Robust Peer-to-Peer Architecture for File Sharing in Wireless Network

Supriya Jamale, Mohini Kedar, Asawari Kulkarni, Prajkta Jamdade, Ratan Deokar

Abstract— **P2P systems are used to communicate between two interested participants. As P2P network communicate directly between nodes it become difficult to maintain network performance as well as network traffic. Mobile networks add additional challenges such as heterogeneity of operating system, inherited resources, characteristics of wireless network.**

We implemented a robust Peer to Peer system where files are our prime resources. A user while doing work on any node can access the files of any other peer provided he is an authenticated user. A graphical user interface is made available at each peer for registering itself to the network, for searching and if found, displaying the contents of required file. So it is secured than traditional distributed system.

We present *Robust Peer to Peer*, a robust architecture to create mobile Peer to Peer networks and efficiently maintain the files content of nodes, file log information, network state. Robust Peer to Peer introduces a novel super-peer selection protocol based on an aggregate utility function that takes into account peers' capability and context. It also presents an scheme through which super-peers can delegate their responsibilities to more powerful and stable joining or existing peers. Our proposed system provide semi centralize approach helps to reduce network traffic results in increase in network performance.

Index Terms—**Peer-to-peer files sharing, peer-to-peer networks, P2P,selection of super peer, super-peer, mobile nodes,file sharing.**

I. Introduction

Peer-to-peer (P2P) systems are constructed to provide content sharing among interested participants (peers) in semi-centralized way. We can also define our user based topology for peer to peer network. Mobile environments pose additional challenges on P2P networks due to heterogeneity of nodes, built in limited content, dynamic context and wireless network characteristics. When it comes to P2P file sharing specifically, there are two models: centralized and decentralized. In the centralized p2p model, the central servers maintain directories of the files shared by the users of the system. The servers allow query of their database and provide results that allow a user doing a query to establish a direct connection with a user who is sharing a desired file.

In super-peer overlay infrastructures, the network topology is constructed in three layers. First layer contains server, which maintain directories for super peer. Second layer contains nodes called *super-peers* (or *super-nodes*) that have relatively higher capability and assume special responsibilities. The third layer contains all ordinary peers (remaining *peers*). Super-peers handle the communications inside their corresponding cells as well as exchange information with other super-peers. Query resolving (resource discovery) in super-peer architecture is much faster than any other P2P topology. However, super-peer selection is challenging due to the many factors that govern the selection decision which have direct impact on the super-peer performance.

The choice of the underlying network architecture has a great impact on the overall system performance. Super-peer networks take advantage of centralized schemes. It also introduces a reliability improvement scheme that reduces the network maintenance overhead, while improving the overall network reliability and stability. In addition, it reduces the overdue burden on files-constrained nodes by distributing loads evenly across the network.

P2P file sharing networks can be classified into four basic categories: the centralized, decentralized, hierarchical and ring P2P systems. These topologies may exist on their own it is usually the practice for distributed systems to have a more complex topology by combining several basic systems to create, what is known now as hybrid systems. This thus gives P2P systems two main key characteristics:

I. **Scalability**: There is no limitation for extension of size of system, e.g. the performance measure of the system should be somewhat constant regardless of number of nodes in the system.

II. **Reliability**: The failure of any given node will not affect the whole system (or maybe even any other nodes).

III. Allowing users of the network to schedule batch-jobs that are processed by the computers on the network during their idle time thereby decreasing the need for new computing files; and

IV **File Transfer**: Allow users to exchange data directly without storing files on a centralized server thereby avoiding the need to establish a centralized server and allowing two businesses to communicate with each other directly.

II. Related Work

Content sharing in P2P systems is done in ad-hoc fashion. Peers that request or offer access to files voluntarily participates in process and have the option to join and leave the network. Many research efforts are done in the super-peer selection algorithms and maintenance schemes of topology. Some researches introduced several modifications to the original design of Gnutella in order to accommodate node heterogeneity. Some proposed a double-layered P2P system, in which super-nodes are selected based on their mobility pattern in order to enhance the system stability and reliability. They also believe that the

ISSN: 2278 – 7798

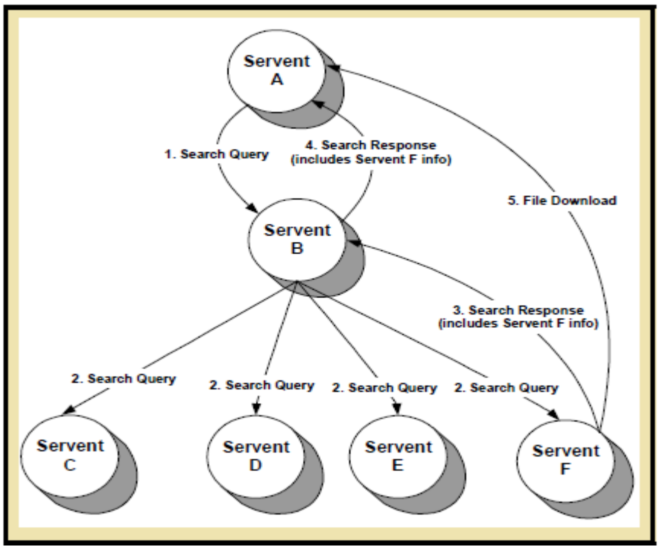
node energy level should be taken into consideration along with the mobility of nodes.

In contrast to these previous research efforts, Khalid et. Al [1] integrated many factors to efficiently select super peers, including the node's current mobility, immediate energy level, and connectivity degree. Current energy level assigns a higher priority to nodes with higher network connectivity. The connectivity degree aims to balance the node load (number of served peers) with its capacity and uniformly distribute the peer load across the network topology.

Gnutella protocol, a type of the peer-to-peer networking system, which provides decentralized file-sharing capabilities to its users. To share files on the Gnutella network, a user (node A1 for example) starts with a networked computer that runs one of the Gnutella clients. Since this node will work both as a server and a client, it is generally referred to as a (Gnutella) “servent”. Node A1 will then connect to another Gnutella-enabled networked computer (node B1 for example) and then A1 will announce existence to B1. Node B1 will in turn announce to all its neighboring nodes (nodes C1, D1, E1, and F1 for example) that A is alive. The user at this node can now query the contents of the data shared across the network.

Limitations of gnutella:

- More time require for query resolution.
- Increase network overhead traffic.
- It works as blind search for files.
- Search query aren't instantaneous because they have to bounce around.



Before proceeding it is necessary to identify some of the flaws of the Gnutella network with respect to the defined design goals. In the Gnutella protocol this is accomplished by using the servant identifier field in the description header for routing purposes, but it is absolutely not hard to track the IP of one servant by just using simple network sniffing tools. Another main point of attack is a malicious user. Since every servant routes packets of other network participants it is very easy for an attacker to harm the core of network. Nodes may easily become a serious bottleneck in the network.

It had also been observed that the shared data is not evenly distributed. So 71% of the ¯les shared were provided by 10% of the nodes.

III. OVERVIEW DESCRIPTION

We design robust peer to peer architecture for Files sharing among mobile devices by providing semi-centralized control over system with the help of efficient super peer selection. The design of RobP2P boils down to a three-fold objective:

1) Develop a robust and efficient super-peer selection protocol;

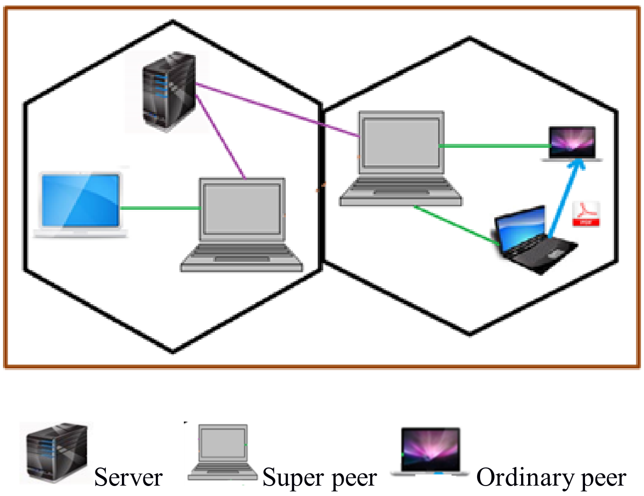
2) Fast query resolution;

3) reduce the overhead traffic of network topology maintenance;

4) Search queries are instantaneous for efficient searching.

5) Increase the reliability and stability of the network infrastructure through enabling peers to flexibly change their role.

We are considering one main server, single super peer for each cell and multiple ordinary peers. Every element has its specific work. This system design is mainly used to select super peer among all peers that leads to solve query efficiently.

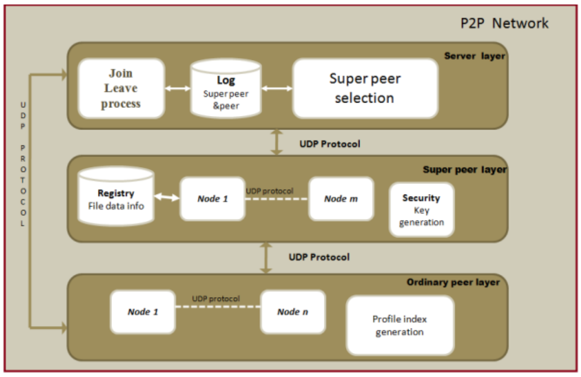


Super peer manages co-ordination among its connected nodes. It maintain log of files which are share by ordinary peers. When any peer wants to share any file it get ip address from super peer and peer- to –peer file sharing occur.

Files sharing can be done within nodes which are in own cell through super peer. But in case when file sharing has to be done within nodes belonging to different cells, it will be done through server. Link management and data updating works when any node leaves its cell or it destroys. This information needs to be update at server side.

IV. SYSTEM ARCHITECTURE

System Architecture is structure of components, their externally visible properties and relationship between the components. We are considering three layered architecture in which first layer is 'Server Layer' which maintains database of super peers. Selection of super peer is done in this layer. Second layer is 'Super Peer layer' which controls Ordinary peers within their own cells. Third layer is 'Ordinary peer layer' which are involved in the sharing of files. The content sharing among ordinary peers is authenticated by super peer.



System Architecture

The P2P network is divided into multiple regions. Each region represents a location-based group that contains peers that are physically located with the region boundaries. Algorithm 1 shows the group initialization procedure. Each group selects a super-peer that represents the group head, while the rest of the peers become ordinary peers. All peers calculate their profile index using the function in Equation (1) and participate to the super-peer selection following Algorithm 2. Once super-peers are selected, all advertisements and queries within groups are sent to respective super-peers. Super-peers collect and index the group information including active peers, advertised files, and offered services in order to manage the group communications and resolve queries addressed to the group. The super-peer is also responsible for maintaining the group state including selecting new super-peers, in case it moves away from the group's centered. The selection of super peer is done only when the performance of current super peer degrades and it send signal about its status.

Join/Leave

If any peer want to join the network it send request to server. The newly joining peer calculates its profile index based on the network *utility function* and. If the performance of current Super Peer degrades then it will inform to server and server will give command to perform selection of super peer algorithm. Then new super peer will select on basis of profile index. Super-peer, the new peer takes over the super-peer responsibility. Then, the current super-peer work as ordinary peer and sends an update message to the group declaring the new super-peer. This message updates the role of the current super-peer and provides the ID of the new super-peer.

- *Join Algorithm*

This algorithm is used when any new node want to enter into the network. First newly entered node is connected to server then server peer information log get updated and for next file transfer purpose super peer ip is given.

Input: nnew: new node to join Gi,

Output: null

Join (Gi)

1. nnew.profile= calculateEqu1(1)
2. Update server_peer-info log.
3. Get Super-peer ip.

- *Leave algorithm*

This algorithm is used when any node leave network. If that node is ordinary peer then all records stored in super peers database get deleted and database gets updated. But if that node is super peer then it needs to more stuff that is super peer has database of all ordinary peers within that cell. If that peer leaves the node firstly it had to broadcast message so that server can choose new super peer and also transfer all database to new super peer.

Input: ***nj***

Output: nnew. Leader
Leave (Gi)

1. Update sever_peer-info log.
2. Update super_peer-fileinfo log.
3. If(super_peer== ***nj***)
   3.1 Choose new super peer.
   3.2 Inform to all cell member.
   3.3 Transfer super peer log
   3.4 Update server_super-peer log.

Super peer selection

Super peer is nothing but an ordinary peer which manages control over a cell in a network. Main functionality of super peer is link management among peers. Selection of super peer is done by using super peer selection algorithm. Selecting super-peers in P2P systems is always challenging. A super-peer must be capable to improve the overall performance of P2P networks.

Battery - super-peers must possess sufficient resources to handle the group communications and resolve queries with reasonable delay.

ISSN: 2278 – 7798

Mobility - peers with low mobility profiles must be given higher preferences to avoid frequent super peer selections.

Network connectivity - Super peer must connect more devices within cell and provide sufficient connectivity to each node.

- *Super peer selection algorithm*

To measure whether a peer nj is a candidate to assume super-peer responsibilities in a group Gi, we define the peer profile using the utility function in Equation (1). In this equation, b is the current battery power level on nj, Emax denotes the maximum energy level that any peer belongs to Gi might have, m is the current mobility pattern of nj, Mmax is the maximum mobility nj can reach, ut is the normalized mean uptime of nj, which denotes how stable the peer is, bandwidth which require, NC represents the network connectivity, i.e. how many peers in Gi can reach nj, w1 w4 are weights that represent the factor importance, where 5k =1 wk= 1. In this utility function, we reverse the peer mobility, since peers with low mobility pattern are of higher preferences. The peer profile ranges from 0 to 1. The higher the profile value, the more possibility a peer could be selected as a super-peer. Each peer in Gi calculates and shares its profile with other peers. The peer with the highest profile declares itself the super-peer serving Gi.

Input: *nj:new joining node*

Output: *currentleader*

1. nj.profile = calculateprofileindex(nj)

2. nk= find current super peer(nj.cell))

2.1 update nk.profile

2.2 currentleader=nk

3. if nk.profile < threshold && nk.profile < nj.profile

3.1 currentleader ← nj

3.2 Broadcast msg_change_sp in all cell member

4. end if

5. end

- *Profile Index equation*

$$n_j.profile = \frac{1}{4}\left(w_1.\frac{n_j.b}{b_{max}(G_i)} + w_2.\frac{M_{max} - n_j.m}{M_{max}} + w_3\frac{n_j.BW}{BW_{max}(G_i)} + w_4\frac{n_j.NC}{|N_i|}\right)$$

Where,

nj = node for which we calculating profile index.

wi = weight assigned.

b = battery of node j.

bmax = Maximum battery level in group.

m = mobility of node j.

Mmax = Maximum mobility in group.

BW = Bandwidth of node j.

BWmax = Maximum Bandwidth in group.

NC = connectivity degree in group.

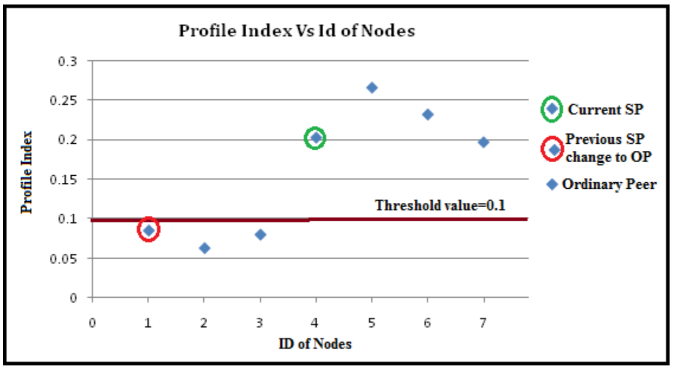
|***Ni*** | = Total node in network.

V. Constructing the Robust P2P Architecture for Files Sharing

We have implemented this topology using Java (Socket programming) and WiFi as communication medium. For the experiment we consider 2 cells each cell having 10 ordinary peers including super peer. The experimental result considering various factors is as shown as below

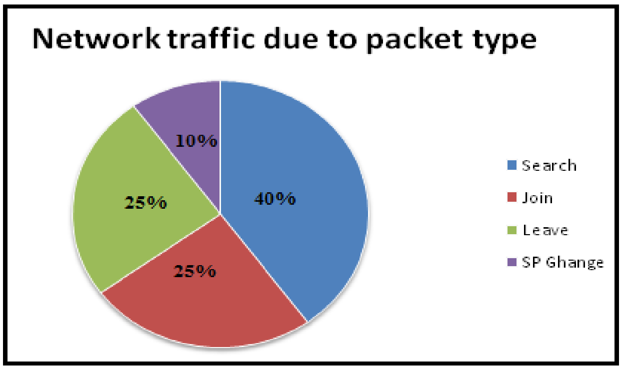
- Selection of Super peer:

The first experiment to investigate proper super peer selection when first peer enter in cell it automatically become super peer. For this experiment we consider threshold value for super peer 0.1 considering all parameter. Here node having ID 1 possess profile index 0.9 but as it is first node to enter in cell network it become super peer, when ID 2,3 come in network profile index generated but it is less than threshold value super peer does not change. Consider node ID 4 come with profile index 0.2 which is greater than threshold and current super peer so it become super peer. Now super peer does not change till profile index of current super peer came become threshold value.



- Network traffic:

To improve the network performance it is necessary to reduce network traffic in our system network traffic due to various packet is as shown in figure if we compare the result with current system network traffic occure due to searching or transferring a files is very much less.



Existing system (traffic due to search query)=41%(query fail rate)+3%(query hit rate)=44%

In proposed system query hit rate is 40%.

However, RobP2P is capable of maintaining a stable network state our super-peer selection function takes into account the various factors that accommodate the inherent dynamics of P2P mobile networks, which by itself makes the super-peer selection efficient.
The role-changing scheme, that we introduced to handle the network churn and the dynamic change in the node context, enables peers to request changing their role

VI. Conclusion

This paper presents *RobP2P*, a robust P2P architecture for mobile nodes that enables efficient file sharing.RobP2P introduces an aggregate utility function that determines whether a peer is a candidate to assume super-peer responsibility. This utility function takes into account both the mobile node constraints and mobile network dynamicity. RobP2P also introduces a scheme that enables peers to call for changing their role based on a significant change in their current profile index and threshold frequency. This scheme renders the P2P network topology, constructed with RobP2P, more stable. It also significantly reduces the network traffic while maintaining a high level of reliability. Simulation results show that RobP2P outperforms other P2P architectures. We plan to continue improving RobP2P by introducing a middle layer that contains *relay peers*, whose profiles are close to the selected super-peer. Relay peers are intended to extend the structure of the overlay network, while maintaining same level of reliability.

A robust P2P architecture enables efficient resource sharing, which is based on super peer selection.
Correct Super peer selection leads to:
– Efficient resource sharing
– Increase in query hit rate
– Reduces network traffic overhead

It helps us to improve overall network performance.

References

[1] Khalid Elgazzar, Walid Ibrahim, Sharief Oteafy, and Hossam S. Hassanein “RobP2P: A Robust Architecture for Resource Sharing in Mobile Peer-to-Peer Networks”

[2] Maria Papadopouli “Resource Sharing in Mobile Wireless Networks (Columbia University Journal 2002).”

[3] Claudia Canali, Member, IEEE Computer Society, Maria Elena Renda, PaoloSanti, and Simone Burresi “Enabling Efficient Peer-to-Peer Resource sharing in wireless Mesh Networks.”

[4] Lee W. McKnight, James Howison (Syracus University), Scott Brandner (Harvard University) “Distributed Resource Sharing by Mobile, Nomadic, and Fixed Devices (IEEE 2004)”

[5] Xing Li, Xiaoqin Zhang, Qiyun Han “A Construction Approach of Small World Network in Large-scale Distributed system.”

Supriya Jamale Student at P.C.C.O.E, Nigdi, Pune
Mohini Kedar *Student at P.C.C.O.E, Nigdi, Pune*
Asawari Kulkarni *Student at P.C.C.O.E, Nigdi, Pune*
Prajkta Jamdade *Student at P.C.C.O.E, Nigdi, Pune*
Ratan Deokar *Professor at P.C.C.O.E, Nigdi, Pune*

ISSN: 2278 – 7798