Advanced Computer Networks

Project Report

CSN-505

Computer Science and Engineering



INDIAN INSTITUTE OF TECHNOLOGY ROORKEE

ROORKEE – 247667

Submitted to: Submitted by:

Dr. P Sateesh Kumar Alamjot Singh (18535002)

Assistant Professor Anjana KK (18535005)

Dept. of CSE Ashish Mahawal (18535008)

IIT Roorkee L Rahul Kiran (18535015)

Monica Jangpangi (18535018)

**Contents**

[Introduction 3](#_gjdgxs)

[The peer-to-peer paradigm](#_30j0zll) 4

[Definitions](#_1fob9te) 4

[Definitions based on locality and connectivity of nodes 4](#_3znysh7)

[Definitions based on decentralization and self-organization 4](#_2et92p0)

[Gnutella](#_tyjcwt) 5

[The Protocol](#_3dy6vkm) 5

[Performance](#_1t3h5sf) 5

[Scalability 5](#_4d34og8)

[Implementation](#_2s8eyo1) 6

[Elements of the Local P2P node](#_17dp8vu) 6

Working 7

Future Aspect: Evolution to Web-Based Nodes10

# 

# Introduction

The main understanding about peer-to-peer (p2p) systems are the ordinary PC-users that have something valuable to share. It may be network capacity, CPU-cycles or storing capacity. The p2p networks are not new a buzzword, even if the actual term peer-to-peer is of a fairly recent date.These p2p nodes could locate other nodes, share files, and take part in distributed computing.

Communication is done with the help of five message types: Ping, Pong, Query, QueryHit, Push. ​

Ping - An empty message sent by a client requesting an 0x01 from everyone on the network.

Pong- Sent in response to a 0x00, this message contains the host ip and port.

Query- This is a search message and contains the query string.

QueryHit- Response to query message if query is matched.

Push- Message type for pushing file.

Each node in a Peer-to-Peer network acts as both a client and server where the nodes are called “Servants”.​

​

# 

# The peer-to-peer paradigm

## Definitions

The p2p concept has been delimited against client/server concepts. Although there is an abundance of p2p definitions, the majority can be distilled into three concepts: decentralization, self-organization and resource-sharing. There is also a line of thought that emphasizes the locality and connectivity of the nodes participating in the investigated network. This results in two sets of p2p criteria for classifying protocols and applications. One emphasizes locality and connectivity of the nodes which does not consider the direct communication between nodes to be a constituting feature of the concept of p2p. The other gives prominence to decentralization, self-organization and direct communication between nodes.

### Definitions based on locality and connectivity of nodes

An application meets the p2p criteria if it allows for variable connectivity and temporary network addresses, and if it gives the nodes at the edges of the network significant autonomy.

The novelty does not lie in the direct communication between nodes. The breakthrough consists of the chances to utilize previously unused resources offered by devices that bypass the Domain Name System (DNS).

### Definitions based on decentralization and self-organization

In the present context, decentralization refers to topological properties. A decentralized network develops as a consequence of local decisions made by the nodes of the network on the basis of local information. This means that these networks have routing mechanisms of their own. Gnutella and Freenet are examples of networks that are designed to function in a decentralized way. In practice, many p2p network architectures in use represent a hybrid version which includes both decentralized and centralized features. Self-organizing nodes frequently have the ability to build an application level network. The nodes cooperate for the benefit of the whole network, and the result is an application that could not be provided by a single node. The management functions of the nodes are indispensable for the self-organization of the nodes.

# Gnutella

## The Protocol

All nodes have routing function. ping and query messages are used to register and search the nodes in the network respectively replies with pong or query hit message respectively through the same nodes those forwarded from source node. The expression “search horizon” is set of hosts reachable by a search request, Gnutella node standard horizon is seven hops.

In order to avoid traffic in network, modern Gnutella protocol forwards the ping messages to the nodes that do not have addresses in pong cache.

The Gnutella protocol version 0.6 suggests that if ping messages have pong data then they can wait for the peers to announce themselves instead discovering neighbours. Ping messages are sent on periodical basis.

## Performance

Ping and pong messages require a very large percentage of bandwidth in the network.

Number of connections of each node and number of hops a message can propagate through the network also affects the network performance.

## Scalability

Gnutella supports few hundred to a few thousand users only in order to increase scalability Gnutella is modified into party hierarchical networks by using super peers.

The next version of Gnutella protocol is node can operate in three different modes, ie, peer, client and server.

# 

# 

# Implementation

## Elements of the Local P2P node

A P2P node consists of 3 structural entities:

1. The Node
2. The Command-line user interface
3. Data structures to store information about files and peers.

**The Command-line user interface**: Command-line user interface is implemented using various python libraries and has the same local host as the node. Command-line user interface communicates with the node using queries to which the node sends query hit messages.

**The Node**: The node can receive messages from both the network and the user interface. The same protocol is used in the user interface and node. Node act both as client, server and route various messages to another node.

**Data Structures**: The two data structures used in a node are the file repository, and the peer table.

The ***file repository*** is implemented using python dictionaries which represent various files present in node. For every matching string in a query for a file, the node uses local function calls for obtaining the file’s name, size, etc.

The ***peer table*** contains address of the peers to which the node communicates directly.

## 

## 

## 

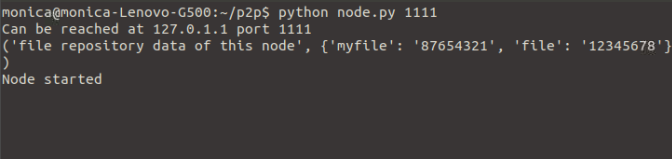
## 

## Working:

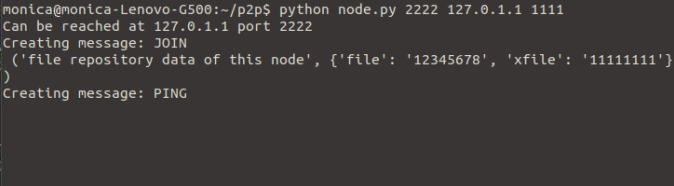
Our node is implemented using sockets in python. A socket is one endpoint of a two-way communication link between two programs running on the network. A socket is bound to a port number.

We start our gnutella network by starting a primary node, we will call it node A. We then start user interface and another node which we will call node B. We then connect both user interface and node B to node A. We then send queries through user interface to our node A.

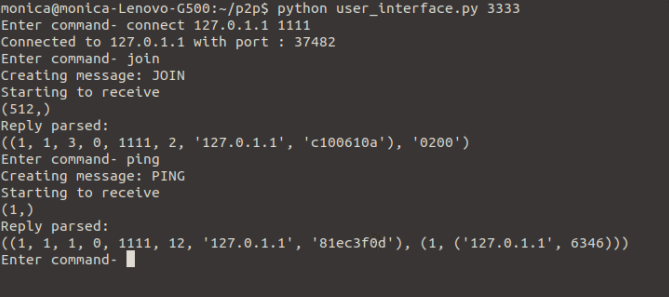
We started first node, node A



We started node B and connected it with node A



We started command line user interface and connected it with our node A



Now we will send queries, below is file repository table for nodes:

Node A: myfile: 87654321, file: 12345678

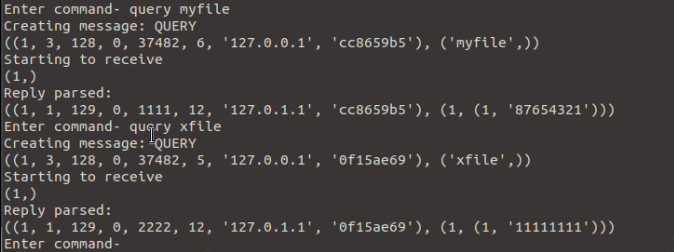
Node B: xfile: 11111111, file: 12345678

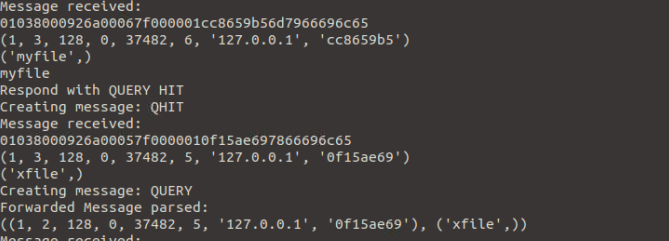
We will first send query with parameter “myfile”, Since this node A has this file so in response it will send QueryHit with result “12345678 ”

We will then send query with parameter “xfile”, Since this node A doesn’t has this file so it will forward request to other node in network. Node B will receive it, since it has this file so in response it will send QueryHit with result “11111111”

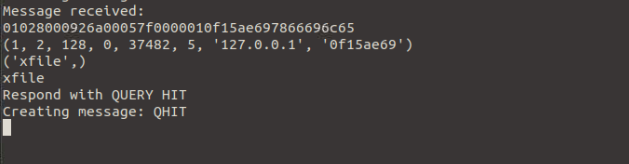
Also we will send an random query to check, since any node doesn’t has this file so user interface will receive no response.

Responses in Command line user interface



Responses in Node A

Responses in Node B



## Future Aspect: Evolution to Web-Based Nodes

The Gnutella protocol uses a mixture of ASCII and binary code. ASCII encoding is used for handshake before opening every session, sending queries and to download files using the HTTP protocol which is ASCII encoded. The packets send and received using Gnutella consists mostly of binary codes.

Since the user interface can connect to any node if it knows the address and port number of the node interface, the user interface can be used to send requests and download files without uploading anything. This is similar to the working of a web-based system.

They will develop a web-based node, which uses HTTP as the communication protocol between P2P nodes.

## P2P Communication using HTTP

### Web-Based P2P Node

The web-based P2P node consists of 3 structural entities:

1. The Node
2. User Interface
3. Data structures to store information about files and peers.

**The Node**: The node will use same protocol for communication between the user, other node and the network. The node also contains a GUI, an HTTP server and an HTTP client.

**User Interface**: Instead of command line, we use HTML to implement the user interface. The user sends queries to the web server through the web server. The router handles these queries when they enter the server.

**Data Structures**: Peer table are same as that of the data structures used in local P2P nodes. The Repository now contains information on links to resources.

### 

### P2P Messages Contained in HTTP Methods

In our design we use two P2P request messages:

1. PING

The PING message along with information as payload is included as part of the HTTP POST method. The response to PING messages are in plain text.

1. QUERY

The QUERY messages are included as dictionary in the POST method. The response to QUERY messages are in plain text.