Transaction Management for Sender/Receiver-Payment Schemes in Charging and Accounting Systems for Interconnected Networks

Junseok Hwang[σ θ], *Jörn Altmann*[ϕ κ], *Ibrahim Okumus*[θ], *Praveen Aravamudham*[θ]
[σ] College of Engineering, Seoul National University
Seoul, 151-742, Korea
[θ] School of Information Studies, Syracuse University
Syracuse, NY 13224, U.S.A
{jshwang, iokumus, paravamu}@syr.edu
[ϕ] School of Information Technology, International University in Germany
76646 Bruchsal, Germany
[κ] Department of EECS, University of California, Berkeley
Berkeley, CA 94720, U.S.A
jorn.altmann@acm.org

Abstract

In this paper, we present an Internet transaction management system for sender/receiver payment schemes. This system allows an arbitrary split of transaction charges between two communication partners. Using this kind of system, new business models can be implemented on the Internet. The new system provides more flexibility than existing charging schemes. Under these new business models, service providers can pick up a share of the cost for the transaction with any of their customers; offer collect-call type of services; or provide services as the 900 services on the telephone network.

This paper describes in detail the transaction management protocol (TMP), its implementation, and the transaction management service platform (TMS). The TMP specifies the protocol state diagram as well as the process of how the costs for resource usage can be allocated to communicating end-users. The TMS platform defines the architecture and the modules, simplifying the implementation of the TMP on the Internet. The TMS provides a module-based transaction management environment, carrying transaction signals such as message schema, accounting policy information, communication reference information, and end-user agreement information. In addition to this, an application of the TMS in the framework of bandwidth broker interconnection networks and a short evaluation of the proposed transaction management system are given.

Keywords

Pricing, Charging, Sender/Receiver Payment Schemes, Resource Allocation.

1. Introduction

Currently, Internet service providers are charging their customers a flat rate or a connect-time-based fee for using the Internet. That means, end-users are charged either a monthly, flat rate or for the number of minutes connected to the Internet. These kinds of charging schemes are dominant because of two technical reasons: first, flat-rate charging requires only a very simple charging and billing system, second, connect-time-based charging can easily be implemented by using the POTS billing system [2].

These charging systems not only limit the deployment of new business models but also have many shortcomings. Flat-rate charging allows cross-subsidy, no flexibility, and over-usage of resources. Connect-time-based charging inhibits the use of the Internet. End-users are reducing the time that they are connected to the Internet, since they are over-charged under this pricing plan. For example, even if end-users do not download any data off the network, they are charged. Consequently, end-users avoid surfing on the Internet.

With respect to the discussion about introducing QoS into the Internet and a fair way of charging, pricing plans based on the data volume usage are considered more reasonable and relevant. Such pricing plans would charge users and networks for the actual amount of data transferred. These pricing schemes require new charging technology, which is currently under development by several research communities. The Internet Engineering Task Force (IETF) works on this issue within the AAA working group [1]. However, these charging systems lack one significant characteristic. End-users, which participate in a communication, cannot split the incurred costs in the way they like. Even more, all end-users are getting charged for incoming as well as outgoing data.

The characteristic of the charging and accounting mechanism that we are presenting in this paper allows an arbitrary split of charges between communication partners. It enables more flexible transactions on the Internet. It allows sender-based or receiver-based payment schemes or any combination thereof. This provides flexibility to Internet-based businesses, enabling them to pick up a certain share of the incurred transaction cost. For example, using this mechanism, application service providers can offer to pay part of the network costs for the customers to attract business. Internet service providers can use this scheme to implement collect-call type of services or services like the 900 services on the telephone network.

Considering peer-2-peer technology or grid technology, the allocation of costs is an important factor for the success of these technologies. Taking Napster as an example, the free rider problem is significant, that means, the percentage rate of people that provide resources (e.g. music files, photos) for others on their computer is extremely low [9]. The reason is that there is no economic incentive for people to make their computer resources accessible by others. Although, most end-users do not face any usage-based cost for Internet connectivity in the USA, all end-users face cost for electricity, memory, and processing power. But, there is no way for end-users to recover these costs from whoever accesses the data.

In order to make such a charging and accounting system widely applicable, we developed a platform, which manages different kind of transaction information such as accounting policy information, communication reference information, and end-user agreement information. It is called Transaction Management Service (TMS). A main component of TMS is the Transaction Management Protocol (TMP), which supports the cost splitting negotiations between end-users.
The paper is organized as follows: Section 2 gives background information about charging systems and discusses related studies. The transaction Management Protocol (TMP) for sender-based and receiver-based payment schemes is described in section 3. The discussion of the advantages of the TMP technology is given in section 4. Section 5 explains the architecture of the Transaction Management Service Platform, and discusses an example on how TMS is used in Bandwidth Broker Interconnection Networks. Concluding remarks and future research are in section 6.

2. Pricing and Charging between ISPs

2.1 Money Flow in the Current Internet

Looking at the money flow in the current Internet at large, it becomes obvious that a business-limiting charging infrastructure is in place. The money flow within the current Internet is hierarchical and benefits tier-1 network service providers, regardless that end-users and tier-2 Internet service providers (ISPs) can to have multiple connections to hierarchically higher network service provider. An example for this situation is shown in Figure 1.

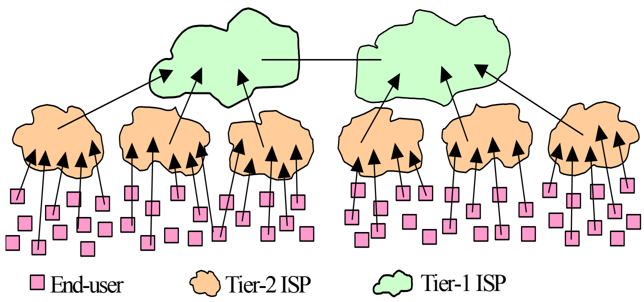

Figure 1. Money Flow in the Current ISP Market

Figure 1 shows that end-users are paying their charges to tier-2 ISPs, which, in turn, pay tier-1 ISPs (also called backbone providers) for the access to the backbone. Tier-2 ISPs are customers of backbone providers. There is no money exchange between tier-1 ISPs. They usually interconnect based on a peering agreement. The basic assumptions for a peering agreement are: the amount of traffic flowing between two peers is approximately the same; the size of the network is the same; and both peers have the same number of subscribers, etc [11].

2.2 Existing Pricing and Charging Solutions

Research on pricing considered schemes ranging from pure flat rates to pure usage-based pricing schemes [3][12]. The work in this area considers prices per network provider as well as how to calculate these prices. The underlying charging system is assumed to support any kind of pricing. In the past, the charging research has been focused on providing a feasible and scalable charging system across networks with multiple service classes [4][5][6][7][8].

In [7], the authors proposed a charging scheme for Diffserv networks. The per-unit bandwidth charge is based on the destination and the usage scheme (e.g. bulk bandwidth, per unit bandwidth). Each destination in the BGP routing table has a price associated with it. In [8], the authors propose to extend RSVP to support charging. In this scheme, the sender sends a PATH message to determine the availability of the capacity, and the returning RESV message sums the cost of each individual link on the path. Upon receiving the total cost, the sender calculates a bidding factor, which reflects the total cost of the link and the amount that the sender is willing to pay. In [6], the authors propose an auction mechanism that allows a user to bid simultaneously on multiple auctions that corresponds to multiple links on a path.

However, with the proliferation of peer, this kind of charging system is not sufficient anymore.

2.3 Enabling Bi-Directional Money Flow

In order to enable new business models on top of applications as peer-2-peer or grid technology, a charging system has to be in place that allows bi-directional money flow. Consequently, cost can be shared between communicating partners. Figure 2 shows an example between any adjacent stakeholders on the Internet. Money can flow between adjacent ISPs or adjacent ISP and end-user in both directions.

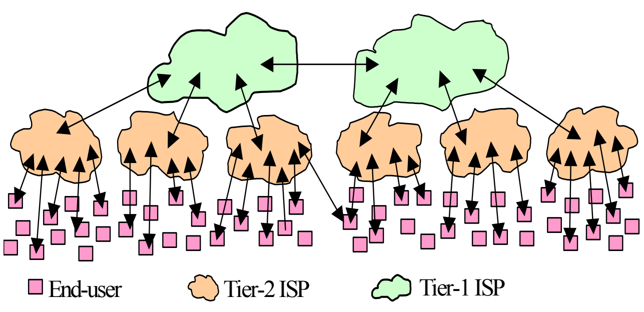

Figure 2. Money Flow in the Future ISP Market

If both communication partners agree on continuing to pay their charge for the communication, the scheme is identical to the example in Figure 1. If both communication partners agree on any other split of the cost, they simply transfer

the money to their respective tier-2 ISP. Under this scheme, the ISPs take on the role of a clearinghouse. They transfer the charges between them if necessary.
The scheme proposed requires a more sophisticated edge pricing. Each ISP must not only be capable to measure the number of incoming and outgoing bytes, but also provide a charging service, enabling money transfer to and from a ISP that is not directly connected. This aspect of the

3. Transaction Management Protocol (TMP)

Internet transaction accounting is used to denote the process of collecting, interpreting, and reporting of costs and charging information about service and resource usage. The aspect, which is considered in this paper, is the allocation of costs to communicating end-users for service / resource usage. The Transaction Management Protocol (TMP) has been developed to achieve exactly that.

3.1 Example

In the current architecture of the Internet without TMP (Figure 1), each tier-2 service provider can only charge its end-users. Considering Figure 3, it means that *ISP A* charges its *end-user A* and *ISP B* will charge its *end-user B* for the traffic. The backbone service provider will charge both (*ISP A* and *ISP B*) for the transmitted data. This limitation does not exist if TMP is applied.

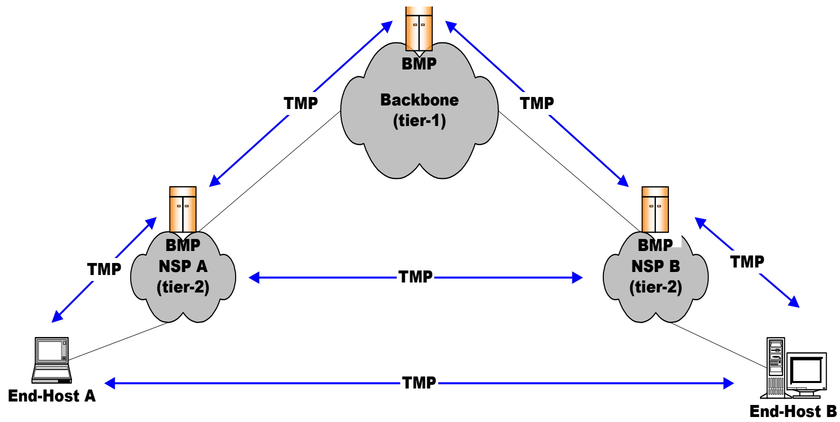

Figure 3. Example of the TMP Protocol Application between End-Users, between End-Users and Service Providers, and between Service Providers

As illustrated in Figure 3, the transaction management protocol can be applied in different situations. The significant application of TMP is between two communicating end-users (*A* and *B*). These are the parties that are being charged for consuming resources. The parties can use TMP in order to negotiate on the share of cost for the communication that they want to pay. This determines the responsibility for the incurred cost. This negotiation process can comprise several iterations and is repeated till both parties' shares add up to cover the entire cost. For example, the initiating end-user *A* sends a TMP message to the other end-user *B* with the information about the price of its connection. After receiving the price that end-user *B* has to pay, both end-users can calculate the total cost per unit and the percentage of the cost each of them have to pay (*x' %* and *y' %*) without any cost share. Based on this calculation, end-user *A* sends the percentage of the cost he is planning to cover: *x%*. *End-user B* replies to this message with another TMP message that includes the percentage that it will be responsible for. Let's assume, the reply says that he (*end-user B*) will pay for *y%* of the cost. This message exchange continues until $x + y = 100\%$.

At this point, both end-users are aware of the cost of the service and know which party is responsible for how much. Without any fraud-committing end-users, network service providers (NSP) would simply have to count the number of bytes, which they can charge their end-user for. However, in order to prevent fraud by end-users, which cooperate with each other to commit fraud, the TMP protocol has also to be applied between end-users and the tier-2 ISP. TMP is used to let the service providers know which end-user is paying how much and to whom.

Considering the example in Figure 3, after both end-users agreed on the share, each end-user informs its tier-2 service provider about the agreement using the TMP protocol. *End-user A* will tell *ISP A* "that for traffic class *G*, I will pay *x* percent of the cost and the other host will pay *y* percent of the cost". Thereafter, both network service providers verify whether the information that they received from their customer (end-user) is correct. They exchange the received information with each other to compare the information about who is paying how much to whom. If the check is negative, providers send back a *Cancel* message to the end-users. If the check is positive, they send a TMP message to their clients, allowing the cost split.

After the communication stopped, the final charge is calculated. Lets assume *x* is less than *end-users A*'s regular percentage of cost *x' %* and the total cost of the traffic is *C*. This means, *end-user B* pays *y C* to *ISP B* and *ISP B* transfers $(y - y')\ C$ to *ISP A*. In addition to this, *ISP A* gets *x C* from *end-user A*.

The same mechanism as described for two end-users can also be applied between tier-2 ISPs or backbone providers (Figure 3).

3.2 State Diagram

Figure 4 shows the finite state machine of the TMP protocol. In addition to the transitions shown in the state diagram, there is a transition from each state to the *Idle* state or *Wait Close* state, if a *Cancel* message is received or sent or a timeout for the negotiation is reached. In TMP, there are three basic message types:

- *Open / OpenAck* Message
- *Charging Query* Message (CQ)
- *Close / Cancel* Message

The *Open* message is used to initiate a session between two parties for negotiating the split of communication charges. One of the TMP parties sends the message to invite the other party. This *Open* message carries service information (e.g. price). If the other party wants to negotiate, it replies with an *OpenAck* message. When the first host receives the *OpenAck* message, the session is initiated and the parties are ready to negotiate.

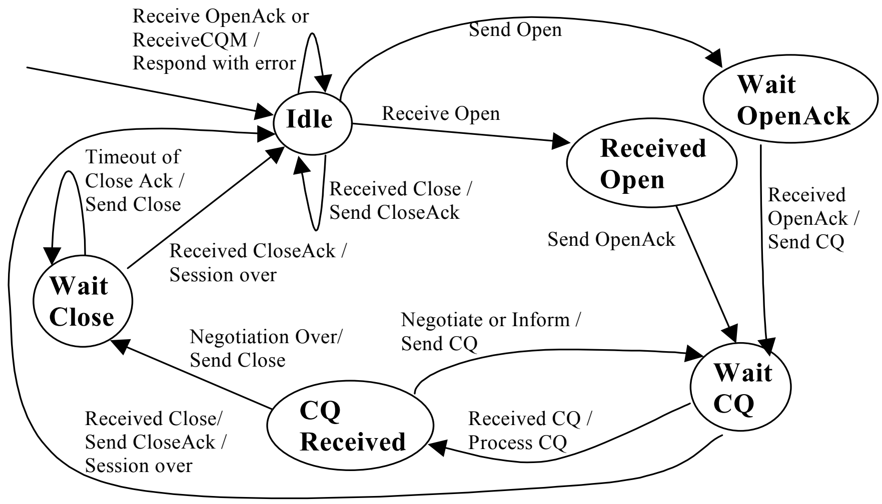

Figure 4. TMP Finite State Machine

The *Charging Query* message enables parties to exchange negotiation information. This message can be either for informational purposes or for negotiation purposes. *Information CQ* messages are used to let a third party know about the cost split between two parties. *Negotiation CQ* messages are used between two hosts to negotiate on a cost split. During the negotiation phase, the *CQ*-sending host sends the percentage of the cost per bandwidth that it is willing to pay. When the other host receives the message and the offer is not satisfactory, the other host sends its own cost split offer to the original sender. This negotiation continues until the parties agree on a cost split or they disagree and terminate the negotiation by sending a *Cancel* message to the other party. During the negotiation phase, both parties keep in the *Wait CQ* or *CQ Received* State. At the end of the communication, one party sends a *Close* message and awaits a *CloseAck* message to terminate the session.

In order to relieve users from the burden of negotiating each communication, the negotiation should be automated by using software agents acting on behalf of end-users. In any case, the negotiation is always performed in real-time. In the context of negotiation between tier-2 ISPs, an automated approach for real-time negotiation is necessary.

4. Deployment of TMP

Although the introduction of the TMP would provide more flexibility to Internet users and would allow network service provider to build new business models, the cost for introducing a new technology such as TMP is very high. Network service provider would have to make an infrastructure investment up front before they would get any benefit at all. In order to make a deployment of TMP attractive for network service provider, we will present in the next section a platform (TMS) that integrates TMP into a service environment. Such an environment also makes the management and maintenance of the system very simple.

Network service provider can benefit from TMP as the following revenue calculation shows. The idea is that network service provider can offer this TMP as a service to their customers. Whenever end-users want to split up their cost in a customized way, network service provider might charge their customers a small service fee for this service. This is a similar business model as known from the 800 and 900 telephone network services. In the US, the toll-free (i.e. 800) service market in the USA generated $17.82 billion in revenue in 2000 with a growth rate of 11.6 percent. The total US telecommunication market size was 280 billion in 2000 [10]. Based on this market data, we assume that there is the same market opportunity (6% of the total market) in the Internet. Assuming a $31 billion ISP market, information service provider (e.g. Internet retailer, application service provider, news service provider etc.) would be willing to spend $1.68 billion if they could offer 800-like services to their customers. This additional revenue would be a valuable market niche for network service providers.

5. Transaction Management Service Platform (TMS)

Since the Internet comprises a high volume of information exchange and many different kinds of transactions between ISPs and end-users, as well as between many different applications, the need for a standard on the transaction management is given. The Transaction Management Services (TMS) platform that facilitates the development of a standard can efficiently process a large volume of transactions.

5.1 TMS Architecture

The TMS platform has been implemented, using principles of Web Services [15] and the TMP protocol as the communication/negotiation protocol. In our implementation, the web service modules have been constructed as ASP.NET classes using the .NET platform.

Figure 5 shows TMS's interaction with a client using SOAP, HTTP, and XML. The transaction information is stored by the TMS in two databases (BMP TTA, TMS PCA). The TMS platform consists of three primary modules. The *XML Messaging Module* responsible for wrapping and unwrapping the messages in accordance to the implemented messaging schema. The *Database Communication Module*, which updates the existing user database by extracting information from the MIB database. The *Transaction Accounting Module*, which handles the implementation and calculations based on existing accounting policies.

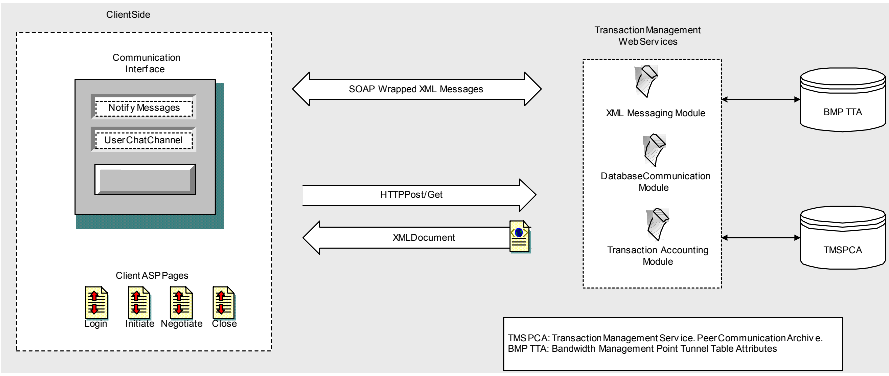

Figure 5. Transaction Management Web Services Modules

Those TMS modules interact with each other by generating and exchanging the following XML documents:

- **Messaging Schema:** The message schema enables a coordinated sequence of user / application and communication system actions in order to distribute information between end-users / end-points.
- **Accounting Policy Document:** End-users may have different transaction accounting policies implemented. TMP's accounting policy document enables each end-user to set its accounting system such that it also records the transaction data based on the accounting policy of the other end-user.
- **Communication Reference Document:** This document lists the accessibility option of each end-point. For example, it can list whether an end-user is behind a firewall, which ports are open, the capacity, etc.
- **Charge Reference Document:** Charge Reference Document contains all the variables, attributes, and formulae that are required to compute a charge. This allows an application that uses TMP to know which data has to be recorded.
- **End-User Agreement Document:** This document is exchanged between end-users in order to determine the agreement in force for a particular transaction (e.g. cancellation of services before expiration date leads to a penalty).

Changes in the contract environment (e.g. accounting policy, communication medium, charging scheme, and end-user agreements) are immediately updated using those XML documents. Changes will occur because of the nature of TMP and the dynamic of various distributed system environment.

The Transaction Management Service platform enables applications to process appropriately a large volume of messages between each other automatically. User intervention is only required when threshold values are to be set, agreements have to be modified, or changes in the accounting policy have to be implemented. This allows TMS to run as a fully automated service, which can communicate with other TMS applications for exchanging relevant data.

The interaction between end-users can be implemented in different ways. Figure 6 illustrates an example how two end-users communicate via a web site using TMS-enabled web server(s). The two end-users shown access a TMS platform web service that is owned by a third party and offered through the web sites of independent network service providers.

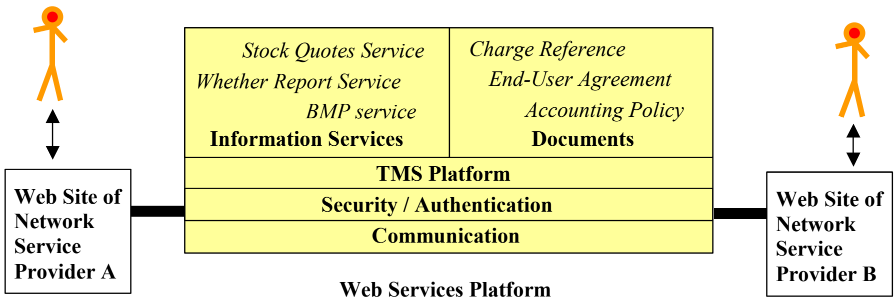

Figure 6. Parties Communicating-using TMS.

When end-users are involved in a transaction and negotiation process using TMS, they have instant access to reports on on-going transactions and accumulated charges for the communication cost (and cost for the computational power and storage) via their own web pages of the TMS system. TMS is only used for negotiation the split of costs for communication, computation, and storage, not for the cost of information services. The use of TMS enables the separation of charges for resource usage and charges for information. Without TMS, a separation of information charges and resource usage charges is not possible.
The TMS platform needs a protocol for authentication of end-users. RADIUS is such a protocol, which also provides negotiation of configuration data [13].

5.2 Example: TMS Application in Bandwidth Broker Interconnection Networks

The TMS has been tested to work on a Diffserv architecture with a bandwidth broker system called Bandwidth Management Point (BMP) [14]. Under our model, every ISP runs a Diffserv network and every Diffserv network has a BMP, which is responsible for admission control, resource control, and policy control. The BMP of each ISP keeps a Diffserv tunnel to all destination ISPs that are used by its customers. Traffic of the same QoS class to the same destination is aggregated into the same tunnel. Tunnels are established unidirectional. As part of the resource management tasks of a BMP, each BMP keeps a database of the available inter-domain tunnels with destination information and utilization level of that tunnel.
In our example, every BMP has a web presence through TMS. Every customer of a BMP has also a web presence through TMS. In order to invoke TMS, it requires users to authenticate through a web-based user client (TMS client page). Once authenticated, the TMS client contacts the TMS of the BMP of that customer and downloads the current available tunnel information and tunnel utilization

information. Then, the list of user access services is published online. To begin negotiations and transactions, TMS client communicates through pre-established BMP tunnels, using Remote Object Invocation (RMI) calls to respective TMS services. Once a negotiation is finalized, the appropriate service is made available for users. The BMP databases are MySql databases available on every TMS service domain. These databases are accessed through .NET RMI calling sequences.

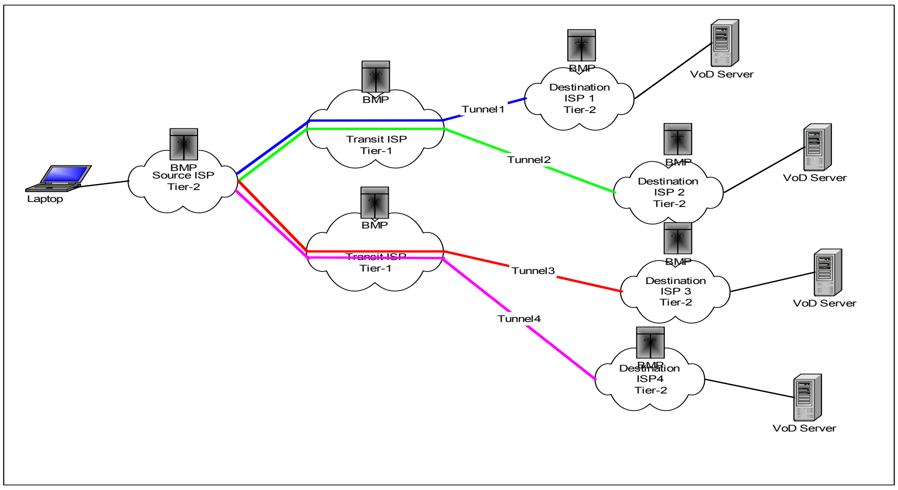

Figure 7. Example of a Video-on-Demand Network

Figure 7 shows an example of a video-on-demand network. The source ISP has a choice of four VoD servers at four different ISPs. All domain BMPs, VoD servers, and customers are logged into the TMS system, and have the capability to talk to each other through the TMP protocol. In this example, the source and destination domains can be classified as tier-2 ISP and transit domains as tier-1 ISP.

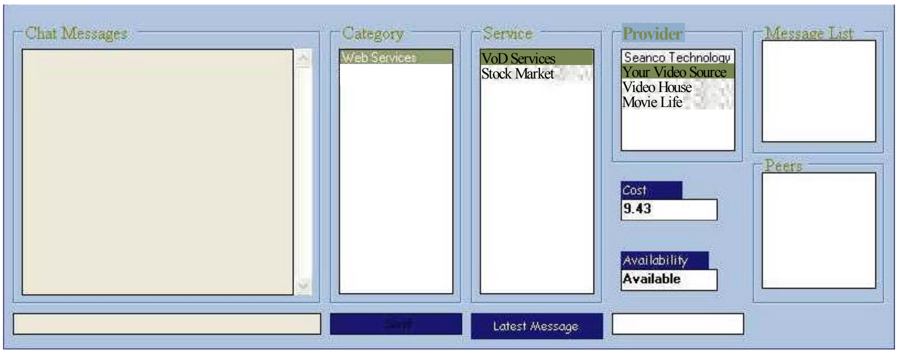

Figure 8. TMS Client Page

When a customer logs into the TMS system, he sees the available services. When the customer chooses one of the available services, all the available service providers are displayed on the web page (Figure 1). In our example, the customer is looking for VoD Services. There are 4 VoD service providers available at 4 different locations. Tunnels to those destinations have different levels of utilizations. Utilization level of the tunnel determines the unit price of bandwidth on that tunnel. If the utilization is high, unit bandwidth price is high; if the utilization is low, unit bandwidth price is low. Based on this simple pricing scheme, the customer can select the less expensive server.

In the next step, the customer and the VoD service providers negotiate the cost share of the network resources allocated to the service. If necessary, the customer can negotiate with all providers simultaneously. The negotiation process starts with an *Open* message. This message includes service specific QoS parameters, authentication information, and authorization information. The customer sends the *Open* message to the chosen provider. Upon receiving the *Open* message, the provider first checks the message for authentication and authorization information. If the customer is eligible, the provider checks whether the service QoS parameters can be accepted. If the parameters are not acceptable or if the sender is not authenticated, the receiver discards the message. If both checks succeed, the provider sends back an *OpenAck* back to the end-user to indicate that it is ready to negotiate. This message will contain similar information as the *Open* message, Information about the category of the service, the service type, the provider name, the end-users' names, the start time, the duration as well as the price (Figure 9).

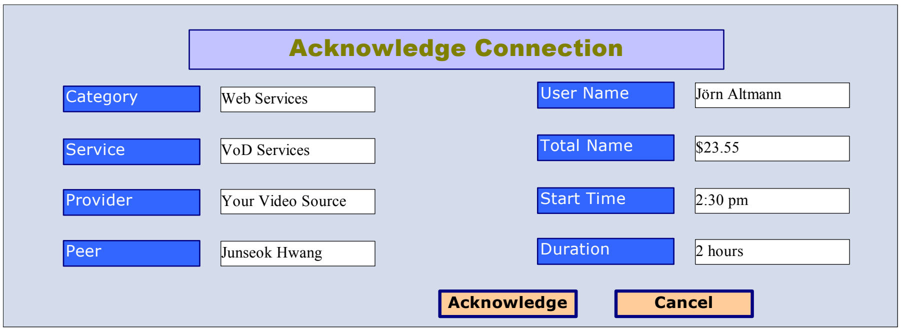

Figure 9. OpenAck Message

Both parties negotiate using *CQ* messages until the find an agreement or disconnect (section 3.2). If the negotiating entities reach an agreement on a cost share, they let the charging authorities (i.e. tier-2 ISPs) know who is paying how much. For this purpose, each end-user starts a new TMP session by sending an *Open* message to the source ISP, respectively to the destination ISP (Figure 7). Upon receiving *OpenAck* from the respective ISP, each end-host sends a *CQ* message with the informational bit set to the charging authority. The charging authority authenticates the message and updates its records according to the information embedded into the CQ message. The session will be ended with a Close message. This approach of informing the tier-2 ISPs does not require any involvement of third parties in the TMP.

5.3 Performance Issues of TMS

The important challenges of the future Internet lie in the management, sharing, and exchange of resources like bandwidth, data storage and computational power that reside across a distributed network. Any participant on the Internet will be able to buy CPU time from any other participant - just like electricity or water is bought. In theory, this is a perfect computing grid because of its flexibility and standards-based access to shared computing resources. The performance of TMS will be effective in such an environment since it is distributed system. We designed TMS as a "Web Service" application that offers a web-based interface for peers to locate other peers and negotiate service parameters (bandwidth, computational power, etc.). Under normal Internet traffic load, the test of its functionality on our bandwidth broker test network (16 PCs) showed that configuration and dynamic updates with TMP could be done in a matter of hundreds of milliseconds.

Our system is scalable since the communication overhead for setting up the charging parameters will be constant for each communication partner. Assuming usage-based charging system in place at the tier-2 ISPs, the changes to the ISP infrastructures that are required to record and account the information about cost splits are minimal. The rule sets in the accounting system simply have to be adapted. Since the negotiation of cost split will only happen for communications that require better-then-best-effort service or for p2p network applications where end-users providing free information services, the TMP communication overhead will be low.

6. Concluding Remarks

In this paper, we presented our idea of the Transaction Management Service (TMS) platform for network service providers that supports sender / receiver-payment schemes. In particular, the TMS platform offers inter-operability between accounting services and integrates with bandwidth management services across heterogeneous computing platforms. TMS enables the separation of resource usage charges and information charges, which is not possible in the current Internet. We demonstrated the effectiveness of our architecture through the implementation on a DiffServ test network.

In addition to this, we described the protocol for negotiating the cost split between end-users and / or ISPs. We presented the state diagram of the protocol and explained how this protocol can be beneficial to network service providers.

For transaction services in networked, distributed computing environments, a middleware application layer, such as TMS, would enable novel solutions to problems of directory services, QoS parameters, discovery of services and overall efficient accounting of service usage. Especially, its application in p2p computing would be beneficial for a successful deployment of p2p applications.

The research environment that we create will provide means to address several behavioral and economic questions that arise as researchers attempt to understand how service providers can establish thriving new networking services. In the short run, we will inform and influence design choices for market management of

Internet services. In the long run, we hope to formulate several economic theories for efficient market management of services available on the Internet.

7. References

[1] Internet Engineering Task Force Authentication, Authorization and Accounting working group, RFC2924, RFC2975, RFC2989, RFC3539.

[2] J.Altmann and L. Rhodes, "Dynamic Netvalue Analyzer - A Pricing Plan Modeling Tool for ISPs Using Actual Network Usage Data," IEEE WECWIS2002, International Workshop on Advance Issues of E-Commerce and Web-Based Information Systems, Newport Beach, USA, June 2002.

[3] J. Altmann, H. Daanen, H. Oliver, and A Sánchez-Beato Suárez, "How to Market-Manage a QoS Network," IEEE InfoCom2002, Conference on Computer Communications, New York, New York, USA, June 2002.

[4] A. Barmouta and R. Buyya, "GridBank: A Grid Accounting Services Architecture (GASA) for Distributed Systems Sharing and Integration," 17th International Parallel and Distributed Processing Symposium (IPDPS), Workshop on Internet Computing and E-Commerce, Nice, France, April 22-26, 2003.

[5] Ca$hman Project, European Acts Program, "Charging and Accounting Schemes in Multiservice ATM Networks."

[6] C. Courcoubetis, M. P. Dramitinos, and G. D. Stamoulis, "An Auction Mechanism for Bandwidth Allocation Over Paths," 17th International Teletraffic Congress (ITC), December 2001.

[7] G. Dermler, M. Günter, T. Braun, and B. Stiller: "Towards a Scalable System for Per-flow Charging in the Internet," Applied Telecommunication Symposium, Washington D.C., U.S.A., April 17-19, 2000.

[8] N. Foukia, D. Billard. P. Reichl, and B. Stiller, "User Behavior for a Pricing Scheme in a Multi-provider Scenario," Workshop on Internet Service Quality Economics, MIT, Cambridge, Massachusetts, U.S.A., December 2-3, 1999.

[9] E. Adar and B. A. Huberman, "Free Riding on Gnutella," First Monday, October 2000.

[10] ICBtollfree.com, white paper (article 5834), "Security, and How We Buy," 2002.

[11] W. B. Norton, "Internet Service Providers and Peering," Proceedings of NANOG 19, Albuquerque, June 2000.

[12] B. Stiller, J. Gerke, P. Reichl, P. Flury, and Hasan,"Charging Distributed Services of a Computational Grid Architecture," IEEE International Workshop on Internet QoS for the Global Computing, May 2001.

[13] C. Rigney, "RADIUS Accounting," RFC2866, April 1997.

[14] J. Hwang, "A Market-Based Model for the Bandwidth Management of Intserv-Diffserv QoS Interconnection: A Network Economic Approach," Ph.D. Thesis, University of Pittsburgh, 2000.

[15] World Wide Web Consortium, Web Services Activity, working paper, "Web Services Architecture."