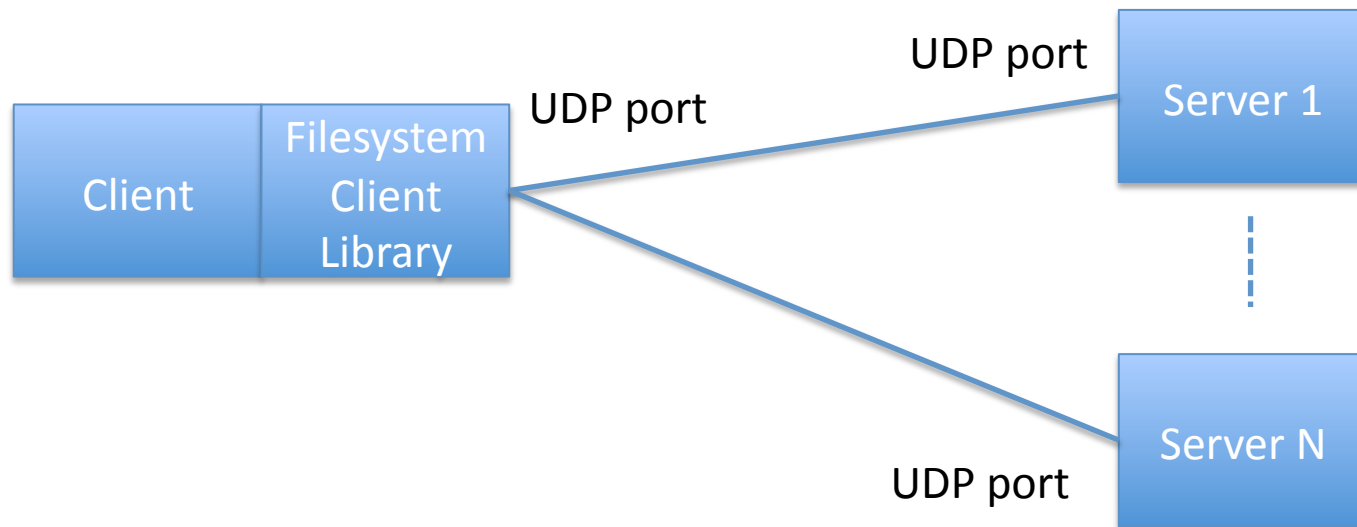


# CS244B Project 2: Replicated Filesystem

Review Session, 5/14 3:15pm B01

<http://www.stanford.edu/class/cs244b/replFs.html>

# Filesystem Overview



- Transactional semantics: commit file updates atomically, or abort
- Replicated: files stored on all available servers
- Reliable: handle drops, failed servers, corrupted packets

# Required Client API

```
intInitReplFs(int portNum, int packetLoss);  
intAddServer(char *id);  
intOpenFile(char *name);  
intWriteBlock(int fd, void *buffer,  
    intbyteOffset, intblockSize);  
intCommit(int fd);  
intAbort(int fd); (Use int instead of void)  
intCloseFile(int fd);
```

# 2 Phase Commit

## Commit request phase (or voting phase)

1. Client sends a query to commit message to all servers
2. Servers execute the transaction up to the commit request
3. Each server replies with a yes (commit) or no (abort)

## Commit phase (or Completion phase)

**Success** - Client received a yes vote from all servers:

1. Client sends a commit message to all servers
2. Each server completes the operation
3. Each server sends an ack to the client
4. The client completes the transaction when acks have been received

**Failure** - Client received an abort message during the commit-request phase:

1. The client sends a rollback message to all the servers
2. Each server undoes the transaction
3. Each server sends an ack to the client
4. The client undoes the transaction when all acks have been received

More details at [http://en.wikipedia.org/wiki/Two-phase\\_commit\\_protocol](http://en.wikipedia.org/wiki/Two-phase_commit_protocol)

# What to Turn In

- Implementation (70%)
  - Server app
  - Client library
- Report (pdf, 30%)
  - Protocol Spec
  - Evaluation
  - Future Directions