RB-Tracker: A Fully Distributed, Replicating, Network-, and Topology-aware P2P CDN

Andri Lareida, Thomas Bocek, Martin Waldburger, Burkhard Stiller
University of Zurich, Department of Informatics (IFI), Communication Systems Group (CSG)
Binzmühlestrasse 14, CH–8050 Zürich, Switzerland
Email: [lareida|bocek|waldburger|stiller]@ifi.uzh.ch

***Abstract*—Finding a good balance between link congestion and over-provisioning for network service providers is a challenge. Recent advances in transport protocols promise a smoother transfer of bulk background data — traffic that P2P (Peer-to-Peer) file sharing systems are significantly contributing to still today. This paper focuses on reducing the link congestion, and traffic peaks originating from P2P systems by replicating data during periods of low network traffic. For this purpose, RB-Tracker is proposed as an extension of B-Tracker in order to automatically manage the overlay network and replication of content, similar to a Content Delivery Network (CDN). RB-Tracker is designed to manage autonomously a P2P network and to use state-of-the-art mechanisms, such as biased neighbor selection.**

I. Introduction

P2P file sharing applications still account for a large portion of Internet traffic [4], thus, the Future Internet will have to deal with this traffic type, too. P2P-induced traffic contributes to link congestion and drives cost of operations, since it does not consider ISP (Internet Service Provider) domains. Traffic shaping techniques, which discriminate P2P traffic, can have a negative effect on P2P customers [13]. Handling peaks with higher capacity infrastructure leads to over-provisioning. Recent advances in transport protocols [14] promise a smoother transfer for bulk background data compared to TCP (Transmission Control Protocol). Biased Neighbor Selection [2], [3] enables peers to prefer locally close peers, when establishing connections and has proven to reduce intra-domain traffic.

In P2P systems, a tracker manages the overlay network by serving neighbors to peers. Experiments with B-Tracker [8], a balanced and fully distributed tracker, showed higher efficiency in terms of upload bandwidth consumption and better load balancing compared to a DHT (Distributed Hash Table) tracker such as [10], [15]. Motivated by the effectiveness of B-Tracker, RB-Tracker was designed with the goal to reduce traffic peaks and inter-domain traffic — by which RB-Tracker is forming an integrated resource management of networked services, here P2P services. RB-Tracker is based on the main idea of exploiting locality and replicating content during non-peak hours similar to a CDN.

The newly proposed RB-Tracker incorporates three mechanisms in a fully distributed system, which form an integrated management approach for P2P services combined with content-aware distribution demands: (1) the identification of content that is popular and of interest to the user in order to replicate it to the local cache. (2) the identification of network status to determine a feasible time to replicate, that is an off-peak hour. (3) the identification of close peers to replicate from in order to avoid additional intra-domain traffic induced by replication. This paper's main contribution is the design of RB-Tracker and a report on the respective work in progress.

The remainder of this paper is structured as follows: Section II presents work related to RB-Tracker. Section III explains the design of RB-Tracker's mechanisms. Finally, the work is concluded in Section IV where also an outlook on future work is given.

II. Related Work and Concepts

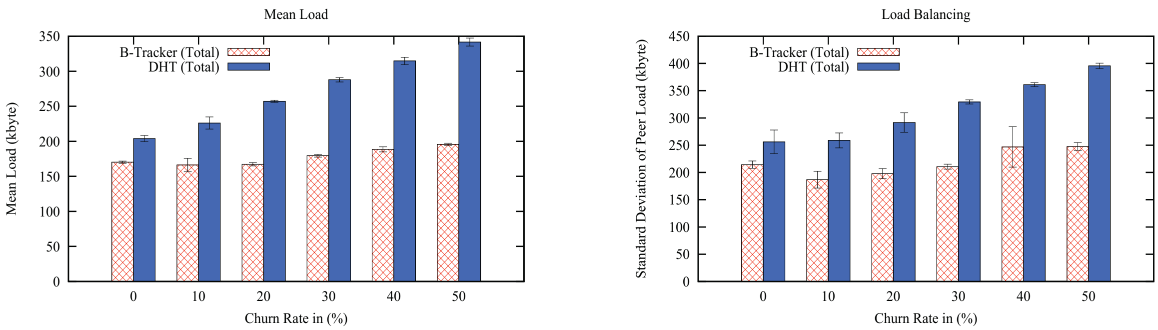
The different concepts to solve the problems stated above include: (1) reducing link congestion, (2) reducing inter-domain traffic, (3) influencing the P2P overlay through neighbor selection, (4) caching and replication, and (5) applying a fully distributed system. After presenting B-Tracker (RB-Tracker's basis) and intermediate results, additional related work addressing one or more concepts (1-5) is introduced.

A. B-Tracker

B-Tracker is a balanced tracker for P2P systems and its usage is not limited to a special kind of P2P system. B-Tracker relies on two stages for tracking. First, it uses a DHT to bootstrap a new peer into the system. Second, once a peer has B-Tracker neighbors, it uses direct messages to query other peers for new providers. Simulations [8] have shown that B-Tracker reduces the tracking overhead in comparison to a DHT tracker as used in [10], [15] and that B-Tracker balances the load among all peers, especially for large swarms and high churn. Load in the context of the B-Tracker is defined as upload bandwidth consumption, since upload is the scarcest resource in P2P systems.

B-Tracker has been implemented as a plugin for the popular BitTorrent client Vuze [9], in order to show simulation results in a realistic experiment, which is based on Vuze instances. The experiment follows a flash crowd scenario, where a new file starts being shared by few seeders and suddenly becomes very popular. For the experiment, a 714 MByte large Ubuntu image was used. The experiment was considered finished when all peers completed their download. The experiment was executed on 5 test bed servers, where each server was running 200 Vuze instances concurrently. The experiment was repeated with the same churn rate 5 times. The experiments applied churn by stopping a percentage, defined by the churn rate, of randomly selected regular peers. Churn rate is defined as the

978-3-901882-50-0



(a) Mean Load

(b) Load Balancing

Fig. 1. Preliminary Results of Ongoing B-Tracker Experiments. Load is defined as the average size of uploaded tracking messages.

percentage of regular peers which are stopped. Once peers are stopped, they are immediately replaced by new peers.

The experiment involved 1,000 active Vuze instances representing peers. There are three types of peers: (a) a DHT bootstrap node, which supports the DHT and does not take part in file sharing. (b) original seeders, which store the file from the beginning. Seeders are present from the beginning of the experiment and show an upload limit of 250 kByte/s. This reflects a standard ISP offering. (c) regular Peers joining the swarm with an exponentially growing inter-arrival time similar to a flash crowd. They also suffer from churn. Regular peers have variable upload limits distributed around the same 250 kByte/s the seeders have.

Figure 1a depicts the amount of tracking data uploaded by one peer on average. Error bars represent the standard deviation between those five repetitions of the experiment. Those results show that B-Tracker is more efficient in terms of upload bandwidth consumption than a DHT tracker, although a DHT is used in both approaches. These results indicate that tracking load increases with increased churn and that B-Tracker scales even better, once churn increases.

Figure 1b outlines current results regarding load balancing. To measure load balance, the standard deviation of uploaded tracking data per peer was calculated. A smaller standard deviation means better load balance. The graph shows that DHT approaches to load balancing are turning worse with churn, whereas B-Tracker first improves and then worsens as well. Moderate churn has positive effects on load balancing in case of B-Tracker.

B. Related Projects

Addressing concept (1), LEDBAT [14] defines an experimental transport protocol with a congestion control mechanism based on delay measurements. It targets background data transfers, which are less important than interactive applications like HTTP (Hyper Text Transfer Protocol). In contrast to TCP, LEDBAT does not wait until packets are dropped. Instead it measures delay and decides based on that measurement if a link is congested. This way LEDBAT uses as much bandwidth as possible, but yields this bandwidth to other protocols.

Addressing concept (2), in support of a reduction of inter-domain traffic, two approaches for P2P file sharing exist: Biased Neighbor Selection (BNS) [2], [3] and Biased Unchoking (BU) [12]. The idea behind BNS is to manipulate a peer's neighbor set in a way that it contains mostly peers that are closely located in terms of network distance. Thus, inter-domain traffic can be reduced significantly. An external service called *oracle* is asked to determine the closeness of two peers. [2] proposed an alternative tracker, which returns a locality biased set of providers. The client, in turn, only asks for additional AS (Autonomous Systems) external providers, once its external neighbor number is below a certain threshold. [3] developed a Vuze plugin called *Ono*, which uses CDNs as oracle exploiting DNS (Domain Name System) redirection-based load balancing done by CDNs. Using a peer's different redirections, a similarity score is calculated. More similar peers are preferred when establishing neighbor relations, which results in a biased overlay network, determined as concept (3). To find sa ufficient number close peers more tracker queries are required than with a regular client. However, in [2] the solution needs a change in the tracker software, which is not yet proven to scale for multiple large swarms. The *Ono* plugin cannot influence the response from the tracker, so it has to query it more often than a regular BitTorrent client until it eventually finds a sufficient number of close providers. Both approaches rely on external server-based services, which contradict the P2P paradigm, concept (5).

[12] proposes BU, which is a BitTorrent-based addition to BNS. BNS can only influence a peer's neighbor set, how data is exchanged is decided by a BitTorrent-specific choking algorithm. BU modifies the choking algorithm in a way that peers being locally close have a higher chance of being unchoked or of receiving data from the peer, respectively. These simulations show that the benefits of BU are most prominent in combination with BNS. [12] also state that beneficial effects already arise, once only a fraction of the swarm uses BNS and BU.

The caching concept (4) originates from CDNs like the Akamai network [1]. It relies on thousands of servers spread across more than 70 countries to deliver content to users. Akamai applies layered caching to bring content closer to the consumer and, in turn, reduces delays. CDNs rely on infrastructure, which has to be upgraded if the number of users increases. However, in P2P systems resources increase as the number of users increases. CDNs focus on user experiences, whereas RB-Tracker's goal is to improve the traffic character-

TABLE I. COMPARISON OF RELATED WORK

Concept	RB-Tracker	MP	CDN	BNS	BU	LEDBAT
(1) Reducing link congestion	+	+	+	-	-	+
(2) Reducing inter-domain traffic	+	+	+	+	+	-
(3) Influencing P2P overlay	+	-	-	+	-	-
(4) Caching and replication	+	+	+	-	-	-
(5) Applying a fully distributed system	+	-	-	-	-	+/-

istics of the content distribution in P2P systems.

MirrorPlane (MP) [11] (MP) applies P2P caching (4) to reduce inter-domain traffic (2). MP is a modified BitTorrent client, which monitors swarms and schedules replication. MP's most prominent drawback is scalability, since it relies on an ISP deployed server which monitors all peers. Their calculations state that roughly 130 Mbit/s of bandwidth are required for monitoring one million peers. A fully distributed system (5) can overcome this issue by spreading the load to all peers using the system. Furthermore, the server maintenance cost can be avoided. Despite these drawbacks, simulation results of [11] show significant potential for a P2P CDN.

Thus, Table I indicates an overview of the related work and its concepts (1-5) applied. On one hand, BNS and BU significantly reduce inter-AS traffic by influencing the P2P overlay structure, while they do not consider network load and cannot help reduce traffic peaks. LEDBAT, on the other hand, is considering local observations in order to identify congested links, which occur often during peak hours. Caching is applied by CDNs like Akamai.

III. DESIGN

The RB-Tracker approach incorporates three distributed mechanisms: Mechanism (1) determines, which files to replicate, mechanism (2) determines when to replicate, and mechanism (3) determines, which peers to select for a replication.

A. Determining Files to Replicate

CDNs use caching to bring content closer to their users [1]. RB-Tracker does the same, but additionally replicates content to peers since these are less stable than CDN servers. The determination of files to replicate depends on the goals of caching. For RB-Tracker this goal is to reduce peak loads. Therefore, popular content is replicated, since popular content causes most traffic. Furthermore, similar to the Radiommender [7] approach, RB-Tracker replicates recommendable files creating an incentive for the replicating peer to provide disk space.

A peer is downloading and sharing files that the peer itself has an interest in. Therefore, a peer's file set is an expression of its interest. Under these assumptions, RB-Tracker can exploit B-Tracker's mesh structure to compare file sets of neighbors. Highly overlapping file sets indicate common interest and, therefore, additional content can be found and replicated without any user interaction.

Figure 2 shows an example of a neighbor set of peers, each letter represents a file being shared. Such a mesh structure is typical in B-Tracker and can be exploited by RB-Tracker to gather information about files shared among neighbors.

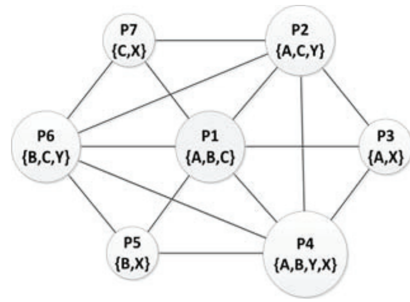


Fig. 2. Example of a Simplified Neighbor Set over Several Files

The following example mechanism has been designed to address popularity: Peer P_1 collects file sets of its neighbors: $\{ACY\}$, $\{AX\}$, $\{ABYX\}$, $\{BX\}$, $\{BCY\}$, $\{CX\}$ and for each file set the similarity is calculated by counting common files to its file set $\{ABC\}$: $P_2 = 2$, $P_3 = 1$, $P_4 = 2$, $P_5 = 1$, $P_6 = 2$, $P_7 = 1$. In the next step, unknown files from these file sets receive the similarity score of their set. Multiple occurrences are added: $Y = 2+2+2 = 6$, $X = 1+2+1+1 = 5$. This shows that, even though X is more popular (shared by 4) than Y (shared by 3), Y is the better candidate to replicate, because it has a higher chance of being in the P_1's interest. Further evaluations of additional recommendation schemes determine work in progress.

B. Finding the Right Time

An RB-Tracker peer uses passive network measurement approaches similar to LEDBAT. By adding a time stamp to messages sent between peers, a peer measures delay and builds statistics based on its measurements. Based on these statistics a peer can determine when delay is increasing. An increase in the delay between two peers can have two causes: (1) the link is congested or (2) the neighbor is busy. In any case the neighbor is not a feasible source for replication and another peer needs to be found.

C. Determining Closeness

A service that provides a locality function for two IP addresses can be used to determine how close two peers are. One example of such a service comes out of the SmoothIT project [6] and was used in [12]. Since RB-Tracker is a fully distributed system, the following mechanism is used. A peer in RB-Tracker decides, if it should replicate from a neighbor using the number of AS hops between them. To calculate the AS hop distance vector, a trace route tool in combination with an IP to AS map is used. Such a hop distance vector has to be calculated only once, when two peers first meet. Evaluation of different threshold values determines work in progress.

A simplified example illustrates this mechanism. A peer P_1 queries its neighbor P_3 for additional providers for file A. P_3 already knows those AS hops to P_1 from the trace route performed, when the connection was established. The AS hop vector from P_3 to P_1 is: $V_{P3-P1} = \{AS33,\ AS22,\ AS11\}$. P_3 knows two potential peers for replying to P_1. Their AS hop vectors are known as well: $V_{P3-P2} = \{AS33, AS66, AS88\}$,$V_{P3-P4} = \{AS33, AS22, AS44\}$. Based on these vectors P_3 returns the peer with the longest matching vector to V_{P3-P1}. In this example, this is V_{P3-P4}, since the first two hops match and, therefore, P_4 is returned in response to the query.

IV. Summary and Conclusions

This paper describes a first step in the process of developing an automatically configured P2P CDN, termed RB-Tracker, which aims to reduce peak load and inter-domain traffic. Combining concepts from P2P systems and CDNs, RB-Tracker incorporates three mechanisms to improve automatic network management in the application layer. A larger set of further and extensive experiments are required to validate these claims under many additional considerations.

While the Future Internet will see P2P traffic as well as content distribution services integrated, the new proposal of RB-Tracker's approach in this paper incorporates the respective demands into a fully distributed system. As such, the RB-Tracker forms an integrated management approach for P2P services combined with content-aware distribution demands by addressing mechanisms for identification of interesting content, network status, and close peers for replication.

For the evaluation of RB-Tracker in respect to inter-domain traffic, it is necessary for the experiments to run in different ASs, because inter-AS traffic has to be measured to show the expected benefits. It is planned to move the experiment into EmanicsLab [5], which consists of several servers spread across Europe with nodes located in different ASs. In this paper, a direct replication and caching mechanism was presented, which combines the P2P and CDN requirements for a Future Internet.

Additional and more sophisticated mechanisms will be evaluated and respective experiments with full files, compared to replicating chunks, have to be carried out. Further recommendation mechanisms, such as Radiommender [7] and the one described in Section III, have to be evaluated. Since this is a key aspect, evaluation of several recommendation schemes will have a high priority. Furthermore, work is required in the design and evaluation of network monitoring mechanisms. Besides the simple approach described in this paper, where monitoring of link congestion is done during file transfers, more sophisticated algorithms have to be evaluated that also include forecasting of link congestion for optimal scheduling of upcoming file replication.

The recommendation-based caching of RB-Tracker leads to a biased network of peers. This bias is based on common interest in files and locality. Therefore, common interests and locally close peers are clustered similar to a social network. Adding a social network component to the file sharing community opens a whole new field.

Another question is if users have an incentive to contribute resources in a P2P CDN. The success of P2P systems shows that there exist enough altruistic users which are willing to provide resources [11]. However, for a replication mechanism it is beneficial when users provide as much storage space as available. This means, on the one hand, incentive mechanisms need to be evaluated how to encourage users to contribute as much as possible. On the other hand, an evaluation should examine the impact of storage size with respect to the overall performance of a P2P CDN.

Acknowledgments

This work has been performed partially in the framework of the EU ICT STREP SmartenIT (FP7-ICT-2011-317846) as well as EU ICT NoE Flamingo (FP7-ICT-2011-318488) supported by the European Commission under its Seventh Framework Programme.

References

[1] Akamai Technologies. (2013, Jan.) Akamai homepage. [Online]. Available: http://www.akamai.com

[2] R. Bindal, P. Cao, W. Chan, J. Medved, G. Suwala, T. Bates, and A. Zhang, "Improving Traffic Locality in BitTorrent via Biased Neighbor Selection," in *26th IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2006, pp. 66–66.

[3] D. Choffnes and F. Bustamante, "Taming the Torrent: A Practical Approach to Reducing Cross-ISP Traffic in Peer-to-Peer Systems," in *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 4, 2008, pp. 363–374.

[4] CISCO. (2013, Jan.) Cisco Visual Networking Index: Forecast and Methodology, 2011-2016. [Online]. Available: http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360_ns827_Networking_Solutions_White_Paper.html

[5] EMANICS. (2013, Jan.) EmanicsLab. [Online]. Available: http://www.emanicslab.org

[6] J. Gimenez, M. Rodriguez, H. Hasan, T. Hoßfeld, D. Staehle, Z. Despotovic, W. Kellerer, K. Pussep, I. Papafili, G. Stamoulis *et al.*, "A New Approach for Managing Traffic of Overlay Applications of the SmoothIT Project," in *2nd International Conference on Autonomous Infrastructure, Management and Security (AIMS08)*, 2008.

[7] F. Hecht, T. Bocek, N. Bar, R. Erdin, B. Kuster, M. Zeeshan, and B. Stiller, "Radiommender: P2P On-line Radio with a Distributed Recommender System," in *IEEE International Conference on Peer-to-Peer Computing (P2P)*, sept. 2012, pp. 73 –74.

[8] F. Hecht, T. Bocek, and B. Stiller, "B-Tracker: Improving Load Balancing and Efficiency in Distributed P2P Trackers," in *IEEE International Conference on Peer-to-Peer Computing (P2P)*, 2011, pp. 310–313.

[9] A. Lareida, "Development and Implementation of the B-Tracker Approach on a BitTorrent Client," Master's thesis, University of Zurich, Zurich, 2012.

[10] A. Loewenstern. (2013, Jan.) DHT Protocol. [Online]. Available: http://www.bittorrent.org/beps/bep_0005.html

[11] A. Mondal, I. Trestian, Z. Qin, and A. Kuzmanovic, "P2P as a CDN: A new service model for file sharing," *Computer Networks: The International Journal of Computer and Telecommunications Networking*, vol. 56, no. 14, pp. 3233–3246, 2012.

[12] S. Oechsner, F. Lehrieder, T. Hoßfeld, F. Metzger, D. Staehle, and K. Pussep, "Pushing the Performance of Biased Neighbor Selection through Biased Unchoking," in *IEEE International Conference on Peer-to-Peer Computing (P2P)*, 2009, pp. 301–310.

[13] R. Pereira and T. Vazao, "On the Impact of P2P File Sharing Traffic Restrictions on User Perceived Performance," in *International Conference on Information Networking (ICOIN)*, jan. 2008, pp. 1 –5.

[14] S. Shalunov, G. Hazel, I. BitTorrent, J. Iyengar, F. College, M. College, and M. Kuehlewind, "Low Extra Delay Background Transport (LEDBAT)," IETF RFC 6817, Dec. 2012.

[15] Vuze Inc. (2013, Jan.) Vuze The Only Way to Download Torrents. [Online]. Available: http://www.vuze.com