Carleton University
Department of Systems and Computer Engineering
SYSC 3303 Real-Time Concurrent Systems Winter 2015

Term Project: Electronic Voting System

The goal of the term project is to develop the Carleton University version of the Electronic Voting System**.** The traditional paper-based voting system consists of a voter manually marking the paper ballot and then the ballot being counted by hand by election officials. In elections using electronic voting or counting technologies, one or both of these processes are automated electronically. In this project, both processes must be automated electronically. There are several electoral systems. In this project, we use the Canadian electoral system which is based on a parliamentary system of government, modelled on that of the United Kingdom. The people in each electoral district vote for a candidate of their choice. The candidate who receives the most votes becomes Members of Parliament (MP) for that electoral district. There are 308 electoral districts in Canada. After an election, the party with the most elected representatives becomes the party in power. The Prime Minister is the leader of the political party with the most elected Members of Parliament.

Our user objectives within the limits of the term are:

- We do not address the communication security component.
- The system must be constituted a number of clients/servers and Fig.1 shows an overall view of Electronic Voting System.
- The voters and the candidates must be registered before vote.

Our technical requirements

- The system will all be written in Java, in the hope that you'll continue and extend your work.
- All properties should be configurable, such as the number of:
    - voters,
    - candidates,
    - districts,
    - clients/servers …
- The system shall be built as a distributed system using a client/server model that implements the Model-View-Controller pattern.
- Servers must contain all the logic of the electoral system.
- The clients shall contain a view of the electoral system.
- The system shall be able to be run on separate machines.
- All network communication shall be done with UDP.
- The servers must concurrently queue incoming clients and process the controls in FIFO order.
- The view of the electoral system shall be a "graphical User Interface" table of the current election results:
    - The servers 3 shall provide periodic updates of the election results. The period must be configurable.

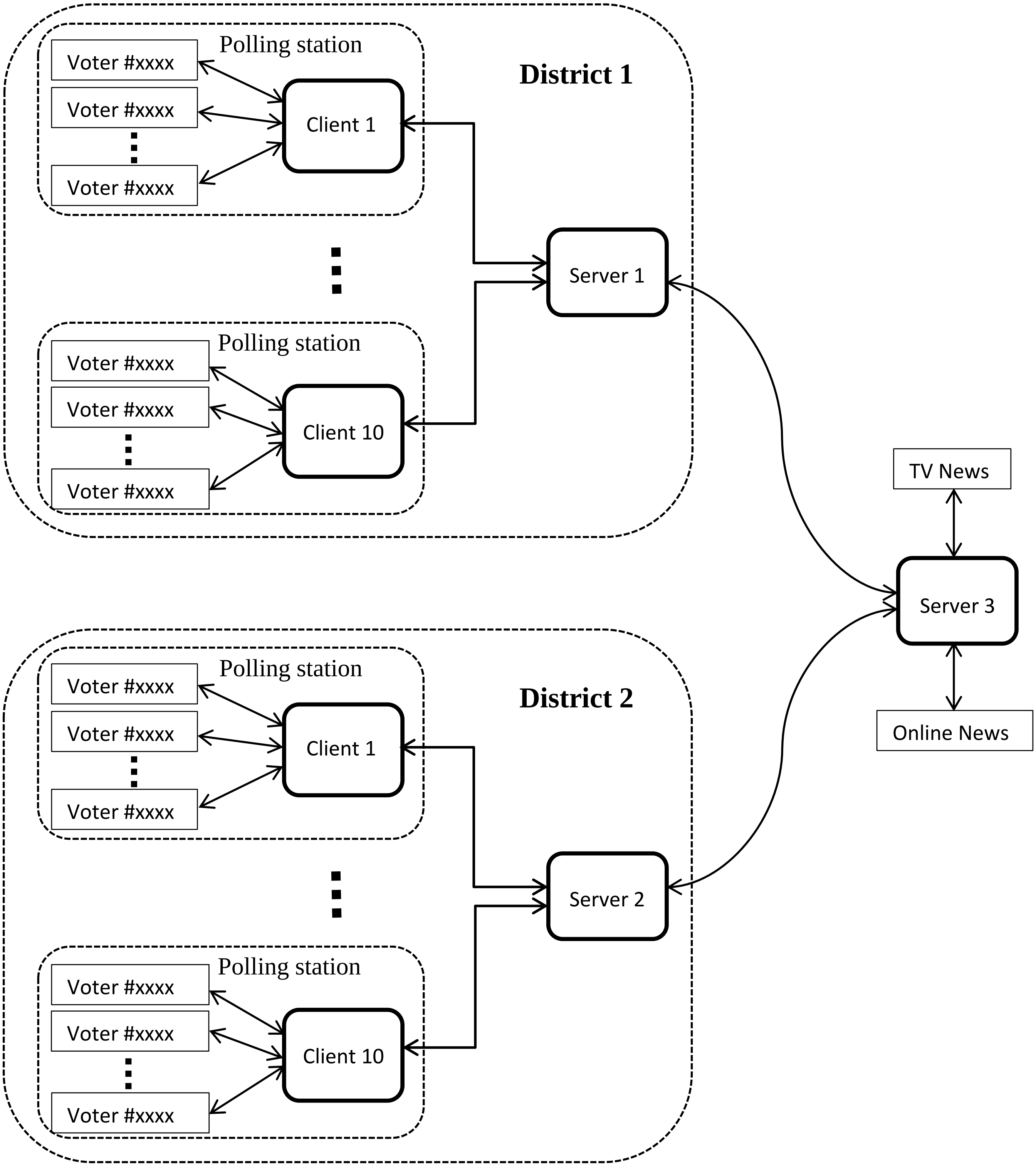

Fig. 1: overall view of Electronic Voting System

To demonstrate the proper functionality of Electronic Voting System, consider three districts. Each district consists of 10 client applications and one server. The latter must implement the concept of semaphore to allocate a connection to each client application. The Each district server stores data election in a table. As shown in Fig. 1, the data are from different client applications. To ensure data integrity, you must use the concept of mutual exclusion. The table contains the profile of each voter and the list of candidates. The profile of voter contains: last name, first name, ID, address, district name, etc. This profile is done during the voter registration. In each district, the client application must identify the voter before voting. Note that only the registered voter has the right to vote. The client application must have an interface that allows registering and voting. The server of each district must send the preliminary election result to the server 3 periodically.

Once the polling stations are closed, everybody can see the results of election via server 3 to know the party that won, elected members and those who are defeated. Since the server 3 contains the cumulative results of the election from different servers, ensure the integrity of the data.

Milestones

The project will have three milestones. Dates will be posted online. The expectations for each milestone will be posted. The details for each milestone are explained in general terms below, while the exact expectations will be enumerated in the marking scheme posted before the milestone's deadline.

1. During your lab period, each team will demo their current system to their assigned TA. All team members must be present.
2. At the end of your lab period, the team must submit all code and documents for that milestone as a single .zip file. Use the CULearn LAB page for submission.
    - The zip file must include a README.txt that provided detailed instructions for running your program. If the TA cannot run your program, you will lose any marks for parts that cannot be verified by the TA.

Project marking

The project is worth 25 percent of the final grade. All team members will receive the same grade, if all members contribute equally. See the course outline for the grading of malfunctioning groups. A single peer review will be done at the end of the term to allow confidential concerns about members' contributions. You are encouraged to come forward early – either to your TA or to the instructor – so that the problem can be remedied quickly.

General Criteria of Project Evaluation

Scope of the project

- Embedding in/of existing paradigms, systems, software or algorithms
- Scale of analysis/design/implementation/testing

Realization of the projects

- Have the goals been accomplished?
- Quality and method of the solution
- Has the team done what was asked?
- Has the team done more than was asked and taken additional initiatives?

Project text

- Correctness
- Completeness
- Clarity of the text
- Structure
- Language
- Clear and well interpretable graphs and figures

Score Project

The qualitative score assignment is as follows:

- 90% - 100% = exceptional result
- 80% - 89% = very good result
- 70% - 79% = good result
- 60% - 69% = acceptable result
- 50% - 59% = minimally acceptable
- 49% or less = unacceptable