Internet voting: Security and Performance Issues

Mohammed Ismael Ahmed, Mohammed Abo-Rizka

Faculty of Computing and Information Technology, Arab Academy, Cairo, Egypt

mohammed-ismael@live.com

Abstract

The growth of internet applications and services have led to apply the security issue in order to achieve a good performance.

In spite of the ongoing technological enhancementssuch as (faster servers and clients, multi-threaded browsers supporting several simultaneous and persistent TCP connections, access to network with larger bandwidth for both servers and clients), the network performance is captured by response time and throughput do not keep up and progressively degrades.

One of these online systems is remote Internet voting systems.Nowadays, many governments are using electronic voting as adifferent voting channel to allow voters to cast their votes remotely. Therefore, in order to provide such a system,it is necessary to evaluate the security and performance by taking into account all security and performanceissues. The implemented measurements of security and performance must be identified and their effectiveness on these risks should be evaluated.

In this paper, willproposea genericmodel of remote internet voting to evaluate performance and security requirements.

Keywords: *Electronic Voting, Two-way factor authentication, Mobile-ID, RemoteInternet voting, Performance evaluation.*

1. Introduction

Electronic voting (e-voting) would be more convenient, relatively secure and utilize fewer resources. To be able to access e-voting system from personal, business or even public computers may be more suitable for many people needing to vote. This could potentially be a solution for the low voter turnout at the polls. However, it is still questionable whether elections can be conducted online or over the Internet due to the high level of concern over security.

There is a wide range of different voting systems that have based on traditional paper ballots, mechanical devices, or electronic ballots.

As figure 1 mentions ,one of the electronic voting categoriesis the internet votingwhich has two different types, either a controlled environment or an uncontrolled environment. A controlled voting environment means the voting machines such as (computers) is under control of the election authority. Wheras, an uncontrolled environment means the voter

could use their personal computers, workplace computer, or any public computer to cast their votes [1].

The e-voting system should be based on the following requirements:

1. The e-voting system must be available during election time.
2. The system must provide ease of use.
3. The system must prevent a voter from casting more than one vote.
4. A system must be able to verify voter.
5. The system must count votes correctly.

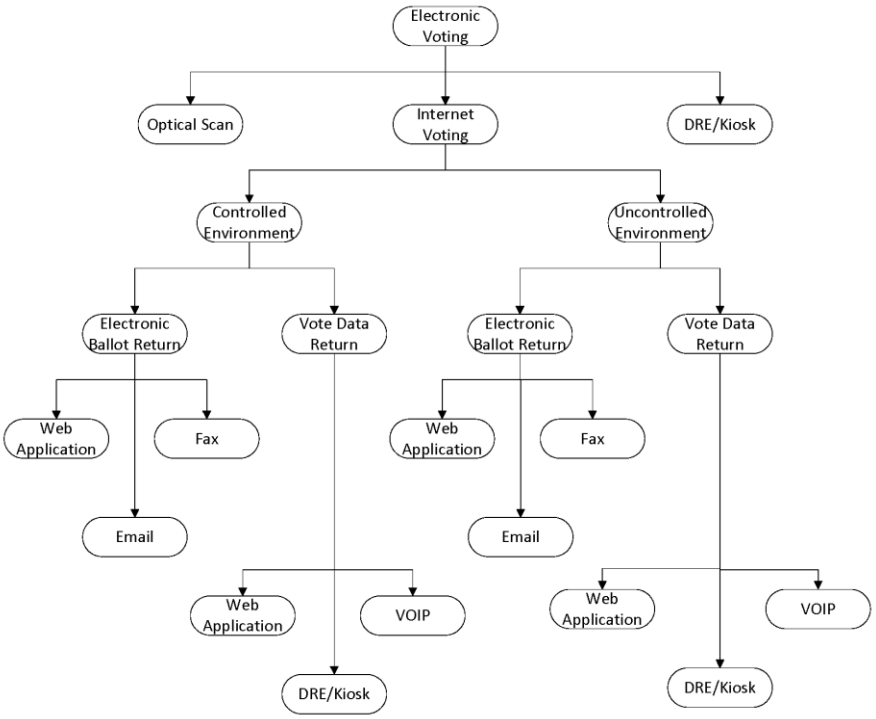

Figure 1: internet voting classifies

The advantages of implementing a voting system over the Internetare : [2]

- **Increasing voter turnout:** By helping older, disabled or sick people or any one cannot easily wait in a large queue and waste time.
- **Cost reduction:** Traditional voting (paper base) it cost too much money starting from preparing the ballot to the announcement of election results.
- **Decrease of invalid votes:** There is no way to doinvalid votes in an online voting because there is no other choice but only one provided.

In this paper, we have designed a prototype of a remote Internet voting system is desgined to match with the security and performance metrics.

The rest of this paper is organized as follows: In Section (2), related works have reviewed. In section (3), the security requirements are discussed. Section (4) presents the performance requirements for online voting. Section (5) describes in details the proposed model. Section (6), security evaluation presented. Section (7) investigates the performance evaluation and tested results. Finally, a conclusion will follow in Section (8).

2. Related Works

This section investigates two systems : the Egyptian E-Voting protocol and the Geneva E-voting.

First for the egyptian E-Voting [3], the authors discovered an Electronic Voting System in Egypt (EVSE) scheme. This scheme is designed to fit in the environment and conditions of Egypt, trying to solve problems in the old system (conventional system).

This system offers a certain degree of flexibility and convenience to the voter to ensure a maximum contribution in the democratic process. If the voter registered for voting in a constituency, e.g.ALX but the voter works in another, e.g.Agouza, then he/she can vote in the Agouza polling station near his/her workplace. However, he/she will only have access to the Ballot Server of ALX to participate in the local election of his/her constituency as shown in figure 2.

Exposed a new electronic voting protocol based on the bit operation XOR and the use of blind signatures. Specifically it is an algorithm designed expressly for the circumstances in which is necessary to choose between two candidates or two options.

It is shown that the proposed algorithm satisfies the important requirements of any E-Voting scheme: anonymity, completeness, correctness and uniqueness.

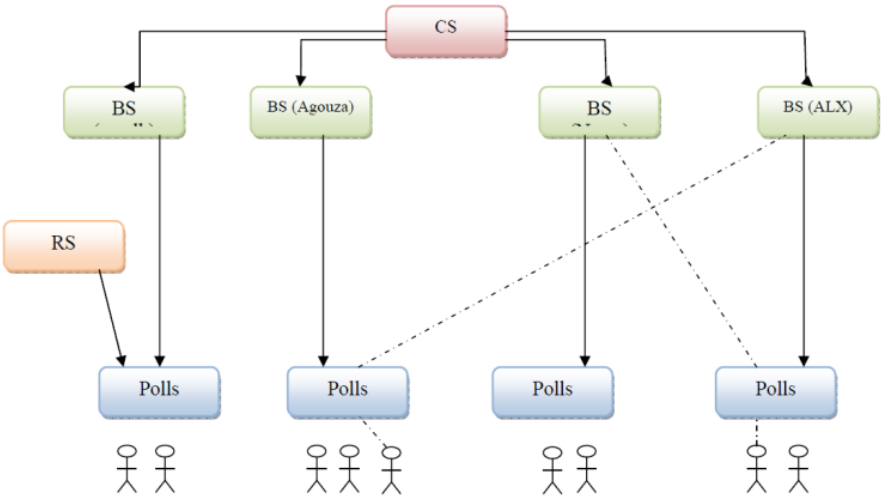

Figure 2: Egyptian E-Voting system hierarchal

Second for the Republic and Canton of Geneva, he decided to develop a remote e-Voting system and reach voters in their homes because there is a real need to entice over-solicited citizens to vote. Moreover, the granularity, locality and distribution of voting enable authorities to establish a new system incrementally. Geneva has introduced voting through the Internet in a controlled manner. [4]

The Geneva project proposal was not simply taken from a technical point of view, but they adopted a multidisciplinary approach. They involved the University commissioning legal and social-political studies and asked private companies to develop some parts of the

application, test his security, attempt to penetrate the system and analyze it. The Geneva Government is the owner of its online voting application.

3. Security Requirements

Many researches on electronic voting have conducted a comprehensive list of security requirements for electronic voting the following requirements are: [5, 6, 7]

- **Privacy**: All voters in an election should be confident that their individual choices will remain hidden.
- **Completeness**: all valid votes should countcorrectly.
- **Soundness**: no one can interrupt the voting.
- **Unreusability**: all voters vote only one.
- **Eligibility**: Only eligible voters can take part in voting, and every voter can cast only one vote.
- **Fairness**: No one can infer partial results before the ballotis closed.
- **Uniqueness**: No voter should be able to vote more than once.
- **Accuracy**: Voting systems should record the votes correctly. All valid votes have counted.
- **Integrity**: Votes should not be able to be modified without detection.
- **Verifiability**: It Shouldbe possible to verify that votes have correctly counted for in the final tally.
- **Auditability**: There should be reliable and demonstrably authentic election records.
- **Reliability**: Systems should work robustly, even in the face of numerous failures.
- **Secrecy**: No one should be able to determine how any individual voted.
- **Non-coercibility**: Voters should not be able to prove how they voted.
- **Flexibility**: Equipment should allow for a variety of the ballot question formats.
- **Integrity**: After casting, votes cannot be altered, deleted, or substituted.
- **Secrecy**: No one can tell how a particular voter or any possible subgroup of voters actually voted.
- **Anonymity**: No one can tell who actually voted.
- **Receipt-Freeness**: a voter does not gain any information (areceipt) which can be used to prove to a coercer that voter voted in a certain way.
- **Robustness**: A small set of broken, unavailable, or corrupt system components or a small group of conspiring parties (election authorities, system administrators, voters, external attackers, etc.) Cannotdisrupt the election process or compromise correctness or privacy.
- **Coercion-Resistance**: a voter cannot cooperate with a coercer to prove to him that she voted in a certain way.

We can classify those requirements as the basicsecurity requirements and any electronic voting system must meet these.

When information is particularlysensitive or vulnerable, using a password alone may not be enough protection. A stronger means of authentication, something that is harder to compromise is necessary. One of the strongest authentication methods is Two-factor

authentication also called strong authentication that is used to identify and authenticate users.

3.1 Identification and Authentication

Identification and authentication (I&A) are the processes that can be used to identify and verify the voter on the system. In the multi-user system, the voter must identify himself / herself, and then the system will authenticate the identity before using the system. Therefore, the identification and authentication processes successfully through the following three traditional ways:

- Something knows: password
- Something has: a smart card or token.
- Biometric feature: fingerprint

There are several systems for dealing with two way-factor authentication. They may differ in delivering the password to the authorized user or a differententity based on the security constraints.

3.2 Mobile-ID

Mobile Id offers a strong two way authentication by authenticating the user to the service and service to the user. The mobile id works as such a way that the user has required to send the code generated by the application after which the Mobile id generates a code to identify the user with the service. [8]

4. Performance Requirements

The remote Internet voting systems need to have a quick response time. If voters become frustrated with request processing time, they will abandon the system, perhaps before they had a chance to vote. It is important to know where potential bottlenecks may reside whether with the servers, network, or applications and to be able to handle peak traffic loads without having to over-allocate resources, which can be costly and inefficient.

We have proposed a generic model of HTTP traffic at the user session level. So one of the scheduling algorithms that provide good performance and high-throughput is First-Come-First-Serve (FCFS).

We present the performance metrics for each request could affect the server:

- **System throughput**: The requests processed by voting server at the rate. The system throughput denoted X.
- **Waiting tasks**: the average number of tasks is waiting to execute. The denoted waiting task is W, completion task is C, and overall tasks are A. So the W = A / C.
- **Response time**: In real-time systems, the response time of the task is the average time between executing the task and finishing tasks. The response time denoted R.
- **Probability of request loss**: it is the average incoming request would be discarding because the system reaches the maximum number of requests. This happened when the system goes under overloading cause. The requestloss denoted L.

Therefore, we assume that the incoming requests arrive at the e-voting web server randomly (arrival time), the most important feature is the service time of the operation is μ The capacity of the queue is M, and service time for completing tasks is Y.

Figure 3 shows the generic queuing model for remote Internet voting.

The proposed performance model can implement on different queuing systems such as multi-server queue and multiple single-server queues.

We have applied some changes in order to fast turnaround environments. First, we have modeled two session classes: First class is User Authentication and Authorization (UAA) class that is responsible for identifying the voter after logging and check if voters had voted or not. Second Keep Alive Turnover (KAT) class that responsible for:

1. Check servers' capabilities for new arrival tasks and assign tasks to an available server.
2. If the task's execution time exceeded, KAT redirectstask to another available server.

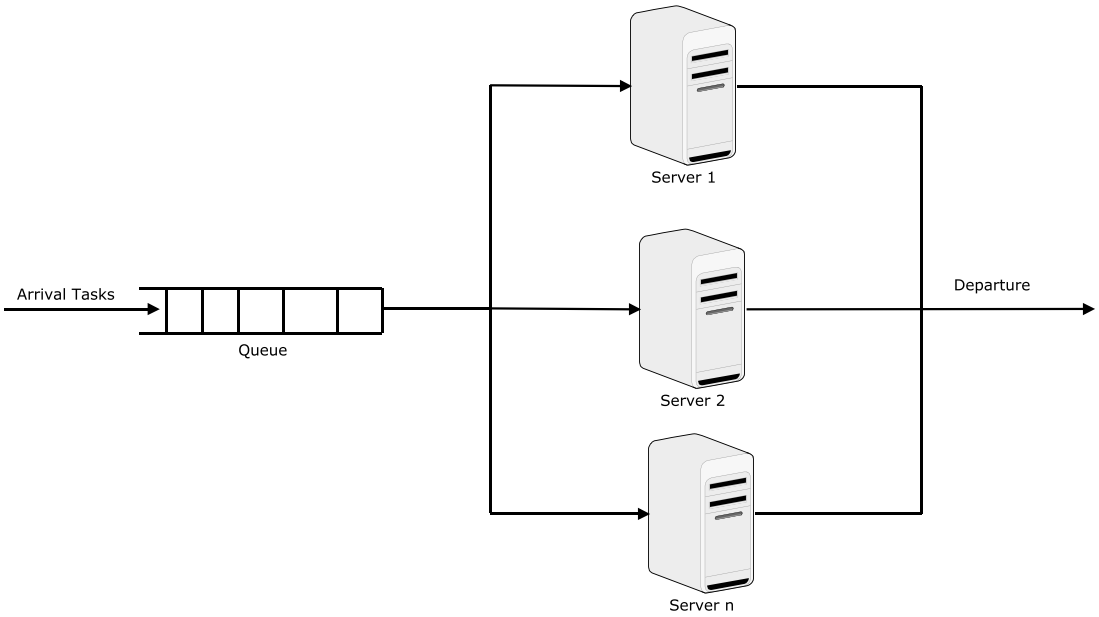

Figure 3: Evaluating Multi-Server queuemodel of remoteInternet voting

The main purpose for Keep Alive Turnover (KAT) class is to reducethe workload over the system, there are some requirements on both user and server sides. In general, we believea user with low bandwidth connection (such as a modem) and avoid multiple parallelcontinualconnection from one browser to the same server. On the server side, the server should setthe minimum timeout value to prevent high workload.

According to the famous network rule (8 Seconds-Rule), it is a way of determining the adequateresponse timeof aweb serverthrough differentbandwidthconnections. It specified that if the load-time of a web page exceeds eight seconds, users are unlikely to waitfor its completion. In order to increase the response of anonline system, faster ways to deliver the content to the user needed to be devised.[9]

We have set time out (T_O) for each request to be (T_x-seconds), T_x-seconds have been set for voter's connection methods (taking into account all connection methods for high speed and low speed connections).

5. The Proposed remote Internet Voting System

While a lot of actions have been taken to enhance voting protocols in order to make them more secure, and especially for remote voting system it does not require security enhancements but, in other have it requires performance enhancement as well.

In this section, we describe the overall architecture and detailedprotocol steps of the remote voting system.

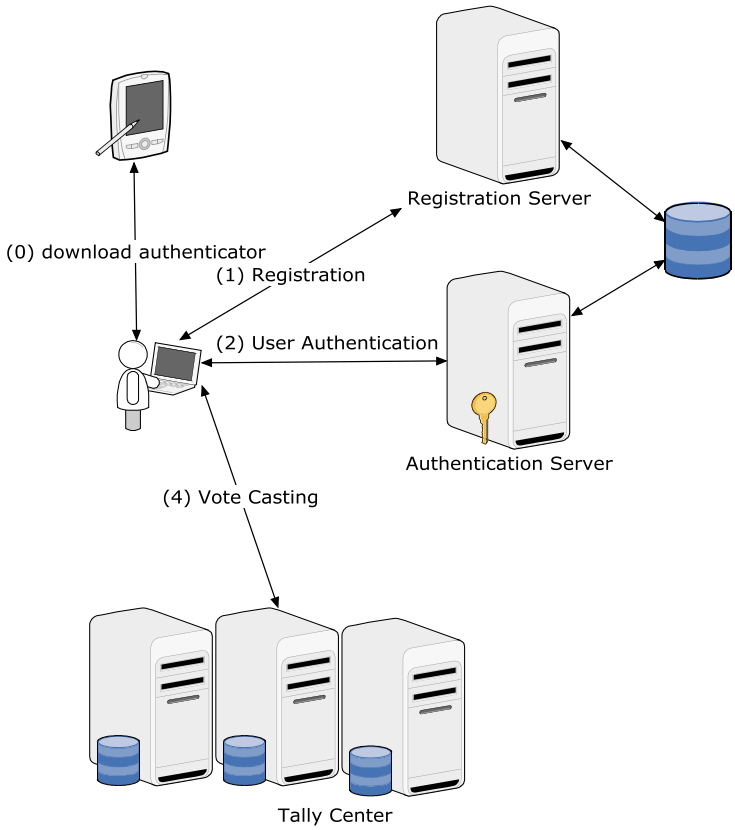

Figure 4: Generic E-Voting architecture

As shown in figure 4 the main basic entities for remote internet voting system: voter, authentication server, registration server, and tallying center.

Proposed model consists of three phases:

- **Registration**
- **Voting**
- **Counting**

5.1 Registration Phase

As shows in figure 5, the voter (V_i) downloads his mobile authentication software on his mobile device, hence mobile device act as e-Token or smart ID that contains the voter'sprivate and public keys.

Then voter accesses the registration server (RS) to download registration form and input his information in secure communication channel using SSL layer.

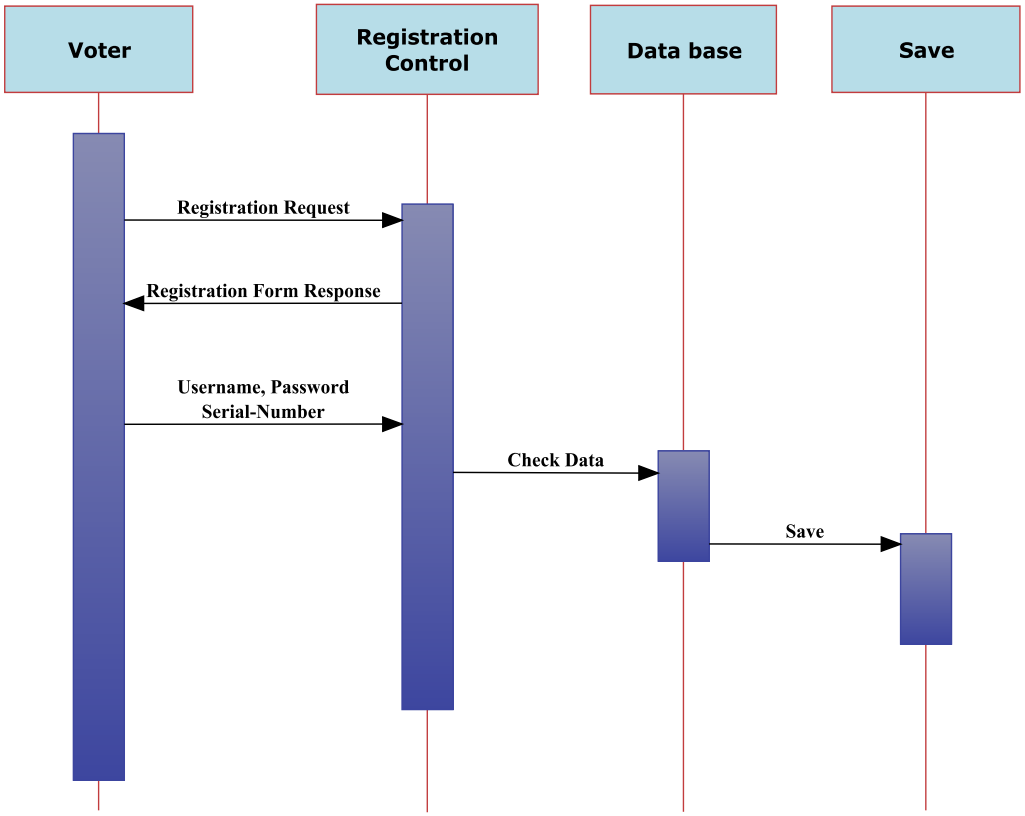

Figure 5: Sequence diagram for registration phase

5.2 Voting Phase

As figure 6 display the sequence of voting phase:

1- Voter provides authentication data (username, password). Authentication server checks either the voter has voted or not. If voters had already voted, AS rejects authorization. Otherwise, AS gives voter right to castthe vote.

Before casting vote, there is a verification step, it is about to prove the voter's identity to prevent voting on behalf. Voter's public and private keys downloaded to the voter's mobile device K- and K, where K is the public key and K- is a private key.

2- After voter providing his / her identity, the voter selects vote and encrypts the vote with the voter's private key. $X_i = K^-(v_i)$, where $X_i$has randomly chosenfactor.
3- AS verifies the signature s_i of messagee_i. If s_i is valid, then AS signe_i as $d_i = S_A(e_i)$ and sends d_i to Ballot.

At the end of voting the authentication server (AS) announces the number of voters receiving AS's signature and publish a list as {ID, e_i, s_i}

4- Tally center (TC) checks the signature of posting messages by using a voter public key.

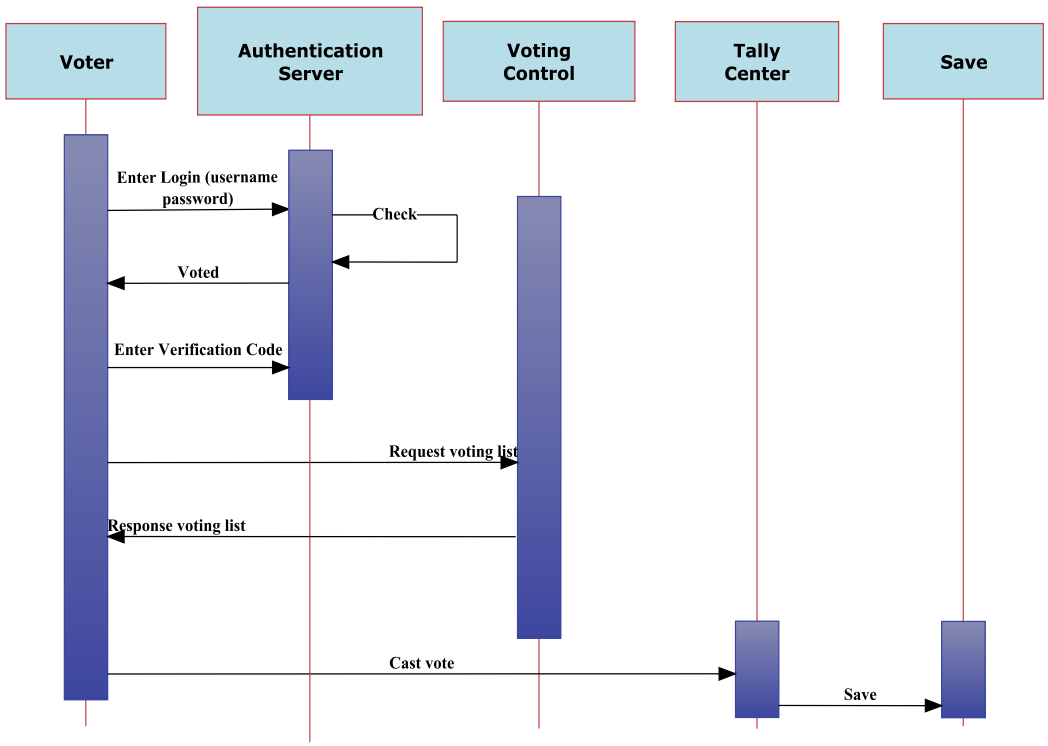

Figure 6: Sequence diagram for voting phase

5.3 Counting Phase

As it displays in figure 7, tally center (TC) verifies the signature of authentication server (AS) d_i if verification fails; the tally center claims that d_i is not a valid signature. If it not fails; tally center decryptsvote and retrieve {ID, e_i} without revealing voter identity.

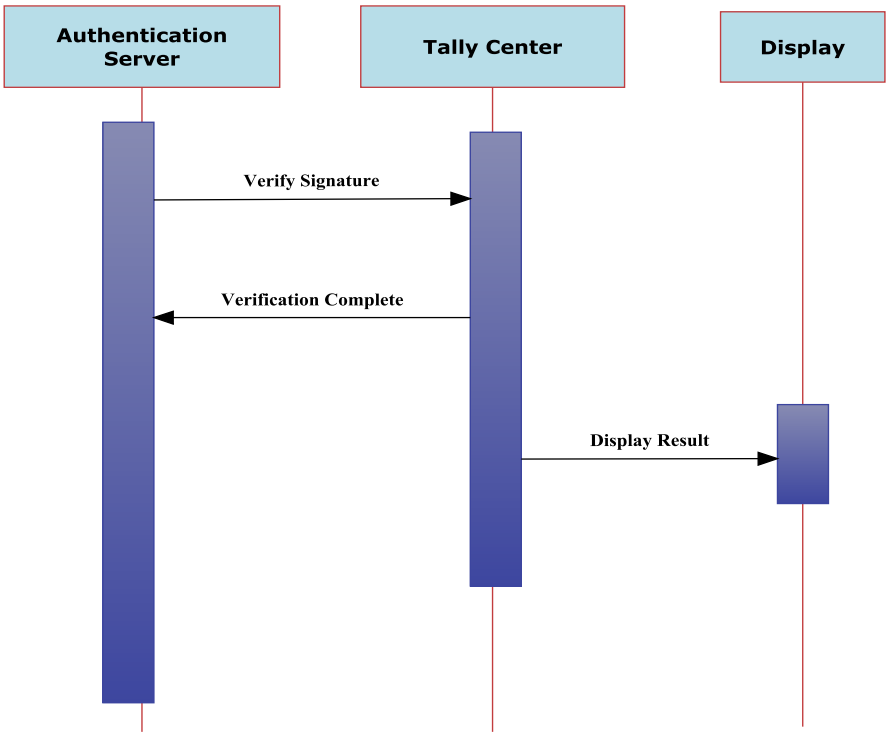

Figure 7: Sequence diagram for voting phase

5.4 Completely Automatic Public Turing Test to Tell Computers and Humans Apart "CAPTCHA"

One of the most attacks on the web is man-in-the-middle (MiTM) this attack interception and retransmission of messages in a way that the original parties will suppose that their communicationhas secured. [10, 11]

According to Dimitris Mitropoulos and DiomidisSpinellis, they have proposed model to prevent treating to electronic voting by brute force attack also called dictionary attack and search engine bots [11]. Figure 6 shows a sample of CAPTCHA that uses to prevent MITM attack.

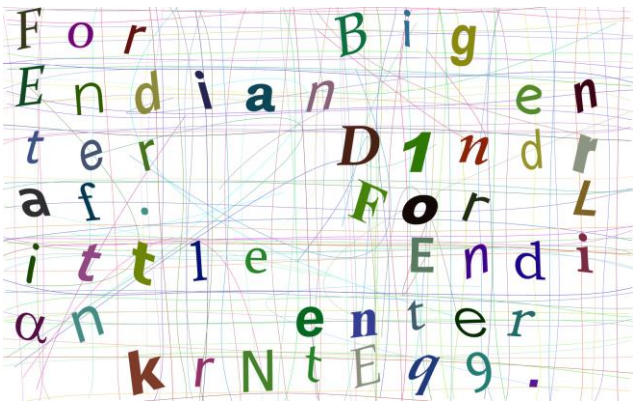

Figure 8: An example of CAPTCHA

They have proposed e-voting scheme that securesthe integrity of the voter's vote from MITM. That during elections, a test can utilize to determine whether a votehascastedby human or malicious software. They have used Transaction Anonymity Number (TAN) instead of voter anonymity.

6. Security Evaluation

When evaluating an Internetvoting platform, it is important to evaluate the security risks.

The following most common Internet voting risks are: [12]

- **Unauthorized voters' casting votes**: ineligible voters could try to cast a vote.
- **Voter impersonation**: voter could try to cast a vote on behalf another person.
- **Voter privacy compromise**: an attacker could break the voter privacy, identifying the voter's voting options, and breaking the vote secrecy.
- **Authentication methods**: one important issue in Internet voting is how voters prove their identities in a remote way.

According to those risks, the proposed model has been tested to evaluate Internet voting risks:

- Voters tried to vote repeatedly.
- Unregistered voters tried to vote.
- Bad passwords were used.
- Bad CAPTCHA were used.
- Votes were lost by the counter.
- Duplicate votes were given to the counter.
- Prevent votes on behalf

7. Performance Evaluation

The goal of the algorithm is to deliveridealquality of security while maintaining high performance for tasks running on the voting system. To achieve the goal, the proposed model manages to reducethe degree of security insufficiency of each task without performance deterioration. [13]

Model algorithm, which integrating security requirements into scheduling for performance enhancement with security. For task T_i, the earliest start time on site $M_j$is es_j (T_i), which can be computed as

$$es_j(T_i) = r_j + \sum_{T_i \in W_i} \left(e_1^j + \sum_{K=1}^{q} c_{1j}^k (s_1^k) \right) \quad (1)$$

Where r_jRepresents the remaining overall execution time of a task currently running on the j^{th} site.

And $e_1^j + \sum_{K=1}^{q} c_{1j}^k (s_1^k)$,it is the overall execution time (security overhead is factored in) of waiting task T_l assigned to site M_j prior to the arrival of T_i. Hence, the earliest start time of T_i is a sum of the remaining overall execution time of the running task and the overall execution times of tasks with earlier arrival on site M_j. Therefore, the earliest completion

time for task Ti on site Mjcalculate as:

$$ec_j(T_i) = ec_j(T_i) + e_i^j + \sum_{k=1}^{q} c_{ij}^k(s_i^k) = r_i + \sum_{\text{Ti}\in\text{Wi}} [e_i^j + \sum_{k=1}^{q} c_{ij}^k(s_i^k)] + e_i^j + \sum_{k=1}^{q} c_{ij}^k(s_i^k) \quad (2)$$

Next are the algorithm steps:

1. For each task, T_i submitted to the queue schedule
2. For each site M_j in the system
3. Use Equation (1) to computees_j (T_i), the earliest start time of T_i on site M_j;
4. Use Equation (2) to computeec_j (T_i), the earliest completion time of T_i on site M_j
5. End for
6. Sort all sites in earliest completion time in a non-decreasing order

8. Results Discussion

The rate at which voters arrive at voting centers has a direct impact on overall system performance. Hence, a heavy arrival rate in a voting center may require more voting stations in order to complete the voting process in a timely manner. [14]

We focus on different parameters that used to evaluate Internet-voting performance under workload. We describe the results in two situations, first in normal condition, which measure performance characteristics of the system. Second is the overloading or high workload over a server that checks the system's ability to respond to an excessive load.

By simulating the HTTP, requests have generated by hundreds or thousands of users who can access voting servers (authentication server and tallying center) and simulate a high workload.

8.1 Normal condition monitoring

We expect during election time the arrival rate is changing; we may expect low arrival rate at morning and above medium at night, but we expect high incoming at midday.

Table 1 represents the system performance in normal condition.

Table 1: Model observation under normal condition

User No.	Clicks	Hits	Errors	Avg. Click Time [ms]	Bytes	Kbit/s
1	59	58	0	238	220,052	374.17
2	55	55	0	540	474,100	376.40
3	58	57	0	295	267,742	380.08
4	58	57	0	294	266,817	377.03
5	58	57	0	295	268,242	376.74
6	58	57	0	296	269,240	376.93
7	58	57	0	238	216,258	377.93
8	55	54	0	540	465,480	377.86
9	57	57	0	294	266,665	378.55
10	57	56	0	720	761,815	476.79

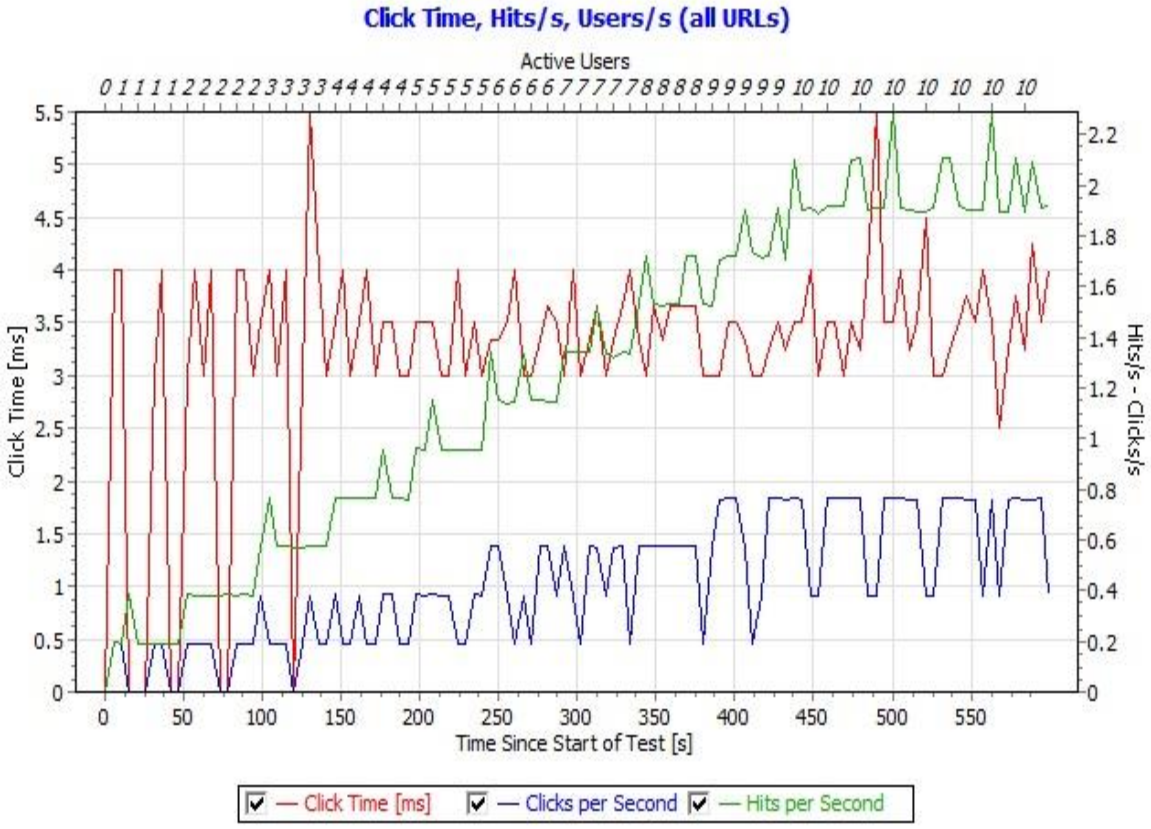

Figure 9: Click Times, Hits/s and Clicks/s

Figure 9 shows the average time a user waited for his request to be processed (Including redirects, images, frames, etc., if enabled), the hits per second and the users per clicks.

We can see that with 10users, the two lines for "clicks per second" (blue) and"hits per second" (green) differ more and more. The reason is that hits includerequests that produce errors, but clicks are only calculated from the requests that were successful.

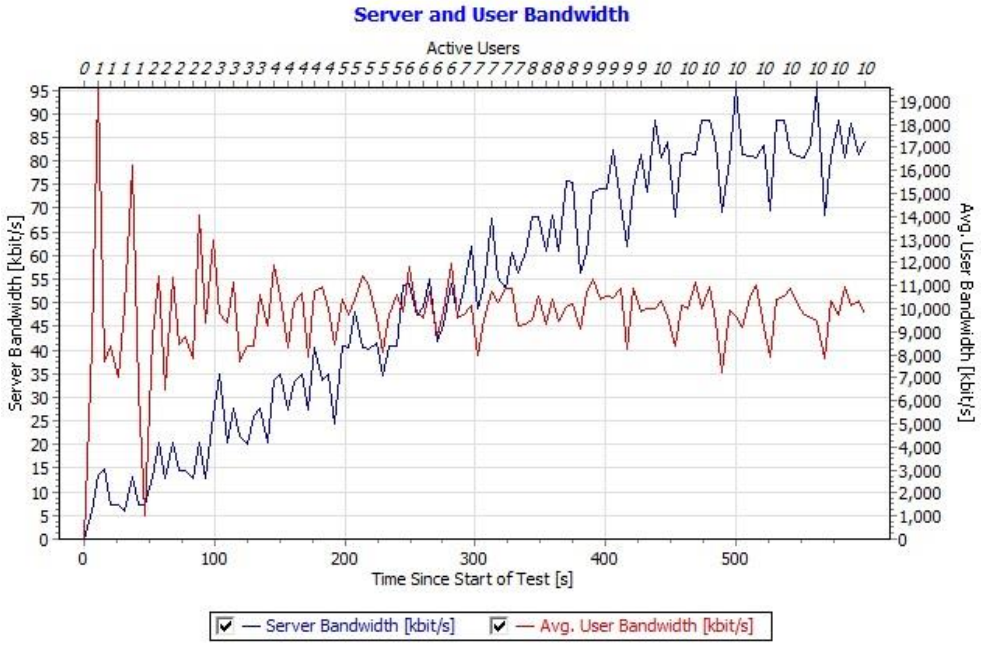

Figure 10: Server and User Bandwidth

Figure 10 displays the bandwidth the server was able to deliver (as a total) as well as the average bandwidth that was experienced by the simulated users.

In this graph we can see that the average bandwidth available per user goes upfrom 500Kbit to 19,000Kbit when the number of users climbs from 1 to 10 users.

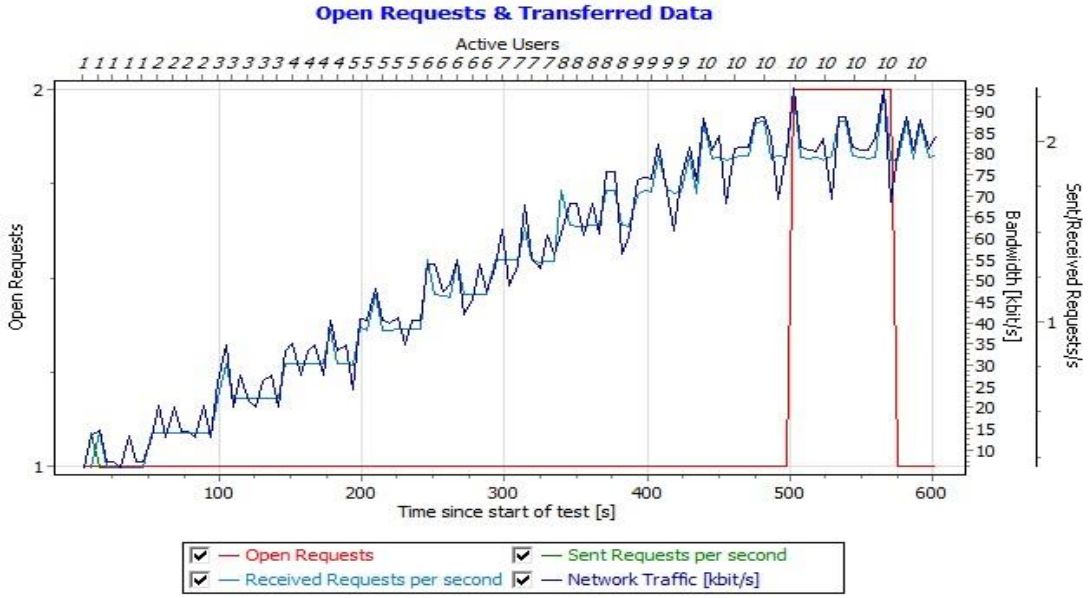

Figure 11: Open Requestes& Transferred Data

Figure 11 shows the number of open requests as well as the number of sent and received requests in comparison with the network traffic. The network traffic goes high according to user network bandwidth as we can see in user 10 the open request (red) because of his / her bandwith.

8.2 Heavy workload condition monitoring

Overloading or server works under high workload it affects the server performance, for that we have tested server. Table 2 displays server conditions under high workload, and the response to users.

Table 2: Model observation under high workload

Users	Clicks	Hits	Errors	Avg.Click Time (ms)	Bytes	Kbit/s
1	3,160	3,160	0	481	19,977,258	9,398.56
2	3,155	3,155	0	483	19,945,448	9,982.91
3	3,155	3,155	0	482	19,945,448	10,249.86
4	3,152	3,152	0	483	19,926,484	10,515.74
5	3,156	3,156	0	483	19,950,128	10,187.72
6	3,152	3,152	0	484	19,926,484	10,205.34
7	3,160	3,160	0	485	19,977,258	10,340.80
8	3,162	3,162	0	483	19,988,056	10,616.73
9	3,159	3,159	0	482	19,969,092	10,528.52
10	3,158	3,158	0	481	19,964,412	10,350.55

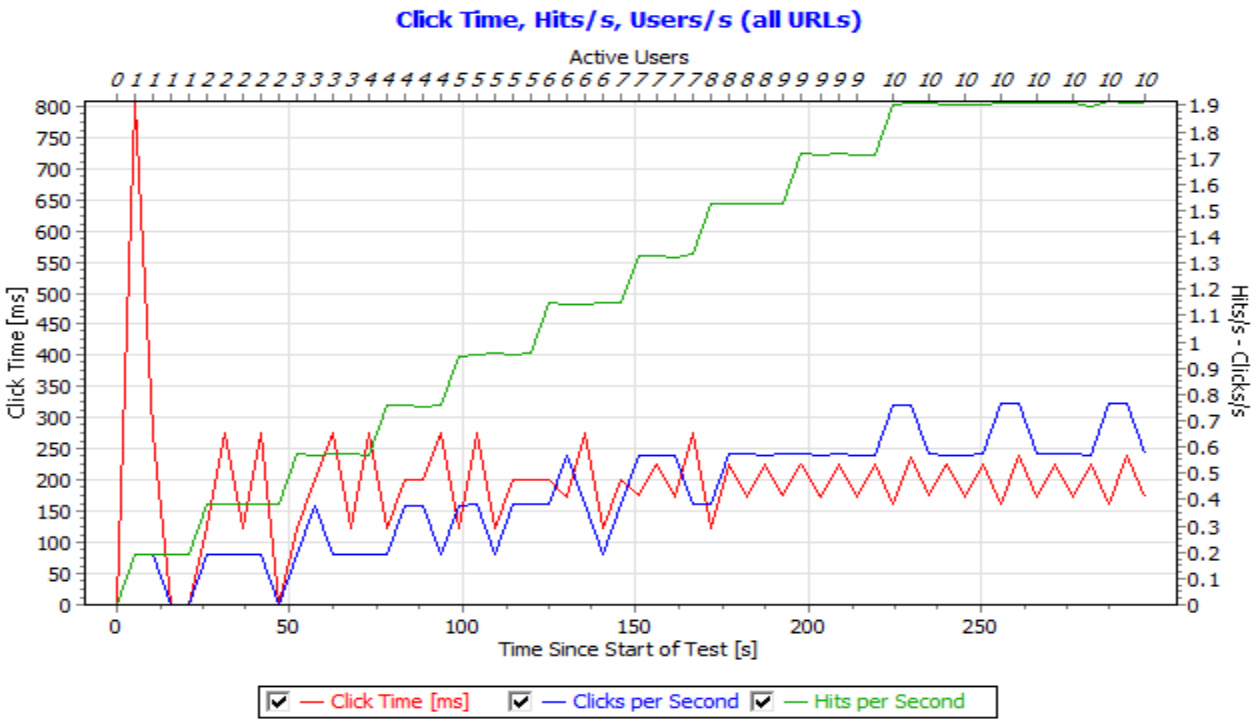

Figure 12: Heavy Workload Click Times, Hits/s and Clicks/s

Figure 12 shows the average time a user waited for his request to be processed (Including redirects, images, frames etc., if enabled), the hits per second and the users per clicks.

We can see that with 10 users the two lines for "clicks per second" (blue) and"hits per second" (green) differ more and more. The reason is that the system reachs limits of its resources and still can handle incoming requests.

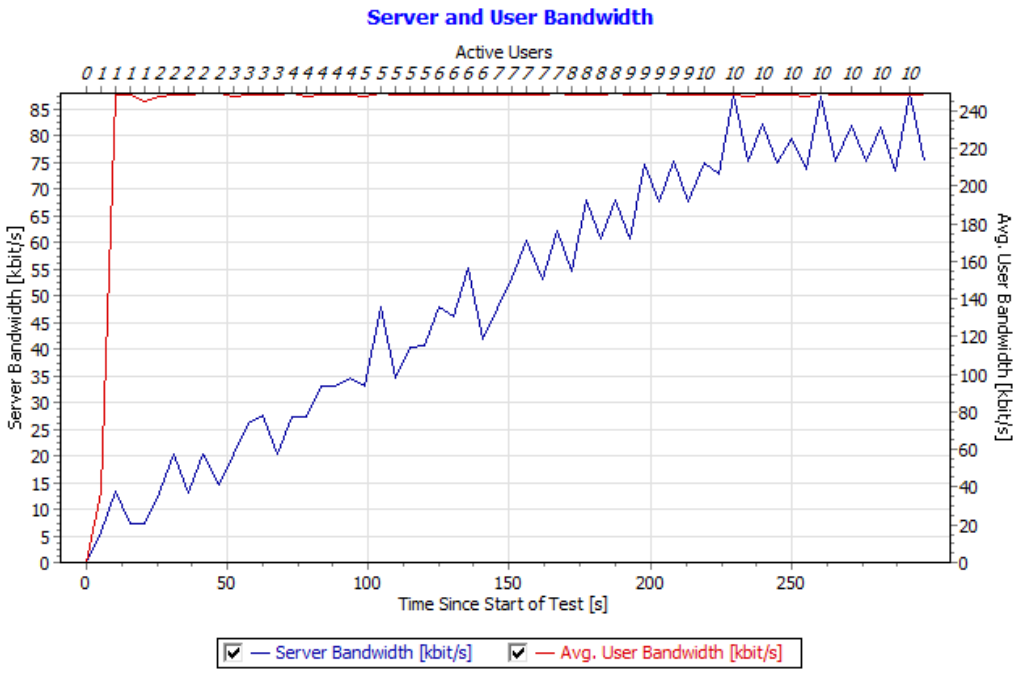

Figure 13: Heavy Workload Server and User Bandwidth

In figure 13 we can see that the average bandwidth available per user goes up from 10 Kbit to 240 Kbit when the number of users climbs from 1 to 10 users.

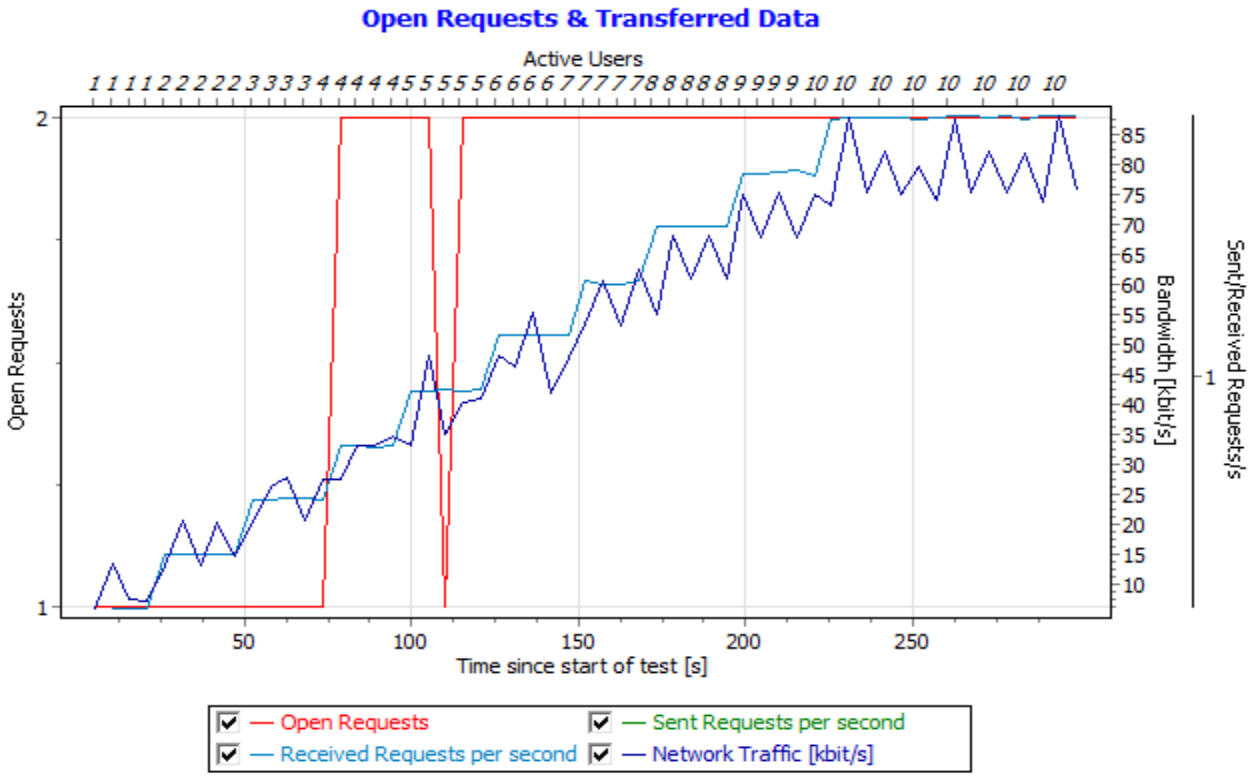

Figure 14: Heavy Workload Open Requestes& Transferred Data

In figure 14, it displays that the open requests per user going up and use high bandwidth because the system reaches the limit of resources and all users have been under high network bandwidth.

The main purpose of normal and heavy workloads tests is:

For normal workload isto measure the performance characteristics of system. Performance test will help system to find answers for the following questions:

- How many users can work with the system concurrently receiving the acceptable quality of service?
- How many voters system can handle during election time?
- What response time is typical for your system under load?

And as for heavy workload or in otherwords it also knows as stress workload is to check the ability of system to respond to an excessive load. Every system has a performance limit. However, even when that limit is reached, the system should meet the "correct overload criteria":

- It should not crash or stop responding completely.
- The number of correct responses per second should not decline significantly.
- It should either provide a user friendly overload message for the requests that it cannot serve, or delay all responses equally.
- After a period of overload it should return to the normal operation without performance degradation.

9. Conclusion

In this paper, we have proposed a generic model of remote internet voting systemwhich has many advantages over thetraditional voting system (paper-based). Some of these advantages are lesser cost, faster tabulation of results, improved accessibility, greater accuracy, and lower risk of human and mechanical errors.

The challenge is that the remote Internet voting system must be able to handle the traffic, or else it will be as useless. System availability is the key to having a successful remote voting system.

Availability includes the presence of ample bandwidth and sufficient server processing capacity to handle steady traffic loads as well as large peaks that might occur during different times of the day.

We presented a solution that uses mobile-ID to guarantee both identification and authentication voters. We alsopresented a performance evaluation of the proposed model and performance results obtained with real data; we have tested data under normal and heavy workload conditions.

We have shown how to reduce the heavy workload on the system in case of the system has insufficient resources and response time take longer to execute.

In order to manage security and performance metrics, we have designed two classes: one for handling security requirements and the other class for handling traffic workload.

References

[1] A Survey of Internet Voting - The US Election Assistance Commission, September 14, 2011.

[2] Schryen, G. (2003). “E-Democracy: Internet Voting In Proceedings of the IADIS”, International Conference WWW/Internet 2003, Algarve, Portugal, 5-8 November 2003.

[3] M. Abo-Rizka, and H. Ghounaim, (2007) ” A Novel in E-voting in Egypt”, IJCSNS International Journal of Computer Science and Network Security, VOL.7, No.11.

[4] Michel Chevallier, Michel Warynski and Alain Sandoz, State Chancery,(2006). “Success Factors of Geneva’s e-Voting System,”, Republic and Canton of Geneva, Switzerland.

[5] Kwangjo Kim, Jinho Kim, Byoungcheon Lee, and Gookwhan “Experimental Design of worldwide internet voting system using PKI”.

[6] J. C. Benaloh and D. Tunistra. (1991). “Receipt-free secret ballot elections”, proc. Of 26th ACM STOC, pp.544 553.

[7] V. Niemi and A.Renvall (1994).“ how to prevent buying of voters in computer election”, advances on in cryptology-asiacrypt’94, LNCS vol.917, pp.164 170, springer-Verlag.

[8] Harish Dinne&KarthikMandava.(2010)“ Two Way Mobile Authentication Systems”,.

[9] Zona Research. (2008)'The Need for Speed II".

[10] Dimitris Mitropoulos and DiomidisSpinellis. (2009). "Securing e-voting against MITM attacks". 13th panhellenic conference on informatics, corfu, Greece, September 2009.

[11] L. V. Ahn, M. Blum, and J. Langford. (2003). "CAPTCHA: Using hard ai problems for security," in In Proceedings of Eurocrypt. Springer Verlag, pp. 294–311.

[12] JordiPuiggalí, JesúsChóliz, Sandra Guasch.(2009) "Best Practices in Internet Voting".ScytlSecure Electronic VotingTuset 20, 1-7, 08006 Barcelona, Spain.

[13] Tao Xie, Xiao Qin. (2003). "Performance Evaluation of a New Scheduling Algorithm for Distributed Systems with Security Heterogeneity", National Security Agency.

[14] Mohammad Malkawi, Mohammed Khasawneh, Omar Al-Jarrah. (2009). "Modeling and Simulation of a Robust e-Voting System".Communications of the IBIMA. VOL.8