Editor in Chief: Ezendu Ariwa

- Visiting Professor, Gulf University, Bahrain
- Visiting Professor, University of Lagos, Nigeria
- Visiting Professor, Kano State Polytechnic, Nigeria
- Chair, IEEE Consumer Electronics Chapter, UK&RI
- Chair, IEEE Broadcast Technology Chapter, UK&RI
- London Metropolitan Business School
- London Metropolitan University
- United Kingdom

Disclaimer

It is our editorial policy to accommodate broad diversity of viewpoints on various issues of the scope of journal. Nevertheless, any views expressed in this publication are the views of the authors and not of GreenField Advanced Research Publishing House.
# INDEX

<table>
<thead>
<tr>
<th>1.</th>
<th>MULTI SINK SCHEDULING SCHEME FOR WIRELESS SENSOR NETWORKS</th>
<th>1-8</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>U. Hari Haran</td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>TOWARDS MAKING WORLD-CLASS UNIVERSITIES: CASE STUDY OF THE ROLE OF INFORMATION AND COMMUNICATION TECHNOLOGY</td>
<td>9-25</td>
</tr>
<tr>
<td></td>
<td>Christian A. Bolu, Adewole Adewumi, Ken Egbo</td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>PROPOSED A METHOD BASED ON FUZZY LOGIC TO DETECT THE DRIVER'S VOICE COMMANDS AND APPLIED TO AUDIO SYSTEM IN A NOISY ENVIRONMENT</td>
<td>26-34</td>
</tr>
<tr>
<td></td>
<td>Navid Samimi Behbahan, Zohreh Mousavinabas</td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>A FRAMEWORK OF TRANSACTION MANAGEMENT IN DISTRIBUTED DATABASE SYSTEM ENVIRONMENT</td>
<td>35-53</td>
</tr>
<tr>
<td></td>
<td>Dr. Farid Ahmad</td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>BULLWHIP EFFECT REDUCTION THROUGH MULTI-AGENT SYSTEM USING JADE TOOL</td>
<td>54-68</td>
</tr>
<tr>
<td></td>
<td>Sandeep Garg, Dr. S. Srinivasan</td>
<td></td>
</tr>
</tbody>
</table>
MULTI SINK SCHEDULING SCHEME FOR WIRELESS SENSOR NETWORKS

U. Hari Haran*

Abstract: In wireless sensor networks WSN increasing the network lifetime, where the information delay caused by moving the sink should be bounded. Some of the combinational complexity of this problem, most previous proposals focus on heuristics and provable optimal algorithms remain unknown. By build a unified framework for analyzing this joint sink mobility, routing, delay, and induced sub problems and present efficient solutions for them. We generalize these solutions and propose a polynomial-time optimal algorithm for the origin problem. Furthermore, we developing with multiple sink for the network and study the effects of different trajectories of the sink and provide important insights for designing mobility schemes in real-world mobile WNNs.

Index Terms—Wireless sensor networks, delay-constrained mobility, network lifetime

*Final Year M.E-CSE, Selvam College of Technology, Namakkal, Tamilnadu, India.
1. INTRODUCTION

In the past decades, wireless sensor network (WSN), one of the fastest growing research areas, has been attracted a lot of research activities. Due to the maturity of embedded computing and wireless communication techniques, significant progress has been made. Typically, a WSN consists of a data collection unit (also known as sink or base station) and a large number of sensors that can sense and monitor the physical world, and thus it is able to provide rich interactions between a network and its surrounding physical environment in a real-time manner.

The capacity-limited power sources of small sensors constrain us from fully benefitting from WSNs. Due to the unique many-to-one (converge-cast) traffic patterns, the traffic of the whole network will be converged to a specific set of sensor nodes (e.g., neighbouring nodes of the sink) and results in the hotspot problem. Much research effort has been dedicated to resolve this issue, for example, energy efficient communication protocols, multi-sink systems. However, as long as the sink and sensor nodes are static, this issue cannot be fully tackled. Therefore, there is a recent trend to exploit mobility of the sink as a promising approach to the hotspot problem.

By the way of using sink mobility, we can classify them into two categories: random mobility based and controlled mobility based. For the first category, the sink is designed to move randomly within the network. For example, Rahul et al. presented an architecture on which mobile entities (named MULEs) pick up data from sensors when in close range in sparse sensor networks. Schemes based on random mobility are straightforward and easy to implement. However, they suffer from shortcomings like uncontrolled behaviours and poor performance. Hence, recent research resorts to controlled mobility to improve the performance.

For the controlled mobility, the key problem is to deterministically schedule the sink to travel around the network to collect data. It is shown that by properly setting the trajectory even limited mobility would significantly improve the network lifetime. However, the mobility also brings new issue, i.e., the delay of the data delivery caused by the movement of the sink. Some previous proposals tried to avoid this issue by considering the so-called fast mobility, whereas the speed of the sink is sufficiently high so that the resulting delay
can be tolerated. While others address this delay-bounded mobility problem by heuristics with little theoretical understanding.

To this end, we study the delay-bounded sink mobility problem (DeSM) of WSNs in this paper. We assume that WSNs are deployed to monitor the surrounding environment and the data generation rate of sensors can be estimated accurately. We constrain the mobile sink to a set of sink sites. First, we propose a unified framework that covers most of the joint sink mobility, data routing, and delay issue strategies. Based on this framework, we develop a mathematical formulation that is general and captures different issues. However, this formulation is a mixed integer nonlinear programming (MINLP) problem and is time consuming to solve directly. Therefore, instead of tackling the MINLP directly, we first discuss several induced sub problems, for example, sub problems with zero/infinite delay bound or connected sink sites (sink sites are connected if for any two sites there exists a path that connects them and each edge of that path meets the delay constraint). We show that these sub problems are tractable and present optimal algorithms for them. Then, we generalize these solutions and propose a polynomial-time optimal approach for the origin DeSM problem. We show the benefits of involving a mobile sink and the impact of network parameters (e.g., the number of sensors, the delay bound, and so on.) on the network lifetime. Furthermore, we study the effects of different trajectories of the sink and provide important insights for designing mobility schemes in real-world mobile WSNs.

Our main contributions are the following:

1. We provide a unified formulation of DeSM, which is general and practical. We discuss sub problems of DeSM and offer efficient algorithms for them to guide the design of our algorithm for the origin DeSM.
2. We generalize algorithms for sub problems and present an optimal algorithm with polynomial complexity for the DeSM.
3. We study the effects of different trajectories of the sink and provide important insights via extensive simulations.

2. RELATED WORK

Mobility management is one of the most important issues in wireless networks, and it has
received extensive research efforts in different areas of wireless networks such as mobile ad hoc network (MANET), wireless mesh network, vehicular ad hoc network. Recently, there is a trend to investigate mobility as a means of relieving traffic burden and enhancing energy efficiency in WSN. We can classify sink mobility into two categories: random mobility and controlled mobility. Sinks in the first category move randomly within the network. Schemes based on random mobility are easy to implement, but they suffer from shortcomings like uncontrolled behaviors and poor performance. Recent research tends to use controlled mobility to improve the performance. The hardcore is to jointly schedule different issues (e.g., sink mobility, data routing, information delay, and so on.) to optimize the network lifetime.

For this paradigm, Gandham et al. first challenged this problem and proposed a heuristic algorithm. Wang et al. relaxed the problem by doing the sink scheduling and data routing separately, and their proposed routing scheme can work only in a grid network topology. Recently, Shi and Hou developed the first algorithm with performance guarantee with a single sink. Liang et al. extended Shi’s work by considering issues like multiple sinks and the maximum number of hops from each sensor to a sink. A three-stage heuristics has been developed to find high-quality trajectory for each sink as well as the actual sojourn time at each sojourn location. In our recent research, we proposed a generalized column generation-based algorithm that can be applied to a set of sink mobility problems with near-optimal performance.

In above proposals, they assume that sinks are high-speed so that information delay caused by moving the sink can be ignored. However, on the one hand, mobile sinks in physical worlds usually have limited speed. On the other hand, underlay applications like the real-time surveillance demand a delay upper bound. Therefore, it is natural to take the delay issue into consideration.

Keung et al. studied the message delivery capacity problem in delay-constrained mobile sensor networks where the sink nodes are static while sensor nodes are mobile. They focused on maximizing the percentage of sensing messages that can be successfully delivered to sink nodes within a given time constraint. Their network model is fundamentally different with ours and is somehow similar to the DTN.
In our previous study, we also addressed the problem of lifetime maximization with delay bound in a mobile WSN. The major improvements of this paper over the previous one are twofold. First, we present some new theoretical results like the connectivity analysis of a WSN. Second, we design a set of new simulations to study the effects of different trajectories of the sink and provide important insights for designing mobility schemes in real-world mobile WSNs.

3 DeSM PROBLEM

Fig. 1 shows reference architecture for a WSN with a mobile sink (i.e., $s_0$). Sensor nodes, which are stationary, keep monitoring the surrounding environment and generating data. A mobile sink is used to gather sensed data by traveling around the network. We assume that only at certain locations, the sink can communicate with the outside network and then deliver cached data to users. For example, due to interference and security issues, for a sensor network deployed in the battle field for the surveillance mission, it is reasonable that the sink can connect with the headquarters only at certain locations using wireless techniques like WiMAX or LTE. These locations are represented by squares in the figure. The sink has a maximum speed $V_{\max}$ (in m/s). We assume that while the sink is moving, sensors will buffer their newly generated data, as in. Only when the sink stays at one of sink sites, sensors will start transmitting data to the sink through multi-hop routing. This could potentially cause a high delay for data packets. Here, we define the delay of data as following,

**Definition 1** (Delay of data).

The delay of data is defined as the time spent by the mobile sink moving from one sink site to the next sink site.
To limit such delay, a delay bound is set according to the underlay applications. Moreover, as pointed out in the previous study, whenever a sink has been relocated to a new site, it will take some time to rebuild the routes of sensors. Thus, we set $\epsilon$ as the minimum residual time of any sink site.

### 4 EXTENDED SSDR (E-SSDR) ALGORITHM FOR DESM

#### 4.1 E-SSDR Algorithm

To solve the origin DeSM problem, we prove the following conclusion:

For an instance of DeSM, if its sink site graph $G^0$ is not connected, we can divide $G^0$ into connected subgraphs, each of which can be solved optimally by the SSDR. The overall optimal solution for this instance is the same solution of the sub graph with the longest network lifetime.

The proof is based on contradiction. Assume that for an instance of DeSM, we have an optimal solution which involves two sites from two different sub graphs. This means that we find a sink path including these two sites that meets the delay constraint. Thus, these two sites are connected and should be in the same sub graph.

![Fig. 2. Probability of the full connection](image)

we propose an E-SSDR approach to solve the origin DeSM optimally:

**Step 1.** Divide $G^0$ into connected sub graphs.

**Step 2.** Apply the SSDR approach to each sub graph and obtains the optimal sink path as well as corresponding routes.

**Step 3.** Choose the solution of the sub graph with the longest network lifetime as output.

### 5 NUMERICAL RESULTS

In this part, we evaluate the proposed algorithms using three typical trajectories of the sink,
namely:

1. Linear trajectory. This case simulates that the sink travels along one predefined path, for example, a vehicle carrying a sink moves along the only path across the forest to gather sensed data daily.

2. Boundary trajectory. Luo and Hubaux suggested that it is the most efficient way to collect data in a dense network.

3. Arbitrary trajectory. In this case, we have little control over the distribution of sink sites, for example, in a battle field. Due to page limit, we prepare a supplement file, available online, for the simulation results the arbitrary trajectory.

6 CONCLUSION AND FUTURE WORK

We proposed a unified framework to analyze the sink mobility problem in WSNs with delay constraint. We presented a mathematical formulation that jointly considers different issues such as sink scheduling, data routing, bounded delay, and so on. The formulation is general and can be extended. However, this formulation is a MINLP and is time consuming to solve directly. Therefore, we discussed several induced subproblems and developed corresponding optimal algorithms. Then, we generalized these solutions and proposed a polynomial-time optimal approach for the origin problem. We show the benefits of involving a mobile sink and the impact of network parameters (e.g., the number of sensors, the delay bound, and so on.) on the network lifetime. Furthermore, we study the effects of different trajectories of the sink and provide important insights for designing mobility schemes in real-world mobile WSNs.

As for the future work, we plan on extending current work to accommodate networks with multiple sinks. Furthermore, using the centralized optimal algorithm developed in this paper as performance benchmark, we want to design distributed online algorithms for fast execution in large-scale networks and test them in real-world experiments.

7 REFERENCES


Towards Making World-Class Universities: Case Study of the Role of Information and Communication Technology

Christian A. Bolu*
Adewole Adewumi**
Ken Egbo***

Abstract: Characteristics of a world-class university include indicators such as quality of faculty, research reputation, talented undergraduate, international presence, proper usage of resources, alliances and networks, embrace of many disciplines, technologically smart, good management practices, internationalism of all aspects of the university. Thus, economic activity, innovation, international diversity, institutional indicators and research indicators are manifestations of a world-class university. In today’s digitally connected world, it is impossible to attain this status without a world-class Information and Communication Technology (ICT) infrastructure. This paper presents impact of ICT activities on universities ranking of three Nigerian universities thus enhancing their quest towards world-class status.

Key words: World-Class, Ranking, ICT, e-Learning, Repository, Nigeria

*University of Nigeria, Nsukka,
**Covenant University, Ota, Nigeria
***Federal University-Oye-Ekiti, Nigeria
1.0 INTRODUCTION

As Nigeria aspires to be one of the top twenty world economies by year 2020, she must have significant educational aspirations. Among these are the quests to raise existing universities to “World Class” stature or to establish “World Class” universities. Characteristics of a world-class university include indicators such as quality of faculty, research reputation, talented undergraduate, international presence, proper usage of resources, alliances and networks, embrace of many disciplines, technologically smart, practice the art of good management, internationalism of all aspects of the university.

Thus, economic activity and innovation, international diversity, institutional indicators, research indicators are manifestations of a world-class university. In today’s digitally connected world, it is impossible to attain this status without a world-class Information and Communication Technology infrastructure. This paper presents impact of information and communication technology activities in three Nigerian universities and the improvement of the universities’ Webometrics world universities ranking thus enhancing the universities’ quest towards world-class status. The three universities studied were:

- University of Nigeria, Nsukka – established in 1960, is one of the oldest public universities with about 40,000 students in four (4) campuses. It has 15 Faculties and over 120 academic departments.
- Federal University Oye-Ekiti – established in 2011, is one of the newest public universities with about 1000 students in 2 campuses. It has four (4) Faculties and 27 departments.
- Covenant University, Ota – established in 2002, is one of the leading private universities in Nigeria with about 9,000 students in one campus. It has nine (9) Schools (or Faculties) and 22 departments.

2.0 WORLD UNIVERSITIES RANKING METHODOLOGIES

University ranking has become a major source of providing indicators and parameter for classifying university performance in teaching and research and therefore identifying world class universities. There are several ranking methodologies:

- Times Higher Education (THE)
- Academic Ranking of World Universities (ARWU)
- Webometrics World Universities Ranking (WR)
2.1 The **UK Times Higher Education** methodology for the 2013-2014 World University Rankings is identical to that used since 2011-2012, offering a year-on-year comparison based on true performance rather than methodological change. The 13 performance indicators are grouped into five areas (THE, 2013):

- Teaching: the learning environment (worth 30 per cent of the overall ranking score)
- Research: volume, income and reputation (worth 30 per cent)
- Citations: research influence (worth 30 per cent)
- Industry income: innovation (worth 2.5 per cent)
- International outlook: staff, students and research (worth 7.5 per cent).

2.2 **Academic Ranking of World Universities (ARWU)**

The Academic Ranking of World Universities is compiled by Shanghai Jiaotong University. The rankings have been conducted since 2003 and then updated annually and uses six indicators:

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Indicator</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quality of Education</td>
<td>Alumni of an institution winning Nobel Prizes and Fields Medals</td>
<td>10%</td>
</tr>
<tr>
<td>Quality of Faculty</td>
<td>Staff of an institution winning Nobel Prizes and Fields Medals</td>
<td>20%</td>
</tr>
<tr>
<td></td>
<td>Highly cited researchers in 21 broad subject categories</td>
<td>20%</td>
</tr>
<tr>
<td>Research Output</td>
<td>Papers published in Nature and Science</td>
<td>20%</td>
</tr>
<tr>
<td></td>
<td>Papers indexed in Science Citation Index-expanded and Social Science Citation Index</td>
<td>20%</td>
</tr>
<tr>
<td>Per Capita Performance</td>
<td>Per capita academic performance of an institution</td>
<td>10%</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>100%</td>
</tr>
</tbody>
</table>

2.3 **Webometrics Ranking Web of Universities**

(Aguillo, 2009), in his paper attempts to provide an alternative, although complementary, system for the evaluation of the scholarly activities of academic organizations, scholars and researchers, based on web indicators. He found that three large groups of indicators are feasible to obtain and relevant for evaluation purposes: activity (web publication); impact (visibility) and usage (visits and visitors). It observed that ranking results are similar to those obtained by other bibliometric-based rankings; and there is a concerning digital divide.
between North American and European universities, which appear in lower positions when compared with their USA and Canada counterparts. (Aguillo, 2006), described the testing of the feasibility of cybermetric indicators for describing and ranking university activities as shown in their Web sites, using a large set of 9,330 institutions worldwide in 2006 rising to over 20,000 in July 2013 edition.

Specically, (Aguillo et al, 2008), presented the Webometric Ranking of Universities using a combined indicator called WR that takes into account the number of published web pages (S) (twenty five percent), the number of rich files, those in pdf, ps, doc and ppt format (R) (12.5 percent), the number of articles gathered from the Google Scholar Database (Sc) (12.5 percent,) and the total number of external inlinks (V) (fifty percent). They suggest that this kind of rankings using web indicators should be used to measure universities’ performance in conjunction with more traditional academic indicators. From July 2010 edition the four indicator names were changed as in Table 2 and the definition of the indicators from the July 2010 edition are shown below.

Table 2: Webometrics Web Ranking Indicator Names

<table>
<thead>
<tr>
<th>No</th>
<th>Before July 2010</th>
<th>From July 2010 Edition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(S) Web Size (20%)</td>
<td>Presence (20%)</td>
</tr>
<tr>
<td>2</td>
<td>(V) Visibility (50%)</td>
<td>Impact (50%)</td>
</tr>
<tr>
<td>3</td>
<td>(Sc) Google Scholar (15%)</td>
<td>Openness (15%)</td>
</tr>
<tr>
<td>4</td>
<td>(R) Rich Files (15%)</td>
<td>Excellence (15%)</td>
</tr>
</tbody>
</table>

**Presence** (20%): The global volume of contents published on the university webdomains as indexed by the largest commercial search engine (Google). It counts every webpage, including all the formats recognized individually by Google, both static and dynamic pages.

**Impact** (50%): The quality of the contents is evaluated through a "virtual referendum", counting all the external in-links that the University web-domain receives from third parties. Those links are recognising the institutional prestige, the academic performance, the value of the information, and the usefulness of the services as introduced in the web pages according to the criteria of millions of web editors from all over the world. The link visibility data is collected from the two most important providers of this information: Majestic SEO and ahrefs, that provides an overlapping scenario very close to a true global coverage.
Openness (15%): The global effort to set up institutional research repositories is explicitly recognized in this indicator that takes into account the number of rich files (pdf, doc, docx, ppt) published in dedicated websites according to the academic search engine Google Scholar.

Excellence (15%): The academic papers published in high impact international journals are playing a very important role in the ranking of Universities. The data is largely provided by the Scimago group.

In a very interesting work, (Ortega et al, 2009), present visual display of the most important universities showing the topological characteristics and describes the web relationships among universities of different countries and continents. Examining the link relationships of the first 1000 higher education institutions using social network analysis techniques found out that the world-class university network is constituted from national sub-networks that merge in a central core where the principal universities of each country pull their networks toward international link relationships. The United States dominates the world network, and within Europe the British and the German sub-networks stand out.

A comparison between ARWU and Webometrics methodologies shows that they differ largely on prestige indicators which are largely subjective.

<table>
<thead>
<tr>
<th>Table 3: Comparison of the main World Universities' Rankings, 2010</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CRITERIA</strong></td>
</tr>
<tr>
<td>Univ's Analyzed</td>
</tr>
<tr>
<td>Univ's Ranked</td>
</tr>
<tr>
<td>Quality of Education</td>
</tr>
<tr>
<td>Internationalization</td>
</tr>
<tr>
<td>Size</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Research Output</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Prestige</td>
</tr>
</tbody>
</table>

2.4 Other international universities rankings are:
- Asia’s Best Universities (Asia Week)
- CHE-Excellence Ranking (CHE)
- Global University City Index
• Performance Ranking of Scientific Papers for World Universities (Higher Education Evaluation and Accreditation Council of Taiwan)
• World University Rankings (THES & QS)

3.0 ICT STRATEGY AND ACTION PLAN

We present here the ICT activities resulting from the improved infrastructure of the three universities studied. We discuss the ICT Policy and Strategy Programmes, Virtual Learning Environment, Institutional Repositories, Integrated Business processes and Internationalisation through Global classroom. We begin with the scope of ICT infrastructure development in the three universities.

3.1 ICT Infrastructure

a. University of Nigeria: Working with Google Inc, the University drew out a very detailed ICT Strategy Programme. The goal, objectives and scope of the University of Nigeria ICT strategy Programme are designed to transform the University to a world-class university in the shortest possible time. Specific actions for university ranking are detailed in Appendix A. The scope of the ICT strategy programme includes but not limited to the following:

<table>
<thead>
<tr>
<th>Area</th>
<th>Infrastructure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internet Connectivity</td>
<td>Provision of a minimum of 310Mbps (2-STM(^1)) of Internet Bandwidth starting with 155 Mbps (1-STM).</td>
</tr>
<tr>
<td>University Network</td>
<td>A comprehensive, ubiquitous, always-on, wired and wireless network that covers the entire geography of the four campuses of the university.</td>
</tr>
<tr>
<td>Central Storage</td>
<td>A storage area network of 400TB (Terabytes) on which students and faculty can store data, but starting with 100 TB.</td>
</tr>
<tr>
<td>Data Centre</td>
<td>Tier 2 Data Centre and Network Operating Centre consisting of Servers, cooling system, automatic fire system, surveillance and sophisticated hybrid power backup system of Solar, Inverters, UPS, generators and public power supply.</td>
</tr>
<tr>
<td>Power Supply</td>
<td>Dedicated Power Supply of the order of 1MW preferably from renewable power source such as solar.</td>
</tr>
<tr>
<td>Virtual Super Computing</td>
<td>Virtual Super Computing facility will be required that can be used for modeling, rendering of bioinformatics and computer intensive research as well as e-Business.</td>
</tr>
<tr>
<td>Pedagogy and</td>
<td>Use of Open source software to drive a new digital environment for</td>
</tr>
</tbody>
</table>

\(^1\) STM-1 refers to a transmission format used in fiber optic networks. STM-1 is an abbreviation for Synchronous Transport Module level-1. It has a bit rate of 155.52 Mbit/s.
Administration teaching, learning and research.

Student and Staff Computing

Each Staff and student will be encouraged to have a computer (preferably a mobile computer) of a minimum configuration of 2GB RAM at least 250 GB HDD which is internet-ready.

Integrated Local Intercom/Voice Service

In locations selected by the University, provision of a packet based voice service (VoIP).

Call Centre and Hardware Repair Centre

Establishment of a call centre to manage communication between teams and users and a hardware repair centre of Laptops, Notebooks and PCs

ICT Resource Centre

Comprehensive ICT Resource Centre consisting of Webinar rooms, networking rooms, software rooms, hardware rooms for teaching; global classrooms, e-learning laboratories, international certification centres for learning; software and hardware testing laboratories, data analysis centre, imaging and printing room, document management centre, videoconferencing rooms for research; infrastructure rooms for data storage, network operating centre, maintenance and repair shop, call centre and charging bay

b. Covenant University: Working with Google Inc and the University taskforce on University Ranking, they drew out a very detailed ICT Strategy Programme. Covenant University, an ICT-driven university, with estimated active ICT users of about 10,000 will require the following minimum ICT infrastructure:

<table>
<thead>
<tr>
<th>Last Mile</th>
<th>Optical fibre cabling from the nearest point of presence of one or two Internet Service Providers such as Glo Communication Networks Ltd or Main One Cables Ltd.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internet Bandwidth</td>
<td>Bandwidth density is 10-15 Mbps per 1000 active users (students and staff). The recommended bandwidth density for 10,000 active users is therefore 1 STM (155 Mbps). Terrestrial bandwidth will be preferred.</td>
</tr>
<tr>
<td>University Network</td>
<td>A comprehensive, ubiquitous, always on wired and wireless network that covers the entire geography of Covenant University.</td>
</tr>
<tr>
<td>Central Storage</td>
<td>A storage area network of at least 400-600 TB (Terabytes) on which students and faculty can store data is recommended.</td>
</tr>
<tr>
<td>Data Resource Centre</td>
<td>Comprehensive ICT Resource Centre consisting of Webinar rooms, networking rooms, software rooms, hardware rooms for teaching; global classrooms, e-learning laboratories, international certification centres for learning; software and hardware testing laboratories, data analysis centre, imaging and printing room, document management centre, videoconferencing rooms for research; infrastructure rooms for data storage, maintenance and repair shop, call centre.</td>
</tr>
<tr>
<td>Dedicated Power Supply</td>
<td>Dedicated Power supply of the order of 1MW preferably from renewable power source such as Solar.</td>
</tr>
<tr>
<td>High Performance Computing</td>
<td>High Performance Computer facility will be required that can be used for modelling, rendering, bioinformatics and computer intensive research.</td>
</tr>
</tbody>
</table>
To provide the information robustness required as the University works towards becoming a world-class university, the information management must be run on an integrated enterprise solution that integrates the University business processes in procurement, operations, planning, project management, student lifecycle management, customer service, asset management, financial accounting, human resources, and analytics application.

Call Centre
A 24x7 call centre to manage communication between teams and users. The call centre will be equipped with the latest state-of-the-art communications facilities.

c. Federal University Oye-Ekiti: The founding management resolving to make the university an ICT-driven university developed an ICT Policy with an event-driven ICT implementation plan. The ICT Policy covers the following areas:

- Application of ICT in Education
- Application of ICT in Administration
- Infrastructure
- Network development and management
- Access management and control
- Capacity Building
- Equal Opportunities Guidelines
- Maintenance of ICT Facilities
- Collaborative Services and Resource Sharing
- World universities Ranking

For the implementation process, an event-driven approach was adopted. There are a total of 128 activities or milestones to be completed during the 36 months life of the Policy (FUOYE, 2012). Overview of the ICT Policy implementation plan is shown in Appendix B.

The implementation work breakdown structure is grouped into 12 quarters as follows:

<table>
<thead>
<tr>
<th>Quarters</th>
<th>Period</th>
<th>No of Milestones</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q4-2012</td>
<td>October-December 2012</td>
<td>17</td>
</tr>
<tr>
<td>Q1-2013</td>
<td>January-March 2013</td>
<td>27</td>
</tr>
<tr>
<td>Q2-2013</td>
<td>April-June 2013</td>
<td>19</td>
</tr>
<tr>
<td>Q3-2013</td>
<td>July-September 2013</td>
<td>18</td>
</tr>
<tr>
<td>Q4-2013</td>
<td>October-December 2013</td>
<td>13</td>
</tr>
<tr>
<td>Q1-2014</td>
<td>January-March 2014</td>
<td>7</td>
</tr>
<tr>
<td>Q2-2014</td>
<td>April-June 2014</td>
<td>5</td>
</tr>
<tr>
<td>Q3-2014</td>
<td>July-September 2014</td>
<td>6</td>
</tr>
<tr>
<td>Q4-2014</td>
<td>October-December 2014</td>
<td>5</td>
</tr>
<tr>
<td>Q1-2015</td>
<td>January-March 2015</td>
<td>5</td>
</tr>
<tr>
<td>Q2-2015</td>
<td>April-June 2015</td>
<td>3</td>
</tr>
<tr>
<td>Q3-2015</td>
<td>July-September 2015</td>
<td>2</td>
</tr>
<tr>
<td>Q4-2015</td>
<td>October-December 2015</td>
<td>1</td>
</tr>
<tr>
<td><strong>TOTAL Milestones</strong></td>
<td><strong>128</strong></td>
<td></td>
</tr>
</tbody>
</table>
3.2 Virtual Learning Environment

a. University of Nigeria: The University of Nigeria established a vibrant Virtual Learning Environment using open-source, Learning Management System, Moodle. Over 400 Academic staff members were trained and were required to convert part of their course to an e-Learning platform. By the end of 2010, over 200 courses were in various stages of development (Figure 1). An e-learning Intranet was also developed for the General Studies.

Figure 1: University of Nigeria e-Learning Portal [http://learn.unn.edu.ng](http://learn.unn.edu.ng)


c. Federal University Oye-Ekiti: Has an e-learning platform using open-source, Learning Management System, Moodle. It is very active with tens of courses and tutorials for students.

3.3 Online Institutional Repository

a. University of Nigeria: as at December 27, 2013, over 21,000 documents have been digitized and uploaded to the University online repository at [http://unn.edu.ng/chart/repo](http://unn.edu.ng/chart/repo). The repository is registered under the OpenDOAR - Directory of Open Access Repositories. This is the largest online academic repository in
any Nigerian University. The repository contains about 10,000 University of Nigeria PhD, Masters, selected Bachelors Theses and University owned academic publications. The breakdown is shown in the table below:

Table 4: Breakdown of Digitised and Uploaded Documents on Repository

<table>
<thead>
<tr>
<th>Category</th>
<th>Number of Documents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arts</td>
<td>1,048</td>
</tr>
<tr>
<td>Agriculture</td>
<td>2,361</td>
</tr>
<tr>
<td>Biological Sciences</td>
<td>749</td>
</tr>
<tr>
<td>Business Administration</td>
<td>3,229</td>
</tr>
<tr>
<td>Dentistry</td>
<td>0</td>
</tr>
<tr>
<td>Education</td>
<td>5,609</td>
</tr>
<tr>
<td>Engineering</td>
<td>811</td>
</tr>
<tr>
<td>Environmental Studies</td>
<td>429</td>
</tr>
<tr>
<td>Health Sciences and Technology</td>
<td>300</td>
</tr>
<tr>
<td>Law</td>
<td>42</td>
</tr>
<tr>
<td>Medicine</td>
<td>1,894</td>
</tr>
<tr>
<td>Pharmaceutical Sciences</td>
<td>659</td>
</tr>
<tr>
<td>Physical Sciences</td>
<td>796</td>
</tr>
<tr>
<td>Social Sciences</td>
<td>2,836</td>
</tr>
<tr>
<td>Veterinary Medicine</td>
<td>402</td>
</tr>
<tr>
<td>Others</td>
<td>294</td>
</tr>
<tr>
<td>TOTAL</td>
<td>21,459</td>
</tr>
</tbody>
</table>

b. **Covenant University**: Covenant University has two Institutional Repositories using open source Dspace and Eprints software. The repositories with thousands of documents are both registered under the OpenDOAR - Directory of Open Access Repositories and are accesses at http://eprints.covenantuniversity.edu.g/ and http://dspace.covenantuniversity.edu.ng/

c. **Federal University Oye-Ekiti**: has one Institutional Repository using open source Dspace software. The repository can be accesses at http://repository.fuoye.edu.ng

### 3.4 Integrated Business Processes

To provide information robustness required towards becoming a world-class university, the information management must be run on an integrated enterprise platform that integrates the University business processes in procurement, teaching, planning, project management, student lifecycle management, asset management, logistics, financial accounting, human resources, and analytics application.

a. **University of Nigeria**: All university business process are been implemented using SAP and SAGE ERP solutions.
b. **Covenant University**: All university business process are been implemented using SAGE ERP solutions

c. **Federal University Oye-Ekiti**: All university business process are been implemented using open source solution, OpenERP.

3.5 Internationalisation through Global Classroom.

a. **University of Nigeria**: Created global classroom for teaching and learning with several overseas universities including the Earth Institute, USA, as well as with top Nigerians in the Diaspora.

b. Covenant University: Established international Linkages with close to 30 Universities and world-class organizations. It is a member of the East Carolina University Global Understanding Initiative.

c. Federal University Oye-Ekiti: Established a Webinar room for teaching and learning from across the globe.

3.6 Other Areas of Action

Other areas of action which the three universities are actively pursuing are:

- Employment and student admission policy encourages the employment of International and Overseas faculty and admission of international students. This will improve the International Diversity of the University.

- Raising the level of the Research and Institutional Indicators such as Academic papers, Research income, ratio of PhD/undergraduate degrees and PhD awarded.

- Commercialisation of research by their respective Innovation Centres thus assisting in attracting funds for PhD research.

- Increasing collaboration and linkages with both international and national universities and research institutions.

4.0 RESULTS AND DISCUSSIONS

The ranking history of the three universities on the Webometrics World Universities Ranking since they became focused on it is shown in the table below. The results show dramatic improvements in global ranking of the three universities.

a. **University of Nigeria**: In the July 2010 Webometrics World ranking, the University of Nigeria was ranked, for the first time ever, amongst the top 100 African Universities. It is today occupying the 31st position in Africa (Webometrics July 2013 edition).
b. **Covenant University**: was ranked amongst the top 100 African universities for the first time, in July 2011. (Webometrics, 2011)

c. **Federal University Oye-Ekiti**: has consistently topped the table of all the 12 new universities established by the Federal Government of Nigeria from year 2011 (Webometrics, 2012 and 2013 editions).

Table 5: Ranking\(^2\) History of Universities Studied ([www.webometrics.info](http://www.webometrics.info), various years)

<table>
<thead>
<tr>
<th>Date</th>
<th>University of Nigeria (UNN)</th>
<th>Covenant University (CU)</th>
<th>Federal University Oye-Ekiti (FUOYE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jul 2008</td>
<td>15,000+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jan 2009</td>
<td>13,000+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jul 2009</td>
<td>10,340</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jan 2010</td>
<td>8,285</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jul 2010</td>
<td><strong>7,170</strong></td>
<td><strong>9320</strong></td>
<td></td>
</tr>
<tr>
<td>Jan 2011</td>
<td>5176</td>
<td>8835</td>
<td></td>
</tr>
<tr>
<td>Jul 2011</td>
<td>5396</td>
<td>7169</td>
<td></td>
</tr>
<tr>
<td>Jan 2012</td>
<td>3228</td>
<td>7730</td>
<td></td>
</tr>
<tr>
<td>July 2012</td>
<td>4032</td>
<td>5491</td>
<td>15034</td>
</tr>
<tr>
<td>Jan 2013</td>
<td>2827</td>
<td>4289</td>
<td>12476</td>
</tr>
<tr>
<td>Jul 2013</td>
<td>2640</td>
<td>7856</td>
<td>12140</td>
</tr>
</tbody>
</table>

Figure 2: Improved Ranking Trend for UNN, CU and FUOYE (out of over 20,000 HEIs)

\(^2\) Lower is better
REFERENCES


APPENDIX A

Source: Proceedings of Deans and Associate Deans Workshop on Action Plan on World Universities Ranking, March 29, 2010

a. External Links (Ranking Weight-50%)

<table>
<thead>
<tr>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
</tbody>
</table>

b. Web Pages (Ranking Weight-20%)

<table>
<thead>
<tr>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
</tr>
<tr>
<td>9</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>11</td>
</tr>
<tr>
<td>12</td>
</tr>
</tbody>
</table>
c. Rich Files (Ranking Weight-15%)

<table>
<thead>
<tr>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
</tr>
<tr>
<td>14</td>
</tr>
<tr>
<td>15</td>
</tr>
<tr>
<td>16</td>
</tr>
<tr>
<td>17</td>
</tr>
</tbody>
</table>

d. Google Scholar (Ranking Weight-15%)

<table>
<thead>
<tr>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
</tr>
<tr>
<td>19</td>
</tr>
<tr>
<td>20</td>
</tr>
</tbody>
</table>

e. Usability & Policy

<table>
<thead>
<tr>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
</tr>
<tr>
<td>22</td>
</tr>
<tr>
<td>23</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

APPENDIX B

Table 6: ICT Policy Implementation Plan, FUOYE

<table>
<thead>
<tr>
<th>#</th>
<th>ACTIVITIES</th>
<th>2012</th>
<th>YEAR 2013</th>
<th>YEAR 2014</th>
<th>YEAR 2015</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Q4-12</td>
<td>Q1-13</td>
<td>Q2-13</td>
<td>Q3-13</td>
</tr>
<tr>
<td>A</td>
<td>VISION &amp; MISSION</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Discussions of ICT Policy Document</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Approval of the Minimum Guidelines Document</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>APPLICATIONS OF ICT TO EDUCATION DELIVERY AT FUOYE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>E-books/e-journals subscriptions, ScienceDirect, etc</td>
<td>14</td>
<td>14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Page</td>
<td>Title</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>E-Assessment/E-Testing - Moodle</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>E-learning - Moodle</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Voice and Video Conferencing - BigBlueButton, Google+</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>E-collaboration - Google+, Google hangout, Google Apps</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Plagiarism Policy - approval</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### APPLICATIONS OF ICT TO ADMINISTRATION IN FUOYE

<table>
<thead>
<tr>
<th>Page</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Identification - Biometrics RFID</td>
</tr>
<tr>
<td>11</td>
<td>Integrated Business Process using E-Business Solutions</td>
</tr>
<tr>
<td>12</td>
<td>Physical Planning - Georeferencing-GIS</td>
</tr>
<tr>
<td>13</td>
<td>Electronic Mail Policy Enforcement</td>
</tr>
<tr>
<td>14</td>
<td>e-Government: Digital Document Management System</td>
</tr>
</tbody>
</table>

### INFRASTRUCTURE

<table>
<thead>
<tr>
<th>Page</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>Last Mile - OFC</td>
</tr>
<tr>
<td>16</td>
<td>Internet Bandwidth</td>
</tr>
<tr>
<td>17</td>
<td>University Network - Wireless</td>
</tr>
<tr>
<td>18</td>
<td>University Backbone - Optical Fibre Cabling (OFC)</td>
</tr>
<tr>
<td>19</td>
<td>Central Storage</td>
</tr>
<tr>
<td>20</td>
<td>Data Centre &amp; Network Operating Centre</td>
</tr>
<tr>
<td>21</td>
<td>Data Resource Centre</td>
</tr>
<tr>
<td>22</td>
<td>Dedicated Power Supply</td>
</tr>
<tr>
<td>23</td>
<td>High Performance Computing</td>
</tr>
<tr>
<td>24</td>
<td>Integrated Business Process using E-Business Solutions</td>
</tr>
<tr>
<td>25</td>
<td>Call Centre</td>
</tr>
</tbody>
</table>

### NETWORK DEVELOPMENT AND MANAGEMENT POLICY

<table>
<thead>
<tr>
<th>Page</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>26</td>
<td>General Network Policy</td>
</tr>
<tr>
<td>27</td>
<td>Campus Area Networks</td>
</tr>
<tr>
<td>28</td>
<td>Inter-Campus Networks</td>
</tr>
<tr>
<td>29</td>
<td>Private Networks</td>
</tr>
<tr>
<td>30</td>
<td>External access to servers on the backbone network</td>
</tr>
</tbody>
</table>
### INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN IT AND ENGINEERING

**ISSN: 2278-6244**

**Vol. 3 | No. 2 | February 2014**

**<www.garph.co.uk> IJARIE | 25**

<table>
<thead>
<tr>
<th>#</th>
<th>ACTIVITIES</th>
<th>2012</th>
<th>2013</th>
<th>2014</th>
<th>2015</th>
</tr>
</thead>
<tbody>
<tr>
<td>31</td>
<td>Data Protection Policy - approval</td>
<td>24</td>
<td>50</td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>Back Up and Disaster Recovery Plan</td>
<td>7</td>
<td>25</td>
<td>51</td>
<td></td>
</tr>
<tr>
<td><strong>F</strong></td>
<td>ACCESS MANAGEMENT AND CONTROL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>Physical Access Control Policy - approval</td>
<td>26</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>Usage Policy - approval</td>
<td>27</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>Antivirus Policy - approval</td>
<td>28</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>Anti-Piracy Policy</td>
<td>8</td>
<td>29</td>
<td></td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>Maintenance Policy - approval</td>
<td>30</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>G</strong></td>
<td>CAPACITY BUILDING</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>ICT Literacy Campaign</td>
<td>9</td>
<td>31</td>
<td>52</td>
<td>76</td>
</tr>
<tr>
<td>39</td>
<td>Technical Team Training</td>
<td>32</td>
<td>53</td>
<td>77</td>
<td>91</td>
</tr>
<tr>
<td>40</td>
<td>Training Resources</td>
<td>10</td>
<td>33</td>
<td></td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>E-Learning Policy - approval</td>
<td>34</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>H</strong></td>
<td>EQUAL OPPORTUNITIES GUIDELINES</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>Accessibility Policy - enforcement</td>
<td>35</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>Gender / Ethnic / Religious Issues Policy enforcement</td>
<td>36</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>I</strong></td>
<td>MAINTENANCE OF ICT FACILITIES</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>Inventory of ICT Facilities - (digital, geo-referenced, etc)</td>
<td>11</td>
<td>37</td>
<td></td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>Hardware Maintenance Management Policy approval</td>
<td>38</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>Repair Centres</td>
<td>12</td>
<td>39</td>
<td>54</td>
<td></td>
</tr>
<tr>
<td><strong>J</strong></td>
<td>COLLABORATIVE SERVICES AND RESOURCE SHARING</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>Sharing of Infrastructure &amp; Resources Policy</td>
<td>40</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>Physical Connection</td>
<td>55</td>
<td>78</td>
<td>92</td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>Hosting services</td>
<td>56</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>Remote Laboratory activities</td>
<td>57</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>Computer-supported collaborative learning</td>
<td>58</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>52</td>
<td>Tele-participation</td>
<td>59</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>53</td>
<td>Collaborative Capacity Building</td>
<td>60</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>K</strong></td>
<td>WORLD UNIVERSITIES RANKING</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>54</td>
<td>Presence</td>
<td>13</td>
<td>41</td>
<td></td>
<td></td>
</tr>
<tr>
<td>55</td>
<td>Openness</td>
<td>42</td>
<td>61</td>
<td></td>
<td></td>
</tr>
<tr>
<td>56</td>
<td>Excellence</td>
<td>62</td>
<td>79</td>
<td>93</td>
<td></td>
</tr>
<tr>
<td>57</td>
<td>Impact</td>
<td>63</td>
<td>80</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
PROPOSED A METHOD BASED ON FUZZY LOGIC TO DETECT THE DRIVER'S
VOICE COMMANDS AND APPLIED TO AUDIO SYSTEM IN A NOISY
ENVIRONMENT

Navid Samimi Behbahan*
Zohreh Mousavinasab*

Abstract: This research investigates the recognition of limited letters in the noisy environment. This issue would be considered, when the driver listening to the music and driving the car simultaneously, which cause to reduce the concentration of driver on driving the car. The letters that driver would use are Next (next track), Previous (previous track), Louder (increase the volume), Lower (decrease the volume). In order to describe any frame, the LPCC constants and the sound energy with the first and second differentiations (39 characterizations in total) have been used.

*Sama Technical and Vocational Training College, Islamic Azad University, Omidiyeh Branch, Omidiyeh, Iran.
1- INTRODUCTION

The value of creating the technology to combine and recognize voices is enormous. Speaking is the most popular way to communicate fast and effective between humans. Writing and typing the letters with keyboards and other electrical controllers that have several keys and buttons can be replaced by the voice recognition technology. But, this new technology needs to be improved for commercial purposes. Voice recognition can make the computer easier for physically disabled people with good hearing and talking abilities. Voice combination not only uses the voice recognition techniques, but also can be used as a friendly outgoing device to diagnose different sounds and key words in order to replace the vision signs (such as traffic lights and etc.) and hearing signs (such as alarms and etc.) with these key words in different situations.

It should be considered that the progressing in voice recognition technology, not only covers the DSP area, but also, needs to have a good knowledge in artificial intelligence and artificial neural networks areas. Using these various sciences not only doesn’t make to find the best and ideal [1].

Using several scientific areas to investigate the voice recognition technique not only doesn’t make it hard to investigate, but also, increases the chance of achieving the ideal and efficient system. The voice recognition technology is the new technique to distinguish messages and voice commands in this area.

In the field of voice pattern recognition using the fuzzy logic, there isn’t enough publications in Iran, and the limited publications in this area only focused on general investigations and the introduction for these types of technique. The results conducted by this study are applicable and outcomes by the software with Matlab coding. The results have been represented in graphs and tables at the end of this research.

2- DATA RANGE

The range of words is one of the important factors in determining good quality discrete speech recognition. The purpose of this research is to achieve recognition in the range of four words. The word range of this research includes the words "previous" (means previous track), "next" (means next track), "low" (means low voice) and "high" (means high voice). Every word was repeated twenty times, ten of which were in noiseless environment and ten others in noisy environment. Afterward
the data were categorized into the trained and experimental data. They are used as the main data in this part of the research project of Persian speech processing.

### 3- FEATURE EXTRACTION

Linear prediction Coding (LPC) is an alternative method for spectral envelope estimation. This method is also known by the names, all-pole model, or the autoregressive (AR) model. It has good intuitive interpretation both in time domain (adjacent samples are correlated) and in frequency domain (all-pole spectrum corresponding to the resonance structure) [2].

The signal $s[n]$ is predicted by a linear combination of its past values. The predictor equation is defined as

$$s[n] = \sum_{k=1}^{p} a_k s[n-k]$$

Here $s[n]$ is the signal, $a_k$ are the predictor coefficients and $\hat{s}[n]$ is the predicted signal.

The prediction error signal, or residual, is defined as

$$e[n] = s[n] - \hat{s}[n]$$

The coefficients $a_k$ are determined by minimizing the residual energy $E[(e[n])^2]$ using the Levinson-Durbin algorithm. As shown in Fig.1, $s[n]$ is the speech signal, $e[n]$ is the voice source (glottal pulses), and $H(z)$ is the response of the vocal tract filter.

![Fig.1: speech signal](image)

$$e[n] = s[n] - \hat{s}[n]$$

$$= s[n] - \sum_{k=1}^{p} a_k s[n-k]$$

$$E(z) = S(z)[1 - \sum_{k=1}^{p} a_k z^{-k}]$$
Thus the spectral model, representing the vocal tract is

\[ H(z) = \frac{S(z)}{E(z)} \]

The predictor coefficients \( a_k \) are rarely used as features but they are transformed into the more robust Linear Predictive Cepstral Coefficients (LPCC) features. A recursive algorithm proposed by Rabiner and Juang can be used for computing the cepstral coefficients from the LPC coefficients [3]. However, unlike MFCC, the LPCC are not based on perceptual frequency scale, such as Mel-frequency scale. This led to the development of the Perceptual Linear Predictive (PLP) analysis.

4- FUZZY LOGIC

Sound recognition process contains fuzzy logic set creation process (Fig.2) and identification process (Fig.3). At the beginning of fuzzy logic set creation process the signals are sampled and normalized [4]. Afterwards data are converted through the Hamming window. Next data are converted into a frequency band through the fast Fourier transform. The fast Fourier transform creates feature vectors. Fuzzy logic set creation process and identification process is based on the same signal processing algorithms. The difference between them is a sequence of execution. All feature vectors are averaged in fuzzy logic set creation process. Two averaged feature vectors are created. Afterwards it is converted into fuzzy logic set. Fuzzy logic set creation process contains following steps: sampling, quantization, normalization, filtration, windowing, feature extraction (two averaged feature vectors) and fuzzy logic set formation [5].
Classification is used in the identification process. It is based on fuzzy logic. There was applied fuzzy logic as a classifier. To obtain results of recognition, it compares feature vector of new sample with averaged feature vector with the help of fuzzy logic functions. Identification process contains following steps: recording of acoustic signal, sound track division, sampling, quantization, normalization, filtration, windowing, feature extraction, classification.

Fig.2: Fuzzy logic set creation process

Fig.3: Identification process
4.1. ACOUSTIC SIGNAL RECORDING

The sound card with analogue-digital converter is able to record, process and replay sound. The recording of the acoustic signal is the first part of identification process. Acoustic signal is converted into digital data (wave format) by the microphone and the sound card. This wave file contains following parameters: sampling frequency is 16000 Hz, number of bits is 16, and number of channels is 1 (mono).

4.2. SOUND TRACK DIVISION

Application divides sound track into sound fragments. It divides data. Next it creates new wave header. Afterwards new wave header is copied. Then new wave header is added to each chunk of data. New wave files are obtained. These files are used in the identification process. There are following advantages of such solution: precise determination of sound appearing, precise sound identification, and application does not have to allocate as much memory in identification process.

4.3. SAMPLING

Sampling is a technique to convert an analog signal into a digital signal. It periodically samples an input signal and transforms into a sequence of intensity values. Sampling frequency is basic parameter. Sampling frequency is 16000 Hz in sound recognition application (Fig.4).

![Sound wave amplitude vs. time](image)

Fig.4: Sound of dc machine with shorted coils for five seconds before normalization
4.4. QUANTIZATION
Quantization is a technique to round intensity values to a quantum so that they can be represented by a finite precision. Precision of sample values is specific to number of bits. Common applied number of bits is 8 or 16. Sound recognition application uses 16 bits because it gives better precision. There is a choice of number of bits depending on quantity of input data and calculations speed in sound recognition process. The compromise is important to obtain good results in short time.

4.5. NORMALIZATION
In sound recognition application, the normalization is the process of changing of the amplitude of an audio signal. There is a possibility that some sounds aren’t recorded at the same level. It is essential to normalize the amplitude of each sample in order to ensure, that feature vectors will be comparable. All samples are normalized in the range $[-1.0, 1.0]$. In method the amplitude maximum of the samples is found and then each sample is divided by this maximum.

4.6. FILTRATION
Filtration is a very efficient way of removing the unwanted noise from the spectrum. The filtration is used to modify the frequency domain of the input sample. The filtration is not necessary to sound recognition. However the usage of this can improve the efficiency of the sound recognition.

4.7. WINDOWING
Windowing is a technique used to shape the time portion of measurement data, to minimize edge effects that result in spectral leakage in the FFT spectrum. By using window functions, the spectral resolution of frequency domain will be increased. There are different types of window functions available, each with their own advantage. The Hamming window is used to avoid distortion of the overlapped window functions.

4.8. FAST FOURIER TRANSFORM
The sequence of frequency of a signal obtained by FFT becomes the basis for extracting of the frequency-domain features. It is applied instead of discrete Fourier transform because of shorter time of calculations. Obtained coefficients create feature vectors which are used in calculations.
4.9. CLASSIFICATION

Difference between sounds depends on differences in ordered sequence [6]. Classification uses feature vectors and fuzzy logic functions in the identification process. It compares different values of feature vectors. It compares feature vectors with the help of fuzzy logic functions (feature vector of investigated sample, feature vector of specific category). Fuzzy logic functions use amplitude of the sample to determine probability (Fig. 5–6). If probability of determined fuzzy function is greater than 0.5 and then function is chosen (240 functions). There are some points where probability is 0.5 and then one of fuzzy function is chosen. Category including the most correct values is the result of the identification

\[
p(x) = \begin{cases} 
0 & x < a_1 \\
\frac{x - a_1}{a_2 - a_1} & a_1 \leq x < a_2 \\
\frac{a_3 - x}{a_3 - a_2} & a_2 \leq x < a_3 \\
0 & x \geq a_3 
\end{cases} \tag{1}
\]

Fig. 5: Frequency spectrum of sound of faultless dc machine for ten seconds after normalization with fuzzy logic functions.

Fig. 6. Fuzzy logic function.
5- CONCLUSION

In this research, a hundred men and women with different ages has been chosen randomly, and asked them to repeat four words (Next, Previous, Louder and Lower), which causes to collect and prepared the database with four hundred voice recorded files. The LPCC coefficients and their related differentials have been extracted (39 characterizations). This database divided to four different classes and 100 Fuzzy rules have been derived by processing these data (400 samples). The input data as discussed before have been held in these two different classes by derivation of their properties and comprised with Fuzzy rules. The detection rate by 10 causeways is 96.33.

REFERENCES


A FRAMEWORK OF TRANSACTION MANAGEMENT IN DISTRIBUTED DATABASE SYSTEM ENVIRONMENT

Dr. Farid Ahmad*

Abstract: The main objective of the paper is to determine the transaction management techniques in DDBMS for present time business applications. Distributed database systems (DDBS) poses different problems when accessing distributed and replicated databases. Particularly, access control and transaction management in DDBS require different mechanism to monitor data retrieval and update to databases. It primarily concentrates on the description of the used commit protocols in distributed database. Current trends in multi-tier client/server networks make DDBS an appropriated solution to provide access to and control over localized databases. Oracle, as a leading Database Management System (DBMS) vendor employs the two-phase commit technique to maintain consistent state for the database. There is an example given that demonstrate the steps involved in executing the two-phase commit protocol. This transaction protocol will be beneficial for the organizations to manage the enterprise data resource in DDBS environments efficiently.

Key words: Transaction management, two-phase commit, distributed database systems, atomic commitment protocol (ACP) and Enhanced three phase commit (E3PC) etc.

*Assistant Professor, Adama Science & Technology University, Adama (OROMIA), Ethiopia (Africa).
INTRODUCTION:

Distributed database systems are systems that have their data distributed and replicated over several locations, not like the centralized database system, where one copy of the data is stored. Data may be replicated over a network using horizontal and vertical fragmentation similar to projection and selection operations in Structured Query Language (SQL). In other words, Distributed database as a collection of multiple, logically interrelated databases distributed over a computer network. A distributed database management system (DDBMS) is then defined as the software system that permits the management of the distributed database and makes the distribution transparent to the users. Sometimes “distributed database system” (DDBS) is used to refer jointly to the distributed database and the distributed DBMS. Both types of database share the same problems of access control and transaction management, such as user concurrent access control and deadlock detection and resolution. On the other hand, however, DDBS must also cope with different problems. Access control and transaction management in DDBS require different rules to monitor data retrieval and update to distributed and replicated databases. Oracle, as a leading Database Management Systems (DBMS) employs the two-phase commit technique to maintain a consistent state for the databases. The objective of this paper is to explain transaction management in DDBMS and how Oracle implements this technique. To assist in understanding this process, an example is given in the last section. It is hoped that this understanding will encourage organizations to use and academics to discuss DDBS and to successfully capitalize on this feature of Oracle. It provide discussions on the fundamentals of transaction management, two-phase commit, Oracle’s implementation of the two phase commit, and, finally, an example on how the two phases commit works.

FUNDAMENTALS OF TRANSACTION MANAGEMENT

Transaction Management deals with the problems of keeping the database in a consistent state even when concurrent accesses and failures occurs, [1].

Transaction:

A transaction consists of a series of operations performed on a database. The important issue in transaction management is that if a database was in a consistent state prior to the initiation of a transaction, then the database should return to a consistent state after the transaction is completed. This should be done irrespective of the fact that transactions were
successfully executed simultaneously or there were failures during the execution [2]. A transaction is a sequence of operations that takes the database from a consistent state to another consistent state. It represents a complete and correct computation.

Two types of transactions are allowed in our environment: query transactions and update transactions. Query transactions consist only of read operations that access data objects and return their values to the user. Thus, query transactions do not modify the database state. Two transactions conflict if the read-set of one transaction intersects with the write-set of the other transaction. During the voting process, Update transactions consist of both read and write operations. Transactions have their time-stamps constructed by adding 1 to the greater of either the current time or the highest time-stamp of their base variables. Thus, a transaction is a unit of consistency and reliability. Each transaction has to terminate. The outcome of the termination depends on the success or failure of the transaction. When a transaction starts executing, it may terminate with one of two possibilities:

1. The transaction **aborts** if a failure occurred during its execution
2. The transaction **commits** if it was completed successfully example of a transaction that aborts during process 2 (P2). On the other hand, an example of a transaction that commits, since all of its processes are successfully completed [3].

**Properties of Transactions**

A Transaction has four properties that lead to the consistency and reliability of a distributed data base. These are Atomicity, Consistency, Isolation, and Durability, [1, 3].

**Atomicity**

This refers to the fact that a transaction is treated as a unit of operation. Consequently, it dictates that either all the actions related to a transaction are completed or none of them is carried out. For example, in the case of a crash, the system should complete the remainder of the transaction, or it will undo all the actions pertaining to this transaction. The recovery of the transaction is split into two types corresponding to the two types of failures: the transaction recovery, which is due to the system terminating one of the transactions because of deadlock handling; and the crash recovery, which is done after a system crash or a hardware failure.
Consistency

Referring to its correctness, this property deals with maintaining consistent data in a database system. Consistency falls under the subject of concurrency control. For example, “dirty data” is data that has been modified by a transaction that has not yet committed. Thus, the job of concurrency control is to be able to disallow transactions from reading or updating ‘dirty data’.

Isolation

According to this property, each transaction should see a consistent database at all times. Consequently, no other transaction can read or modify data that is being modified by another transaction. If this property is not maintained, one of two things could happen to the database. 

a. Lost Updates: this occurs when transaction (T2) updates the same data being modified by the transaction (T1) in such a manner that T2 reads the value prior to the writing of T1 thus creating the problem of losing this update.

b. Cascading Aborts: this problem occurs when the first transaction (T1) aborts, then the transactions that had read or modified data that has been used by T1 will also abort.

Durability

This property ensures that once a transaction commits, its results are permanent and cannot be erased from the database. This means that whatever happens after the COMMIT of a transaction, whether it is a system crash or aborts of other transactions, the results already committed are not modified or undone.

1. DISTRIBUTED TRANSACTION MANAGEMENT

This section provides general background on the atomic commit problem and protocols.

1.1. Problem Definition

A distributed transaction is composed of several sub-transactions, each running on a different site. The database manager at each site can unilaterally decide to abort the local sub-transaction, in which case the entire transaction must be aborted. If all the participating sites agree to commit their sub-transaction (vote Yes on the transaction) and no failures occur, the transaction should be committed. I assume that the local database server at each site can atomically execute the sub-transaction once it has agreed to commit it. In order to ensure that all the sub-transactions are consistently committed or aborted,
the sites run an atomic commitment protocol such as two phases commit. The requirements of atomic commitment [7]) are as follows:

**AC1**: Uniform Agreement: All the sites that reach a decision reach the same one.

**AC2**: A site cannot reverse its decision after it has reached one.

**AC3**: Validity: The commit decision can be reached only if all sites voted Yes.

**AC4**: Non-triviality: If there are no failures and all sites voted Yes, then the decision will be to commit.

**AC5**: Termination: At any point in the execution of the protocol, if all existing failures are repaired and no new failures occur for sufficiently long, then all sites will eventually reach a decision.

**The Prepare and Commit Phases**

**Prepare Phase**

The first phase in committing a distributed transaction is the prepare phase in which the commit of the transaction is not actually carried out. Instead, all nodes referenced in a distributed transaction (except one, known as the commit point site) are told to prepare (to commit). By preparing, a node records enough information so that it can subsequently either commit or abort the transaction (in which case, a rollback will be performed), regardless of intervening failures. Prepare phase; the global coordinator (initiating node) asks participants to prepare (to promise to commit or rollback the transaction, even if there is a failure).

**Commit phase**: if all participants respond to the coordinator that they are prepared, the coordinator asks all nodes to commit the transaction. If any participants cannot prepare, the coordinator asks all nodes to roll back the transaction.

When a node responds to its requestor that it has prepared, the prepared node has made a promise to be able to either commit or roll back the transaction later and not to make a unilateral decision on whether to commit or roll back the transaction [6].

Queries that start after a node has prepared cannot access the associated locked data until all phases are complete (an insignificant amount of time unless a failure occurs). When a node is told to prepare, it can respond with one of three responses:

**Prepare Phase Actions by Nodes**

To complete the prepare phase, each node performs the following actions:
The node requests its descendants (nodes subsequently referenced) to prepare.

The node checks to see if the transaction changes data on that node or any of its descendants. If there is no change, the node skips the next steps and replies with a read-only message.

The node allocates all resources it needs to commit the transaction if data is changed.

The node flushes any entries corresponding to changes made by that transaction to its local redo log.

The node guarantees that locks held for that transaction are able to survive a failure.

The node responds to the node that referenced it in the distributed transaction with a prepared message or, if its prepare or the prepare of one of its descendents was unsuccessful, with an abort message.

These actions guarantee that the transaction can subsequently commit or roll back on that node. The prepared nodes then wait until a COMMIT or ROLLBACK is sent. Once the node(s) are prepared, the transaction is said to be in-doubt. Prepared Data on the node has been modified by a statement in the distributed transaction, and the node has successfully prepared. Read-only No data on the node has been, or can be, modified (only queried), so no prepare is necessary. Abort The node cannot successfully prepare.

**Read-only Response**

When a node is asked to prepare and the SQL statements affecting the database do not change that node’s data, the node responds to the node that referenced it with a read-only message. These nodes do not participate in the second phase (the commit phase).

**Unsuccessful Prepare**

When a node cannot successfully prepare, it performs the following actions:

- That node releases any resources currently held by the transaction and rolls back the local portion of the transaction.

- The node responds to the node that referenced it in the distributed transaction with an abort message.

These actions then propagate to the other nodes involved in the distributed transaction to roll back the transaction and guarantee the integrity of the data in the global database.
Again, this enforces the primary rule of a distributed transaction. All nodes involved in the transaction either all commit or all roll back the transaction at the same logical time.

**Commit Phase**

The second phase in committing a distributed transaction is the commit phase. Before this phase occurs, all nodes referenced in the distributed transaction have guaranteed that they have the necessary resources to commit the transaction. That is, they are all prepared. Therefore, the commit phase consists of the following steps [6]:

1. The global coordinator sends a message to all nodes telling them to commit the transaction.
2. At each node, Oracle commits the local portion of the distributed transaction (releasing locks) and records an additional redo entry in the local redo log, indicating that the transaction has committed.

When the commit phase is complete, the data on all nodes of the distributed system are consistent with one another. A variety of failure cases, caused by network or system failures, are possible during both the prepare phase and the commit phase.

**1.2. Two Phase Commit**

The simplest and most renowned atomic commitment protocol (ACP) is two phases commit [7]. Several variations of 2PC have been suggested (e.g., presume abort and presume commit); the simplest version is centralized; one of the sites is designated as the coordinator. The coordinator sends a transaction (or request to prepare to commit) to all the participants. Each site answers by a Yes (“ready to commit”) or by a No (”abort”) message. If any site votes No, all the sites abort.

The 2-phase commit (2PC) protocol is a distributed algorithm to ensure the consistent termination of a transaction in a distributed environment. Thus, via 2PC a unanimous decision is reached and enforced among multiple participating servers whether to commit or abort a given transaction, thereby guaranteeing atomicity. The protocol proceeds in two phases, namely the prepare and the commit phase, which explains the protocol’s name. The protocol is executed by a coordinator process, while the participating servers are called participants. When the transaction’s initiator issues a request to commit the transaction, the coordinator starts the first phase of the 2PC protocol by querying—via prepare messages—all participants whether to abort or to commit the transaction.
initiates the first phase of the protocol by sending *PREPARE* (to commit) messages in parallel to all the cohorts. Each cohort that is ready to commit first force-writes a *prepare* log record to its local stable storage and then sends a *YES* vote to the master. At this stage, the cohort has entered a *prepared* state wherein it cannot unilaterally commit or abort the transaction but has to wait for the final decision from the master. On the other hand, each cohort that decides to abort force-writes an *abort log* record and sends a *NO* vote to the master. Since a *NO* vote acts like a veto, the cohort is permitted to unilaterally abort the transaction without waiting for a response from the master. After the master receives the votes from all the cohorts, it initiates the second phase of the protocol. If all the votes are *YES*, it moves to a *committing* state by force-writing a *commit* log record and sending *COMMIT* messages to all the cohorts. Each cohort after receiving a *COMMIT* message moves to the committing state, force-writes a *commit log* record, and sends an *ACK* message to the master. If the master receives even one *NO* vote, it moves to the aborting state by force-writing an *abort log* record and sends *ABORT* messages to those cohorts that are in the prepared state. These cohorts, after receiving the *ABORT* message, move to the *aborting* state, force write an *abort log* record and send an *ACK* message to the master [8].

Finally, the master, after receiving acknowledgements from all the prepared cohorts, writes an *end log* record and then — forgets — the transaction. The 2PC may be carried out with one of the following methods: Centralized 2PC, Linear 2PC, and Distributed 2PC, [3].

### 1.3. The Centralized Two-Phase Commit Protocol

In the Centralized 2PC shown in Figure 3, communication is done through the coordinator’s process only, and thus no communication between subordinates is allowed. The coordinator is responsible for transmitting the PREPARE message to the subordinates, and, when the votes of all the subordinates are received and evaluated, the coordinator decides on the course of action: either abort or COMMIT. This method has two phases:

**1. First Phase:** In this phase, when a user wants to COMMIT a transaction, the coordinator issues a PREPARE message to all the subordinates, (Mohan et al., 1986). When a subordinate receives the PREPARE message, it writes a PREPARE log and, if that subordinate is willing to COMMIT, sends a *YES VOTE*, and enters the PREPARED state; or, it writes an abort record and, if that subordinate is not willing to COMMIT, sends a *NO VOTE*. A subordinate sending a *NO VOTE* doesn’t need to enter a PREPARED state since it knows that
the coordinator will issue an abort. In this case, the NO VOTE acts like a veto in the sense that only one NO VOTE is needed to abort the transaction. The following two rules apply to the coordinator’s decision, [3].

a. If even one participant votes to abort the transaction, the coordinator has to reach a global abort decision.

b. If all the participants vote to COMMIT, the coordinator has to reach a global COMMIT decision.

2. Second Phase: After the coordinator reaches a vote, it has to relay that vote to the subordinates. If the decision is COMMIT, then the coordinator moves into the committing state and sends a COMMIT message to all the subordinates informing them of the COMMIT. When the subordinates receive the COMMIT message, they, in turn, move to the committing state and send an acknowledge (ACK) message to the coordinator. When the coordinator receives the ACK messages, it ends the transaction. If, on the other hand, the coordinator reaches an ABORT decision, it sends an ABORT message to all the subordinates. Here, the coordinator doesn’t need to send an ABORT message to the subordinate(s) that gave a NO VOTE.

1.4. The Linear Two-Phase Commit Protocol

In the linear 2PC, as depicted in Figure 4, subordinates can communicate with each other. The sites are labeled 1 to N, where the coordinator is numbered as site 1. Accordingly, the propagation of the PREPARE message is done serially. As such, the time required to complete the transaction is longer than centralized or distributed methods. Finally, node N is the one that issues the Global COMMIT. The two phases are discussed below, [3]:

First Phase: The coordinator sends a PREPARE message to participant 2. If participant 2 is not willing to COMMIT, then it sends a VOTE ABORT (VA) to participant 3 and the transaction is aborted at this point. If participant 2, on the other hand, is willing to commit, it sends a VOTE COMMIT (VC) to participant 3 and enters a READY state. In turn, participant 3 sends its vote till node N is reached and issues its vote.

Second Phase: Node N issues either a GLOBAL ABORT (GA) or a GLOBAL COMMIT (GC) and sends it to node N-1. Subsequently, node N-1 will enter an ABORT or COMMIT state. In turn, node N-1 will send the GA or GC to node N-2, until the final vote to commit or abort reaches the coordinator, node.
1.5. The Distributed Two-Phase Commit Protocol

In the distributed 2PC, all the nodes communicate with each other. According to this protocol, as Figure 5 shows, the second phase is not needed as in other 2PC methods. Moreover, each node must have a list of all the participating nodes in order to know that each node has sent in its vote. The distributed 2PC starts when the coordinator sends a PREPARE message to all the participating nodes. When each participant gets the PREPARE message, it sends its vote to all the other participants. As such, each node maintains a complete list of the participants in every transaction,[3]. Each participant has to wait and receive the vote from all other participants. When a node receives all the votes from all the participants, it can decide directly on COMMIT or abort. There is no need to start the second phase, since the coordinator does not have to consolidate all the votes in order to arrive at the final decision.

The coordinator collects all the responses and informs all the sites of the decision. In absence of failures, this protocol preserves atomicity. Between the two phases, each site blocks, i.e., keeps the local database locked, waiting for the final word from the coordinator. If a site fails before its vote reaches the coordinator, it is usually assumed that it had voted No. If the coordinator fails in the first phase, all the sites remain blocked indefinitely, unable to resolve the last transaction. The centralized version of 2PC is depicted in Fig. 1. Commit protocols may also be described using state diagrams [7]. The state diagram for 2PC is shown in Fig. 1. The circles denote states; final states are double- circled. The arcs represent state transitions, and the action taken (e.g., message sent) by the site is indicated next to each arc. In this protocol, each site (either coordinator or participant) can be in one of four possible states: q: initial state; A site is in the initial state until it decides whether to unilaterally abort or to agree to commit the transaction. w: wait state; In this state the coordinator waits for votes from all of the participants, and each participant waits for the final work from the coordinator. This is the “uncertainty period” for each site, when it does not know whether the transaction will be committed or not. c: commit state; The site knows that a decision to commit was made. a: abort state; The site knows that a decision to abort was made. The states of a commit protocol may be classified along two orthogonal lines. In the first dimension, the states are divided into two disjoint subsets: The committable states and the non-committable states. A
site is in a committable state only if it knows that all the sites have agreed to proceed with the trans-action. The rest of the states are non-committable. The only committable state in 2PC is the commit state. The second dimension distinguishes between final and non-final states. The final states are the ones in which a decision has been made and no more state transitions are possible. The final states in 2PC are commit and abort, [7].

**The Commit Point Site**

The job of the commit point site is to initiate a commit or roll back as instructed by the global coordinator. The system administrator always designates one node to be the *commit point site* in the session tree by assigning all nodes commit point strength. The node selected as commit point site should be that node that stores the most critical data (the data most widely used). The commit point site is distinct from all other nodes involved in a distributed transaction with respect to the following two issues:

1. The commit point site never enters the prepared state. This is potentially advantageous because if the commit point site stores the most critical data, this data never remains in-doubt, even if a failure situation occurs. (In failure situations, failed nodes remain in a prepared state, holding necessary locks on data until in-doubt transactions are resolved.)

2. In effect, the outcome of a distributed transaction at the commit point site determines whether the transaction at all nodes is committed or rolled back. The global coordinator ensures that all nodes complete the transaction the same way that the commit point site does. A distributed transaction is considered to be committed once all nodes are prepared and the transaction has been committed at the commit point site (even though some participating nodes may still be only in the prepared state and the transaction not yet actually committed). The commit point site’s redo log is updated as soon as the distributed transaction is committed at that node, [6]. Likewise, a distributed transaction is considered *not* committed if it has not been committed at the commit point site.

**Failures that Interrupt Two-Phase Commit**

The user program that commits a distributed transaction is informed of a problem by one of the following error messages:

ORA-02050: transaction ID rolled back,
some remote dbs may be in-doubt
ORA-02051: transaction ID committed,
some remote dbs may be in-doubt
ORA-02054: transaction ID in-doubt

A robust application should save information about a transaction if it receives any of the above errors. This information can be used later if manual distributed transaction recovery is desired.

Note: The failure cases that prompt these error messages are beyond the scope of this book and are unnecessary to administer the system. No action is required by the administrator of any node that has one or more indoubt distributed transactions due to a network or system failure. The automatic recovery features of Oracle transparently complete any in-doubt transaction so that the same outcome occurs on all nodes of a session tree (that is, all commit or all roll back) once the network or system failure is resolved. However, in extended outages, the administrator may wish to force the commit or rollback of a transaction to release any locked data. Applications must account for such possibilities.

Failures that Prevent Data Access

When a user issues a SQL statement, Oracle attempts to lock the required resources to successfully execute the statement. However, if the requested data is currently being held by statements of other uncommitted transactions and continues to remain locked for an excessive amount of time, a time-out occurs. Consider the following two scenarios, [6].

Transaction Time-Out

A DML SQL statement that requires locks on a remote database may be blocked from doing so if another transaction (distributed or non-distributed) currently own locks on the requested data. If these locks continue to block the requesting SQL statement, a time-out occurs, the statement is rolled back, and the following error message is returned to the user:

ORA-02049: time-out: distributed transaction waiting for lock. Because no data has been modified, no actions are necessary as a result of the timeout. Applications should proceed as if a deadlock has been encountered. The user who executed the statement can try to re-execute the statement later. If the lock persists, the user should contact an administrator to report the problem. The timeout interval in the above situation can be controlled with the initialization parameter DISTRIBUTED_LOCK_TIMEOUT. This interval is in seconds. For
example, to set the time-out interval for an instance to 30 seconds, include the following line in the associated parameter file:

```
DISTRIBUTED_LOCK_TIMEOUT=30
```

With the above time-out interval, the time-out errors discussed in the previous section occur if a transaction cannot proceed after 30 seconds of waiting for unavailable resources.

### 1.6. Quorums

In order to reduce the extent of blocking in replication and atomic commit protocols, majority votes or quorums are often used. A quorum system is a generalization of the majority concept. Enhanced three phase commit (E3PC), like Skeen’s quorum-based three phase commit protocol, uses a quorum system to decide when a group of connected sites may resolve the transaction. To enable maximum flexibility the quorum system may be elected in a variety of ways (e.g., weighted voting). The quorum system is static; it does not change in the course of the protocol. The predicate \( Q(S) \) is true for a given subset \( S \) of the sites iff \( S \) is a quorum. The requirement from this predicate is that for any two sets of sites \( S \) and \( S' \) such that \( S \cap S' = \emptyset \), at most one of \( Q(S) \) and \( Q(S') \) holds, i.e., every pair of quorums intersect. For example, in the simple majority quorum system \( Q(S) \) is true iff \( |S| > n/2 \), where \( n \) is the total number of sites running the protocol. Numerous quorum systems that fulfill these criteria were suggested. An analysis of the availability of different quorum systems may be found in [9].

For further flexibility, it is possible to set different quorums for commit and abort (this idea was presented in [7]). In this case, a commit quorum of connected sites is required in order to commit a transaction, and an abort quorum is required to abort. For example, to increase the probability of commit in the system, one can assign smaller quorums for commit and larger ones for abort. In this case, the quorum system consists of two predicates: \( Q_c(G) \) is true for a given group of sites \( G \) iff \( G \) is a commit quorum, and \( Q_a(G) \) is true iff \( G \) is an abort quorum. The requirement from these predicates is that for any two groups of sites \( G \) and \( G' \) such that \( G \cap G' = \emptyset \), at most one of \( Q_c(G) \) and \( Q_a(G') \) holds, i.e., every commit quorum intersects every abort quorum.

### 1.7. The Extent of Blocking in Commit Protocols

The 2PC protocol is an example of a blocking protocol, [7]: operational sites sometimes wait on the recovery of failed sites.
Locks must be held in the database while the transaction is blocked. Even though blocking preserves consistency, it is highly undesirable because the locks acquired by the blocked transaction cannot be relinquished, rendering the data inaccessible by other requests. Consequently, the availability of data stored in reliable sites can be limited by the availability of the weakest component in the distributed system; [7] proved that there exists no non-blocking protocol resilient to network partitioning. When a partition occurs, the best protocols allow no more than one group of sites to continue while the remaining groups block. Skeen suggested the quorum-based three phase commit protocol, which maintains consistency in spite of network partitions. This protocol is blocking in case of partitions; it is possible for an operation site to be blocked until a failure is mended. In case of failures, the algorithm uses a quorum (or majority)-based recovery procedure that allows a quorum to resolve the transaction. If failures cascade, however, a quorum of sites can become connected and still remain blocked.

Since completely non-blocking recovery is impossible to achieve, further research in this area concentrated on minimizing the number of blocked sites when partitions occur. Define optimal termination protocols (recovery procedures) in terms of the average number of sites that are blocked when a partition occurs. The average is over all the possible partitions and all the possible states in the protocol in which the partitions occur. The analysis deals only with states in the basic commit protocol and ignore the possibility for cascading failures (failures that occur during the recovery procedure). It is proved that any ACP with optimal recovery procedures takes at least three phases and that the quorum-based recovery procedures are optimal. I construct an ACP that always allows a connected majority to
proceed, regardless of past failures. To our knowledge, no other ACP with this feature was suggested. The ACP suggested in uses a reliable replication service as a building block and is mainly suitable for replicated database systems. In this paper, I present a novel commitment protocol, enhanced three phase commit, which always allows a connected majority to resolve the transaction (if it remains connected for sufficiently long). E3PC does not require complex building blocks, and is more adequate for partially replicated or non-replicated distributed database systems; it is based on the quorum-based three phase commit, [7].

AN EXAMPLE OF A DISTRIBUTED DATABASE SYSTEM

Fig. 2 illustrates the steps homogenous distributed database performs in order to PREPARE, Select the COMMIT Point Site, and COMMIT. The example in the figure depicts a company that has several branches located in different cities numbered A to G. Each site has to have access to most of the data in the company in order to check on the status of purchase orders, material acquisition, and several other issues. Since new projects are awarded and older projects are completed, project sites tend to change locations. Also, depending on the size and duration of a project, different COMMIT point strength can be assigned and thus, in the same area, different COMMIT point sites can be chosen, for a given location, over a period of time. In this example, City E is the head office and thus posses the highest COMMIT point strength. The other sites are assigned the COMMIT point strength based on the rupee volume of the project. Higher monetary value for a project requires more resource allocation, and as such, will lead to more transactions executed against the data for that project. Since the amount of data involved is large, each site will have the portion of the database pertaining to its operations replicated and stored on a local server. Any transaction will at least affect the database at the head office and one of the sites. If, for example, a material rate, description of an item, accomplished progress, or purchase order is entered, a transaction is initiated that will affect the database at the head office and the database at the concerned site, [3].
Additional modifications, such as those involving employee transfer or equipment transfer from one site to another, will affect two or more sites. The following discussion explains the steps that entail in processing a distributed transaction: An employee is to be transferred from City F to City B. The transaction is initiated by City E by a personnel employee. The affected sites need to participate in the transaction. The processes that transfer one employee from one site to another should be grouped under one transaction so that either all or none of the processes are carried out. An explanation of these steps follows, [3]:

1. Since City E is initiating the transaction, it becomes the root of the session tree, i.e. the global coordinator. Since City 1 updates data in City F and City B, it becomes a client. Since City E updates data on City G and City B, the two nodes become database servers.

2. When the application issues the COMMIT statement, the two-phase commit is started.

3. The global coordinator determines the COMMIT point site.

4. The global coordinator issues the PREPARE statement to all nodes except the COMMIT point site. If any of the nodes cannot PREPARE, the transaction is aborted; otherwise, a PREPARED message is sent to the node that referenced it.

5. The global coordinator instructs the COMMIT point site to COMMIT. The COMMIT point site commits the transaction locally and records the transaction in its local redo log.

6. The COMMIT point site informs the global coordinator that it has committed and the global coordinator informs the other nodes by sending the COMMIT message.

7. When all the transactions have committed, the global coordinator informs the COMMIT point site to ―forget‖ about the transaction. The COMMIT point site, after ―forgetting‖ about the transaction, informs the global coordinator, and the global coordinator, in turn, ―forgets‖ about the transaction.
2.1. Failures in Distributed DBS

Several types of failures, [2] may occur in distributed database systems:

**Transaction Failures:** When a transaction fails, it aborts. Thereby, the database must be restored to the state it was in before the transaction started. Transactions may fail for several reasons. Some failures may be due to deadlock situations or concurrency control algorithms. Site Failures; Site failures are usually due to software or hardware failures. These failures result in the loss of the main memory contents. In distributed database, site failures are of two types:

1. **Total Failure:** where all the sites of a distributed system fail,
2. **Partial Failure:** where only some of the sites of a distributed system fail.

**Media Failures:** Such failures refer to the failure of secondary storage devices. The failure itself may be due to head crashes, or controller failure. In these cases, the media failures result in the inaccessibility of part or the entire database stored on such secondary storage.

**Communication Failures:** Communication failures, as the name implies, are failures in the communication system between two or more sites. This will lead to network partitioning where each site, or several sites grouped together, operates independently. As such, messages from one site won’t reach the other sites and will therefore be lost. The reliability protocols then utilize a timeout mechanism in order to detect undelivered messages. A message is undelivered if the sender doesn’t receive an acknowledgment. The failure of a communication network to deliver messages is known as performance failure, [1].

**CONCLUSION:**

The present time Transaction management is a fully grown thought in distributed data base management systems for research area. Homogenous Distributed Database Systems based replication proposal is able to inherit and reduces the communication traffic the best characteristics of the Database Systems. However, Oracle was the first commercial DBMS to implement a method of transaction management: the two-phase commit. Though it was very difficult to obtain in order on homogenous DBMS implementation of this method were able to pull together sufficient in sequence to put in writing homogenous transaction for the database system. Many associations do not implement distributed databases because of its difficulty. They simply resort to centralized databases. However, with global organizations and multi-tier network architectures, distributed implementation becomes a necessity. It is
hoped that this paper to will assist organization in the implementation of distributed databases when installing homogenous DBMS, or give confidence organizations to journey from centralized to distribute DBMS, [7]. Organizations could also contribute to this process by having graduates with the knowledge of homogeneous DBMS capability.

This paper demonstrates, how the commit protocols can be made more efficient and simply by maintaining two additional counters and by changing the decision rule and defines the distributed transaction management in efficient way with others necessary factors. The new protocol, E3PC, always allows a quorum of connected sites to resolve a transaction: At any point in the execution of the protocol, if a group G of sites becomes connected and this group contains a quorum of the sites, and no subsequent failures occur for sufficiently long, then all the members of G eventually reach to a decision.

REFERENCES


BULLWHIP EFFECT REDUCTION THROUGH MULTI-AGENT SYSTEM USING JADE TOOL

Sandeep Garg*
Dr. S. Srinivasan**

Abstract: The bullwhip effect is the magnification of demand fluctuations, not the magnification of demand. Whenever demand increases and decreases, the bull whip effect are evident in a supply chain. When the supply chain is large, the issue becomes more complex. Also Bullwhip effect is caused from distortions in information along the supply chain Some of the bull whip effects are excess inventories, problems with quality, increased costs, overtime expenditures, lost customer service, lost sales and more. This leads to poor forecasting of sales, incorrect information along the supply chain. In order to solve bullwhip effect, it is proposed that Multi-agent based Supply chain management which ensures entire supply chain working on real time. Intention of our work is to develop a supply chain management application based on MAS and SOA and build a model how it can help in reducing bullwhip effect in a manufacturing unit. The Supply Chain Management application consists of three different services i.e. MAS, SOA and SCM. These services are designed, integrated and architected separately and brought them together using MAS and SOA technologies.

*Research Scholar, NIMS University, Jaipur, India
**Professor Department of Computer Applications, PDMCE Bahadurgarh, Jhajjar, India
I. INTRODUCTION

A supply chain consists of all parties involved, directly or indirectly, in fulfilling a customer request, or demand. The supply chain not only includes the manufacture and suppliers, but also transporters, warehouses, retailers, and finally the end consumers themselves. The objective of every supply chain is to maximize the overall value generated. The value a supply chain generates is the difference between what the final product is worth to the customer and the effort the supply chain expends in filling the customer’s request.

Supply chain management involves the management of flows of information, product, or funds between and among stages in a supply chain to maximize total supply chain profitability. An important phenomenon in SCM, known as the bullwhip effect, which suggests that the demand variability increases as one moves up a supply chain. It is 1989 that Sterman first introduced regarding this effect. Since then, worldwide researches have been carried out by various authors to study different aspects of SCM, causing the bullwhip effect and suggested a number of methods to reduce its effect.

The impact of the bullwhip effect is to increase Manufacturing cost, Inventory cost, Replenishment lead time, Transportation cost, Labor cost for shipping and receiving, for building surplus capacity and holding surplus inventories. The impact of the bullwhip effect is also to decrease ‘Level of Product Availability’, since More run out of stocks in supply chain, and to decrease ‘Relationship Across the Supply Chain’, since each stage tends to blame other stages of the supply chain.

There are so many minor causes which gives rise to bullwhip effect. But, they can never be quantified through mathematical equations, however, can be controlled through effective managerial levers. Some of these causes can be pointed out as below,

- Lack of supply chain coordination
- Lack of information sharing
- Lack of trust among the members in SC
- Lack of proper incentive scheme
- Lack of proper trained sales forces etc

The major causes which increase in variability are projections of future demand expectations, which result in over-exaggerated responses to changes in demand.
In 1997 Lee et al. identified five major causes of the bullwhip effect which was all the consequence of the rational behavior of the supply chain members:

They are the use of

- Demand Forecasting
- Batch purchasing OR Ordering Lots
- Replenishment lead times
- Rationing & Supply Shortages
- Price Fluctuations and Safety Stock

The loss due to this can be quantified through mathematical equations, and can be controlled effectively, if the factors affecting the bullwhip effect are analyzed properly through proper method.

All previous works were only limited on quantifying the bullwhip effect based on common methods of reducing its impact. However, with all these previous works, it is difficult to obtain graphical illustration of the bullwhip effect.

II. RELATED WORK

Lee et.al (1997) stated that the bullwhip effect occurred when the demand order variabilities in the supply chain were amplified as they moved up the supply chain. Distorted information from one end of a supply chain to the other could lead to tremendous inefficiencies. Companies could effectively counteract the bullwhip effect by thoroughly understanding its underlying causes. Industry leaders were implementing innovative strategies that posed new challenges: 1. integrating new information systems, 2. defining
new organizational relationships, and 3. implementing new incentive and measurement systems.

Fransoo et al. (2000) stated discussed increased demand variability in supply chains (the bullwhip effect) in the literature. The practical measurement of this effect, however, entailed some problems that had not received much attention in the literature and that had to do with the aggregation of data, incompleteness of data, the isolation of demand data for defined supply chains that were part of a greater supply web. This paper discussed those conceptual measurement problems and discusses experiences in dealing with some of these problems in an industrial project. Also presented empirical results of measurements of the bullwhip effect in two supply chains.

Merkuryev et al. (2002) described the impact of two different information sharing strategies – decentralized and centralized information – combined with two inventory control policies – min-max and stock-to-demand inventory control – on the bullwhip effect. To investigate and measure this impact, simulation models were developed using the Arena 5.0 software package for a four-stage supply chain, consisting of a single retailer, wholesaler, distributor and manufacturer. The experiments with the developed models were described and the results are analyzed.

Warburton et al. (2004) described the Bullwhip Effect as problematic: order variability increases as orders propagate along the supply chain. The fundamental differential delayed equations for a retailer’s inventory reacting to a surge in demand were solved exactly. Much of the rich and complex inventory behavior was determined by the replenishment delay. The analytical solutions agreed with numerical integrations and previous control theory results. Managerially useful ordering strategies were proposed. Exact expressions were derived for the retailer’s orders to the manufacturer, and the Bullwhip Effect raised naturally. The approach was quite general and applicable to a wide variety of supply chain problems.

Croson et al. (2005) proposed a new behavioral cause of the bullwhip, coordination risk, arising when players deviated from equilibrium to build inventory to protect against the perceived risk that others would not behave optimally. They tested two strategies to mitigate coordination risk: (1) holding additional on-hand inventory, and (2) creating common knowledge by informing participants of the optimal policy. Both strategies
reduced, but did not eliminate, the bullwhip effect. Holding excess inventory reduced order amplification by providing a buffer against the endogenous risk of coordination failure. Such coordination stock differed from traditional safety stock, which buffers against exogenous demand uncertainty. Surprisingly, neither strategy reduced supply-line underweighting. They concluded that the bullwhip could be mitigated but its behavioral cause appeared robust.

Mujaj et.al (2007) addressed the problem of increasing order variances in multi-tier supply chains. The majority of current approaches for reducing this problem, namely the bullwhip effect, relied on information sharing and/or cooperative planning in inter-organizational systems. Due to multiple barriers in implementing these approaches, they maintained the local autonomy of the participants in the supply chain and provided a multiagent-oriented solution to the problem. In particular, they designed an agent-based reverse pricing model for matching supply and demand between independent agents. They adopted reverse pricing for operational procurement decisions and matchmaking that could be automated to a large extent. They evaluated our proposal by conducting a simulation study using a multiagent-based simulation system, and showed that the novel approach results in a significant reduction of the bullwhip effect.

Chaharsooghi et.al (2008) stated that two-echelon supply chain, which included two products based on the following considerations, had been studied and the bullwhip effect was quantified. Providing a measure for bullwhip effect that enabled us to analyze and reduce this phenomenon in supply chains with two products was the basic purpose of this paper. Demand of products was presented by the first order vector autoregressive time series and ordering system was established according to order up to policy. Moreover, lead-time demand forecasting was based on moving average method because this forecasting method is used widely in real world. Based on these assumptions, a general equation for bullwhip effect measure was derived and there was a discussion about nonexistence of an explicit expression for bullwhip effect measure according to the present approach on the bullwhip effect measure. However, bullwhip effect equation was presented for some limited cases. Finally, bullwhip effect in a two-product supply chain was analyzed by a numerical example.
Duc et al. (2010) studied whether a third-party warehouse can help to reduce the bullwhip effect in a supply chain. We compare the bullwhip effect in a three-stage supply chain with one supplier, one third-party warehouse, and two retailers and that in a two-stage supply chain with one supplier and two retailers. As a result, they exactly quantified an upper bound on variance of order lead time when order lead times of the warehouse and the retailers had the same mean value. In addition, they showed that the bullwhip effect pooling exists (i.e., the existence of third-party warehouse is beneficial for mitigating the bullwhip effect in supply chains) regardless market shares of the retailers if the variance of order lead time of the warehouse is less than the upper bound value.

Barlas et al. (2011) discussed that Supply chain inventories were prone to fluctuations and instability. Known as the bullwhip effect, small variations in the end item demand created oscillations that amplified throughout the chain. By using system dynamics simulation, they investigated some of the structural sources of the bullwhip effect, and explored the effectiveness of information sharing to eliminate the undesirable fluctuations. Extensive simulation analysis was carried out on parameters of some standard ordering policies, as well as external demand and lead-time parameters. Simulation results showed that (i) a major structural cause of the bullwhip effect was isolated demand forecasting performed at each echelon of the supply chain, and (ii) demand and forecast sharing strategies could significantly reduce the bullwhip effect, even though they could not completely eliminate it. They specifically showed how each policy is improved by demand and forecast sharing.

Future research involves more advanced ordering and forecasting methods, modelling of other well-known sources of bullwhip, and more complex supply network structures.

III. MULTI-AGENT SYSTEM

Computers are not very good at knowing what to do: every action a computer performs must be explicitly anticipated, planned for, and coded by a programmer. If a computer program ever encounters a situation that its designer did not anticipate, then the result is not usually pretty—a system crash at best, multiple loss of life at worst. This mundane fact is at the heart of our relationship with computers. It is so self-evident to the computer literate that it is rarely mentioned.

And yet it comes as a complete surprise to those encountering computers for the first time. For the most part, we are happy to accept computers as obedient, literal, unimaginative
servants. For many applications (such as payroll processing), it is entirely acceptable. However, for an increasingly large number of applications, we require systems that can decide for themselves that they need to do in order to satisfy their design objectives. Such computer systems are known as agents.

Agents that must operate robustly in rapidly changing, unpredictable, or open environments, where there is a significant possibility that actions can fail are known as intelligent agents, or sometimes autonomous agents. Here are examples of recent application areas for intelligent agents:

- When a space probe makes its long flight from Earth to the outer planets, a ground crew is usually required to continually track its progress, and decide how to deal with unexpected eventualities. This is costly and, if decisions are required quickly, it is simply not practicable. For these reasons, organisations like NASA are seriously investigating the possibility of making probes more autonomous— giving them richer decision making capabilities and responsibilities.

- Searching the Internet for the answer to a specific query can be a long and tedious process. So, why not allow a computer program—an agent—do searches for us? The agent would typically be given a query that would require synthesising pieces of information from various different Internet information sources. Failure would occur when a particular resource was unavailable, (perhaps due to network failure), or where results could not be obtained.

The multi-agent system (M.A.S.) is a computerized system composed of multiple interacting intelligent agents within an environment. Multi-agent systems can be used to
solve problems that are difficult or impossible for an individual agent or a monolithic system to solve. Intelligence may include as given below:

- some methodic,
- functional,
- procedural or algorithmic search,
- find and
- processing approach.

Although there is considerable overlap, a multi-agent system is not always the same as an agent-based model (ABM). The goal of an ABM is to search for explanatory insight into the collective behavior of agents (which don't necessarily need to be "intelligent") obeying simple rules, typically in natural systems, rather than in solving specific practical or engineering problems. The terminology of ABM tends to be used more often in the sciences, and MAS in engineering and technology.

Agents communicate in order to achieve better the goals of themselves or of the society/system in which they exist. Note that the goals might or might not be known to the agents explicitly, depending on whether or not the agents are goalbased. Communication can enable the agents to coordinate their actions and behavior, resulting in systems that are more coherent.

Coordination is a property of a system of agents performing some activity in a shared environment. The degree of coordination is the extent to which they avoid extraneous activity by reducing resource contention, avoiding livelock and deadlock, and maintaining applicable safety conditions. Cooperation is coordination among nonantagonistic agents, while negotiation is coordination among competitive or simply self-interested agents. Typically, to cooperate successfully, each agent must maintain a model of the other agents, and also develop a model of future interactions. This presupposes sociability.

IV. MULTI-AGENT SYSTEM BASED SUPPLY CHAIN SYSTEM

A multi-agent approach is applied for designing the system in order to deal with the complexity of the domain and to provide flexibility regarding the system architecture. Indeed, agent technology has become the most popular tool for designing distributed SCM systems as it provides an adaptable and dynamic way for managing separate links within the chain.
Unlike centralised approaches, agent-based SCM systems can respond quickly to changes and disturbances (either internal or external) through local decision making. Another advantage of designing the SCM solution as a multi-agent system (MAS) is that it allows different tasks within the SCM to be separated and explored both independently and in relation to each other. This feature is particularly important for the project presented in the thesis, as the research is mainly focused on one side of the supply chain, namely, its demand part, which deals with selling products to customers. The main problem which sellers are facing when managing their supply chains is of deciding on the details of offers to be made to customers: which prices to set, how many items, when and to whom to sell in order to increase profit, and when to sell available stocks without being penalized for late deliveries at the same time. The ability to predict market prices is crucial for developing better selling strategies. The task is not easy to solve in the context of e-Commerce, where prices are established dynamically. Using the principles of the MAS, this problem is studied in the thesis: a number of predictive models are deployed in the Demand agents of the MAS, and their effect on the behaviour of other internal agents as well as the whole system is investigated. The principles of the multi-agent approach help to meet the requirements of SCM in the following way:

- **Distributed**: The functions of SCM are divided among a set of separate, asynchronous software agents.
- **Dynamic**: Each agent performs its functions asynchronously as required, as opposed to in a batch or periodic mode.
- **Intelligent**: Each agent is an “expert” in its function; it uses artificial intelligence and operations research problem-solving methods.
- **Integrated**: Each agent is aware of and can access the functional capabilities of other agents.
- **Responsive**: Each agent is able to ask for information and/or a decision from another agent - each agent is both a client and a server.
- **Reactive**: Each agent is able to respond to events as they occur, modifying its behaviour as required, as opposed to responding in a pre-planned, rigid, batch manner.
- **Cooperative**: Each agent can cooperate with other agents in finding a solution to a problem - that is, they do not act independently.

- **Interactive**: Each agent may work with people to solve a problem.

- **Anytime**: No matter how much time is available, an agent is able to respond to a request, but the quality of the response is proportional to the time given to respond.

- **Complete**: The total functionality of the agents must span the range of functions required to manage the supply chain.

- **Reconfigurable**: The SCM system itself must be adaptable and must support the “relevant subset” of software agents.

- **General**: Each agent must be adaptable to as broad a set of domains as possible.

- **Adaptable**: Agents need to quickly adapt to the changing needs of the human organization. For example, adding a resource or changing inventory policy should be quick and easy for the user to do.

- **Backwards Compatible**: Agents need to have a seamless upgrade path so that the release of new or changed features does not compromise existing integration or functionality.

Various type of hybrid agent is used here which describe as below:

- **Demand Agent** used for handling the point-of-sale data and provide the information to central agent.

- **Forecasting Agent** used for analysing the market and upcoming trends and up and down and gives a optimistic forecasting values.
Figure 4: Forecasting Agent

- **Supply Agent** used for sequencing and handling the order data with high managing skills and control supply for the manufacturing process.

Figure 5: Supply Agent

- **Delivery Agent** used for studying everyday price of market and mention the peak high and deliver the product below the price in day and their reasons and provide this information to central agent.

Figure 6: Delivery Agent

- **Stock Control Agent (SCA)** used for handling the stock and control each steps of stocking related to market demand and up-down, and all these work done by sharing the information to central agent.
• Inventory Agent (IA) used for handling the inventory control system.

![Figure 7: Inventory Agent](image)

• Central Information Managing Agent (CIMA) used for sharing the information from all the agent and provide the information which is required for particular agent.

• Production Agent (PA) used for handling the production of the product which is related to supply chain agent, because production low and high demand also cause the bullwhip.

![Figure 8: Scheduling Agent](image)
V. BULLWHIP EFFECT REDUCTION

Supply Chain encompasses all those activities needed to design, manufacture and deliver a product or service needs a mechanism or framework for information sharing. Agent-based manufacturing is a new way of thinking about and applying information. With this idea an effort is made to provide a multi agent system model for the supply chain management in order to reduce the bull whip effect. Basic interface is given below:

Figure 9: MAS based SCM system

Figure 10 GUI of the JADE RMA
In the proposed model (bull whip effect model) each agent performs a specific function of the organization and share the information with information agent. There by the most important requirement of successful supply chain i.e information sharing is achieved besides controlling the demand-supply trouble in the proposed model. In the current work a part of the model related to control agent is designed and it may be the first ever such system to reduce the bull whip effect

VI. CONCLUSION

Supply Chain networks are multi stage complex dynamical systems consist of various involved organizations performing different processes and activities in each and consequent stages which are connected through upstream and downstream linkages to produce value in the form of products and services Demand forecasting and decision making processes are among the key activities which directly affect the performance of this complex systems. The variability of the demand information between the stages of Supply Chain networks and the increase in this variability as the demand data moves upstream from the customer to the consequent stages which cause the bull whip, and there are lot of buff whip calling factor is available in supply chain, so control these factor proposed multi-agents are being designed, and it control very effectively and reduce the bull whip in the supply chain networks

REFERENCES


[9] Dragana Makajic, Biljana, Bull whip effect and supply chain modeling and analyzing using CPN Tools, Operation Research Laboratory, University of Belgrade.