A la carte: An Economic Framework for Multi-ISP Service Quality

Cristian Estan
Aditya Akella
Suman Banerjee

Technical Report #1591

March 2007

À la carte: An Economic Framework for Multi-ISP Service Quality

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

March 2, 2007

Abstract

Internet quality of service is required by many applications such as interactive voice and video that could fuel the further growth of the network, but it is not widely available to end-users. While ISPs are providing QoS within VPNs, end-hosts 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. Our core observation is that the existing contracts between networks are an obstacle to multi-ISP service quality because remote ISPs on the path of the traffic have no incentive to provide good service when needed. We propose *À la carte*, an economic framework that addresses this problem. Among its properties are: 1) end-users can choose the ISPs their traffic flows through and the level of service within each ISP 2) end-users pay remote ISPs for their services, but need contracts only with the ISPs they connect to 3) the prices ISPs charge for various classes of traffic are static and public, thus the cost to the end-user is predictable 4) a scalable accounting framework reduces the trust one needs to place in remote ISPs and provides a detailed log of expenditures.

1 Introduction

The Internet has evolved from a collaborative social experiment to a large distributed federation of competing commercial ISPs. While this transformation has fueled the network's growth and has turned the Internet into a vast economic force, it has not helped end-users achieve end-to-end service quality[1] for their traffic *across multi-ISP paths*.

End-to-end service quality is quite desirable to many users in the Internet for a large set of applications, including but not limited to interactive voice, video conferencing, telepresence, and distributed games. While these applications are supported on the Internet today, the experiences of users at different locations and times vary widely. To address this issue, ISPs offer enterprises Virtual Private Networks (VPNs) with service guarantees in the form of Service Level Agreements (SLAs). Applications sensitive to service quality are supported over such VPNs, but all endpoints have to be on the network of a single organization — a far cry from the near universal reach that helped email and the web rival telephone and television as top communication conduits.

To enable the Internet to support applications sensitive to service quality, a number of technical solutions such as IntServ [ZDE+93] and DiffServ [Wro98, dif] have been developed. In particular, DiffServ is widely supported by network equipment and used by ISPs to offer service guarantees within their networks. But when the traffic crosses from the user's home ISP to subsequent ISPs on the path, the service quality does not always match the needs of the end-users. In particular, there is no mechanism available today through which end-users can influence service quality for their traffic at remote networks.

To achieve end-to-end service quality in an Internet with multiple ISPs, the end users need the collaboration of remote ISPs. But in the current Internet, remote ISPs have neither the contractual obligation nor the economic incentive to offer quality service to traffic between users who are not their customers. Therefore, in our search for a solution we consider both technical and economic facets of the problem: contracts between end-users and networks, inter-ISP traffic accounting, measurements of service quality, packet scheduling, end-host route control, and end-host congestion control. Our goal is to find an economic framework for service quality in multi-ISP networks that gives all the stakeholders incentives to take

[1] We intentionally use the term "service quality" instead of the more specific Quality of Service (QoS) in this paper, since the latter is usually associated with stronger guarantees. In contrast, our objective is to provide end-users the *flexibility to improve* the performance experienced by their traffic, as and when desired.

technologically feasible steps that will allow end-users to obtain the service quality they need. In this paper, we primarily report on our detailed thought experiment on one such framework, which we call *À la carte*, shaped by many discussions between authors, external colleagues, and the exposition and discussion of a recent related idea, called *Bill-Pay* [EAB06], which we consider the precursor of the framework presented in this paper.

1.1 Problem description

The goal of the economic framework we present is to enable *any pair of nodes in a future Internet to achieve the service quality they desire for the traffic they exchange*. We note that the network must have the ability to deny service if it does not have the resources to accomodate the traffic or if the resources it has are used to carry other, more important traffic. Furthermore, the network may require a payment to provide the service quality the end-users want.

While in the spirit of recent clean-slate network design initiatives we assume that changes to network protocols and contracts are possible, we present here some factors that constrain and shape the design of any economic framework that aims to achieve our goal.

- **Multiple ISPs** will exist and communication between some endpoints will cross ISP boundaries. While it is true that consolidation is taking place and the number of tier-1 ISPs is decreasing, we expect that differences in business model, differences in customer demographics, differences in technology, cultural and legislative differences among countries, and anti-monopoly regulation will ensure that a large number of ISPs will continue to exist.

- **Congestion** will occur within and between networks (though not necessarily frequently). A viable solution for end-to-end service quality must be sensitive to congestion on end-to-end paths. While careful provisioning of the network together with an understanding of usage patterns can make congestion uncommon, there will be economic pressures on all ISPs to keep their costs down, so congestion will not be fully eliminated.

- **Scalability** of the functionality required from the network is important. If routers need to keep per-flow state, if contracts impose unreasonable accounting burden, or if heavyweight performance monitoring is required, the cost of equipment might increase significantly. Therefore we favor solutions that lend themselves to lightweight and scalable implementations.

- **Security** becomes paramount when payments are used as economic incentives: systems with the authority to originate payments and those holding accounting-related information present an attractive target for attackers. While it is likely that any solution that allows payments will increase the attack surface, we prefer solutions where the number of vulnerable systems and the types of attacks they are exposed to are kept at a minimum. Furthermore, the framework should ensure that there is little opportunity for the stakeholders to cheat by manipulating the accounting and monitoring infrastructure in ways that benefit them financially.

- **Bi-lateral contracts** already exist between neighboring networks. We espouse this form of contracts mainly due to their scalability and simplicity. Frameworks requiring contracts between entities that are not directly connected to each other have two problems: they often require heavy-weight monitoring to track compliance with the contract and the number of contracts a network must enter into is often large. Trusted third parties (e.g. PayPal) can enable transactions between entitites that don't have a contract, but solutions that work even without a trusted third party have an even better chance of adoption.

2 *À la carte*

With the framework we propose, ISPs publish a menu of prices for various classes of service, and the end-users choose which ISPs and service classes to use (hence the name *À la carte*) to achieve the desired end to end service. End-users control the service they receive at the level of individual packets. Conceptually end-users pay all the ISPs on the path of the traffic, but the actual payments are only to the ISP(s) they connect to. These ISPs forward some of the payment to downstream ISPs as appropriate. The accounting system used by our framework ensures that ISPs are paid based on the number of packets they deliver to the "next hop" ISP and based on the service class selected by the sender. The accounting system does not enforce that the service quality matches that advertised by the ISP for the given class, but end-users can very quickly switch to other paths if the service quality is below what they expect.

À la carte does not require any change in the way best-effort packets are handled. Packets that require better service will specify the equivalent of an ISP-level loose

Term	Meaning	Section
Path descriptor	A field listing the ISPs the packet travels through and the service classes at each ISP	2
Service class	A specific set of scheduling policies applied to a set of packets from various users	2.1
Price list	A public document listing for each ISP the available service classes and their prices	2.1
Confirmation	Message generated by ISPs on the path and the destination to confirm the service of the previous ISP – the entire accounting system is based on these messages	2.2
Value of a confirmation	The size of the payment triggered in by a confirmation message	2.2
Inspector	A device that compares confirmations against a log of actual traffic to detect fraud	2.3
Digital cartographer	An end-host module keeping knowledge of network topology and traffic conditions	2.4
Digital secretary	An end-host module tracking the user's willingness to pay for various types of traffic	2.4
Sponsor	The end-user willing to pay for improving service quality (can be sender or receiver)	2.5
Boomerang packets	Packets that can be used when the receiver is the sponsor	2.5
Reverse service	A service class at the sender that can be used when the receiver is the sponsor	2.5
External service provider	An organization connected to the network that provides some service	2.6

Table 1: **Summary of *À la carte* terms:** We summarize here the terms we use in this section for the most important concepts of the *À la carte* framework. The last column lists the section where the given concept is defined.

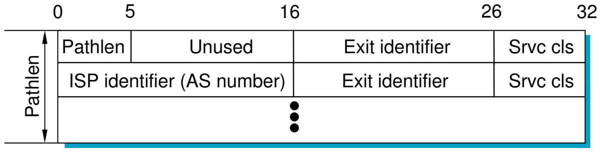

Figure 1: **The basic *À la carte* path descriptor:** For each ISP whose services the sender wants to use, the path descriptor specifies the service class and the identifier of the exit point from the ISP (e.g. which peering point to use).

source route that we call the *path descriptor* of the packet. Figure 1 shows the structure of the path descriptor. While the role of some of the fields will become clear only later, the most important elements of the path descriptor are straightforward: the pathlen field specifies the number of ISPs on the path and all ISPs other than the one the sender is sending through are listed as 16 bit AS numbers. The list also contains for each ISP the service class the sender requests as a 6-bit codepoint, and a 10-bit identifier for the exit point from the ISP where the packet should be handed over to the next ISP. Based on the path information, one can look up in the price lists published by the ISPs on the path how much money the sender owes each ISP for the successful delivery of the packet to the next hop, but it is not necessary that one perform such lookup in the data plane. As an input to the accounting system, all ISPs on the path will generate confirmation messages that acknowledge that a given ISP delivered the packet to the next hop, but for scalability reasons, these confirmation messages are generated only for sampled packets.

2.1 Prices and service classes

To achieve the cooperation of remote ISPs, *À la carte* gives them a financial incentive to offer good service to the packets requesting it. The ISPs offer multiple *service classes* and the user pick at the level of individual packets the service class the packet should be mapped to. All packets within a given service class receive the same treatment. The ISPs must publish *price lists* for the service classes they offer, the lists should specify the price for servicing a single packet. Based on current prices we expect the price for handling a single packet to be on the order of nanodollars. These prices are meant to be fairly long term commitments from the ISPs. For example the ISPs may be required to announce any price change a month in advance of it taking effect. This contributes to a certain stability and predictability for the prices end-users pay for various network services, and significantly simplifies the distribution of price information. The prices of servicing a packet may depend on a number of factors besides the service class: the size of the packet, the points where the packet enters and exits the ISP's network, the time of the day, the day of the week and the day of the year. The price list published by the ISP may also include a description of the service offered to the various classes of traffic, but these descriptions are not used by the accounting system.

We envision two different strategies ISPs can use with *À la carte* to provide improved end to end service quality to end-hosts that require it: a guarantee-centric approach and a guarantee-less approach. The guarantee-centric approach builds on the methods ISPs use today to ensure that they deliver the QoS guarantees they commit to in their SLAs: careful provisioning of their network coupled with policing at the edges and/or an understanding of traffic patterns. The guarantee-less approach is based on the Andrew Odlyzko's Paris Metro Pricing [Odl97] and it does not provide strong QoS guarantees, but it relies on the end-users' sensitivity to prices to ensure that good service quality is possible (if a given user picks an expensive enough service class, the amount of traffic in that class will be small enough to allow quality service). These two approaches can coexist in the *À la carte* framework, and we can leave it to the market to decide the types of settings in which each is preferable.

The guarantee-centric strategy has the advantage that the burden of ensuring service quality rests entirely with the ISPs and the end-hosts need not perform any network probing or measurement. The problem with this method is that it is difficult for the ISPs to fulfill their guarantees. While the problem is hard even in the single-ISP case, it becomes even harder when the traffic can originate in other ISPs. Also, there is a level of predictability of the traffic matrix for any given ISP which makes provisioning and dimensioning easier, but the traffic patterns that *À la carte* would lead to may be more dynamic, making the problem harder. The *À la carte* framework does not require the existence of infrastructure for measuring to what degree ISPs comply with the service quality they advertise for the various classes of traffic, but ISPs that want to increase the level of trust the users place in their QoS guarantees may deploy such infrastructure voluntarily.

The guarantee-less strategy has the advantage is that it works even with unpredictable dynamic traffic. The ISP can define a number of progressively more expensive classes of traffic that receive strict priority service in the network (prices can still depend on time of day or network path). If the network is uncongested the users will use the cheapest class and receive good service. As the network is more congested, users will start using higher and higher classes to achieve the level of service they want. Since high prices discourage users to send and encourage them to find alternate cheaper paths if possible, the actual traffic will decrease, reducing congestion. The main disadvantage of this method is that it places on end-hosts the burden of monitoring network performance and exploring alternate paths and various service classes. Such a strategy can lead to good service quality, but is different from the guaranteed quality of service typically addressed by the QoS literature.

2.2 Scalable accounting

The goal of the accounting system is to track payments due to ISPs for packets successfully delivered to the next hop on the path descriptor and to provide to end-users and ISPs and audit trail of expenditures. The accounting system is built around the concept of *confirmation messages* generated by the receiver and the first router in each ISP that acknowledge that *the previous ISP* has delivered the traffic. The use of aggressive sampling ensures that the number of confirmation messages[2] is low enough so that cryptographic operations such as signing are feasible. A confirmation message indicates to the accounting infrastructure of each ISP that the ISP whose service it acknowledges needs to be paid by the source of the packet for its services. Since the payment needs to be transferred through the same chain of ISPs as those who carried the traffic, confirmations must propagate to the sender through the same ISPs that carried the packet (but in reverse order). While it may seem unusual that *À la carte* confirmations acknowledge only the services of the last ISP, and not those of the ISPs before it, with this arrangement ISPs depend on their neighbors and not on untrusted end-hosts for confirmations that lead to payments for their services. Furthermore, such of confirmations make it easier to achieve scalability through independent sampling.

To ensure the scalability of the *À la carte* accounting infrastructure, confirmations are generated only for sampled packets. While this sampling leads to errors, if the sampling rate is high enough, these errors will be negligible. For example, if the price of certain level of service offered by one ISP is 1 nanodollar per packet and a sampling rate of one in ten thousand is used, a user sending ten billion packets will receive on the order of a million confirmations, each indicating that the user must pay ten thousand nanodollars (a thousandth of a cent). The user will pay close to 10 dollars for the network service: the probability that the amount is off by more than 3 cents in either direction is less than 0.2%. For a router processing small 40-byte packets at line rate on a 10Gbps link, a sampling rate of one in ten thousand means that is has to generate only 3,125 confirmations per second. We note here that not all routers need to generate confirmations, only the first router in each ISP. We use the term *the value of a confirmation* to denote the size of the payment triggered by a confirmation message which is equal to the price of the service it acknowledges times the inverse of the sampling rate applied when confirmations are generated.

If there are big differences in the price of servicing various packets using a sampling rate that is proportional to

[2] The function of confirmation messages is similar to that of acknowledgments, but there are a number of important differences: they are generated only for sampled packets, they are also generated by ISPs on the path, and they are processed by intermediate ISPs on their way back to the source. Because of these differences we use the name "confirmation" and not "acknowledgment".

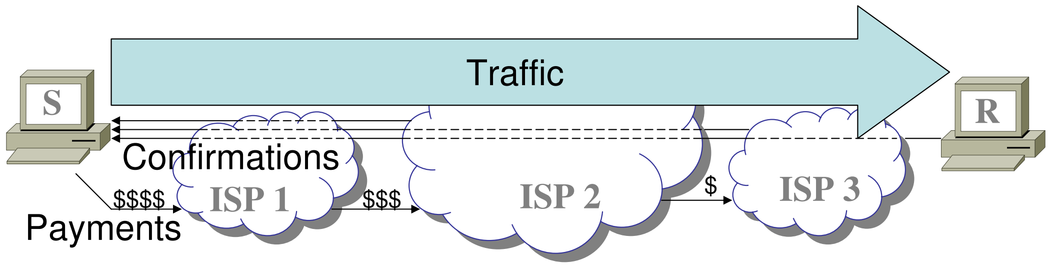

Figure 2: **The *À la carte* accounting system:** For sampled packets, the receiver R and ISPs 2 ad 3 generate confirmation messages that acknowledge the service of the previous ISP. Based on these confirmations, the sender S pays ISP 1 who transfers some of the payment to ISP2, who transfers some of the payment to ISP3. The amount each ISP keeps is determined by the number of confirmations from the next hop, the type of packets they refer to, and the public price list each ISP announces.

the price of the packet reduces the variance of the payments [DLT01]. Applying this technique directly requires the border router to look up in the neighbor's price list the exact price associated with the current packet. If this operation threatens to become a performance bottleneck because the exact price depends on too many factors, the sampling rate can still be set based on just the most important of these factors (e.g. time of day/week and service class) in ways that reduce the variance of the payments. As in the case of a fixed sampling rate, the actual sampling rate applied to the packet that generated the confirmation must be included in the confirmation so that the accounting infrastructure can compute the correct payments.

The *À la carte* framework allows upstream ISPs to perform further sampling of the confirmation messages, but the additional sampling should be recorded in the confirmation message by updating the sampling rate of the confirmations that are not discarded. We expect that this will not become necessary and guidelines about acceptable sampling rates will emerge (e.g. on average generate a confirmation for every thousandth of a cent worth of service) all ISPs will follow them. Since confirmation messages travel through the accounting system of each ISP we assume that, unlike TCP acknowledgments, they cannot get lost unless there are massive equipment failures that also affect the traffic. We also place a timeliness constraint on the confirmation messages to make it possible for ISPs on the path to the sender to verify it by comparing it to in-memory logs of traffic. A simple way to define this timeliness requirement is to give a generous time budget of say 100 ms for each ISP on the path, which could easily fit the propagation delays and the time required to process the confirmations.

Assuming that the guidelines for sampling rates suggested in the previous paragraph are applied, it is easy to estimate the accounting burden on an ISP that is due to payments that only transit it towards downstream ISPs: each packet imposes a small burden (in expectation) that is proportional to the total price of the services of downstream ISPs. To allow ISPs to recuperate the cost of this accounting burden, classes of traffic can impose caps on the amount of downstream payments a packet can carry, and packets with payments above these thresholds may have to use more expensive classes. Alternatively, we could allow the prices of the service for packets to also depend on the total price paid to downstream ISPs.

The accounting system of each ISP needs to store the confirmation messages so that they can be used by the billing system to generate the appropriate payments and as an audit trail. For end-users, it is easy to compute the payment they owe at the end of the month: it is the sum of the values represented by the confirmations they received (after compensating for the effect of sampling). The operation is similarly simple for two ISPs that connect through multiple links: after computing the sum of the values of *all* confirmations sent in the two directions, the accounting system computes the size of the payment as the difference between the two sums. The direction of the payment is from the ISP who received more confirmations (weighted by their value) to the other. Note that if the size of the payment accumulated by the end of the billing cycle is a concern, other arrangements are possible: for example the ISP may charge the user's credit card whenever the amount reaches some agreed upon limit (say a hundred dollars). The *À la carte* accounting system allows ISPs to track balances in real time.

The fact that prices can depend on time does not pose a problem to the accounting system as long as it is unambiguously defined which of the relevant events (the time the packet was sent, the time the packet entered the ISP providing the service, the time the confirmation was received by the sender) determines the price. We propose to use the time the confirmation was generated as the time determining the price of the service. Thus the router generating the confirmation can add a timestamp to it and it is clear to everyone what price applies. This means that a sender can not be sure what cost a packet will incur (even in expectation), as delays along the path may cause it to arrive somewhat later and be charged a different price. To minimize this variability we propose that prices should not change fast. For example instead of suddenly changing the price from 10 nanodollars to 1 nanodollar at 7PM, one should gradually decrement the price by 1 nanodollar every 10 seconds for the first 100 seconds of the hour.

2.3 Trust, cheating and attacks

À la carte builds on the existing trust relationships between organizations that are connected by a network link. No direct payments are exchanged between organizations that do not have a connection. Such organizations are already in a contractual relationship, so *À la carte* does not require new contracts, just amendments to the existing ones. But since actual money is involved, it is important to ensure that the framework does not allow ISPs to collect payments for services they have not performed or end-users to use services they don't pay for.

One undesirable scenario is that of an end-user who refuses to pay the bill to her ISP at the end of the billing cycle. Note that the contracts the ISP has with its neighbors still obligate it to pay for the services of other ISPs used by the end-user. But this is not a new problem as end-users may refuse to pay their flat fees in today's Internet, so the current solutions to this problem can be used with *À la carte*. While *À la carte* leads to variable, usage-based payments this does not pose challenge as phone and other utility bills are also variable.

A situation with more severe consequences for the credibility of the proposed framework is if ISPs are able to collect payments for services they have not performed. Since the payments an ISP receives are triggered by confirmations generated and signed by its neighbors, the ISP cannot just generate fraudulent confirmations and collect the payments. We note here that if the receiver is a large organization with a long-term presence on the Internet, their signature on a confirmation message can bear as much weight as that of a neighboring ISP. However, for recipients that are individual end-users with a dynamically assigned network address, tracking the associated public keys and the times they are valid for might impose too large of a burden. For such scenarios it is acceptable for the last hop ISP to generate confirmation messages for its own services at their last router. While this allows the ISP to generate fraudulent confirmations for its own services, the solutions we discuss below that handle the case of colluding neighbors apply to this scenario also.

Without further protective mechanisms, by colluding with their next hop neighbor, ISPs can still reap the benefits for fraudulent confirmations. To guard against such behaviors we add to our framework network devices called *inspectors* whose role is to detect when ISPs engage in such activities. *À la carte* works under the assumption that most ISPs are honest, and the role of the inspectors is not to ensure that every single confirmation is legitimate. Their role is to ensure that schemes relying on fraudulent confirmations are detected and honest ISPs can cease to cooperate with the cheaters and, if possible, trigger punitive action by law enforcement agencies. Inspectors need not be deployed universally throughout the network, but just on selected links. We defer the discussion of how the confirmation inspectors can be implemented scalably to Section 2.3.1, here we only describe the functionality they perform.

A simplest form of fraudulent confirmations is confirmations for packets that the ISP did not actually carry. A variant of this form of cheating is confirmation messages that pretend that the class of service the ISP gave to the packet is a more expensive one than the one the packet actually requested. To defend against these types of cheating, the inspectors keep an in-memory log of the relevant fields of (some of the) packets, and match the confirmation stream against this log. To facilitate this type of check, *À la carte* requires the senders to add to the packets they send a unique identifier send. Uniqueness is required only within packets using the same path descriptor and going to the same destination. Also, after a time on the order of a few seconds, identifiers may repeat. We only impose a weak uniqueness constraint. Packets with the same identifier will not have a serious impact on the operation of the inspectors, and the effect of repeated identifiers is that it is harder to detect fraud against those packets. The operation the inspector has to perform is simple: when it receives the confirmation message, it checks to see if the corresponding packet is in its log, and if so, to see whether the service class matches.

A more refined form of cheating is to confirm actual packets, but to do so more often than claimed in the confirmations. If the confirmation claims that one in ten thousand packets are sampled, but in reality one in a thousand packets are, the ISP can collect ten times the price of the services offered. To defend against this, the inspectors also need to track the aggregate amount of payment to each ISP carried in the actual traffic and compare it against the aggregate amount requested through the confirmation messages. While the randomness of sampling can lead to mismatches between the two quantities, standard statistical tests can be applied to determine whether a difference is conclusive indication of cheating.

ISPs have dual roles: to provide services and to forward payments to downstream ISPs providing services. The previous two forms of cheating tried to exploit the first function, the next form of cheating tries to exploit the second: the cheating ISP can reduce the service classes for downstream ISPs to cheaper classes and retain the difference between the price the sender pays for the original classes and the price of the reduced service classes. But for this to work the cheating ISP should also change the confirmation messages so that they reflect the original class of service, not the one that was in the packets at the time they reached the downstream ISP whose service is being confirmed. Since confirmation messages are cryptographically signed by the originator, such modifications can be detected, thus this type of cheating is not feasible.

Another type of cheating is based on ISPs colluding with hackers who control zombies that can send *À la carte* traffic. The ISP can advertise an expensive class of service and have zombies send traffic that it services at those unrealistically high prices. While the accounting framework cannot determine whether prices are unrealistically high and traffic fraudulent, it provides good support for combating such behavior. Once ISPs notice such behavior, they can immediately stop further losses to their clients by filtering out traffic sent to the expensive classes of the dishonest ISP, or even disallowing altogether *À la carte* packets with that ISP in their path descriptor. If the contracts and the legal framework support it, the honest ISPs can even "undo" old payments to the dishonest ISP, since the accounting system preserves the confirmations, providing an audit log that can be used to determine the exact amounts paid to the dishonest ISP.

It is also useful to look at how *À la carte* affects another undesirable activity zombies are used for: network flooding attacks. If the zombies are flooding with best-effort traffic, legitimate clients can switch to higher priority classes of traffic available through *À la carte* and bypass the effects of the flood at a cost that we expect to be small. If zombies flood with high priority classes of *À la carte* traffic, they increase the cost to legitimate users of the service attacked, and cause financial losses to the actual owners of the zombies. While this situation is undesirable, it presents three advantages over how the current Internet behaves in such cases. Firstly, the legitimate users willing to pay the increased price, can get access to the service under attack. Secondly, the high priority traffic sent by the zombies may be a very reliable sign that the ISP the zombies connect to can use to detect that an attack is under way and apply countermeasures (e.g. filtering) to protect the victim from the attack and its customer from the expenditure. Finally, if such measures fail and the owner of the zombie must pay, this gives him a financial incentive to secure his computer against intrusions.

Finally ISPs must guard against malicious behavior that does not benefit anyone, but it denies them the payments they deserve for their services or damages their reputation by planting incriminating indications that they engage in cheating. The two types of damages correspond to the case when the router of a neighboring ISP generates too few or too many confirmation messages. It is irrelevant for the rest of the discussion whether the router is under the control of the neighbor, or under the control of a malicious hacker, we focus our attention on how the *À la carte* accounting system can help the ISP detect and counter these behaviors. The ISP's router can check locally at the link level whether the total volume of confirmations matches the value of services it delivered. If the volume is too low, it may be an indication of packet drops on the link or in the neighbors router, so a logical first step is to investigate in conjunction with the neighbor whether the reason is a technical problem. If the ISP is not content with the way the problem is resolved it can stop accepting *À la carte* traffic routed through its untrustworthy neighbor and thus avoid carrying traffic it would not get properly compensated for. If the volume of confirmations is too large, the ISP only needs to drop some until the volume matches the price of services it delivers. Of course, it is appropriate to escalate the issue to the management of the neighbor. Finally if the neighbor generates confirmations for inexistent packets, as long the volume of confirmations is right, this evidence will incriminate the neighbor, not the ISP whose services it confirms.

2.3.1 Scalable confirmation inspectors

The confirmation inspectors have two roles: to detect if the volume of confirmations for any downstream ISP significantly exceeds the value of the services justified by traffic, and to detect ISPs that send confirmation messages for packets that were not in the traffic (or packets that had different service classes). To achieve their role, inspectors keep logs of traffic and compare them against the confirmation stream. These logs need not keep packet content, just the packet's unique identifier, the source and destination addresses, and the information relevant to the prices of service: the path descriptor and the packet size. To limit the amount of time these logs need to be kept for, *À* *la carte* introduces a timeliness requirement for the confirmations. For example the constraint might be that the confirmation needs to reach the ISP within an amount of time proportional to the number of hops in the path descriptor to the ISP generating the confirmation. The time budget can be generous (say 200 ms per hop) to ensure that occasional delays do not cause legitimate confirmations to be flagged as fraudulent.

Even with limits on the size of the records and the lengths for which they must be stored, for high speed links they can push the limits of the available memory sizes and more importantly updating the log for every packet would become a performance bottleneck. Since the role of the inspectors is to detect cheating eventually, not to detect every single fraudulent confirmation message, an acceptable way of handling the scaling is by logging only sampled packets. But while this allows the detection of anomalously large volumes of confirmations it makes it hard to detect promptly when ISPs generate confirmations that do not correspond to actual packets. Therefore, instead of random sampling, we use hash based sampling [DG00] where the sampling decisions are based on a hash of invariant fields present both in the packet and the confirmation, and if the hash falls in a certain range, the packet is logged, otherwise it is ignored. The hash function is seeded with a random number known only to the inspector. If there is a confirmation whose hash value falls within the range used to select packets, but the corresponding packet is not in the log, it is certain that the confirmation is fraudulent.

2.4 End-host route control

End-hosts or campus-level servers will have the task of building the path descriptors. This task can be separated in two components: a *digital cartographer* whose role is to keep knowledge about the topology of the network and the current path conditions, and a *digital secretary* that uses some knowledge of the user's priorities to decide whether the price required to achieve the desired level of service is within the user's willingness to pay.

The digital cartographer will have a leading role in building the path descriptors for packets in a way that minimizes cost, but achieves the desired service quality. The cartographer's initial knowledge of the network's topology will come from the price lists published by ISPs. This information would be augmented by constant monitoring of the actual service quality experienced by user traffic. 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 large number of end users.

The digital secretary's 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.

2.5 Service quality on the reverse path

So far our discussion of *À la carte* assumed that the sender is the one willing to pay for ensuring that the packets receive an appropriate level of service. However it is possible that a higher level of service is required for the packets flowing towards the end-user willing to pay for good service quality whom we call the *sponsor* of the traffic. For example a person browsing a web site experiencing slow download times may be willing to sponsor the traffic coming from the web server so that it receives good service. Here we present two ways in which *À la carte* supports such scenarios. The first solution requires that the sponsor trusts only minimally the opposite end and it is analogous to it sending a self-addressed envelope with a stamp to be used for the return traffic[3]. The second solution requires slightly stronger trust in the opposite end and it is analogous to sending money the opposite end-host can use at is sees fit to buy stamps for the return traffic. We note here that for both solutions, the "opposite end" managing service quality related tasks can be either the actual end-host or its ISP that can act as a proxy on its behalf.

The first solution requires the introduction of *boomerang packets*. These are sent by the sponsor

[3] Unlike in the case of a stamped envelope, if the opposite end does not send return traffic, the sponsor is not be charged for the "unused stamp" on the return envelope.

of the traffic and have a loop-shaped path descriptor: a first part of the path descriptor gets them to the opposite end, and a second part of the descriptor gets them from the opposite end back to the sponsor. When a boomerang packet arrives to the opposite end, it is not immediately returned to the sponsor, but it waits for the end-host at the opposite end to generate some data to send, and the boomerang is sent back when such data is available (but the packet identifier is not changed). Just like with normal *À la carte* packets, the sponsor will be charged only for the portion of the path that the boomerang actually traversed. Specifically if the opposite end does not send it back to the sponsor, the sponsor will not be charged for the return portion of the path. Just as with normal packets, confirmations follow the exact reverse path of the traffic. Besides obvious simple modifications to the accounting infrastructure and to confirmation inspectors due to the fact the the path descriptor of boomerang packets has a slightly different format and the source and destination addresses are switched on the return path, there are two changes required to make boomerang packets more useful. The first regards the timeliness constraints of the confirmations: for boomerang packets on their path from the sponsor to the opposite end confirmations from the return path should be accepted for a longer time interval (say an extra 5 seconds) than warranted by the length of the path descriptor. This allows the opposite end to keep the boomerang for a short time (up to 5 seconds) before it expires and it can no longer be used to send data from the opposite end to the sponsor. A second change regards the size of the packet. We do expect that on the return path the boomerang may be of a different size. In the web browsing example above, the sponsor may send a small request and expect a large document in return). Therefore, boomerang packets on the forward path also specify a maximum size for the packet on the return path that can be much larger than the size of the packet on the forward path.

A second way of solving the problem of service quality for traffic sent from the opposite end towards the sponsor is to use *reverse service* classes. These are high-priced service classes available at the opposite end used to "transfer money" to be used by the opposite end to pay for service quality for packets sent towards the sponsor. When the sponsor sends a first packet using a reverse service class, the opposite end establishes a budget initialized to the price of the reverse service. This budget is used to support service quality in the opposite direction: whenever a packet is sent towards the sponsor the total price of the service it receives from all ISPs on the path is subtracted from the budget, whenever a reverse service packet arrives from the sponsor, the budget is increased accordingly. By keeping the budget positive throughout the exchange the opposite end can ensure that overall it will not be paying for the traffic[4]. The use of reverse service classes is more flexible than the use of boomerang messages, since a single message from the sponsor can be used to support a large number of packets in the reverse direction and the balance of the budget can persist for much longer than the boomerang messages since it places a burden only on the opposite end and not on the infrastructures of all ISPs on the path. But this flexibility comes at the cost of increased trust the sponsor has to place in the opposite end.

2.5.1 Internet value flows

One of the criticisms of the current economic framework of the Internet is that is enforces rigid value flows: the end-users pay the cost of reaching backbone network. In contrast the telephone network supports caller-pays arrangements (by default) toll free numbers (1-800 numbers in the U.S.) where the callee supports the entire cost of the conversation, as opposed to the caller paying, and hybrid models where if the callee uses a mobile phone, the caller pays part of the cost, but the callee may also pay for the "air time". While *À la carte* does not affect the direction of value flows for best-effort traffic, for *À la carte* traffic either end can support the cost of the communication. This flexibility allows arrangements not possible in the current Internet. For example a web site (say a banking site) that wants to ensure good service even when its customers access it from an often-congested low cost network, can do so with *À la carte*. Also if a small site hosted in a large data center gets popular and the traffic exceeds the traffic budget its owner has paid for, a special class of traffic paid for by visitors through *À la carte* could still be used to ensure that those interested can visit it.

2.6 Micropayments beyond the network

While the goal of the *À la carte* framework is to support service quality in a multi-ISP Internet, the fact that with *À la carte* the ISPs act as conduit for payments can be used for another purpose: micropayments to remote end-users (not just to remote service providers). End-points

[4] This is not a strict guarantee since the randomness introduced by the sampling used to generate confirmation messages can lead to situations where the opposite end pays a small amount at billing. But actually the system is biased in favor of the opposite end as the remaining budget after the conversation ends is to its advantage and because it does not have to pay the full price for the service of packets in the reverse direction that get dropped in the network.

that wish to receive micropayments will join the accounting system as an *external service provider* that provides non-network services and announce special services similar to the reverse service class described in Section 2.5. A packet sent to such a special class of an external service provider could represent a payment of say one cent, or one dollar. While such an use will likely fall under a different regulatory framework, and extensions to the *À la carte* accounting framework may be required we briefly discuss it here because it can enable new uses for the network and it can turn into a significant new source of revenue for ISPs. We discuss here a few security considerations and three possible applications for network micropayments.

We expect that due to security concerns, micropayment ability will never become as prevalent as network connectivity. The ability of hijacked micropayment-enabled devices to send payments to the attackers will act as a deterrent that ensures that only the most secure computers and devices will be authorized to send payments to external service providers. But we expect that cryptographic methods may ensure that trusted devices will be able to authorize payments that would travel through untrusted devices. For example if a user trusts his personal digital assistant (PDA), but not his desktop computer, he could still send micropayments through the computer as long as the PDA signs the packets with a signature recognized by the first hop ISP and appropriate measures are taken to avoid replay attacks.

Pay per page web would enable new business models for the publishers of electronic content currently limited to advertisement-sponsored or subscription-based solutions. For example an unaffiliated writer of a popular blog could turn his passion into a full-time job by charging say a cent per view and without annoying his users with intrusive or distracting advertisements.

Spam could never compete with legitimate emails carrying a cent to certify that they are worth the recipient's attention. Since very few spam messages result in a purchase by the recipient [Min06], the only reason spammers can still make a profit is that it is very cheap to send email messages. While linking micropayments to email messages is not a new idea (spam defense services such as "Bonded Sender" [bon] implemented this idea by requiring organizations sending large amounts of email to escrow money), *À la carte* micropayments can provide a convenient way to implement it. Note that messages without micropayments could still reach the recipient if they passed white-lists and various aggressive spam filtering solutions.

Direct payments between mobile devices may prove an appealing alternative to some uses of cash. Paying through say one's cell phone instead of cash could provide a widely accepted payment mechanism for cheap items (e.g. newspapers, some meals and food items, etc.) where the overhead of credit cards is too high, yet an audit trail is desired.

3 Related Work

Quality of Service has been recognized as an important capability missing from the original Internet architecture. The intserv line of research [ZDE$^+$93] proposed a resource reservation protocol, RSVP, that allows end-hosts to communicate to the network their QoS-related resource requirements. Diffserv [Wro98, dif] is a newer proposal that defines packet marking and scheduling operations capable of providing service quality guarantees in conjunction with network provisioning and policing of the traffic admitted to the network. Both RSVP and diffserv are widely supported by current routers. A significant volume of research has looked at (dynamic) provisioning and pricing for differentiated network services built on top of various network technologies [LV93,DaS00,SLCL01,WS01].

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 and service level agreements (SLAs) describe the terms of the agreement between the customer and the ISP. Such contracts never apply to the service quality the traffic receives outside the contracting ISP's network and solutions based on DiffServ [Wro98, dif] are typically used to implement QoS inside the ISP's network. Typically, 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.

Congestion-based pricing for the Internet has been considered in simplified settings [MMV95, PT00, Odl97]. In MacKie-Mason and Varian's "smart market" proposal [MMV95], 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 [GK99]. In Odlyzko's Paris Metro Pricing [Odl97], an ISP network is divided into several service classes each offering best effort service but at different prices. Traffic classes with higher prices attract less traffic, and thus offer improved service. These papers assume a single ISP. In contrast, we focus on the more realistic scenario where packets traverse multiple ISPs and our goal is to build an economic framework that supports payments to remote

ISPs without a direct contract.

In recent work, Argyraki et. al. [AMCS04] proposed the notion of packet obituaries — on packet losses, ASes are required to send reports to prior AS hops and sources the location of this loss. Such a mechanism is an useful to end-hosts in better selecting paths. However, obituaries do not provide any information on how different ISPs differentially treated successfully delivered packets — a mechanism critical to implementing our desired objective of end-to-end service quality.

Micro-payment solutions such as Micali and Rivest's Peppercoin [MR02] 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 [pay]. Compared to *À la carte*, these solutions have the advantage of working without support from the network. However, unlike *À la carte*, 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.

Finally, the ideas proposed in this paper, build on the core idea developed in *Bill-Pay* [EAB06] and extends it in the following four aspects: (i) unlike *Bill-Pay*, *À la carte* requires each ISP to publish a *menu* of prices for different services that simplifies the AS path selection problem at end-hosts. (ii) *À la carte* provides a lightweight accounting system implemented through simple cryptographic operations, on a tiny fraction of packets chosen through aggressive sampling. (iii) the sender has controls how the payment is distributed among the ISPs on the path. (iv) ISPs get paid only for the packets they deliver to the next network in the path. We believe that these differences may simplify end-host route selection and may facilitate adoption.

4 Conclusions and future work

The economic framework we propose, *À la carte*, aims to enable multi-ISP service quality by giving ISPs financial incentives to provide good service quality to the packets that need it. All the data plane technologies it relies on are widely supported by current routers. The required accounting infrastructure poses no scalability challenges.

While the thought experiment presented in this paper is promising, further work is required to decide whether it can fulfill its potential. Challenges remain in demonstrating that the digital secretary and the digital cartographer can be accurate enough. It is also necessary to explore how the changes in incentives affect ISPs' strategies for provisioning their networks and setting their prices.

References

[AMCS04] K. Argyraki, P. Maniatis, D. Cheriton, and S. Shenker. Providing packet obituaries, 2004.

[bon] Bonded sender. http://www.senderscorecertified.com/.

[DaS00] L. DaSilva. Pricing for qos-enabled networks: a survey, 2000.

[DG00] N. G. Duffield and M. Grossglauser. Trajectory sampling for direct traffic observation. In *Proceedings of the ACM SIGCOMM*, pages 271–282, August 2000.

[dif] Differentiated services. `http://www.ietf.org/html.charters/diffserv-charter.html`.

[DLT01] Nick Duffield, Carsten Lund, and Mikkel Thorup. Charging from sampled network usage. In *SIGCOMM Internet Measurement Workshop*, November 2001.

[EAB06] Cristian Estan, Aditya Akella, and Suman Banerjee. Achieving good end-to-end service using Bill-Pay. In *ACM HotNets-V*, Irvine, CA, December 2006.

[GK99] R. J. Gibbens and F. P. Kelly. Resource pricing and the evolution of congestion control. *Automatica*, 35:1969–1985, 1999.

[LV93] Steven H. Low and Pravin P. Varaiya. A new approach to service provisioning in ATM networks. *IEEE/ACM Transactions on Networking*, 1(5):547–553, 1993.

[Min06] Alex Mindlin. Seems somebody is clicking on that spam. The New York Times, July 3 2006.

[MMV95] J. Mackie-Mason and H. Varian. *Public Access to the Internet*, chapter Pricing the Internet. MIT Press, 1995.

[MR02] Silvio Micali and Ron 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.

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

[PT00] I. Ch. Paschalidis and J.N. Tsitsiklis. Congestion-Dependent Pricing of Network Services. *IEEE/ACN Transactions on Networking*, 2000.

[SLCL01] N. Semret, R. Liao, A. Campbell, and A. Lazar. Pricing, provisioning and peering: Dynamic markets for differentiated internet services and implications for network interconnections, 2001.

[Wro98] J. Wroclawski. Differential Service for the Internet. http://diffserv.lcs.mit.edu, March 1998.

[WS01] Xin Wang and Henning Schulzrinne. Pricing network resources for adaptive applications in a differentiated services network. In *INFOCOM*, pages 943–952, 2001.

[ZDE$^+$93] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala. Rsvp: A new resource reservation protocol. In *IEEE Network*, pages 8–19. September 1993.