Peer to Peer Edge Caches Should be Free

Nicholas Weaver

Abstract

Bulk-data peer to peer systems have been promoted as reducing the cost of content distribution. Unfortunately, this isn't generally true. Rather, bulk-data P2P shifts the costs of delivery from the content provider to the ISPs or content recipients. A very simple economic model of content delivery shows how this occurs and that, under some conditions, P2P can grossly magnify costs. Even with localization this cost increase may still occur.

However, if the P2P system includes the notion of in-network caches, the economics change greatly. A P2P caching infrastructure allows both content providers and ISPs to realize significant cost savings, and the benefit is maximized when the caches are free to use.

1 Introduction

Bulk-data peer to peer systems have been promoted as reducing the cost of content distribution. Unfortunately, this isn't generally true. Rather, bulk-data P2P is designed to shift the costs of delivery from the content provider to the ISPs or content recipients.

This cost shifting can be significant. A simple linear model of content delivery shows how this occurs and that, under some conditions, P2P can grossly magnify costs. Even with localization this cost increase may still occur, in particular for cable and wireless networks where the last mile uplink is a particularly expensive resource.

Caching can change the economics. If the P2P system includes in-network caches [17], now not only does the content provider see cost savings, but the ISP does as well. There are many requirements in building such caches, including support for attribution, cache discovery and customer preference, and designs which emphasize unreliability and partial deployment.

Most critically, such caches need to be free to use: the maximum benefit to all parties occurs when the cache represents just another network resource, not a value-added service beyond normal connectivity.

2 Terminology and Background

Bulk Data P2P systems: Bulk-data P2P is a peer to peer system centered around the delivery of large data files, either as a single entity (such as BitTorrent [3]) or organized in a streaming fashion (such as OctoShape [12]). The salient features of bulk-data peer to peer is simply that: it is used to transfer large files between a significant group of participants. This contrasts with conventional content delivery where the content provider is responsible for providing the source file to all the recipients.

Peer to Peer localization: Localization is an enhancement for a peer to peer system to preferentially chose "local" nodes. Localization's primary goal is to both increase performance and reduce overhead. Work on this area includes P4P [16] and the ALTO working group in the IETF [8].

HTTP caches: An HTTP cache is an in-network device which provides a common cache for content among multiple users. Thus if a piece of data is in the network's cache because a previous user fetched the item, the cache will return the item without having to fetch it from the network.

P2P Edge caches: A P2P cache is a device for caching P2P traffic rather than HTTP traffic. There have been existing P2P cache designs, but these caches are protocol specific and require some form of deep packet inspection to discover client requests. There is a proposal from Yang et al [17] which proposes building caching into a peer to peer infrastructure as part of the IETF DECADE group.

3 A simple linear cost model for content delivery

Consider the costs involved in distributing a file of B bytes from a content provider to N total customers on I ISPs, both for the content provider and an ISP which is serving M of the N customers.

The content delivery cost is C_{con} per byte for the content provider, while the ISP's cost is broken into four areas, C_{bd} for each byte downloaded from the Internet into the ISP's internal network, C_{bu} for each byte transfered to the Internet from the ISP's internal network, C_{ld} for each byte downloaded from the ISP's internal network to the customer over the "last mile", and C_{lu} for each byte uploaded from the customer to the ISP's internal network.

Of course, these prices are a simplification. For some purposes, there is a strict linear or almost linear relationship (eg, if the content provider is using a delivery mechanism like Amazon S3[13]), while for others there can be significant nonlinearities: if there is no congestion, the cost of bandwidth is 0, but if there is congestion, the cost of bandwidth is extremely high because other customers are limited in service and increasing the available bandwidth would require a slow and expensive network upgrade.

3.1 HTTP and cached HTTP

Under this model, the cost for conventional HTTP delivery is straightforward: the content provider incurs BNC_{con}, while the ISP incurs $BM(C_{bd} + C_{ld})$. This should be taken as the baseline cost for content delivery.

One natural optimization is HTTP caches. If the ISP has an HTTP cache, the total cost will be somewhere between $BM(C_{bd} + C_{ld})$ and $BC_{bd} + BMC_{ld}$, depending on the effectiveness of the cache, with the content provider also seeing comparable savings. Likewise, the content provider also sees savings equal to the successfulness of the cache.

Unfortunately, HTTP caches are not without problems. If coded incorrectly, they may cache data incorrectly, leak information, or even introduce vulnerabilities [4]. Some websites may use encoding and distribution information that doesn't favor caching. Finally, HTTP caches are failure-intolerant: if the cache has a problem, this will cause a denial-of-service condition in the network. Thus although some in-network HTTP caches exist, they are relatively rare.

Finally, even with perfect caching, *a*ny content delivery mechanism, be it HTTP, P2P, or some not yet invented protocol, will always incur at least $BC_{bd} + BMC_{ld}$ cost for the ISP, as this represents the cost of delivering a single copy of the data from the Internet to all the ISP's customers.

3.2 Bulk Data P2P

In the absence of HTTP caches, content providers naturally feel the pressure to minimize delivery costs. Thus there is a natural inclination to turn towards peer-to-peer delivery, such as Norway Public Television's use of BitTorrent[2]; or CNN's use of Octoshape's P2P for its streaming videos [5].

The natural savings from P2P delivery are substantial, simply because every byte delivered by a peer to another peer is a byte the content provider did not have to deliver.

Thus consider a P2P network where a fraction K is delivered from other peers rather than a content provider $(0 < K < 1)$. Thus the content provider's cost is now $(1-K)BNC_{con}$. And the more efficient the P2P system is, the more the savings. In the limit, the P2P content provider only needs to distribute a single copy, which would cost BC_{con} to deliver, a huge potential savings.

Recent usage by CNN has some rumors suggesting that K can be roughly .3, which alone is substantial. If a high volume site like Hulu or YouTube [1] could see similar savings, the incentive would be substantial. The only problem such providers face is that a P2P delivery mechanism requires violating the browser's "same origin" policy present in most scripting languages such as Flash or Java. However, at least one content provider site, ABC, uses a signed Java applet for the player which is able to bypass the same origin policy, and the Octoshape framework is an authorized Flash extension.

Unfortunatly, the savings to the content provider are offset by additional costs to the ISP. The ISP must see $KBMC_{lu}$ for the uplink, as well as additional cost on the border with the Internet. This can be described by the localization factor L, the fraction of P2P traffic which does not leave the ISP.

Thus with P2P, the content providers now sees a cost of $(1 - K)BNC_{con}$, while the ISP sees a cost of $(1 - L)BM(C_{bd}+KC_{bu})+BMC_{ld}+KBMC_{lu}$. Even with *p*erfect localization and perfect P2P delivery ($K = 1$ and $L = 1$), the ISP will see $BMC_{ld} + BMC_{lu}$ cost for delivering traffic. If localization is ineffective, then the ISP will see a total cost of $BM(C_{bd} + KC_{bu} + BMC_{ld}) + KBMC_{lu}$.

In particular, there are some ISPs, including cable providers and wireless ISPs, where C_{lu} is particularly high. If $C_{lu} > C_{bd}$, Peer to Peer content delivery must increase ISP costs, no matter the localization, and the more effective the P2P system (the higher K) is at shifting delivery from the content provider to user-connections, the greater the cost increase the ISP will incur.

Additionally, for most cases, $C_{lu} > C_{con}$, simply because it is far cheaper to deliver a hundred units of bandwidth to one location then it is to deliver one unit of bandwidth to a hundred locations. Thus the effect of bulk-data P2P is generally not cost savings but cost shifting, transferring the burden of content delivery from the provider to the recipients or recipients' ISPs.

4 Edge Caches

Caches offer a solution to this cost dilemma. In most ISP networks there exist locations where C'_{lu} is effectively zero within the internal infrastructure. Thus if a peer to

[1] Some estimates [14] suggest that YouTube's bandwidth costs Google roughly $300 million a year.

peer client, acting as a cache, is located in the infrastructure, P2P distribution will result in substantial cost savings for the ISP, as any data which is sent from the cache only costs BMC_{ld}, the theoretical minimum, just like an HTTP cache.

However, peer to peer edge caches have some properties not present in HTTP caches, including failure tolerance (which can enable low-cost designs) and partial deployability, both arising from the nature of P2P systems. They need to support attribution, edge-cache discovery, opaque data, and customer preference. However, they do not need to support anonyminity or reliable storage, and doing so may not be of benefit to the ISP. For example, the DECADE IETF informal working group is focusing on this approach [17].

4.1 Failure Tolerance

A P2P system, in order to be robust, *m*ust assume that nodes are added and removed all the time, a process known as churning. As long as the edge caches are treated like other nodes, the Peer to Peer system will tolerate edge caches which fail. And since P2P edge caches are failure tolerant, they can be inexpensive. A system which can *r*eliably store and deliver several terabytes of content must be expensive, requiring redundant disks, power supplies, and other features.

But if the nodes can be unreliable, then low cost components can be used: no redundancy, basic disks, and COTS systems. A 1U server (based on a Mini-ITX motherboard) capable of holding 4 SATA disks costs less than $800. With a price of $130 for a 1.5TB drive, an edge cache costing less than $1400 could cache over 5 TB of data. Such a low-cost system will suffer significantly higher transient and permanent failure rates than a higher-quality server, necessitating a reboot, reimage, and automatic disabling of bad disks, but as failures are low-consequence, such caches can be cheap to deploy.

And if an ISP's cost of delivery is $.10/GB, the cache only needs to offload 14 TB of total transfer to cover the cost of deployment over the system's lifetime.

4.2 Partial Deployment

Another advantage is partial deployment. If an edge cache is treated like other P2P nodes, then the absence of an edge-cache only degrades performance or increases ISP cost, but does not result in delivery failure.

This is significantly different from other content delivery mechanisms, such as Akamai or other distributed CDNs [1], which require that the nodes are deployed in most ISPs in order to see a general benefit.

4.3 Cache discovery and customer preference

One natural mechanism that needs to be built into the system is cache discovery and customer preference. As part of the localization process, the client software needs to discover the addresses of the ISPs caches, to allow it to communicate with the caches.

The caches should also act on a user's behalf. If a user desires a file and the cache does not have it, the cache itself should participate in the P2P network so that it receives a copy of the file, forwarding each block as it is received onto the final client. This enables the cache to provide the uploading required for peer to peer operation, so even if the data does not get reused by another ISP customer, the cache is still able to prevent the load on the last-mile uplink. And if another customer requests the same data, the cache now has the necessary data to provide it.

Likewise, the caches should only support the ISP's customers, not external freeriders. A cache may exchange information in a tit-for-tat manner with a non-customer, but only when a customer has requested this data and the cache is acting on the customer's behalf. Otherwise, the cache would be engaging in traffic which does not benefit the ISP.

4.4 Volatile Storage

An important feature of edge caches is they should not be regarded as persistent storage. Rather, it should be the responsibility of the content provider to ensure that the source files remain available and seeded. This greatly relaxes the design constraints on the caching infrastructure, as there is no longer a need to ensure that the distributed cache architecture maintains persistant copies of data.

Additionally, this means that the caches can be independent: each cache serving a group of clients and acting on their behalf, but not needing to develop a global view to ensure reliable data storage.

4.5 Anonymity and Attribution

Bulk data P2P, unless it uses a deliberate and inefficient layer of indirection, can not provide anonymous access, as participating peers will always see other peers and those wishing to map a P2P network can use sybils [15] to further map the participants involved in transferring a file.

Thus although the content may be opaque, the cache can not help but understand and should record the participants of the file. Although potentially recording more information, this information is already avaliable to anyone who participates in the P2P swarm. Thus the cache should be designed to provide attribution, not anonymity, for those reading the file.

Attribution, at least for local clients, is required for deployability. Without attribution, a caching infrastructure would be used for significantly illegal content. For example, rougly 40the bytes of Tor traffic is BitTorrent, dispite the poor performance, simply because Tor provides anonymity for participants [10]. Yet with attribution, the caching infrastructure is not useful for illegal content, as it allows copyright enforcement and law enforcement.

5 Edge Caches Should Be Free

Thus the observation: the maximum benefit to everyone occurs when the edge caches are free for all to use, a service provided by the ISP. The shift to P2P bulk content delivery may be inevitable, and content providers will benefit from the shift regardless of the presence of caching in the infrastructure.

Thus although P2P edge caches would seem to represent an increased cost for the ISP as an additional service, there is a distinct advantage to them being free. If access to the caching infrastructure is limited to paying content providers, these providers will not have an incentive to use them since there exists a free alternative: uncached P2P, where the additional costs are not born by the content provider but the ISPs.

5.1 A P2P Library

Yet in order to see this benefit, the P2P programs themselves need to be edge-cache aware. Although edge-caches can be built for many P2P applications using deep packet inspection [2], such caches would be application specific. Thus the best solution is not to design the caches around the application, but to develop a library which enables applications around the P2P caching infrastructure.

One possibly solution is to develop a generic P2P library, under the GNU Limited Public Licence (LGPL [7]) version 2 which could be used by multiple P2P applications in conjunction with a cache architecture. In particular, a library which integrates into either Adobe Flash [6], Java [9] or Silverlight [11] for ease of browser integration.

Using the LGPL version 2 also has a significant advantage over other licenses. This license is very permissive for integration, allowing arbitrary applications and easy adoption to use the library without restriction. Yet any changes to the library itself must be made public.

[2] The DPI required is rather limited: it is usually sufficient to identify the file being distributed and an entry point into the P2P swarm, which would allow the cache to participate in the network.

Thus the LGPL would enable caches and other systems to take advantage of development and changes to the library, while attempting to keep a common, unified codebase which would ease cache implementation, but without the development latency commonly seen in standards development.

6 Alternative Strategies

There are two alternative strategies ISPs could employ should peer to peer become a primary mechanism for distributing legal content. The first is to simply tolerate it. In such case, the ISPs would be less efficient at processing customer data, but it would not impede the development of P2P content distribution.

The second is for the ISP to restrict the use of P2P software on the network. This case can be made when bulk-data P2P is primarily used for pirated or illegal content [3], but would prove difficult if a major content distributor such as Hulu, Netflix, or YouTube shifted to a P2P model.

7 Conclusions

A shift to peer to peer content delivery may be inevitable, as the savings for content providers are substantial. Yet unless caches are integrated into the peer to peer architecture, this shift may substantially increase ISP costs. Yet if caches are part of the peer to peer design, both the content providers and the ISPs should experience significant cost savings, and the maximum benefit for all should occur when the caches represent a new, free resource for the ISPs customers.

8 Acknowledgements

Significant feedback was provided from Richard Yang, Richard Woundy, Jason Livinggood, Richard Alimi, Alex Song, Vern Paxson, Christian Kreibich, and Ning Zong. This work has been sponsored in part by the National Science Foundation. All opinions are those of the author and not those of the funding institution.

References

[1] Akamai, http://www.akamai.com/.

[2] Nate Anderson. Norway's public broadcaster launches bittorrent tracker, http://arstechnica.com/tech-policy/news/2009/03/norways-public-broadcaster-nrk-receives.ars.

[3] BitTorrent. `http://www.bittorrent.com`.

[3] Although recent experience in the US suggests that even this is politically difficult for most ISPs.

[4] Us cert advisory 435052, intercepting proxy servers may incorrectly rely on http headers to make connections, http://www.kb.cert.org/vuls/id/435052.

[5] David Charter. Cnn p2p video streaming raises questions, http://arstechnica.com/web/news/2009/02/cnn-p2p-video-streaming-tech-raises-questions.ars.

[6] Adobe flash, http://get.adobe.com/flashplayer/.

[7] The gnu lesser general public liscence version 2.1, http://www.gnu.org/licenses/lgpl-2.1.txt.

[8] The ietf application layer traffic optimization (alto) research group.

[9] Java software, www.java.com.

[10] Damon McCoy, Kevin Bauer, Dirk Grunwald, Tadayoshi Kohno, and Douglas Sicker. Shining light in dark places: Understanding the tor network. In *Eighth International Symposium on Privacy Enhancing Technologies (PETS)*, 2008.

[11] Microsoft silverlight, http://www.microsoft.com/silverlight/.

[12] Octoshape large schale live streaming solutions, http://www.octoshape.com/.

[13] Amazon.com Simple Storage Service (S3), http://aws.amazon.com/s3.

[14] Todd Spangler. Youtube may lose 470 million in 2009: Analysts http://www.multichannel.com/article/191223-youtube_may_lose_470 _million_in_2009_analysts.php.

[15] The sybil attack, http://en.wikipedia.org/wiki/sybil_attack.

[16] Haiyong Xie, Yang Richard Yang, Arvind Krishnamurthy, Yanbin Liu, and Avi Silberschatz. P4p: Portal for p2p applications. In *ACM SIGCOMM*, 2008.

[17] Yang Richard Yang. Sailor: Effective p2p application design using in-network data lockers, ietf 74, March 2009.