

SevenDeadlySYN - Design Doc

Matthew McFarland, Jacob Weiss, Adam Grounds, Dave Harmon

Abstract: Here is a high level description of the project.

Contents

1	Client	2
1.1	Subsection title!	2
2	Tracker	2
2.1	Subsection title!	2
3	Network	2
3.1	Client <-> Network Communications	2
3.1.1	Client <- Tracker (Incoming)	2
3.1.2	Client -> Tracker (Outgoing)	3
3.1.3	Client <- Client (Incoming)	3
3.1.4	Client -> Client (Outgoing)	3
3.2	Client Network Operation	3
3.3	When a Client Connection is Made	4
3.4	Tracker <-> Network Communications	4
3.5	Tracker <- Client (Incoming)	4
3.6	Tracker -> Client (Outgoing)	4
3.7	Tracker Network Operation	4
4	Data Structures	5
4.1	Subsection title!	5

1 Client

Write about this section here!

1.1 Subsection title!

This is a subsection!

2 Tracker

Write about this section here!

2.1 Subsection title!

This is a subsection!

3 Network

The client and tracker logic threads are separated from all actual network functions by asynchronous queues that the tracker and client logic threads can use like APIs. These asynchronous queues deliver information received by the network thread to the logic thread and deliver network requests to the network thread to disseminate. By separating the logic from the actual network calls, we can preserve a simple two thread but asynchronous communication model between the logic and the network. The advantage of these communication queues is that each thread can focus on reacting to incoming data and acting appropriately. The single thread model also prevents Heisenbugs from developing.

3.1 Client <-> Network Communications

3.1.1 Client <- Tracker (Incoming)

- Receive a master file system from the tracker
- Receive a master file table from the tracker

- Receive an update to the file system from the tracker (peer changed something)
- Receive a new peer added message / receive a peer deleted message
- Receive a peer acquisition update message

3.1.2 Client -> Tracker (Outgoing)

- Send a chunk acquisition update
- Send a request for master file system and file table
- Send locally updated files

3.1.3 Client <- Client (Incoming)

- Receive a chunk request
- Receive a chunk (request fulfilled or rejected)

3.1.4 Client -> Client (Outgoing)

- Send a chunk request
- Respond to chunk request by sending chunk or rejection response

3.2 Client Network Operation

The network thread continually scans its set of open socket descriptions with a `select()` call. After a timeout period, if no data is received on any of the sockets, then all of the queues from the logic thread are checked. (If there is data on the open sockets, then that data is processed before moving into the communication queues.)

The set of open socket connection will include the following connections. If the network thread is servicing a client, the network will always maintain a connection to the tracker. The network will also leave a listening socket open for new peer connections. If there are any existing client connections, those sockets will also be watched.

The client network thread also automatically sends a heartbeat message to the tracker every 6 seconds to assert its "alive" status.

3.3 When a Client Connection is Made

When the client logic makes a chunk request, the logic must specify a peer to request a chunk from. The network thread inspects its record of that peer, and if no connection exists, a connection is opened. A record of the number of requests made on a connection is kept. For every response, that record of requests is decremented until the outstanding requests reach zero, at which point, the originator of the requests will close the client to client connection.

3.4 Tracker <-> Network Communications

3.5 Tracker <- Client (Incoming)

- Receive file system update
- Receive request for master file system and master file table
- Receive a file acquisition update from a client
- Receive notice that a client has joined or left a network

3.6 Tracker -> Client (Outgoing)

- Distribute a file acquisition update of a client
- Distribute a file system update to clients
- Distribute information about a new peer or lost peer
- Send master file system and master file table

3.7 Tracker Network Operation

The tracker thread scans its set of open socket connections for new data to process and then checks each of the request queues from the tracker logic to the network thread. The tracker keeps an open socket devoted to listening for new client as well as open connection to each existing client in the network.

The tracker network thread keeps the master peer list of clients. This master peer list contains information about each client including: id (the identifier given to the logic side of things), the ip address, the time last alive, and the socket. (The tracker network thread periodically scans all of the open client connections and terminates any client that hasn't sent a heartbeat within a 6 second time frame.)

4 Data Structures

Write about this section here!

4.1 Subsection title!

This is a subsection!