AFIQ: An accounting framework for inter-domain QoS with limited trust

Cristian Estan Aditya Akella Suman Banerjee
Computer Sciences Department, University of Wisconsin-Madison
{estan,akella,suman}@cs.wisc.edu

October 3, 2007

1 Introduction

Applications sensitive to service quality, such as interactive video, could fuel the growth of the Internet if the network could support them. ISPs do provide QoS within VPNs, but endhosts connected to the public Internet and linked by paths that cross multiple ISPs can do little to influence the quality of the service for their traffic. While over-provisioning helped eliminate congestion in backbones, congestion is still present at the edges and it is not clear that if inter-domain QoS were feasible, major over-provisioning would remain the most cost-effective solution.

RFC 3869 "IAB Concerns and Recommendations Regarding Internet Research and Evolution" [AF04] identifies two main open problems that must be solved by a future inter-domain QoS architecture. "Deploying existing QoS mechanisms (...) across an inter-domain boundary creates a significant and easily exploited denial-of-service vulnerability for any network that provides inter-domain QoS support. (...) Also, current business models are not consistent with inter-domain QoS, in large part because it is impractical or impossible to authenticate the identity of the sender of would-be preferred traffic while still forwarding traffic at line-rate. Absent such an ability, it is unclear how a network operator could bill or otherwise recover costs associated with providing that preferred service." A distributed accounting system owned and operated by ISPs could solve the billing problem. We list below in rough order of importance the properties we believe such a system should have.

1. **Payments to ISPs:** It should ensure that ISPs are paid for the services they perform, based on the amount of traffic handled and on the price of the services.
2. **Flood protections:** It should work together with data plane mechanisms to identify and limit traffic that illegitimately uses improved QoS.
3. **Either end can pay:** If one of the two communicating endpoints is more eager to support the costs, it should be able to pay for all services used.
4. **Untrusted endhosts:** Users with computers compromised by malicious hackers should be allowed to participate in legitimate communication.
5. **Overhead:** The various forms of overhead imposed by the accounting system should be minimized.
6. **Discouraging cheating:** ISPs manipulating the accounting system to gain undeserved payments or to hurt others should be easy to detect.

2 Core components of *AFIQ*

We present here the core ideas of the accounting framework we propose, *AFIQ*. For simplicity, our presentation assumes that *AFIQ* will be part of a clean-slate re-design of the Internet architecture. While we believe that it is possible to retrofit *AFIQ* to work with IPv4, some changes to border routers would be required.

AFIQ, builds on our earlier proposal, *Bill-Pay*, but it goes significantly beyond it through its support for operation in the presence of untrusted (possibly hijacked) endhosts and routers and its ability to support stronger QoS guarantees. We present a more detailed comparison of the two proposals in the related work section.

We define the accounting framework as a component of the network whose task is to allow users to pay the various ISPs on the path of their traffic for services with improved QoS. Within this general definition there are many ways to delineate the exact role of the accounting system. We make the following choices: we associate payments for each ISP on the path with individual packets (not with flows or aggregates), we make payment for an ISP conditional on the packet reaching the next ISP (as opposed to it being unconditional or it depending on the packet having met some deadline), and we relax the requirement

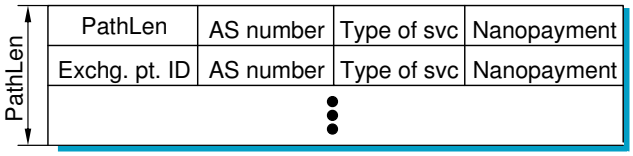

Figure 1: **The QoS accounting header** specifies the list of ISPs the sender is requesting service from, the type of service at each ISP and the amount of the nanopayment the sender is committing to. The header also specifies what exchange point the packet should use when crossing from one ISP to the next.

that payments exactly reflect the cost of the services performed by allowing small random deviations. We believe that these choices do not prevent *AFIQ* from supporting a wide variety of service models for inter-domain QoS.

The QoS accounting header (see Figure 1) is added to every packet using inter-domain QoS. This header makes it explicit for each ISP what service (as defined by QoS and the entry and exit points) the sender requests, and what price the sender commits to pay for the service. Since the per packet prices are very small, we refer to the amounts in the accounting header as *nanopayments*.

Recursive payments along the path make it unnecessary for money to exchange hands between organizations that are not directly connected and have an existing contract. The sender pays the first ISP the prices of the services of all ISPs on the path, and every ISP on the path pays the next ISP the prices of the services of all downstream ISPs. Thus the balance for each ISP is the price of the service it provided to the packet. Of course actual payments happen at the much coarser time scale of the billing cycle: the payment the sender owes its ISP at the end of the month is the sum of the nanopayments in the packets it has sent throughout the month. For ISPs that connect with multiple links on which payments flow in both directions, the amount and direction of the payment will be determined by the aggregate balance for all links.

This simplified view suggests that a simplistic accounting solution akin to SNMP counters (see Figure 2) would be enough. On inter-ISP links ISPs would need only to sum the total amount of downstream nanopayments in packets and read the counters at the end of each billing cycle. But packets can get lost and in such cases we want the source to pay only for services actually performed: the ISP dropping the packet and the downstream ISPs should not be paid. But with the simplistic accounting architecture, the ISP dropping the packet receives full payment and the payments for downstream ISPs. For the example from Figure 2, if ISP A drops the packet it makes 11 nanodollars, if it delivers it to ISP B, it makes 2. See Sections 2.1 and 3.3 for further cases in which such an incentive misalignment can have a negative effect.

Confirmation messages (see Figure 3) used by *AFIQ* enable fairness in the face of systematic or random packet losses and are key to deterring many types of manipulations of the accounting system ISPs could engage in. The first router of each ISP on the path and the last router of the final ISP generate confirmation messages that propagate on the same path as the packet, but in reverse direction. Confirmations propagate through a reliable, closed multi-ISP overlay network that is part of the accounting system and mirrors the physical interconnection patterns. The accounting header of the packet is included in the confirmation message. The confirmation's role is to signal to the accounting system that the packet made it past a certain ISP and payment is due. Payment counters are incremented as confirmation messages return, except on the outgoing link of the ISP whose service is being confirmed.

Sampling ensures the scalability of *AFIQ*: confirmations are generated only for sampled packets with each ISP making independent sampling decisions. Based on the confirmations generated from the sampled packets we get an unbiased estimate of the total volume of nanopayments an end user or an upstream ISP has to cover. As the prices of services can vary widely, we reduce the variance of these estimates by using price-based smart sampling [DLT01]. We set a small threshold expressed in currency units and all packets whose service costs more than the threshold generate confirmations, while those whose service costs less generate confirmations with probability proportional to the price of the service. Based on such samples the accounting system computes unbiased estimates of the relevant nanopayment volumes. Note that since the sampling decisions for individual packets are independent, the variance of those estimates does not depend on the timing of packets (short intense flows or long low-rate flows), but on the total volume of nanopayments.

2.1 Limited trust

Data plane filtering and rate limiting based on the accounting header are required as protection measures against hijacked, untrusted or misconfigured endhosts or networks. If hijacked endhosts or routers send illegitimate traffic using expensive services, the owner of the hijacked system is liable to pay for those services. To limit the extent of the liability, we propose that contracts contain per link filtering and rate limiting clauses that protect the upstream organization from excessive payments. For example the filtering clause may specify that no matter what traffic is sent, the upstream organization will not pay for the services of a certain untrusted remote ISP suspected

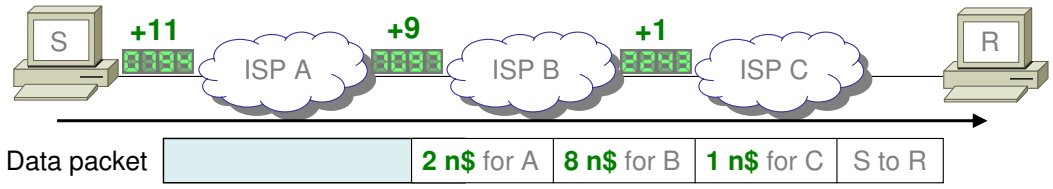

Figure 2: **A simplistic accounting system** can rely exclusively on payment counters on the external links. The counters are incremented for each packet by the total value of the downstream service for the packet.

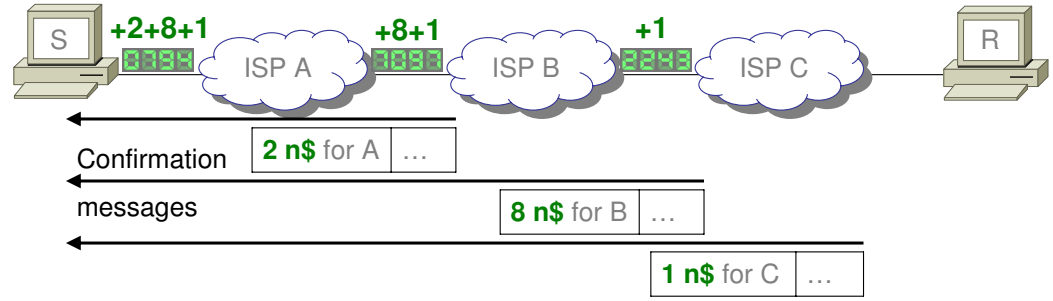

Figure 3: **Confirmation messages** are generated by the first router of each ISP on the path and the last router of the final ISP. They acknowledge the services of the previous ISP and payment counters are incremented in response to them. For scalability, confirmations are generated only for sampled packets.

of cooperating with hijackers to increase its revenue. The corresponding filter at the downstream ISP would *downgrade* all nanopayments for that ISP to 0. Contracts may also limit the total amount of downstream nanopayments that a packet of a given size may carry. A rate limiting clause may use multiple leaky bucket descriptors to define the rate at which payments can flow through a given link at various times of the day or the week. A separate rate limiting clause can limit the flow of payments at larger timescales (e.g. months). The downstream ISP should implement all filters and rate limits in the data plane of its first router, so that even if its neighbor's routers are hijacked and they send more traffic than allowed, the excess will be downgraded or dropped.

During normal operation, *AFIQ* can rely on per link payment counters (Figure 3) to determine payments at the end of the billing cycle. To allow quick settlement of disputes, the downstream ISP (the payee) should also keep a log of the confirmations submitted to the upstream entity (the payer) until at least the end of the billing cycle and the payer should periodically give the payee cryptographic acknowledgments of the confirmations as they are received. To help detect manipulations of the accounting system, the originator of any confirmation should cryptographically sign it.

2.2 Service models

AFIQ can support a number of service models for improved QoS. One such model could involve a fixed per second cost for as long as the communication channel is active coupled with a limit on the rate at which traffic can be sent. If QoS mechanisms outside the accounting system can handle the resource reservations and the data plane scheduling and the policing, *AFIQ* can handle the payments. Note that despite the fact that the actual rate at which the packets are sent may fluctuate, by modulating the size of the payments, the sender can ensure that the payments arrive at a constant rate.

AFIQ can support a service model akin to Andrew Odlyzko's Paris Metro Pricing [Odl97] without the need for any machinery beyond integrating with existing Diff-Serv functionality. ISPs can map various nanopayment levels to traffic classes with different priorities and not give any performance guarantees for the traffic in a given class. The actual performance will fluctuate according to network conditions and the users will pick the class that gives them the desired service or quit using the network if it becomes too expensive. If users can choose between many paths, this type of service model may lead to a natural load balancing of the network traffic as traffic moves from congested, expensive ISPs to cheaper, uncongested ones. While this service model has the disadvantage that the prices paid by end users may fluctuate throughout the

duration of the call, it may still prove a popular alternative if the overall cost of communication is low enough.

We do not have yet a solution that would allow *AFIQ* to support service models where ISPs are penalized for not meeting performance targets unless the end users trust ISPs to some extent.This remains an interesting direction for future research.

3 Discussion and extensions

Here we present a discussion of how *AFIQ* satisfies the requirements outlined in the introduction and introduce some extensions to *AFIQ*. We note that *AFIQ* sidesteps the difficult authentication problem described in RFC 3869 as ISPs in the middle of the network do not need to authenticate the sender of the packet. The very fact that their upstream neighbor sent the packet ensures that as long as it is within the parameters of the filtering and rate limiting clauses, the ISP (and the ones further downstream) will be compensated for their services.

3.1 Payments to ISPs

For various service models, *AFIQ* ensures that ISPs get compensated for their services. There is the risk of the upstream neighbor not paying what he owes at the end of the billing cycle, but this risk already exists with flat fee payments, and we believe that the methods that are currently used to handle the problem can be adapted. Note that the downstream always has the technical means to limit its losses and incentivize the upstream to pay its dues by downgrading all traffic to nanopayments of size 0.

3.2 Fighting floods of priority traffic

Floods of traffic with improved QoS can be particularly damaging because of the high priority treatment the flood receives in the network. As the level of priority is correlated with the cost of the service, the total of the payments carried by the flood packets within a given time is a good measure of the severity of the flood. The rate limiters deployed to enforce the safety clauses of *AFIQ* contracts can significantly dampen the magnitude of the floods. For particularly untrustworthy portions of the network the ISP may configure filters that ensure that no traffic with improved QoS originates there. The filtering and rate limiting capabilities we proposed can be used inside the network as quick reaction measures throttling floods as they happen.

AFIQ logs of confirmations can offer helpful sources of information in a post-mortem analysis after an attack where the ISP lost control of routers in its network. Even if its own logs are compromised, the downstream neighbors keep logs that can help the ISP understand the severity and characteristics of the floods.

Despite the protection measures discussed so far, end users whose computers can originate traffic with improved QoS can incur some losses if their computers are hijacked and used in DDoS attacks. Thus if further in-network anti-DoS defenses [MBF+02, YPS04, YWA05] are available, the ISPs who deploy them can use this as competitive advantage to attract customers from ISPs who do not use them. This fortunate alignment of incentives is especially encouraging for solutions that require deployment of various types of filters at ISPs close to the computers perpetrating the flood [MPR02, AC05].

3.3 Either end can pay

One party often benefits more from the communication taking place. If the network allows that party to support the cost of communication, the network receives more traffic. This is why the phone network goes beyond the default caller-pays model: users can make toll-free calls (1-800 numbers) and even toll calls (1-900 numbers). We discuss how *AFIQ* gives the same type of flexibility.

As presented so far, the sender is always the one paying for the traffic. But with *courier packets* or *boomerang packets*, either participant in two-way communication can sponsor the communication. Courier packets require stronger trust in the ISP of the non-paying participant and are akin to the sponsor sending money that the ISP at the non-paying participant can use to pay for "stamps" in the reverse direction. Boomerang packets require less trust and are akin to sending self-addressed envelopes with stamps. Courier packets require no modifications to *AFIQ*, they just require the ISP connecting the non-payer to define a *virtual service* that will be used by the sponsor to transfer money. Boomerang packets require small changes so that the accounting header will allow circular paths. When the sender sends a boomerang packet and receives a reply it will look to the accounting system like a single packet on a circular path that changed its size at the ISP of the non-paying participant and it was delayed there for slightly longer than at other ISPs. If the non-paying participant does not return the boomerang packet, the sponsor will not be charged for services on the return path. Note that these solutions do not require symmetric paths for the two directions of traffic.

It is also possible to configure *AFIQ* to implement the equivalent of toll calls where one participant makes an actual payment to an organization providing an application layer service through the network. This enables content providers to move beyond the currently dominant advertisement based and subscription based models. For example, in exchange for a micropayment of a fraction of a cent, Yahoo could offer users a version of the weather

page that does not have annoying mortgage advertisements with dancing green extraterrestrials. The ISP treats the organization providing the application layer service as a neighbor which provides a virtual service. Note that it is not necessary for the organization providing the application layer service to have an AS number of its own. The ISP can delegate to this organization the authority to generate cryptographically signed confirmations for its portion of the address space and thus both will have an accurate image of the amount the ISP owes at the end of the billing cycle. Note that if a packet with a large nanopayment is dropped before reaching the organization providing the service, the sender will not actually pay because the confirmation for the large nanopayment will never enter the accounting system. The sender can re-send after an appropriate timeout without risking to pay twice.

3.4 Untrusted endhosts

Endhosts vulnerable to hijacking need not be excluded from using services with improved QoS. As proposed by *Bill-Pay*, a trusted module not vulnerable to hijacking and with the ability of occasionally interacting with the user can act as a *digital secretary* that makes decisions about the level of service warranted for various packets. This module can be protected by running it on a separate piece of hardware or by running the vulnerable operating system of the endhost inside a virtual machine and the digital secretary outside it.

Vulnerable endhosts without such host-based protection and those belonging to users not willing to pay for improved QoS can still benefit from *AFIQ* without posing any danger. They may still send packets with improved QoS when communicating with endhosts that support both directions of the communication through courier packets or boomerang packets.

3.5 Overhead

For devices on the boundary of a domain, *AFIQ* requires the addition of four forms of processing that handle the accounting header in the data plane: filtering, rate limiting, counting payments (see Section 3.6), and sampling. Since the filters mandated by any given contract need to be deployed on a single link we expect the number of filtering rules for any individual interface to be small enough not to present significant implementation challenges. Rate limiting is similar in complexity to scheduling algorithms already implemented by routers of all speeds. Counting payments and sampling are even simpler.

The sampled packets trigger more expensive cryptographic operations. The most expensive of these is the signing of the confirmation message. When the confirmations move from one upstream ISP to the other it is possible to batch acknowledgments so the dominant processing cost is the much cheaper cryptographic checksum computation. ISPs can use the sampling rate to trade off signing overhead against accuracy. With a threshold of 100 microdollars (a hundredth of a cent), if the total amount of service for an end user is a dollar, with probability 99.9% the confirmations will show within 3.3 cents of the correct amount. If the total amount of service an ISP performs is one million dollars, with probability 99.9% the confirmations will be within $33 of the correct amount. Our measurements show that a 6-core UltraSPARC T1 (Niagara) processor can perform 512 bit RSA signing operations at a rate slightly above 28,000 per second, thus if we operated smart sampling with a threshold of 100 microdollars, the processor working at peak throughput would "pay for itself" within a few minutes[1]. Note that the rate confirmations are generated at depends on the volume of payments they confirm, not on the speed of the link. As we expect the size of most accounting headers to be under 30 bytes, simple calculations show that bandwidth and storage overheads will be less of a concern.

3.6 Discouraging cheating

ISPs can manipulate the accounting system to increase their income beyond the price of the services they perform, to reduce their competitors' income, or to make it look like their competitor is cheating. These are unlikely events since once an ISP is found to have cheated, the other ISPs (including its neighbors) can quickly cut off all nanopayment-carrying traffic through it and legal action may have severe consequences for the cheater. Thus the goal of the accounting framework is not to make it impossible to cheat, but to ensure that any cheating involving significant payments can be promptly detected and that the accounting system provides unambiguous incriminating evidence pointing to the cheater(s).

To protect against cheating we propose using both payment counters for packets (as in Figure 2) and payment counters for confirmations (as in Figure 3) and comparing them against each other to detect inconsistencies. Instead of a single counter for downstream payments, there would be a separate counter for each downstream ISP and a counter for the upstream ISP. For the counters tracking downstream ISPs, the volume of payments reflected in the confirmations should not exceed the volume in the actual traffic (it can be lower due to losses and downgrading), and for the upstream counter they should match. Due to the randomness of sampling instead of exact tests one should use statistical tests that determine whether devia-

[1] Actually the signed confirmations pay for the services of the upstream neighbor of the ISP doing the signing.

tions are significant or not. Furthermore, to ensure that these tests can be performed in a timely manner, we introduce a promptness requirement for confirmations: the delay between the packet and the confirmation cannot exceed say 500ms per downstream ISP.

An ISP can try to get a larger income than due by exploiting its two roles in which it receives payments: providing services and acting as a counduit for payments to downstream ISPs. By generating more confirmations for its own services, the ISP can increase payments to itself, but per ISP payment counters can easily detect this form of cheating and the signatures on the fraudulent confirmations will not match the keys of the valid downstream ISPs. Alternatively, the ISP can generate fake confirmations for downstream ISPs and retain the payments, but mismatching payment volumes and fake signatures can be detected upstream. A more subtle form of cheating is to downgrade the services of downstream ISPs but to manipulate the confirmations to show the original higher payment amount, and this will not lead to mismatches in upstream payment counters, but it can be detected by mismatches in signatures. Since ISPs sign (batches of) confirmations as they move from one ISP to the other, once the cheating is detected, it is possible to find which downstream ISP is the cheater. There is one form of cheating *AFIQ* cannot detect: if an ISP colludes with its downstream neighbor to increase the rate of confirmations to account for packets that were lost inside the ISP. Since this requires collusion and very tight coordination and since we expect packet losses to be infrequent we do not consider this a major limitation.

An ISP can try to reduce the income of other ISPs by exploiting its two roles in which it triggers payments to others: generating confirmations for its neighbors' services and acting as a conduit for payments to downstream ISPs. Generating fewer confirmations than due can hurt an ISP's upstream neighbors, but the neighbors can quickly limit their losses by refusing to accept traffic that flows through the cheating ISP. "Not accepting" confirmations from downstream neighbors would hurt them since they would have to pay to ISPs further downstream from which they accepted the confirmations. But there is no concept of "not accepting" confirmations in *AFIQ*, even more, ISPshave to cryptographically acknowledge (batches of) confirmations to their downstream ISPs so that if a dispute arrises they cannot repudiate confirmations.

A form of cheating specifically targeted at incriminating the cheater's neighbor is to generate the right volume of confirmations, but preferentially confirm traffic from one of its' neighbor's upstream ISPs over others. Thus that ISP upstream of the cheater's neighbor will see a larger volume of payments in the confirmations than in the traffic, but sice the cheater has to sign its confirmations, they will incriminate him. There is a variant of this attack incriminating a downstream ISP by preferentially dropping some of the confirmations while generating additional fraudulent confirmations for another portion of their traffic (or replaying legitimate confirmations). The fake signatures or the existence of duplicate confirmations can be used to trace the cheater. We do have additional scalable verification tools for detecting these and other forms of cheating and for accelerating the finding of the cheater. But since we expect them to be rarely used and since many of them would be deployed only in spots where problems are detected, we do not discuss them here.

4 Related work

The work most closely related to *AFIQ* is our earlier proposal, *Bill-Pay*. The two share the core idea of in-band recursive nanopayments associated with individual packets to obtain improved service along paths that traverse multiple ISPs. But there are critical differences that make *AFIQ* a much better accounting framework. The first one is that *AFIQ* allows the sender to explicitly partition the payment between the ISPs along the path. The second one is that payments are triggered by confirmation messages that are generated only after an ISP delivers the packet to its downstream neighbor. The third one is that *AFIQ* includes protection measures in the form of filters and rate limiters. As a consequence of these three primary differences, many incentives for misbehavior are eliminated, the damage that hijacked computers and routers can inflict is much smaller, ISPs need to place less trust in each other, and a wider variety of service models is possible.

Feldman et al. [FCSS05] discussed the use of recursive contracts to achieve good service quality along a path with multiple ISPs. Their focus is on a game-theoretical analysis of how the contracts affect ISPs' behaviors (dropping or not dropping traffic) and prices, whereas our focus is on building an accounting framework that can ensure that payments reflect the services requested by the packet.

Argyraki et al. proposed [AMCS04] an *accountability* framework for Internet traffic. Their feedback reports on individual packets are similar in some ways to the confirmation messages we propose. But since *AFIQ* solves another problem we use confirmations differently: we have distinct confirmations for each AS, we use independent sampling decisions to generate them, and we link confirmations to nanopayments.

Micro-payment solutions such as Micali and Rivest's Peppercoin [MR02] use cryptographic techniques to ag-

gregate very small payments (on the order of cents) into payments large enough to justify the fees associated with money transfers (say $10). Compared to *AFIQ*, these solutions have the advantage of working without support from the network. However, unlike with *AFIQ*, it is hard to see how such solutions could be adapted to support fine-grained service in a multi-ISP Internet. The main problem is that 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

In this paper we present *AFIQ*, an accounting framework for inter-domain QoS. We also briefly touch on aspects that are not strictly part of the accounting framework such as selecting the combination of per-ISP services that achieves the end to end service quality desired by the user, packet classification, traffic shaping and anti-DDoS defenses. Our treatment of such aspects is necessarily incomplete as the focus of this paper is the accounting framework, not the entire network architecture (and not even the entire QoS architecture).

AFIQ uses five main techniques: explicitly listing the nanopayments in an accounting header that is part of the packet, recursive payments along the path, confirmation messages generated by border routers, sampling to achieve scalability, and data plane filtering and rate limiting as protection measures against untrusted endhosts or networks. We argue that the resulting design satisfies to a great extent requirements that go beyond the primary goal of conveying payments to ISPs for their services. *AFIQ* offers assistance in identifying and fighting floods, it supports various ways of splitting the cost of communication between the participating endpoints, it allows limited participation of untrusted (potentially hijacked) computers, it has an acceptable overhead, and it offers strong support for detecting ISPs who engage in dishonest manipulation of the accounting system. We hope that our proposal will contribute to removing a critical impediment to the deployment of inter-domain QoS in the public Internet.

References

[AC05] K. Argyraki and D. Cheriton. Active internet traffic filtering: Real-time response to denial-of-service attacks. In *USENIX Technical Conference*, April 2005.

[AF04] R. Atkinson and S. Floyd. IAB concerns and recommendations regarding internet research and evolution. RFC 3869, August 2004.

[AMCS04] K. Argyraki, P. Maniatis, D. Cheriton, and S. Shenker. Providing packet obituaries. In *HotNets-III*, November 2004.

[DLT01] N. Duffield, C. Lund, and M. Thorup. Charging from sampled network usage. In *IMW*, November 2001.

[FCSS05] M. Feldman, J. Chuang, I. Stoica, and S. Shenker. Hidden-action in multi-hop routing. In *EC '05: 6th ACM conference on Electronic commerce*, 2005.

[MBF+02] R. Mahajan, S. Bellovin, S. Floyd, J. Ioannidis, V. Paxson, and S. Shenker. Controlling high bandwidth aggregates in the network. *ACM SIGCOMM CCR*, 32(3):62–73, July 2002.

[MPR02] J. Mirkovic, G. Prier, and P. Reiher. Attacking DDoS at the source. In *ICNP*, November 2002.

[MR02] S. Micali and R. L. Rivest. Micropayments revisited. In *Cryptography Track at RSA Conference*, 2002.

[Odl97] A. M. Odlyzko. A modest proposal for preventing Internet congestion. Technical report, AT&T Research Lab, 1997.

[YPS04] A. Yaar, A. Perrig, and D. Song. SIFF: A stateless internet flow filter to mitigate DDoS flooding attacks. In *IEEE Symposium on Security and Privacy*, May 2004.

[YWA05] X. Yang, D. Wetherall, and T. Anderson. A DoS-limiting network architecture. In *ACM SIGCOMM*, August 2005.