ISSN 2395-1621

Distributed Hybrid System For File Sharing

[#1]Naina Kumari, [#2]Abhishek Nair, [#3]Yashwant Padekar, [#4]Vinod Thorat

[1]naina.sinha925@gmail.com
[2]abhishek21nair@gmail.com
[3]yash.padekar@gmail.com
[4]vinod.b.thorat@gmail.com

[#1234]Department of Computer Engineering
JSPM's
Imperial College Of Engineering & Research
Wagholi, Pune – 412207

ABSTRACT

Wireless Peer-to-Peer (P2P) file sharing is widely used as one of the major applications of ad hoc networks. This is mostly motivated by the current advances in fast speed technologies of wireless communication and high traffic demand for applications of P2P file sharing. For achieving the ambitious goal of wireless P2P network, we require a scalable topology control protocol for solving the neighbor discovery problem and network connectivity problem. Hence, we know that the file sharing topology control mechanism should be application driven. So that we have to try to accomplish an efficient connectivity among mobile devices in order to better serve the file sharing application. We propose a protocol which consists of two parts, namely Community-Based Asynchronous Wakeup (CAW) and Adjacency Set Construction (ASC). Our proposed protocol is able to enhance the fairness and provide incentive mechanism in wireless P2P file sharing applications and capable of increasing the energy efficiency. In this paper, we propose a scheme to apply peer-to-peer file sharing on network coding which employs a peer-to-peer network to distribute files resided in a web server or a file server.

Keywords—wireless networking, topology control, file sharing, P2P systems, network protocols.

ARTICLE INFO

Article History

Received: 26th October 2015

Received in revised form :

28st October 2015

Accepted: 2nd November, 2015

Published online :

4th November 2015

I. INTRODUCTION

Peer-to-peer (P2P) computing has been proposed as a promising technology that will reconstruct the architecture of distributed computing. This is because it provides various resources (including computation, storage) at the edge of the Internet, with lower cost of occupancy, and at the same time enjoy many desirable features (e.g., scalability, autonomy, etc.). Since, from last 10 years P2P computing technology has spurred increasing interests in both academic and industrial communities. As such, there are increasingly more applications being developed based on this criterion. For example, scientific computation (e.g., BOINC and Folding@home), digital content sharing (e.g., Naspter, Gnutella, and Shareaza), collaborative groupware (e.g., Groove), instant messages (e.g., ICQ) and so on.However, there has not been difficulty to study the architecture and concepts of P2P systems. The architecture of a system of high-level applications is the cornerstone that are implemented upon it, the conception of P2P architecture is crucial to realizing its full promising. Hence a study is important because: (a) It helps developers, researchers, and users to better appreciate the differences and relationships between other distributed computing paradigms and P2P systems (e.g., client-server and grid computing). (b) It allows us to be conscious of the hypothetical merits of P2P computing for recently emerging application demands, and to determine the most suitable architecture for them. (c) It enables us to determine the architectural issues that are crucial to a P2P system's performance, reliability, scalability, and other features. Therefore, we address this chapter to examine and summarize the framework of P2P systems and some related issues.

Based on existing systems we first present a classification of P2P architectures that have been developed. On one part, some P2P systems are supported by centralized servers. On the other part, pure P2P systems are completely decentralized.Between these two extremes are hybrid systems where nodes are categorized into two layers: the upper tier servers and the lower tier common nodes. Second, we will conduct an extensive comparison among these three types of architectures. Third, we will check how peers in different architectures define their neighbors (those that are directly connected)—statically or dynamically, and figure out the holding techniques for dynamic reorganization of peers that allow communities to be designed based on some common interests between the nodes. We will also observe how nodes that are relatively powerful can be exploited to shoulder more responsibilities. This survey paper is sectionalized as follows: In the next section, we discuss the framework and related work on P2P networking. In Section 3 and 4, we describe our proposed topology control protocol. Finally, we provide some concluding remarks in the last section.

II. ADVANCEMENT

There is a plethora of previous work related to the topology control problem.The first category is topology control for ad hoc wireless network, without any P2P concept. The major local points of these previous results are energy efficiency and connectivity. Some of these previous schemes can generate performance gain in terms of throughput and delay. Only and explicitly take fairness into consideration.The second category is wired P2P, including P2P file sharing protocols. They are usually divided into three classes: centralized P2P file sharing protocol, decentralized unstructured P2P file sharing protocols, and decentralized structured P2P file sharing protocols. Recently, proximityaware configuration of P2P network connection topology has also been extensively studied. Most of these protocols are designed for file exchange in a wired network. Respectively, these protocols do not participate in constructing link layer topology.

III.NAPSTER: SHARING OF DIGITAL CONTENT

Music file sharing is perhaps one of the fastest growing applications in the Internet,and Napster certainly played a critical role in facilitating music file exchange over the net. In Napster, each user (computer) acted as a producer of content. Hence, one can view the system as a collection of MP3 files that are distributed over the personal computers of Napster users. To enable users to locate music files, Napster employed a centralized server that stored the locations of the nodes that own the files. Thus, a request will be channeled to the central server to obtain the list of owner nodes. Certain data exchange between two peers can proceed without any further intervention from the central server. Generally, Napster provided three basic functions: search engine, file sharing, and Internet relay chat. The search mechanism is a dedicated server, which realizes the function of resource location. File sharing provides a mechanism to trade MP3 files among peers, without using the storage space of the central server. Internet relay chat provides a way to find and chat with other online peers. A simple MP3 trading procedure in Napster can be divided into three phases: joining Napster, resource discovery, and downloading files. First, through various connections (e.g.,LAN or dial-up), a user was able to join the Napster network by connecting to the central server and completing a registration process on the central server for the connection. Second, a peer queries the central server by sending out a lookup message. After receiving the message, the central server first looked up against the index in its local repository and then returned a list of nodes that contained the desired files. Thus, resource discovery was accomplished with the help of the central server. Note that the central server maintained an index of metadata of shared files on all online peers and corresponding data (e.g., IP address), but it did not store MP3 files itself. Finally, the query peer established direct connections with the desired peers, and downloaded files without the involvement of the central server.

- **Gnutella: The First "Pure" P2P System**

Gnutella is a purely decentralized P2P system. No central authority is in charge of the network's organization, and there is no prejudice between the client and the server. Nodes in the system connect to each other directly through a specific software application. The Gnutella network expands as new nodes join the network and collapses as all nodes leave the network. In this sense, it is a software-based network infrastructure. Routers, switches, and hubs are not necessary to enable communication at this level.Briefly speaking, the basic operations of Gnutella include joining or leaving network, searching and downloading files: − Joining or leaving network. When a node joins the Gnutella network, it indicate its presence by sends a "PING" message. The "PING" message is forwarded to other nodes by broadcast scheme. When other nodes receive the "PING" message, they feed back "PONG" messages as replying, which means they get know about the entry of the newcomer. From the "PONG" messages, the newcomer node can get the information about existing nodes and settle its own communal with some of them. When a node leaves the network, it is not necessary for the node to broadcast its neighbors. On the converse, each node needs to probe its neighbors with "PING" messages at regular intervals to confirm whether its neighbors are still online. If no response is returning from the node, it will take as these nodes have left the network, and then update the list of its neighbors. − Searching and downloading files. When a node wants to find certain files, it asks its neighbors by issuing a "lookup" message, and its neighbors in turn forward the message to all of their own neighbors in the same manner. Those who have the desired files send "hit" messages to the query initiator, which are reversely routed back along the same path as the "lookup" message has routed. Through broadcast the "lookup" message via nodes' neighbors, a good recall of the searching can be achieved. The relaying process goes on until the entire network is covered or the TTL (time-to-live) value of the lookup message reaches zero. Now the original node may have many replies at hand, and can choose some nodes to connect and then download the desired files. Moreover, each message is attached to a unique identifier. When a node receives a message with an identifier it has seen, it will drop the message so that message loops can be avoided.

IV.DISTRIBUTED FILE SYSTEMS

In subjected paper, a method to play multimedia data smoothly on a pure P2P-based distributed e-Learning system is proposed and its implementation is described. In our system, multimedia data is divided into multiple fragments by time set, each fragment manages by each media agent, and their media agents are linked bidirectional. If division range of multimedia data is large, multimedia data cannot be played smoothly because a lot of agents have to migrate to a requesting node simultaneously. Therefore, we hatched the timing of sending a requesting message to next agent. This method enabled smooth play of multimedia data[1].

Looking further on the the principles of P2P file-sharing define basic operations for users to manage a local repository, by publishing andexchanging documents with their peers.In the paper[2] applying these principles to wiki pages,proposed have obtained an original wiki system that supports a non-traditional collaboration model. In this collaboration model, users create new documents, download them from others, and store them in their local repository just like any other files they could share in the underlying file-sharing application. The wiki functionality is simply an additional layer used by the web browser, whereby this web browser becomes an editing tool. Users can edit their local documents and this way produce new versions, that they link to the previous versions and publish locally. This versioning process, materialized by a link, produces a graph of linked documents,defined in the greater context of our file-sharing model. This graph of documents can then be queried using general graph query principles. The useful queries in P2Pedia include finding the parent versions ofa document, finding the children versions (edits), and by applying a transitive closure we can find all the descendent edits of a document.For this to work, the underlying implementation of the queries does not need to know the semantics of the “parent” relation, any more than the database needs to know that the documents being stored are wiki pages. Ultimately, this application – a full-fledged wiki engine – is simply an application of general file-sharing principles, with the extra feature of links between documents. Similarly, the notions of trust that support the collaboration model, by allowing users to select the existing versions, are applicable to other file-sharing applications.In future work, we intend to further customize our application P2Pedia, and deploy it as a note-taking tool, for several courses at Carleton University, and in an area high-school.

In the surveyed paper[3] the conjunctive matching criterion of today’s P2P file-sharing systems, poor data description bounds query accuracy. Concerns help solve to this problem by automatically tuning local descriptors using those of other peers. Our provisional findings demonstrate that it is possible to improve query accuracy up to 30% at a tunable cost by probing for better file descriptions and sampling result sets. Furthermore, this is accomplished in a fully distributed pattern and is therefore suitable for general P2P environments. Ours is the first work we know of that automatically tunes the descriptions of shared data and could be extended beyond the P2P environment to describing all non-self-describing binary data. We are now considering ways of better controlling exactly where probes are directed (i.e., less or more popular files). We are also developing models to help in tuning probing threshold and sampling values manner. We are also considering different ways of sampling the network to yield cardinality estimations on shared data for the sake of tuning sampling rates.Finally, proposed system is building test data from traces from the Gnutella network using a tool we recently developed – no such data currently exists, esp. the lack of relevance judgment between queries and shared files in Gnutella network.

V. HYBRID P2P SYSTEMS

This paper presents a work that developed a P2P file sharing application for use on mobile devices that have a connection to the Internet through the GPRS network. For achieving this, the various existing P2P architectures were weighted. At the end of the review and having taken into consideration the peculiarities of mobile devices with their small resources, the semi centralized or hybrid architecture was proposed for a feasible mobile P2P file sharing system. The work has then made it possible to create a working solution to the problem of file sharing irrespective of the location of the mobile peers. Moreover, the work if properly optimized is a feasible option for mobile file sharing in second generation mobile networks with GPRS capability. It has been shown that the network operators which is major units of the system that need to focus on for the success of the P2P file sharing are the GPRS connection, MMS and SMS[4].

The Gnutella network continues to be obsessed by poor search performance. On improving the network query mechanisms prior works are focused . In this skill, we show that the failure stems from a mismatch between the way different query for objects and users name . Advancements in network connections and search mechanisms alone cannot focus this problem. To transform the failed queries we propose simple mechanisms. We carefully describe mechanisms to keep pertinent transformations to the intent of the original query. Through extensive imitations, we show that we can achieve significant improvements in query concert. Across a middleware, we show that our approach is practical. We show hints that the approach is relevant through subjective analysis. By using information retrieval techniques future work should build better transformation mechanisms that are amenable to peer to peer systems of the distributed nature. More sophisticated information retrieval techniques are likely to added to improve on our results[5].

We proposed a new hybrid peer and object reputation system as well as an improvement to the Scrubber peer reputation system, as strategies to reduce content pollution in P2P file sharing systems. We also performed an extensive evaluation of both systems, comparing their effectiveness against the Credence object reputation system, for various system and peer behavior configurations. Our main conclusions are: (1) the hybrid system converges much faster to a maximum efficiency than the other strategies, even under collusion and Sybil attacks, (2) the hybrid system is less sensitive to parameter setting than Scrubber, providing costeffectiveness for various configurations, and (3) all three systems depend greatly on user cooperation to give reliable feedback on the status of downloaded objects, although the hybrid system is still able to restrain pollution dissemination even in very uncooperative and unreliable communities. Future work includes further evaluating the systems, especially under more sophisticated attacks (e.g., traitor, whitewashing), prototyping and extending our systems to other application domains[6]. We propose a novel hybrid shared-data/shared-nothing storage architecture for large-scale and data-intensive applications. We first started an overview of the system architecture, and after that we describe the methodology of data organization, data partitioning, query and transaction processing in the proposed system. We match the proposed system with existing architectures of parallel database and conduct experiments to verify our design. The proposed scheme can have the assets of two main parallel database architectures, and it is much more convenient applications with complex query patterns. The experimental and analytical results have shown the possibility of the hybrid architecture. In the future, we conceive to work on detailed design and implementation of the hybrid system, including data partitioning and query processing methods[7] .

VI. CONCLUSION

In this paper, we have proposed a new localized topology control protocol, which is application driven.The proposed scheme is designed for a wireless P2P file sharing network. Our proposed scheme is based on several useful but simple policies, which, we believe, when efficiently deployed in the environment considered in our study, could enhance the life-time as well as the effectiveness of file sharing among the peers. In our proposed algorithms, we do have physical layer control actions (e.g., controlling who is the neighbor). We strongly agree that an important next step in this research is to propose an efficient “crosslayer” design for file sharing network topology control. Moreover, in the current paper, we have taken a more objective approach in considering the various topology configuration criteria. Furthermore, an even more detrimental situation would be having some malicious users who drop important control messages or fake them, possibly leading to the formation of an inefficient cluster.

REFERENCE

[1]M.Higashino, Tadafumi Hayakawa, Kenichi Takahashi, Takao Kawamura, and Kazunori Sugahara,”Management of Streaming Multimedia Content using Mobile Agent Technology on Pure P2P-based Distributed e-Learning System”,

[2]Alexander Craig, Alan Davoust and Babak Esfandiari,"A Distributed Wiki System Based on Peer-to-peer File Sharing Principle",

[3]Dongmei Jia Wai Gen Yee Linh Thai Nguyen Ophir Frieder,"Distributed, Automatic File Description Tuning in Peer-to-Peer File-Sharing Systems",

[4]O.O. Abiona1, A. I. Oluwaranti2, T. Anjali3, C. E. Onime4, E.O. Popoola5, G.A. Aderounmu6, A. O Oluwatope7 and L.O. Kehinde8,"Architectural Model for Wireless Peer-to-Peer (WP2P) File Sharing for Ubiquitous Mobile Devices",

[5]Surendar Chandra ,William Acosta,"Using query transformation to improve Gnutella search performance",

[6]Cristiano Costa Jussara Almeida,"Reputation Systems for Fighting Pollution in Peer-to-Peer File Sharing Systems",

[7]Huaiming Song †, Xian-He Sun †, Yong Chen,"A Hybrid Shared-nothing/Shared-data Storage Architecture for Large Scale Databases",

[8]Andrew Ka-Ho Leung ,Yu-Kwong Kwok,"On Localized Application-Driven Topology Control for Energy Efficient Wireless Peer-to-Peer File Sharing",

[9]Q.H. Vu et al., DOI 10.1007/978-3-642-03514-2_2, © Springer-Verlag Berlin Heidelberg 2010,"Peer-to-Peer Computing",

[10]Min Yang ,Yuanyuan Yang, Fellow, IEEE,"Applying Network Coding to Peer-to-Peer File Sharing",

[11]TITLE :Hybrid Peer-to-Peer DNS
AUTHOUR :Ricardo Sancho, Ricardo Lopes Pereira

