

# 15-440: Recitation 7

## Fault Tolerant Request-Reply Communication with Maybe, At-least-once and At-most-once Call Semantics

**Date:** October 20, 2011

### I- Intended Learning Outcomes (ILOs):

The ILOs of this recitation are two-fold:

- 1- Apply multithreaded client-server socket programming.
- 2- Code for a reliable communication using different request-reply call semantics.

### II- Objectives:

In this lab, students will:

- Create a multithreaded socket server using Java socket programming.
- Simulate two scenarios where a client cannot locate a server and a server filters duplicate messages from a single client for the same request.
- Implement maybe, at-least-once and at-most-once request-reply call semantics using a time-based mechanism, retransmissions, duplicate filtering and history.
- Assure that the server never leaves residual threads.

### **III- Problem:**

Develop a client-server application using Java socket programming. The client attempts initially to locate the server but fails. Apply maybe call semantic where the client gives up immediately and reports failure to locate the server. Also apply at-least-once call semantic where the client retransmits its request until either the server replies or the client concludes that the server is most likely down. Lastly, apply at-most-once semantic where the server is made capable of filtering duplicate messages from the same client for the same request. When implementing at-least-once and at-most-once call semantics, have the client start a timer when sending a request. If the timer expires before a reply comes back from the server, have the client act according to the adopted call semantic.