AN ARCHITECTURE FOR REVERSE CHARGING IN THE INTERNET[1]

R. A. M. Sprenkels, R. Párhonyi, A. Pras, B. J. van Beijnum, B. L. de Goede

Centre for Telematics and Information Technology, University of Twente,
P.O. Box 217, 7500 AE Enschede, The Netherlands,
e-mail: {sprenkel, parhonyi, pras, beijnum, goede}@cs.utwente.nl

ABSTRACT

Charging for traffic in the Internet is gaining importance, due to the introduction of different Quality of Service classes and the increasing access speeds of end-users. Current charging schemes are limited since they do not allow Internet Service Providers (ISPs) to charge customers of other ISPs for data that is transferred to these customers. This paper presents an architecture for *reverse charging* in the Internet, that does allow such payments to be made. Reverse charging enables new business scenarios for traffic flow in the Internet.

1 INTRODUCTION

In the last decade the Internet has grown from a research network into a ubiquitous communication platform supporting new applications like video on demand, video conferencing, Internet radio, IP telephony and electronic commerce. To support these applications customers get connected to the Internet via high speed links and new techniques are introduced that allow creation of different Quality of Service (QoS) classes [1]. As a consequence of this development Internet accounting is becoming more and more important; a fact which has been recognised by the Internet Engineering Task Force (IETF) and resulted in the creation of special Working Groups in this area [2].

Currently each Internet Service Provider (ISP) uses its own scheme to charge its customers. However, current charging schemes only allow ISPs to get money from their own customers; it is not possible for ISPs to get money from receiving users who are connected via other ISPs. Still there are cases where this might be desirable. Consider for example a school that has a video server on which students put movies that they produced as part of their education, and which are interesting to a wide audience. The school does not need to receive money for these movies, but is also not willing to invest in an expensive high speed Internet connection with good QoS support. Also the school may not be prepared to go through the burden of setting up a dedicated organisation for selling the movies. As long as the current charging schemes, in which ISPs can only charge their own customers, remain unchanged, others will not be able to enjoy the movies, irrespective whether they may have been prepared to pay for the expenses.

This paper presents an architecture for *reverse charging* in the Internet. The architecture enables ISPs to charge receiving users, who are connected to another ISP, for the costs that would otherwise be paid by the sending user. Since the architecture allows to charge more than the actual costs of forwarding IP packets, the architecture can also be used to charge for the content. The reverse charging architecture presented by this paper enables therefore new Internet business scenarios.

This paper is organised as follows. Section 2 presents an overview and analysis of the State of the Art in charging schemes. Section 3 discusses the requirements for a reverse charging architecture. This architecture is presented in section 4; section 5 provides the conclusions and directions for future work.

1. This research has been performed as part of the Internet Next Generation project (*http://ing.ctit.utwente.nl/*) which is a part of the Dutch Gigaport programme (*http://www.gigaport.nl/*).

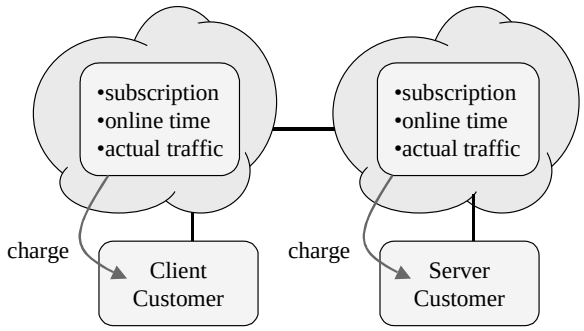

Figure 1. Example configuration

2 STATE OF THE ART AND ANALYSIS OF CHARGING SCHEMES

Charging schemes have been discussed extensively in literature. In [3], for example, charging has been structured in to a *subscription charge* and a *session charge*, that each in turn consist of a setup component and a recurring or usage component. The case where the session charge is always zero is called *flat rate charging*, the case where the session charge depends on the session duration and/or on the session volume is called *usage sensitive charging*. The price per unit duration or per unit volume can be determined statically (static pricing) or dynamically (dynamic pricing). Static pricing parameters can still depend on price modifiers like time-of-day, destination, and usage. For dynamic pricing the price depends on unpredictable, dynamic factors like the current demand for network resources or the overall network load. Two examples of dynamic pricing that are used in conjunction with the Resource Reservation Protocol [4] and Intserv [5] (auction based pricing and adaptive load-sensitive volume pricing) are discussed in [6].

What all these usage sensitive charging schemes have in common is that *customers* pay their *own ISPs* for their part of the sessions (see Figure 1). Current charging schemes do not allow for payments from a customer to an ISP other than his own.

Due to developments like the introduction of different Quality of Service classes and the increasing access speeds, there will be a need for charging schemes that allow customers to pay *remote* ISPs. The concept of a session that involves only the *customer* and *its own ISP*, as it is implicitly used in [3], therefore needs to be extended. A session has to include not only the customer and its own ISP, but also a remote ISP. This will allow the development of a usage sensitive charging scheme where a customer can pay for session costs occurring in other ISPs than its own. We will call this *reverse charging*. Note that in telephony networks this charging model is in fact the most common case. For a normal telephone call, the calling party pays all of the charges (not considering monthly subscriptions for a moment), and the called user pays nothing. The equivalent of this behaviour is what we would like to see possible in the Internet also for data traffic.

The architecture that is presented in this paper can not only be used by ISPs to charge the costs of forwarding IP packets, but can also be used to charge on behalf of content providers for content.

A different approach to charge for content, which has been described in literature [7], is called *off-line back charging*. In this approach end-users subscribe to a content service, and pay for their subscription via some off-line payment mechanism like a credit card. Since the content service provider pays its ISP for a connection with sufficient quality and capacity, it has to increase the price of its content.

While off-line back charging works in a number of cases, there are some problems with this approach that prevents it from being a generally applicable solution. One problem is that payment systems like the credit card system are not fit to handle large numbers of small payments, because the overhead per transaction is relatively high. The problem of high costs for a financial transaction has been researched extensively. This research resulted in various solutions to allow micro payments [11,12]. A micro payment is a financial transaction with a very low (fractions of a cent) overhead. This makes micro payments suitable for paying very small amounts of money.

Another problem with off-line back charging is that this approach requires every content provider to have an administrative system to keep track of subscription records, credit card details, usage logs etc. In addition, every customer needs to have a credit card or similar payment means to use the service, and every service must accept credit card payments.

3 REQUIREMENTS FOR THE ARCHITECTURE

The main feature of the reverse charging architecture that is presented in this paper is that ISPs will be able to charge customers *of other ISPs* for IP packet forwarding as well as delivery of content. An important consequence of this is that money has to be transferred from the client that receives the content to the ISP that connects the server.

3.1 Authorisation for Reverse Charging

The idea behind reverse charging is that one user pays for traffic or content that is sent by another user. To prevent misuse, it is necessary that the receiving user explicitly authorizes any reverse charged traffic that is sent to him. If a user has not authorised reverse charging, but

for some reason receives reverse charged traffic or content, the user should not have to pay. The other way around, if a user actually has authorised reverse charged traffic, it should be impossible to deny this authorisation.

The semantics of an authorisation need to be defined in terms of traffic parameters and amounts of money to be paid. Two main classes of authorisations can be identified: duration based authorisations and volume based authorisations.

- A *duration based* authorisation is for a specified charge per unit time. This means that the ISP to which the server is connected receives an amount of money that is proportional to the amount of time the client receives reverse charged traffic. It is not specified at what minimum or maximum rate the traffic will flow.
- A *volume based* authorisation is for a specified charge per number of bytes. The client authorises the server ISP to send a certain amount of traffic, and agrees to pay a specified amount. It is not specified how much time the transmission of this traffic will take.

3.2 Deployment Strategy

Reverse charging should be introduced in the Internet bit by bit. Given the current installed base of equipment and software, it is not feasible to change this installed base overnight; a coexistence strategy is therefore a strict requirement. The architecture should therefore work for any mixture of reverse charging enabled ISPs, ISPs that are not yet enabled, as well as backbone networks.

3.3 Scalability

The architecture should scale well, preferably to the same size as the Internet. This means, for example, that there should be no state information for every single reverse charged flow in every single router along the path. Such a design would suffer from the same scalability problems as the RSVP protocol [8].

When content is sent from a server to a client, the architecture should be usable even in the case where that content is free, and only end to end transport of that content is to be charged to the client. As a result, the per transaction cost should be very low; in fact it should be considerably lower than the cost of the transportation of the free content itself. Therefore, the requirement of very low transaction overhead costs as it is found in micro payment transaction systems [11,12] applies here very strongly as well.

Another scalability issue concerns the number of required trust relations between ISPs. A design where every pair of ISPs is required to have a formal bilateral business agreement before they can send reverse charged traffic, will not scale. If the number of reverse charging enabled ISPs is *n*, this would result in the order of (n^2) relations. When all ISPs support reverse charging, this number becomes too big. The number of trust relations that the architecture requires should therefore be of an order that is *less than* n^2, for example of the order (n) or ($n \log n$). Note that similar scalabilitiy requirements exists for general micro payment systems.

4 ARCHITECTURE FOR REVERSE CHARGING IN THE INTERNET

In line with the requirement to introduce reverse charging bit by bit, the architecture will only effect the two access providers at both ends of the reverse charging path. As a result, no changes are needed to the backbone networks; those networks will continue to have peering agreements with neighbouring backbone networks and with access ISPs in exactly the same way as before.

To facilitate as many new Internet business scenarios as possible (including the reverse charged payments for transport costs of free content), the architecture aims to add as little complexity as possible to both the client as well as the server. Instead, it aims at keeping as much of the inherent additional complexity inside the ISPs.

To keep the number of required trust relations low, the architecture introduces a Trusted Third Party (TTP). All ISPs that allow their customers to use reverse charging need a trust relation with this TTP.

4.1 Overview of the Architecture

The complete reverse charging architecture is given in Figure 2. The figure shows two access domains (named Client ISP and Server ISP) that each have a bilateral trust relation with the Trusted Third Party. Note that no user data is exchanged between access domains and the TTP, but only information on authorisations for reverse charged traffic.

The example from section 1 is used again to explain the architecture. In that example a client wants to retrieve a movie from a video server, and is willing to pay the server ISP for the transport of that movie (and, of course, its own ISP for the transport inside the client ISP). For the reverse charged traffic the server is the source end-system and the client is the destination end-system. The main interactions needed to obtain authorisation will be described in section 4.2. The subsequent sections describe the details plus individual components of the architecture: the client end-system (section 4.3), the server end-system (section 4.4), the client ISP and client access router (section 4.5), the server ISP and server access router (section 4.6), the Trusted Third Party (section 4.7) and the backbone domains (section 4.8).

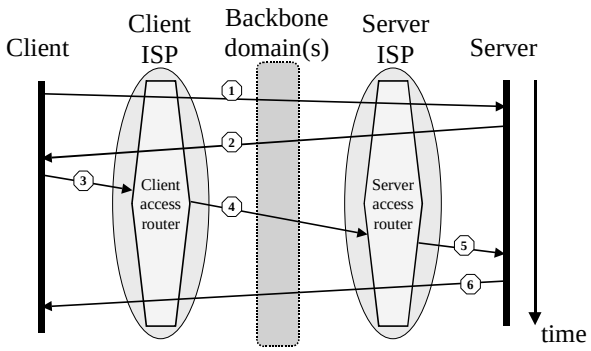

Figure 3. Time Sequence Diagram for the authorisation of Reverse Charged Traffic

4.2 Main Interactions

The main interactions to obtain an authorisation for reverse charged traffic or for charged content is shown in the time sequence diagram of Figure 3. Note that the figure does not show the interactions with the TTP; these interactions will be discussed in the subsequent sections.

1. The client sends a requests for content to server.
2. The server decides that the transport of the content should be reverse charged, or that the content should be charged. The server requests authorisation from the client.
3. The client explicitly authorizes the use of reverse charging and notifies its own access router (the client access router).
4. The client access router stores the authorisation response for bookkeeping purposes, and forwards the response to the server access router.
5. The server access router stores the authorisation response for bookkeeping purposes, updates its authorisation information so that the reverse charged traffic will be forwarded from the server to the client, and forwards the authorisation response to the server.
6. The server now knows that the client has authorised the use of reverse charging, and that both the client access router and server access router have agreed to this. The server starts sending the traffic.

4.3 The Client End-system

The client end-system takes the initiative by sending a request for some piece of content to the server. This can for instance be a HTTP request, or a request to start a video and audio stream.

After some time the client receives from the server an authorisation request for reverse charging. The server supplies the client with all the information the client needs to make a decision, see section 4.4. The request is interpreted by special software that is running on the client's system. This software performs similar functions as the 'purchasing agent' that is defined in [9]. The software can handle the authorisation request in two ways.

- The software can pop up a question on the screen that includes all the information related to the authorisation request.
- The software can be pre-configured with information that enables it to make decisions on its own,

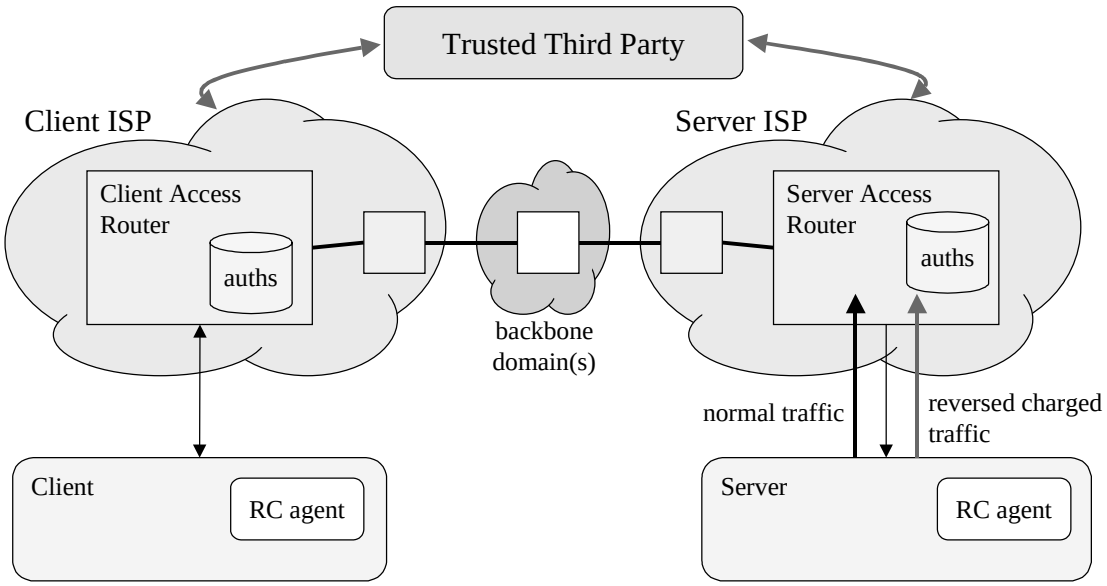

Figure 2. Reverse Charging Architecture Overview

without requiring manual input by a human user. This type of automated decision will greatly speed up the authorisation process.

4.4 The Server End-system

The server has to decide for all the traffic that it is sending if it has to be reverse charged or not. If reverse charging is used, the server sends a request to the client for authorisation. In this request the server specifies what particular kind of authorisation it wants: authorisation for reverse charged traffic, or authorisation for charged content. An authorisation request for reverse charged traffic indicates whether the authorisation will be based on volume or duration, what the price per unit will be and for what period the authorisation will be valid. An authorisation request for charged content includes a description of the content and the price of the content.

The request also contains a reference to the server access router; this reference will later be used by the client access router for sending the authorisation response back.

The server only starts sending the content when it has received a positive response to the request from the server access router. To indicate that the traffic should be reverse charged, a special DiffServ Code Point (DSCP) can be used [10].

4.5 The Client ISP and Client Access Router

The client access router receives authorisation responses from client end-systems, and stores them for bookkeeping purposes. In case the response relates to charged content, the balance of the client is updated, as well as the balance of the server ISP. In case the response relates to reverse charged traffic, the access router may additionally start monitoring the traffic stream to verify that the server ISP is charging in accordance with the actual authorisation.

The client ISP keeps records of all authorisation responses, and periodically settles via the TTP the financial balance with the other ISPs that support reverse charging.

4.6 The Server ISP and Server Access Router

The server access router receives authorisation responses from client access routers.

In case the response relates to reverse charged traffic, the server router will update its configuration to ensure that it can forward the reverse charged traffic. The configuration can have the form of traffic filters. The default behaviour of the router is to drop the reverse charged traffic *unless* there is a filter activated that specifies that the traffic is to be forwarded. The filter may be such that it checks the source and destination IP address, as well as the DSCP.

For a response that relates to charged content the server router updates the financial balances it keeps for the ISP of the client, and for the server.

When the router has processed and stored the authorisation response, it forwards the response to the server.

The server ISP as a whole maintains records that indicate how much reverse charged traffic has been handled for each of the other ISPs that support reverse charging. Periodically the server ISP settles the financial balances with these other ISPs through the TTP, and with its servers and clients.

4.7 The Trusted Third Party

Every reverse charging enabled ISP has a legal agreement with the TTP that embodies the trust relation between the ISP and the TTP. Trust is conveyed by means of trust certificates. Those certificates use cryptographic techniques to ensure authenticity and non repudiation of authorisations. The tasks of the TTP are twofold:

1. to periodically issue a trust certificate to each ISP;
2. to periodically settle the financial balance between all participating ISPs.

A client ISP includes its trust certificate with every authorisation response it sends back to a server ISP. The server ISP verifies the validity of this certificate; it will trust the client ISP to actually pay for the reverse charged traffic that is being authorized. The TTP will keep on issuing new trust certificates to an ISP as long as that ISP is meeting its payment obligations.

The TTP gets aggregated information from each ISP on the financial balances the ISP has with the other ISPs. If all is well (the balance information is consistent across all ISPs) the TTP computes the net balance for each ISP and settles this bill with every individual ISP. If the balance information is not consistent between a pair of ISPs, it is the task of the TTP to determine who has faulty information. An ISP that claims it should receive money from another ISP should have kept the records of the authorisations of the previous period to prove its claim. Because the client ISP has to cryptographically sign every authorisation response it forwards, the client ISP cannot deny having sent the authorisation response.

4.8 The Backbone domain(s)

The backbone domains of this architecture are not effected by the use of reverse charging and remain therefore totally unchanged. The normal peering agreements between adjacent backbone domains and between back-

bone and access domains keep being used. These traditional peering agreements can be symmetrical, in which case no payment from one domain to another is involved, or they can be a-symmetrical. In the latter case one of the peering domains pays money to the other for connectivity.

5 CONCLUSIONS AND FUTURE WORK

Current schemes for Internet charging only support payments from ISP customers to their own ISP. There are business cases, however, where such limited schemes are not sufficient. An example is when users are willing to pay for the costs of transporting a movie or for the costs of the movie itself. Off-line back charging may not always be a good solution to enable such business cases. This paper introduces a new architecture for reverse charging in the Internet. This architecture, which is based on the idea of a Trusted Third Party (TTP), starts from the following requirements:

- The use of reverse charging should be explicitly authorised by the receiver in advance.
- A deployment strategy for reverse charging needs to allow for a step-wise introduction into the Internet.
- The architecture has to be scalable to Internet size.

This research is closely related to the field of micro payment systems for content. When compared that field, our architecture aims to add these two features:

- Support for payments for the *transport of content only*. This in addition to possible payments for the content itself, but the content as such can be free of charge as well.
- Minimal impact on both the content server as on the content client, at the possible expense of additional complexity in the ISPs. This should keep the acceptance barrier for potential suppliers and consumers of free content as low as possible, thereby increasing the use of reverse charging, and ISP turnover.

Further study is needed to resolve the detailed engineering issues of this architecture. These issues include the design of a protocol for transporting authorisation requests and responses between clients, servers and ISP routers. A prototype of such protocol will be implemented to proof the feasibility of this reverse charging architecture.

ACKNOWLEDGEMENTS

The authors would like to thank all members of the Accounting Work Unit (WU5) of the Internet Next Generation project [13] for the stimulating discussions we had and for their contributions to this paper.

REFERENCES

[1] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss, *"Architecture for Differentiated Services"*, RFC 2475, Internet Engineering Task Force, December 1998.

[2] IETF Authentication, Authorization and Accounting working group (aaa), *http://www.ietf.org/html.charters/aaa-charter.html*

[3] F. Hartano and G. Carle, "*Policy-based Architecture for Internet Differentiated Services*", In Proceedings of IFIP Fifth International Conference on Broadband Communications (BC'99), November 1999

[4] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, "*Resource reservation protocol (rsvp) - version 1 functional specification*", RFC 2205, Internet Engineering Task Force, September 1997.

[5] IETF Integrated Services Working Group (intserv), *http://www.ietf.org/html.charters/intserv-charter.html*

[6] G. Fankhauser, B. Stiller, C. Vögtli and B. Plattner, "*Reservation-based Charging in an Integrated Services Network*", In proceedings of 4th INFORMS TELECOMMUNICATIONS Conference, Boca Raton, FL, USA, March 1998.

[7] M. S. Borella, V. Upadhyay and I. Sidhu, *"Pricing Framework for a Differential Services Internet"*, European Transactions on Telecommunications, 10(3), May 1999.

[8] A. Mankin et al., *"Resource ReSerVation Protocol (RSVP) Version 1 Applicability Statement Some Guidelines on Deployment"*, RFC 2208, Internet Engineering Task Force, September 1997.

[9] Richard J. Edell, Nick McKeown, and Pravin P. Varaiy, *"Billing users and Pricing for TCP"*, Berkeley, Technical Report #98-004P, April 1995.

[10] K. Nichols, S. Blake, F. Baker and D. Black, "*Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*", RFC 2474, Internet Engineering Task Force, December 1998.

[11] Steve Glassman, Mark Manasse, Martin Abadi, Paul Gauthier, and Patrick Sobalvarro, *"The Millicent protocol for Inexpensive Electronic Commerce"*, In *World Wide Web Journal, Fourth International World Wide Web Conference Proceedings*, pp. 603-618, O'Reilly, 1995.

[12] Ronald L. Rivest and Adi Shamir, *"PayWord and MicroMint: Two simple micropayment schemes"*, *Cryptobytes*, vol. 2, num. 1, pp. 7-11, 1996

[13] *The Internet Next Generation project, http://ing.ctit.utwente.nl/*