Achieving Good End-to-End Service Using Bill-Pay

Cristian Estan   Aditya Akella   Suman Banerjee
University of Wisconsin-Madison

1 Introduction

Over the past couple of decades, the Internet has rapidly evolved from a collaborative social experiment to an agglomerate of competing commercial providers. This shift has helped maintain growth and has turned the Internet into a vast economic force, but it has also introduced some serious problems. A particularly bad problem is the inability of end-users to obtain the desired levels of performance for their transfers.

Today, an organization can set up a contract with its ISP to ensure that the ISP offers good service to its traffic. But typical transfers in the Internet traverse multiple ISPs and it is clearly infeasible for the organization to have contracts with all of them. It is possible for neighboring ISPs to enter into contracts that require them to offer good performance to each other's "premium" traffic. However, such contracts are extremely rare and, even when used, cannot guarantee good end-to-end service to user transfers.

Our thesis in this paper is that we can support good end-to-end service to user transfers by extending the current model of binding bi-lateral contracts between neighboring entities (e.g. customers and providers or peering partners) with simple mechanisms that produce *tacit incentives for remote ISPs*. Our use of the phrase "good end-to-end service" is intentional: our goal is not to offer "end-to-end QoS" with strict performance guarantees, but rather to provide end users the *flexibility to improve* the performance experienced by their transfers, as and when desired.

Our proposal builds on two main end-user based mechanisms that generate the tacit incentives.

- **The carrot:** We propose that the end-user include in-band payments with their data. Each ISP then retains a portion of the payment, commensurate with the transit performance it offers.
- **The stick:** We propose that end-users be able to influence what path their traffic uses and thus bypass remote ISPs with unjustifiably bad service and/or those requiring unjustifiably high payments.

Our proposal, *Bill-Pay* (*Bil*ateral *l*ocal nano*Pay*ments), enables end-users to apply these mechanisms in a fine-grained manner by adding, to every individual packet: (1) a small payment which we call "nanopayment", and (2) information indicating the user's preferred path and service levels. In its basic form, *Bill-Pay* users pay according to their network usage, but *Bill-Pay* can support "flat fee" pricing for end users as well. We argue that *Bill-Pay* enables good service to end-user transfers and allows more effective protection against DDoS floods and spam.

1.1 Overview of *Bill-Pay*

We illustrate the functioning of *Bill-Pay* using the example in Figure 1. End-host A wishes to transfer data to end-host B. ISP Y along the path experiences congestion that reduces the throughput of the transfer below that desired by A.

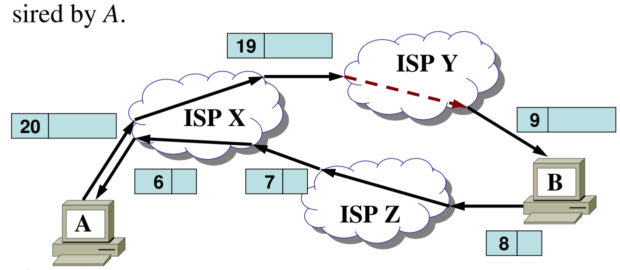

Figure 1: A uses *Bill-Pay* nanopayments to achieve priority service through congested ISP Y.

All networks and users in this example have local bilateral *Bill-Pay* agreements with their neighbors (A has an agreement with ISP X, X and Y have an agreement, etc.). To improve throughput in the face of congestion in ISP Y, A adds a nanopayment of 20 nanodollars to the data packet it sends to B. Since ISP X is not congested it leaves most of the nanopayment in the packet as it forwards it to ISP Y. ISP Y retains 10 nanodollars and gives the packet preferential treatment. B returns most of the remaining nanopayment in the acknowledgment packet it sends through ISP Z.

We note that A owes ISP X 14 nanodollars for this particular pair of packets: while A sent out 20 nanodollars, it received 6 nanodollars in the acknowledgment. In return, A's packet received good service despite the congestion in ISP Y. These nanopayment balances are converted to actual payments at the end of the billing cycle. We note that the profit made by an intermediate ISP is related to the number of packets it carries, and the level of service it offers. For example, X makes 2 nanodollars for two packets and Z makes 1 nanodollar for the one packet (both X and Z offer "regular" service) while Y makes 10 nanodollars for offering premium service to a single packet.

The viability of such an architecture depends on four important questions.

- **How can we ensure that ISPs provide improved service at fair prices?** In Section 2.1 we discuss the incentives ISPs have to limit the amount of nanopayment they retain and to provide a commensurate service quality. We also discuss various mechanisms which ISPs can use to determine how much payment to retain.
- **How can *Bill-Pay* end-users avoid expensive and congested paths?** In Section 2.2 we describe a mechanism that allows the senders to influence the trajectory of packets through the network at a granularity slightly finer than AS-level paths, with ISPs retaining control over the level of detail exposed to end-users.
- **How can the end-user ascertain how large a payment a transfer is worth?** In Section 2.3 we discuss the feasibility of a digital secretary that learns the user's preferences.
- **How can the payments be secured from malicious hackers who take control of an end-host?** In Section 2.4 we discuss mechanisms that ensure that even if the end-host is hijacked, no significant payments can be leaked without the consent of the user.

After discussing each of these issues in turn, we examine how our architecture facilitates solutions to various important problems (Section 3), how it compares to prior related proposals (Section 4), and finally conclude with a discussion of future work (Section 5). The technical report version of this paper [1] also contains a discussion of how *Bill-Pay* can interoperate with existing technologies such as diffserv and how it can be incrementally deployed.

2 Detailed discussion of *Bill-Pay*

The basic *Bill-Pay* contract is very simple and very easy to enforce: the upstream organization has to pay the downstream an amount of money equal to the total of the nanopayments in the packets it sent, and the downstream organization has no contractual obligation. Since the downstream organization has an *economic incentive* to provide good service to the packet, contractual obligations are not needed. More complex contracts linking payment to performance metrics such as loss rate and jitter clearly provide a stronger incentive for the downstream ISP to offer good service to the selected packets, but they require trustworthy measurements of the degree of compliance and the sender needs contracts with all ISPs on the path. In contrast, with *Bill-Pay* the incentives "carry over" along an end-to-end path, even without an explicit contract.

In its simplest form, *Bill-Pay* enables unidirectional nanopayments: the sender is the ultimate upstream and the origin of the nanopayment and all organizations on the path of the packet can retain a portion of the nanopayment. However, in a web browsing scenario (and in many other settings), it is common that the receiver of the packets is the one willing to pay to improve QoS (irrespective of whether the congestion is on the path to, or from, the sender). *Bill-Pay* can easily handle such a scenario with the cooperation of the server: the client sends nanopayments to the server throughout the lifetime of the TCP connection, and the server puts the remaining amount in the packets carrying content. To simplify presentation, in the rest of this paper we assume that the source is the one paying for the network traffic.

2.1 ISP Behavior and incentives

Bill-Pay can deliver benefits to end users only if most ISPs provide an appropriate level of service to packets and retain a reasonably low amount from the nanopayments. Later in this section we discuss the incentive structure which will motivate ISPs to adopt such acceptable behaviors. We start by detailing a central aspect of ISP behavior: the method used to decide the amount to retain from the nanopayment in the packet, which we call a "toll". We argue that at least two toll mechanisms are needed: congestion-based and fixed [1].

Congestion tolls on packets are to be set dynamically based on the level of congestion on links being traversed by the packets. All packets with nanopayments above the congestion toll pass and the congestion toll is deducted. The packets with nanopayments smaller than the congestion toll get dropped with probability proportional to the difference between the toll and the nanopayment. There are two important decisions to be taken at such congested links under this toll model — (i) congestion pricing: the amount of congestion toll to be retained from the packets, and (ii) congestion scheduling: the order in which packets are processed at the congested links. Both of these decisions depend on the level of congestion and the amount of nanopayments included in the packets. We note that the congestion toll can also be used to signal congestion to all senders, analogous to the way packet losses are used by TCP today.

A significant amount of past work has addressed the former issue of congestion pricing – but only in the case of a single ISP (discussed further in the related work). Congestion scheduling, in contrast, is a relatively unexplored area. We hope to address both of these challenging issues in future work.

While congestion tolls are an useful construct, this mechanism alone is not sufficient to guarantee reasonable ISP behavior, because it gives ISPs an incentive to create "fake congestion" in their networks to collect

[1] Other types of tolls such as proportional tolls (a percentage of the nanopayment) are also possible, but we do not discuss them here.

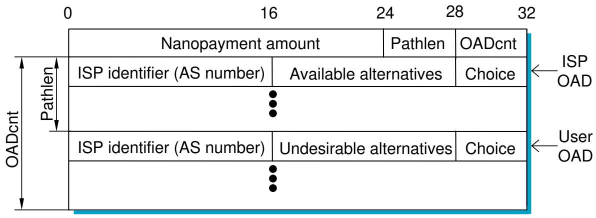

Figure 2: The *Bill-Pay* header structure

more money from packets. This motivates the case for fixed tolls, described next.

Fixed tolls are independent of the level of congestion in the network. They are a suitable mechanism to recuperate the sunken costs of running the network. Based on today's prices, these fixed tolls can be on the order of nanodollars per packet and picodollars per byte, but the amount depends on technology and on the strength of the incentives for keeping tolls low.

If ISPs also collect fixed tolls, in addition to congestion tolls, an ISP artificially inflating congestion tolls faces a loss of fixed toll revenue due to traffic that shifts to other ISPs and this acts as a deterrent for fake congestion tolls.

2.1.1 ISP incentives for acceptable behavior

It may appear that a greedy ISP receiving packets with *Bill-Pay* nanopayments could keep the money (i.e., charge a very high toll) and drop the packet. Even without nanopayments, dropping packets is cheaper than carrying them. But just as we do not see this type of near-sighted greedy behavior with today's ISPs, we expect that ISPs with *Bill-Pay* contracts will not behave in this negative fashion either. The core motivation in both cases is the *promise of future payments*.

Competition is the most important incentive for ISPs. If there is enough path diversity between the sender and the receiver and one ISP imposes unreasonable tolls, the sender can shift subsequent traffic to a different path. While single-homed users must send all their packets through the single ISP they connect to, the threat of them switching to a different ISP provides an incentive for keeping the tolls low. Note that it is not an economic or technological requirement that high speed Internet access be a choke point with high tolls. In the Utopia project [10], for example, access links are managed by a community-owned organization and the users can easily choose between many ISPs who can offer service through these access links.

Legislation can obviously limit the tolls imposed by ISPs. If extensive local monopolies for high speed network access persist, regulation is likely with or without *Bill-Pay*.

The limited willingness of the sender to pay acts as a final incentive to keep tolls lower: if tolls are too high, the sender can choose not to send traffic. Concerns about the ISP's reputation and public image (which affect long term profit prospects) can strengthen the incentive to not impose "unfair" tolls.

Without competition and regulation, the ISPs can charge tolls that amount to monopoly prices and the flexibility of the payment mechanism we propose makes this somewhat easier. While even such an expensive service could still be very valuable to users, we consider monopoly prices an undesirable outcome. Fortunately since competition between ISPs is a reality in many parts of the world where the Internet reaches and monopolies are often regulated, we believe that the basic conditions for the success of *Bill-Pay* are met.

2.2 End-user influenced paths

The incentive structure for *Bill-Pay* works best if users can ensure that their traffic avoids congested and expensive ISPs when alternate paths are available. In addition, a user may want to express other types of preferences: choice between a high latency and a high cost path within an ISP, choice between low jitter and low loss rates, etc. While conveying a limited amount of such information is possible in today's Internet, using the Type of Service field in IP packets, intermediate ISPs have no real incentive to adhere to such user indication. With our incentive-based architecture, user-influenced path and service type selection becomes viable and it can be used to achieve the desired service quality.

We describe here a mechanism that can handle the information exchange between the sender and ISPs. This mechanism uses *opaque alternative descriptors* (OADs) to represent different types of service that users desire and ISPs are able to offer. In particular, there are two kinds of OADs — ISP-OADs with information originating from the ISP and user-OADs with information originating from the sender. ISPs willing to offer choices to the users of *Bill-Pay* packets will use opaque numbers, e.g., choice 1, 2, 3 or 4, to denote the different alternatives. These choices have locally-defined semantics. ISPs mark these available choices in the corresponding bitmap from ISP-OAD fields of the packet's *Bill-Pay* header (see Figure 2). The choice field identifies which specific choice was made for this particular packet. The receiver echoes back an appropriate summary of ISP-OADs together with relevant performance measures to the source of the packet along the reverse path. The sender then uses user-OADs to convey its own preferences to the ISPs along the path. For example, if the sender considers its current path is too expensive for the desired quality, it can communicate this using the undesirable alternatives bitmap of the OAD and mark another alternative in the choice field. Note that the user uses a separate user-OAD for each ISP on the path.

We illustrate the expected operation of OADs using the example from Figure 1. The first packet from A to B would have no user-OADs indicated by A and it would acquire two ISP-OADs, from ISPs X and Y on its way to B. Let us assume that ISP X encodes its choice for this packet as alternative number 1. The acknowledgment from B to A can return these two ISP-OADs to A in the packet payload. If A considers the total toll of 14 nanodollars too high, based on its existing knowledge of topology it can ask ISP X to forward future packets through ISP Z by specifying choice number 2 in user-OADs in subsequent packets. Even if A has no prior knowledge of the topology, it can specify that choice number 1 is undesirable and learn about the alternate path with the next packet.

Note that in user-OADs, the sender expresses a preference which is not binding and any ISP is free to ignore the sender's preference altogether. But ISPs have incentives not to do so unless they have a reason to believe that the sender's choice is based on out of date or erroneous information. Also, it is quite possible that a sender does not indicate user-OADs for every ISP on its path. For example, this is the case when there are no alternatives at a given ISP, or when the sender agrees with the default choices of the ISP.

We propose concentrating the end-host's knowledge of network topology and the service quality associated with various choices exposed by ISPs into a module we call the *digital cartographer*. This digital cartographer will have a leading role in picking user-OADs to influence paths and nanopayment levels for future packets in a way that minimizes cost, but achieves the desired service quality. The cartographer's initial knowledge of the network's topology and of the meaning of various choices ISPs expose can come from descriptions published by the ISPs. To build confidence in such information and to keep up to date with changing network conditions, the cartographer would constantly monitor the acknowledgments for *Bill-Pay* traffic to measure actual service quality and toll levels. For individual home users, the digital cartographer is a service running on the end-host, but for larger organizations it makes sense to consolidate this functionality into a campus-wide service that achieves a more detailed understanding of the network by combining information from the transfers of a many end users.

The overhead imposed by OADs is small. Since not all *Bill-Pay* packets need to use them, the header size in most packets can be as small as 4 bytes. Typical AS path lengths in the Internet are 3 and 4, and most are shorter than 7, so a *Bill-Pay* header recording a typical path fits within 20 bytes. Routers can process the packets by inspecting a few fields and writing at most two (the nanopayment amount and the appropriate ISP-OAD) and they need not change the packet size. We believe that this processing can be implemented in the fast path of routers.

2.3 Accounting and authorization

Accounting of nanopayments between different organizations is easy because *Bill-Pay* agreements are local. The basic mechanism required for accounting is a pair of counters for each link connecting two organizations to track the total volume of nanopayments in the two directions. The two organizations can keep separate copies of the counters. At the end of each month or whenever the difference between the two counters reaches a certain value (say $100) the two organizations settle their accounts with actual payments.If the debtor fails to pay, the organization owed money can limit its losses without recourse to law enforcement, by providing no preferential service to future packets from the debtor.

Authorization of nanopayments should be concentrated in a trusted module on the end-host we call the *digital secretary*. Its primary task is to determine how large a nanopayment the user is willing to spend on any given transfer. A large set of initial rules about the importance a typical user assigns to various types of applications helps the digital secretary make decisions, but to build a better understanding of user preferences it requires some initial guidance from the user. We expect that over time, as the secretary learns from the user's answers, it can become sufficiently unobtrusive. The secretary can make small errors by occasionally making small "unjustified" nanopayments to avoid bothering the user with questions. The fact that the secretary does not need an exact understanding of the user's preferences and priorities makes its task more tractable. The digital secretary can also play a role in assembling a "billing statement" that summarizes for the user what he spent his money on. In an enterprise setting the end-host digital secretary would also interact with a central secretary responsible for setting enterprise-wide policies and producing enterprise-wide spending reports.

2.4 Security considerations

If malicious hackers hijack a device that can generate *Bill-Pay* packets to the outside world, they can direct nanopayments to computers they control and cause significant financial damage to the organization the device belongs to. Defenses recognizing suspicious (sudden, large, unusual) nanopayment streams and filtering them out can limit the amount of damage, but we want to disallow such fraudulent payments entirely. Hence, security mechanisms will be an integral part of our proposed architecture. While security mechanisms are needed for all network elements, such as network access devices and routers, in this section we focus on those most vulnerable — the end-hosts.

End-hosts are regularly hijacked by malicious hackers, and we expect them to remain vulnerable for the foreseeable future. Servers that never originate nanopayments, and those that mirror nanopayments to clients are relatively easy to protect by moving all nanopayment handling into trusted network devices. But clients must be able to originate nanopayments to signal to the network that certain packets are important to the user. The most obvious requirement is to secure the digital secretary and digital cartographer: the attacker should not be able to modify their code or local data, and the digital secretary should be able to interact with the user securely. We can achieve this goal by running the vulnerable operating system and applications inside a virtual machine and placing the secretary and cartographer outside it, or by moving them to a specialized secure device that interacts with the user directly. These trusted modules will need well-specified simple interfaces to interact with applications and protocols running on the end-host. There are two attack models that will gain significance in the proposed architecture.

Impersonation attacks pose a threat to the end-host because a hacker can hijack an application and use it to "impersonate" user behavior and mislead the digital secretary into authorizing unjustified payments. Such threats can be mitigated better if the digital secretary is able to discern regular user behavior from malicious ones. Appropriate research in learning techniques is therefore an important area of future work.

Man-in-the-middle attacks pose a threat because a hacker located on the path between the trusted digital secretary and the ISP can arbitrarily generate new packets or modify packets, including the destination address of packets, and their nanopayments. Such a situation can happen for example if the digital secretary runs on a USB device and the hacker controls the operating system of the end-host. We envision a solution to this problem that involves low-overhead cryptographic checksums, issued by the user's digital secretary and verified by a trusted router at the edge of the network. Similarly, packets that carry sensitive network topology and performance information in the other direction are signed by the router and verified by the secretary. While this would incur additional processing in the data path, we believe that current hardware technologies allow the implementation of such mechanisms in the fast path of enterprise, access, and edge routers.

3 Solutions based on *Bill-Pay*

Once it is adopted by enough ISPs, a network payment architecture such as *Bill-Pay* can contribute to solving many important problems. We briefly sketch one such solution below. The technical report version of this paper [1] also discusses the use of *Bill-Pay* as part of DDoS defenses and the possibility to build micro-payment protocols on top of *Bill-Pay* that can be used (among many other things) to discourage spam.

3.1 Better End-to-end Service Quality

As a first application of *Bill-Pay*, we discuss how an endpoint can achieve the desired service quality in two scenarios: improved throughput for a large transfer (e.g. an unattended download), and low loss rates and delays for time-sensitive traffic (e.g. gaming traffic). The proposed solutions have two main differences with respect to the prevailing view of QoS guarantees for traffic: the price of the transfer is variable, and we rely on active probing instead of explicit negotiation. (Of course, *Bill-Pay* does not provide tight bounds on performance.) The fact that the price of the transfer depends on current network conditions is not a problem if it falls within the amount the user is willing to pay. The extra traffic generated by *Bill-Pay* end-hosts performing active probing is not a problem to the network as they make nanopayments for the probing traffic also.

For large transfers, the overall amount of the nanopayments is likely to be a primary concern and the loss of individual packets, or even large bursts of losses can be acceptable. A reasonable strategy is to not include nanopayments in packets by default. If the performance of the transfer dips below the desired throughput, the sender can choose to add nanopayments. The sender can gradually increase the nanopayments in subsequent packets (while also trying alternate paths) until either the performance of the transfer improves above a threshold or the digital secretary indicates that the limit of the user's willingness to pay has been reached. If the tolls stay lower than the size of the nanopayment for a sustained period of time, the sender decreases the size of the nanopayments.

For time-sensitive traffic, the above strategy is not suitable because a number of important packets can be lost when sudden congestion occurs while the sender is exploring alternative paths and the size of the nanopayment to use. A sender with time-sensitive traffic can discover in advance the paths that provide an acceptable loss rate and delay through active probing with *Bill-Pay* packets done in close cooperation with the local digital cartographer. When the important time-sensitive packets are sent, the amount of the nanopayment is set large enough to get them past typical congestion events (packets may still get dropped, but the probability is much lower). The sender can even exploit the existence of multiple independent paths to increase the probability of timely delivery by sending duplicate copies of important packets along different paths.

4 Related Work

Existing contracts between ISPs either involve flat fees or employ usage-based pricing. Most contracts in the latter category use the 95th percentile traffic volume computed over all 5-minute intervals in a month to determine how much to charge. Customers pay additional amounts for QoS guarantees. Typically, these contracts are negotiated for several months at a time and the customer can re-negotiate or switch ISPs at the end of the contract period. *Bill-Pay* can be easily implemented by extending existing contracts with a clause that obliges both parties to honor the nanopayments included in the packets they exchange.

Congestion-based pricing for the Internet has been considered in simplified settings [4, 6, 7]. In MacKie-Mason and Varian's "smart market" proposal [4], users include "bids" within packets which indicate their maximum willingness to pay the ISP for access. Gibbens et. al show how smart markets can be realized in practice using simple packet marking mechanisms [2]. In Odlyzko's Paris Metro Pricing [6], an ISP network is divided into several service classes each offering best effort service but at different prices. Traffic classes with higher prices attract lesser traffic, and thus offer improved service.

In general, the above mechanisms work as long as users are vying for access from a single network provider. In contrast, *Bill-Pay* generalizes both the smart markets approach as well as Paris Metro Pricing by allowing users to place "bids" on packets traversing multiple ISPs. Moreover, *Bill-Pay* provides a payment mechanism that can be used to pay remote ISPs without a direct contract.

Micro-payment solutions such as Micali and Rivest's Peppercoin [5] use cryptographic techniques to aggregate very small payments (on the order of cents) into payments large enough to justify the fees associated with money transfers (say $10). Such schemes can be used by network endpoints to perform transactions without any special assistance from the network. Another popular solution is account-based micro-payments such as PayPal [8]. Compared to *Bill-Pay*, these solutions have the advantage of working without support from the network. However, unlike *Bill-Pay*, neither category of solutions can be used to offer fine-grained quality of service in the Internet. This is because such solutions face tremendous scalability challenges when one wants to make payments on the order of a billionth of a dollar on millisecond timescales.

5 Conclusions and Future Work

In this paper, we provided a brief description of *Bill-Pay*, a per-packet nanopayment mechanism based on local bilateral contracts. We believe that *Bill-Pay* is an effective mechanism for providing good end-to-end service between arbitrary endpoints. Furthermore, *Bill-Pay* provides a practical way to solve several other key issues, including DDoS mitigation and spam prevention.

The focus of this paper was to present a broad overview of *Bill-Pay*, its advantages and possible applications. Needless to say, we have left several interesting issues unaddressed. Throughout the paper, we outlined several major open issues; below, we mention a few more:

- What should the optimal behavior of a rational ISP be (e.g., what tolls to set, what service levels to offer)? This might depend on the ISP's topology, traffic patterns and interconnections with peers.
- A related question is how should a rational user "modulate" the size of the payments over the duration of a transfer (and how is this impacted by losses and retransmissions)? And, how do the payments interact with the user's congestion response? E.g. Does the user need to cut the congestion window in half to maintain stability even if he has included a high enough nanopayment in his packets?
- How do the tacit incentives of *Bill-Pay* influence the longer-term growth trends of the network? Will it "automatically" steer the Internet toward richer interconnections between ISPs? Note that *Bill-Pay* implicitly encourages end-users and stub networks to multihome. But will it lead to more path diversity in the rest of the network?
- Does the packet-level routing flexibility of *Bill-Pay* introduce undesirable oscillations into the network? What guidelines should ISPs and end-users adhere to when selecting routes for their traffic so as to ensure stable network operation?

References

[1] C. Estan, A. Akella, and S. Banerjee. Achieving good end-to-end service using Bill-Pay. Technical report, University of Wisconsin-Madison, November 2006. `http://www.cs.wisc.edu/~estan/publications/BillPay.html`.

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

[3] T. Karagiannis, D. Papagiannaki, and M. Faloutsos. BLINC: Multilevel traffic classification in the dark. In *SIGCOMM*, Aug. 2005.

[4] J. Mackie-Masson and H. Varian. *Public Access to the Internet*, chapter Pricing the Internet. MIT Press, 1995.

[5] S. Micali and R. L. Rivest. Micropayments revisited. In *Cryptography Track at RSA Conference*, 2002.

[6] A. M. Odlyzko. A modest proposal for preventing Internet congestion. Technical report, AT&T Research Lab, 1997.

[7] I. C. Paschalidis and J. Tsitsiklis. Congestion-Dependent Pricing of Network Services. *IEEE/ACN Transactions on Networking*, 2000.

[8] *PayPal*. `http://www.paypal.com`.

[9] M. Roughan, S. Sen, O. Spatscheck, and N. Duffield. Class-of-service mapping for QoS: A statistical signature-based approach to IP traffic classification. In *Internet Measurement Conference*, Oct. 2004.

[10] The utopia consortium. `http://www.utopianet.org/`.