DISTRIBUTED DATABASE MODEL FOR MOBILE E-POLLING SYSTEM

At present, the two ways of vote casting are Paper- Ballot system and Electronic Voting Machine (EVM). Both these methods do not ensure integrity, secrecy, accuracy, authentication and flexibility [MCH2007]. We have proposed a network efficient, cost effective multilayer peer to peer distributed model for E-Polling System which resolves the above mentioned problems in an effective way.

5.1 INTRODUCTION

Various models proposed for electronic voting system [TKO2004] can be categorized under two broad names [TSD2007]. One is electronic voting (e-Voting) which is less complex but voters are limited to polling stations only [ASG2008]. Second is voting on internet allowing a voter to cast his vote without being physically present at polling station but it is more complex and needs higher security [MHA2005][SAN2005]. Our proposed architecture combines the merits of both these voting systems architectures and reduces the consumption of resources like disk space, communication cost and traffic bottlenecks etc.

R. Mercuri in [RME2002] addressed the concepts of untraceable electronic mail and digital pseudonyms which can apply for electronic voting for anonymity. In recent years, voting equipments which were widely adopted may be divided into five types [TOM2003]:

- **Paper-based Voting**: The voter gets a blank ballot and uses a pen or a marker to indicate the candidate he wants to vote for [SPE2008].

- **Lever Voting Machine**: Lever machine is peculiar equipment and each lever is assigned for a corresponding candidate. The voter pulls the lever to poll for his favourite candidate. But its interface is not user-friendly.

- **Direct Recording Electronic Voting Machine**: Also abbreviated as DRE, integrates with keyboard, touch screen or buttons for the voter to cast his vote. Some of them lay in voting records and counting the votes is very quickly [SMN1995].
Chapter 5: Distributed Database Model for Mobile E-Polling System

- **Punch Card**: The voter uses metallic hole-punch to punch a hole on the blank ballot. It can count votes automatically but if the voter’s perforation is incomplete, the result is probably determined wrongfully [MDB2007].

- **Optical Voting Machine**: After each voter fills a circle correspond to their favorite candidate on the blank ballot, this machine selects the darkest mark on each ballot for the vote, then computes the total result.

AccuVote-TS’s [COM2003] is a Diebold Election System which includes touch screen, card reader, keyboard, head phone and paper tape printer. The voter selects his favorite candidate on touch screen [DCA2007] and the vote will be printed on the paper tape. But all the electoral information (including identity authentication, audit or counting of votes) are stored in Microsoft Access database without setting password, so there is high risk of attack. IVotronic [AMK2005] provides multi-language and uses flash memory to save voting records. Electoral workers use PEB (Personal Electronic Ballot, a device which is similar to disk) to start polling machine up. When the election is finished, the workers use PEB to access voting records in the polling machine. Because the PEB’s password is only three characters, the risk of password breaking exists.

In eSlate 3000 [HAR2004], voter gets a Personal Identity Number (PIN) as four digits from electoral workers, then goes to the booth to input the PIN into polling machine to login. Each terminal connects to the server which is named JBC (Judges Booth Controller). During counting of votes, the information is sent to JBC from every terminal by network which saves it in MBB (Mobile Ballot Box). This system does not encrypt voting data, so there are some risks of data security. Furthermore, the electoral functions are not protected with password, anyone, even the voter, can finish the election.

SAVIOC is an open source E-voting system and all the source code and software can be downloaded from its official website [CAG2005]. SAVIOC’s advantages are its simple disposition and low cost but on the other hand, it is not easy to use. Jung et al. [JYL2009] implemented an E-voting model using the concept of contactless IC cards. The system with contactless IC card can not only make sure voter’s identity but also ensures the validity of IC card. Foreign experience [HEB2001] [BBB2003] revealed that they are often confronted by security issues while the electronic voting system is running. Security threats are not only from outsider (such as voters and attackers) but also from insider (such as system developers and administrators).
5.2 PROPOSED MODEL

We have a multilayer system consisting of peer nodes arranged logically in a hierarchical fashion as shown in Figure 5.1 providing the system its scalability and communication efficiency. Top layer contains the main database comprising of information pertaining to voters and candidates who will compete in the elections. At the time of elections, a copy of these databases is divided into fragments and transferred to the state servers at layer 2 which are linked together through a dedicated communication channel resulting in a virtual or ‘hybrid’ peer to peer system where each state server maintains the data related to that particular state.

State servers further fragment the copies of their databases and send them to district servers at layer 4. Now these district servers send all data to the cluster heads of the clusters connected to them at layer 5. These cluster heads then transfer data to all the clients active in their clusters. An intermediate layer (layer 3) is maintained between state level and district level layer i.e. state cache server layer which maintains two index tables:
First index table contains information about location of the databases maintained at each state. Second index table contains the information about the location of database of each district which comes under that state.

5.3 WORKING OF PROPOSED VOTING SYSTEM

Each layer in the proposed system as shown in Figure 5.1 maintains its own ID. Each user has 15-digit Hexadecimal ID in the similar fashion as proposed by Rohit et al. [RVA2009]. In this format, first 2-digit represent a country code, next 2-digit for state, next 2-digit for district, next 2-digit for village and last 7-digits for representing the user’s personal ID as shown in Table 5.1. The format enables to identify uniquely a total population of 268435456 in a village or in a district. Thus, by using this assumption more than $10^{12}$ million users can be uniquely identified which is quite a good assumption to manage a whole world population. This unique ID is made available to each layer in the system.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1-256</td>
<td>1-256</td>
<td>1-256</td>
<td>1-256</td>
<td>1-268435456</td>
<td>MAXIMUM RANGE</td>
</tr>
<tr>
<td>IA</td>
<td>C4</td>
<td>D3</td>
<td>B6</td>
<td>90A54D3</td>
<td>HEXADECIMAL ID</td>
</tr>
<tr>
<td>8 BIT</td>
<td>8 BIT</td>
<td>8 BIT</td>
<td>8 BIT</td>
<td>28 BIT</td>
<td>SIZE OF ID</td>
</tr>
</tbody>
</table>

To get the unique 15-digits hexadecimal ID, every user has to fill online form at district level or village level. After submission of the form, unique hexadecimal ID is generated and this information is sent to main database. User can register or update his/her profile anytime. A numbers of mobile booths increase the participation of users as well as reduce the complexity and security issues. User can cast votes by logging in at fixed polling booths or at specified mobile booths only.

To cast a vote, user has to click either the candidate’s name or the sign of candidate’s political party. After this, a mark is placed against his/her unique ID in the database at the local layer. This will prevent the same user to cast his/her vote twice. After a fixed period of time, the modifications made to local database are sent to main database and the databases situated at the intermediate layers in the form of packets. Time period of sending packets from different servers at the lowest layer is maintained in such a way that no traffic congestion occurs in the network. After the voting process is completed, results
can be displayed in no time by counting the total number of marks placed against the ID of users.

5.4 CONCLUSION

On the basis of hypothesis, following conclusion can be deduced:

- The search time for data at the state server database is of the complexity $O[1]$ as compared to those with $O[n]$ complexity where $n$ is number of servers, which require the databases to be searched serially, by maintaining index tables at the intermediate cache servers providing one-hop path to the desired data thus saving the search time and enhancing throughput.

- Let $PT$ is the Processing Time of one transaction to update main database after fixed intervals, then.

  $PT = UT + AT + WT$  \hspace{1cm} (1)

Where $UT$ is time taken to update database at topmost layer and $AT$ is time taken to send acknowledgment back to the lowest layer. In our hypothesis, $WT$ (waiting time for each transaction) is 0 as each server does its transaction on its turn only after fixed interval. So processing time will remain constant. We consider one another constant factor $ND$ (Network Delay) which depends upon the execution environment in which proposed model works. Finally, Total Processing Time for $m^{th}$ transaction as $TPT_{(m)}$.

  $TPT_{(m)} = ND + PT$  \hspace{1cm} (2)

As $ND$ and $PT$ are constant, so the total processing time will remain constant as the time goes on.