

# Leader Election Phase 1

---

*Matt Kaiser, Risham Chokshi, Varun Shah, & Bhavin Patel*

## File Descriptions

*BootClients.java* – This program serves only to create the individual voter threads and initiate the election. The variable for number of clients is contained in here.

*Client.java* – The class that represents a client object and contains all logic for carrying out the election from a single voter perspective.

## How it works

The BootClients program creates  $N$  Client instances, where  $N$  is number specified by *BootClients.numClients*. The Boot script randomly selects  $\log_2 N$  clients to be in the candidates in the initial round of voting. The clients are then created knowing whether or not they have been selected, and their id number. The id number is simply assigned by the for loop from 1 to  $N$ . The client's port number then becomes  $3000 + id$ , so that every client can easily know the port of other clients.

In the first round, all the other clients will be voting for only these randomly selected clients. After a timer expires, the selected potential leaders then begin voting amongst themselves. Once a client has  $N/2$  votes, it becomes the leader and alerts the others to stop the election.

At the end, all clients connect to a server at port 3000 to report number of messages.

## Test Environment

All tests were run on the same machine, a Macbook Pro with a quad-core processor running OS X Yosemite. The program was run in eclipse with JDK 1.8.0u20 using a Java program to create the clients and initiate the election.

## Test Methodology

Each run is preformed by hitting the “run” button in eclipse and recording the output for total number of messages, and id of the leader elected. The number of clients is specified in BootClients.java

## Test Results

The first test is preformed with 19 clients. Here are the results for 20 runs:

Run #	Total Msgs	Leader
1	50	10
2	34	3
3	50	2
4	50	5
5	50	16
6	50	5
7	66	5
8	50	10
9	50	7
10	66	8
11	66	13
12	50	11
13	50	8
14	50	10
15	50	6
16	50	8
17	50	6
18	50	6
19	66	13
20	66	6
<b>Average:</b>	53.2	

Now, testing with 20 clients:

Run #	Total Msgs	Leader
1	53	7
2	53	6
3	70	19
4	21	3
5	70	8
6	53	10
7	53	5
8	70	6
9	53	8
10	70	4
11	53	7
12	41	4
13	23	3
14	38	4
15	53	5
16	38	1
17	53	8

18	70	4
19	53	6
20	23	4
<b>Average:</b>	50.55	