this isn't any use. Specifically where little or no value is derived unless more than some minimum threshold bit rate can get through. So we need admission control for demand side control, but we question current approaches to implementing it. The decision on which flows are admitted is a policy decision. Accepting flows on a first-come, first served basis as Intserv [5] does is just one model. Another (far more economically efficient) model is to accept those most willing to pay. We must not embed a business model choice into the technology of every network. We should provide a substrate on which network businesses can make these choices. This is a new criticism of Intserv, which is usually only criticised for non-scalability [6].

Sections 2.2 and 2.3 describe how a network provider can choose to offer admission control under either of the above business models. Under policy control we synthesise either model at the edge, from the same flowless network technology in the core. The crucial addition to the core is explicit congestion notification (ECN). Again the approach is to improve the flow of economic information outward, rather than only focusing on QoS requests *into* the network, contrary to the end to end principle.

Our first solution uses pricing to encourage self-admission control [7,8]. This is similar to time of day pricing, but uses real-time levels of demand at every congestible resource on every path rather than long term predicted averages and is thus far more optimal in economic terms [9]. Effectively, demand can be ranked by value with the price automatically adjusting to ensure the network is always fully used by the most valuable customers.

Unfortunately, there is strong evidence that customers find dynamic pricing unacceptable, despite our user experiments detecting some interest in it. We believe it is a viable model for the QoS of computer to computer interactions in the future, but admit that it is not necessarily appropriate for interactive use. So rather than the customer synthesising admission control, our second solution synthesises it at the network edge from a dynamically priced wholesale service. Not only do we provide hard admission control guarantees without embedding a business model into the network, but we also solve Intserv's scalability problem.

To summarise, we show that a minimal connectionless service is all that is necessary in the network, and session-oriented business models can be synthesised at the edge—the end to end design principle applied to QoS itself.

1.2. What is M3I?

It is often not immediately apparent where M3I technology sits. This is because M3I is a number of complementary things. M3I is:

- IP network middleware *on* customer and provider systems giving their buying and selling policies real-time control over application and network quality
- middleware *between* providers and customers (who may themselves also be providers) along the value chain to transform between different QoS technologies and pricing schemes
- a framework around the middleware to enable switching between pricing schemes
- an approach to managing network resources using pricing even if hidden from customers.

Section 3 on Engineering covers the first three points, while Section 4 on Modelling reports on the considerable body of work on analysis and simulation of QoS pricing behind the M3I approach. This whole paper is an extremely brief overview of a considerable body of work. References to our more detailed reports are given throughout.

2. Requirements and scenarios

These scenarios look forward to a future Internet that is a multi-service network. We believe network service providers will wish to offer differentiated products (services) to their customers as competition in the Internet services market increases. There are several current proposals for technical mechanisms to provide differentiated services in the Internet. M3I builds middleware over these various mechanisms that flexibly allows service providers to implement their business models for pricing and charging for these services. In order to demonstrate this flexibility, M3I has implemented several different business models over the different technical QoS mechanisms. Each of these business model/technical mechanism pairs is described here. Further information can be found in Ref. [10].

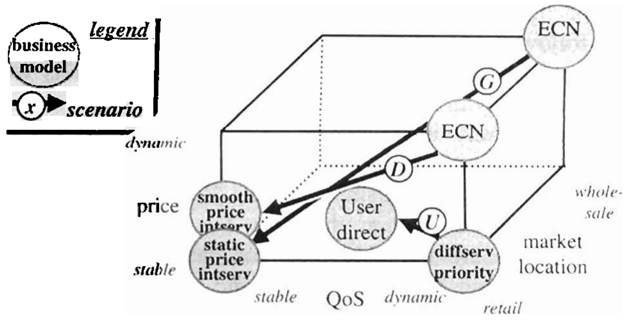

Fig. 1. Scenario dimensions.

Each technical QoS mechanism has traditionally been associated with its own, 'native' business model. The business models implemented in our scenarios can therefore be represented as a transformation from the native model for the QoS technology in question, shown as labelled arrows in Fig. 1. Not only are the scenarios chosen for their commercial feasibility, they are also chosen because they stretch the three dimensions shown in Fig. 1. If we are to avoid embedding session admission control in the connectionless Internet, we have concluded that two of these dimensions—price and QoS stability—are in fundamental tension. More stability in one cannot be achieved without less stability in the other. The third dimension (market location) allows us to relax the tension between the other two. It introduces another link in the value chain between network provider and customer, described in the guaranteed stream provider scenario below.

2.1. User direct scenario

There has been much work on providing different classes of service according to the different needs of different applications. The user direct scenario ('U' in Fig. 1) gives the end user control over quality of service and price, according to his utility for the services. In our basic scheme the user is offered a list of priority levels at which to send his traffic. Traffic sent at a higher level will be sent at a higher priority and at a higher price. The absolute quality of service of each priority level is not guaranteed, but will depend on the current network state. The differences in QoS are relative, may change in real-time and, on that basis, the user may choose to move up or down the levels accordingly, trading relative quality of service for price of service usage. The scheme has parallels with, for example, the airline industry where 'upper class' seats are available at a higher price but where there are no strict guarantees about what better level of service a customer will receive. Customers use their past experience to decide whether to pay extra for the higher level of service (Fig. 2).

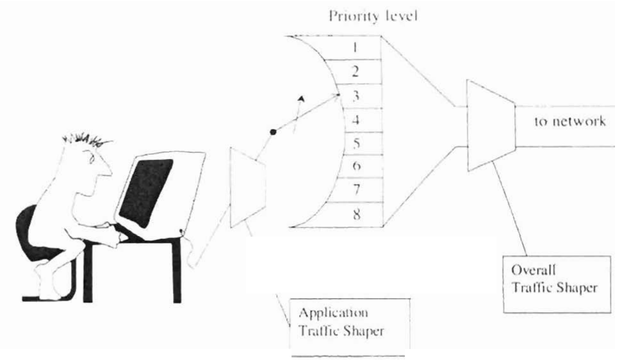

Fig. 2. User direct scenario

The basic pricing plan we propose is straightforward. Each priority level is priced at a different rate. The prices are strictly increasing with regard to priority. The usage of the different priority levels will be monitored. Usage will be metered. This could be as the number of bytes transmitted or the number of packets transmitted. Pricing plans can evolve as usage, and hence demand, of the priority levels is monitored. Moving among the priority levels will be on user time scales so that end-users can respond to the varying quality of service. Their basic choice will be between moving to a better QoS priority level or to a cheaper level. This choice is made on the basis of utility of a session (how important is the quality to current task) or to the nature of the application (send all emails at cheapest, best effort level). The prices of the priority levels are known to the end-user in advance. They may also vary as the service provider alters his pricing strategy but this will be over a long time scale, easily slow enough for the user to keep track. After describing why we want to price on a user time scale and why we want to pursue service differentiation, the technology has to be chosen which can deploy this kind of differentiation. We have chosen to use Differentiated Services with pre-marking. The end user will have to have software available on his system that will allow him to mark his traffic. The user will see a simple selector for the priority level.

Analyses and simulations of the behaviour of the User Direct scenario were performed under various demand models. The main conclusions were:

- With an appropriate pricing structure an ISP is able to increase his revenue over that from a flat fee subscription model
- Also, with an appropriate pricing structure, the social welfare is increased—i.e. the ISP revenue and the aggregate utility of the customers is increased
- The marginal increase in welfare decreases quickly with the number of priority levels—i.e. in practice an ISP should only offer 2 or perhaps 3 levels
- It is not difficult for an ISP, given collected demand statistics, to calculate the optimal prices for the different levels.

2.2. Dynamic price handler (DPH) scenario

This scenario explores the concept of a dynamic price handler (DPH) agent on the customer machines ('D' in Fig. 1) reacting to priced explicit congestion notification (ECN) marks [11]. The idea is to give the agents a price incentive to react to *approaching* congestion on the paths they are using through the Internet, while allowing them to pay to ignore a certain level of this incipient congestion if the value gained from so doing is greater than the charge levied. All that network providers have to do is to deploy ECN on all routers so that the congestion experienced field in the IP packet header is set with a probability related to current load on the egress interface. The receiver's network provider then offers network service at a charge calculated by placing an effectively fixed price on each such mark (the same pricing scheme can be used between network domains too). To avoid increasing pricing for worse service, the marking rate should rise just *before* the queue grows, which we have implemented using a virtual queue [12]. Unlike M3I's guaranteed stream provider scenarios below, edge network providers do not insulate their customers from a potentially variable quality or price. Instead, customers insulate themselves from unpredictability with an agent. It optimises their use of the available service within the constraints of a policy per task either supplied with an application or at session initiation. Each policy is a small data object that encapsulates how utility varies with bit rate for each task [13].

We have implemented such an agent and demonstrated how different policies can give each agent complete control over the network behaviour of various sending applications. Responses to congestion range from completely elastic (like TCP), to a completely inelastic 'non-response', holding a constant bit-rate by paying whatever is necessary during congestion episodes, but only up to a threshold (self-admission control). Policies between these two extremes provide the flexibility to move the bit rate to whatever is considered *best value* for the task in hand given prevailing congestion conditions. Agents controlling flows through the same bottleneck interact, intermediated by congestion signalling. While some inelastic agents are paying to hold their rate, the more elastic agents back-off further to avoid paying (Fig. 3).

2.3. The guaranteed stream provider (GSP) scenario

The motivation of this scenario ('G' in Fig. 1) is to provide a type of service to end-users that incorporates and extends the classical telephony-like service, but without embedding connection-oriented technology in core networks. That is, a service for applications where low bit rates are valued less, pro rata, than high ones (e.g. real-time audio and video). So the benefit to everyone is greater if some users are blocked out while the rest are given capacities over their critical utility thresholds (admission control). Unlike

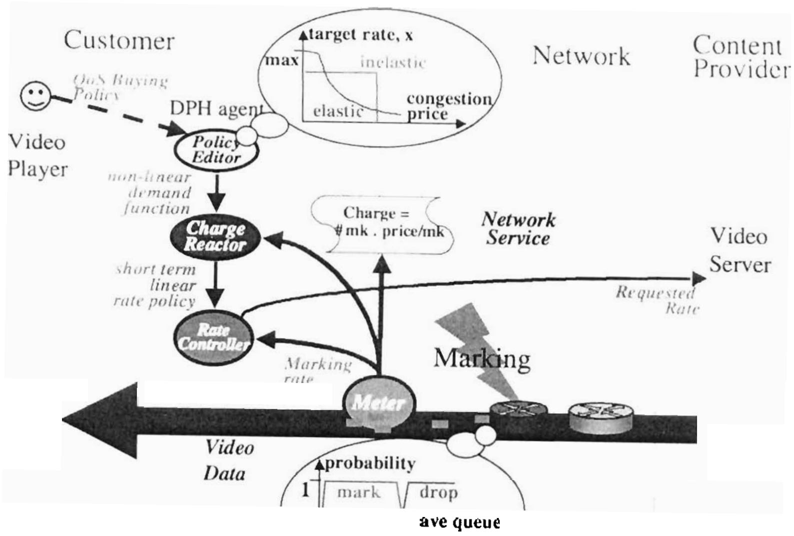

Fig. 3. Dynamic Price Handler Scenario.

agent-based self-admission control above, admission control is offered as a *service* by the network provider. Pricing for capacity reservations can be completely static, or the provider may also choose to vary reservation pricing on slow time scales, perhaps by time of day.

The scenario is based on the following assumptions. Two stakeholders cooperate in order to provide the above services to end-users: an Internet service provider (ISP in Fig. 4) providing a basic communication mechanism and a guaranteed stream provider (GSP) making the refinement into guaranteed services. There is no switched infrastructure. Instead, the basic communication production platform of the ISP is based on packed-switched IP technology, with ECN deployed on every router, as already described in the above DPH scenario. Again we start congestion notification *before* queues start to grow, this time experimenting with load-based rather than queue-based marking. Thus the signalling interface, I3 and I4, between ISP and GSP is simply the packet marking rate. The GSP offers an interface based on the standard reservation protocol (RSVP [14]). The only RSVP-enabled routers are those at the edges of the network. Reservations are forwarded between them across the ISP's non-RSVP-enabled routers which simply treat them as data.

The GSP protects reserved traffic across the ISP by simply clearing the ECN capability field in the packets of unreserved traffic. During congestion, unreserved traffic is therefore dropped by the ISP, while reserved traffic in the same queue is merely congestion marked. When congestion signalling into the GSP rises above a threshold, it starts denying admission to new reservation requests. Thus the level of ECN-capable traffic across the ISP is kept below capacity and queues remain low because marking starts *before* load reaches congestion. Our experiments [15] have shown that the guarantees that result are indistinguishable

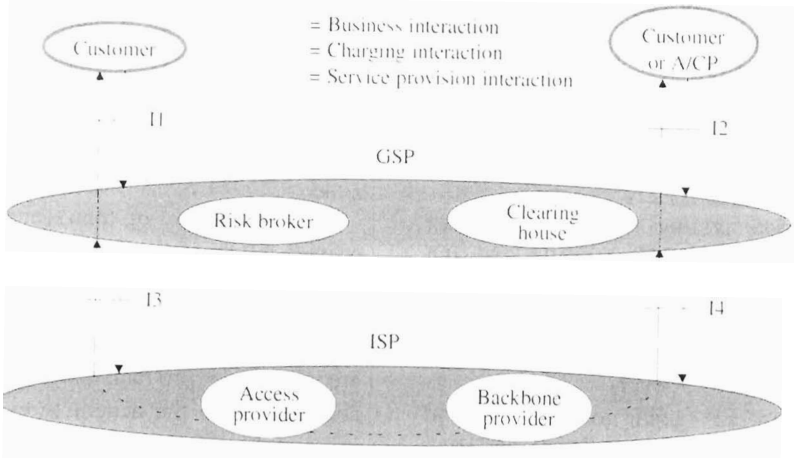

Fig. 4. The guaranteed stream provider scenario.

from those of a homogeneous network with per flow integrated services processing on every router, but without the scalability problem this creates.

2.4. Cumulus pricing scheme (CPS) scenario

The cumulus pricing scheme (CPS) scenario is the scenario that considers explicitly long time-scale pricing. Rather than controlling short term congestion, as in the previous three scenarios, it uses long term over-sizing, just as in Diffserv, but improves the market signals for sizing calculations. Therefore, rather than implying it is comparable with the other scenarios, it has been omitted from Fig. 1. In a sense, CPS can be stated as a dynamic flat rate pricing scheme with an appropriate feedback mechanism. Indeed the scope of CPS claims, since it defines a new approach, investigation on contracting by service level agreements (SLA) and as a consequence investigation on traffic heuristics for correct estimation of customer requirements. In the M3I project, CPS is applied to a differentiated services (DiffServ) environment. The idea is to merge the two systems and to profit from synergies in the areas of contract negotiation and of contract supervision. Pricing schemes form the essential part of a business model for Internet service providers (ISP). A pricing scheme applied to the transport of data in an IP network needs to cope with a number of issues of the IP technology utilised. Therefore, the scheme designed at this stage was termed cumulus pricing scheme (CPS) and has been explicitly developed for the differentiated services internet architecture (DiffServ). CPS proposes a paradigm shift and argues that the problem of Internet pricing is not a matter of complexity, but instead a problem of mapping multiple and multi-dimensional time-scales. The developed scheme shows a simple, transparent, market-managed, and feasible Internet pricing scheme. CPS is a flat rate scheme founding on SLA contracts between customers and ISP, whereby the customer may itself be an ISP. It provides individual and dynamic adaptation of flat rates on long-time scales due to SLA contract ruptures and/or renegotiations. The compliance of the contract is motivated and supported by a feedback mechanism, the cumulus points (CP), and the liberality for deviations on short-time scales, due to statistical metering and average CP accumulation mechanisms.

3. Engineering components

In this section, the engineering components are described, giving information on their design and realisation. The M3I technology components are designed to be put together in different ways to realise various QoS technology and tariffing scenarios. The main sub-systems are described in the following sections:

Tarriff communication is the primary method for distributing tariffing policy to the other sub-systems;
Charging and accounting is the function that applies whatever tariff is chosen to measured data in order to calculate each customer's charges;
Price calculation is the function that calculates optimal prices given current loading. It may calculate *internal* shadow prices that merely guide the provider on the advisability of its *actual* market pricing;
Charge reaction is a function customers use to control their load dependent upon prevailing charging. In many scenarios this function is provided by a human not software (the dynamic price handler is one exception);
Data gathering is used by the provider, both as an input to the charging accounting system, and for price calculation, both via mediation;
Mediation is necessary to aggregate gathered data and do format conversions necessary in practice.

Fig. 5 shows the basic components and their relationships in one of the more important arrangements of the M3I architectural components (enterprise policy control (EPC) and billing are outside this paper's scope):

3.1. Price/tariff communication

The price communication protocol (also called tariff distribution protocol) is a flexible protocol that can use a number of different transport mechanisms like UDP multicast, HTTP and RSVP to distribute tariffs between the ISP's management systems and also to customers. The protocol makes no special assumptions about the QoS architecture used (Intserv, Diffserv etc.).

To give ISPs freedom, tariffs can be distributed as Java code, thus every imaginable tariff can be realised. Dynamic pricing is feasible as the protocol supports a push mechanism and small-sized messages. However, dynamic pricing is usually realised by applying a fixed price to something variable within the network (e.g.congestion signalling) rather than using this protocol for price updates. The Price Communication Protocol is currently planned to be standardised via the IETF under the name 'tariff distribution protocol' [16]. More information about it can be found in Ref. [17]. Introducing a new tariff and updating existing ones cause problems in the charging and accounting system of an ISP. The Protocol includes mechanisms to solve these problems, more information can again be found in Ref. [17].

3.2. Charging and accounting system (CAS)

The CAS has to support economically controlled management, it therefore has to determine and utilise current network resource usage information (e.g. per-customer, usage feedback).

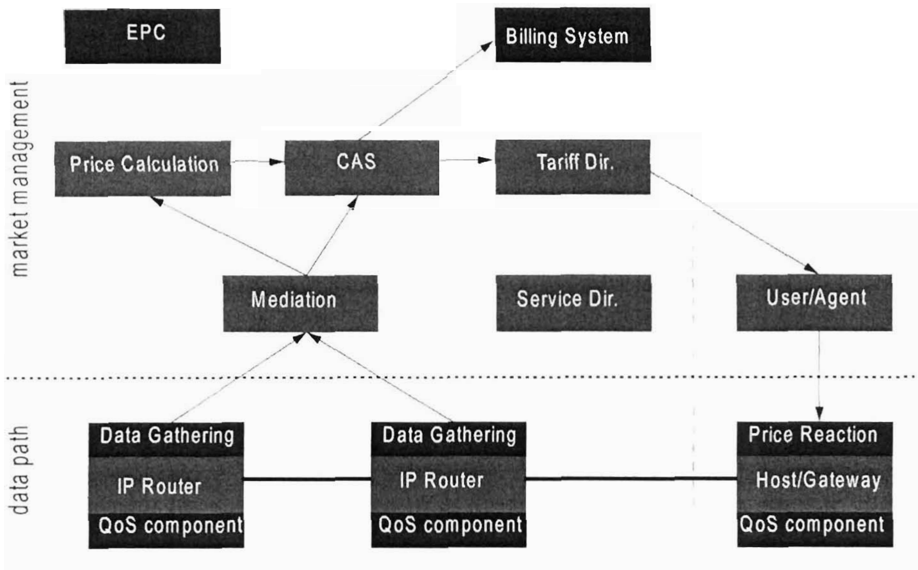

Fig. 5. A typical high level design.

The four basic modules of the CAS are depicted in Fig. 6. The accounting module collects data about the sessions or bulk usage of each customer that is provided by the mediation module. The charging module applies the tariffs that are sent from the price calculation module via the tariff database using the tariff distribution protocol. It calculates the charges for the finished sessions and its output is again the input for the billing mechanisms of the provider or a subcontractor. The customer support module manages the contracts and SLAs with the customers while the user support module can give online feedback to users about their current and past sessions, charges etc. For further information see Ref. [18].

3.3. Price calculation

The price calculation module in M3I is used to set prices automatically, based on the policy of the provider. It supports frequent price updates and therefore dynamic pricing. Inside the module one or more price calculation algorithms are used. These algorithms decide when price changes via a tariff update are necessary. They also calculate the tariff parameters and send out the tariff messages using the price communication middleware. The provider is free to use one price calculation algorithm for one or more than one tariff.

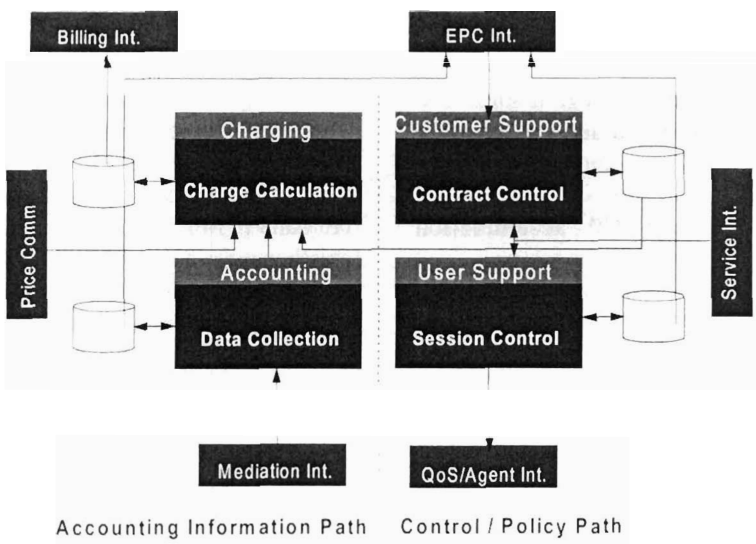

Fig. 6. Charging and accounting system (CAS).

The input of a price calculation algorithm is done via a connector. In M3I three kinds of connectors can be used. First, a connector to mediation (see below); second, a connector that receives information from the policy decision points (also see below) and third, one to the CAS. A connector offers a push and a pull mode, which means it can inform a price calculation algorithm of important events (push) while at the same time the price calculation algorithm is free to request a status update any time it wishes.

The data that is passed through a connector is encapsulated in a normalised meter event (NME), an IUM concept, (see Section 3.6) that contains a number of type/value pairs. See Ref. [19] for more information.

3.4. Charge reaction

A major goal in the M3I project was the investigation of supplementing the prediction of network supply with flexibility in the price domain for fine control of demand. When predictions turn out to have been wrong, the price can be raised to prioritise available capacity for those most willing to pay. Regular price variation can be used to signal congestion.

Price-based QoS control is separated into two parts: charge reaction and QoS control, generally both on customer machines. The aim of the charge reaction function is to produce a policy for the QoS controller. The charge reaction function is a high level, flexible module that produces a policy for the QoS controller. The QoS controller is separated out from this, as it must directly control the flow of network traffic and therefore must sit low in the communications stack, preferably in the kernel (or equivalent) of the operating system. See Ref. [20] for more information.

3.5. Data gathering

Data gathering is the process that provides general ways to meter and sample the usage of the router resources. In M3I, two different data gathering implementations have been realised, which would operate in parallel if two different tariffs requiring them were in force simultaneously:

- COPS-based [21] data gathering for session start and stop events;
- NeTraMet-based [22] data gathering for intra-session packet data.

The common open policy service (COPS) protocol [21] is a simple client-server model for supporting policy control over QoS signaling protocols. The policy server is called policy decision point (PDP) and its clients policy enforcement points (PEPs) [23]. In the COPS based data gathering, edge RSVP routers are PEP entities and there is a central PDP module for every sub-network.

NeTraMet is an implementation of the IETF realtime traffic flow measurement (RTFM) architecture [24]. We have extended NeTraMet to provide counters for the ECN field. Additionally our extended tool is configurable through an interface (API) by policy rather than just manually.

In both cases, either the PDP, or the NeTraMet Reader collects the data provided from the routers, filters them and then, forwards this information to the charging and accounting system (CAS) and the price calculation modules, through the mediation component.

The configuration of the data gathering modules is based on the price calculation algorithm.

3.6. Mediation

Mediation is the component that performs aggregation and correlation techniques on the observed data that have been collected at the data gathering modules. Since the information that is collected is linearly related to the collection time, if collected data is supplied as is to the CAS and price calculation modules, the amount of collected data becomes very large. Instead, mediation correlates and aggregates the observed data and it provides a 'compressed' form to the CAS and to price calculation.

For the realisation of the mediation module, the HP Internet usage manager (IUM) [25] has been used. IUM is a Java-based framework, for which we have provided two interfaces, one for NeTraMet and one for the COPS-PDP module.

4. Summary conclusions from modelling work

4.1. Economic models

Dynamic charging provides good incentives for end-system demand on the network, leading to good economic performance. Dynamic charging schemes such as explicit congestion notification (ECN [11]) charging [26] provide feedback at the fastest timescales, enabling end-systems to control their demand in a way that is appropriate to their application service requirements.

Interconnect agreements dealing with quality of service naturally have the problem of information asymmetry, since each provider generally has more information on the state of its own network. Economic models can expose some of the problems that can arise from relatively inflexible contracts (where no payment takes place if the agreed quality is not delivered), and can show how it is beneficial to all parties if more flexible contracts are used—for example, SLAs offering multiple charge/QoS choices.

Congestion pricing is a form of market segmentation—user demands are differentiated according to their resource requirements and willingness-to-pay. Service providers may therefore favour congestion pricing as a means of extracting value, but if competing service providers use congestion pricing the result is increased competition since they are competing over many different price points.

4.2. Network models

Dynamic charging can be achieved practicably and efficiently through either of two M3I scenarios—user direct and the ECN dynamic price handler. The user direct scenario is based on differentiated services with priority pricing, and designed to provide a user-friendly interface to end-systems where prices for different priority levels are fixed but their performance varies. Queuing analysis, supported by simulation and actual test-bed experiments, of this and similar systems have shown that end-systems can adapt appropriately, thus leading to overall stability and efficient utilisation of network resources [27]. Further work is required to determine how the provider should set prices for priority levels. The ECN charging scheme has been studied in more detail within M3I, and the major results are summarised in the following section.

Further work is needed to investigate network stability under dynamic charging, and whether end-systems will require specific incentives or constraints to ensure stability. Real-time streaming applications are likely to favour rate stability, but non-real-time transfers have an incentive to use on–off type rate control which could lead to instabilities in network traffic.

The Guaranteed Stream Provider role introduced in Section 2 has been shown to be viable in economic terms, and methods have been developed to support the call acceptance and price-setting functions of the risk-broker role based on either predictive models or statistical measurements.

Diffserv pricing schemes can be extended to GPRS networks in a way that ensures economically efficient use of mobile network resources.

4.2.1. Detailed results for ECN charging

The work on ECN charging contains both detailed packet-level simulation studies, as well as experiments with actual implementations in a test-bed and how they depend on the particular characteristics of the packet marking algorithms [28]. Different rate control algorithms, operating in end-systems, were considered, including window-based algorithms and radically different algorithms for file transfer applications. The packet marking algorithms that were investigated, operating in network routers, include RED (random early detection [29]), virtual queue marking [7], and load-based marking.

The results from the simulation experiments, which were also verified with testbed experiments, show how service differentiation and performance, in terms of queuing delay and average throughput, are affected by the rate control algorithms, and how they depend on the particular characteristics of the packet marking algorithms. By service differentiation we refer to the ability of the end-system rate control algorithms, working in conjunction with the marking algorithms in routers, to offer different levels of throughput to connections with different weights or willingness-to-pay values.

The interaction of the marking algorithms and congestion control algorithms was investigated using the marking probability as a function of average utilisation, since the latter function affects the convergence and stability behaviour of the system. For marking algorithms based on the queue length, such as RED, it was found that smoother traffic can result in a steeper marking probability function, hence can increase the degree of fluctuations of the congestion window and the sending rates, and could compromise stability. Moreover, probabilistic marking results in smoother traffic hence higher utilisation. Nevertheless, with appropriate tuning, all three marking algorithms can exhibit the same marking probability as a function of average load; this result shifts the focus of the comparison of marking algorithms away from the achievable utilisation, towards how easy it is to tune the parameters of a particular marking algorithm, and how robust the algorithm is to varying characteristics of the received traffic (smoothness). This observation is in agreement with other works, which however focus exclusively on RED.

If ECN charging were to be widely rolled out to end-users there would be a risk that end-systems might choose to use overly aggressive rate control (in order to ensure quick charge reaction), which might compromise network stability. How to limit such aggressiveness, or how to provide incentives for users, working in their own benefit, to avoid such aggressiveness, is an issue that requires further investigation.

We have developed and analysed a procedure for estimating the average utilisation in equilibrium. Indeed, the equilibrium can be visualised as the intersection of two curves: a curve giving the marking probability as a function of the average load (which is determined by the marking algorithm implemented in the routers and the rate control algorithm operating in the end-systems), and a curve giving the total demand for resources (which is determined by the policy, expressed in the form of a utility function, of the rate control algorithms operating in the end systems). The procedure and corresponding model has been extended to cover cases where both elastic and inelastic traffic coexist.

Based on the above procedure for estimating the average utilisation, and if each ECN mark is charged by a fixed price, one can determine what this price should be in order to achieve a target utilisation; this target utilisation can depend on the average queuing delay or loss ratio that is to be supported. In cases where both elastic and inelastic traffic coexist, the selection of the price per mark can be used to achieve an optimal sharing of resources between elastic and inelastic traffic.

Further information on all aspects of market modelling carried out in the M3I project can be found in the many papers available from the M3I Web site [30].

5. User experiments

A number of focused experiments on user sensitivity to the price of network quality of service have been conducted. Quantitative experiments were used to investigate sensitivity to the stability of quality and of price, and numerous other factors. Qualitative experiments have also been conducted to assess the attitudes of customers to variable quality and pricing, and to pricing quality of service in general. Many interesting results have been produced and reports are being prepared to appear shortly.

6. Conclusions

We have shown how generic market control technology can be implemented so that network providers are free to choose from a wide range of new business models. We have described four example scenarios each of which represents a choice of network QoS technology and of tariff to price quality. A number of further variants on these scenarios are not mentioned here, but have been analysed to varying degrees with respect to feasibility and commercial viability.

The general *engineering* approach has been to use a few simple, minimal components that are capable of providing generic functions, and to combine them in various ways to implement each pairing of technology and business model—*minimise then synthesise*.

We have also shown how the same approach can be used in *commercial* terms, to synthesise business models at the retail edge of the Internet from simpler packet granularity business models within the core—minimise then synthesise.

A number of insights have been reported resulting from our modelling work. There has been a particular focus on economic analysis of the use of explicit congestion notification (ECN) as a shadow pricing mechanism in the core of the Internet, and synthesising other business models at the edge. We have shown that this is a feasible mechanism for fast control of QoS, which is simple, inexpensive and gives the correct economic incentives. This represents a QoS solution that is compatible with the end to end design principle, and therefore will tend to avoid complexity in the network and foster future innovation.

However, further work is required to understand the theoretical possibility of a second order problem that may arise when end systems are given control of QoS in this way. They can be given incentives to use resources responsibly, but we are not yet able to give incentives to constrain the dynamics of each user, and irresponsible dynamics may cause global instability. Further work is required to understand whether anyone can gain more than they lose by such behaviour.

More generally, we have shown that many other, less radical business models are also possible with our approach, which we offer in the belief that it will open up the whole Internet market to a far greater degree of commercial innovation.

Acknowledgments

The background contributions from all the partners of the M3I project are gratefully acknowledged: BT, HP, Telenor, Athens UEB, ICS FORTH, TU Darmstadt, ETH Zürich and FT Wien.

References

[1] J.H. Saltzer, D.P. Reed, D.D. Clark, End-to-end arguments in system design, ACM Transactions on Computer Systems 2(4)(1984)277–288, an earlier version appeared in the Second International Conference on Distributed Computing Systems (April, 1981) pp. 509–512.

[2] K. Nichols, S. Blake, F. Baker, D. Black, Definition of the differentiated services field (DS Field) in the IPv4 and IPv6 headers, Request for comments 2474, Internet Engineering Task Force, URL: rfc2474.txt (December, 1998).

[3] D.D. Clark, J. Wroclawski, An approach to service allocation in the Internet, Internet draft, Internet Engineering Task Force, URL: http://diffserv.lcs.mit.edu/Drafts/draft-clark-diff-svc-alloc-00.txt, expired (July, 1997).

[4] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, An architecture for differentiated services, Request for comments 2475, Internet Engineering Task Force, URL: rfc2475.txt (December, 1998).

[5] R. Braden, D. Clark, S. Shen ker, Integrated services in the Internet architecture: an overview, Request for comments 1633, Internet Engineering Task Force, URL: rfc1633.txt (June, 1994).

[6] F. Baker, B. Braden, S. Bradner, A. Mankin, M. O'Dell, A. Romanow, A. Weinrib, L. Zhang, Resource ReSerVation protocol (RSVP)—version 1 applicability statement; Some guidelines on deployment, Request for comments 2208, Internet Engineering Task Force, URL: rfc2208.txt (January, 1997).

[7] R.J. Gibbens, F.P. Kelly, Distributed Connection Acceptance Control for a Connectionless Network, Proceedings of the International Teletraffic Congress (ITC16), Edinburgh, 1999, URL: http://www.statslab.cam.ac.uk/~frank/dcac.html, pp. 941–952.

[8] B. Briscoe, M. Rizzo, J. Tassel, K. Damianakis, Lightweight, End to End, Usage-Based Charging for Packet Networks, Proceedings of the IEEE Openarch 2000, 2000, URL: http://more.btexact.com/projects/mware.htm, pp. 77–87.

[9] J.K. MacKie-Mason, H. Varian, Pricing congestible network resources, IEEE Journal on Selected Areas in Communications, Advances in the Fundamentals of Networking 13 (7) (1995) 1141–1149.

[10] R. Andreassen (Ed.), M3I; Requirements Specifications; Reference Model, Deliverable 1, M3I Eu Vth Framework Project IST-1999-11429, URL: http://www.m3i.org/ (July, 2000).

[11] K.K. Ramakrishnan, S. Floyd, D. Black, The addition of explicit congestion notification (ECN) to IP, Request for comments 3168, Internet Engineering Task Force, URL: rfc3168.txt (September, 2001).

[12] R.J. Gibbens, F.P. Kelly, Resource pricing and the evolution of congestion control, Automatica 35 (12) (1999) 1969–1985.

[13] S. Shenker, Fundamental design issue for the future Internet, IEEE Journal on Selected Areas in Communications 13 (7) (1995) 1176–1188.

[14] R. Braden (Ed.), L. Zhang, S. Berson, S. Herzog, S. Jamin, Resource ReSerVation protocol (RSVP)—version 1 functional specification, Request for comments 2205, Internet Engineering Task Force, URL: rfc2205.txt (September, 1997).

[15] M. Karsten (Ed.), GSP/ECN Technology & Experiments, Deliverable 15.3 PtIII, M3I Eu Vth Framework Project IST-1999-11429, URL: http://www.m3i.org/ (February, 2002).

[16] O. Heckman, V. Darlagiannis, M. Karsten, B. Briscoe, Tariff dissemination protocol, Internet draft, Internet Engineering Task Force, URL: http://www.ietf.org/internet-drafts/draft-heckmann-tdp-00.txt (March, 2002).

[17] O. Heckmann, V. Darlagiannis, M. Karsten, 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), URL: http://www.kom.e-technik.tu-darmstadt.de/publications/abstracts/HDKS01H-1.html., 2001.

[18] B. Stiller (Ed.), M3I charging and accounting system (CAS) design, Deliverable 4, M3I Eu Vth Framework Project, IST-1999-11429, URL: http://www.m3i.org/ (June, 2000).

[19] M. Karsten (Ed.), M3I Pricing Mechanism (PM) design, Deliverable 3 Pt I, M3I Eu Vth Framework Project, IST-1999-11429, URL: http://www.m3i.org/ (June, 2000).

[20] B. Briscoe (Ed.), M3I pricing mechanism design: Price reaction, Deliverable 3 Pt II, M3I Eu Vth Framework Project IST-1999-11429, URL: http://www.m3i.org/ (July, 2000).

[21] D. Durham, J. Boyle, R. Cohen, S. Herzog, R. Rajan, A. Sastry, The COPS (common open policy service) protocol, Request for comments 2748, Internet Engineering Task Force, URL: rfc2748.txt (January, 2000).

[22] N.J. Brownlee, The NeTraMet System, Software Release Notes, URL: http://www.auckland.ac.nz/net/NeTraMet/ (December, 1997).

[23] R. Yavatkar, D. Pendarakis, R. Guerin, A framework for policy-based admission control, Request for comments 2753, Internet Engineering Task Force, URL: rfc2753.txt (January, 2000).

[24] N. Brownlee, C. Mills, G. Ruth, Traffic flow measurement: architecture, Request for comments 2063, Internet Engineering Task Force, URL: rfc2063.txt (January, 1997).

[25] HP, Smart internet usage, White paper, HP, URL: http://www.hp.com/smartinternet/media/siuwp2.html, 2000.

[26] F.P. Kelly, A.K. Maulloo, D.K.H. Tan, Rate control for communication networks: shadow prices, proportion fairness and stability, Journal of the Operational Research Society 49 (3) (1998) 237–252.

[27] J. Altmann, H. Daanen, H. Oliver, A. Sánchez-Beato Suárez, How to Market-Manage a QoS Network, Proceedings of the IEEE Conference on Computer Communications (Infocom'02), 2002.

[28] V.A. Siris, C. Courcoubetis, G. Margetis, Service differentiation and performance of weighted window-based congestion control and packet marking algorithms in ECN networks, Computer Communications (This volume).

[29] S. Floyd, V. Jacobson, Random early detection gateways for congestion avoidance, IEEE/ACM Transactions on Networking 1 (4) (1993) 397–413.

[30] M3I Partners, Market Managed Multi-service Internet, M3I project Web site, URL: http://www.m3i.org/, 2000–2002.