Programming Assignment 2
Maintaining file consistency in your Gnutella-style P2P system
CS550- Advanced Operating System
Harsh Singh (A20398109)

Verification:

The system is verified by designing (most of) the test cases using equivalence partition testing.

1. Verifying basic Gnutella P2P file sharing system.
   Initially debug codes were inserted to check the functionality of basic abstractions. When those were verified, upper abstractions were tested. These tests follow equivalence partition method.

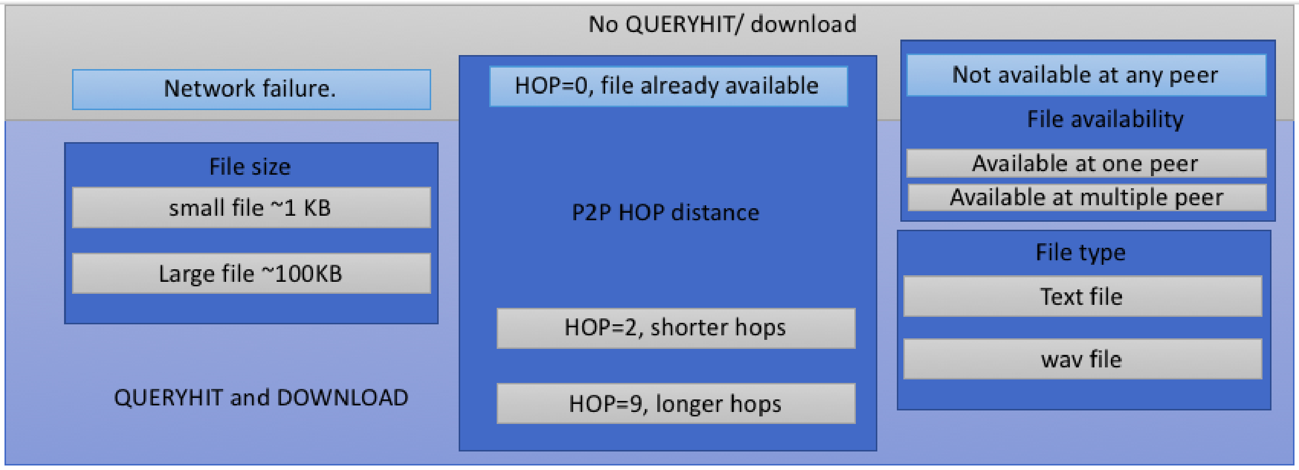

Figure 1 equivalence partition of Gnutella

   Figure 1 shows various scenarios of queried file. File didn't download or QUERYHIT was not received when file was either not available or file was available only at the location of initial request or network failure. In all other cases, file was downloaded.

   Such test cases validate Query Forward, QUERYHIT, file upload and file download. It also verifies handling of various error conditions like peer unavailability, file unavailability, connection error etc.

2. Verifying PULL mechanism.
   Similar to Gnutella testing, debug codes were added to check the functionality of methods. To avoid runtime errors, test cases were designed using equivalence partition.

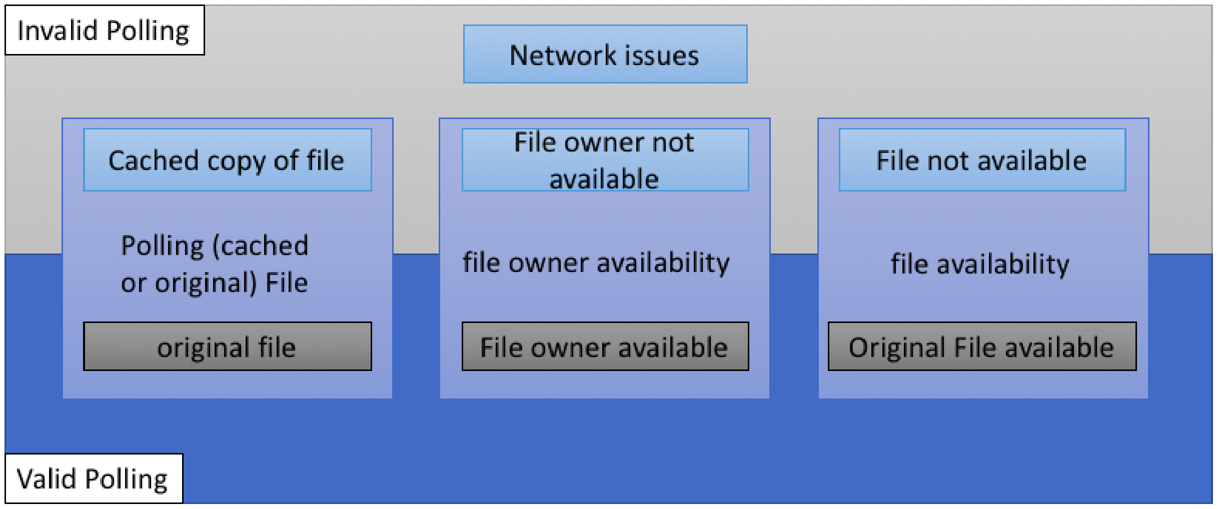

Figure 2 equivalence partition of PULL mechanism

Polling of a downloaded file from a peer to a peer should only work when file is original, available and no network issues. Otherwise, polling should happen later.
Also, try and catch error handlers are applied to minimize failure (due to network etc.)

3. Verifying PUSH mechanism.
   Similar to Gnutella testing, debug codes were added to check the functionality of methods. To avoid runtime errors, test cases were designed using equivalence partition.

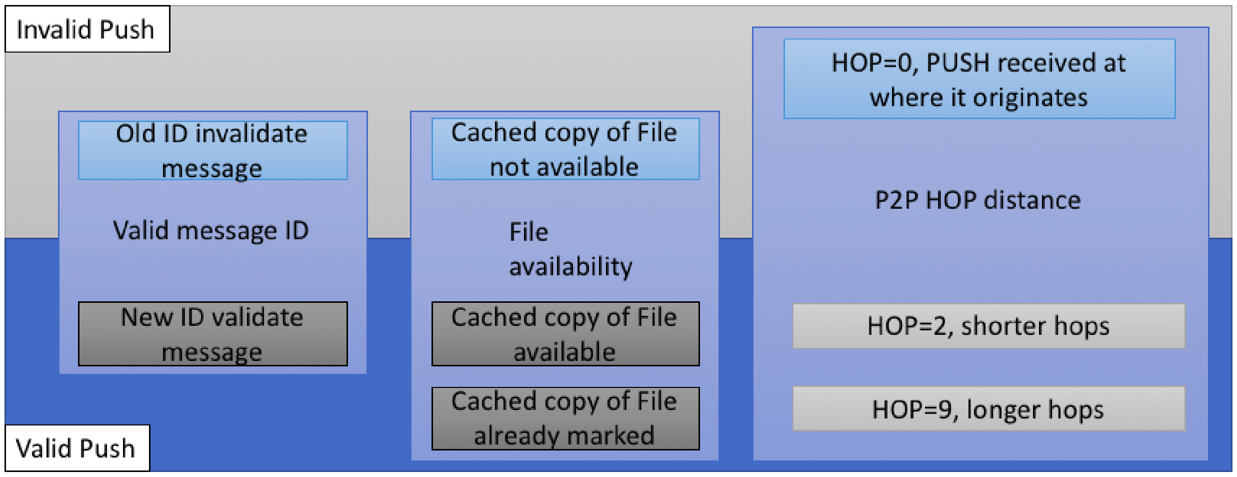
Invalid Push
Old ID invalidate message
Valid message ID
New ID validate message
Cached copy of File not available
File availability
Cached copy of File available
Cached copy of File already marked
HOP=0, PUSH received at where it originates
P2P HOP distance
HOP=2, shorter hops
HOP=9, longer hops
Valid Push

Figure 3 equivalence partition of PUSH mechanism

PUSH (invalidation message) is considered invalid if ID already received or cached copy is not available or cached copy already invalidated or invalidation message origin is same as received peer. Otherwise PUSH is considered valid.
Also, try and catch error handlers are applied to minimize failure (due to network etc.)

4. Others.
   All of the communication, file read and write, file last modified read etc. are accompanied by try and catch error handlers, minimizing failures.

All mentioned test cases were tested and passed.