Adding Service Discrimination to the Internet

David D. Clark
MIT Laboratory for Computer Science
September 1995, Version 2.0

ABSTRACT

This paper explores extensions to the Internet that can provide discrimination in the service offered to different users in times of network congestion. It proposed a scheme which allows different users to adjust their sending rates to different values during overload. This scheme is contrasted with a number of resource allocation schemes under consideration. [1]

1. INTRODUCTION

This paper explores the issue of extending the Internet by adding features that permit allocating different service levels to different users. Specifically, the problem to be solved is sharing bandwidth in times of congestion. One of the most significant performance complaints of real users today is that large data transfers take too long, and that there is no way to adjust or correct for this situation. People who would pay more for a better service cannot do so, because the Internet contains no mechanism to enhance their service. Historically the Internet has not allowed the user to select one or another service. Instead, the Internet has implemented one service class, and used a technical means rather than a pricing means to allocate resources when the network is fully loaded and congestion occurs.

New mechanism and pricing must go hand in hand to provide a range of service levels. Mechanism is needed to control the actual allocation of bandwidth; pricing is needed to regulate the use of this allocation. This paper will not concentrate on pricing policy, but instead on a discussion of mechanism (both for bandwidth allocation and pricing), because it is the mechanism we must get right. Pricing decisions can be changed quickly, but it takes a long time to implement and deploy new features inside the Internet. If we get the mechanism wrong, it may take a number of years to recover from that error. Today there is no agreement in the Internet community as to what a service allocation mechanism should be, and indeed no universal agreement that such an addition to the Internet is appropriate. Thus, the goal of this paper is to stimulate discussion on what the enhanced service of the Internet should be, and what mechanism should be added for this purpose.

---

[1]This research was supported by the Advanced Research Projects Agency of the Department of Defense under contract DABT63-94-C-0072, administered by Ft. Huachuca. This material does not reflect the position or policy of the U.S. government, and no official endorsement should be inferred.

In fact, the service provided by the Internet is rather complex. The problem is to understand what aspect of the Internet service the user actually values. Failure to understand what service features are valued by the user can lead to the implementation of potentially complex control mechanisms that do not meet real user needs. Thus, the structure of this paper is to describe how the Internet currently deals with congestion, speculate on the service features that relate to user satisfaction, assess in this context some existing proposals for service enhancements, and finally to propose a new approach, which seems to provide considerable generality in meeting user needs, and provide a framework for relating the service obtained to the pricing for it.

2. BANDWIDTH ALLOCATION TODAY IN THE INTERNET

The Internet today uses a service model called "best effort". In this service, the network allocates bandwidth among all the instantaneous users as best it can, and attempts to serve all of them without making any explicit commitment as to rate or any other service quality. Indeed, some of the traffic may be discarded, although this is an undesirable consequence. When congestion occurs, the sources of traffic are expected to detect this event and slow down, so that they achieve a collective sending rate equal to the capacity of the congested point.

In the current Internet, rate adjustment is performed by software that runs in the computer that is the source of the data. That software implements the transport protocol of the Internet, which is called TCP. The general approach is specified as follows. A congestion episode causes a queue of packets to build up. When the queue overflows and one or more packets are lost, this event is taken by the sending TCPs as an indication of congestion, and the senders slow down. Each TCP then gradually increases its sending rate until it again receives an indication of congestion. This cycle of increase and decrease, which serves to discover and utilize whatever bandwidth is available, continues so long as there is data to send. TCP, as currently specified and implemented, uses a set of algorithms named "slow start", and "fast retransmit" [Jacobson], which together realize the rate adaptation aspect of the Internet. These rather sophisticated algorithms have been developed over the last several years, and seem to work fairly well in practice[2].

It is sometimes assumed that the consequence of congestion is increased delays. People have modeled the marginal cost of sending packets into a congested Internet as the increased delays that those packets encounter. However, this perception is not precisely correct. Because of the rate adaptation, the

[2]Another fact worth noting about the Internet is the rather long delays in the control mechanisms. With the cross country round trip delay about .1 seconds, the response time of TCP to congestion information is necessarily slow. In this sort of time, many packets can be sent from a single source, and substantial short-term congestion can occur, which must be dealt with by queuing.

queue length will increase momentarily, and then drop back as the sources reduce their rates. Thus, the impact on the user of using a congested network is not constant increased delays to individual packets, but a reduction in throughput for data transfers. Further, observing the delays of individual packets does not give an indication of the throughput being achieved, because that depends not on individual packet delays, but on the current sending rate of the TCP in question. Thus, given TCP today, packet delay is not an indication of service quality.

Observation of real delays across the Internet suggests that wide variation in delay is not, in fact, observed. The minimum round trip delay across the country, due to speed of light and other factors not related to load, is about .1 seconds. Isolated measurements of delay across the Internet usually yield values in this range, whether the measurements are taken in periods of presumed high or low load. MacKie-Mason and Varian [MacKie] have measured variation of delay on a number of Internet links, and observed that in some cases maximum delay is indeed observed to increase under periods of higher loads, but that the average does not usually deviate markedly in most cases.

If the delays of individual packets are not much increased by congestion, how then does a user perceive congestion and its impact on performance? At any particular moment, the user is transferring a data object of some certain size. For remote login, the element is the single character generated by each keystroke. For a Web browser, the element is a web page, of perhaps 2K bytes average. And for a scientist with large data sets to transfer, the element may be many megabytes. The hypothesis of this paper is that in each case, the criterion that the user has for evaluating network performance is the total elapsed time to transfer the typical element of the current application, rather than the delay for each packet.

For an application with a limited need for bandwidth and a small transfer element size, such as a remote login application, the impact of congestion is minimal. Isolated packets sent through a congested network will see an erratic increase in delay and occasional losses, but will otherwise not be harmed. The transfer of a typical web page takes only a few packets, and if these packets are slightly delayed in transit, this effect is masked by the round trip delay across the Internet [3]. However, the measurable effect of congestion becomes more pronounced as the data object gets larger. For a larger transfer, the effect of round trip delay is minimized, since many packets are in transit at once. Thus, for a user moving a large data file, the rate adaptation translates into an total elapsed time for the

[3]The typical round trip for a cross country Internet path is about .1 seconds. The transmission time of a 2KByte data element on today's long distance trunks (45 mb/s) is about .00035 seconds. Thus, even a queuing delay of 100 packets, adding .035 seconds of delay, is substantially less than the irreducible round trip delay.

transfer that is proportional to the size of the file and the degree to which the source slows due to congestion. The delays of individual packets are not a significant factor in this overall transfer time. The rate adaptation can adjust the sending rate over several orders of magnitude, since a properly implemented TCP on an advanced workstation or PC today can fully load even a 45 mb/s trunk, while users on a congested network might see only a fraction of a megabit per second or less achieved throughput.

The next observation about usage of the Internet is that the traffic from an individual user is often very bursty. That is, most applications do not send a continuous stream of data, but instead send infrequent bursts, each representing a separate data object. Consider again the example of a user exploring a series of Web pages. In contrast to a phone call, which represents a stream of bits continuously arriving from the user, the bits for each Web page are all delivered to the network at once, and (presumably) are all to be delivered as soon as possible. It is this characteristic of traffic that is described as bursty[4]. Analysis of actual traffic on parts of the Internet suggest that the bursty nature of traffic is very pronounced. One model for actual Internet traffic that has been proposed [Willinger] is the superposition of a number of on-off sources, where the distribution of on and off intervals is heavy-tailed, or has infinite variance.

In general, the faster a packet network delivers a data object, the greater the user satisfaction. This is in strong contrast with the telephone system, where a phone call on an unloaded network cannot usefully "go faster". But since a TCP cyclically increases its sending rate, it will just send faster if it discovers unused bandwidth. This directly translates into a reduced overall transfer time, and thus (presumably) greater benefit to the user[5]. The goal of delivering the data object as quickly as possible adds to the bursty nature of the observed traffic.

The final observation about traffic on the Internet is that the association between source and destination may change very rapidly. Some packet flows last a long time -- a remote login connection or a large data transfer. But others may come and go with great rapidity. A user searching the Web may

---

[4]If the user fetches a new Web page on the average every 10 seconds, and each page on the average is 2Kbytes, then the data from each user occupies the trunk for .00035 seconds every 10 seconds. 28,000 of this sort of user can be supported in each direction across a 45 mb/s trunk. It is also worth noting that one such user generates data at about 2.5% of the rate of a single phone call. This relative efficiency is one of the characteristics of data transfers over the Internet.

[5]This adaptivity is also what allows TCP to transfer data over links of widely varying speeds, from dialup modems to 100 mb/s LANs and beyond. The current record for a long distance TCP transfer is over 500 mb/s.

go to a different network location for each successive page, and a user sending e-mail typically sends to a succession of different receivers.

Thus, one should envision the traffic on the Internet as a mix of data objects from different users, with different sizes and different objectives as to overall delivery time. One user may be transferring a single keystroke, with the goal of delivery in a fraction of second. Another user may be transferring an image of many megabytes, with the goal of delivery within five minutes. A third user may be connecting to a succession of locations across the Internet, and transferring an unpredictable number of bytes from each before moving on. Somehow, the bandwidth allocation mechanisms of the Internet must combine all these disparate uses in a way that makes each of the users sufficiently satisfied. Currently, the allocation of bandwidth between all of these sources is implicit, based on the rate adaptation that TCP performs as each source encounters congestion. The network does not know the size of the object being transferred; all it sees is the succession of packets into which the data has been broken; further, the network does not know how many bytes will ultimately be transferred, nor the overall target delivery time. This is the context into which we should consider adding some improved scheduling mechanism.

As an aside, there is another dimension to the service quality, beyond the desire for a particular target elapsed time for delivery, which is the degree to which the user is dissatisfied if the target delay is not met. For most services, as the delivery time increases, the user has some corresponding decrease in satisfaction. In some cases, however, the utility of late data drops sharply, so that it is essentially useless if the delivery target is missed. The most common case where this arises is in the delivery of audio and video data streams that are being played back to a person as they are received over the net. If elements of such a data stream do not arrive by the time they must be replayed, they cannot be utilized. Applications with these very sharp loss of utility with excess delay are usually called *real time* applications, and the applications in which the user is more tolerant of late data are sometimes called *elastic*. There is much current work to add support for real time services to the Internet. However, this paper concerns itself with perhaps the more basic but less well explored issue of adding service allocation for elastic applications.

3. EXISTING SCHEMES FOR BANDWIDTH ALLOCATION

A number of approaches have been proposed for control of usage and explicit allocation of resources among users in time of overload, both in the Internet and in other packet networks. As a starting point, it is useful to look at these, and see how well they match the patterns of usage described above.

Guaranteed minimum capacity service --

As usually defined, this service provides an assured worst case rate along the path from a source to a specific destination. There are a number of options for how such a service might be specified. One is that the user would make a long term reservation along each potential path[6]. The problem with this is that the user must specify separately the desired rate along each separate path to any potential recipient. Thus, this approach does not scale well to networks the size of the Internet. This problem might be mitigated by moving from permanent reservation to temporary reservation established as needed. However, the delay and network traffic required to establish a temporary reservation may be hard to justify if the user is only going to transfer a small number of bytes before going on to another destination. The most basic problem with a guaranteed minimum capacity service, however, is that a simple guaranteed minimum capacity presumes that the traffic offered by the user is a steady flow, while in practice the traffic is extremely variable or bursty. Each object transferred represents a separate short term load on the network, which the user wants serviced as quickly as possible, not at a steady rate. To guarantee continuous capacity at the peak rate desired by the user is not feasible; it would result in a network with vastly increased capacity, mostly unused, and thus presumably cost.

Fair Allocation service --

If a provider is selling the same service to two users, and giving one a smaller share when they offer equal load, then that user presumably has a complaint. The point of a fair allocation service is to assure the various users that they are being treated in a equitable way relative to each other. If one could find a useful definition of fairness, adding such a mechanism might enhance user satisfaction. The problem with this approach is to find a useful definition of fairness.

Consider a specific *flow* of packets, a sequence of packets that represent one transfer from a source. Each flow, along its path in the network, may encounter congestion, which will trigger a rate adjustment at the source of a flow. In concrete terms, each TCP connection would thus represent a flow. It would be possible to build a packet switch which assured that each flow passing through it received an equal share. Methods to implement this, such as Weighted Fair Queuing [Demers, Clark], are well known. But this sort of switch would only achieve local equality inside one switch. It would not really insure overall fairness, because it does not address how many flows each user has, and how they

---

[6]The Frame Relay service is defined in this way. The subscriber to a Frame Relay network must purchase a Permanent Virtual Circuit (PVC) between each source and destination for which a direct connection is desired. For each PVC, it is possible to specify a Committed Information Rate (CIR), which is the worst case rate for that PVC. Presumably, the provider must provision the network so that there are sufficient resources to support all the CIRs of all the clients. But capacity not being used can be shifted at each instant to other users, so that the best case peak rate can exceed the CIR. This makes the service more attractive.

interact. What if one user has one flow, and another 10? What if those 10 flows follow an identical path through the net, or go to 10 totally disjoint destinations? If they go to different destinations, what does congestion along one path have to do with congestion along another? If one path is uncongested, should a flow along that path penalize the user in sending along a congested flow? And finally, what about multicast flows, that radiate out from a source to multiple destinations? If the goal is to enhance the network service by offering the users some assurance of overall fairness, all these issues must be resolved.

As hypothesized above, what the user considers in evaluating the service being provided is the total elapsed time to complete the transfer of an object of some typical size, which may be very small or very large. In this context, a simple scheme that gives equal access to a point of congestion may not accomplish the desired allocation of resources. In fact, it may be necessary to allocate a larger share of the link to the user with the larger file, depending on the service expectation of the two users. In the practical case of the Internet, one of its "features" may well be that a user transferring a large file can obtain more than his "fair share" of bandwidth during this transfer. The fairness manifested by this system is not that each user is given an instantaneous equal share, but that each user is equally permitted to send a large file as needed. While this "fairness" may be subject to abuse, in the real world it meets the needs of the users.

Local mechanism inside a switch that allocates traffic to classes and gives a controlled share of capacity to each class does have an important role to play in the Internet, and is being implemented and shipped in routers today. But what it is used for is allocation of capacity to aggregates of traffic, not dealing with the fine-grained service objectives of individual users.

Dynamic bidding for access --

Given the highly bursty nature of most network traffic today, making any sort of continuous reservation for capacity seems a poor match with reality. This leads to the idea of asserting the needed service level for each packet.

One proposal for dynamic allocation of bandwidth at the packet level is the "smart market" proposal by MacKie-Mason and Varian [MacKie]. In this scheme, each packet carries a bid, a price that the user is willing to pay for service. At each point of congestion, all the offered packets are ranked by price, and a cutoff price is determined, based on current capacity, such that only those packets with a bid above the cutoff are serviced. The others are held in a queue, subjecting them to increased delay and risk of being dropped.

This scheme has the desirable feature that it does not lock the user into one service model, such as a constant fixed rate service guarantee, but lets the user bid on each packet as desired. From the perspective of the user, the issue then is what bids to offer, in order to achieve the desired overall service. As was proposed above, what the user cares about (except in real time flows) is not the delay of individual packets, but the overall delivery time of whatever data object is being sent. Thus, to make this scheme useful, it is necessary to determine a linkage between the treatment of individual packets and the resulting overall transfer rate. This objective is possibly made more complicated in this scheme because the user cannot directly know the service he is going to achieve by making a specific bid. The bid is not directly a request for service, but an assertion of price. This raises the possibility that the user will have to hunt for the correct bid in order to achieve the desired overall transfer time[7].

Priority scheduling --

A scheme that has been proposed for allocation of bandwidth among users is to create service classes of different priorities to serve users with different needs. Such a scheme is proposed in [Gupta]. The definition of priority is that if packets of different priority arrive at a switch at the same time, the higher priority packets always depart first. This has the effect of shifting delay from the higher priority packets to the lower priority packets under congestion[8].

What does this mechanism have to do with service differentiation? Slowing down an individual packet does not much change the observed behavior. But the probable effect of priority queuing is to build up a queue of lower priority packets, which will cause packets in this class to be preferentially dropped due to queue overflow. The rate adaptation of TCP translates these losses into a reduction in sending rate for these flows of packets. Thus, depending on how queues are maintained, a priority scheme can translate into lower achieved throughput for lower priority classes.

This might, in fact, be a useful building block for explicit service discrimination, but it is important to note that a simple priority scheme has no means to balance the demands of the various classes. The highest priority can pre-empt all the available capacity and starve all lower priorities, with the consequence that the highest priority user gets much better service than he needed, and the other users

[7]There is another drawback to the smart market scheme, which is that it couples the service model and the price model in a very direct way, which reduces the flexibility that providers have in setting price for service. This issue is explored in [Shenker].

[8]If there is no congestion, then there is presumably no queue of packets, which means that there is not a set of packets of different priority in the queue to reorder. Thus, priority scheduling normally has an effect only during congestion.

get much worse. There is no way to moderate this effect in a simple priority scheme. Some additional usage control must thus be a part of a priority scheme.

The other drawback to a priority scheduler for allocating resources is that by itself it does not give the user a direct way to express a desired network behavior. There is no obvious way to relate a particular priority with a particular achieved service. Most proposals suggest that the user will adjust the requested priority until the desired service is obtained. Thus, the priority is a form of price bid, not a specification of service. This is a rather indirect way of obtaining a particular service; by the time the correct priority setting has be determined, the object in question may have been completely sent. It is much more effective to let the user directly specify the service he desires, and let the network respond.

4. SERVICE PROFILES: RELATING TRANSFER TIME TO PACKET SCHEDULING

What is needed is a mechanism that directly reflects the user's desire to specify total elapsed transfer time, and at the same time takes into account such issues as the vastly different transfer sizes of different applications, 1 byte or 10 million bytes, and the different target transfer times, which may range from tenths of seconds to minutes or hours.

While the network does not know the size of the object the user is sending, or the desired overall delivery time, the user potentially does. If we assume that the user knows these two numbers, simple division yields the needed transfer rate for this object, and this rate becomes the overall service objective for all of the packets that constitute the object.

How might such an overall rate requirement be translated into a sequence of per-packet service requests? One could speculate on approaches that put the desired rate in the packet as a service request, although it is not clear what the network would do with this knowledge. However, in the Internet today, a practical answer can be deduced by noting the behavior of TCP. TCP tries to send as fast as possible, but slows down whenever it receives a congestion signal (which today is a discarded packet). So to control the overall sending rate, one must control the congestion feedback received at the source of the data.

Imagine that the user, for the transfer of each object, computes a minimum rate at which packets must be sent to satisfy the overall delivery objective. This rate becomes a service profile for this transfer, and is then installed in a *traffic meter* at the source. As the packets are sent, the meter flags each packet as to whether it is *in* or *out* of that profile. At any point of congestion inside the network, the packets that are tagged as being *out* are preferentially selected to receive a congestion pushback notification. (In today's routers, this is accomplished by dropping the packet.) If there is no congestion,

there is no discrimination between *in* and *out* packets; all are forwarded uniformly. The router is not expected to take any other action to separate the *in* and *out* packets. In particular, there are no separate queues, or any packet reordering such as priority scheduling. Packets, those both *in* and *out*, are forwarded with the same service (perhaps FIFO) unless they are dropped due to congestion.

If TCP were to send exactly at the specified rate in the service profile, all the packets would be flagged as *in*, thus hopefully avoiding any congestion feedback. However, it is the natural behavior of TCP to speed up if unimpeded, and as the sending rate exceeds the minimum, some of the packets will be flagged as *out*. If the network is congested, those packets may trigger a congestion slowdown. Thus, the TCP will, as before, operate at a higher speed if the network is underutilized, but will slow down under congestion to the desired minimum speed, at which all the packets are flagged as *in*. At this operating point, different users may be getting very different service, depending on how the *in* flags are set in the packets of the various users, which is the service discrimination that the Internet cannot explicitly perform today.

This proposal for an *in/out* flag is thus a way to relate per-packet service requests to overall behavior. It is tied to some extent to the behavior of TCP, although in fact it is more general than that. (If the sender does not adjust its sending rate in response to congestion notification, the mechanism will just continue to discard most of the sender's packets.) It differs from some of the current proposals in that a congested router will allocate service among users, not by delaying some packets more and some less, but by controlling which packets receive congestion pushback indications. As hypothesized earlier, congestion pushback, which triggers TCP rate adjustment, is a much more important factor in the overall service than is the exact delay of individual packets .

5. MAKING SURE THERE IS ENOUGH BANDWIDTH.

There is, of course, no guarantee that just because a user sends a sequence of packets flagged as *in*, that there is capacity to carry them. In fact, any sort of hard guarantee will be very difficult to implement in the Internet. One of the successes of the Internet is its ability to exploit the mixing of traffic from a large number of very bursty sources to make very efficient use of the long distance trunks. To offer hard guarantees is inconsistent with the statistical nature of the arriving traffic, as the discussion of minimum guaranteed capacity illustrated. However, even though the Internet does not offer any guarantees of service, the users do have *expectations*. Experience using the network provides a pragmatic sense of what the response will be to service requests of various sized at various times of day. This idea of *expectation*, as opposed to *guarantee*, is an important distinction.

For the provider, meeting the customer's expectation is a matter of provisioning. Providers will observe actual usage across links in the network to determine needed capacities, making reasonable assumptions about the nature of aggregated traffic. One consequence of the scheme that tags the packets from each user as to whether they are within the desired performance profile is to provide a very clear indication to the provider as to whether the net has sufficient overall capacity. If the provider notices that there are significant periods where a switch is so congested that it is necessary to discard packets that are tagged as being *in*, then there is not sufficient total capacity. In contrast, if the switch is congested, but some of the packets are flagged as *out*, then the situation is just that some users are exceeding their minimum usage target (which is what a TCP will always attempt to do), and so pushing back on those users is reasonable, and not a indication of insufficient capacity.

6. PRICING FOR DIFFERENT SERVICES

Of course, the discussion above is somewhat less than half the story. Once we give the user some means to adjust the level of service, it will be necessary to provide some constraint on the user, lest he just flag all his packets as *in*. An obvious approach is to attach some pricing scheme to the mechanism, so that asking for a better service has a higher price.

A simple form of pricing can be implemented in this scheme without any further mechanism at the switches inside the network. At the point where the user attaches to the network and delivers traffic, the sending of *in* packets can be counted. Associating cost with these packets is a rational basis for pricing, since these packets represent exactly what the user values enough to send during periods of congestion, when the marginal cost of packet transmission is non-zero. The provider can enter into any contract with the user that is mutually agreeable, including charging for actual use, adding a demand component to the charge for large users, or a fixed payment for a particular usage profile negotiated long term.

The tagging scheme also provides a rational means for providers to settle with each other for the capacity that each requires to service its users. By metering the links between providers, and looking at the number of *in* packets carried, the providers can determine how much capacity represents traffic of value that should be carried at times of congestion. This information can inform a rational long-term transfer of payment between providers.

The result of this approach to pricing is that the providers enter into payment arrangements among themselves based on aggregated observation of tagged packets, and for each specific traffic source, the provider serving the sender takes all. There is no per-source inter provider accounting, and no need for accounting mechanism inside the network that tracks the usage of individual users. This approach has

the benefit that providers will probably experiment with pricing schemes for individual users, and would not like to have the price allocation locked to a particular algorithm inside the network.

However, this approach to pricing does not deal in an precise manner what happens when there is an episode of short-term congestion so severe that even the *in* packets cannot all be carried. In that case, which users should be served? One approach, which binds price to service, would be to create a smart market along the lines of MacKie-Mason and Varian[MacKie]. Each packet could carry a bid for service, except in this case the bid is the highest price one will pay to avoid having a packet discarded and thus triggering a rate adjustment in one's TCP. As congestion builds up, one will receive no less than the service one requests until one's bid is insufficient, at which point congestion feedback will force the sending rate below the requested level.

The simpler version of this scheme (as described above) would requires just one control bit in each packet, to flag the packet as *in* or *out*, and leaves unspecified how the price for the service is set. One could wonder whether the more complex tagging scheme with bidding to allocate capacity to those willing to pay the most during periods of congestion would be more efficient, in an economic sense. I will venture a total speculation that the answer is no, and that the simple scheme with only two levels of service will provide an effective allocation of service. The simple scheme, with a one bit tag, provides two critical features. First, it permits different users to specify the level of service they actually desire, and to do so with considerable discrimination, based on how they tag each of their packets, and at what rate they send them. This provides the ability to distinguish users with different needs so that different prices can be set for them. This enhancement alone will permit a great improvement in network utility.

Second, it allows the providers to determine what level of capacity is needed to serve the users. Today the providers provision in a somewhat conservative manner, so that congestion is uncommon. Given the additional information as to which of the packets are acceptable for discard during congestion, the providers will probably continue to provision somewhat conservatively. Only in rare cases will packets marked as *in* be discarded. If this is so, a more complex dynamic bidding system will almost never be useful. Just like the phone system, which only occasionally shows congestion (e.g. Mother's day) there seems little utility in a complex scheme to allocate the network optimally in those cases. This line of reasoning, it should be reiterated, is pure speculation.

7. EXPECTED CAPACITY PRICING

As observed above, one can imagine a number of time scales over which a usage profile could be installed and used. At one extreme, the user could contract with the network on a very dynamic basis to install a usage profile before each transfer. At the other extreme, the user and the network could enter into a long term contract for a profile, which then applies to all of the transfers of that user. Part of the benefit of the tagging scheme is that it does not constrain this sort of decision between the user and the provider. None the less, there are several benefits to an approach based on a long term contract, rather than a highly dynamic scheme. A scheme based on log term contracts, called *expected capacity* pricing, is described in [Clark 95]. In this scheme, a user would purchase a usage profile, called an expected capacity profile, based on the general nature of his usage. For example, a user exploring the web would have need a very different profile from a scientist transferring a sequence of large data sets.

Expected capacity pricing has a number of advantages. First, users with different usage profiles can be charged different amounts, but the price to each user is fixed and predictable, which permits stable budgeting for network use. Many users have expressed the need to have stable prices, and today fixed prices are available. However, the prices today are normally coupled to the peak rate of their access link, and users with a need for high peak rates but a low average rate may find the fee for a high speed link intolerable. By purchasing a high speed access link but a expected capacity that matches their actual useage pattern, they should be able to negotiate a lower monthly fee while still getting the high peak rate.

Second, expected capacity gives the providers a more stable model of capacity planning. If users are permitted to install and use different profiles on demand, the provider must provision somewhat more conservatively, to deal with peaks in demand. This will translate into a higher charge for a usage profile installed dynamically, compared to one that is contracted long term.

Obviously, a long term expected capacity profile must provide some latitude for normal variation in user behavior, and thus must be somewhat more relaxed than a dynamic profile installed for one transfer. But a reasonable speculation is that this degree of imprecision is not an important issue in defining prices to the user. Today, we price networks based on the assumption that there is essentially no constraint, other than the peak rate of the access link, on the worst case user behavior. Even a very permissive expected capacity profile will, for most users today, imply such a restriction on worst case behavior that the price benefit will be substantial. Given the success of today's (non) scheme, there is no reason to think that we need to move to some scheme of very high precision to achieve reasonable user utility. Essentially, entering into a long term expected capacity profile is to view the contract between user and provider as a provisioning relationship. Today we do peak rate provisioning only, and

expected capacity allows for a much wider range of provisioning contracts, with the added benefit that the user can exceed his profile without penalty in times of network underutilization.

Even if providers choose to deal with individual users on a more dynamic basis (a decision which will based on the market success of different approaches), it would seem that between providers, where there is substantial traffic aggregation, that a long term expected capacity profile is the more effective version of this scheme. In other words, the proposal is that where providers interconnect, the arrangement should be the mutual provision of expected capacity for each other, rather than billing based on actual transport of tagged packets. This, again, has the benefit of stable prices, and no need to account for actual packets (expect to detect whether there is adequate provisioning).

It should be noted that the problem of rationalizing inter-provider payments is very complex, and this model only addresses some of the issues.

7. LIMITATIONS TO THIS SCHEME

This scheme, as described, has two key limitations that must be resolved before it could be considered practical. These are the need for receiver payment, and multicast.

To this point, the description of bandwidth allocation has been in terms of the sender of the data. The sender purchases capacity from his immediate provider, which purchases it in turn from next attached providers, and so on all the way to the receiver. In return for this arrangement, the sender is permitted to send packets marked as *in* to the receiver.

In practice, we cannot expect all capacity to be purchased in this way. There are many circumstances in which the receiver of data, rather than the sender, will be the natural party to pay for service. In fact, for much of the current Internet, data is transferred because the receiver values it, and thus a "receiver pays" model might seem more suitable. This assumption may be less universal today; if the World Wide Web is more and more used for commercial marketing, it may be that the sender of the information (the commercial Web server) is prepared to subsidize the transfer. But in other cases, where information has been provided on the Internet free as a public service, it seems as if the natural pattern would be a "receiver pays" pattern. In general, both of the conditions will prevail at different times for the same subscriber.

This pair of patterns somewhat resembles the options in telephony, with normal billing to the caller, but collect and 800 billing to the recipient. But technically, the situation is very different. First, of course, the Internet has no concept of a call; there is no setup phase before traffic is sent, nor any

knowledge inside the network that ties together the sequence of packets that make up the flow. Second, the data flows in each direction are conceptually distinct, and can receive different quality of service. Third, which way the majority of the data flows has nothing to do with which end initiated the communication. In a typical Web interaction, the client site initiates the connection, and most of the data flows toward the client. When transferring mail, the sender of the mail initiates the connection. In a teleconference, whoever speaks originates data.

Abstractly, a mix of sender and receiver payment makes good sense. The money flows from the sender and from the receiver in some proportion, and the various providers in the network are compensated with these payments for providing the necessary capacity. The money will naturally flow to the correct degree; if there is a provider in the middle of the network who is not receiving money, he will demand payment, either from the sender or the receiver. And the resulting costs will be reflected back across the chain of payments to the subscribers on the edge of the Internet[9].

As a practical matter, payments in the Internet today resemble this pattern. Today, each subscriber pays for the part of the Internet that is "nearby". The payments flow from the "edges" into the "center", and it is normally at the wide area providers where the payments meet. That is, the wide area providers receive payments from each of the attached regional providers, and agree to send and receive packets without discrimination among any of the paying attached providers.

What is needed is a way to meld this simple payment pattern with the more sophisticated idea of tagging traffic. The technical issue that must be resolved is that tags in packets flowing from the sender can easily indicate the sender's preference for the importance of the packets, but in a region of the network in which the receiver paid, it should be the receiver, not the sender, that is charged for the packets.

The problem of dealing with receiver payment is made more complex by the Internet mechanism called multicast, which allows one packet from a source to fan out along a tree of routes to a number of receivers. This capability is used today to carry audio and video, both for transmission of single site events, and for multi-site teleconferences. Multicast implies that the cost of a packet should be shared in some way among multiple receivers, who thus reap the benefit of the multicast mechanism. In order to allocate costs for multicast in a rational manner, it may be necessary to add explicit mechanism

[9]Of course, in the future, this whole payment pattern could be inverted. The long distance providers could directly attract the individual subscribers, and then contract with the regional or local area providers to carry their traffic. This pattern now applies when businesses directly purchase long distance telephone service from a provider, who then contracts with a CAP to connect the business to that provider.

inside the network. This is perhaps the most important example of the need for explicit mechanism to support the objective of pricing itself, and the design of this mechanism must be undertaken with considerable care, since the long lead time for changing the Internet extracts a high price for implementing the wrong mechanism.

8. CONCLUSIONS

The service provided by the Internet is a service that mixes a large number of instantaneous transfers of objects of highly variable size, without any firm controls on what traffic demands each user may make, and with user satisfaction presumptively based on elapsed time for the object transfer. We also conclude that while the mechanisms in the Internet seem to work today, a valuable service enhancement would be some means to distinguish and separately serve users with very different transfer objectives, so that each could better be satisfied.

There are two general ways to regulate usage of the network during congestion. One is to use technical mechanisms (such as the existing TCP congestion controls) to limit behavior. The other is to use pricing controls to charge the user for variation in behavior. This paper concludes that it is desirable in the future to provide additional explicit mechanism to allow users to specify different service needs, with the presumption that they will be differentially priced. This paper attempts to define a rational cost allocation and pricing model for the Internet by constructing it in the context of a careful assessment of what the actual service is that the Internet provides to its users.

Key to the success of the Internet is its high degree of traffic aggregation among a large number of users, each of whom has a very low duty cycle. Because of the very high degree of statistical sharing, the Internet makes no commitment about the capacity that any user will actually receive. It does not make separate capacity commitments to each separate user.

The central hypothesis of this paper is that the characteristic of the Internet service most valued by the user is the overall throughput achieved by a user during the transfer of a data object of some size, not the delay of individual packets. Thus, the linkage between the treatment of individual packets and overall user satisfaction is whether a packet triggers a congestion feedback indication, not how much it is delayed. This is because congestion feedback (currently a discarded packet) is taken as an indication to the source that it is to slow its sending rate.

This paper suggests that instead of allocating capacity to users by explicit reservations along a path, we should takes the much simpler step of aggregating all the traffic that is within the usage profile of all the users, as indicated by the tags in the packets, and then viewing the successful transport of this

aggregated traffic as a provisioning problem. This raises the risk that on occasion, a user will not actually be able to receive exactly the expected throughput, but a failure of this sort, on a probabilistic basis, is the sort of service assurance that the Internet has always given, and that most users find tolerable.

Finally, this paper claims that allowing the user to tag packets, because it represents how resources are allocated when they are in demand, represents a rational basis for cost allocation. Cost could be allocated on the basis of actual use, or on the basis of the expectation of use, which is much easier to administer, and again reflects the statistical nature of the sharing that the Internet already provides.

One perspective on this approach to service allocation is that it provides a better limit on the impact of worst-case user behavior, so that users are not disrupted by other users sending in an unexpected pattern. In the current Internet, the limit on any one user is the peak speed of the access link for that user. As noted above, the difference between the traffic pattern of a normal user and the load generated by a constant transmission at the peak rate of the access link may be very considerable, since most users normally generate very bursty and intermittent traffic. In contrast, with this tagging scheme, the provider can limit the user to some prearranged "worst" behavior by limiting the ability of the user to tag *in* packets. Usage beyond that point will be tagged as *out*, and will not interfere with the *in* packets from other users. Again, the provider can offer a range of limits, suitably priced.

The mechanism proposed here, which is the discrimination between packets marked as *in* and *out* for congestion pushback at times of overload, has the virtue that it is simple to implement and capable of implementing a wide range of policies for allocation of capacity among users. It allows providers to design widely differing service models and pricing models, without having to build these models into all the packet switches and routers of the network. Since experience suggests that we will see very creative pricing strategies to attract users, limiting the knowledge of these to a single point, where the user attaches to the network, is key to allowing providers to differentiate their services with only local impact. What must be implemented globally, by common agreement, is the format of the *in*/*out* tag in packets, and the semantics that *out* packets receive congestion indications first. Providers use the level of *in* packets to assess their provisioning needs, and otherwise are not concerned with how, for any particular customer, the expected capacity profile is defined. This design thus pushes most of the complexity to the edge of the network, and builds a very simple control inside the switches. Thus this approach attempts to minimize what we must agree on and deploy in common throughout the Internet, and leaves as much of the total mechanism as a local matter to each provider. (It also permits incremental deployment in parts of the Internet.)

9. ACKNOWLEDGMENT

This paper has benefited from criticism and comment from a number of people. A longer version of the material was presented at the M.I.T. Workshop on the Economics of the Internet in spring of 1995. Many of the participants there provided valuable feedback on the material. I would like to thank Scott Shenker for his comments on that version of this paper, and particularly to thank Marjory Blumenthal, who managed to read and provide detailed comments on two earlier version of that paper . Both of these colleagues have taught me a considerable amount of remedial economics.

REFERENCES

[Jacobson] V. Jacobson, *Congestion Avoidance and Control*, Proceedings of ACM Sigcomm '88, Stanford, Calf.

[MacKie] J. MacKie-Mason and H. Varian, *Pricing the Internet,* presented at Public Access to the Internet, JFK School of Government, May 1993. Latest version available via ftp://ftp.econ.lsa.umich.edu/pub/Papers/Pricing_the_Internet.ps.Z

[Willinger] W. Willinger, M.S. Taqqu and D.V. Wilson, *Self-Similarity Through High-Variability: Statistical Analysis of Ethernet LAN Traffic at the Source Level*, Proc. ACM SigComm 95

[Demers] A. Demers, S. Keshav and S. Shenker, *Analysis and Simulation of a Fair Queuing Algorithm*, in Journal of Internetworking: Research and Experience, 1, pp. 3-26, Also in Proc. ACM SigComm '89.

[Clark] D. Clark, S. Shenker and L. Zhang, *Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanism,* in Proc. ACM SigComm '92, Baltimore.

[Shenker] S. Shenker, D. Clark, D. Estrin and S. Herzog, *Pricing in Computer Networks: Reshaping the Research Agenda,* in Proc. of TPRC 1995.

[Gupta] A. Gupta, D. Stahl and A. Whinston, *Managing The Internet as an Economic System*, available via http://cison.bus.utexas.edu/ravi/pricing.ps.Z

[Clark 95] D. Clark, *A Model for Cost Allocation and Pricing in the Internet,* in "Internet Economics." L. McKnight and J. Bailey, eds. Journal of Electronic Publishing, University of Michigan Press (forthcoming). http://www.press.umich.edu:80/jep/