

Cacheability of bulk content for ISPs

Bernhard Ager
TU Berlin / DT Labs

Fabian Schneider
TU Berlin / DT Labs

Anja Feldmann
TU Berlin / DT Labs

{bernhard,fabian,anja}@net.t-labs.tu-berlin.de

1. INTRODUCTION

The application considered responsible for most of the Internet traffic volume is file-sharing. Among the popular file-sharing applications are P2P-based ones, using e. g., the BitTorrent protocol, Web based offers by direct download services (DDS), e. g., `rapidshare.com`, and NNTP¹ based services, e. g., `giganews.com`. To cope with the ever growing bandwidth demands ISPs are exploring various options ranging from exploiting locality for peer/server selection, e. g., via CDNs or P2P neighbor selection strategies [11, 2, 4], to taking advantage of caching [1].

In this paper, we reexamine the feasibility of caching file-sharing content. In the early days of the Internet Web caches were very popular. But, the efficiency of Web caches decreased dramatically [7] with the introduction of advanced HTTP features, such as cookies, dynamic Web pages, AJAX-based applications, etc. However, recent studies of caching efficiency within Gnutella [9], Fasttrack [10], or YouTube [12] indicate that it is time to revisit the potential of caching.

We base our analysis on anonymized traces from a site that connects more than 20,000 DSL customers of a large commercial ISP in Europe to the Internet. We focus on the most prominent file-sharing applications in our environment: (i) BitTorrent (BT), the most used P2P system, (ii) DDS1, the most popular DDS provider, and (iii) NNTP. We find that caching looks like a promising option for BitTorrent but not for DDS and NNTP.

2. DATA

We base our analysis on anonymized packet level traces. Our results are based on a 24h trace starting at 4:00 am (local time) on a weekday in Sept. 2008. The traces are pre-processed with the Bro IDS [8] to extract anonymized HTTP, BitTorrent, and NNTP header traces immediately on a secured measurement infrastructure.

For HTTP we use the standard protocol analyzer while we developed DPD [6] based analyzers for NNTP and BitTorrent [5], including the Azureus Messaging Protocol [3] and the LibTorrent Extension Protocol. A brief look at NNTP reveals that 99% of the NNTP traffic is binary and from/to fee-based NNTP servers.

We observe that each of the three studied applications contributes between 5% and 10% to the overall volume. A summary is shown in Table 1. Similar protocol distributions have been observed at different times and at other locations of the same ISP.

Table 1: Data set summary

Application	Fraction of traffic	Fraction of users
BitTorrent	9%	2%
DDS1	6%	6%
NNTP	5%	1%

3. RESULTS

In order to estimate the potential of caching, i. e., how much of the content transferred via NNTP, DDS1, and BitTorrent is cacheable, we use a simplified approach. We only consider the first download of any content as unavoidable. All follow-up downloads are considered cacheable, as they might eventually be served from an ideal cache. This implies that if n copies of content X are downloaded the cache efficiency for X is $\frac{n-1}{n}$. As such we need to estimate for each file-sharing system how often each content is accessed. The overall cache performance is an average of each content's cache efficiency weighted by the content's size.

NNTP and Direct Download Services

For NNTP the unit of interest with regards to cacheability are articles. Unfortunately, a vast majority of articles is requested exactly once. We find that only a tiny fraction of all articles, 2%, are downloaded multiple times. With respect to bytes it looks even worse—only 0.03% of the bytes can be saved by caching.

For DDS1 the unit of interest are files which are accessed via HTTP. To identify which URLs refer to the same files of bulk content we consider only URLs starting with `/files/` and strip all parameters. Moreover, we ignore cookies and other HTTP header fields limiting cacheability. We find that 9% of the requests are in principle cacheable. However, most of these are small. As a result only 0.7% of the transferred bytes are requested more than once.

This appears to be very disappointing especially as the results from YouTube indicated that cacheability should be higher. However, even though both services contribute significantly to the overall traffic only a small fraction of the population is using NNTP, less than 1%, or the DDS1 service, roughly 6%. Moreover, the observation period is limited. For that reason we plan to collect longer traces.

¹Network News Transfer Protocol, Usenet, RFC 3977

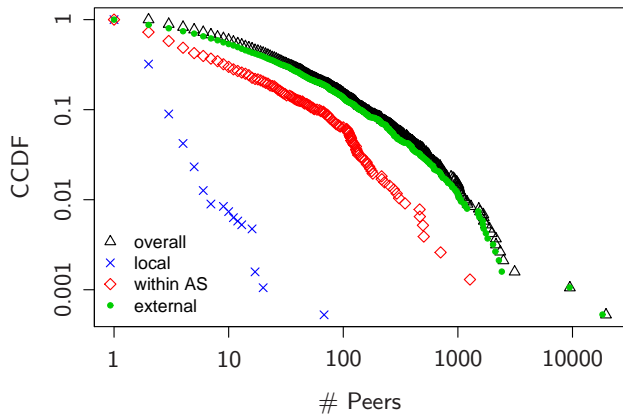


Figure 1: CCDF of peers per torrent by location

BitTorrent

For BitTorrent the unit of interest can be either the Torrent which represents the overall content or a Torrent block which corresponds to a 16kB transfer unit. Let's first focus on Torrents. Figure 1 shows the CCDF of the number of peers per torrent swarm. The reported numbers include peers that are not among our monitored hosts which is enabled by the P2P nature of the BitTorrent protocol. We distinguish between (i) hosts that are downstream from the monitor, the *local* hosts, (ii) hosts that are within the ISP's autonomous system (AS) that are not local, the *AS* hosts, and (iii) *external* hosts.

We see that most torrents have a sizable number of peers within the AS which translates into a high potential for caching. If all peers within the AS would do a complete download 96% of the bytes are downloadable from peers within the AS—corresponding to an AS-wide caching efficiency of 96%. However, note that this only includes peers discovered at a single monitoring point. When we consider only the local hosts the caching efficiency is still 41%. This indicates a substantial potential for caching.

Contrary to those results, when focusing on block downloads, the smallest entity in BitTorrent, we find that the local cache efficiency is only 5% and the AS wide one is 10%. Among the reasons for this substantial reduction is that some BitTorrent downloads started before the start of the monitoring section. As such the peers already have a substantial fraction of the content downloaded.

This can be exploited by both P2P neighbor selection strategies as well as caches. The former can redirect the downloads to local servers while the data may be entered into the cache upon initial download. We are able to identify such content by inspecting the bitfield messages of the peers. Now, the local cache efficiency again increases to 35% and the one that also includes peers within the AS to 77%.

The considerably higher caching efficiencies for BitTorrent result from three properties: (i) inspecting the bitfield messages allows us to peek into the past, i.e., we are able to observe effects of a longer time interval (ii) being a P2P protocol we can also observe users outside our local network, esp. within the AS and (iii) compared to NNTP BitTorrent has a wider user base and compared to DDS1 we speculate that the temporal density of data distributed via BitTorrent is higher.

4. SUMMARY AND OUTLOOK

Contrary to recent work we find that caching is not necessarily beneficial. For NNTP and DDS1 we find hardly any potential. On the other hand caching for P2P protocols seems promising especially when combined with P2P neighbor selection strategies. In future work we plan to extend our analysis in two dimensions: (i) Increasing the monitoring durations and (ii) considering additional applications like eDonkey and extending the HTTP caching analysis to all HTTP traffic, instead of DSS only. Further information can be found at:

http://www.net.t-labs.tu-berlin.de/posters/sigcomm09_content_duplication/

5. REFERENCES

- [1] ABRAMS, M., STANDRIDGE, C. R., ABDULLA, G., WILLIAMS, S., AND FOX, E. A. Caching proxies: Limitations and potentials. Tech. rep., Virginia Polytechnic Institute & State University, Blacksburg, VA, USA, 1995.
- [2] AGGARWAL, V., FELDMANN, A., AND SCHEIDELER, C. Can ISPs and P2P users cooperate for improved performance? *SIGCOMM Comput. Commun. Rev.* 37, 3 (2007), 29–40.
- [3] Azureus messaging protocol. http://www.azureuswiki.com/index.php/Azureus_messaging_protocol, April 2009.
- [4] CHOFFNES, D. R., AND BUSTAMANTE, F. E. Taming the torrent: a practical approach to reducing cross-isp traffic in peer-to-peer systems. In *Proc. ACM SIGCOMM* (2008), pp. 363–374.
- [5] COHEN, B. The BitTorrent Protocol Specification. http://bittorrent.org/beps/bep_0003.html, 2008.
- [6] DREGER, H., FELDMANN, A., MAI, M., PAXSON, V., AND SOMMER, R. Dynamic Application-Layer Protocol Analysis for Network Intrusion Detection. In *Proc. USENIX Security Symposium* (2006).
- [7] FELDMANN, A., CACERES, R., DOUGLIS, F., GLASS, G., AND RABINOVICH, M. Performance of web proxy caching in heterogeneous bandwidth environments. In *Proc. IEEE INFOCOM* (1999), pp. 107–116 vol.1.
- [8] PAXSON, V. Bro: A System for Detecting Network Intruders in Real-Time. *Computer Networks* 31, 23–24 (1999).
- [9] SALEH, O., AND HEFEEDA, M. Modeling and Caching of Peer-to-Peer Traffic. In *Proc. ICNP* (2006), pp. 249–258.
- [10] WIERZBICKI, A., LEIBOWITZ, N., RIPEANU, M., AND WOŹNIAK, R. Cache Replacement Policies Revisited: The Case of P2P Traffic. In *Cluster Computing and the Grid* (2004), pp. 182–189.
- [11] XIE, H., YANG, Y. R., KRISHNAMURTHY, A., LIU, Y. G., AND SILBERSCHATZ, A. P4P: provider portal for applications. In *Proc. ACM SIGCOMM* (2008), pp. 351–362.
- [12] ZINK, M., KYOUNGWON, S., YU, G., AND KUROSE, J. Watch Global, Cache Local: YouTube Network Traffic at a Campus Network. In *Proc. SPIE* (2008), vol. 6818.