Modeling and Simulation of a Robust e-Voting System

Mohammad Malkawi, Associate Professor, Argosy University, and VP, *AIM Wireless USA*
mmalkawi@aimws.com
Mohammed Khasawneh[1], *IEEE Senior Member*, *College of Engineering*, mkha@ieee.org
Omar Al-Jarrah[2], *Associate Professor of Computer Eng.,* and Laith Barakat[2]
[1]University of Illinois at Urbana Champaign
[2]Jordan University of Science & Technology

Abstract

In this paper we present a simulation model for a multifaceted online e-Voting system. The proposed model is capable of handling electronic ballots with multiple scopes at the same time, e.g., presidential, municipal, and parliamentary, amongst others. The model caters for integrity of an election process in terms of the functional and non-functional requirements. The functional requirements embedded in the design of the proposed system warrant well-secured identification and authentication processes for the voter through the use of combined simple biometrics. Of utmost importance are the requirements for correctness, robustness, coherence, consistency, and security. To verify the robustness and reliability of the proposed system, intensive computer simulations were run under varying voting environments, viz. voter density, voter inter-arrival times, introduced acts of malice, etc. Results of the simulations show the impact of several parameters on the performance of the system. These results provide the proper grounds that would guide the decision maker in customizing an e-voting system.

Key-Words: e-voting, modeling and simulation, performance evaluation

1. Introduction

Election is a process in which voters choose their representatives and express their preferences for the way that they will be governed. Correctness, robustness to fraudulent behaviors, coherence, consistency, security, and transparency of voting are all key requirements for the integrity of an election process. There is a wide variety of different voting systems that are based on traditional paper ballots, mechanical devices, or electronic ballots [1]. In a traditional paper ballots, voters choose or mark their favourite choices on ballots and place them in boxes, which are sealed and officially opened under special conditions to warrant transparency. The ballots are then counted manually, which is a tedious process that is subject to human error. With voting via mechanical systems, meanwhile, voters make their choices by pulling down on mechanical levers that correspond to their favourite choice of candidates. Each lever has a mechanical counter that reports the number of votes for that position. These machines are no longer manufactured [1]. On the other hand, some systems use punch cards where voters punch holes in computer readable ballot cards. These systems are not reliable because of problems in reading cards and were replaced by optical scan device systems, which allow voters to record choices by filling in areas on the ballots. The ballots are read using a computer scanner and then the votes are counted automatically using a computer program [1]. Finally, special-purpose computers are used as voting machines where voters use touch screens or push buttons to select choices, which are stored and counted or processed by a special program on the same machine [1].

Often times, however, counting errors take place, and in some cases, voters find ways to vote more than once, introducing irregularities in the final count results, which could, in rare cases, require a repeat of the election process altogether! Moreover, in some countries, purposely introduced manipulations of the votes take place to distort the results of an election in favour of certain candidates. Although such mishaps can be avoided with a properly scrutinized election process, errors can still occur, especially when the number of voters is quite large. Quite often, international monitoring bodies are required to monitor elections in certain countries.

The advancement of information and telecommunications technologies allow for a fully automated online computerized election process. In addition to overcoming commonly encountered election pitfalls, electoral vote counts are done in real time that by the end of elections day, the results are automatically out [2, 3]. The election process can be easily enhanced with various features based on the demand and requirements of different countries around the world. E-Voting is an inter-disciplinary subject and should be studied together with the experts of different domains, such as software engineering, cryptography, politics, law, economics and social sciences. Although many people have worked on this subject, mostly e-Voting is known as a challenging topic in cryptography because of the need to achieve voter anonymity, and therefore, to ensure his/her privacy [4]. However, many studies warn against the adoption of e-Voting because of many challenges in software engineering, security, and auditing [1, 4 - 7].

Due to worldwide advancements in computer and telecommunication technologies supported by the underlying infrastructures, online voting or e-Voting is no longer a North American or Western

phenomenon. This high tech method of casting a ballot has spread far beyond the United States, expanding throughout the entire world. E-Voting, along with its benefits and detriments, can now be found from the developed countries of Europe to the developing countries of Asia and South America. The introduction of electronic voting has been the biggest change, for instance, to the Irish electoral system since the establishment of the state over 80 years ago. E-Voting may soon become a global reality or a global nightmare [8 - 10]. In 2003, a new e-Voting system was introduced in Belgium in two locations to convince citizens that the system was trustworthy [11]. They introduced a "Ticketing" system where the voter prints and approves a hard copy of his/her vote. At the end of the elections, all of the paper votes (tickets) are counted and compared to the electronic result. In this election, there was an electronic voting problem reported where one candidate got 4096 extra votes because of a technical problem [11]. Besides reliable e-Voting technologies, there is a dire need for international standards to govern the technology, the software reliability and accuracy, the processes and algorithms deployed within the technology, and the verification of all hardware, software and protocols involved. Such standards will eventually allow elections to proceed in any part of the world without the need for monitoring bodies. The design of a "good" voting system, whether electronic or using traditional paper ballots or mechanical devices must satisfy a number of sometimes competing criteria including a high degree of security and accuracy, eligibility and authentication, integrity, verifiability and auditability, reliability, flexibility, performance and scalability [12, 13].

More importantly, there is a real need for a good simulation model which can guide the deployment of e-Voting resources such that the election process can proceed with minimal faults and performance issues. In this paper, we provide a simulation model for a generic e-Voting process. The model is designed to be flexible enough to be adapted to different election environments. The objective of the simulation model is to study the effect of several parameters on the course of an election process. Simulation results provided by the model, for a particular election process, allow offices administering an election process to deploy adequate hardware and networking resources to make the process as successful as possible.

There are several parameters which impact any voting process. The rate at which voters arrive at voting centers has a direct impact on overall system performance. Hence, a heavy arrival rate at a certain voting center may require more voting stations in order to complete the voting process in a timely manner. The simulation model allows for a good estimation of the number of voting stations at each center based on a predicted average arrival rate of voters and the total number of registered voters in a certain district. The available bandwidth for the communication links is an important factor as well. Note that the bandwidth may vary within the same country. For example (in some countries) DSL links are available only to particular localities, while dial up links are used more often in other areas. The message size used for communication between a voting station and the central servers has a notable impact on overall performance. This, in turn, will dictate the type and size of authentication traffic that can be accommodated by the system. Another important parameter is the architecture of the data management system. Here, the performance of the system is directly impacted with the use of either a centralized or distributed approach for data storage, manipulation and management. A distributed approach, however, introduces more challenges for maintaining accuracy/currency of the voting process. These parameters are implemented in a generalized simulation model.

In the simulation model, several metrics are used to evaluate overall performance and system behavior. The main metrics will be the voting (simulation) time, which represents the ability of the system to execute the voting process in the allotted timeline. Internally, the average queue length (average number of voters waiting to vote) is another metric. The average waiting time per voter is yet another important metric. One of the key tuning parameters would be the number of voting stations required at each voting center. This parameter is important, because it, by and large, is the only parameter which can be tuned during the voting process (given the availability of hardware resources). The voting center manager can (in principle) add or remove stations as deemed appropriate.

The rest of this paper is organized as follows. In the next section (section II) we describe in more detail the general e-Voting model, while the simulation model is introduced in section III. Simulation results are presented in section IV. Finally, the paper is concluded in section V.

2. The Proposed e-Voting Model

Automating an election process, while relying on state-of-the-art in computer and ICT technologies, can significantly mitigate many of the factors that would hamper a healthy progress of a given election process. For automated e-Voting processes to be fully acceptable worldwide, several issues must be addressed and resolved. Among these issues are authentication/validation, security, robustness, performance and correctness. Given the short history of e-Voting systems across the world and the inherent limitations in the scope of implementation, it is very difficult to measure the success or failure of any or all of the issues mentioned above. In addition, any voting process, as mentioned earlier, is

bound by regulations and cultural values that characterize the different societies involved. Hence, the example of one country may not directly suite the example of another. As a result, it is highly recommended to build a simulation model whereby an e-Voting system can be evaluated and various attributes adequately assessed before one is deployed.

This paper introduces one simulation model, where we address the main factors which directly contribute to the success of a voting process. The simulation parameters can be changed based on the peculiarities of any entity. The main components of the architecture of the model are shown in Figure 1. This is a client/server web-enabled architecture.

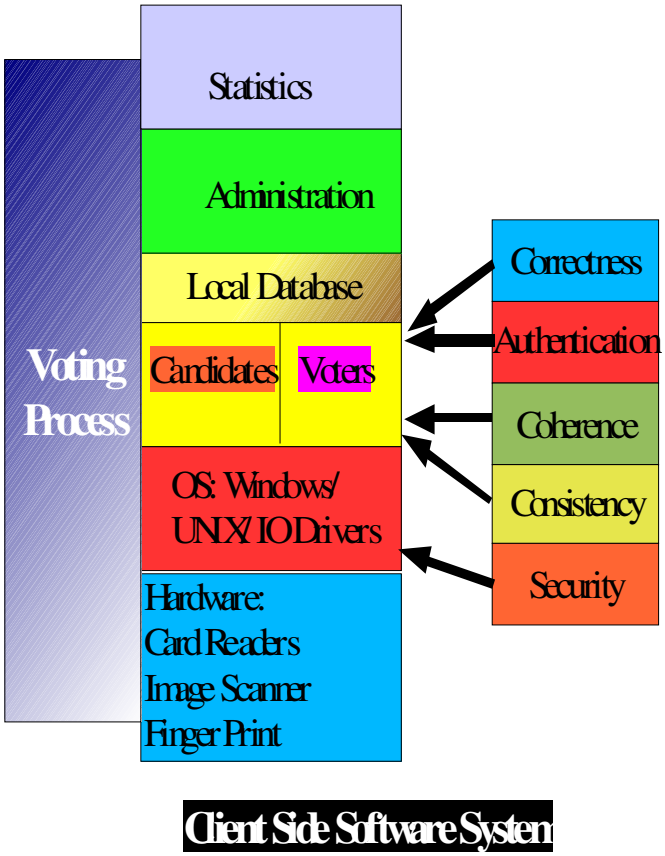

Figure 1a

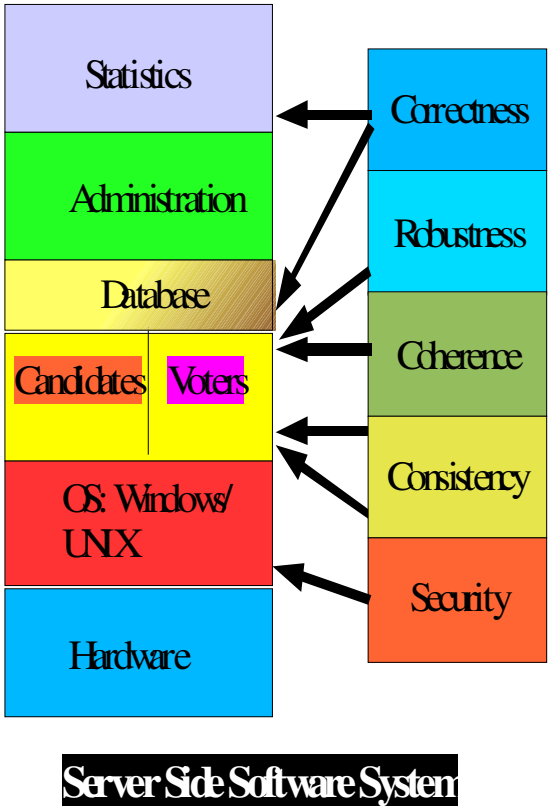

Figure 1b

The server side hosts the central database for the voting as well as the candidate population. The server also collects basic statistics related to an ongoing election process (some statistics can be turned on or off based on the needs and requirements of each election unit). Besides the main functional properties of a voting system, as described in the previous section, the e-Voting system must cater for several essential non-functional requirements. Of utmost importance are the requirements for correctness, robustness, coherence, consistency, performance and security.

The client side represents a voting station, where voters cast their votes. Note that the hardware on the client side includes IO devices for verification and authentication (e.g., image scanners, ID card readers, finger print readers, etc.). In addition to that, two more requirements are necessary. In order to reduce the traffic rate on the network links, a local database at the client side is required to host the data which pertains to the local voting center. This DB is a rather dynamic one, in the sense that the data stored in its tables may vary over the election time period. The size of the local DB at any voting center is only a small fraction of the global DB at the server side. The use of a local DB enhances the performance of the voting process. However, this approach creates a synchronization problem, which will be addressed later in this paper. The alternative is to use one centralized DB. The second requirement is the transparency of the voting process. In essence, a voter on an electronic voting station casts his/her vote to a computer. The voter does not have an insight on how his/her vote is translated and/or tallied. In a paper-based election, the ballot is filled out by the voter and dropped into a sealed box by the voter himself/herself. Votes are

counted in the presence of candidates or their representatives.

The voter is certain that his/her exact ballot with his/her vote selection is placed in the appropriate box. Of course, ambiguity in the ballot formats (as was the case in the US presidential election in 2000) may render the transparency a rather deceiving one. In an electronic version, the voter puts his trust into computer hardware, software and network infrastructure that process his/her vote. Hence, the e-Voting system in its broadest form may render the process a non-transparent one [8, 14]. This issue can be resolved by printing a hardcopy of each vote for the voter to keep for his/her records. Another copy is printed, possibly in the form of a bar code, and saved for later verification. In order to verify the accuracy and correctness of the process, a random sample of the saved copies can be scanned and verified against the votes stored in the DB.

The identification of a voter is done via a card reader which reads off his/her official ID card and retrieves the voter record from the local DB (on the client side) or loads the record from the central DB if it is not already in the local one. Records are loaded dynamically from the central DB to the local DB's either on demand or on a pre-fetch basis. The voter record includes, amongst others, a biometric description of the voter in question. In this study, we use a fingerprint authentication method (other methods can be added to the model). The voter will be rejected if his/her fingerprints do not match the stored record. In order to reduce false rejections, we store for each voter several copies of his/her fingerprints taken at different time periods. Fingerprints are stored as an encoded text in order to reduce storage consumed by images. This dual process should guarantee that no voter can falsely impersonate another. Note that the use of fingerprints or any other scanned image directly impacts the message size and hence the performance of the network. Hence, a distributed database approach is preferable over a centralized approach.

The accuracy and correctness of the e-Voting process can be further jeopardized if the same voter casts two or more votes, or a vote is not properly added to the overall count of the right candidate. Such mishaps may come about as a result of synchronization conflicts at the central DB level. In order to prevent two or more votes per voter, we use a "voting status flag" in the voter record. This flag is initialized to FALSE. The voting status flag is set to TRUE in the central DB whenever a voter identity is verified (before authentication takes place). If the authentication fails, the flag is reset to FALSE. If the voter leaves the station without completing a vote, the flag is also reset to FALSE; thus allowing the voter another chance to try again and cast his/her vote. If the voter successfully completes the voting process, the flag remains set to TRUE. Note that even if the result of the vote is not committed to the central DB in due time, the flag in the voter's central record is set to TRUE, thus eliminating the possibility of another attempted voting by the same voter, or by someone who carries a counterfeit ID card. This requires that whenever the record of a voter is accessed for identification, even when the record is found at the local DB, the flag on the central record must be checked. If it has already been set to TRUE, the voter is denied access and his/her attempt fails to go through. If two people carrying the same ID card (one is authentic while the other is counterfeit) attempt to vote simultaneously, the first one to access the record will set the flag to TRUE, load the record and prevent the other one from accessing the record. Of course, if the one with the counterfeit card obtains the record first, the vote cast will fail at the next authentication step. It is possible that a record gets loaded into two different voting centers due to block transfer from the central DB into local DB's. When a voter attempts to access the record from any of the stations, the client will verify the central record flag. If it has been set to TRUE, access is denied; otherwise it sets the flag to TRUE and access is granted. Note that simultaneous requests to the same record will be synchronized by the DB query serialization process (only one query may access any table at any given time). This mandatory check of the flag in the central DB will add extra overhead on the network. This overhead is already included in evaluating the simulator performance and is reflected into the ensuing simulations.

Another synchronization resolution is required when a vote is to be tallied into the record of a candidate. If a candidate is being selected by several voters at the same time, then a certain assignment plan needs to be put in place so that all votes will be tallied (no misses) and added to the candidate's record. Again, we use a "COUNT" flag/mutex for the candidate's record. The COUNT flag is initially set to FALSE. When the record is selected by a voter, the flag is set to TRUE until the record count is updated, then the flag is reset to FALSE. All votes for the same candidate will be queued until the flag is reset to FALSE. In order to improve the 'hit' performance, a

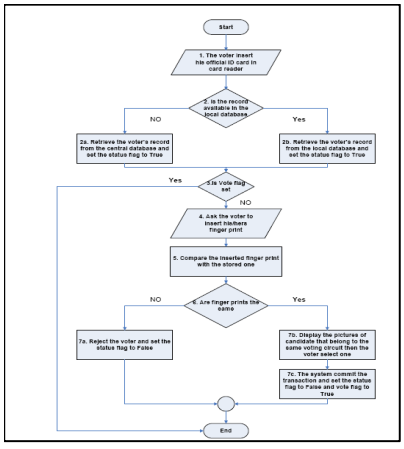

Figure 2: Voting process flow chart

counting semaphore COUNT can be used instead. A copy of the vote will be printed only when the vote is successful and the candidate's record is updated. This requirement, initially made for transparency purposes, provides a final test for the accuracy and correctness of the process, especially in the presence of thread hang-ups. Figure 2 shows a flowchart of the voting model.

The overall architecture of the system is shown in Figure 3. The central database in Figure 3, which is mirrored out for reliability reasons, is used to store all relevant information on the candidates and voters. Voting centers are distributed around the given country. One or more voting centers could share a local database. A voting center consists of one local DB server and one or more voting stations.

Information security is very important to our system. There is inherent need to secure all the communications between the clients and their local DB servers. We also need to secure the communications between the local DB(s) and the central DB server [15]. Security concerns and solutions will be presented in a separate work by the authors.

Each voting station is equipped with a card reader, a fingerprint scanner, a touch screen, and a multimedia subsystem. The multimedia subsystem is used for people with special needs (physically challenged), such as the blind and those with difficulties in reading or comprehending images, texts, or sounds. Figure 3 also shows the structure of a voting station.

The proposed system is capable of handling electronic ballots with multiple scopes at the same time, e.g. presidential, municipal, parliamentary, and others. However, the simulation environment in this study is designed only for a single voting scope.

3. Simulation Study of the Proposed Model

The simulation model is a general (M/S/G) queuing model. Voters randomly arrive at a voting center according to a Poisson random process. The inter-arrival rate is controlled by the mean time between two successive arrivals (mean interarrival time,μ) and is governed by an exponential distribution. The simulator allows for as many voting centers as needed. The rate of voter arrival varies over time; low arrival rates characterize early hours of the day; heavy arrival rates characterize mid morning times and close to elections closing time. In the simulator, we choose μ =10, 5 and 2 for low, moderate and large inter-arrival rates, respectively.

Voters arrive at a voting center in a rather clustered manner; *i. e.*, in groups. The average size of a cluster is a Poisson random variable with mean (λ). In our simulations, we use λ=2 and λ=5 for low and high voter densities, respectively. For example, λ=5 and μ=10 represents the case when voters arrive in large clusters but at rather dispersed time intervals. The number of voters expected to vote at a given voting center is predefined. The model simulates centers with as few as 2000 voters and centers with as many as 20,000 voters. In general, the simulator model allows for a random number of voters to be selected per center. The simulator is expected to run until all voters registered at each center have cast their votes. Simulation is normally set to complete within 12 hours (typically voting begins at 7:00 am and completes at 7:00 pm). However, the simulator can be tuned for any required simulation time period.

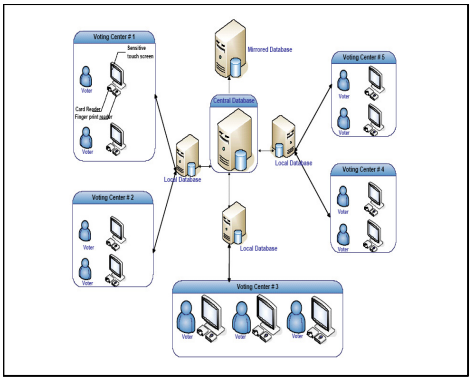

Figure 3: General view of system architecture

Voters are queued at the voting stations within each voting center. A voting center consists of (N) voting stations, and hence N queues. When a group of voters arrive, the simulator moves the voters to the appropriate queue either in a round robin manner (one voter per queue at a time) or enters the whole cluster of voters into the least loaded queue. Both scheduling policies are evaluated in the simulator. Other policies can be added and evaluated.

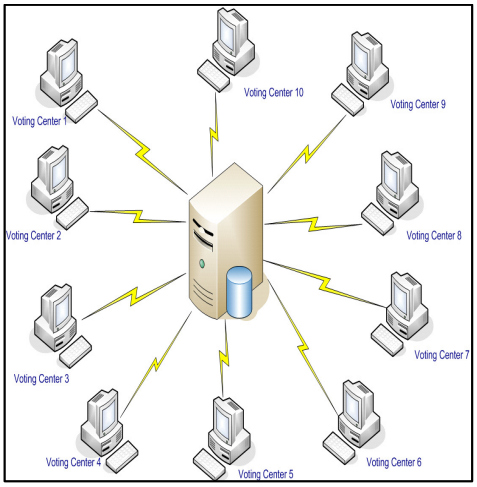

Figure 4: simulation environment

The service time (i.e., the time it takes to complete one whole voting transaction) is a blend of real time execution and random delay due to several factors. The random delay is made up of the average time required to read the voter's ID and scan the fingerprint when a voter is de-queued and selected to vote. This average is empirically determined using typical card readers, scanners and touch screen monitors. The verification and authentication processing time consists of real-time access to an Oracle database. DB transactions undertaken in the simulator include setting the voter flag as well as the candidate Counter semaphores.

In the simulation model, as figure 4 shows, the central DB server and the local voting station servers are located within the same local network segment. Therefore, we introduce a random delay to compensate for inter-net transfer time. The transfer time is a function of the available network bandwidth and message size. We use several bandwidth sizes in this study. A 1 Mbps is a rather conservative bandwidth and is typical of many voting locations around the world.

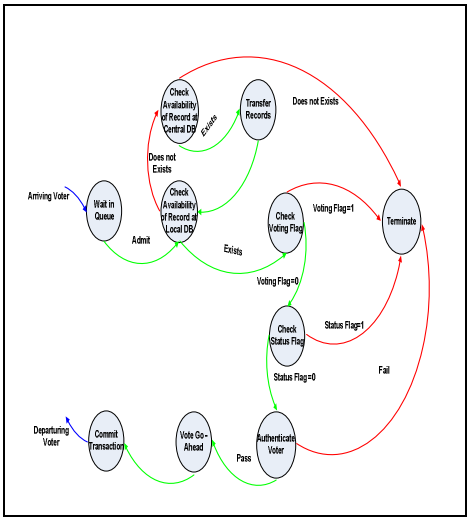

Figure 5: State Diagram of the Proposed Simulation Model

We also use a smaller bandwidth (128 Kbps) which is typical of dialup locations. Furthermore, we use a 10 Mbps bandwidth for more developed network infrastructures. The simulator uses a Weibull distribution to model the bandwidth which accounts for delays in the network. The message size is a function of the voter and candidate records. The simulator runs for several message sizes. Note that the use of biometric measures for verification and authentication produces larger message sizes and, hence, increases the overall transfer time of data across the network.

The voting process is shown in the state diagram of Figure (5). The simulations were averaged over five voting runs. This is particularly important because the model entails several random factors.

The simulation environment entails a live Oracle database system for voters and candidates. Besides personal identification information, the records include authentication information and locality of a voter and/or a candidate. The simulator, also, includes modules which emulate the arrival of voters at voting centers and the voting process itself. The simulator allows a voter to cast a vote at any voting center, irrespective of his actual voting district (locality). This is one of the main advantages of an online e-Voting system.

We have conducted a fairly large number of simulations of the proposed voting system, taking the number of voters over a sample range starting at 2000 voters per voting center and ending at 20,000 voters per voting center. We realize that the number of voters in a given locality may be much larger than the numbers we used in the simulator. However, the simulation results are fairly scalable where the simulation model is capable of modeling fairly large number of voters. We fixed the number of voters at a given voting center in the simulator. Although in reality, this number may vary by a small percentage due to the fact that people will be allowed to vote at any other center they choose for the sake of voting convenience, especially those voters residing at townships outside their voting districts, or those voters casting their votes through embassies away from their home country/ies.

4. Simulation Results:

In the following subsections we discuss the impact of various model parameters on the overall model and its performance.

4.1. *Centralized Versus Distributed DB*

The architecture of the proposed e-Voting model can implement either a centralized or distributed DB approach as discussed earlier. The centralized approach keeps all the records of the voters and the candidates in a central DB located at a central server. Each voting transaction must interact with the central DB. In a distributed approach, however, local servers at voting centers download voter and candidate records most relevant to the local center on demand or on a pre-fetch basis. In the centralized approach, data consistency is not a serious issue since all data is maintained at one central DB. However, this approach causes a serious performance problem both in DB response time and network traffic. In the simulated model, we implement a quasi-distributed approach. A central DB is used to host all candidate and voter data. Each voting center has a local DB server which loads the candidates as well as the voters registered at the local voting center. The data consistency issue and the accuracy of voting are maintained by means of

synchronization flags as discussed earlier (the VOTING and STATUS flags shown in Figure 5). When the flags were turned off we noticed several violations of the voting accuracy. In the worst case scenario, we noticed 1.2% error rate; where the error is manifested in a cast vote not being reported in the final results for a given candidate. With the use of the synchronization flags, errors of the like are totally eliminated.

4.2. *Number of Voting Stations*

When the number of voters at a given center is relatively large, or the arrival rate of voters is high, it is recommended to add more voting stations to the center. In a centralized approach, the addition of voting stations may not improve the results significantly.

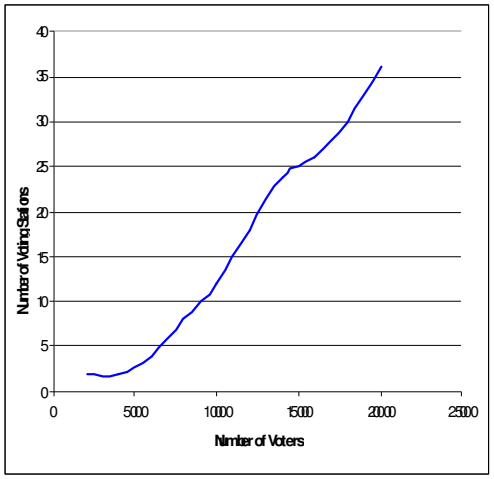

Figure 6: Voting Center Configuration

More voting stations simply shift the overload from the local station on to the main (central) server. However, in a distributed approach, the addition of local stations will distribute the load locally without significantly impacting the main server.

Figures (6) shows the number of voting stations required to complete all voting transactions within 12 hours for different voting populations. For example, 36 voting stations are needed to allow all 20000 voters to vote within 12 hours (7:00 am to 7:00 pm). The results shown in the figure assume a 1 Mb/s bandwidth and a clustered policy in a distributed architecture.

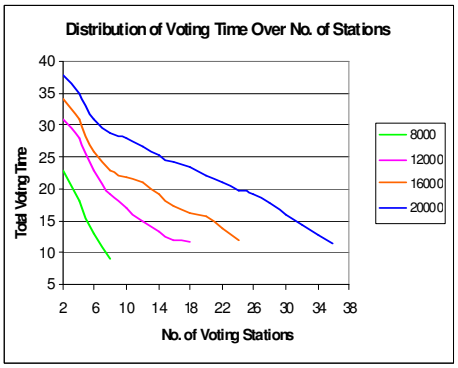

Figure 7

This allows the administering body in a certain country to properly size a given voting center given the number of voting populace and the available bandwidth. Figure (7) shows the time required to complete all voting transactions given a certain number of voting stations and a voting population. This figure illustrates the trade-off between the voting time and the number of voting stations. In countries, where the cost of voting stations can be a real burden, the voting time period can be extended. For example, 20000 voters can be served by 2 stations over 35 hours or can be served in 12 hours using 36 voting stations.

4.3. *Network Bandwidth*

The wide area network bandwidth has a direct impact on the model performance. Figure (8) shows the average service time for three different bandwidth values (128 Kb/s, 1 Mb/s, and 10 Mb/s).

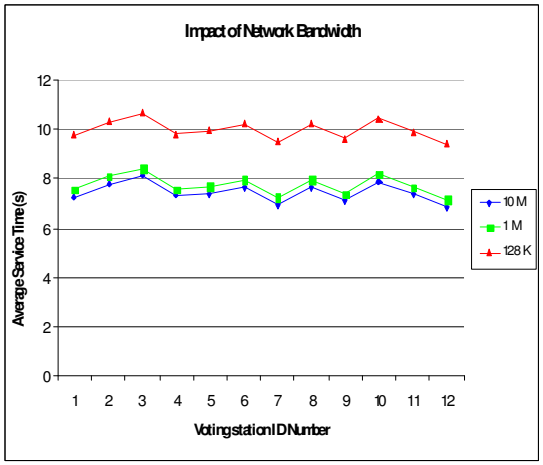

Figure 8

We show the results for different voting stations; here, we show the network effect at the various

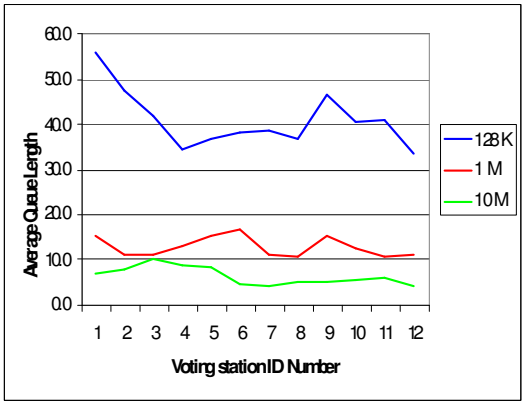

Figure 9: Network Bandwidth

voting stations (in this case 12 stations) being incubated at a particular voting center. The service time is defined as the time it takes to process a vote

once the voter has been admitted to the system. This time includes identity verification, authentication, flag checks, the update to the central DB and the statistics update as shown in Figure 8. Beyond a bandwidth of 1 Mb, any pronounced improvement begins to diminish. The x-axis represents the voting station ID used to process the voter population. The results in Figure 8 are averaged over a 12-hour simulation timeframe for 10,000 voters.

Furthermore, the simulator was tuned to run and accommodate all voters within a 12-hour timeframe. Figure (9) shows the average queue length for different bandwidths. The shorter the queue is the less time a voter expects to wait before casting a vote at a voting station. The average queue length can be 3 times as big for low speed network connections (128 Kb/s). Figure (9) also confirms that the performance improvement beyond 1 Mb/s is rather insignificant.

4.4. *Scheduling Policy*

The scheduling of voters to the available voting stations at a given voting center has a direct impact on the overall performance of the voting process.

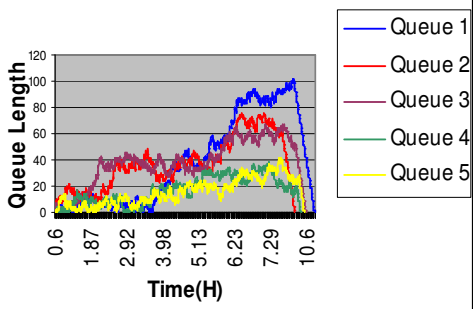

Figure 10a: Clustured Scheduling

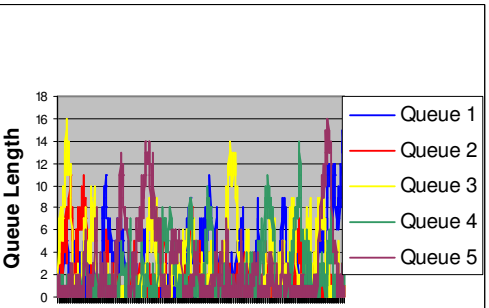

Figure 10b: Round Robin Scheduling

Voters arrive at a station at an average rate of (1/ μ) according to an exponential inter arrival rate. Furthermore, voters arrive in clusters (groups at a time) with a mean (λ) according to a Poisson random process. The scheduling policy used in this study (clustered) allocates all (λ) voters to the station with the shortest queue length. We observed that this scheduling policy, although intuitive, leads to the fact that some stations may become free, while others remain busy, especially with large number of voting stations in the voting center. For comparison purposes, we implemented another policy (round robin) which aims at keeping a balance between the various voting stations. This policy allocates voters in a round robin manner across all voting stations. Figures (10a and 10b) show the average queue length at 5 voting stations using clustered and round robin scheduling policies, respectively. Note that the round robin policy maintains a fairly good balance of voters among all queues. The clustered policy shows a wide variation among the voting stations in terms of the queue lengths and the completion time. Some of the stations were observed to finish up all the voters within 10 hours, while others persisted for more than 12 hours. The main advantage of the clustered policy is the simplicity of implementation and low cost, since it does not require any distribution of voters among stations; whereas the round robin requires more personnel to manage the voting process.

5. Conclusions

In this paper, we have proposed an online e-Voting system which can tackle all earlier issues encountered in a conventional (manual) voting system. The new system maintains voting statistics in real-time while preserving the integrity of the voting process from the minute a voter steps in to cast his/her vote until the cast vote is registered in favour of the chosen candidate at a globally allocated DB repository. While observing full-fledged voting transparency, at the voter as well as the system levels, the proposed system is capable of denying access to any illegal voter/s, preventing multiple votes by the same voter, and blocking any introduced forms of malice that would adversely affect the voting process altogether. Moreover, the proposed voting system caters for the needs of the physically challenged voters by providing special multimedia amenities that would facilitate voting to a voter's convenience.

Simulation results of the system, while running a live DB backend server, reveal a number of important factors that ought to be assessed carefully by the party adopting a system like this one, for any form of election activities, prior to its final deployment. These factors address the number of voting stations needed at any voting center, as outlined by the voting needs of a given voting

district, the network bandwidth requirement by a given voting center, and the size of the local DB to support the needs of a given voting locality, amongst others. The system, via these simulations, has shown ruggedness and sustained reliability in terms of preventing multiple votes by the same voter, and maintaining internal system audits that would warrant no missed votes, per candidate, in the process of voting.

With the use of an e-Voting system, as the one proposed in this paper, many of the issues, that had long challenged traditional voting systems, are bound to be resolved providing a peace of mind to both voters and election candidates. It is well expected that with a well administered/designed e-Voting system, countries that have long been observed by international monitoring bodies, while carrying out election processes of their own, will soon be able to work on their own and, yet, achieve the election integrity they have longed for.

Issues of security and reliability at the central DB, the communication links and local servers will be further addressed by the authors in a separate study.

References

1. National Science Foundation. Report on the National Workshop on Internet Voting: Issues and Research Agenda, Mar. 2001. http://news.findlaw.com/cnn/docs/voting/nsfe-voterprt.pdf
2. R. Mercuri. Electronic Vote Tabulation Checks and Balances. PhD thesis, University of Pennsylvania, Philadelphia, PA, October 2000.
3. A. D. Rubin. Security considerations for remote electronic voting. Communications of the ACM, 45(12):39–44, December 2002. http://avirubin.com/e-voting.security.html
4. Cetinkaya, O. and Cetinkaya, D. (2007) "Verification and Validation Issues in Electronic Voting" The Electronic Journal of e-Government Volume 5 Issue 2, pp 117 - 126, available online at www.ejeg.com
5. "How E-voting Works", [Online], Available: http://people.howstuffworks.com/e-voting.htm/printable
6. Caltech/MIT Voting Technology Project, "What Is, What Could Be," July 2001, http://www.vote.caltech.edu/Reports/2001report.html ; Carter et al, To Assure Pride and Confidence; The Constitution Project's Forum on Elections, "Building Consensus for Election Reform," and "Recommendations for Congressional Action," August 2001
7. California Internet Voting Task Force. A Report on the Feasibility of Internet Voting, Jan. 2000. http://www.ss.ca.gov/executive/ivote/.
8. McGaley Margaret, McCarthy Joe, "Transparency and eVoting: Democratic vs. commercial interests", 2004. www.cs.nuim.ie/~mmcgaley/Download/Transparency.pdf
9. Online Voting. Parliamentary Office of Science and Technology. May 2001. www.parliament.uk/post/pn155.pdf
10. McGaley, Margaret. "Irish Citizens for Trustworthy Voting." 6 July 2004. http://evoting.cs.may.ie/
11. "Electronic voting in Belgium",[Online], Available: http://en.wikipedia.org/wiki/Electronic_voting_in_Belgium
12. "The Problem with Electronic Voting Machines", 2004. [Online], Available: http://www.schneier.com/blog/archives/2004/11/the_problem_wit.html
13. "Development of remotely secure e-voting system" Keshk, A.E.; Abdul-Kader, H.M.; Information and Communications Technology, 2007.
14. TADAYOSHI KOHNO, ADAM STUBBLEFIELD, AVIEL D. RUBIN, DAN S. WALLACH: Analysis of an Electronic Voting System, IEEE Symposium on Security and Privacy 2004 Introduction to Computer Security, by Matt Bishop (ISBN: 0-201-44099-7), Addison-Wesley 2005.