ISSN 2395-1621

Distributed Hybrid System for File Sharing

[#1]Abhishek Nair, [#2]Yashwant Padekar, [#3]Naina Singh

[1]abhishek21nair@gmail.com
[2]yash.padekar@gmail.com
[3]naina.sinha925@gmail.com

[#123]Department of Computer Engineering

JSPM's
Imperial College of Engineering & Research
Wagholi, Pune--412207

ABSTRACT

File sharing is an essential daily task. This includes for instance the sharing of pictures, music, and learning material. Common file sharing applications, however, are not suitable for mobile networks as the induced network load is too high. Wireless Peer-to-Peer file sharing is widely used as one of the major applications of ad hoc networks. This trend is largely motivated by the recent advances in high speed wireless communication technologies and high traffic demand for P2P file sharing applications. To achieve the ambitious goal of realizing a practical wireless P2P network, we need a scalable topology control protocol to solve the neighbor discovery problem and network organization problem. Indeed, we believe that the topology control mechanism should be application driven in that we should try to achieve an efficient connectivity among mobile devices in order to better serve the file sharing application. We propose a new protocol which consists of two components, namely Adjacency Set Construction and Community-Based Asynchronous Wakeup. Our proposed protocol is shown to be able to enhance the fairness and provide incentive mechanism in wireless P2P file sharing applications. It is also capable of increasing the energy efficiency.

Keywords: topology control, wireless networking, P2P systems, file sharing, energy efficiency, fairness, incentive, network protocols.

ARTICLE INFO

Article History

Received: 28th May 2016

Received in revised form :

28th May 2016

Accepted: 31st May 2016

Published online :

1st June 2016

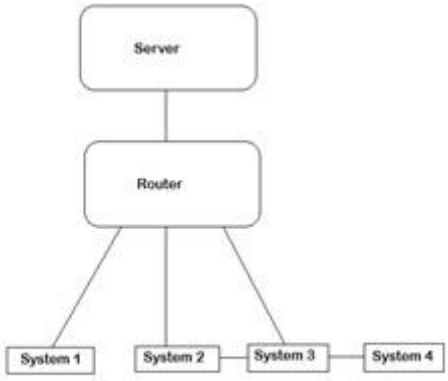
I. INTRODUCTION

Peer-to-peer (P2P) computing has been proposed as a promising technology that will reconstruct the architecture of distributed computing (or even that of the Internet). This is because it provides various resources (including computation, storage and bandwidth) at the edge of the Internet, with lower cost of ownership, and at the same time enjoys many desirable features Since, from last 10 years P2P computing technology has spurred increasing interests in both industrial and academic communities. As such, there are increasingly more applications being developed based on this paradigm. For example, digital content sharing, scientific computation collaborative groupware, instant messages and so on. However, there has not been much effort to study the architecture of P2P systems. As the architecture of a system is the cornerstone of high-level applications that are implemented upon it, an understanding of P2P architecture is crucial to realizing its full potential. Such a study is important because: (a) It helps researchers, developers, and users to better appreciate the relationships and differences between P2P and other distributed computing paradigms (e.g., client-server and grid computing). (b) It allows us to be conscious of the potential merits of P2P computing for newly emerging application demands, and to determine the most suitable architecture for them. (c) It enables us to determine the architectural factors that are critical to a P2P system's performance, scalability, reliability, and other features. Therefore, we dedicate this chapter to summarize and examine the architecture of P2P systems and some related issues.

We first present a taxonomy of P2P architectures based on existing systems that have been developed. On one extreme, some P2P systems are supported by centralized servers. On the other extreme, pure P2P systems are completely decentralized. Between these two extremes are hybrid systems where nodes are organized 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 supporting techniques for dynamic reorganization of peers that allow communities to be formed based on some common interests among the nodes. We will also examine how nodes that are relatively powerful can be exploited to shoulder more responsibilities.

II. PROPOSED SYSTEM



A hybrid unstructured P2P network allows the existence of infrastructure nodes, often referred to as super-peers (or super-nodes or overlay nodes). This creates a hierarchical overlay network that addresses the scaling problems on pure unstructured P2P networks such as Gnutella. A peer in such network can typically change roles over time. For example, a regular peer can become a super-peer that takes part in coordinating the P2P network structure. KaZaA (http://www.kazaa.com/), which is based on the Fast-Track (http://www.fasttrack.nu/) protocol, is an example of a hybrid unstructured P2P network. This network uses specially designated super peers with high bandwidth, disk space and processing power. When a peer joins the network, it is assigned to a super-peer. The peer then informs its super-peer about the content that it will share. The super-peer facilitates the search by maintaining a database that maps content to peers, and tracks only the content of its assigned peers. Similar to the centralized design, the super-peer plays the role of a directory server, although only to its assigned peers. The super-peers together create a structured overlay of super-peers, which makes search for content more efficient. A query in this network is routed to a super-peer. Then, as in the decentralized design, the query is flooded in the overlay super-peer network. The super-peer then responds to the peer that originated the query with a list of peers having the content. The hybrid networks are no longer dedicated to a single server, since the database is distributed among the super-peers. Moreover, the size of the database is relatively small, since each super-peer tracks only the contents of its assigned peers. However, the drawback of this approach is that it is considerably complicated, and requires non-trivial maintenance of the overlay network. Moreover, the fact that super- peers may have more responsibilities than ordinary peers can result in a bottleneck.

Peer-to-peer networks generally implement some form of virtual overlay network on top of the physical network topology, where the nodes in the overlay form a subset of the nodes in the physical network. Data is still exchanged directly over the underlying TCP/IP network, but at the application layer peers are able to communicate with each other directly, via the logical overlay links (each of which corresponds to a path through the underlying physical network). Overlays are used for indexing and peer discovery, and make the P2P system independent from the physical network topology. Based on how the nodes are linked to each other within the overlay network, and how resources are indexed and located, we can classify networks as unstructured or structured (or as a hybrid between the two).

Socket is an object that represents a low level access point to the Internet Protocol (IP) stack, it is used to send and receive data, thus it can be opened and closed, the data to be sent is always sent in block known as Packet.Packets must contain the IP address of both the source of the packets and the destination computer where the data is being sent, and optionally it may contain a Port number. A port number is between 1 and 65,535. A port is a communication channel or endpoints on which computers can communicate. It is always recommended that programs use port number higher than 1024 to avoid conflicts with other applications running on the system, because no two applications can use the same port. Packets containing port numbers can be sent using UDP (User Datagram Protocol) or TCP/IP (Transmission control protocol). UDP is easier to use than TCP because TCP is more complex and has longer latencies, but when the integrity of the data to be transferred is more important than performance, then TCP is preferred to UDP, and thus for our file sharing application TCP/IP will be used because it guarantees that our file does not become corrupt while being transferred and if during the transmission process a packet is loss, it is retransmitted thus making sure that our file integrity is maintained.

III. PACKAGES

ClientDeskTopSharing:

1. Jrdesktop
2. Jrdesktop.images
3. Jrdesktop.rmi.client
4. Jrdesktop.rmi.server
5. Jrdesktop.server
6. Jrdesktop.utilities
7. Jrdesktop.utilities.screenCaptureComp
8. Jrdesktop.viewer
9. Jrdesktop.viewer.FileMng
10. Org. Jrdesktop.layout
11. Org. Jrdesktop.swingworker

ServerDeskTopSharing:

1. Jrdesktop
2. Jrdesktop.images
3. Jrdesktop.rmi.client
4. Jrdesktop.rmi.server
5. Jrdesktop.server
6. Jrdesktop.utilities
7. Jrdesktop.utilities.screenCaptureComp
8. Jrdesktop.viewer
9. Jrdesktop.viewer.FileMng
10. Org. Jrdesktop. layout
11. Org. Jrdesktop.swingworker

Procedure:

1. On client click on configuration enter self IP address (pc address) and port no .and click on start button.

2. On server computer start net-beans and open project run main.java file code in neat-beans (6.8)

3. On server click on configuration enter default IP address as (127.0.0.1) and Port no same as specified on client comp.

4. Click on new connection Enter client IP address and port no. and click on start button.

5. Client desktop view on server side

6. You can perform any operation of client from Server.

IV. 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 "cross-layer" 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.

V. REFERENCES

[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 BabakEsfandiari,"A Distributed Wiki System Based on Peer-to-peer File Sharing Principle",

[3]DongmeiJiaWai Gen Yee Linh Thai Nguyen OphirFrieder,"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]SurendarChandra ,WilliamAcosta,"Using query transformation to improve Gnutella search performance",

[6]Cristiano Costa JussaraAlmeida,"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-KwongKwok,"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

