Peer-to-peer infrastructure: Case for cooperative P2P caching

Alan Arolovitch (PeerApp)
alan at peerapp dot com

1. Problem statement

The Internet traffic has been growing steadily in recent years, with year-over-year growth rate commonly put between 75% and 100%.
According to multiple studies [1-2], the increased consumption of bandwidth by multimedia traffic in all forms is the main driver of this growth.
The continuing adoption of the Internet as a mainstream platform of content distribution by content providers and ongoing growth in quality of online media are likely to drive similar growth rates in the future as well.

The increase in the Internet traffic has led to congestion of various segments of service provider networks, including last-mile, aggregation network and transit, degrading service levels for all Internet applications as a result.
Furthermore, the bandwidth crunch has led to an increase of service providers' costs related to traffic, with no prospect for increase in revenues.
This situation could lead to disruption of the business model that lies at the heart of the Internet-as-we-know-it – the Internet access funded purely by subscribers.

The Internet architecture today optimizes media delivery to work around limitations and variability of Internet throughput. Thus, traditional CDNs employ combination of network-aware sourcing and content caching to minimize cost and deliver data from a location closest to the end user.

Most recently, as complexity of the client applications increased, the logic of network optimization started to shift from server side to application.
The new breed of "network-aware" applications, that peer-to-peer applications are a subset of, emerged. These applications typically have high bandwidth requirements and adapt content sourcing depending on bandwidth and content availability. The techniques used include multi-source delivery (or "swarming") [3-4], adaptive media playback rates [5] and selection of source out of many driven by actual network service level.
Such "network-aware" application architecture offers high degree of scale and resilience to congestion.

At the same time, the primary goal of these optimizations is application performance, i.e. sustained high bandwidth throughput, not network cost.

The unilateral application techniques that aim to take network proximity of content sources as an input to source selection, based on measured latency or proximity estimates, are suboptimal and do not necessarily align with the network provider traffic engineering policy and actual traffic costs.
On the other hand, the network providers have engaged in various techniques [6-7] for management of the congestion through traffic prioritization.
The common practice is to apply selective packet loss and delay as a way to control utilization of the congested network segment, at the expense of application performance.
These congestion management practices have led to emergence of application traffic obfuscation [8, 9], potentially leading to counter-productive arms race between application and network providers.

Thus, the need exists for cooperative infrastructure-based solutions between application and network providers that would optimize media delivery for both application performance and network cost.

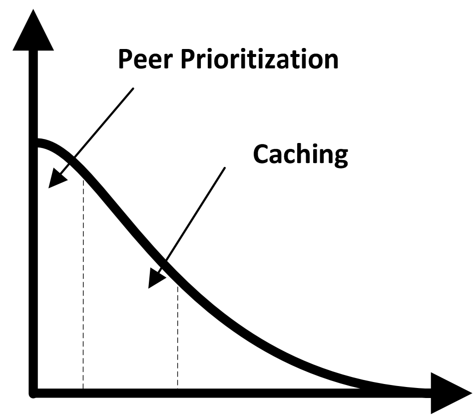
2. Traffic localization

The common network optimization practice that can achieve both application performance gains and reduce network costs is traffic localization.
Both caching and peer prioritization [10] are two approaches for traffic localization.
The above class of network-aware applications stands to gain the most from both.

The peer proximity optimization is likely to have a positive impact on both application performance and traffic locality mainly when applied to “head” of content popularity distribution curve.

The traditional peer-to-peer applications assume voluntary participation of the P2P nodes and create incentives [11, 12] to maximize such participation. As a result, the peer-based content sources in P2P networks are mostly ephemeral and short-lived.
The premise of the peer proximity optimization is that low-cost content sources are available in the network and the purpose of the cooperative framework is therefore to help the application to locate these sources.

As a result, the impact of proximity-based peer prioritization is going to be the largest for popular content items that attract a large number of concurrent downloaders of the same content item, for example in P2PTV applications or in flashcrowd situations in download P2P applications.



Peer-to-peer caches, on the other hand, are persistent, always available and have large storage capacity. The cache therefore is capable to provide locally a content item, which was downloaded into the particular network previously, but is no longer available from peer sources.

As a result, caching in peer-to-peer environment can provide the localization and acceleration benefits for larger part of the content distribution curve.

3. Caching benefits

Caching is traditionally applied to architectural issues involving limited throughput, from memory to storage systems to Web.
Media caching as a local acceleration infrastructure is able to provide significant acceleration benefits, especially when combined with application use of swarming. Given the statistical nature of content and peer availability, a persistent "always-on" cache component helps guarantee service levels and complements bandwidth capacity when peers are in short supply or peer discovery is in process.

Most of commercial peer-to-peer applications today combine peer offload with server-side acceleration components (the so-called "hybrid" architecture), based out of traditional CDN or own hosting facility.
Cache elements within service provider network help achieve both application performance requirements and reduce network provider cost at the same time.

At the same time, cache infrastructure is not a distributed resource and cannot scale as well as peer swarms. To preserve the scalability advantages of peer-to-peer architectures, peer-to-peer caching needs to combine approaches boosting peer participation, such as proximity peer prioritization discussed above or tiered peer indexing (tracking) [13].
There is the fore a value in combining peer optimization techniques with caching in a common tiered architecture.

References

[1] A. Parker, "**The true picture of peer-to-peer filesharing**", http://www.cachelogic.com/,

[2] Cisco whitepaper, **"The Exabyte Era"**

[3] **Swarmcast Autobahn**, http://www.swarmcast.com/

[4] **Bittorrent DNA,** http://www.bittorrent.com/dna/technology/

[5] **"Move Adaptive Stream",** http://www.movenetworks.com/

[6] S.Blake et al.**, "An architecture for Differentiated Services",** RFC 2475

[7] **"Deep Packet Inspection"**, Wikipedia, http://en.wikipedia.org/wiki/Deep_packet_inspection

[8] **"Message Stream Encryption",** http://www.azureuswiki.com/index.php/Message_Stream_Encryption

[9] **"Tracker Peer Obfuscation",** Bittorrent Enhancement Proposal #8, http://bittorrent.org/beps/bep_0008.html

[10] H. Xie, A. Krishnamurthy, A. Silberschatz, Y. R. and Yang. **"P4P: Explicit Communications for Cooperative Control Between P2P and Network Providers"**, http://www.dcia.info/documents/P4P Overview.pdf

[11] J.A.Pouwelse et al.**, "TRIBLER: a social-based peer-to-peer system",** http://www.tribler.org

[12] Bram Cohen, **"Incentives Build Robustness in Bittorrent"**, http://www.bittorrent.org

[13] Bram Cohen, **"bt2 protocol features",** http://lists.ibiblio.org/pipermail/bittorrent/2004-June/001178.html