PeerCollaboration

Thomas Bocek[1] and Burkhard Stiller[1,2]

[1] Department of Informatics IFI, University of Zurich, Switzerland
[2] Computer Engineering and Networks Laboratory TIK, ETH Zurich, Switzerland
[bocek|stiller]@ifi.uzh.ch

Abstract. Increasing traffic due to increased bandwidth or the number of users calls for scalable systems, which can be built with peer-to-peer (P2P) mechanisms. Scalability is a key issue for systems that rely on many participants, such as large-scale collaboration system. This paper introduces PeerCollaboration, a fully decentralized P2P collaboration system for documents, which is robust against malicious behavior, provides an efficient content search, and offer mechanisms for distributed control. Typical tasks in PeerCollaboration are searching, retrieving, creating, changing, and maintaining documents in a collaborative manner. Three problem areas are investigated within this system. The first problem focuses on similarity search on top of existing P2P networks and highlights a novel algorithm, which outperforms compared approaches. The second problem deals with incentive schemes, which work with indirect reciprocity. Thus, the novel and robust incentive scheme finds more reciprocities than with compared approaches. The third problem focuses on user-based voting mechanisms.

1 Introduction and Motivation

The Internet is steadily growing, because Internet users and Internet bandwidth of end-users are steadily increasing. According to [8], Internet network capacity grows 55% per year. This growth is also observed for end-users bandwidth. Nielsen's Law of Internet bandwidth states that high-end user's connection speed grows by 50% per year, which yields in a 57 times increase in 10 years [7].

This situation leads to the need for scalable systems, which scale with the number of participating users. Client-server systems have resource limits and as soon as this limit is reached, the system needs to be upgraded or replaced. Scalable P2P systems do not have such limitations, because every participating user contributes resources. Such P2P systems can reduce upgrading costs, increase stability, and improve fault-tolerance.

Large-scale centralized collaboration systems, such as Wikipedia, which depend on monetary donations for hardware infrastructure, cannot offer services, if donations decrease and infrastructure costs cannot be paid. Thus, the goal of PeerCollaboration is to demonstrate a scalable, robust, fault-tolerant, and fully decentralized P2P collaboration system for documents that provides robust incentives, an efficient content search, and offers mechanisms for distributed control. PeerCollaboration can be used for any kind of collaborative document writing, such as scientific reports, project deliverables, or online articles.

2 Decentralized P2P Collaboration System

In collaboration systems, users work together to reach a common goal. Users collaborate using communication and management tools. Examples for communication tools are Instant Messenger (IM), Email, or Voice over IP (VoIP) conferencing tools, and examples for management tools are calendar applications, project management tools, or social software. Decentralized collaboration systems work without central servers. Coordination among users is achieved in a P2P manner. Existing decentralized collaboration applications, such as Microsoft Groove, or P2P Wiki approaches PIKI [6], Wookie [10], DistriWiki [5], and [9] support users in a coordinated creation and sharing of documents. Table 1 compares distinct features of those P2P-based Wiki systems.

Table 1. P2P-based Wiki applications comparison

Application	*Distinct Features*
PIKI [6]	Concurrent editing, version control, full-text exact keyword search, semantic linking
Wookie [10]	Consistency and replication
DistriWiki [5]	No distinct features
Urdaneta et al. [9]	Fault tolerance, security issues, placement mechanisms
PeerCollaboration	Keyword similarity search, incentive schemes, user-based voting mechanism

The following use cases show the functionality, which are of particular interest for PeerCollaboration, of a general collaboration system. In use case 1, (cf. Figure 1), a user searches for the “Linux” article. The user misspells the search term and types “lniux” instead of “Linux”. PeerCollaboration searches for similar entries to “lniux”. The search can find misspellings up to the edit distance of 2. PeerCollaboration shows a preview of articles with a download link containing the keyword searched for. In use case 2, a user searches for an article about “Java”. The user types “java”. The result list including the preview does not contain this article. Thus, the user writes and submits the missing article. PeerCollaboration stores and indexes this articles after verifying that the user has contributed to the system beforehand. In use case 3 a user wants to search for an article about “Firefox”. One of the preview article shows the right article and the user downloads it. However, the user finds wrong information in this article. Thus, the user corrects and submits the article. This starts the voting process, where previous authors vote for or against the modification. Once the majority of previous authors agrees, the corrected article is published.

3 Problem Statement

Free-riders are peers that do not provide resources, *e.g.,* uploading documents. Such behavior limits scalability, because fewer peers have to provide these resources. Thus, incentive schemes are used to encourage peers to contribute. A very popular and widely used incentive scheme is Tit-for-tat (TFT). With TFT, users may only download as much as they upload. This incentive scheme keeps a per-peer history of resource transactions on every peer that is solely based on local observation. Thus, peers have a limited view of all transactions to peers with direct reciprocity. Peers with direct reciprocity can exchange resources from each other, while peers with indirect reciprocity can exchange resources only with at least one intermediate peer in between. However, with TFT indirect reciprocity is not detectable and exploitable. Thus, an incentive scheme that works with indirect reciprocity is essential in networks with many indirect connections.

Similarity search in P2P networks is important because otherwise peers that search for misspelled documents fail to find those. In flooding-based P2P systems, each peer can evaluate the search query and do a similarity lookup, however, these systems do not scale, because search queries are sent to all peers. Scalable P2P systems such as Distributed Hash Tables (DHT) allow for an exact search only. Thus, a fast similarity search algorithm on top of a DHT is essential to find documents with misspelled text.

Changing data in a P2P system is prone to attacks, because malicious peers can insert incorrect information or delete correct information, which leads to a lower quality of data. While quantitative parameters like text length and variation of words can be measured automatically, the qualitative assessment of human contributions can be performed only by humans. Thus, a voting mechanism with human interaction is essential to maintain the quality of data.

All three problem statements are addressed by PeerCollaboration, which discourages peers that download but never upload documents, enables similarity search to find documents with misspelled text or with a misspelled search query, and provides decentralized control for managing changes.

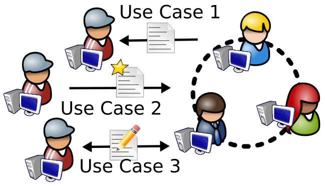

Fig. 1. Three use cases. Article search, publishing article, and modifying article

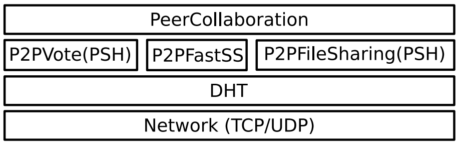

Fig. 2. Layered architecture with PeerCollaboration on top

4 Design and Evaluation

The PeerCollaboration architecture can be split up into four layers as shown in Figure 2. The base layer determines the network layer, with Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) providing a socket interface to the DHT. Based on that, the DHT layer provides the interfaces `put()` and `get()` to the upper layers. On top of the DHT, P2PVote, P2PFastSS [2], and P2PFileSharing are built. The incentive scheme PSH [1] is built into P2PVote and P2PFileSharing. These three components offer the following interfaces. For P2PFastSS the interfaces are `search` and `index`, for P2PFileSharing, `publish` and `download`, and for PeerVote, `proposeChange` and `vote`. On top of these three components, the PeerCollaboration application is built.

All three mechanisms and algorithms addressing the aforementioned problem statements have been evaluated. First, PSH [1] has been evaluated in a simulation and on EMANICSLab using input data from ThePirateBay. PSH has been compared to TFT, and two variants of PSH: PSH_r, and CompactPSH. Second, the underlying algorithm of P2PFastSS [2], FastSS and its variants have been evaluated in a simulation and on PlanetLab. FastSS has been compared to Burkhard-Keller tree, dynamic programming, NR-grep, keyword tree, neighborhood generation, and n-grams. [3]. Third, PeerVote [4] has been simulated on EMANICSLab evaluating malicious behavior, churn, and other parameters.

References

1. T. Bocek, Y. El-khatib, F. V. Hecht, D. Hausheer, and B. Stiller. CompactPSH: An Enhanced Private and Shared History-based Incentive Scheme. *under review.*
2. T. Bocek, E. Hunt, D. Hausheer, and B. Stiller. Fast Similarity Search in Peer-to-Peer Networks. In *11th IEEE/IFIP Network Operations and Management Symposium (NOMS)*, Salvador, Brazil, April 2008.
3. T. Bocek, E. Hunt, and B. Stiller. Fast Similarity Search in Large Dictionaries. Technical Report ifi-2007.02, Department of Informatics, University of Zurich, April 2007.
4. T. Bocek, D. Peric, F. Hecht, D. Hausheer, and B. Stiller. PeerVote: A Decentralized Voting Mechanism for P2P Collaboration Systems. In *Proceedings of the 3nd International Conference on Autonomous Infrastructure, Management and Security (AIMS 2009)*, Twente, The Netherlands, July 2009.
5. J. C. Morris and C. Lüer. DistriWiki: A Distributed Peer-to-Peer Wiki. In *Proceedings of the 2007 International Symposium on Wikis (WikiSym '07)*, pages 69–74, Montreal, Quebec, Canada, October 2007.
6. P. Mukherjee, C. Leng, and A. Schürr. Piki - A Peer-to-Peer based Wiki Engine. In *Proceedings of the 2008 8th International Conference on Peer-to-Peer Computing (P2P '08)*, pages 185–186, Washington, DC, USA, September 2008.
7. J. Nielsen. Nielsen's Law of Internet Bandwidth, 2008. http://www.useit.com/alertbox/980405.html.
8. L. Roberts. Beyond Moore's Law: Internet Growth Trends. *Computer*, 33(1):117–119, Jan 2000.
9. G. Urdaneta, G. Pierre, and M. van Steen. A Decentralized Wiki Engine for Collaborative Wikipedia Hosting. In *Proceedings of the 3rd International Conference on Web Information Systems and Technologies (WEBIST)*, Barcelona, Spain, March 2007.
10. S. Weiss, P. Urso, and P. Molli. Wooki: A P2P Wiki-Based Collaborative Writing Tool. In *Proceedings of the 8th International Conference on Web Information Systems Engineering (WISE)*, pages 503–512, Nancy, France, 2007.