Scalable Usage Based Internet Accounting

Jérôme Tassel Bob Briscoe Mike Rizzo
Konstantinos Damianakis
<bob.briscoe@bt.com> <www.btexact.com/people/briscorj/>
BT Research, B54/130, Adastral Park, Martlesham Heath, Ipswich, IP5 3RE, England
Tel. +44 1473 645196

1 Mar 1999

Abstract

In this paper we present a novel scalable accounting infrastructure to support charging for network usage and Quality of Service (QoS) in an Internet context. Measurement and accounting are two core processes of a charging mechanism that make heavy demand on resources. They reduce the resource available for the core processes of the network i.e. routing and forwarding packets. Increasing the processing power and data storage capacity of the affected network elements lead to an increment in the cost of the network. The underlying principle of the infrastructure we propose is the transfer of these processes onto the edge systems to reduce their impact on the cost of the network. The measurement, accounting, applying pricing and billing processes and related sets of data are relocated on the edge systems of the network while allowing the provider to retain control over them. This paper focuses on the measurement and accounting aspects of this infrastructure. To achieve scalability we propose not to meter all the data or QoS control packet but only samples of them. We also discuss which controls both users and network providers could desire over the relocated processes. Early implementation of this work is introduced as a practical example of the concepts we present.

Keywords: Data Communication, Networks, Internet, Charging, Accounting, Measurement, Metering.

1 Introduction

The Internet is moving from a free best effort network used for research purposes by a relatively small number of users to a commercial multi-service network used by millions. As multimedia applications hit the Internet a new infrastructure is required to support them. Originally a best effort network, the Internet is now becoming a multi-service network through the use of the differentiated services and integrated services architectures. Those architectures define protocols that reserve resources for given streams of data or prioritise traffic as it passes through network elements in a attempt to overcome delays due to congestion. Streams of data with different requirements (e.g. maximum delay, jitter) belong to different classes of service. This implies that more resources (e.g. router CPU cycles, buffers) are used by those classes of service that have the highest requirements. While those resources are in use the effectiveness of statistical multiplexing for data transfer is decreased as resources are exclusively reserved (at the access control stage at least in the case of RSVP).

As a result in a QoS enabled Internet, ISPs will need to be able to find out about the usage of different classes to achieve fair charging, as straight network usage figures or flat monthly rate will not reflect the actual consumption of resources which will depend instead on the service class being used .

When a number of streams compete for the same limited amount of resources not all can be satisfied at the same time. QoS control mechanisms do not solve the problem of congested and overloaded networks that have high demands at peak times. Currently static access control mechanisms are in place to account for the usage of the resources and accept or reject requests based on the amount of resources remaining, the last request is the one that is denied if resources are limited.

Another way to solve this problem is to use differentiated pricing in order to manage demand. As in any economic market, raising and lowering prices would lower or increase the demand for bandwidth. If all service classes are priced equally then users would always request the highest class even when they do not need it. Pricing each service class differently would therefore solve this first problem we can assume that users would only want to spend

the minimal amount of money to access the level of service they require. Secondly when congestion happens prices can be increased so that users with lower priorities back off, enabling other users with higher priorities to use those resources. This solution is a well know one to the PSTN world where the load of the network is balanced over the period of a day by imposing higher rates during day time and lower rate at night. However we envisage that such rate variation could be much more dynamic on the Internet [9] while at the same time allowing people to choose stable pricing.

Some might argue that controlling bandwidth demand using pricing is an answer to a problem that does not exist, as the network can simply be over-provisioned as bandwidth is becoming cheaper and cheaper [8]. One of the long term objectives of our work is to find out if bandwidth over-provisioning really is more cost effective than price managed bandwidth. The work presented in this paper does not cover the design or selection of an appropriate pricing model [8] but focuses on the accounting infrastructure required to support such models (auctions, markets... etc.).

One of the main issue in measuring network usage or service usage is that of performance. Adding measurement processes in the data path can increase the processing time of the data packets on this same path [5]. In a multi service network this processing time is already increased by the enforcement of some QoS mechanism irrespective of whether intserv or diffserv frameworks are being used. Our approach is to transfer as much as possible of the measurement, accounting and pricing loads from the service provider network elements onto the user machines at the edge of the network. The provider still retains control over these processes and the data they produce by using remote control facilities. To ensure that those processes are taking place correctly on the user machines, the provider performs some auditing of the data generated by the users. Auditing consists of comparing a sample of the user's generated data with an equivalent sample at the provider end, the provider only having to perform the measurement for the duration of the sampling period. Moreover, to ensure that the user is unaware of the auditing periods, they take place at random times.

The infrastructure for enabling the relocation of the accounting and measurement processes to the user machines is the core of this paper. The first part of this paper presents some background to this work. The second part describes the main concepts regarding the accounting and measurement aspects of the infrastructure we propose followed by a description of some early prototyping for which an understanding of UML and Java is useful.

2 Internet Accounting Requirements

In this section we describe some of the desired requirements for an effective and flexible Internet charging infrastructure. We draw on some earlier requirements presented in [7, 5, 4] and current IETF work in the draft stage. There are requirements both from the service provider and the user point of view. From the service provider point of view the requirements include the following:

Commercial flexibility It is unlikely that there will be a single accounting solution that fits the requirement of all service providers. To ensure that ISPs can inter-operate, if they decide to, a clean and stable general design of their specific accounting system would enable them to easily provide open interfaces. Those open interfaces would only provide external access to the functions of their systems that are required to inter-operate. Moreover with well-defined interfaces it will be easier to inter-operate with legacy systems. For an Internet accounting system to be successful it needs to be flexible enough to support any business charging models. It should not embed a specific charging model but provide the appropriate hooks to support any of them. The final aspect regarding commercial flexibility is that the data generated by the accounting processes is also valuable for other purposes such as network management, capacity planning, marketing and should therefore be available outside the accounting context.

Scalability and performance The number of customers, transactions and data related to usage based accounting for the Internet can be expected to be larger than any existing accounting base. If the accounting processes are in the data path then the cost of the network elements is increased as more processing power and storage space if required on them to achieve equivalent performance. The design of the accounting infrastructure must therefore ensure that the impact on the existing network infrastructure is low and that it can scale to support the expected large set of customers. Accounting does not add to the service being provided so it should incur minimal costs.

Robustness and scalability If some accounting data is lost or duplicated because of hardware failures (device crashes and reboots) then it must be possible to recover to a stable state or generate statistically the missing set of data.

Simplicity The accounting system needs to be simple to be understood, interfaced with other systems and support any business charging model. It will also permit its re-use for other accounting purposes than network related ones.

Technical flexibility The accounting system needs to be able to deal with the different network technologies being used. It should be able to support any QoS frameworks and protocols. Moreover the accounting system needs to be independent from the application or transport protocols being used. The service that is actually being provided is the routing of IP packets and any applications or transport protocols can be used by the users, including their own proprietary ones.

Security The accounting data must be available only to the indented recipients and it should not be possible to forge or fake it. The control of the accounting processes must also be secured in a similar manner. Denial of service attacks are also a potential threat to the accounting system that needs to be catered for.

The above requirements are also valid for the users of the services but open interfaces are of particular importance to enable the integration with their own accounting and network management systems.

Now that we have described the requirements of an Internet accounting system, we follow by presenting our infrastructure to meet those requirements..

3 Internet Accounting Infrastructure

From the requirements we described in the previous section and the intended purpose of our Internet charging infrastructure (presented in the introduction to this paper) we have designed a novel Internet charging architecture. Figure 1 illustrates our architecture and describes its components:

The main concepts of our architecture are:

- Sampling of measurement instead of continuous measurement on the service provider domain
- Transfer of the charging processes and state onto the users' machines
- Measurement at the IP layer
- Common architecture for end-user charging and inter-provider charging. In this paper we use the word customer both for the end-user and provider that is also a customer of other providers.

In this paper we only focus on the accounting part of this infrastructure, readers interested in other aspects of the work can find more information in [1, 9].

The underlying principle of the infrastructure we propose is the relocation of the processes that are required for charging from the network provider's systems (computers and routers) onto the customers' machines. This is possible due to the increasing intelligence of personal computers, mobile and Internet devices. Moreover the software technology is now available (Java) to facilitate software upgrades in dynamic environment through distributed programming technologies. The functions we propose to transfer to the users' machines are:

- The measurement component that meters the network consumption
- The accounting component that aggregates, stores measurement data and relates it to customer and tariff information
- The pricing component that gathers and keeps track of the tariffs and applies them
- The billing component that creates bills from the accounting data
- The payment component that can be initiated from a bill on the user's machine

Although those processes are on the customers' machines, the provider retains the control over them, as illustrated on 1. The terms of this control can be defined in contractual terms between the customers and the service provider. For example, the provider makes decisions over and control what measurement is to be done for each user. The obvious issue regarding this approach is what happens if the user falsifies the measurement or accounting data. To solve this issue the user machine reports to the provider system the data it produces. This reporting process is controlled from the provider's system which checks the validity of this received data by comparing it with samples of accounting data it generates on its own local measurement over a randomly chosen sampling period.

The randomness factor is used to ensure that the user is unaware of the period when the data its system generates is audited. The provider is therefore able to discover breaches by the discrepancies

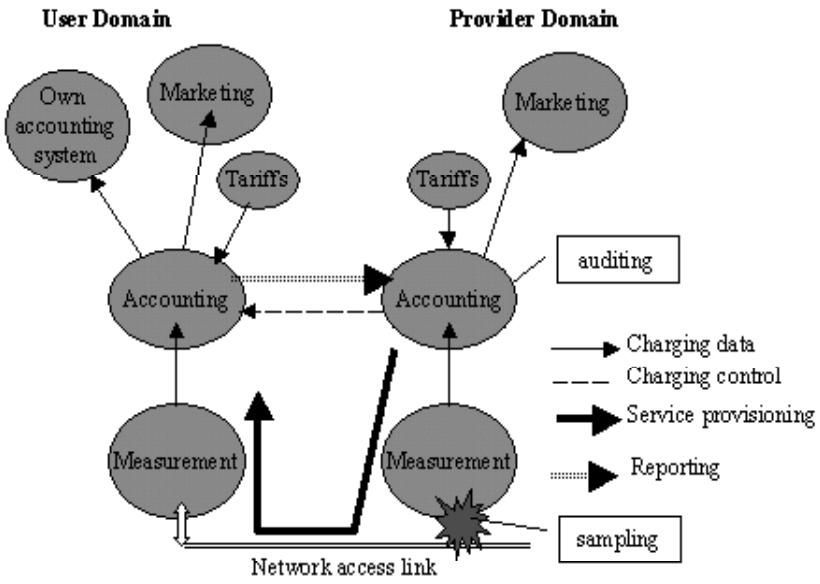

Figure 1: Accounting Infrastructure

between its own system's accounting data and the reported data.

We assume that there is always a minimum amount of bandwidth available for reporting and control to take place. Alternatively some bandwidth could be reserved for this sole purpose as it would be insignificant in comparison with the amount of data traffic. In the case where the link between the user and the provider systems is broken then the data path is broken as well so reporting is not needed (this is the same as when the user's machine is switched off).

All the data generated by charging resides on the customer machines and the provider has access to it in the granularity and intervals it desires. For example a user might desire daily bills but her network provider only monthly bills. This decreases the processing power and storage space required in the provider management domain, which to date accounts for about 6% [1] of the revenue of a telecommunications company. In addition this gives more control to the user in respect of her spending.

With this architecture, the processes required for a per-packet-charging model, no longer constitute a source of bottlenecks on the network provider routers and support systems. In the following section we describe in more detail the metering aspects and the accounting component of our infrastructure.

4 Accounting & Measurement Concepts

In this section we detail the concepts of the measurement and accounting components of our architecture.

4.1 Metering component Concepts

The main aspect about the measurement in our work is that it is not centralised at the service provider entry point but dispersed to the customer machines. Having a single measurement point for all users of a network creates a bottleneck, which affects all users. Dispersing this measurement to the user machines removes this bottleneck. As a result each user suffers a smaller degradation in network throughput from measurement but will experience some CPU degradation. One aspect of our work is to try to measure this degradation depending on the data traffic.

We have decided to measure network usage at the IP layer because in an "IP over everything" network this will produce data that can be understood by all parties involved in the charging process and our architecture is heavily based on comparisons. Measuring lower down the stack would require specific tools for every protocol (ATM, SDH, Ethernet...) and would produce data that would need to be reworked to be compared. Moreover it is possible, at the IP layer, to differentiate data packets from signalling packets (such as RSVP PATH messages)

and therefore classify them accordingly. Tariffs can then be applied according to the charging model in use by the business (signalling packets might be free, cheaper or more expansive than data packets).

Another important aspect of the meter is that it has not got a unique controller. The meter delivers information to the accounting process but the information it produces might also be useful for other purposes such as QoS monitoring or network management. The type of information and its granularity depends on the recipient of the information. As a result the meter can be controlled by different processes and must be able to optimise the measurement to provide all the requested information. Each process might require a different control interface to the meter, so far we have designed the meter interface for the accounting process.

We also ensure that any processing that is not required to be done by the measurement process is delegated to the recipients of the measurement information such as the accounting process. For example the accounting component and not the meter does customer related aggregation.

4.2 Accounting Component Concepts

The role of the accounting part of the system is to gather measurement information from the meter and create, store, aggregate, report and delete accounting records. Furthermore, on the provider side only, accounting is involved in the auditing process because it holds the information that is used to do so: i.e. the set of measurement data from the customers and the sampling data from its own meter. Another objective of the accounting process is to relieve the meter process of any function that does not need to be performed per packet.

The accounting process is independent from the type of data that might be measured and is therefore independent of the technology that is to be accounted for. Introspection techniques (with which the set of accessible methods and attributes of a class can be found at run time) can be used to find at run time the type of information that is accounted for. Dynamic class loading techniques (with which a class type can be loaded at run time) can be used to introduce new types of measurement at run time. This combination of techniques is very effective to provide a flexible and future proof accounting component. As a result this accounting component can be used to account for other usage than network usage, such as sessions or Web content.

The impact of the accounting information on the network is minimised by using aggregation techniques. For example records dealing with the same IP addresses and port numbers and of the same nature, e.g. packet size, can be aggregated to further reduce the load of the accounting data on the network. The combination of compression techniques with aggregation techniques can also reduce further this impact. Aggregation takes place both in the meter that we presented in the section above but also in the accounting component. The meter can only aggregate using network information (such as IP addresses, port numbers). The accounting component has more information available to further aggregate this data such as the source and destination customer identifiers.

A single provider accounting system could potentially be dealing with a large base of customers. In order to lower the load on the service provider accounting system the reporting of accounting data operates under a push model. The service provider's system registers its interest with the customer's accounting system and it is then the customer's system that sends accounting data to the service provider. The customer reporting process supports multiple recipients with different reporting specification and the service provider system supports different reporting properties for each customer. This is to allow for the re-use of the accounting information in other processes and the individual control of the users' accounting processes. Other processes might be for network management or marketing related purposes, which require information of different granularity. As a result the accounting process on the customer or provider systems must be able to multiplex reporting requests with different parameters. For example the provider's own network management system might require measurement figures on a different time frame than the payment system.

An accounting process residing on the customer machine is controlled remotely by its provider accounting system. We have identified the following needs regarding remote control of a customer accounting process:

- Specify which services should be measured for which users.
- Define how long to store, delete and aggregate records.
- Specify the identity of the party (account number mapped to network address) whose usage needs to be measured.
- Define whether accounts should simply record usage or also include cost calculation. This is to allow billing to take place on the user machines or the service provider machines.

- Specify how often and where to reporting is required, this is the main item that varies depending on what charging model is to be used [4]. Changing the reporting properties also leads to modification of the meter proprieties in order to match the new requirements. As a result the service provider indirectly controls the users' meter.

5 Implementation Example

We have implemented a Java version of the accounting component we described in the section above. We chose Java because it is an object oriented language with built in introspection and dynamic capabilities. We use the same component for the service provider and the customer. With our current implementation it is possible to run a set of customer accounting systems that report to a service provider system that controls each of them. Each of those systems can run on different machines. Our current implementation also includes a minimal measurement object, we are currently working on a more advanced and complete measurement tool based on NeTraMet [3].

We use the Java event model available in the JDK1.1 onwards as the push model for reporting the accounting and measurement information. To enable the distribution of the reports and measurements across a network different technologies can be used: Java RMI, Java Sockets, Flexinet and others. In our prototype we have tested all of those three methods. We finally settled for Flexinet because it is very easy to add or remove non-functional behaviours such as what transport protocol is used (currently TCP), multicasting, security, auditing and this without modifying the original accounting specific code. What we have effectively achieved is the implementation of a distributed Java event mechanism that we use for reporting accounting and measurement information. More information about Flexinet can be found in [6].

The remote control of the customer accounting system is also done using Flexinet. With Flexinet it is not only possible to send data from a machine to another but also to access remote objects. Each customer accounting component exports a control interface to the Flexinet trader that can then be used by the service provider system to remotely control each customer's system. We use digital signatures and object guards available in the JDK1.2 to restricts access to these interfaces.

In order to be able to account for any type of measurement we make use of the introspection and dynamic class facilities available in Java as illustrated in the figure 2:

The accounting object accepts objects of the type `MeasurementRecord` as valid measurement information. The only constraint on the actual measurement information is to implement the Measurement interface. In our current implementation we cater for two types of measurement, straight packet counts and RSVP related packets. The actual type of measurement received at run time by the accounting object can be found using the introspection facility available in Java.

The service provider can control individually each customer's system by using the GUI illustrated in figure 3. The customer uses the same interface to control which other processes (such as network management tools) might receive the accounting information it generates.

Figure 3 illustrates the case where the service provider BTNet is interested in controlling two of its customers: Private1 and Corporate1. As we have used an object oriented distributed technology, objects that have to be accessed remotely need to have a unique name in a trader that can then give out object references for those objects. The top of the window shows the local naming information: the name of the accounting object i.e.: BTNet and the name of the local meter that is used by it: local. The next line down is used to add reporting sources: i.e. customers. The second half of this GUI is divided into two frames, the left one lists the current customers the accounting process controls and receives reports from. The frame on the right side defines those aspects of the remote customer charging system that can be controlled locally i.e.: the reporting rate and the local auditing process parameters (sampling rate and length). Any changes to the reporting properties of a customer are propagated to this customer accounting system which will trigger the modification of the metering properties on this customer system to match the new requirements. The greyed-out options are subject to further work

In section 2 we presented the requirements for an effective Internet accounting system, drawing on some earlier and current work from the IETF. We have not directly addressed three of those requirements in the presentation of our infrastructure: simplicity, coping with device crashes and reboot and open interfaces. Simplicity is provided by the fact that we use standard tools and protocols and componentise each function of the infrastructure. We have not yet fully addressed the failure transparency requirement but the auditing process on the provider side permits at least to find out about errors. Regarding open interfaces, we are using open tools and protocols (RTFM [2], NeTraMet, Java) and using object oriented technologies permit to define clean and clear interfaces that can

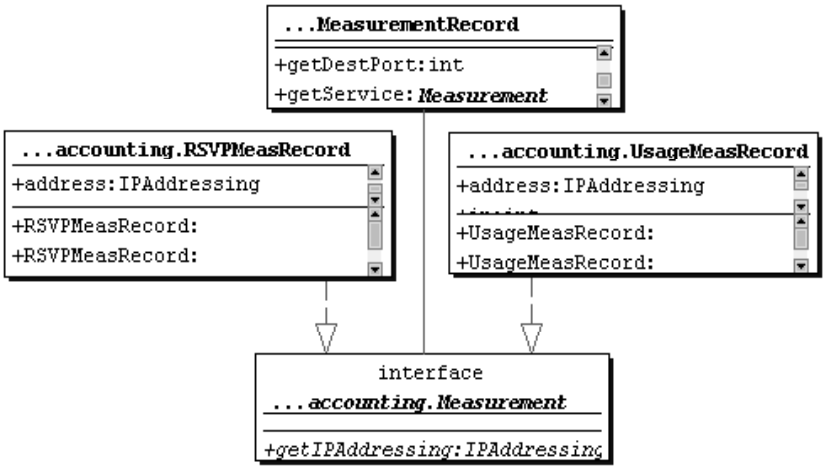

Figure 2: Flexible measurement

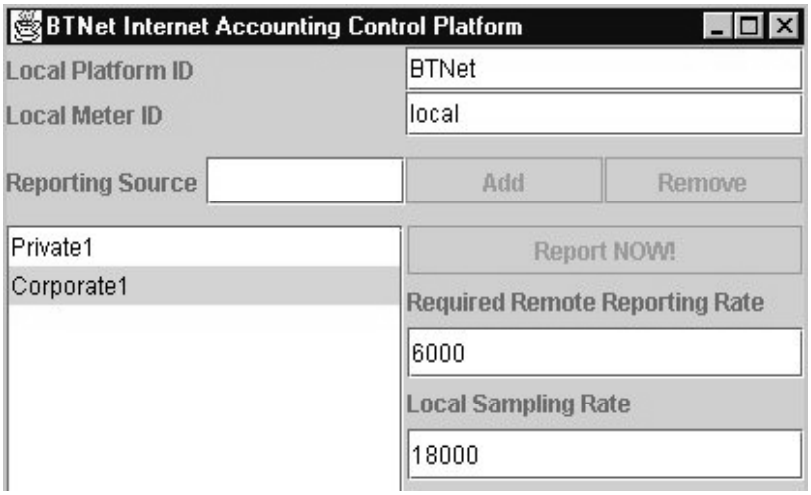

Figure 3: Accounting Control GUI

later be open.

6 Conclusions and further work

As we mentioned in our introduction usage based accounting is strong requirement for the charging systems of the future multi service Internet. It is necessary to be able to match actual consumption of network resources to user charges that differ according to the QoS requirements of their applications. It is also necessary if pricing is used to manage the demand and supply of bandwidth on heavily accessed network elements.

The main concern with usage based accounting for Internet is the impact of measurement on the network elements and the amount of data that this could potentially generate. The infrastructure that we propose solves both those issues. Distributing measurement onto the users' machines and only doing it on a sampling basis on the provider's system decreases the impact of measurement. The accounting processes are also transferred onto the user's machines together with the data they produce in order to decrease the storage requirements on the provider's system. Although this information is stored on the users' machines, the provider retains control over them and can check their validity through auditing. This has not only got advantages for the service provider but also for the users who have more flexible access to their billing data and can integrate it with their own accounting systems or network management tools.

Early prototyping has proved that the infrastructure is practical and that the technology is available to support it. Further work is going to focus on completing the prototype and integrating it with the tariff and measurement prototypes that we are developing. We also plan to add an electronic payment mechanism into the system. Customer trials are also being explored and more advanced modelling work on the economics of using such a model for charging and price distribution. We also believe that this infrastructure can be used to provide a charging mechanism for higher level services such as sessions or Web content. Charging for digital mobile networks (UMTS) is also an area we are looking at for a possible application of this work, as bandwidth is limited and therefore over-provisioning is unlikely to be economic.

References

[1] Bob Briscoe, Mike Rizzo, Jérôme Tassel, and Konstantinos Damianakis. Lightweight, end to end, usage-based charging for packet networks. In *Proc. IEEE Openarch 2000*, pages 77–87, URL: http://www.labs.bt.com/projects/mware/, March 2000.

[2] N. Brownlee, C. Mills, and G. Ruth. Traffic flow measurement: Architecture. Request for comments 2063, Internet Engineering Task Force, URL: rfc2063.txt, January 1997.

[3] Nevil J. Brownlee. *The NeTraMet System, Software Release Notes*. URL: http://www.auckland.ac.nz/net/NeTraMet/, December 1997.

[4] David D. Clark. A model for cost allocation and pricing in the Internet. In *Proc. MIT Workshop on Internet Economics*, URL: http://www.press.umich.edu/jep/works/ClarkModel.html, March 1995.

[5] George Fankhauser, Burkhard Stiller, Christoph Vögtli, and Bernhard Plattner. Reservation-based charging in an integrated services network. In *Proc. 4th INFORMS Telecommunications Conference, Boca Raton, FL*, URL: ftp://ftp.tik.ee.ethz.ch/pub/people/stiller/paper/informs98.ps.gz, March 1998.

[6] Richard Hayton, Andrew Herbert, and Douglas Donaldson. FlexiNet — A flexible component oriented middleware system. In *Proc. SIGOPS'98*, URL: http://www.ansa.co.uk/, 1998.

[7] C. Mills, D. Hirsh, and G. Ruth. Internet accounting: Background. Request for comments 1272, Internet Engineering Task Force, URL: rfc1272.txt, November 1991.

[8] Andrew Odlyzko. The economics of the Internet: Utility, utilization, pricing, and quality of service. *Proc. ACM SIGCOMM'98, Computer Communication Review*, 28(4), September 1998.

[9] Mike Rizzo, Bob Briscoe, Jérôme Tassel, and Kostas Damianakis. A dynamic pricing framework to support a scalable, usage-based charging model for packet-switched networks. In *Proc. Int'l W'kshp on Active Networks (IWAN'99)*, volume 1653, URL: http://www.labs.bt.com/projects/mware/, February 1999. Springer LNCS.