

# Project TiniTorrent: Phase Two

Javed I. Khan  
Spring 2010, Deadline: April 15, 2008  
ST: 6/759995 Foundations of Peer-to-Peer Computing  
Department of Computer Science, Kent State University  
[javed@kent.edu](mailto:javed@kent.edu)

## 1 Assignment-2.1: MiniTorrent with Keep-Alive: (200+200+600=1000 points)

Problem: In this step you will implement two of the features deferred in earlier design. (a) Allow each peer to send and check periodic keep-alive messages via the connection. Unless messages are received in designated interval peer should close an inactive connection.

Documentation: none.

Experiment: none.

## 2 Assignment-2.2: MiniTorrent with Strategy: (200+200+600=1000 points)

Modify the TiniTorrent so that it can employ *optimistic choking and unchoking* strategy described in class note. You connect to all peers supplied by the tracker. Initialize the game in choked & uninterested state. Then based on who have the pieces, send interest message. Based on upload/ download rates send unchoked messages to selective peers as per strategy. Also include print statement in your peer so it can create a log/trace of all the incoming and outgoing messages with timestamp.

Documentation: (a) State the strategy implemented. (b) Explain the strategy program with flowcharts, and explain how each of the rules is supported.

Experiment: Let the odd/even peers exchange the test file. Trace the incoming and outgoing messages in both. Explain the trace. Give a count of each type of messages and network load due to each.

## 3 Assignment-2.3: Utilities for MiniTorrent: (200+200+600=1000 points)

Problem: In this step you will develop few utility systems. (a) Build a parsers program that can decode a standard torrent file. (b) Build a routine that can send HTTP GET message to a standard tracker and decode standard tracker reply. (Optional: (c) Build a mini personal tracker. (d) Build a program which crates a Torrent file.).

Documentation: none

Experiment: none.

## 4 Assignment-2.3: MiniTorrent: (200+200+600=1000 points)

Problem: Trace all the messages that a standard BitTorrent peer will use to download a file. Modify TiniTorrent messaging to reflect these messages. Also include print statement in your peer so it can create a log/trace of the incoming and outgoing messages with time.

Documentation: none.

Experiment: Let your peer join a real internet swarm and download a file. Trace all the messages that your peer received. Explain the trace.

## **5 Assignment-2.3: Creative MiniTorrent: (200+200+600=1000 points)**

Step-7: In this challenge final step you can implement an advanced feature into the Torrent. Discuss with me if you have any idea. Deploy and experiment on PlanetLab, and do comparative performance study with Standard TiniTorrent.