Avoiding Network Capacity Collapse*

John Kristoff
DePaul University

To appear in SANS 2001

March 12, 2001

Abstract

A number of educational institutions were taken by surprise when an easy to use MP3 file sharing application called *Napster* quickly arose to make use of all available Internet capacity. Many institutions have been using various techniques to manage their Internet capacity more pro-actively since the first file sharing applications became available. Some institutions simply banned the use of such applications while others did nothing. Still other institutions altered various network configurations and tried new products from vendors to adjust traffic usage in one form or another. This paper briefly details some of the various technical solutions that might be used to pro-actively manage Internet capacity as well as some of the potential pitfalls in each case.

1 Introduction

The Internet has seen its fair share of killer applications, but not until recently have these newest killer applications brought the concept of managing Internet bandwidth usage to the forefront of challenges facing educational institutions large and small.[1] This paper focuses on avoiding the problems these new applications can inflict upon an institution's network. Here we refer to this undesirable result only as capacity collapse. In this context we're concerned with an institution's Internet link becoming overly congested due to scarcity of capacity. A collapse will occur when large percentages of traffic are dropped due to congestion, reducing the *goodput* and response time an application sees.[1]

File sharing using FTP has been around since the earliest days of the Internet, but with the advent of applications such as Napster, CD-ROM burning hardware and high speed connectivity into the dorms and homes it became easy to move high volumes of large files from one host to another.[2][3] Putting the copyright issues aside, network traffic growth as a result of these newest file sharing applications has forced many educational institutions to deploy various solutions that monitor, block, slow or prioritize traffic which traverses links most affected by increased traffic demands. Typically the link most susceptible to capacity collapse is an institution's Internet connection, hence this paper will focus on solutions that manage capacity between an institution and the public Internet. In addition, this paper will assume institutions do not have multiple active Internet connections unless otherwise noted. This is the simplest case and although many of the solutions apply in all cases, some solutions become extremely difficult to deploy in sites with multiple entry and exit points. Furthermore, many of the solutions in this paper may apply throughout an institution's own autonomous internet as well, but this paper will not address those cases specifically.

*Thanks to the UNISOG mailing list members for their helpful feedback and a special thanks to Joe St Sauver for his valuable insight and comments.

[1] The term bandwidth is often misleading so from here on out the generic term *capacity* will be used instead.

As network capacity management alternatives are outlined, this paper will also address some of the possible consequences in various scenarios. Where necessary, institutions are cautioned on the short and long term implications a particular solution may impose. A brief overview of a few of the underlying technologies are also included to help an institution formulate its network capacity management decision. This paper will avoid any detailed coverage of standards based *class of service* (CoS) or *quality of service* (QoS) solutions, since much of the work in these areas is still largely experimental and subject to change.[2] This paper also focuses on network based solutions rather than host, application software or social changes.

2 Adding Capacity

Perhaps the most obvious solution to capacity collapse is to simply get more capacity. In fact, many institutions have done just that. The overwhelming problem with this solution is the increased cost associated with public Internet connectivity. There may also be provisioning problems in acquiring the necessary capacity. It may also be a difficult decision to determine how much capacity is enough and how long the increase in capacity can remain adequate. This solution however is arguably the easiest to implement of any described in this paper. Simplicity itself may be worth the cost of the capacity increase.

3 Access Blocking

Network administrators are half jokingly perceived to be control freaks. A popular reaction and long term strategy of many network admins is to strictly prohibit certain types of network usage. An institution's acceptable usage policy (AUP) may allow them the authority to use technical means to limit certain types of activity. In fact many institutions have installed comprehensive rules in their network routers to prevent communication using certain IP addresses, protocol types, application types and even specific content.

Access blocking using router filters may be a relatively straight forward process and it generally requires very little start up cost. Comprehensive router filters can work to block most unacceptable network traffic. However, as long as there is a minimal window out of the institution, it is quite possible and very likely that applications or users will be able to find ways to circumvent blocks through the use of protocol tunneling, port hopping, encryption, proxy servers and other tricks. Access blocking may lead institutions down a difficult path of trying to control user and application behavior through imperfect technological means. In addition, when blocking certain types of traffic, other unrelated traffic may be penalized. For example, if a router filter is configured to prevent the use of TCP port `6688` in order to block Napster traffic, other applications that happen to be assigned port `6688` by the operating system will also mysteriously fail.

In addition to the potential difficulty to block specific usage while allowing legitimate traffic, forcing network devices to inspect all packets and block unauthorized usage can be a computationally expensive task. In some cases, access blocking can actually degrade overall system and network performance.

Unauthorized network access may simply be the result of an application's default configuration rather than user intent. In these cases, blocking might provide a quick and easy solution to capacity collapse. However, as many educational institutions discovered early on, complete access blocking may give rise to an ugly political backlash from the user community.

4 Rate Limiting

Rather than completely sever access many institutions have configured limits on the amount of capacity certain protocols, applications and users are allowed. Limiting rather than blocking is generally perceived as a more reasonable approach to dealing with capacity issues resulting from the usage of large capacity applications. The technical solutions to rate limiting traffic usage are not unlike those of blocking

[2] The reader is referred to [4], [5] and [6] for further information in these areas.

configurations. With both management techniques the network administrator must know the specific characteristics of the traffic. This again might include IP addresses, protocol types, application types and possibly specific content. Additionally, rate limiting requires the network administrator to select appropriate limits for restricted traffic, which in itself may not be trivial.

Strict rate limiting is often the easiest solution, but many network administrators are now using more sophisticated mechanisms such as limiting traffic only during peak usage times or when a host is nearing a daily allowance of bytes transferred. In situations where dynamic rate limits are used, an institution must maintain accurate traffic accounting, robust automated configuration management and scalable systems.[7][8] These more complex mechanisms attempt to provide a greater level of *fairness* over time. For the moment many institutions have found that some forms of rate limiting are successfully preventing capacity collapse.

Rate limiting however has a number of subtle, but often dangerous consequences. First, it matters a great deal where the rate limits are set. For example, if rate limits are setup on the interface nearest the end host, should the limits be in the upstream (from the end host) or in the downstream (to the host) direction? If for example rate limits are in the downstream direction, or the egress from the network, this means some traffic is being carried a long way and then suddenly dropped. Dropped packets are sometimes necessary in periods of congestion, but the penalty incurred can be harsh. A transport or application layer protocol timer will need to expire before recovering from a lost packet. A TCP retransmission for example may not occur for at least a couple of seconds at minimum, a noticeably long time in today's networks. Furthermore, a lost packet will probably just be resent, which is not a very efficient use of the network especially if the retransmitted packet *almost made it* the last time.

Secondly, rate limits inhibit temporary traffic bursts. Historically Internet traffic is bursty on all time domain scales. If the unlimited capacity goes unused, the limited traffic cannot take advantage of the available capacity and this eliminates one of key advantages in using packet switching networks.

Thirdly, rate limits can be circumvented. Unless all traffic is rate limited users and applications may disguise unacceptable traffic as unlimited traffic possibly through the use of tunneling or port hopping tricks. If rate limits apply to all traffic, users may use alternative physical paths around the limits.

5 Active Queue Management

Internet routers have a finite amount of buffer memory with which to store packets traversing from one interface to another. In most networks, the potential capacity of the end stations far exceed the capacity of the uplinks, backbones and expensive Internet service. Historically most end stations in data packet switched networks operate at their potential load carrying capacity for very brief periods of time. The statistical multiplexing advantage prevents the total potential aggregate capacity from being realized in practice. When backbone connections and uplinks do become saturated, the network may be forced to drop packets as queues fill. Almost all routers currently deployed will drop all incoming packets that run into a full queue on an output interface by what is commonly referred to as *tail drop* or FIFO queueing.[9]

Recently there has been a great deal of interest in active queue management (AQM) to help complement TCP's congestion avoidance mechanisms.[10][11] With the addition of flow, priority or end station signalling, active queue management may provide one of the most promising methods for capacity management in the future.[3]

One of the most popular AQM algorithms is random early detection (RED).[13] RED enabled routers will mark incoming packets with a drop probability based on an average running queue size. As the queue size increases, packets are marked with higher probabilities and thus are more likely to be dropped by the algorithm. Using RED as an AQM is particularly helpful with TCP connections since dropped packets are implicit congestion signals for senders to slow

[3] Some institutions are experimenting with a combination of AQM and traffic rate limits.[12]

down. RED has been shown to be effective in achieving stable link utilization at the expense of increased packet drops. It has also been shown that large packet flows will be dropped more frequently than short flows, which prevents starvation and excessive delays for light applications and users. Coupled with explicit congestion notification (ECN), packet drops can theoretically be eliminated resulting in near perfect link utilization.[14] Of course, a perfect world is still a long way off and mechanisms such as ECN are still in the experimental stage.

AQM and signalling mechanisms such as ECN are still areas of active research, but are being tested by a few organizations and vendors. In fact the IETF currently recommends the use of RED on Internet routers.[9] It is also worth noting that certain protocols such as UDP may not respond to implicit network congestion signals or explicit limits set by an administrator. Prohibiting unresponsive flows from causing capacity collapse is a ongoing area of research.[1]

6 Scheduling

Instead of managing queue lengths, scheduling algorithms manage packet transmission order. So for example, a simple scheduling mechanism may decide to prioritize and transmit HTTP traffic flows ahead of FTP traffic flows. Scheduling algorithms will give varying amounts of link capacity to the configured classes of traffic. A class of traffic can be a protocol type, a TCP flow, an IP address or other such identifying characteristic of network packets.

Various flavors of scheduling algorithms are available from numerous router and capacity management vendors. Generally speaking, scheduling algorithms have not yet seen widespread use, because their long term impact on end-to-end traffic patterns is still uncertain. In addition, just as users and applications can force their applications to get around blocks and rate limits by altering traffic characteristics, so to can traffic be crafted to conform to the highest weighted class used in a scheduling algorithm. What many say is needed is proper *admission control* at the edges of the network.[15] Scheduling algorithms are still in the early stages of research and experimentation As of yet, they are not widely deployed on the Internet.[4]

7 Traffic Shaping

One popular third party capacity management product implements a technique known as *TCP rate control* that adjusts traffic patterns to fit into a profile defined by a network administrator.[18] This technique, also known generically as traffic shaping, attempts to modify TCP window size advertisements as packets flow through the network. In addition, TCP ACKs returned by the receiver can be *paced* to control the entry of new packets into the network by the sender.

Traffic shaping in this manner, when it is not performed by the end hosts, requires the use of a middle box architecture.[19]. Middle box architectures have a number of documented complications for the traditional transparent architecture of the Internet.[20] As with any mechanism within the network that alters packet transmission, it takes a great deal of insight and experimentation to understand the consequence traffic shaping may have on the overall operation of the Internet. It may also be difficult to implement traffic shaping policies if users or applications are able to disguise traffic using some of the tricks previously mentioned.

8 Caching

Throughout the world of computing, caching technology has always been on the forefront of performance enhancements. From CPU caches to local disk caches, fetching a local copy of data can decrease response time by orders of magnitude. In fact, many administrators in many different types of organizations have been using world wide web caching for a number of years. Most web browsers support local caching of pages by default. By placing a cache at the border of an institution's Internet connection, caches may help slow the growth of expensive Internet costs

[4] For current activity in this area see [16] and [17] for further information.

by returning requested data much more quickly than if packets had to traverse the entire end-to-end path. Even with reported cache hit rates as low as 30 to 40 percent, the savings are often significant enough for many network administrators to take an interest in network caching technology.

Like most solutions discussed so far, a number of caveats should be considered. Scaling issues, data staleness and middle box complexities may give an institution hesitation in fully implementing network caching systems. As long as traffic demands are increasing, an institution must also recognize that caching systems only delay the inevitable need for more capacity. Assuming an optimistic cache savings rate of 50% on current capacity, the rate at which an institution's traffic doubles is the length of time the institution can go without needing more capacity. This may only be a few months for many institutions. Use of a voluntary cache system on the network may be a nice middle ground approach. For increased performance, users would be given an incentive to use the cache system, but can avoid it if they want to.

9 Private Peering

Private peering agreements provide the advantage of adding capacity without many of the financial burdens for Internet service. Agreements by one or more organizations to peer will occur in common Internet access locations such as public exchange points.

Typically any two organizations agreeing to peer will configure traffic destined between each other to travel directly over the peering link rather than over a common Internet transit provider link. Most often, peering arrangements do not involve money being exchanged between participants. Therefore, the more traffic that travels directly between peers the better since commodity Internet transit costs are reduced by this amount. Since an institution's link must terminate in a place where peering configurations can be made, this may not be a viable option for some.

10 Other Technologies

A number of other solutions to managing capacity have been proposed and are beginning to see use by a growing population of institutions. Another group of solutions were originally designed to solve other problems, but in many cases have been, sometimes inadvertently, used to control capacity issues.

10.1 Monitoring

Though not a solution per se, monitoring network capacity must be the first step to avoiding network capacity collapse. Certainly network capacity problems can crop up quickly, but with sufficient monitoring, areas of concern can be quickly identified. Minimally an institution should be monitoring Internet link utilization with a tool such as the Multi Router Traffic Grapher (MRTG).[21] It would be even better if an institution monitors individual subnet utilization with a tool like MRTG and Internet flow statistics with a tool like *cflowd*.[22]

Other tools and features of networking equipment can help identify looming capacity issues as needed. It is important to be able to visualize short and long term trends and traffic patterns. It also helps to be able to understand the traffic characteristics, even if only as an aggregate to an application class or geographic location. Monitoring also helps to justify future capacity solutions.

10.2 Consortia

A possible capacity enhancing solution is when an institution can gain access to a local, state, national or global consortium offering some type of capacity enhancing service. For example, participation in Internet2 includes significant traffic carrying capacity to other Internet2 connected sites at significantly reduced prices, albeit to a restricted set of sites. This solution is often inappropriate solely as a capacity solution, but is an added advantage to consortium participation. Often politics, institutional goals and budgetary constraints will dictate participation in consortia.

10.3 Proxy Servers

Proxy servers have most often been used as a way to control access to internal and external resources often with the use of authentication and authorization controls. When used as a central *choke point* that users must filter through, an institution can obtain detailed accounting and auditing information about network usage. If coupled with other technologies a proxy server can do much more. Address translation, web caching and content filtering are some common services that are often coupled with proxy servers.

With regards to managing capacity, proxy servers can do a lot of the functions that have discussed up to this point and perhaps all of them in a single framework. Therein lies the primary problem with proxy server deployment. Requiring the use of a proxy server may presuppose many of the limitations of middle box architectures.

10.4 Content Distribution

Content distribution networks are related to caching systems, although they differ in one important respect. The content providers place copies of their data as close to the users as possible. This has all the advantages of caching without many of the middle box architecture issues an end user might encounter. On the downside, it is more costly and complex for the content provider to maintain replicas of their data across the Internet. Akamai is one example of a leading provider of content distribution solutions.[23]

10.5 Content Subscription

Similar to content distribution an institution can subscribe to a content subscription service in order to reduce the necessity of using expensive Internet links to obtain the same data. So for example, rather than users going off-site to download and watch movies, an institution can subscribe to a service such as iBEAM and have current content available locally.[25] This not only reduces strain on the Internet links, but may also help alleviate potential copyright infringement issues. The primary barrier to this service is the subscription cost an institution may incur.

10.6 Compression

Compressing data before transmission is another technique, which may be used by end hosts or middle boxes to conserve network capacity. However, the use of compression for Internet traffic is not a common practice. Compression is often not practical in packet switched data networks. Many data packets are only a few dozen bytes long. Compressing individual packets does not save much and often may make a packet bigger depending on the compression algorithm being used. Additionally, compression at high speeds can be computationally difficult without slowing things down further. Perhaps the most important reason network based compression isn't practical is because a significant amount of data is already compressed (e.g. MP3 files, JPG images, and other multimedia content).

Compressing files for transmission and decompressing them on arrival is very common, but typically not automatic.[5] Packeteer has recently developed a new product, which among other things claims to help speed Internet performance by compressing and customizing content to capacity constrained networks.[18] This product however must be deployed near content provider networks before traffic hits constrained capacity links.

10.7 Network Address Translation

The use of private IP addresses and network address translation was originally intended as a temporary solution to a public IP address shortage problem.[24] In recent years, many organizations have come to believe that using private addresses and doing network address translation (NAT) between their network and the public Internet is just standard operating procedure. Unfortunately NAT makes internetworking in the traditional Internet transparency model much more difficult.[20] The use of NAT has slowed some deployment of applications and among these include many of the latest high capacity ones such as Napster. The use of NAT makes it harder for applications to act as servers, but not impossible. Applications only

[5] As in the case where many software applications are packaged with `tar` and `gzip`.

need to change their behavior to operate in a NAT environment and in fact many are now doing just that.[6] A potential, but as of yet largely unpopular consequence, would be protocol stacks and applications implementing standard protocols in completely nonstandard ways. If an institution is not currently using NAT it is not recommended as a solution to capacity collapse due to the numerous other unrelated problems NAT can impose.

11 Conclusion

Institutions must often wrestle with conflicting demands on capacity. For example, many users and applications need low latency and strict jitter control, while others need sheer throughput. Being able to offer a high level service in both domains is important, but also very difficult. Unfortunately there is no single, all encompassing solution to avoid capacity collapse. A number of the alternatives are not even that attractive, particularly with considering long term consequences. Capacity management solutions can prevent short term capacity collapse, but great care must be taken not to paint oneself into a corner with a short sighted approach. Blocking, limiting, shaping and other similar solutions do not add more capacity, but rather, they try to slow things down. There will probably be a breaking point when the solution really does call for more capacity. It is not a matter of if, it is a matter of when. Over time an institution finds that adding capacity is a solution that cannot be avoided.

References

[1] Sally Floyd and Kevin Fall. Promoting the Use of End-to-End Congestion Control in the Internet. IEEE/ACM Transactions on Networking, August, 1999.

[2] J. Postel and J. Reynolds. File Transfer Protocol (FTP). RFC 959, October 1985.

[3] Napster home page: http://www.napster.com.

[4] IETF Differentiated Services Working Group.

[5] Paul Ferguson and Geoff Huston. Quality of Service: Delivering QoS on the Internet and Corporate Networks. Wiley Computer Books, January 1998.

[6] Internet2 QoS home page: http://www.internet2.edu/qos/.

[7] Charley Kline, University of Illinois at Urbana-Champaign. Usage Pattern Adaptive Rate Limiting. APAN/TransPAC/NLANR/Internet2 Workshop, January 2001.

[8] Information Systems and Technology (IST), University of Waterloo. Excessive Use from a Residence Connected Computer, University of Waterloo Computer Directives and Related Documents. February 2001.

[9] B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Patridge, L. Peterson, K. Ramakrishan, S. Shenker, J. Wroclawski, and L. Zhang. Recommendations on Queue Management and Congestion Avoidance in the Internet. RFC 2309, April 1998.

[10] J. Postel. Transmission Control Protocol (TCP). RFC 793, September 1981.

[11] M. Allman, V. Paxson, and W. Stevens. TCP Congestion Control. RFC 2581, April 1999.

[12] Rich Graves. Brandeis University. Relevant information at: http://offshore.unet.brandeis.edu/bandwidth/.

[13] Sally Floyd and Van Jacobson. Random Early Detection Gateways for Congestion Avoidance. IEEE/ACM Transactions on Networking, August 1993.

[6] For example, as long as one of the two hosts in a TCP session has a public IP address, the application can have the host behind a NAT box initiate the connection whether it is acting as a client or server.

[14] Sally Floyd. TCP and Explicit Congestion Notification. ACM Computer Communications Review, October 1994.

[15] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource ReSerVation Protocol (RSVP). RFC 2205, September 1997.

[16] Bernhard Suter, T. V. Lakshman, Dimitrios Stiliadis and Abhijit Choudhury. Efficient Active Queue Management for Internet Routers, November 1997.

[17] M. May, J. Bolot, C. Diot, and B. Lyles. Reasons Not to Deploy RED. Proceedings of IEEE/IFIP IWQoS June 1-3, 1999.

[18] Packeteer home page: http://www.packeteer.com.

[19] IETF Middlebox Communications Working Group.

[20] B. Carpenter. Internet Transparency. RFC 2775, February 2000.

[21] Multi Router Traffic Grapher home page: http://www.mrtg.org.

[22] CFLOWD home page: http://www.caida.org/tools/measurement/cflowd/.

[23] Akamai home page: http://www.akamai.com.

[24] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, and E. Lear. Address Allocation for Private Internets. RFC 1918, February, 1996.

[25] iBEAM home page: http://www.ibeam.com.