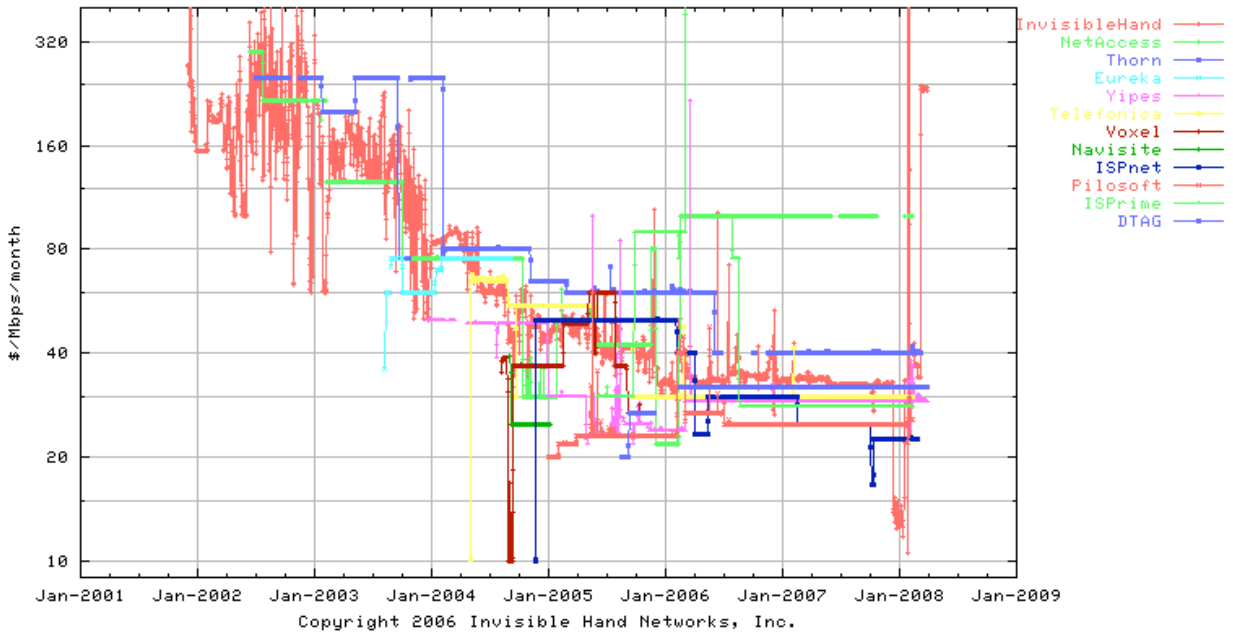
Henning Schulzrinne
Columbia University

ENCOURAGING BANDWIDTH EFFICIENCY FOR PEER-TO-PEER APPLICATIONS

Introduction

Bandwidth costs for high-volume data can be significant. If we were to assume that all roughly four hours of daily TV viewing were streamed at HDTV rates of 18 Mb/s, a household would consume about 972 GB per month. Currently, capped broadband[1] ranges from 10 GB to 150 GB per month. (Columbia University caps connections at 350 MB/hour, equivalent to 252 GB/month.)

The economics of bandwidth



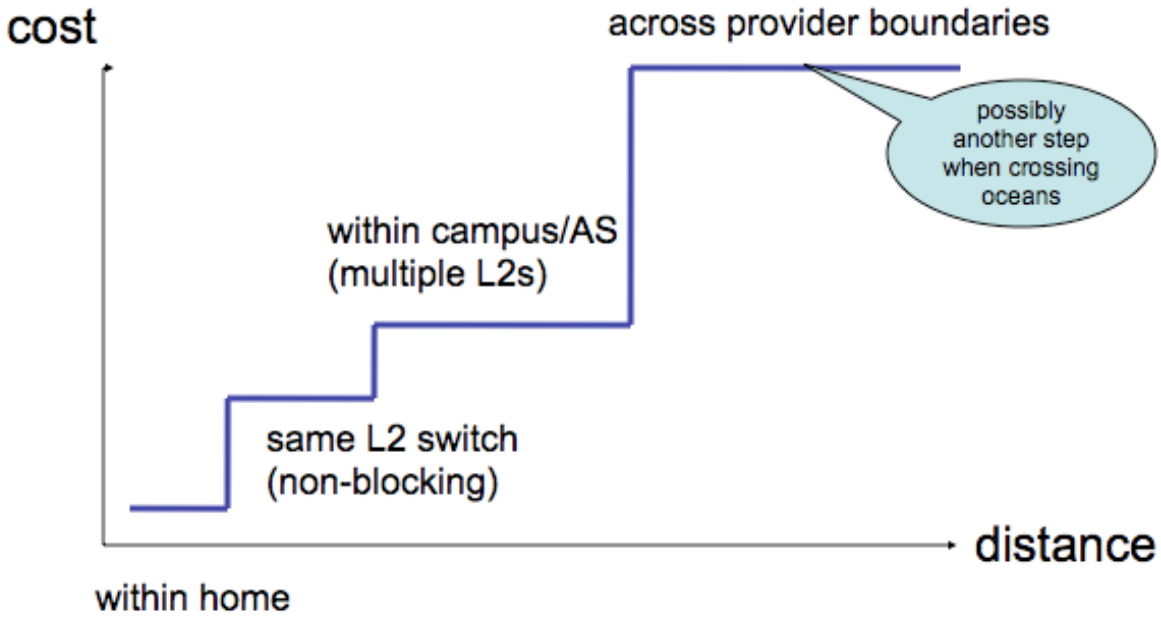
The cost of Internet bandwidth can be estimated in a variety of ways. The figure above, from Invisible Hand Networks, shows the market price of bandwidth, in dollars per Mb/s per month. One Mb/s/month corresponds to 324 GB, so that the price of $40, say, corresponds to a delivery cost of $0.12/GB. (It is interesting to note that the cost seems to have settled on around $40/Mb/s/month since 2005.) US colocation providers, from an informal sample, charge about $0.30/GB to $1.75/GB. CDN costs for every large customers in 2007Q4 are quoted[2] as about $0.08 to $0.19. Thus, these numbers are all in the same general neighborhood. Assuming $0.15/GB, shipping a DVD's worth of data costs about $1.05. Netflix is said to pay about $0.70 in postage for a round-

[1] shaw.ca; quoted at http://hnorth.wordpress.com/2007/09/14/

[2] http://blog.streamingmedia.com/the_business_of_online_vi/2007/11/cdn-pricing-dat.html

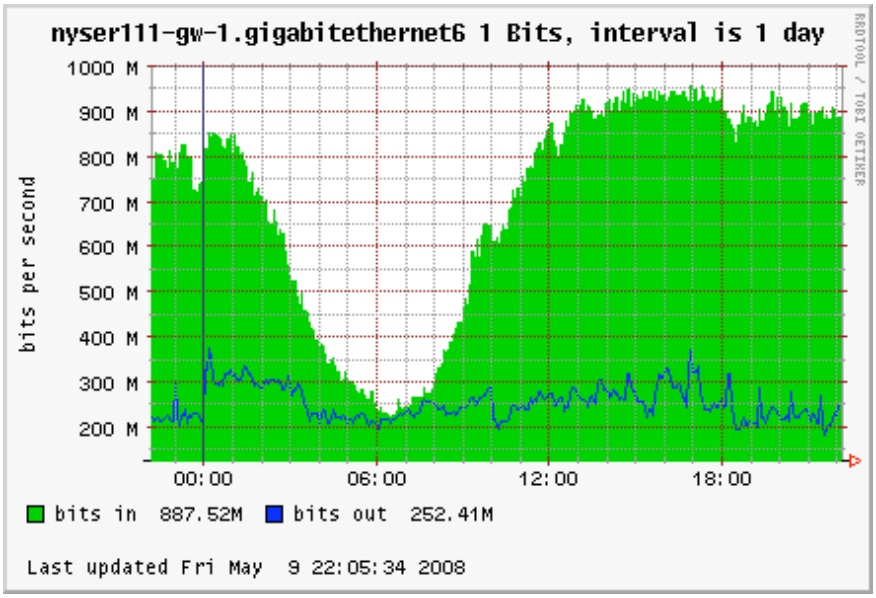
trip of a DVD[3]. Thus, in a variation on the old saying attributed to Tanenbaum that nothing beats the bandwidth of a station wagon full of magnetic tapes, the US postal service is still the cheapest volume data delivery mechanism, albeit with ping times measured in days.

The tiering of bandwidth costs



Because of the small number of cross-provider hand-off points and the distance-insensitive pricing of wide-area Internet bandwidth, one can probably approximate the cost-vs.-distance function of bandwidth by a step function, as illustrated in the figure above. The precise boundaries depend on the local network architecture and whether switching and transmission facilities are owned by the service provider (and thus amortized as capital) or paid by volume.

[3] http://www.hackingnetflix.com/netflix/2005/07/postmaster_gene.html



When a provider uses its own facilities, the cost is the amortization expense of the infrastructure, rather than traffic volume, and is thus largely driven by the daily peak demand. Thus, just like power companies, ISPs have an incentive to smooth diurnal variations. An example of such variations is shown in the figure above, which illustrates bandwidth usage for the commodity Internet connection at Columbia University.[4] The graph shows a ratio of more than 4-to-1 between the trough in the early morning and the daytime peak load, although the average load is probably closer to 70% of the peak in this particular case.

Network services for more efficient data delivery

There are three straightforward network services and protocols that can allow application writers to build better tools:

1. Ability for an application to determine its network location and proximity to other nodes;

2. Support for cost and charging information for application so that applications can make appropriate trade-offs;

3. Support for DiffServ code points that avoid bulk data interference with high-value, latency-sensitive applications.

We will describe each of these issues in more detail below. All three address network congestion, while the first two are likely to reduce wide-area network costs (or allocate them to users disproportionally using such services.) All are content- and source-neutral.

Determining Network Location and Distance

[4] http://www.columbia.edu/acis/networks/bandwidth.html

Currently, it is difficult for an end node to determine accurately how far it is away from other nodes, particularly in a network cost sense. At best, it can try to translate a public IP address into an AS number using services such as RADB, but these do not scale to Internet-wide use and are not applicable to end systems that only have an RFC 1918 (NAT) address. The end system's AS number provides a simple indication of whether a system is served by the same provider and is useful particularly for , but there is no relationship between AS numbers. For example, Columbia University has AS number 16 and is close to other networks that are on Internet 2, as well as to systems that are served by its immediate upstream commercial Internet provider, but it is essentially impossible to discover this relationship. Thus, a service that finds nearby AS, as defined by the AS provider, would all an application to preferentially select peer nodes from that set. (However, the mechanism described in the next section may obviate the need for this.)

Making the AS number available via DHCP would provide an immediate, low-complexity mechanism for end systems to discover this information. Given the difficulty of propagating new DHCP types across consumer-grade NAT devices, it may be worthwhile to also consider other mechanisms, such as an extension to STUN, but as an additional mechanism, not a replacement.

In addition, there should be a simple mechanism to discover the AS number of any IP address, as this is required, in conjunction with the mechanism described below, to discover the potential cost of communication to another network.

Given the transition to 32-bit AS numbers, there is no shortage of such numbers, so one could easily create pseudo-AS if it becomes necessary to distinguish between different parts of large networks, even if they are not advertised separately in the routing infrastructure.

Cost-of-bandwidth information

Relying on cooperative applications to be "nice" is probably insufficient unless there is an incentive. There are at least three possibilities, namely either volume limiting, rate limiting or volume-based charging. For example, a network could offer a tiered tariff that allows unlimited bandwidth within the same AS, 100 kb/s rate-limited but unlimited prioritized DiffServ to anywhere and volume-limited bulk traffic to any AS. There are several ways to convey this information to applications, allowing them to make appropriate trade-offs. For example, a network could provide an information service with a list of service descriptors, similar to the RSVP TSpec. Alternatively, a protocol such as GIST could discover the cost of particular destination.

The service classes may change over time. For example, a network may offer "midnight specials" when the network is idle during off-hours, encouraging bulk downloads during that time.

Scavenger Service and Real-Time Services

A number of years ago, the Internet2 QoS working group investigated a bulk data service class, referred to as scavenger service[5]. Packets marked with that DSCP value would get the lowest priority and thus would lose out in competition with other TCP flows, for example. The service

[5] http://qos.internet2.edu/wg/wg-documents/qbss-definition.txt

would limit interference of (peer-to-peer) bulk data with other data on both the user's access network and shared network elements.

As a complementary mechanism, providing (rate-limited and/or volume-limited) DiffServ EF (RFC 3246) services may help improve service for latency and loss-sensitive applications.

Security Considerations

Any high-priority or volume-charged service incurs a potential security risk to the user. For example, an attacker could launch a denial-of-service attack that exhausts the user's traffic quota or incurs bandwidth charges. A compromised host could run up a tab as it delivers spam or denial-of-service packets as a member of a bot net. Clearly, these risks have to be mitigated, but measures to prevent these may also encourage better end system hygiene and provide incentives to OS vendors to address security issues.

Risk mitigation strategies include requiring user confirmation before volume-based charges are incurred or user-configurable volume limits.

A more complicated solution involves a user-setable traffic filter, so that only solicited traffic can reach the host. (We have proposed using NSIS to establish such filters.) A simple version of that is a stateful TCP filter, similar to the NAT tables in use today, that only allows inbound TCP traffic if there is an active TCP connection. This obviously does not address unsolicited UDP packets. It may be possible to leverage ICE for creating such state.