# CS 6378: Advanced Operating Systems
# Programming Project 3

Instructor: Ravi Prakash

Assigned on: November 20, 2015
Due on: December 9, 2015

   You are required to work on this project in groups of two. Sharing of code across groups, or using code from other sources without permission of the instructor and/or TA is strictly prohibited. You are expected to demonstrate the operation of your project to the instructor or the TA.

## Project Description

Implement the **Jajodia-Mutchler voting algorithm**. Let there be eight servers, $A$, $B$, $C$, $D$, $E$, $F$, $G$ and $H$. There is only one data object $X$, replicated across the eight servers, that is subject to writes. Initially, the version number (VN), replicas updated (RU) and distinguished site (DS) values for $X$ at all the servers are 1, 8 and $H$, respectively. The rule for selection of distinguished site favors the server that is alphabetically largest among the candidates.

   Refer to Figure 1 for the sequence of network partitioning/mergers that the system should go through.
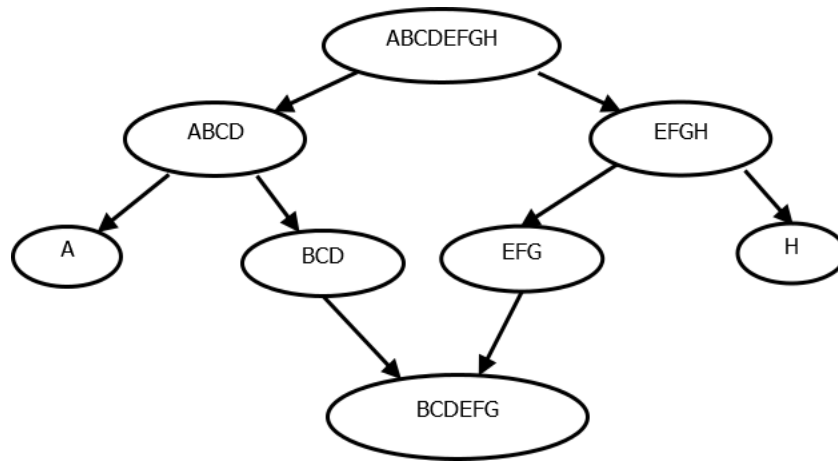


Figure 1: Sequence of network state transitions.

   Each server runs on a different machine. Initially, all the servers have reliable socket connections with each other forming one connected component as represented by the oval at the top of the figure. Subsequently, partitions and mergers happen as shown. In order to emulate partitioning and mergers, all socket connections between servers in the same partition are maintained/created, while all socket connections between servers in different partition(s) are severed.

1. You must attempt at least two writes in each of the network components shown in Figure 1. A write can originate at any of the servers in the corresponding partition.

2. After each write attempt, successful or unsuccessful, output the VN, RU and DS values for each server.

## Submission Information

Please submit your files (source code, makefile, log files) as one "zipped" or "tarred" archive using eLearning. The report should list the output of the eight servers after each write attempt.