Mitigating the coming Internet crunch: multiple service levels via Precedence*

Roger Bohn
rbohn@ucsd.edu
University of California, San Diego
La Jolla, CA 92093-0519

Hans-Werner Braun
hwb@sdsc.edu
Kimberly C. Claffy
kc@sdsc.edu
San Diego Supercomputer Center
San Diego, CA 92186-9784

Stephen Wolff
steve@nsf.gov
National Science Foundation, DNCRI
Washington, D.C.

March 22, 1994

Abstract

The current architecture and implementation of the Internet assumes a vast aggregation of traffic from many sources and stochastic distribution of traffic both in space (traffic source) and time (burstiness of traffic volume). Given this general assumption, Internet components typically have little if any ability to control the volume and distribution of incoming traffic. As a result the network, particularly from the perspective of the router, is vulnerable to significant consumption of networking resources by high-volume applications, with possibly little stochastic behavior, from a few users. This often impacts the overall profile of network traffic as aggregated from many clients. An example is the continuous flows introduced by real time applications such as packet audio, video, or rapidly changing graphics.

This situation creates a time window where applications exist on a network not designed for them, but before an appropriately architected network can augment the current infrastructure and cope with the new type of workload. We propose a scheme for voluntarily setting Internet traffic priorities by end-users and applications, using the existing 3-bit Precedence field in the Internet Protocol header.

Our proposal has three elements. First, network routers would queue incoming packets by IP Precedence value instead of the customary single-threaded FIFO. Second, users and their applications would voluntarily use different and appropriate precedence values in their outgoing transmissions according to some defined criteria. Third, network service providers may monitor the precedence levels of traffic entering their network, and use some mechanism such as a quota system to discourage users from setting high precedence values on all their traffic. All three elements can be implemented gradually and selectively across the Internet infrastructure, providing a smooth transition path from the present system. The experience we gain from an implementation will furthermore provide a valuable knowledge base from which to develop sound accounting and billing mechanisms and policies in the future.

1 The problem

In such situations of moderate scarcity, however, not all people can have whatever means of communications they want. The means are rationed. The system of rationing may or may not be equitable or just. There are an infinity of ways to partition a scarce resource .. egalitarian .. meritocratic .. [recognizing] privilege .. cultural values .. [rewarding] skill and motivation, as that which allows communications institutions to earn profits that depend on their efficiency. –**Ithiel de Sola Pool, Technologies of Freedom, 1983, p.240.**

*Hans-Werner Braun and Kimberly Claffy acknowledge support from the National Science Foundation for studying engineering and architectural issues of the National Research and Education Network.

Most architecture and instrumentation for accounting and traffic control in the Internet reflect its historical status as a typically government bulk-funded service for the academic community. It has been a research environment with usage-insensitive costs that are often transparent to the end-users. As a result the current Internet architecture is not conducive to allocating network resources among multiple entities or at multiple qualities of service. Presently there is no incentive to limit the use of bandwidth, and the appetite for bandwidth is growing far faster than the Internet can support. As traffic grows, and new applications with fundamentally different traffic characteristics come into widespread use, resource contention will become a problem.

Earlier networks, such as the ARPAnet, relied on inflow control into switching nodes within the network to reduce the likelihood of resource contention. However today's Internet is inherently based on a datagram architecture which aggregates traffic from many endpoints through distributed packet forwarders, few if any of which implement admission control. Most entrance points into transit networks cannot provide significant back pressure to peer points that inject more traffic than the transit network can handle. This situation pervades the Internet environment, leaving end systems able to unfairly monopolize available bandwidth and cause significant congestion in the transit networks they use.[1]

During the mid-80s on the 56kbps NSFNET backbone, congestion developed to a dangerous degree. In response the NSFNET engineers deployed an emergency measure to provide certain interactive network applications, specifically telnet, preferential treatment over other traffic. The priority transit allowed interactive users requiring better network responsiveness to continue working under highly congested circumstances. At the same time the NSFNET backbone established support for separate queues in the routers according to the *IP Precedence* value in the IP header field. Because the backbone administrators did not have any way to provide an incentive to not use the highest priority, they did not publicize the priority-based treatment of traffic, and end users did thus not know it was possible to give high precedence to other applications.

When the NSFNET was upgraded to T1 capacity, offering a 24-fold bandwidth increase and a richer topology, the designers did not re-introduce the priority queuing for end-user traffic. The new infrastructure used multiple queues only to differentiate between user traffic and network management traffic. An overabundance of bandwidth rendered superfluous the use of multiple queues. In the case of the NSFNET backbone, the project partners bore all the costs of maintaining this bandwidth ahead of demand. The subsequent upgrade to the T3 network further exemplified this method of coping with network congestion by increasing the bandwidth and switching capacity ahead of demand.

However, today software developers are building advanced network applications which can consume as much bandwidth as network operators provide. In particular, real-time applications using packet voice and video do not exhibit the same stochastic burstiness characteristics of more conventional applications such as interactive login sessions and electronic mail. The latter traffic is typically aggregated across a large number of individual sources with relatively modest resource requirements [4]. Real-time applications rather require prolonged delivery of large amounts of traffic in near real-time, and thus continuously consume significant amounts of bandwidth. Clearly usage of such applications will not scale in the current Internet architecture, which has no facilities to support continuous high volume services for many individual connections, as it does with the current traffic cross-section.

Many recent videoconferencing applications require 125kb/s to 1Mb/s over LANs. People accept lower quality and bandwidth over WANs because of cost. For example, CU-SeeMe developed at Cornell University uses compression to reduce the bandwidth requirements [2] to under 100kb/sec [3]. Current Macintoshes come instrumented with a microphone and hardware enabling transmission of audio across the Internet; within a year it will be feasible for an undergraduate with a $2000 Macintosh AV and a $500 camcorder to send real-time video to friends on another continent. (How much will arrive at its destination will depend on actual network contention.)

---

[1] We will use the term *bandwidth* loosely, to include both transmission and switching. At present and in the near future, which factor is most constraining will depend on the environment: typically the bottleneck will be switching capacity for environments which can support high bandwidth, such as national backbones using large telecommunications pipes or local area networks using FDDI technology. In contrast the intermediate transmission between a LAN and national backbones is more often constrained by link or transmission bandwidth in campus or regional networks.

[2] The exact bandwidth requirements depend on compression techniques used and the motion within the video frame.

Most video and audio applications are currently still in a prototype phase; they enjoy a wide-area infrastructure which is not yet subject to overwhelming congestion. Once network resource congestion occurs, however, parts of the infrastructure may become quite unpredictable due to switching contention. One way to accelerate toward that point is continued wide-spread deployment of these applications. Their success and popularity bodes ominously for an infrastructure not able to: distinguish among certain traffic types; provide more than a best-effort guarantee to datagram traffic; or upgrade in a time efficient way towards an availability of higher bandwidth (if only due to lack of accounting and billing mechanisms to enable cost recovery).

Appropriate ways to deal with the situation involve considerations of fairness to network clients, whether to penalize larger flows first, or implement preferred service classes, latency-sensitive applications, etc. No solution will satisfy every constraint, especially if the scale of the answer must encompass the integrity, operability, and manageability of the global infrastructure.

At present, users have no motivation other than altruism to conserve their use of bandwidth. Ewing and Schwartz measure that more than half of all FTP file transfers do not use compressed format [5]. If the current use of FTP is any indication of the awareness of needless bandwidth consumption, video will make matters worse. There is plenty of evidence of widespread dissemination of video streams pointing at trees outside people's offices, or even empty offices at night where people have neglected to turn the video transmission off as they leave their office in the evening. While an FTP session eventually completes, or a telnet session essentially exchanges no data without a person actually typing, a video stream can continue to transmit large volumes of data indefinitely. Indeed, it is difficult to overestimate the dramatic impact which digital continuous media will have on the Internet fabric.

Digital continuous media traffic profiles are fundamentally different with volumes of much higher mean and lower variance, and are not designed for sharing as equitably since they typically use UDP instead of TCP.[3] We lose the "sharing" part of stochastic sharing, and although potentially mitigated by compression techniques, with no back pressure to the traffic sources by intermediate systems nor incentive to implement bandwidth-economizing mechanisms, the Internet is defenseless against them. Because of the lack of inflow control and the lack of cost recovery mechanisms based on actual traffic composition, the infrastructure critically depends on the conscientious behavior of the end systems, and their users, to utilize well-mannered mechanisms for bandwidth requisition, such as window-based flow control and backoff in the face of detected congestion. It is only such mechanisms, which window-based transport protocols such as TCP exhibit, that can render relatively predictable the traffic patterns at aggregation points within environments such as today's Internet.

In fact, one may argue that the impact of the new, specifically real-time, applications will be disastrous: their high bandwidith-duration requirements are so fundamentally at odds with the Internet architecture, that attempting to adapt the Internet service model to their needs may be a sure way to doom the infrastructure. Proponents of this argument also might claim that there is thus no point in implementing usage-based accounting or congestion control mechanisms in an infrastructure which will break if it really has to support those applications to any realistic degree. Those applications may have to be integrated into a different environment, specifically one architected for greater service integration, and one which may eventually absorb the Internet. In the meantime, however, they should leave the Internet to do that for which it was architected.

Despite the pleasant thought that somehow the increasing population of high bandwidth-duration applications will find a more appropriate environment and leave the Internet for traditional applications, we consider it at the very least a moot argument. Such real-time applications exist even on today's Internet, and have already exhibited a detrimental impact on the traditional shared traffic that is so far carried largely by applications using relatively small TCP flows for transport layer communication.[4]

[3] UDP is a lesser-used transport protocol designed for higher volume traffic, with much less error checking. Unlike TCP, it does not automatically back off its transmission rate in the face of congestion, and can thus saturate a path.

[4] We admit that the growth of real-time applications may be self-limiting since each new user contributing to the congestion degrades performance for themselves as well as everyone else, and while degraded performance only makes e-mail late, it makes real-time video/audio streams useless. Nonetheless, we note that such a process will yield a rather high "equilibrium" level congestion.

2 Strategies for solution

We see four possible strategies for dealing with these problems.

1. Do nothing, and wait for congestion to cause poor service which chases users to more expensive networks with better service. In the meantime, the nature of existing FIFO queuing implies that congestion will result in random allocation of delays and lost packets, meaning that many users will experience lower quality of service, even those who request only very small amounts of resources.

2. Implement sophisticated usage tracking, real-time pricing, or resource reservation schemes. We certainly expect that in the long run the community will redesign and reengineer networks to support mixed traffic profiles, including real-time continuous media traffic, with orders of magnitude more bandwidth capacity, complex resource reservation, accounting and pricing schemes. However these proposals may well require major redesign, preceded by considerable discussion [2] [6] [8] [7] [9]. In the interim we have a time window in which the Internet must accommodate traffic for which it was not designed.

3. Hope that service providers will be able to continue to upgrade the network ahead of demand, with no transference of resource consumption and upgrade costs to users. Never viable in the long-term, this strategy was tenable only during the period of plummeting prices for T1. That is, a 24-fold increase in capacity was affordable well before the possibility of a 24-fold increase in demand. A similar improvement in cost-performance for T3 leased circuits is unlikely to occur soon.

4. Provide an interim solution which can be implemented quickly to alleviate the impact of new patterns of traffic behavior. We propose such a solution here.

Many believe that Internet dependence on the altruism of the end systems has become unrealistic, and that protection needs to become a network-based mechanism implemented in routers, especially given the most recent TCP optimizations which support very large windows and thus the consumption of large fractions of total network bandwidth. Even the version of TCP [1] which supports these features took several years after the initial TCP deployment before widespread implementation. Similarly, it will take years to address the problems with new protocols and to broadly integrate appropriate solutions. Architecturally we believe we need a mechanism for the existing Internet that is consistent with the existing protocols and can be implemented with only minor changes to existing software.

In pursuit of incremental progress, we propose a strategy for the *existing* Internet, not in order to support new real-time multimedia applications, but rather to shield, albeit in limited way, the existing environment from applications and users whose behavior conflicts with the nature of resource sharing. In the future, the Internet needs its own self-defense mechanisms; our proposal is only an interim measure to address current and short term problems.

We targeted four goals in our proposal. First, we aim for an increased *effectiveness* in making use of available bandwidth. In particular we strive for multiple service classes on a single physical network. Second, we anticipate a need for the proposed scheme to be *socially accepted*, since one cannot easily mandate behavior on the Internet; one must rely on peer pressure or other incentives. Third, related to the second, we want a scheme which is *equitable*, at least in some loose sense. Fourth, it should be *easy to implement* and reward early local implementors even before the full system is in place throughout the global infrastructure.

3 Proposal

We propose a scheme for setting voluntary priorities for Internet traffic, and provide incentives to users to limit their use of high precedence levels. Where peer networks interconnect (e.g., at either a Federal Interagency EXchange (FIX), a Commercial Internet EXchange (CIX), or more broadly at generic Network Access Points (NAPs)), the mutual treatment of precedence levels is, potentially at least, a further item for negotiation among the connected networks that exchange routes and traffic. We intend this proposal to serve to catalyze the community into greater consideration of issues surrounding resource consumption in the network itself, rather than to necessarily provide a solid answer.

Our proposal requires cooperation of designers of communications software, who would set appropriate default priorities for transmitted traffic. Our proposal involves three components. First, the application layer under user control will place the precedence level for packets into the 3-bit precedence field of the IP header. Second, Internet routers would maintain multiple queues, and weight service toward packets in the higher priority queue. [5] Finally, network service providers (NSPs) and local network administrators may monitor traffic entering or leaving their systems, and provide incentives to users to limit their use of high precedence levels.

3.1 Precedence value assignment

Assigning precedence values will ideally occur by loose consensus within the Internet community. We do not advocate tying a precedence value directly to an application type (or TCP port), or more abstract qualifiers that lack scalability in the global infrastructure, but rather to the qualities of the individual flow, as judged by the user and system administrators of the end systems.

Because the IP Precedence field is largely unused at this time, most packets today contain a default value of zero. Packets with this zero value would receive lowest priority, whether they have that value because they are not participating in the scheme or because they choose the lowest priority. The default priority for bulk traffic would be 1. We propose discouraging use of the highest precedence value, 7, reserving it for traffic surrounding issues of network management, routing or other traffic considered critical.

Software developers should set default priority levels for the IP precedence field as they develop and distribute their applications, based on some general guidelines. Users of the applications would then be able to modify the priority, either on a per session basis or permanently. Criteria to modulate priorities could include:

- latency sensitive (e.g. *telnet* is high priority; *ftp* bulk transfer and *netnews* are low priority)
- low volume (e.g. *ftp* directory listing transfers are high priority; color video is low priority)
- sponsored (e.g., agency traffic across its own infrastructure could be high priority, while recreational game playing should use low priorities)

The priorities could vary within the same application; some may even dynamically vary within a session. For example, e-mail on behalf of a mailing list could use lower priority than e-mail from one individual to another. Table 1 shows an estimate of current Internet backbone traffic in different categories.[6] One could use this table to easily classify more than 35% of traffic as low priority. Although one could simply have the routers assign priorities according to Table 1, we do not advocate a situation where a network owner would dictate what traffic travels on its network.

Developers and users will implement use of the precedence field only as they so desire. As described earlier, the current default setting for IP Precedence is typically 0, which is the lowest priority defined in the IP specification. If users encounter no problems with their priority 0 traffic, they will see no reason to change. End users who want better service will be able to obtain new versions of applications that default to a higher IP Precedence value, and perhaps allow the end user to modify the value further.

Obviously, skilled system users will be able to rewrite or override the priority set to any value by their applications. In the future, however, most users of Internet will not have the necessary expertise to do this themselves. Moreover in some cases (such as netnews) the application operator will have little or no incentive to cheat by assigning their own outgoing packets a higher priority. We will discuss more elaborate methods of fostering compliance with priority guidelines in Section 3.3.

[5] Whether lower priority queues starve when higher queues are non-empty is up for debate [6] [8]. We assume that they do in the numerical example of Figure 1.

[6] Based on monthly data for September 1993 as maintained by Merit, Inc. on `nis.nsf.net`.

Table 1: Example stratification of traffic into precedence levels

priority	Suggested default priority (percent of traffic)						Sum
7	ntp 0.3	icmp 1.8	bgp 0.1	egp 0	ospf 0	routed 0.1	2.3
6	tn 16.5	rlogin 0.7	X0 0.9	X1 0.0	X2 0.0	X3 0.0	18.1
5	ftp-ctrl 1.9	talk 0.6	gopher 1.8	www 0.5			4.8
4	dns 5.8						5.8
3	baseline 22.7						22.7
2	ftp-data 22.2	IP-prot4 4.5					26.7
1	smtp 7.4						7.4
0	irc 2.4	nntp 8.8	vmnet 0.8	uucp 0.2			12.2

file transfer: ftp-data gopher www
interactive: tn rlogin X0 X1 X2 X3 talk ftp-cntl
e-mail/news: smtp nntp irc vmnet uucp
network mgt: ntp icmp bgp egp ospf routed dns
misc: IP-prot4

3.2 Routers

The second component of our proposal requires that administrators update router software to maintain multiple queues based on IP Precedence. The effect of the precedence value is that when resource contention occurs, it affects mainly the lower priority traffic. This scheme does not require global changes; routers which do not implement the scheme merely treat all traffic as before, at the same priority with FIFO-based queueing. But the scheme can achieve substantial local improvement in the face of congestion. Network service providers will have the opportunity to selectively enable IP Precedence based queuing as they perceive a need, and even disable it in case of misuse. The NSPs thus have a great deal of flexibility. As soon as a few routers implement multiple queues (especially on the backbone), the incentive for users to implement precedence in their packets is expected to increase significantly.

3.3 Soft quotas on total priority

The third part of our proposal involves an opportunity for network service providers to set soft quotas on the total volume of traffic by specific IP Precedence levels. Between peer networks, quotas will have to be negotiated through settlements, which although violating the spirit of the "no settlements" policy of some interconnections (e.g., the CIX), will be necessary to control and limit the import of congestion from a correspondent network. In the more prevalent non-peer case, each client of an NSP may receive from the service provider(s) from whom it buys service a quota on the weighted sum of the priority values in its packets per unit time. We suggest a formula such as:

$$Q = \sum_{i=2}^{6} x_i \alpha^{i-2}$$

where Q is the total quota used by the customer; x_i is the number of packets sent with priority i during the metered period; and α is a parameter greater than 1; we suggest $\alpha = 2$. Alternatively, a bandwidth-limited system may set the quota in terms of transmitted bytes, or perhaps a weighted average of bytes and packets.

Note that we do not sum over i = 0,1, or 7. In this scheme one can send both priority 0 and priority 1 traffic freely since they have no effect on the value of Q. Only network management software will send priority 7 traffic; others using that priority would incur penalty by a special mechanism.

One fear is that users will simply set all their traffic to the highest priority. The purpose of the quota is to set loose incentives, monitored after the fact. If a customer exceeds its quota, the local network service provider assesses an after-the-fact penalty, such as an increase in connection charge for the next month; a more careful monitoring of traffic from that client; or if the abuse persists a threat to terminate services to the customer. The NSPs will specify the quota in the contracts with their users (and in contracts between the NSP and its neighbors and backbones). If a customer finds that it is exceeding its quota regularly, it has three choices: negotiate a higher quota, put and enforce quotas on its subordinate units, or pay the penalties assessed on it.

Each service provider decides what original quota to allocate to each customer, with obvious heuristics such as assigning them in proportion to pipe capacity that attaches the customer, or in proportion to total packets sent the previous year.[7] Customers could also pre-purchase a higher quota.

We propose that quotas be handled in a decentralized way, with measurement and enforcement only at gateways between networks, and only as desired. For example, suppose that backbone N gives a quota to NSP R. R can ignore the quota until its traffic (whether original customers or through gateways) begins to approach the quota. This recurses all the way down to end users. End users then have a direct or indirect incentive to limit their use of high priority packets. Each level chooses its own method of enforcing the quotas for the next lower level. Thus for a long time Internet will have a variety of coexisting quota systems. For some systems, peer pressure and informal mechanisms (such as publicizing a list of offenders) may be the most cost-effective means of enforcement. Metering, when used, has several simplifying characteristics:

- only occurs at entry points;
- not real-time but rather examined after the fact (real time enforcement at this time is unlikely);
- may be done by random sampling (sampling would in fact be necessary, in conjunction with post analysis, if a service provider wants to monitor and penalize for misuse).

Note that we do not tie this scheme directly to billing or pricing. Over time we imagine tying multiple service qualities to more direct monetary incentives for compliance with the otherwise voluntary standard, such as a "pay per priority" scheme. Again, each individual service provider has this choice, and different systems may coexist even in a single network.

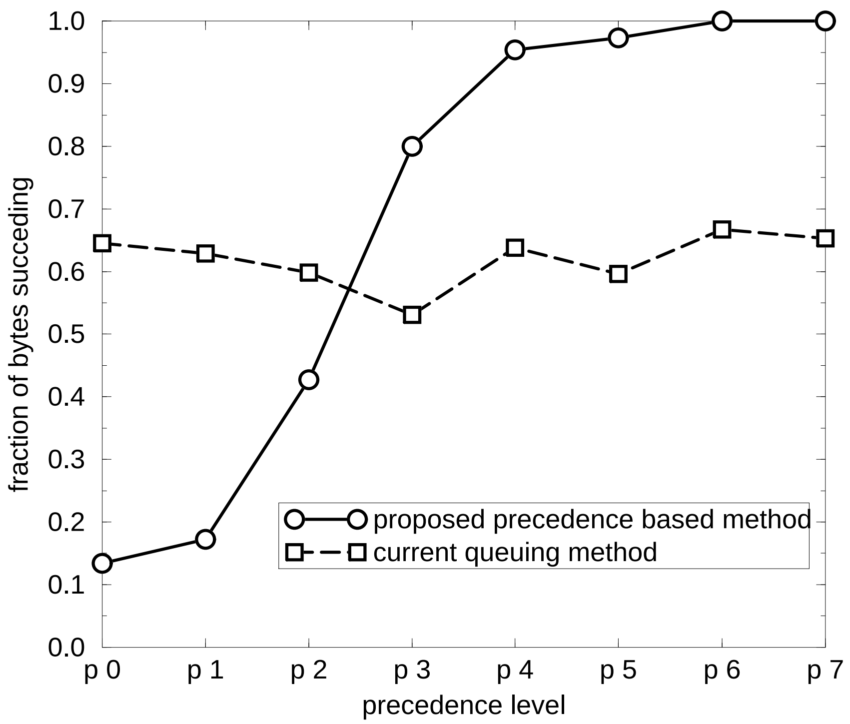
We illustrate with an example. A university may receive a bulk quota from the regional network to which it attaches. The campus network operations center may reallocate the overall quota to faculty, students, and staff, or to different departments, according to established criteria at that campus. Some universities may use a very egalitarian system, while others may tie the quotas to overhead payments or academic rank. Tracking quota usage by individuals would be too expensive, but various proxies and limiting mechanisms are possible if desired. For example the network operations center may impose control, or at least guidelines, on standard software used (e.g., a special version of NCSA telnet). The NOC can set filters according to user ID, or even in principle take the extreme step of automatically reducing packet precedence levels before the packet enters the regional network.

3.4 Illustration

To illustrate how precedence might work, we ran a simple simulation, using a one-hour trace of traffic at an inflow point of the T3 (45 Mbps) NSFNET backbone during March 1993. We divided the trace into one-second buckets, and assumed that the capacity each second was limited to 193,000 bytes, i.e., T1 (1.544

[7] We discarded the idea of setting the quota proportional to current usage, because that provides an incentive to flood the network with priority zero packets.

Figure 1: Effective transmission success by precedence using a congested system (simulates precedence stratification in Table 1 on a T1 connection using a busy one-hour packet trace of traffic from UIUC connection into T3 NSFNET backbone during March 1993)



Mbps) capacity. If more than 193,000 bytes arrive in any second, the simulated router will drop some packets. Although this traffic was from the T3 network, we simulated its transmission through a link of T1 capacity to assess artificial congestion, allowing the illustration of the effect of our proposal. For this line capacity and our data set, the router dropped about 41% of the arriving bytes. We are thus simulating a heavily loaded system.

We compared the loss performance of two queuing rules: FIFO allocation, as routers now perform, and precedence allocation according to the priority assignment in Table 1. For the latter scheme we used the application information based on IP protocol and TCP/UDP port number of each packet in the trace, and simulated its transmission had it carried a precedence value according to Table 1. Figure 1 plots the success rate for all transmitted bytes for each precedence level. The dotted line indicates that under FIFO service (the present system), the router drops approximately the same fraction of packets from all precedence levels.[8] The solid line indicates that in a precedence-based queueing system, in which a higher precedence packet always preempts a lower precedence packet, packets of highest precedence are subject to zero loss, while packets of the lowest precedence suffer extreme attrition.

This experiment represents just one example of a simulated environment, and other scenarios may be more reasonable depending on constraints of a particular environment. Examples include: (1) assuming packets rather than bytes were the bottleneck; (2) use a queueing rule that does not completely starve lower priority queues when serving higher priority queues; (3) apply an alternative stratification to that of Table 1.

4 Special issues

Several specific cases invite questions. An obvious example is reverse traffic, e.g., that of anonymous FTP servers. Since the quota is based on the source of the traffic, the FTP server could quickly draw down its quota if it sent requested files at high priority levels. One mechanism to address this issue is to provide FTP clients with user accounts that receive a priority quota as part of their account. Anonymous users would generally receive low IP Precedence values on their traffic, which should be an adequate approach to deal with that situation.

Defining a better user account structure for widespread dissemination of files may also be useful in other situations, e.g., high volume research between separated organizations (e.g., supercomputer centers). Owners of user accounts would, in principle or in fact, receive quotas which they could then allocate as needed. A flexibly defined user structure would also address the issue of reverse traffic at the packet level, e.g., TCP acknowledgement (ACK) packets, by qualifying outgoing packet priorities based on the actual user.

Another issue is whether there is incentive for interconnecting peer networks (i.e., at NAPs) to discriminate against competitors' packets e.g., by placing them on low-priority queues. We see no such incentive except possibly when a network is receiving pure transit traffic from a communicating peer. That is, if an NSP receives a packet with either the source or destination address being one of the NSP's own customers, it is presumably in the NSP's best interest to treat the packet neutrally by respecting the assigned precedence value. If, however, neither the source nor destination address is a customer of the NSP, it might seem advantageous to handle the packet at lower priority, either by rewriting the precedence field or (more likely) by shunting it to a lower-priority queue. But when considering the interconnected web of NSPs that form the Internet, this situation appears to be a form of the Prisoners' Dilemma, so that "doing the right thing" results in better performance for all participants. Since flows at NAP's are bidirectional, a simple reciprocal policy will encourage cooperation.

5 Future evolution

The system may evolve over time in several directions, depending on the reception within the community. It will certainly affect the role of application software developers, who will modify programs to provide the

[8] In our data set, some types of traffic were burstier than others, leading to somewhat varying attrition rates under a FIFO scheme.

end user with partial although not necessarily total control over precedence, as well as for network service providers that would be able to offer a variety of levels of service qualities.

Application developers may enhance their software to optimize use of fixed quotas, while staying within the general scope of priority assignment guidelines. Some of these changes will be helpful to total system performance, while others will be neutral or detrimental. For example, time shifting of bulk transmissions to times of less congestion helps everyone, but is non-trivial when considering time zones across the entire globe. Monitoring transmission delays and setting precedence dynamically to procure high bandwidth is neutral. Violations by small numbers of highly skilled users does not vitiate the overall value of the proposal; if the violations become egregious, they will require detection and tracing.

The precedence-setting scheme may become arbitrarily sophisticated. One useful enhancement would be to dynamically adapt the IP precedence level to actual user requirements. For example, the user would raise the precedence level for interactive or urgent e-mail traffic requiring immediate response. The user would lower the precedence level for file transfers or other deferable services.

The quota formulas could be sensitive to the time of day, destination, or other factors. One could envision a secondary market in precedence points, as customers engage in side trades among themselves. Monetary value attached to precedence points would provide another signal to service providers to upgrade their networks.

In any case, the strategy we propose will provide an incentive directly to the end user to be sensitive to network resource consumptions and to make more efficient use of available bandwidth, whether with compression techniques, user-friendly delayed-transmission methods (e.g. "*ftp* this file within four hours"), or other mechanisms. Downstream customers could implement such mechanisms gradually, as their upstream providers find the need to provide an incentive to do so.

6 Comparison with other approaches

The theme of any approach to prioritization of traffic is that if congestion occurs, a logical decision of what to drop or delay is superior to a random one. Figure 1 provided emprical confirmation of this claim using a recent packet trace from an entrance point into the NSFNET backbone. Our scheme does not require a central standard-setting; the IP precedence field is already a standard, albeit a non-enforced and essentially unutilized one. It also does not require central enforcement. As long as there is no resource contention, there is no need to prioritize traffic. The only question is that as congestion occurs, who gets hurt and how much.

An important characteristic of our proposal is its decentralized nature. Players, whether network users, clients, or service providers can participate in the scheme, via implementation of their portion, or not, as they choose. Players can also choose whether to impose incentives to lower level customers. Incentives can be of many different kinds; quotas are only one alternative. Some networks could decide to bill by priority, rather than set quotas. Each decision is between the provider and its customers.

Other approaches favor classifying traffic by more abstract definitions, such as local funding agencies or specific applications being used. As the Internet matures to a market driven environment based on varieties of funding sources and billing arrangements, as well as multitudes of varying requirements for millions of users and thousands of applications, we believe that qualifying traffic priorities according to applications and users will not scale to a global environment, largely due to the inability to manage and administer such an environment while ensuring global consistency. Decentralized arrangements in client-provider relationships without global impact would likely be easier to implement, operate, and manage. The implementation we suggest here is only one example of a step in the direction we advocate. It is by no means a permanent solution, but gears the community toward the use of multiple service levels, which we view as the essential architectural objective.

7 Summary

We feel that the cost and priority structure which many users have enjoyed thus far, transparent to their behavior and often facilitated by government bulk subsidized funding, is becoming untenable in the current environment that is rapidly evolving towards a market driven infrastructure. Presently users perceive little if any incentive to limit their use of network bandwidth, and the appetite for bandwidth is growing far faster than the government can reasonably support. We propose a strategy to address this situation in the short-term, taking advantage of an existing but unused aspect of the Internet Protocol (IP).

On the surface, the technical changes required for our proposal are twofold: software in Internet routers would require modification to support the IP Precedence field for queuing; and participating application software would also need modification to set non-zero values of the IP precedence field. However we see the social obstacles as at least as great as the technical ones. Suggesting voluntary compliance with a policy to lower one's own favor in the eyes of the network may seem absurd in the Internet community. The present decentralized structure of the Internet has not only allowed incredible technological advancements in application technology (video, audio) and overall network usability (gopher, mosaic), but has also had major social impacts on the notions of individualism and autonomy. In short, users who inject real-time video may be able to clog the backbones no matter what. Incentives are necessary to address such behavior, while at the same time still allowing the Internet to evolve.

However, we also see the potential for the culture to shift somewhat, as people see lower performance for small applications (e.g. massive video streams slowing down simple ftps). Tracking and publishing statistics on usage and congestion can also raise awareness and perhaps accelerate this cultural shift. We advocate phasing in an implementation, over several months and possibly years. A typical Internet communication may involve two campus networks, two regional networks, and one or more very large scale backbone networks, for a total of five or more concatenated networks. Yet if any one of them implements precedence queuing, performance will improve. That is, incremental implementation yields incremental improvement; there are no "flag days", and implementations are backward-compatible with the existing Internet technology. We anticipate that long before network and application conformance reaches 80%, or even if it never does, the system will be providing substantial benefits to those users and routings that are compliant.

We recognize that an IP Precedence-based service, even in conjunction with accounting and/or charging, is not going to protect the Internet from applications that do not fit the basic premise of the current Internet: stochastic sharing. That is, traffic priorities, or even accounting, are not a solution to this problem. But accounting can help this problem in the transitory phase of the current environment. In addition to billing applications, maintaining statistics for the variety of multiple services can also yield a better understanding of network demands, a valuable knowledge base to develop interim policies, capacity planning, and fairer and potentially more competitive cost recovery in the future.

References

[1] D. Borman, R. Braden, and V. Jacobson. Tcp extensions for high performance. RFC 1323, obsoletes RFC1185, 05/13/1992.

[2] H.-W. Braun, K. Claffy, and G. C. Polyzos. A framework for flow-based accounting on the Internet. In *Proc. Singapore International Conference on Networks (SICON'93)*, pages 847–851, September 1993.

[3] S. Brim. personal communication, October 25 1993.

[4] K. Claffy, G. C. Polyzos, and H.-W. Braun. Internet traffic flow profiling. UCSD TR-CS93-328, SDSC GA-A21526, November 1993.

[5] D. J. Ewing, R. S. Hall, and M. F. Schwartz. A measurement study of internet file transfer traffic. Technical report, University of Colorado, Boulder, 1992.

[6] S. J. Floyd. Hierarchical link sharing. Technical report, LBL, August 1993.

[7] J. Mackie-Mason and H. Varian. Pricing the Internet. Technical report, U. Michigan, May 1993.

[8] Scott Shenker (Xerox PARC), David C. Clark (MIT), and Lixia Zhang (Xerox PARC). A scheduling service model and a scheduling architecture for an integrated services packet network. Technical report, USC, Xerox Park, MIT, March 1994.

[9] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala. RSVP: a new resource reservation protocol. Technical report, USC, February 1993.