Service Models and Pricing Policies for an Integrated Services Internet

Scott Shenker

Palo Alto Research Center

Xerox Corporation

3333 Coyote Hill Road

Palo Alto, California 94304-1314

shenker@parc.xerox.com

Abstract

This paper addresses the integration of services in the Internet and the resulting impact on pricing policies. I first address why an integrated services Internet is desirable, and give an overview of the services it is likely to offer. I then argue that an integrated services Internet, in order to be efficient, must employ per-user, quality-of-service sensitive, and usage-based pricing policies.

1 Introduction

In the next five years, the Internet will undergo significant technical changes. These will probably include dramatic increases in bandwidth[1] on the backbone transmission links, better physical access from homes and businesses, and a more sophisticated network architecture. Internet policies are also likely to change; these policy changes will probably include allowing more public access, increasing privatization of service provision, reduced or at least modified government subsidies, and new pricing schemes. These policy and technical changes will reinforce each other: some forthcoming technical developments will enable or force the Internet community to contemplate new policy options; some policy choices will dictate certain technical design choices.

Many current workstations and personal computers can transmit and receive audio and video signals. As a result, multimedia teleconferencing applications and other forms of remote multimedia communications are becoming increasingly common. In recent years, many researchers have been investigating ways to provide integrated services in packet networks; integrated services is essentially the ability to carry video and voice as well as data. This integration is not just a matter of increased speed; instead, integration will require a major change in the basic network architecture. This paper introduces policy makers to the technical aspects of integrated services, and explores the impact this technical development will have on pricing policies. This paper does not address the technology, economics, or policies of the Internet in a comprehensive manner, but rather focuses on the specific topic of integrated services.[2]

This paper has four sections. Section 2 explains why the integration of services requires a new network architecture; in essence, this is because simultaneously delivering adequate service to

[1] Bandwidth is the amount of data a link can move per unit time, and is the usual measure of capacity of a link.

[2] References [6, 18, 20, 29, 30] provide more general treatments.

video, audio, and data in the current Internet architecture would require a prohibitive amount of bandwidth. Section 2 also describes the service requirements of various applications such as video, voice, and data, and then briefly sketches an appropriate service model. Section 3 discusses how the integration of services, combined with increased public access and privatization, will affect pricing policies. In particular, I argue that an integrated services Internet must employ per-user, quality-of-service sensitive, and usage-based pricing policies. Section 4 discusses some important steps along the path to the future integrated services Internet.

2 Integration of Services

2.1 Background

Since 1985, the speed of the Internet's backbone links[3] have increased by roughly three orders of magnitude (from 56kbps to 45mbps), and in the next decade they are likely to increase by another two orders of magnitude. Similarly, since 1985 the number of sites and hosts have increased by several orders of magnitude (from roughly 50 and 1000 respectively to over 10,000 and 1,000,000 respectively). Moreover, in the next decade many if not most homes will likely have some access to the Internet or an equivalent network. This astounding growth in both speed and size was achieved without changing the basic network architecture. The Internet's basic network architecture, as embodied in the underlying TCP/IP [24, 25] protocols, has remained virtually unchanged since its inception; this is a powerful testimonial to the robustness of the original design. However, a recent flurry of research activity has focused on building *integrated services packet networks*,[4] or ISPN's. Such networks represent

[3]The figures cited in this paragraph and the next were taken from Reference [30]. We adopt their terminology, in which Internet refers to Arpanet but not to Milnet.

[4]The term "integrated services packet networks" is not completely standard.

a major departure from the basic network architecture currently used in the Internet.[5]

The search for a new network architecture is driven by the emergence of a new generation of of computer-based applications that make extensive use of the network. These applications include multimedia teleconferencing, remote video, computer-based telephony, computer-based fax, telemetry, remote visualization, virtual reality, and many others. It is widely expected that many future remote business, scientific, and social interactions will utilize such tools to enhance the quality of communication and to reduce the need for co-location. These applications require very different *Qualities of Service* (QoS) from the network.[6] For instance, a casual telephone conversation requires relatively little bandwidth, and can tolerate occasional dropped packets[7], but is rather intolerant of network-induced delay.[8] A video connection used for remote control of an experiment requires a relatively high bandwidth rate, cannot tolerate any dropped packets, and also needs low delay. A video broadcast of a lecture requires a relatively high bandwidth rate, but can tolerate a few dropped packets and some delay. Bulk data transfers, such as computer-based fax, file transfer, and electronic mail, do not have an intrinsic bandwidth rate; they use as much bandwidth as is available to minimize the transfer time. These bulk data transfer applications can tolerate high delays for individual packets, but are sensitive to the loss of many packets. The current

[5]For a discussion of such research efforts, see [1, 7, 8, 9, 11, 12, 13, 14, 15, 16, 17, 19, 21, 22, 26, 28, 31, 33, 34].

[6]It is perhaps useful to clarify that networks such as the Internet are *packet-switched*, in that transmissions of data from one site to another are broken up into many small packets of data and each packet is transmitted separately. The network switches (also called routers or gateways) determine the route each packet travels to its final destination, where the packets are reassembled into the original data. The bandwidth requirement of an application is determined by the rate at which these packets are transmitted (and their size). The delay referred to below is the time taken by these packets to traverse the network.

[7]Human speech is quite redundant and human conversation has a built-in recovery mechanism, so occasional dropped packets have little ill effect.

[8]We will discuss the delay requirements of such applications in great detail in Section 2.2.

Internet architecture cannot efficiently support this emerging generation of applications, in that it cannot simultaneously meet all of their service needs unless the network is extremely overprovisioned (i.e., the average utilization is quite low and so all applications receive good service).[9]

Our current telecommunications infrastructure handles different applications on separate networks; the telephone network carries voice and fax traffic; the cable TV network carries broadcast video; the Internet and other similar data networks carry data traffic. As computers begin supporting many video, voice, and fax applications, separate networks might be developed to support these different applications; however, a network offering an integrated set of services seems more attractive. First, it would be extremely awkward to implement multimedia applications across several different networks. Second, it would be expensive to wire buildings for several different networks. These two problems could be resolved by combining the networks where they enter the office or home. A more fundamental advantage of a single integrated services network is that it uses bandwidth more efficiently[10] than a collection of separate networks. Segregating network traffic by application type leads to a substantial loss of efficiency; a simple example at the end of this section illustrates this point.

Thus we have a formidable technical challenge: to build a single network architecture that can meet the service needs of the emerging generation of applications. To support these applications the network must accommodate extreme variations in delay and bandwidth requirements. Packet-switching, as used in the Internet, is widely regarded as the technology of choice to meet this challenge (see Chapter ?? [MacKie-Mason and Varian] for an intro-

[9]Such inefficiency is only a problem if bandwidth continues to be the scarce commodity in networks. Some, such as in [10], maintain that all-optical networks will soon make bandwidth plentiful and switching resources scarce. I disagree and think that bandwidth will continue to be scarce, but this paper is not the place to debate this point.

[10]Efficiency is measured by the total application performance achievable on the network for a given amount of bandwidth; see Section 2.3 for a more precise definition of efficiency.

ductory presentation).[11] Its principle advantage is that it allows statistical multiplexing to occur on a packet-by-packet level, and thus wastes no bandwidth.[12] The Internet, and most other data networks, employ a first-in-first-out (FIFO) packet scheduling algorithm; that is, the first packet to arrive at a network switch is the first one sent. Such scheduling algorithms cannot provide different service to different clients, such as low delay to one client and high delay to another. Thus, to provide a variety of QoS, the network must employ nontrivial packet scheduling algorithms.[13] Incorporating these algorithms into high-speed switches is one of the technical hurdles facing designers of ISPNs.

2.2 Service Model

The set of services offered by the network is called the *service model.* The service model is embodied in the *service interface*, which allows users to request various kinds of QoS from the network. This service interface plays a key role in network design, because it regulates the interaction between applications and the underlying network and keeps the two cleanly separated. A network can employ any technology that supports the service interface; similarly, any application that expresses its needs through the service interface can

[11] ATM is also a packet-switching architecture. The Internet uses a connectionless packet-switching architecture, while ATM is a connection-oriented packet-switching architecture. While the technical community is in agreement that for today's transmission technologies packet-switching is the correct choice, there is still an active debate about the relative merits of connection-oriented and connectionless packet-switching architectures.

[12] Statistical multiplexing occurs when several entities with variable resource demands share a single resource. The aggregation of these multiple demands has a lower variance than the individual demands, and so the resource is used more efficiently. Packet switched networks can share on a packet basis, whereas circuit switched networks can only share at the level of circuits which is less efficient.

[13] There has been much research in the past few years designing and analyzing these packets scheduling algorithms. See [1, 4, 9, 11, 12, 13, 17, 19, 21, 26, 31, 34] for some examples.

use the network. A stable service interface allows rapid technological improvements in both applications and network technology, without the need for coordination.

The service interface is embedded within applications; it is thus very hard to change without disrupting service to current network clients. Thus, the price for quick progress in applications and networks is that the service interface must remain stable. Because the service interface is essentially a particular parameterization of the service model, the service model must also remain stable.

ISPN designers face the question: what set of services should an ISPN offer? The answer should be based on conjectures about future application and institutional requirements, as well as technical feasibility. For designers, the basic technical assumption is that the more closely aligned the service model is to the needs of applications, the more efficient the network will be. Before addressing this issue in detail, it is useful to provide some context by first reviewing the service offerings of the current telephone network and the Internet.

The telephone network is a circuit switched network. Phone calls require an explicit preallocation of resources while the connection is being established. Calls are blocked if sufficient resources are not available. The service model for ISDN telephone service is the delivery of data at a fixed bandwidth with a fixed delay; all data arrives at the receiver a fixed time after it was transmitted. For the purposes of this discussion, this can be considered one kind of bounded-delay service.

The telephone network serves one application, spoken voice conversation.[14] Specific data rates and delays were chosen to accommodate speech production and recognition. The performance of a phone call is independent of the speed of the underlying phone lines. In case of overload, excess calls are blocked, rather than allowing all calls to connect and delivering degraded service to them all.

[14] I will ignore fax, as this is in some sense a data application that has been overlaid on the phone network because of its ubiquitous connectivity.

Data networks such as the Internet are quite different. First, they are packet-switched rather than circuit-switched. In addition, there is no explicit call set-up, no preallocation of resources, and no admission control;[15] the network offers only *best-effort* service. This means that the network attempts to deliver packets as quickly as possible, but makes no guarantees about delivery and delays. The switches typically use the FIFO packet scheduling algorithm and thus deliver the same quality of service to all applications. Since there is no admission control, the network cannot prevent overloads by refusing service. When the network is overloaded, delays increase and packets are dropped.

Thus, the telephone network and the Internet provide very different service models. The best-effort service model is scalable; it adapts to whatever bandwidth is available. Programmers rarely build real-time requirements, or even specific time-scales, into software.[16] As machines speed up, the program works better; if system performance degrades, the program still works, just not as quickly. This is the appropriate service model for traditional computer-based applications, such as Telnet, FTP, and electronic mail, that also improve with better service (more bandwidth and/or less delay), and degrade gracefully with deteriorating service (less bandwidth and/or more delay). This scalability is amply verified by the fact that this application family has remained relatively unchanged even though network speeds have increased by several orders of magnitude since their introduction.

Applications such as voice conversations or video transmissions are not scalable; they have some fundamental bandwidth and delay requirements.[17] Their performance does not improve greatly if their service requirements are exceeded, but does quickly degrade if their

[15]That is, applications need not ask the network's permission before sending data.

[16]In fact, one of the important paradigms of computer software design is to *abstract* away performance issues by using concepts such as virtual processors, virtual memory, and abstract machines to organize programs.

[17]The bandwidth requirements can be modified by the encoding algorithm, but that entails a change in the signal quality. Also, we will be more specific about the lack of scalability later in this section.

requirements are violated. The best-effort service model, with its inability to provide any assurance about the quality of service, is obviously inappropriate for this class of applications. Applications such as video and voice are better served by service models which resemble the telephony service model with its bounded delays and fixed bandwidth.

One problem shared by both the Internet and telephony service models is that they do not offer different services to different applications. Any ISPN must overcome this limitation by allowing applications to specify their requirements through a service interface which offers a wide variety of services.

Several prototype ISPN's are in operation,[18] but the designers have not reached a consensus on the appropriate service model; in fact, there is strenuous disagreement about several fundamental issues. There is, however, one point of widespread agreement – the service model should be more varied, or *richer*, than the current telephony and Internet offerings, and should combine these paradigms by offering (1) quantitative delay bounds, and (2) several levels of best-effort service. The remainder of this section describes one such proposed service model[19] (see References [1, 28] for a much fuller exposition).

The service model is based on the requirements of applications and institutions. Applications can be roughly divided into those that are *elastic* or scalable, and those that are *real-time*. Elastic applications adjust easily and flexibly to delays in delivery; that is, a packet arriving earlier helps performance and a packet arriving later hurts performance, but there is no set need for a packet to arrive at a certain time. Typical Internet applications are elastic in

[18]The networks cited in [9, 16, 17, 19] are examples of operating prototypes; the design my colleagues and I work on is operational on DARTnet, an ARPA-funded T1 testbed linking roughly a dozen industrial and academic research sites.

[19]There are other aspects to the service model that are not addressed here (such as policy routing, multicast, etc.); this discussion is restricted to services relevant to packet scheduling. The inclusion of other services only strengthens the argument.

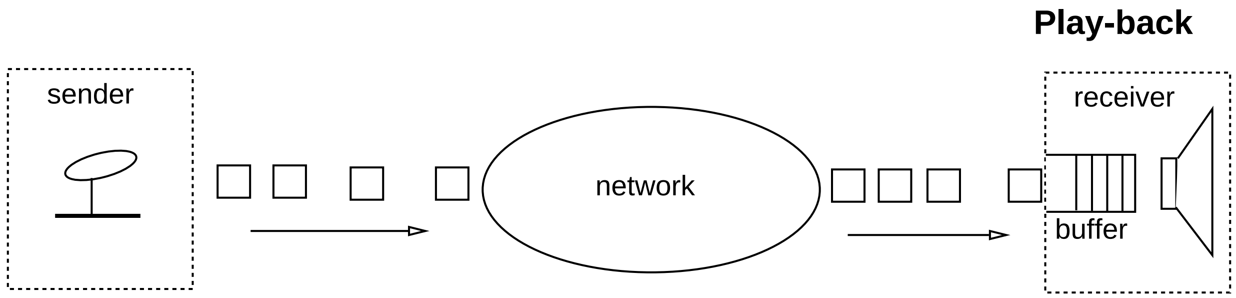

Figure 1: Playback Applications: the receiver plays the data back at the playback point where the playback point is the sum of the generation time and the offset delay.

nature, and the Internet service model has performed well for them. Thus, for elastic applications we propose a service model consisting of several classes of best-effort service. These classes allow applications to indicate their relative sensitivity to delay so that the network can distinguish between the delay requirements of, for example, interactive burst transactions (e.g., Telnet and X-protocol), interactive bulk transfers (e.g., FTP), and asynchronous bulk transfers (e.g., electronic mail and fax).

Real-time applications have more stringent requirements. Some real-time applications are *playback* applications (see Figure 1). In these applications, the source digitizes some signal and transmits it over the network; the data packets are delivered to the receiver with varying delay; the receiver buffers the data and plays the signal back at some specified moment; this moment is called the playback point and is the generation time plus some essentially fixed offset delay. Data that does not arrive at the receiver before its playback point cannot be played back on time; it is of essentially no use. To choose a reasonable value for the offset delay, an application needs some a priori characterization of the maximum delay; this could either be provided by the network in a delay bound, or through the observation of the delays of earlier packets. While our discussion has treated the offset delay as essentially fixed, in reality the application can slowly adjust its offset delay during use as its estimate of the maximal packet delays change.

Delay can affect the performance of playback applications in two ways. First, the value of the offset delay (which is determined by predictions about the future packet delays) affects the interactive nature of the application. Applications vary greatly in their sensitivity to this offset delay. Some playback applications, in particular those that involve interaction such as a phone call, are extremely sensitive to this delay; transmissions of a movie or lecture are less sensitive.

Second, the signal becomes degraded when packet delays exceed the offset delay. Applications vary greatly in their sensitivity to late packets. We can divide these playback applications into those that are tolerant of occasional dropped or late packets, and those that are not. Intolerant playback applications require an absolutely faithful playing back of the original data, either because the hardware or software is unable to cope with missing data, or because the users are unwilling to risk missing any data. On the other hand, users of tolerant applications, as well as the underlying hardware and software, are prepared to accept occasional losses of data. Most casual telephone conversations are tolerant, since human speech tends to be redundant and, if necessary, users can request that the other participants repeat the missing parts of the conversation. It is important to note that the distinction between tolerant and intolerant applications is not just a function of the software and hardware involved but also depends on the needs of the users themselves.

In essence, the performance of elastic applications is more closely related to the average delay of the packets, whereas the performance of real-time applications is more closely related to the maximum delay of the packets. Best-effort service produces reasonable values for average delays but, because the service has wide variations in delay, the maximal delays are intolerable. To keep the maximal delays within a reasonable range, real-time applications need a service with a bounded delay. For intolerant applications, the service model we propose is a firm worst-case bound on delay; this bound should not be violated as long as the network switches and links function properly. Tolerant applications do not need such a reliable bound; for these applications the service model we propose is a loose bound on delay that incorporates predictions about the aggregate traffic load; this bound will occasionally be violated when the predictions are wrong.

Future video and audio applications will likely be playback applications[20], and we conjecture that the vast majority of real-time traffic will be produced by such playback applications. Thus, we expect that our real-time services will fit the needs of most future real-time applications. The taxonomy of applications, and the relevant service offerings, is depicted in Figure 2.

Services for both tolerant and intolerant real-time applications involve admission control; before commencing transmission, applications must request service from the network. This service request consists of a traffic descriptor, in which applications specify their traffic load, and a QoS descriptor, in which applications specify the desired quality of service. We envision employing a traffic descriptor that specifies bandwidth and burstiness. After receiving a service request, the network decides whether or not to accept the request based on whether or not it can deliver the desired QoS. In contrast, there is no admission control for the best-effort service classes. Thus, the predominant failure mode for real-time service is that requests can be blocked, and for best-effort service that best-effort packets can be dropped. Most failures should be suffered by users who indicate – in return presumably for cheaper service – that they are willing to experience a higher failure rate. Thus, the service model incorporates the notions of preemptable packets, the first packets discarded when the network is overloaded[21], and preemptable connections, which are terminated when incoming connections would otherwise be blocked.

These best-effort and real-time service offerings are designed to meet application requirements. Institutional requirements are less frequently discussed. Currently, many private firms partially bypass the regular phone system and lease lines directly for their own inter-

[20]Current video and audio applications on circuit switched networks do not fit this model, since there is no jitter in the network delays; however, the video and audio applications in use on the Internet today, such as *vat* and *nv*, do fit this model.

[21]See [23] for an example of how this might work.

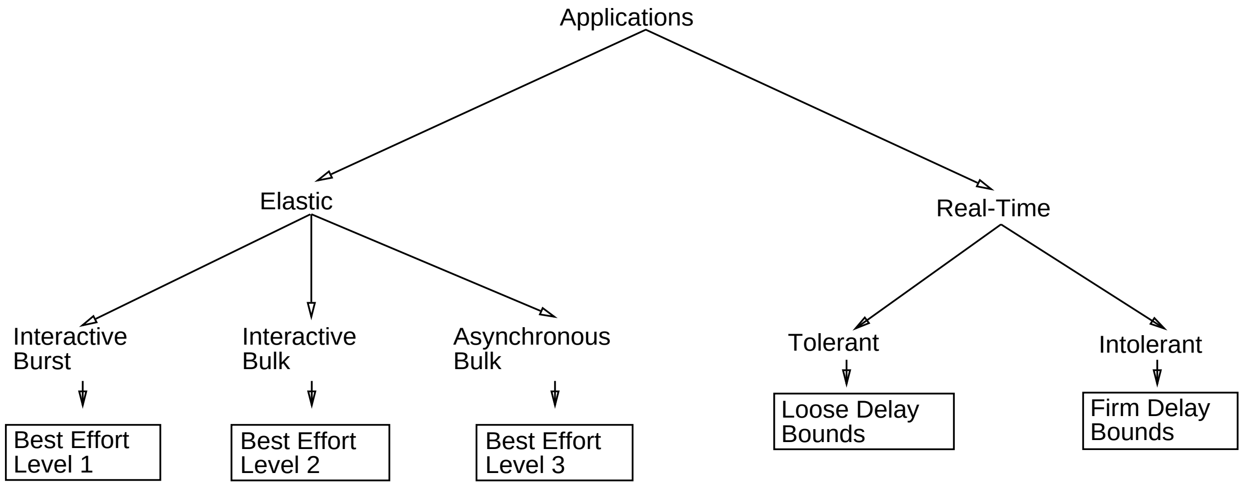

Figure 2: Application Taxonomy: the application classes and the associated service offerings.

nal computer and telephone networks. Leased lines are the telecommunications equivalent of buying wholesale, and the need to lease will likely persist. However, such bypass is harmful to an ISPN because it reduces the revenue stream and prevents the network from taking full advantage of the increasing returns to scale of statistical multiplexing. For an ISPN to remain economically viable, it should keep bypass at a minimum by offering firms a service which is at least equivalent to and preferably superior to directly leasing lines.

One possibility would be to offer *virtual* leased lines consisting of a long-term reservation of real-time service with some bounded delay and fixed bandwidth. This service would be essentially equivalent to directly leasing a line. In order to make the virtual leased line service more attractive than directly leasing lines, two problems common to both virtual leased lines and direct leased lines must be addressed. First, leasing a virtual line leads to inefficient use of bandwidth because the network's QoS scheduling mechanism is circumvented; not all packets sent over this real-time channel need the real-time service. For instance, if a firm leases a real-time line but uses it mostly for file transfer or electronic mail traffic, the internal network switches would schedule the traffic as if it needs the real-time bounded delay service when in fact it needs much less stringent service. Second, such a line cannot be shared by several entities in a controlled way without the firm exercising its own scheduling at the entrance to the connection. This is especially important if several divisions of a firm, or departments of a university, choose to jointly lease some bandwidth and want to ensure that every entity gets its fair share when the link is fully loaded.

We propose a modified virtual leased line service that allows link-sharing through the specification of a hierarchy of link ownership shares. For instance, a link could be shared by several universities; these shares could be split between their various academic departments; these shares in turn could be split among the staff and students of the department, so that finally each individual in the department would have his or her own individual share. In addition, this modified virtual leased line service would offer the QoS service classes discussed above, so that the service needs of the packets are revealed to and utilized by the network in the packet scheduling algorithm. Thus, real-time requirements will always take precedence over elastic ones, but admission control will ensure that every entity gets its share of the bandwidth. Such a service would meet the needs of institutions better than direct leased line service because of its more efficient use of bandwidth and more flexible sharing options.

Thus, our proposed service model incorporates two kinds of real-time service, several classes of best-effort service, and a modified virtual leased-line service. This service model is designed to meet the needs of the entire spectrum of future Internet applications. Of course, a period of experimenting and rethinking will be needed before the field reaches – if it ever does – a consensus on the appropriate service model for an ISPN; while we expect the general form of the eventual service model to resemble our proposal, there will likely be important differences in the details.

2.3 Integration and Efficiency

Our case for the desirability of ISPN's is based largely on the efficient use of the network bandwidth, a notion explored in this subsection. Efficiency is not measured by link utilization or some other network related quantity; instead, efficiency must be connected to the performance of applications using the network. The following model provides provides such a notion of efficiency. Consider some network with a set of clients. Let s_i denote the network performance delivered to client i; s_i describes all the relevant service parameters (such as bandwidth, delay, etc). Assume that every network client's degree of satisfaction with its service is expressed by some function $V_i(s_i)$; this function quantifies how much money the client would be willing to spend for such service. The network has a finite capacity, and so only certain $\vec{s}$ are feasible; let $\mathcal{F}$ denote the feasible set of service allocations. A measure of the efficiency of some service allocation $\vec{s}$ is just the sum $V_{total}(\vec{s}) = \sum_i V_i(s_i)$. The most efficient feasible service allocation $\vec{s} \in \mathcal{F}$ is the one that maximizes $V_{total}(\vec{s})$. The service interface and the underlying packet scheduling algorithm determine the feasible set of allocations. This section illustrates two claims: first, that service models which are more closely aligned with the service needs of the applications are more efficient; and second, that networks with a heterogeneous set of clients are more efficient than ones with a homogeneous set of clients.

Consider a network with a single link modeled by an exponential server (of rate $\mu = 1$) and Poisson arrival processes. Consider two types of network clients, with Poisson arrival rates $r = 0.25$ and with $V_1 = 4 - 2d_1$ and $V_2 = 4 - d_2$ where d_i represents the average queueing delay delivered to client i.[22] Thus, we have two clients with different sensitivities to delay. If we use FIFO service in the network, then $d_1 = d_2 = \frac{1}{(1-0.5)} = 2$ and so $V_{total}^{FIFO} = 2$. If we use strict priority service with preemption, and give client 1 priority, then $d_1 = \frac{1}{(1-0.25)} = 4/3$ and $d_2 = \frac{1}{(1-0.25)(1-0.5)} = 8/3$ and $V_{total}^{priority} = 8/3$. Thus, the strict priority scheduling algorithm is more efficient than FIFO, since the client that is less sensitive suffers more delay. In fact, when compared to all possible scheduling algorithms, the strict priority scheduling algorithm

[22]Recall that the average delay in the M/M/1 queueing network considered here is just $d = \frac{1}{(\mu-r)}$. If we have two priority levels with preemption, with arrival rates r_1 and r_2 respectively, then the delays are given by $d_1 = \frac{1}{(\mu-r_1)}$ and $d_2 = \frac{\mu}{(\mu-r_1)(\mu-r_1-r_2)}$.

gives the most efficient feasible allocation of delay.[23]

If we double the linespeed, $\mu = 2$, and double the number of applications, the performance numbers become: $V_{total}^{priority} = 32/3$ and $V_{total}^{FIFO} = 10$.[24] Now consider two networks, with separate bandwidths $\mu_1 + \mu_2 = 2$, each carrying two applications. If we divide the clients so that each network carries one application of each type, and the bandwidths are split evenly (which is optimal here), then we revert back to our original case and the most efficient choice is to use priority on each link and achieve a total efficiency of 16/3. If we partition the clients so that one network carries two delay-sensitive applications and the other carries the two less sensitive applications, then the optimal arrangement is to use FIFO on each network and to split the bandwidth as $\mu_1 = 5/2 - \sqrt{2} = 1.086...$ and $\mu_2 = \sqrt{2} - 1/2 = .914...$, which yields an efficiency of $10 - 4\sqrt{2} = 4.343....$ Thus, there is greater efficiency when the application types are mixed than when the application types are segragated.[25]

This formulation of efficiency reveals two problems with offering a variety of QoS. First, not every client gains directly from the resulting increase in efficiency; that is, $V_{total}(\vec{s}^1) > V_{total}(\vec{s}^2)$ does not imply that $V_i(s_i^1) > V_i(s_i^2)$ for all i. For instance, in the simple example with just two clients, $V_2^{FIFO} = 2$ but $V_2^{priority} = 4/3$ even though $V_{total}^{priority} > V_{total}^{FIFO}$.

[23]This example is misleading in that a slightly overprovisioned FIFO network ($\mu = 17/16$) has the same efficiency value as the priority network. If we consider more realistic examples, where the delay preferences are more varied, a much greater degree of overprovisioning is needed to make a FIFO network match the efficiency of a priority network. In fact, this necessary degree of overprovisioning increases without bound as the variation in delay requirements increases.

[24]Because the network's value for $V_{total}^{priority}$ and V_{total}^{FIFO} has more than doubled when we doubled both the linespeed and application load, this example illustrates the increasing returns to scale of statistical multiplexing. Variations in load can be more fully averaged out when more users share the network, and smoothing of the load leads to more efficient use of the network's resources.

[25]Using the formula for delay, we have $V_{total} = \{8 - \frac{4}{\mu_1 - 0.5}\} + \{8 - \frac{2}{\mu_2 - 0.5}\}$. When combined with the constraint $\mu_1 + \mu_2 = 2$ we can solve for the optimal value of μ_1 and μ_2.

Efficiency in the heterogeneous networks is gained by shifting resources from applications that are not extremely performance-sensitive to those that are; the performance-sensitive clients gain from using more sophisticated scheduling algorithms, but the less performance-sensitive clients lose. Considering only network service, the increase in efficiency benefits only the performance-sensitive applications and in fact harms the less performance-sensitive applications. Why should users of less performance-sensitive applications be in favor of richer service models?

Second, in order to achieve efficiency, clients must request the appropriate service for their application. In our simple example, the increased efficiency of priority service can only be realized if the network can recognize which client is more delay-sensitive. Any reasonable service interface will allow this, but the clients have to use this interface appropriately; high quality service should only be requested by users of performance-sensitive applications. This raises an important incentive issue: if efficiency depends on the truthful revelation of preferences, but the less performance-sensitive clients lose when they reveal their lack of performance sensitivity, what will motivate clients to reveal their preferences honestly?

Careful pricing of network services provides a solution to both of these problems. Pricing can provide the appropriate incentives to clients to reveal their service requirements honestly. Pricing can also spread the benefits of the increased efficiency to all clients. Clearly, the network should charge less for low quality service than for high quality service. More specifically, the pricing scheme should be designed so that the increase in quality of service outweighs the increase in cost for performance-sensitive applications, and the decrease in cost outweighs the decrease in quality of service for the less performance-sensitive applications. In this case, when both the cost and quality of network service are considered, all application classes benefit from the increased efficiency of ISPNs. Furthermore, such a pricing scheme will provide the proper incentives so that only users of performance-sensitive applications will request high quality service; then the selfish choices of individual clients will allow the network as a whole to achieve optimal efficiency.[26] We return to the issue of pricing in Section 3.3.

3 Policy Developments and Pricing Schemes

Much of the design and prototyping of ISPN's takes place in research settings that are quite insulated. In particular, for the various testbed networks where these designs are or will be tested, the test user community is small and cooperative, the designs are determined more by technical factors than market pressures, and network service is free. However, in the Internet, or in any other similar public network, these conditions will no longer hold. In particular, users will be charged, either directly or indirectly, for network service. In this section, I discuss how the integration of services, along with the policy developments of public access and privatization, determine the form of the pricing schemes.

3.1 Public Access

With the rapid fall of policy and technical barriers, access to the Internet is becoming much more widespread – within the next decade, a sizable fraction of homes may have Internet access. A dramatic increase in the Internet user population would have several important ramifications. First, once a home consumer standard is widely adopted, there is tremendous pressure for it to remain stable. Thus, once Internet usage becomes a staple for many American families, the service interface and other network protocols will become even harder to change than they are now (and they are already quite difficult to change). It is extremely important that the networking community make the right technical decisions about the proper service interface now.

[26] See [32] for a discussion of such priority pricing and [2, 3] for an application of these ideas to networks. For a more theoretical treatment of incentive issues and efficiency, see [27].

Also, once artificial access restrictions are lifted, the Internet will no longer have a small, technically knowledgeable, and very cooperative user community. As with other widely used public facilities, informal enforcement of behavioral norms is unlikely to be sufficient to ensure socially desirable behavior; not all network clients will truthfully reveal their service preferences, unless it is in their own interest. One cannot rely on policing for this, so pricing will have to provide incentives for proper network usage.

3.2 Privatization

In the future, multiple private carriers will compete to offer network services, perhaps much as we now have multiple long-distance phone companies. While the issue of competition raises many issues, here I discuss only that of reselling bandwidth. Assuming that the principle of common carriage[27] applies to these networks (there are those that doubt the viability of this principle; see [20]), then a firm can purchase network service from one provider and *resell* it to others. There is nothing intrinsically wrong with reselling; in some sense it is simply the transfer of marketing responsibilities. However, reselling can result in decreased efficiency when the service requirements of the traffic is not accurately represented. For example, buying real-time service in bulk and then reselling this service to elastic traffic leads to inefficient use of the network; a different allocation of service – giving the bulk elastic traffic best-effort rather than real-time service – would produce a higher total level of satisfaction. Thus, for a pricing scheme to encourage efficient allocation of resources, such reselling must not be profitable.

Our modified virtual leased line service proposal discourages the mislabeling of service needs since it allows users to buy in bulk and still utilize the QoS service classes; the network is presented with each packet's true service needs and can schedule the packets accordingly.

[27]Common carriage is the requirement that networks provide service to all customers on an equal footing. In particular, common carriage means that networks cannot refuse to sell service to competitors.

As we stated above, in order to achieve efficiency the pricing scheme must also be designed to make reselling unprofitable. However, other factors besides pricing and the service model can influence the profitability of reselling. In a connection-oriented technology, such as ATM, if connection set-up involves substantial delays then there will be strong market pressure to resell service from pre-set-up connections to clients with intermittent traffic needs. Thus, if ATM and other similar technologies are to avoid being primarily a leased line service, they must provide rapid connection establishment.

3.3 Pricing

With widespread competition among network service providers, pricing schemes will likely lead to a high degree of efficiency because providers must guard against new competitors with more efficient schemes. Thus, we will assume that network efficiency is an important pricing goal. Chapter ?? [MacKie-Mason and Varian] discusses these efficiency issues in much greater detail.

There are a wide variety of pricing policies. The granularity of the pricing policy can vary by *what* is priced – access, connections, packets, etc. – and by how charges are identified – institution, user, application, etc. This is not a question of who pays the bill (in many cases that will be the institution), but rather how detailed the bill should be.

There is tremendous pressure to offer access-based pricing schemes; that is, to charge only for the size of the access link.[28] This has two important advantages: it is technically easy (in fact, it is currently the predominant charging scheme in the Internet), and predictable.[29]

[28]Most pricing schemes will include at least some charge for access; I use the term access-based pricing to refer to pricing schemes which *only* charge for access.

[29]While predictable costs are indeed desirable, in many facets of life such as utility and phone usage we have adjusted quite well to variable costs. However, both the newness of the basic technology and the rapidly changing usage patterns, in addition to the historical legacy of access-based pricing, contribute to the rather

In a network with a nontrivial service model, to achieve efficiency the pricing scheme must (1) provide incentives for users to specify the appropriate service class, and (2) ensure that reselling is not profitable. Access-based pricing fails to meet either requirement.

To provide incentives for appropriate service requests, the network must tie costs to the individual clients who make the QoS choices, and must also price the various services differently. Pricing based exclusively on access charges does neither. These requirements are met by QoS-sensitive pricing schemes that charge for both the establishment of the reservation or connection and its duration.[30]

Furthermore, access-based pricing makes reselling profitable, since a firm should attempt to sell all of its unused capacity. This reselling problem persists even when the pricing scheme includes charges for individual connections. For example, if I establish a video connection with a reserved bandwidth of 100mbps, but my video source has an average bandwidth of 10mbps (and a peak rate of 100mbps), then if I am charged for 100mbps I should resell the remaining 90mbps. I would make sure that my packets get priority when my video surges to the peak rate, and thus the service I actually sell is not real-time but best-effort. To ensure that such reselling is not profitable, the pricing policy should also be usage-based; that is, my cost should not be based only on the size of my access pipe, my QoS, and my connection parameters, but also on my actual usage; the per-packet charge must be greater than what I can resell the best-effort service for. Thus, in addition to the access charge, the appropriate pricing scheme for real-time connections is a QoS-sensitive multipart tariff with a charge for the establishment of the reservation, a charge for the duration of the reservation, and a charge for the actual usage of the reservation. For best-effort traffic, a QoS sensitive per-packet charge is appropriate.

---

vocal demand for predictable costs in networking.

[30]These might most naturally be called connection charges, but to avoid the possible confusion because often access charges are called connection charges, we will adopt the term reservation charges.

Such a scheme would impose severe accounting requirements on the underlying network architecture. There is almost no accounting infrastructure currently in place. For the kind of QoS-sensitive and usage-based pricing advocated here, the network needs an accounting infrastructure that can record individual packets and assign their costs to the appropriate clients. The network and the operating system of the host would have to cooperate to identify individual users (or perhaps applications) as the responsible entities. It is not clear where this division of labor should split; in the ATM context, the network would most likely assign the charge to a VCI (virtual circuit identifier) and the operating system would be responsible for associating each VCI to a user or application. The network accounting infrastructure must be built into the underlying network protocols, which must support some degree of authentication so that charges are not misassigned. While the integration of services is a fashionable topic in academic research circles, accounting and authentication are not.[31] If we are to succeed in building an integrated services Internet, we must address these problems.

The fact that users will generate costs based on their network usage (even if they themselves don't pay the bill) will be a dramatic shift away from the recent trend in computer systems to hide the underlying technology from users. Operating systems will need to account for network usage by user, and user interfaces will need to inform users about the charges their actions are incurring. Again, little work has been done in this area, but it must if an integrated services Internet is to be widely used.

One objection is that a highly detailed pricing scheme will create huge transaction costs. While it is true that a sizable fraction of the telephone system's cost is in billing, it is not clear how much of that cost is for accounting (collecting data and computing the amount to be charged) and how much is for processing bills for individual customers (sending out bills, handling incoming envelopes, cashing checks, etc.). My uninformed guess is that the cost of doing the accounting is probably not nearly as great as the cost of the actual billing; our

[31] Reference [5] is one of the few contributions in this area.

proposal necessitates accounting, but for commercial customers the billing should probably be sent to the firms, not broken down by individual. Thus, at least in the beginning when most users of the Internet are commercial or academic, not residential, the billing overhead may be manageable. A network that provides a variety of qualities of service must have detailed billing; my guess is that, despite its cost, such billing is more cost-effective than doing without a QoS mechanism.

4 Next Steps

An efficient integrated services Internet must offer a rich service model that combines real-time service, best-effort service, and a modified virtual leased line service. Moreover, such a service model will only be used efficiently if it is combined with a usage-based and QoS-sensitive pricing scheme. However, the current Internet has neither a rich service model nor an accounting infrastructure capable of supporting sophisticated pricing schemes. What are the critical next steps that will take us from the current Internet to our vision of the future Internet?

First, and most importantly, the Internet must adopt standards that mandate a full accounting infrastructure and a rich QoS service interface. Such a radical change in the Internet would have been difficult enough in the Internet's infancy when technical decisions were made by a rather small and cohesive technical community. In the current environment, changing the Internet's basic architecture is especially problematic. Because the government no longer plays a significant role in determining the Internet's architecture, the companies who produce network equipment and the consumers of this equipment must reach a consensus on this architectural transformation. Managing this transition will be the ultimate test of the Internet's organizational coherence.

Second, since the cost of usage will become important, operating systems and user interfaces need to be modified so that users are more aware of their network usage and its cost.

Organizations will feel comfortable with usage-based pricing only when users are familiar with network costs and can make informed decisions. Most organizations can handle usage-based phone charges because users understand the pricing structure and are able to control their own costs effectively; we need to create a similar situation in the Internet.

Finally, while it may lead to inefficiencies in the short term, the pricing structure should accommodate users who require either a fixed fee cost structure or extremely inexpensive service. There are important user communities, such as schools and libraries, where the variability in usage-based costs would cause extreme difficulties. We must find ways to accommodate these communities.[32] There are also likely to be many potential users who, at competitive prices, could not afford even the most basic level of Internet access. However, if we expect the Internet to become an important part of out telecommunications infrastructure, we must strive for universal access either through price regulations or user subsidies.

5 Acknowledgments

The ideas in this paper were developed through many hours of discussion with my colleagues David D. Clark, Deborah Estrin, Bryan Lyles, David Sincoskie, and Lixia Zhang. Furthermore, the service model presented here represents joint work with David D. Clark and Lixia Zhang. In addition, I would like to thank Hal Varian, Jeff MacKie-Mason, Vint Cerf, Steve Deering, Padmanabhan Srinagesh, and Abel Weinrib for helpful discussions. However, not all of the discussants agree with everything written here, and all errors or misconceptions in this draft are solely the responsibility of the author.

[32]This could involve using other forms of incentives to control network usage, or merely accepting the inefficiencies that result when the proper incentives don't exist.

References

[1] D. Clark, S. Shenker, and L. Zhang. *Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanism* In **Proceedings of SIGCOMM '92**, pp 14-26, 1992.

[2] R. Cocchi, D. Estrin, S. Shenker, and L. Zhang. *A Study of Priority Pricing in Multiple Service Class Networks*, In **Proceedings of SIGCOMM '91**, pp 123-130, 1991.

[3] R. Cocchi, S. Shenker, D. Estrin, and L. Zhang. *Pricing in Computer Networks: Motivation, Formulation, and Example*, In **IEEE/ACM Transactions on Networking**, 1(6), pp 614-627, 1993.

[4] A. Demers, S. Keshav, and S. Shenker. *Analysis and Simulation of a Fair Queueing Algorithm*, In **Journal of Internetworking: Research and Experience**, 1(1), pp. 3-26, 1990.

[5] Deborah Estrin and Lixia Zhang. *Design considerations for usage accounting and feedback in internetworks* In **ACM Computer Communication Review**, 20(5), pp 56-66, 1990.

[6] G. Faulhaber *Pricing Internet: The efficient subsidy*, In **Building Information Infrastructure**, edited by B. Kahin, 1992.

[7] D. Ferrari. *Client Requirements for Real-Time Communication Services*, In **IEEE Communications Magazine**, 28(11), 1990.

[8] D. Ferrari. *Distributed Delay Jitter Control in Packet-Switching Internetworks*, In **Journal of Internetworking: Research and Experience**, 4(1), pp 1-20, 1993.

[9] D. Ferrari and D. Verma. *A Scheme for Real-Time Channel Establishment in Wide-Area Networks*, In **IEEE JSAC**, 8(3), pp 368-379, 1990.

[10] P. Green. *The Future of Fiber-Optic Computer Networks*, In **IEEE Computer**, 24(9), pp 78-89, 1991.

[11] S. J. Golestani. *A Stop and Go Queueing Framework for Congestion Management*, In **Proceedings of SIGCOMM '90**, pp 8-18, 1990.

[12] S. J. Golestani. *Duration-Limited Statistical Multiplexing of Delay Sensitive Traffic in Packet Networks*, In **Proceedings of INFOCOM '91**, 1991.

[13] S. J. Golestani. *A Framing Strategy for Congestion Management*, In **IEEE JSAC**, 9(9), pp 1064-1077, 1991.

[14] R. Guérin and L. Gün. *A Unified Approach to Bandwidth Allocation and Access Control in Fast Packet-Switched Networks*, **Proceedings of INFOCOM '92**, 1992.

[15] R. Guérin, H. Ahmadi, and M. Naghshineh. *Equivalent Capacity and Its Application to Bandwidth Allocation in High-Speed Networks*, In **IEEE JSAC**, 9(9), pp 968-981, 1991.

[16] J. Hyman and A. Lazar. *MARS: The Magnet II Real-Time Scheduling Algorithm*, In **Proceedings of SIGCOMM '91**, pp 285-293, 1991.

[17] J. Hyman, A. Lazar, and G. Pacifici. *Real-Time Scheduling with Quality of Service Constraints*, In **IEEE JSAC**, 9(9), pp 1052-1063, 1991.

[18] B. Kahin. *Overview: Understanding the NREN*, In **Building Information Infrastructure**, edited by B. Kahin, 1992.

[19] C. Kalmanek, H. Kanakia, and S. Keshav. *Rate Controlled Servers for Very High-Speed Networks*, In **Proceedings of GlobeCom '90**, pp 300.3.1-300.3.9, 1990.

[20] E. Noam. *The Impending Doom of Common Carriage*, preprint, 1993.

[21] A. Parekh and R. Gallager. *A Generalized Processor Sharing Approach to Flow Control-The Single Node Case*, In **IEEE/ACM Transactions on Networking**, 1(3), pp 344-357, 1993.

[22] A. Parekh. *A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks*, In **Technical Report LIDS-TR-2089**, Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, 1992.

[23] David Petr, Luiz DaSiliva, and Victor Frost. *Priority discarding of speech in integrated packet network.* **IEEE Journal on Selected Areas in Communications**, 7(5), pp 644-656, June 1989.

[24] J. Postel. *Internet protocol.* Request For Comments 791, Information Sciences Institute, University of Southern California, August 1981.

[25] J. Postel. *Transmission control protocol.* Request For Comments 793, Information Sciences Institute, University of Southern California, September 1981.

[26] H. Schulzrinne, J. Kurose, and D. Towsley. *Congestion Control for Real-Time Traffic*, In **Proceedings of INFOCOM '90**.

[27] S. Shenker. *Efficient network allocation with selfish users.* In **Proceedings of Performance '90**, edited by P. J. B. King, I. Mitrani, and R. J. Poole, pp 279-285, 1990. Edinburgh, Scotland. North-Holland.

[28] S. Shenker, D. Clark, and L. Zhang. *A Scheduling Service Model and a Scheduling Architecture for an Integrated Services Packet Network* preprint, 1993.

[29] M. Sirbu. *Telecommunications Technology and Infrastructure*, In **A National Information Network**, Institute for Information Studies, 1992.

[30] L. Smarr and C. Catlett. *Life after Internet: Making room for new applications*, In **Building Information Infrastructure**, edited by B. Kahin, 1992.

[31] D. Verma, H. Zhang, and D. Ferrari. *Delay Jitter Control for Real-Time Communication in a Packet Switching Network*, In **Proceedings of TriCom '91**, pp 35-43, 1991.

[32] Robert Wilson. *Efficient and competitive rationing.* **Econometrica**, 57(1), pp 1-40, 1989.

[33] L. Zhang. *A New Architecture for Packet Switching Network Protocols*, In **Technical Report LCS-TR-455**, Laboratory for Computer Science, Massachusetts Institute of Technology, 1989.

[34] L. Zhang. *VirtualClock: A New Traffic Control Algorithm for Packet Switching Networks*, In **ACM Transactions on Computer Systems**, 9(2), pp 101-124, 1991.