

# CS550 Design Document (PA#3)

---

Cyril Trosset A20316209

Section 1

Uzair Ahmed A20303372

CS 550

## Introduction

In this assignment, we have added consistency mechanisms to the Gnutella style P2P file sharing system. Here each peer acts as both a client and a server. The objective of this project is to ensure that all copies of a file in the P2P system are consistent with one another. There is a master copy for each file in the system. The master copy is typically stored at the peer where the object was initially created called the origin server who is the owner of the file and only the copy present with the origin server can be modified.

## Requirements

- 1: Implement push based approach by adding a new message type and invalidation message in the system. The server will broadcast an invalidation message out whenever there's a modification to a file. From the Server's perspective, there should be at least two directories for each peer, one for the files owned by the peer and other for the files downloaded from others. Only the file's owner can change the files. From the client's perspective, the following information for each downloaded file must be stored. version number of the file, the origin server id for the file, and consistency state of the file, where the consistency state is (valid | invalid | TTR expired).
- 2: Implement pull-based approach, where a static TTR based approach. From the server's perspective, whenever a download request comes in, the server must send the following info along with the file: the origin server ID for the file, TTR, and last-modified-time of the file. If the file is downloaded from a peer other than the origin server, then that peer simply uses the information stored with that copy and passes it to the new downloader. From the client perspective, a peer will periodically check its local store and poll the server for those copies whose TTR expired. And the server in reply will send a new TTR if the file is not modified since. Or a negative reply if the file has been modified. Otherwise the client updates the file's TTR and proceeds. The client side should provide users an option to "refresh" an outdated file, which downloads a new copy of the file.
- 3: Implement both the push and pull based approach as described above but have configuration parameters that can turn on and off these features.

## Design

The network is initialized using a config file read by each peer at startup. Consistency is achieved either using Push or Pull.

In push approach, whenever the master copy is modified, the server broadcasts an “Invalidate” message for the file. This invalidate message propagates exactly like a "query" message. Upon receiving an "Invalidate" message, each peer checks if it has a copy of the object. Further, it propagates the “Invalidate” message to all its neighboring peers. In this manner, the invalidate message propagates through the system and invalidates all cached versions of the object. Here the origin server does not maintain any state about which peer is caching an object, - an invalidate message is simply broadcast through the system and reaches all peers regardless of whether they have the object or not.

In the pull approach, it is the responsibility of each peer to poll the origin server to see if a cached object is valid. The poll message contains the version number of the cached object. The origin server, upon receiving this message, will check the version number of the master copy and respond accordingly. If the master copy is newer, the origin server sends a "file out of date" message back and the peer then discards its cached copy and notifies the user (by printing an appropriate message on the screen). The effectiveness of the pull approach depends on when and how frequently a peer polls the origin server. In general, the more frequent the polls, the better are the consistency guarantees that can be provided. If the master copy is modified between two successive polls, then the peer is left with a stale copy until its next poll. When downloading an object, the origin server attaches a TTR value with the object to indicate the next time the peer should poll the server (e.g., TTR=5min or TTR=30min). The peer simply polls the origin server when the TTR expires.

The queries used in this assignment are of the type

Type: UID: TTL: Fname: IP :Port: versionNumber

- Type refers to the type of the query.
- UID: is the unique identifier that identifies individual query, which informs the client if it had already dealt with the query.
- TTL: is the time to live, which is associated with each query which controls the time the query lives in the network.
- Fname: denotes the name of the file that needs to be queried.
- IP: refers to the IP of the host
- port: is the port of the client
- versionNumber: refers to the version number of the file

The program gives three options, to search for a file, download a file and to Invalidate a file. When the search option is picked, then the program returns all the file that the peer contains. When the download option is selected, the peer checks if the version number of the file it contains is the same as the original copy of the file. When the Invalidate file option is opted, then the version number of the file is incremented and based on the approach, appropriate action is taken. The ways by which the peer checks if it contains the latest copy of the file and the ways the invalidate message is sent is explained below.s

When the file is changed, the server sends an invalidation message for the file to all the peers in the push method. Upon receiving the invalidate message, the peer looks if it has the copy and if the copy is found, it discards it. And after discarding the cached copy of the file, it forwards the query to its neighboring peers.

In the pull based approach, the peer has to poll the origin server to see if the object is valid. Every peer sends By making use of the TTR( time to refresh), the peer polls the server at end of every TTR. Once the server gets this poll message that contains the version number, it check if the version number of the file it contains is the same as the one sent in the message. If the version number of the file contained is newer, then the server sends a file out of date message back to the peer. The peer now discards the copy of the file.

## **Enhancements**

Future enhancements can be the following:

1. Dynamic initialization of the network can be a future enhancement for this assignment.
2. Dynamic network modification, which would allow a network to add and remove nodes dynamically.
3. Another enhancement can be the automatic discovery of the network, where a node automatically discovers the other nodes present in the network.
4. File modification can be done by other peers as well, who aren't the owners of the file.