# International Journal of Computer Science and Security (IJCSS)

ISSN : 1985-1533

VOLUME 3, ISSUE 2

PUBLICATION FREQUENCY: 6 ISSUES PER YEAR

**Editor in Chief Dr. Haralambos Mouratidis**

# International Journal of Computer Science and Security (IJCSS)

Book: 2009 Volume 3, Issue 2
Publishing Date: 31-10-2009
Proceedings
ISSN (Online): 1985-1553



IJCSS Journal is a part of CSC Publishers
http://www.cscjournals.org


Published in Malaysia

Typesetting: Camera-ready by author, data conversation by CSC Publishing Services – CSC Journals, Malaysia

**CSC Publishers**

# Table of Contents

Volume 3, Issue 2, April 2009.

# Behavior Based Anomaly Detection Technique to Mitigate the Routing Misbehavior in MANET

**T.V.P.Sundararajan** tvpszen@yahoo.co.in
*Assistant Professor/Department of Electronics and Communication Engineering*
*Bannari Amman Institute of Technology*
*Sathyamangalm-638401, Tamilnadu,, India*

**Dr. A.Shanmugam** dras@yahoo.co.in
*Principal*
*Bannari Amman Institute of Technology*
*Sathyamangalm-638401, Tamilnadu,, India*

**ABSTRACT**

Mobile ad hoc network does not have traffic concentration points such as gateway or access points which perform behavior monitoring of individual nodes. Therefore, maintaining the network function for normal nodes when other nodes do not route and forward correctly is a big challenge. This paper, address the behavior based anomaly detection technique inspired by the biological immune system to enhance the performance of MANET to operate despite the presence of misbehaving nodes. Due to its reliance on overhearing, the existing watchdog technique may fail to detect misbehavior or raise false alarms in the presence of ambiguous collisions, receiver collisions, and limited transmission power. Our proposed scheme uses intelligent machine learning techniques that learns and detects each node by false alarm and negative selection approach. We consider DSR, AODV and DSDV [1] as underlying routing protocol which are highly vulnerable to routing misbehavior. Analytical and simulation results are presented to evaluate the performance of the proposed scheme.

**Keywords:** intrusion detection, anomaly detection, mobile ad hoc network, security.

## 1. INTRODUCTION

A Mobile Ad Hoc Network (MANET) is a collection of mobile nodes (hosts) which communicate with each other via wireless links either directly or relying on other nodes as routers. The network topology of a MANET may change rapidly and unpredictably. In a MANET, different mobile nodes with different goals share their resources in order to ensure global connectivity. However, some resources are consumed quickly as the nodes participate in the network functions. For instance, battery power is considered to be most important in a mobile environment. An individual mobile node may attempt to benefit from other nodes, but refuse to share its own resources. Such nodes are called selfish or misbehaving nodes and their behavior is termed selfishness or misbehavior [2]. One of the major sources of energy consumption in the mobile nodes of MANETs is wireless transmission [3]. A selfish node may refuse to forward data packets for other nodes in order to conserve its own energy.

In order to mitigate the adverse effects of routing misbehavior, the misbehaving nodes need to be detected so that these nodes can be avoided by all well-behaved nodes. In this paper, we focus on the following problem:

**1.1 Misbehavior Detection and Mitigation:** In MANETs, routing misbehavior can severely degrade the performance at the routing layer. Specifically, nodes may participate in the route discovery and maintenance processes but refuse to forward data packets. How do we detect such misbehavior? How can we make such detection processes more efficient (i.e., with less control overhead) and accurate (i.e., with low false alarm rate and missed detection rate)?

The existing two extensions to the Dynamic Source Routing algorithm (DSR) [4] to mitigate the effects of routing misbehavior: the watchdog and the path rater. In this technique, Watchdog's weaknesses are that it might not detect a misbehaving node in the presence of 1) ambiguous collisions, 2) receiver collisions, 3) limited transmission power, 4) false misbehavior, 5) collusion, and 6) partial dropping.

In this paper we explore a behavior based intrusion detection techniques which assume that an intrusion can be detected by observing deviations from the normal or expected behavior of the nodes. The intrusion detection systems compare this behavior model with activities of normal node. When the deviation is observed, an alarm is generated.

We have implemented behavior based anomaly detection to the underlying DSR, AODV and DSDV Source routing algorithms. The basic idea of the behavior-based approach involves Negative Selection Algorithm (NSA).The detectors are capable of distinguishing well-behaving nodes from the misbehaving nodes with a good degree of accuracy. The False positives (or False Alarms) could be minimized to good extent though some False Negatives exist because of subtle differences between good and bad behaviors in this experimentation.

The rest of the paper is organized as follows: In Section 2, we summarize the various approaches for router misbehavior detection and mitigation that have been proposed and studied in the literature. In Section 3, we discuss the details of Negative Selection Algorithm (NSA), and section 4 describes router misbehavior and attacking scenarios. In Section 5, we present simulation environment, performance metrics and discuss our simulation results that compare which protocol is better for behavior based detection. We conclude the work in Section 6.

## 2. RELATED WORK

The security problem and the misbehavior problem of wireless networks including MANETs have been studied by many researchers, e.g., [4], [5], [6], [7]. Various techniques have been proposed to prevent selfishness in MANETs. These schemes can be broadly classified into two categories: credit-based schemes and reputation-based schemes.

### 2.1 Credit-Based Schemes

The basic idea of credit-based schemes is to provide incentives for nodes to faithfully perform networking functions. In order to achieve this goal, virtual (electronic) currency or similar payment system may be set up. Nodes get paid for providing services to other nodes. When they request other nodes to help them for packet forwarding, they use the same payment system to pay for such services[8], [9], [10].The main problem with credit-based schemes is that they usually require some kind of tamper-resistant hardware and/or extra protection for the virtual currency or the payment system. We focus on reputation-based techniques in this paper instead.

### 2.2 Reputation-Based Schemes

The second category of techniques to combat node misbehavior in MANETs is reputation-based [11]. In such schemes, network nodes collectively detect and declare the misbehavior of a suspicious node. Such a declaration is then propagated throughout the network so that the misbehaving node will be cut off from the rest of the network.

In [12], Marti et al. proposed a scheme that contains two major modules, termed watchdog and pathrater, to detect and mitigate, respectively, routing misbehavior in MANETs.Nodes operate in a promiscuous mode wherein the watchdog module overhears the medium to check whether the next-hop node faithfully forwards the packet. At the same time, it maintains a buffer of recently sent packets. A data packet is cleared from the buffer when the watchdog overhears the same packet being forwarded by the next-hop node over the medium. If a data packet remains in the buffer for too long, the watchdog module accuses the next hop neighbor of misbehaving. Thus, the watchdog enables misbehavior detection at the forwarding level as well as the link level. Based on the watchdog's accusations, the path rater module rates every path in its cache and subsequently chooses the path that best avoids misbehaving nodes. Due to its reliance on overhearing, however, the watchdog technique may fail to detect misbehavior or raise false alarms in the presence of ambiguous collisions, receiver collisions, and limited transmission power, as explained in [12].

The CONFIDANT protocol proposed by Buchegger and Le Boudec in [13] is another example of reputation-based schemes. The protocol is based on selective altruism and utilitarianism, thus making misbehavior unattractive. CONFIDANT consists of four important components—the Monitor, the Reputation System, the Path Manager, and the Trust Manager. They perform the vital functions of neighborhood watching, node rating, path rating, and sending and receiving alarm messages, respectively. Each node continuously monitors the behavior of its first-hop neighbors. If a suspicious event is detected, details of the event are passed to the Reputation System. Depending on how significant and how frequent the event is, the Reputation System modifies the rating of the suspected node. Once the rating of a node becomes intolerable, control is passed to the Path Manager, which accordingly controls the route cache. Warning messages are propagated to other nodes in the form of an Alarm message sent out by the Trust Manager. The Monitor component in the CONFIDANT scheme observes the next hop neighbor's behavior using the overhearing technique. This causes the scheme to suffer from the same problems as the watchdog scheme.

There exist some wireless intrusion detection systems WLAN IDS [14] that encompasses components such as data collection, intrusion detection and an additional secure database to log anomalies. Another IDS were designed to secure wireless networks (WLAN). One such IDS [15] suggest to detect intrusions such as abnormal routing table updates and attacks at the MAC layer.

## 3. Negative Selection Algorithm (NSA)

The Negative Selection Algorithm (NSA) is based on the principles of self/non-self discrimination in the immune system. It can be summarized as follows:

1. Define self as a collection S of elements in a feature space X, a collection that needs to be monitored. For instance, if X corresponds to the space of states of a system represented by a list of features, S can represent the subset of states that are considered as normal for the system.
2. Generate a set F of detectors, each of which fails to match any string in S.
3. Monitor S for changes by continually matching the detectors in F against S. If any detector ever matches, then a change is known to have occurred, as the detectors are designed not to match any representative samples of S.

In this work, we propose a hybrid approach for misbehavior detector generation

### 3.1 Anomaly detection

The anomaly detection process aims at distinguishing a new pattern as either part of self or non- self, given a model of the self (normal data) set. The problem space, denoted by X in an n - dimensional space; the self set is denoted as S and let N be the complementary space of S It is assumed that each attribute is normalized to [0, 1],

then $S \subseteq [0,1]^n$ $S \cup N = X$, $S \cap N = \Phi$

Given the normal behavior of a system S the characteristic function of S defined

as $N_S(p) = \begin{cases} 1, p \in S \\ 0, p \in N \end{cases}$ is used to distinguish between self and non-self

In order to generate detectors through an evolutionary process, we used a structured genetic algorithm (sGA), which is suitable for encoding different detector shapes [16].

### 3.2 The Structured GA

A structured GA (sGA) is a type of evolutionary algorithm [17] that incorporates redundant genetic material, which is controlled by a gene activation mechanism. It utilizes multi-layered genomic structures for its chromosome i.e. all genetic material (expressed or not) is 'structured' into a hierarchical chromosome. The activation mechanism enables and disables these encoded genes. The implicit redundancy has the advantages of maintaining genetic diversity necessary in solving complex search and optimization applications. The capacity to maintain such diversity however depends on the amount of redundancy incorporate in the structure.

The sGA as shown in Figure.1 interprets the chromosome as a hierarchical structure; thus, genes at any level can be either active or passive, and high-level genes activate or deactivate sets of low-level genes. Thereby, the dynamic behavior at any level, whether the genes will be expressed phenotypically or not, is governed by the high level genes.

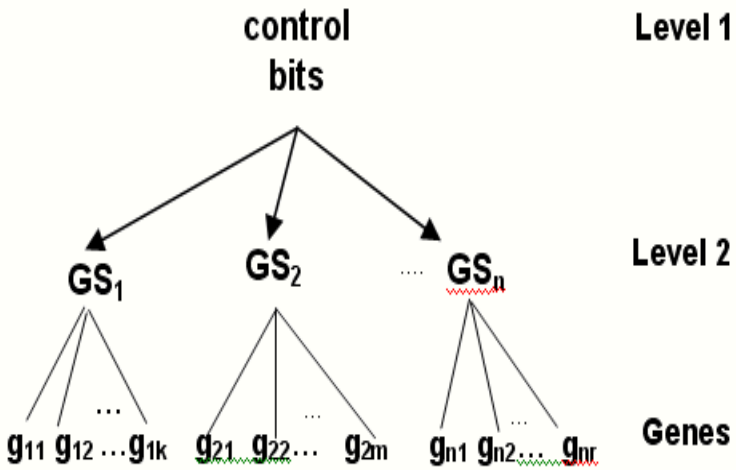



**FIGURE 1:** Structured GA representation of a chromosome with n different gene sets

This illustration shows that the representation scheme for a chromosome tree has a control gene activating one of the shapes in phenotype space where each shape identifies a detector shape. A detector is defined in an n- dimensional space as a geometrical shape, such as a hyper sphere, a hyper-rectangle or a hyper-ellipse. The matching rule is expressed by a membership function associated with the detector, which is a function of the detector-

sample pattern distance [18] (Euclidean or any other distance measure). A set of good samples (also known as self) represented by n- dimensional points are given as inputs to the algorithm.

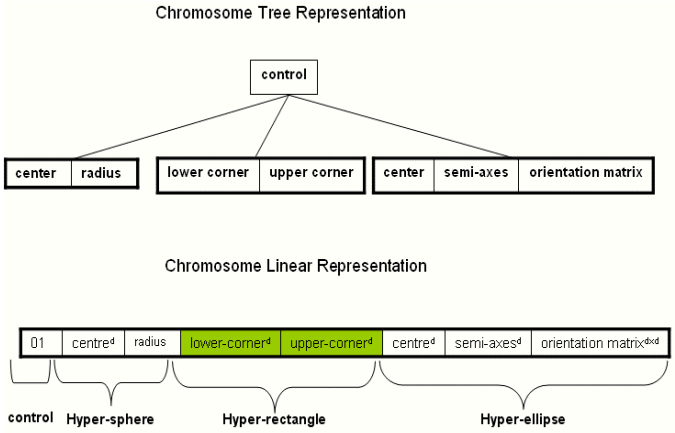



**FIGURE 2:** Encoding multi-shaped detectors: a chromosome having high level control and low level parameters for three different shapes: hyper spheres, hyper-rectangles and hyper-ellipses detectors.

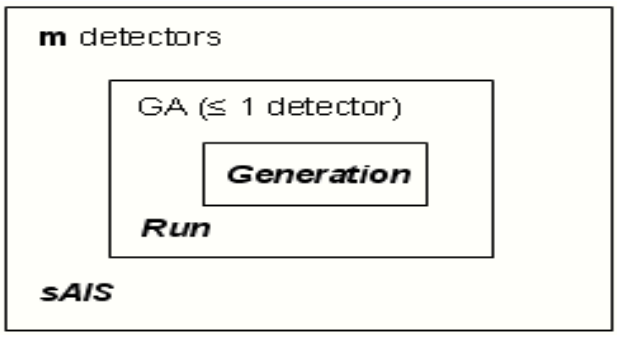



**FIGURE 3:** General framework for detector generation

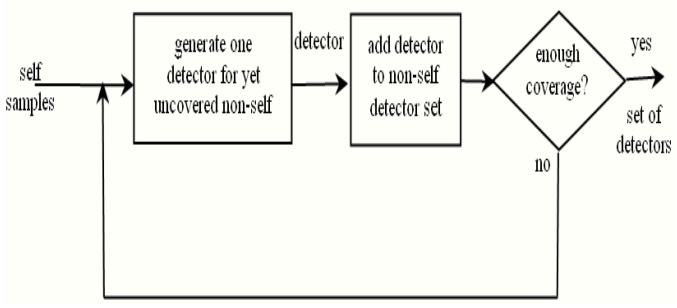



**FIGURE 4:** Modular subdivision of the detector generation process

As depicted in Figure 3, the goal of the algorithm is to evolve a set of detection rules to cover the non-self space. The iterative process to generate a set of detectors (Figure 4) is driven by two main goals :Minimize overlap with self, and Make the detectors as large as possible and keep them separate from each other, in order to maximize the non- self covering (This is referred to as the coverage parameter in all our experiments).In this work, we assumed that the self data points representing the events of good traffic network behavior while non-self data points represent the misbehaving event sequences.

## 4. PROBLEM OF ROUTING MISBEHAVIOR

In this section, we describe the problems caused by routing misbehavior. Then , we illustrates detection of flooding attacks, and DoS attack scenarios.

### 4.1 Routing Misbehavior Model

We present the routing misbehavior model considered in this paper in the context of the DSR,AODV and DSDV protocol . Due to its popularity and recognized by IETF MANET group, we use these routing protocols to illustrate our proposed anomaly detection add-on scheme. The details of DSR,AODV and DSDV can be found in [5].

We focus on the following routing misbehavior: A selfish node does not perform the packet forwarding function for data packets unrelated to itself. However, it operates normally in the Route Discovery and the Route Maintenance phases of the routing protocol. Since such misbehaving nodes participate in the Route Discovery phase, they may be included in the routes chosen to forward the data packets from the source. The misbehaving nodes, however, refuse to forward the data packets from the source. This leads to the source being confused. The existence of a misbehaving node on the route will cut off the data traffic flow. The source has no knowledge of this at all.

In this paper, we propose the intelligent machine learning technique to detect such misbehaving nodes. Routes containing such nodes will be eliminated from consideration. The source node will be able to choose an appropriate route to send its data.

### 4.2 Detecting Router Misbehavior

Wireless Ad hoc networks using routing protocol such as DSR, AODV and DSDV are highly vulnerable to (packet) routing misbehavior due to misbehaving, faulty or compromised nodes . We describe detection of flooding attacks and two attack scenarios.

### 4.3 Detecting Flooding Attacks

From the work of [19] it was seen that 802.11b wireless networks suffers from some inherent flaws and are therefore prone to more attacks than wired networks because there is no need of any physical access to wireless networks. We first show how the above approach can be used to detect flooding attacks. We implemented two familiar attacks based on the guidelines in our previous work [19]. The two attacks are Denial of Service attack from an attacker machine outside the wireless network, and Denial of Service attack from a compromised machine inside the wireless network. The first attack was launched in a simulation environment consisting of some wireless stations and an Access Point (Figure 5), while the second one is implemented using a network simulator tool called Qualnet 4.5 (Figure 6).

The detection results indicate that in all the three cases the Negative Detectors were able to detect the attacks with good detection rate. We briefly illustrate the attack scenario and the results for the Ad-hoc network case. The simulation scenario comprises of an Ad-hoc WLAN with three wireless stations communicating with each other (Figure 5), and established normal traffic flow. An attacker machine lunches a PING flood attack to a wireless node This attack is launched by setting DoS attack in the configuration setting .

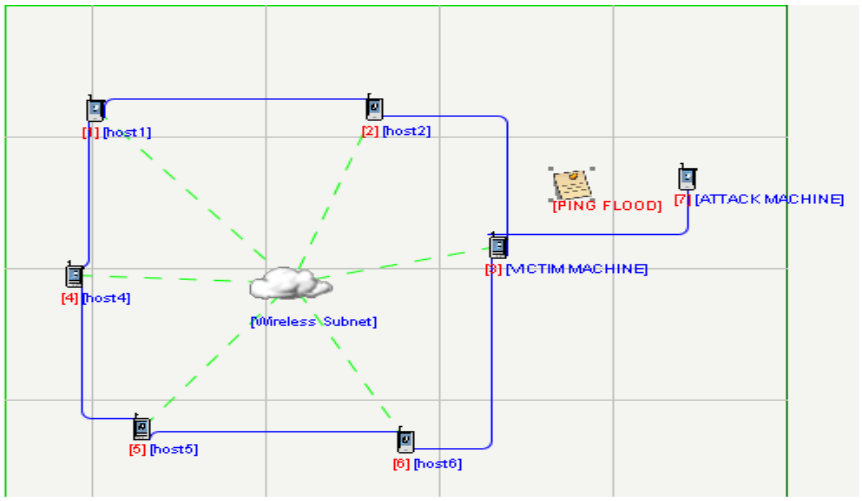



**FIGURE 5:** Ping Flood DoS in Ad-Hoc Network from an attacker machine

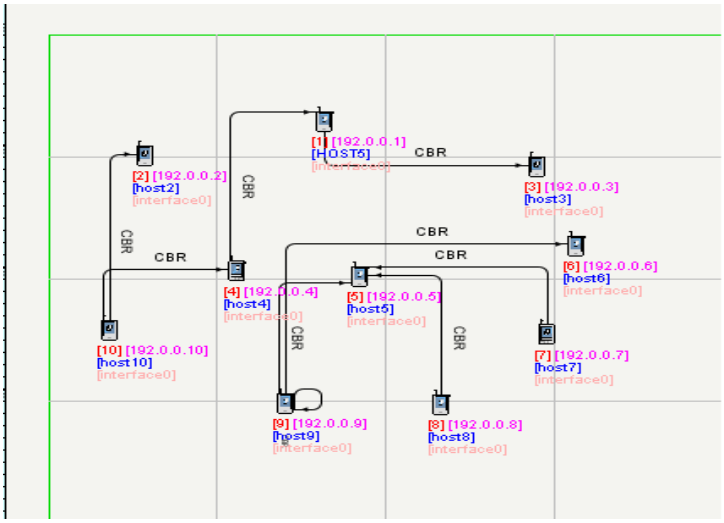



**FIGURE 6:** DoS attack by a compromised node in the Wireless network implemented in Qualnet 4.5

The ICMP (Internet Control Message Protocol) packet sequence numbers are collected at the Ad-hoc node different from the victim machine, which keeps pinging, on the victim machine throughout the experiments. The time taken for each packet to reach the destination node is noted when each ping (ICMP) packet is sent. It is observed that during the occurrence of an attack, there are some drops in the ICMP packets. In addition, the time taken to reach the destination host increased when the attack was launched.

A network simulator (Qualnet version 4.5) is used to simulate an Ad-hoc network. Figure 6 shows a scenario with 10 nodes and the traffic flow among them. These nodes are labeled, ranging from Node 0 to Node 9. Constant Bit Rate (CBR) traffic is defined between Node 0 and Node 2 Node 3 to Node 4, Node 4 to Node6, Node 5 to Node 3 Node 6 to Node 7,

Node 1 to Node 8, Node 9 to Node 2, Node 8 to Node 7 and File Transfer Protocol (FTP) traffic flows between Node 1 and Node2. The start times for all these traffics are preset. The attack is launched from Node 0 to Node 2. The attack is simulated as DoS with heavy traffic flow in a short time interval. During these periods when the attack was launched, the number of legitimate packets received by the victim node (Node 2) was reduced The sequence numbers resulting from the connection between different nodes and Node 2 were collected.

We are using this approach in order to test our detection approach, although a more realistic approach (the complete network profile rather than monitoring a single node) is used in this work for misbehavior detection [20],[21]. The primary objective was to evaluate our model in terms of good detection rates and low alarm rates for wireless ad-hoc network's routing layer.

## 5. Simulation Environment

Qualnet simulator [22] provides a scalable simulation environment for large wireless and wireline communication networks. Its scalable architecture supports up to thousand nodes linked by a heterogeneous communications capability that includes multi-hop wireless communications using ad-hoc networking.

| Simulation System parameters | |
|---|---|
| Parameter | Default value(s) |
| Routing protocol | DSR,AODV and DSDV |
| Simulation time | 60 minutes |
| Simulation area in meters | 800x1000 |
| Number of nodes | 60 |
| Radio range | 380 m |
| Propagation Path loss model | Two-ray |
| Mobility model | Random |
| Mobility speed (no pauses) | 1m/s |
| Misbehaving nodes | 5, 10, 20 |
| Traffic type | telnet,CBR |
| Payload size | 512 bytes |
| Frequency/rate | 0.2-1s |
| Radio-Bandwidth/link speed | 2Mbps |

**Table 1** : Simulation System parameters

Qualnet simulator version 4.5 from has been used to analyze the reactive routing protocols DSR,AODV and DSDV. The under lying MAC protocol defined by IEEE 802.11 was used Traffic sources of both continuous bit rate (CBR) based on TCP for 10 sources were generated. The CBR and TCP mobility scenario of 20 nodes with a maximum speed of 20 seconds and for a simulation area of 500 × 500 with 4.0 kbps was generated. Each model had five scenario files

generated with different pause times of 0, 10, 30, 40, and 50 seconds.

| Detection System parameters | |
|---|---|
| Parameter | Default value(s) |
| Upper limit for Events sequence sets of a Monitored Node for learning | 500 |
| Number of subsequences in a sequence set | 4 |
| Upper limit for the number of events in a sequence set | 40 |
| Upper limit for a sequence set collection | 10s |
| Misbehavior probability | 0.8 |
| Two-ray Learning data threshold | 0.001 - 0.1 |
| Threshold for detection (% of Detection rate) | 0.25 |
| Mutation probability | 0.05-0.1 |
| Crossover probability | 0.6 |
| Normalized space range | [0.0, 1.0] |
| Number of dimensions | 4, 2 |

**Table 1 :** Detection System Parameters

### 5.1 Experimental Details

The incorporation of misbehavior into the network is the same as done in [19]. We reiterate for clarity. The nodes can be set to misbehave as a Boolean parameter. It can be set or reset. Using this implementation capability we could have different numbers of misbehavior set up (In our experiments, 5 10 and 20 were involved). The misbehavior are implemented in two ways - (1) Nodes neither forward route requests nor answer the route replies from their route cache. (2) The nodes do not forward data packets. The misbehavior probability is a control parameter such that the misbehaving nodes behave badly only during certain times in the simulation run. The same or different probabilities could be utilized in either case. We used an implementation of the routing packets traffic data using DSR ,AODV and DSDV protocol, in Qualnet [22] that provides an excellent environment for wireless mobile ad-hoc networks. For data set collection, detection and analysis, crucial simulation and detection parameters, as defined in Table 1 and 2 were used.

### 5.2 Performance measures

The experimental simulation aims at finding and reporting the detection behavior of the generated nodes in correctly identifying the misbehaving nodes as well as how well it could identify such deviations in behavior. Our experimental results are based on the following metrics.

1. Average detection rates for the misbehaving nodes is defined as detection rate (D.R.) = (true positives)/ (true positives + false negatives).
2. Average False Alarm rates for misclassifying the well behaving nodes is defined as false alarm rate (F.A.R) = (false positives) / (false positives + true negatives).

where, true positives refer to the number of abnormal cases identified as abnormal in the given data set of vector points. False positives refers to the number of normal cases mistakenly identified as abnormal; true negatives refer to the number of normal event sequences (normal cases) in the entire sequence correctly identified as normal while false negatives are the count of the number of those abnormal sequences that the detector set classified as normal. Whenever and wherever we refer to positive detection and misclassifications, we refer to these metrics respectively.

**5.3 Simulation Results and Analysis:**

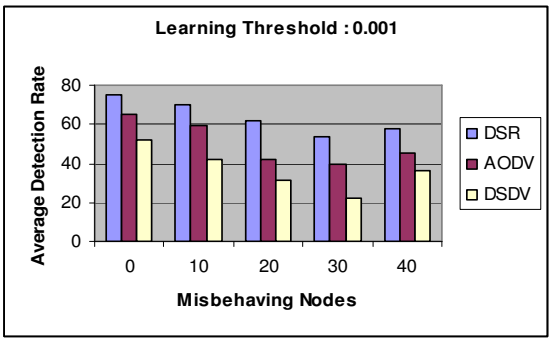
Based on the Performance metrics, we present the following results here for clarity. To study the feasibility of our proposed detection scheme, we have implemented behavior based anomaly detection in a network simulator and conducted a series of simulations to evaluate its effectiveness. We choose three specific ad-hoc wireless protocols as the subjects of our study. They are DSR protocol, AODV protocol, and DSDV protocol. They are selected as they represent different types of ad-hoc wireless routing protocols, proactive and on-demand. We now show how our anomaly detection methods can be applied to these protocols and demonstrate the effectiveness of our models can be used on other different scenarios.

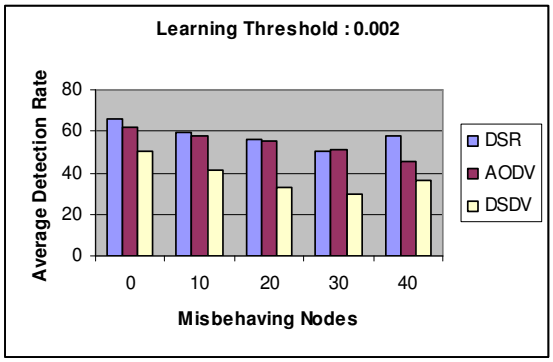
It is interesting to observe that DSR with its results outperforms AODV, DSDV protocols a lot, while the DSDV is the worst. The simulation results as shown in figure.7 and figure.8 demonstrate that an behavior based anomaly detection approach can work well on different wireless ad-hoc networks. That is, the normal behavior of a routing protocol can be established and used to detect misbehaving nodes.

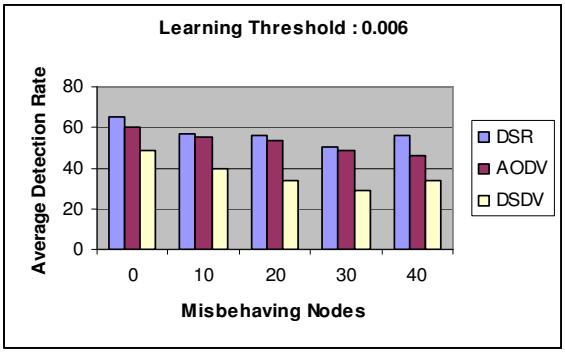
Having done the simulations on three ad-hoc routing protocols, we now attempt to answer this question - which type of protocol is “better" for misbehavior detection. Our solution tends to prefer DSR and AODV,even in the first look its route update is not as ‘regular" as DSDV. After detail analysis of these protocols, we believe that misbehavior detection works better on a routing protocol in which a degree of redundancy exists within its infrastructure. DSR embeds a whole source route in each packet dispatched, hence making it harder to hide the intrusion by faking a few routing information. We call this a path redundancy. Further, the DSR and AODV route update depends on traffic demand, which makes it possible to establish relationships between routing activities and traffic pattern. We call this a pattern redundancy. DSDV, in contrast, has a very weak correlation between control traffic and data traffic, even when we preserve the traffic feature. Note that DSR and AODV are both on-demand protocols. We therefore believe that those types of redundancy have helped on-demand protocols to have a better performance.

**Detection capabilities** : In all these experiments, average detection rates as shown in figure.7 were substantially higher than the detection threshold. This demonstrates that all misbehaving nodes were detected and classified as misbehaving while the well-behaving nodes were classified as normal. Depending on the detector sets the average rates of detection and average false alarm rates (misclassifications) as in figure.8 were close to the median of the best and worst cases. We show that average false alarm rate as a function of number of misbehaving nodes. It can be observed that the average false alarm rate reduces as number of misbehaving nodes increases for DSR,AODV and DSDV protocols. However, false alarm rates of DSR is greater than AODV,DSDV for various threshold values. This is due to the fact that routes are broken frequently when the network is under the influence of Denial of Service attacks.

**Impact of the learning threshold parameter** : It is clearly discernable from the experimental results that most of the misbehaviors are subtle and hence difficult to distinguish from the benevolent behaviors. This results in high false negatives thus lowering the detection rates. Thus a lower threshold value (0.001). has a higher detection rate compared to the higher ones For a given learning threshold value, the number of misbehaving nodes also play a distinguished role. As the number of misbehaving nodes increases, the detection rates decrease slightly.


Learning Threshold : 0.001
Average Detection Rate
80
60
40
20
0
0
10
20
30
40
Misbehaving Nodes
DSR
AODV
DSDV


Learning Threshold : 0.002
Average Detection Rate
80
60
40
20
0
0
10
20
30
40
Misbehaving Nodes
DSR
AODV
DSDV


Learning Threshold : 0.006
Average Detection Rate
80
60
40
20
0
0
10
20
30
40
Misbehaving Nodes
DSR
AODV
DSDV

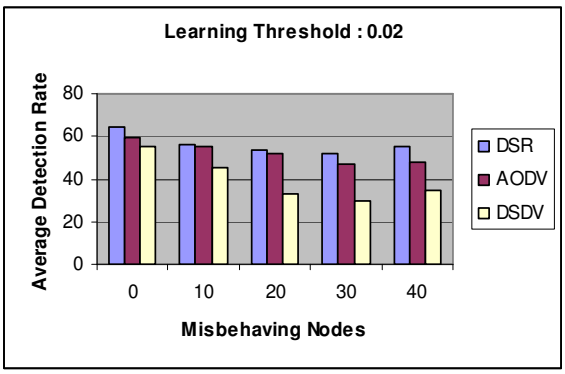
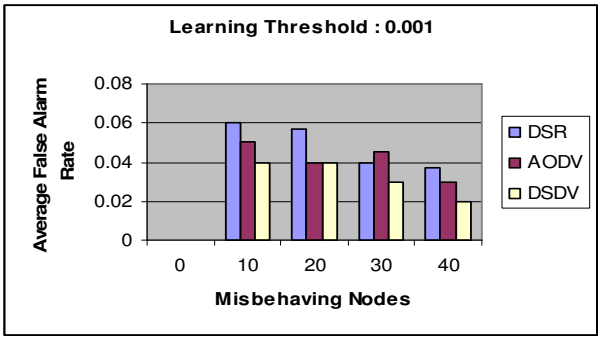
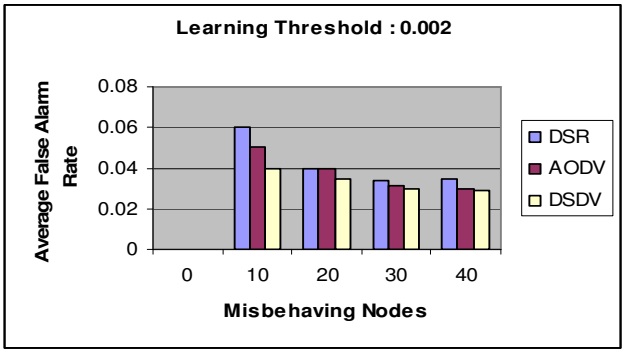
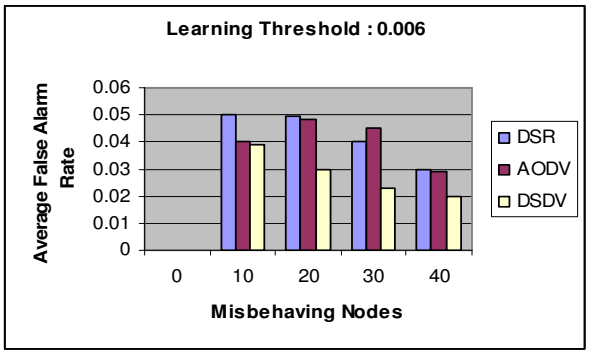

Learning Threshold : 0.02
Average Detection Rate
80
60
40
20
0
0
10
20
30
40
Misbehaving Nodes
DSR
AODV
DSDV

**FIGURE 7:** Detection Performance of DSR, AODV and DSDV



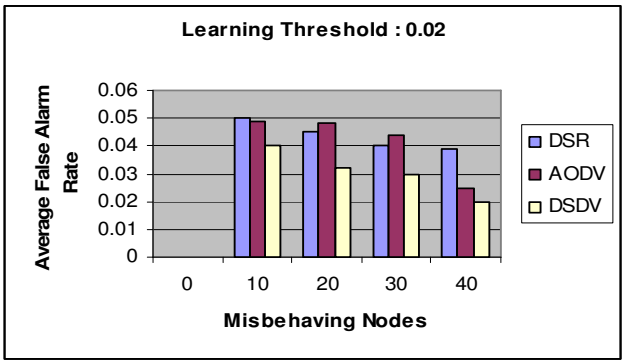




**FIGURE 8:** False Alarm Rates of DSR, AODV and DSDV

## 6. CONCLUSIONS AND FUTURE WORK

We proposed to use behavior based anomaly detection learning technique in which learning of good behavior node and detection of misbehavior node is carried out in a simulated ad-hoc wireless environment. The cooperative approach generated multi-shaped detectors (a set of rules) defining a boundary for the unknown regions of the behavior space. In certain cases, it would require a large amount of training (good) data as well as the number of efficient detectors[23],[24]. Sometimes, for very subtle misbehavior detection, more detectors with better coverage are necessary. We used a three set of protocol like DSR, AODV and DSDV, to test the effect of our proposed approach. With the use of learning threshold for the detectors learning, certain subtle abnormalities are supposedly captured. We have found that the anomaly detection works better on DSR rather than AODV and DSDV. However performance of the DSR is very sensitive to some parameters that require careful tuning. More specifically, on-demand protocols usually work better than table-driven protocols because the behavior of on-demand protocols reflects the correlation between traffic pattern and routing message ows[25].The preprocessing of raw data for matching the misbehavior needs to be carefully analyzed.

It is more difficult to decide the behavior of a single node. This is mainly due to the fact that communication takes place between two nodes and is not the sole effort of a single node. Therefore, care must be taken before punishing any node associated with the misbehaving links. When a link misbehaves, either of the two nodes associated with the link may be misbehaving. In order to decide the behavior of a node and punish it, we may need to check the behavior of links around that node. This is a potential direction for our future work.

## 7. REFERENCES

[1]. C.Perkins and P.Bhagwat, "*Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers*," in proceedings of the ACM SIGCOMM'94 Conference on Communications Architectures, Protocol and Applications, London, UK, August 1994, pp.234-244.

[2] L. Buttyan and J.-P. Hubaux, "*Security and Cooperation in Wireless Networks*," http://secowinet.epfl.ch/, 2006.

[3] L.M. Feeney and M. Nilsson, "*Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc Networking Environment*," Proc. IEEE INFOCOM, 2001.

[4] L. Zhou and Z.J. Haas, "*Securing Ad Hoc Networks*," IEEE Network Magazine, vol. 13, no. 6, Nov./Dec. 1999.

[5] F. Stajano and R. Anderson, "*The Resurrecting Duckling: Security Issues in Ad-Hoc Wireless Networks*," Proc. Seventh Int'l Workshop Security Protocols, 1999.

[6] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang, "*Providing Robust and Ubiquitous Security Support for Mobile Ad-Hoc Networks*," Proc. IEEE Int'l Conf. Network Protocols (ICNP '01), 2001.

[7] I. Aad, J.-P. Hubaux, and E-W. Knightly, "*Denial of Service Resilience in Ad Hoc Networks*," Proc. MobiCom, 2004.

[8] L. Buttyan and J.-P. Hubaux, "*Enforcing Service Availability in Mobile Ad-Hoc WANs*," Proc. MobiHoc, Aug. 2000.

[9] J.-P. Hubaux, T. Gross, J.-Y. LeBoudec, and M. Vetterli, "*Toward Self-Organized Mobile Ad Hoc Networks: The Terminodes Project*," IEEE Comm. Magazine, Jan. 2001.

[10] L. Buttyan and J.-P. Hubaux, "*Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks*," ACM/Kluwer Mobile Networks and Applications, vol. 8, no. 5, 2003.

[11] S. Zhong, J. Chen, and Y.R. Yang, "*Sprite: A Simple, Cheat-Proof,Credit-Based System for Mobile Ad-Hoc Networks*," Proc. INFOCOM, Mar.-Apr. 2003.

[12] S. Marti, T. Giuli, K. Lai, and M. Baker, "*Mitigating Routing Misbehavior in Mobile Ad Hoc Networks*," Proc. MobiCom, Aug.2000.

[13] S. Buchegger and J.-Y. Le Boudec, "*Performance Analysis of the CONFIDANT Protocol: Cooperation of Nodes, Fairness in Dynamic Ad-Hoc Networks*," Proc. MobiHoc, June 2002.
[14] Y. Lim, T. Schmoyer, J. Levine and H. L. Owen. " *Wireless Intrusion Detection and Response*". In Proceedings of the 2003 IEEE workshop on Information Assurance United States Military Academy, NY: West Point.
[15] Y. Zhang and W. Lee. Au gust 6-11, 2000. "*Intrusion Detection in Wireless Ad-Hoc Networks*". In Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking, Boston: Massachusetts.
[16] S. Balachand ran, D. Dasgupta, F. Nino, D. Garrett, " *Framework for Evolving Multi-Shaped Detectors in Negative Selection*". Submitted to the IEEE Transactions on Evolutionary Computation, January 2006.
[17] D. Dasgupta and D. R McGregor, " *sGA: A structured Genetic Algorithm*". Research Report IKBS-11-93, April 1993.
[18] F. González, "*A study of Artificial Immune Systems Applied to Anomaly Detection*", . PhD. Dissertation, Advisor: Dr. Dipankar Dasgupta, The University of Memphis, May 2003.
[19] M. Kaniganti. "*An Agent-Based Intrusion Detection System for Wireless LANs*", Masters Thesis, Advisor: Dr. Dipankar Dasgupta. The University of Memphis, December 2003.
[20] S. Sarafijanovic and J.Y. Le Boudec. "*An Artificial Immune System for Misbehavior Detection in Mobile Ad-Hoc Networks with Virtual Thymus, Clustering, Danger Signal and Memory Detectors*". In Proceedings of ICARIS-2004 (Third International Conference on Artificial Immune Systems), pp . 342-356, Septemb er 13-16, 2004, Catania, Italy
[21] J. Kim and P.J. Bentley. "*The Artificial Immune Model for Network Intrusion Detection*", 7th European Conference on Intelligent Techniques and Soft Computing (EUFIT'99), Aachen, Germany.
[22]. Scalable Network Technologies, "*Qualnet simulator-version 4.5*," Software package 2008,[online]. Available : http://www.qualnet.com
[23] H. Miranda and L. Rodrigues, "*Preventing Selfishness in Open Mobile Ad Hoc Networks*," Proc. Seventh CaberNet Radicals Workshop, Oct. 2002.
[24]. Meera Gandhi, S.K.Srivatsa," *Detecting and preventing attacks using network intrusion detection systems", in the* International Journal of Computer Science and Security, Volume: 2, Issue: 1, Pages: 49-58.
[25]. N.Bhalaji, A.Shanmugam, Druhin mukherjee, Nabamalika banerjee." *Direct trust estimated on demand protocol for secured routing in mobile Adhoc networks*", in the International Journal of Computer Science and Security, Volume: 2, Issue: 5, Pages**:** 6-12.

**T.V.P. Sundararajan** received the BE Degree in Electronics and Communication from Kongu Engineering College , Perundurai in 1993 and the ME Degree in Applied Electronics from the Government college of technology, coimbatore in 1999. He is Assistant Professor ,working in Bannari Amman Institute of Technology, Sathyamangalam. He is doing a part time research in Anna University, Chennai . His current research focuses on mobile ad hoc networks and wireless security. He is member of the IEEE, ISTE and the IEEE computer society.
E-mail : tvpszen@yahoo.co.in

**Dr.A.Shanmugam** received the BE Degree in PSG College of Technology in 1972, Coimbatore and ME Degree from College of Engineering, Guindy, Chennai in 1978 and Doctor of Philosophy in Electrical Engineering from Bharathiyar University, Coimbatore in 1994.From 1972–76, he worked as Testing Engineer in Testing and Development Centre, Chennai. He was working as a Lecturer Annamalai University in 1978. He was the Professor and Head of Electronics and Communication Engineering Department at PSG College of Technology, Coimbatore during 1999 to2004. Authored a book titled "Computer Communication Networks" which is published by ISTE, New Delhi, 2000.He is currently the Principal, Bannari Amman Institute of Technology, Sathyamangalam..He is on the editorial board of International Journal Artificial Intelligence in Engineering & Technology (ICAIET), University of Malaysia, International Journal on "Systemics, Cybernetics and Informatics (IJSCI)" Pentagram Research Centre, Hyderabad, India. He is member of the IEEE, the IEEE computer society. E-mail : dras @ yahoo.co.in

# Embedding Software Engineering Disciplines in Entry-Level Programming

**Lung-Lung Liu** llliu@mcu.edu.tw
*Associate Professor, International College*
*Ming Chuan University*
*Gui-Shan, Taoyuan County, Taiwan, ROC 333*

**ABSTRACT**

Software engineering disciplines can be embedded in entry-level programming assignments as a very basic requirement for teachers to the students in the classrooms and mentors to their apprentices in the offices. We are to use three examples to demonstrate how easily some of these software engineering disciplines can be embedded, and we will then prove that they are helpful for quality and productive software development from the point of being with "no source code modification" when some requirements are changed. In fact, convergence can be confirmed even there have been these changes. If the entry-level programming works are with software engineering disciplines, then the total software development effort should be decreased. With this concept in mind for project managers, actually, there are simple refactoring skills that can be further applied to those programs already coded.

**Keywords:** Software Engineering Practice, Preventive Maintenance, Requirement Change.

## 1. INTRODUCTION

We had the experience to give software engineering courses to computer science (junior/senior) students in the campus and software company (programmer) employees in the industry. They are with good programming language skills, such as the handling of syntax details by using Java or C#; but to most of them, software engineering is just like extra requirements to their previous works. They have to change even from their minds. The reasons are clear. The conventional education strategy in computer science is to divide the software courses into programming groups and software engineering groups, and the programming ones are the mandatory. Having studied the programming related courses, the students may want to select the software engineering related ones. That is, when people are to take a software engineering course, their personal programming styles have been set already. However, the styles may be improper for software development if necessary attentions were not paid.

Why don't we embed some of the software engineering disciplines in those programming related courses, especially the entry-level ones? In the following sections we are to demonstrate this by providing three examples: the popular "hello, world" program [1] introduced as the first trial in various programming languages, the state transition [2] handling program, and the sorting [3] program. We will show that some software engineering disciplines can be easily embedded in these entry-level programming exercises. These disciplines are for "no source code modification" to possible requirements change, which are usually to happen. In other words, a considerable programming style (with software engineering disciplines) can help reduce the chance to modify

the source code. It is true that if code need not be modified when there is a requirement change, then there is the higher possibility for quality and productive software.

The theoretical background of the disciplines is preventive maintenance [4], or a convergence software process, which makes sure that consecutive processes can really approach to the target. We will discuss this after the demonstration of the examples. The other software engineering technology can be directly applied here is refactoring [5]. It is for programs already coded by students or junior employees. To the teachers and managers, asking them to do the refactoring works is a challenge. Nevertheless, automated skills but not labor-intensive routines should be considered. When the students and programmers are used to embedding software engineering disciplines in their daily coding works, Personal Software Process [6] is then significant.

## 2. THE "HELLO, WORLD" EXAMPLE

The program titled "hello, world" has been the first trial in learning different kinds of programming languages for years. Although there are versions of the program, the major function is to output the words: hello, world. In the following, we use a common pseudo code to specify the program. A typical and simple version may look like this:

```
function hello_world()
begin
        print("hello, world')
end
```

From a programmer's point of view, it is well done. However, even the program is titled as hello_world, a customer or stakeholder may request a change for alternative output of the words: hi, globe. They are the same, technically in programming skills, but they are actually different programs since the code should be modified. There are risks to introduce human errors in the modification processes.

Literals should be avoided in programs, and variables and parameters are suggested in general. We skip the use of variables and go straight for the use of parameters. The new version of the program may now look like this:

```
function hello_world()
begin
        print(get_whatever_requested())
end
```

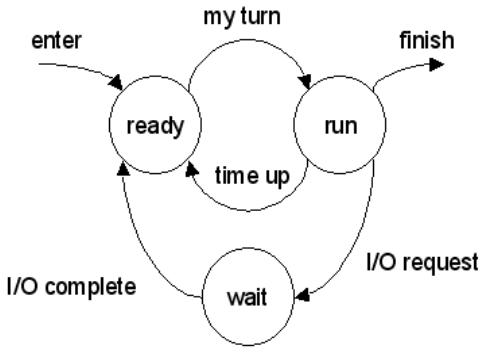
The nested function get_whatever_requested() is flexible and powerful. It can be another system supported function just like the print(something) function. The programmer is then waiting there, for all possible words requested as the desired output, since there is no need to modify any of the code. (Actually, no re-compilation is necessary.)

There are ways to let the system know where to get those whatever requested by providing a customizable profile, and the contents may indicate the input source such as the keyboard, a file, a database entry, or those through a network. The format of the input data can be further specified, such as whether the requested words are in English or Chinese.

In the past, we seldom saw students (or junior programmers) go this way at their earlier programming learning stages. Their styles were pure "hello, word" by referencing the books of introduction to programming languages. Although the programming (development) environments have been greatly improved since Java and C# were introduced years ago, the design of them together with software engineering basics is still blocked.

## 3. THE STATE TRANSITION HANDLING EXAMPLE

The handling of state transition is another typical exercise for computer science students and business application oriented junior programmers. The processes in a multitasking operating system is basically synchronized by a scheduler, and the processes are well controlled to be with ready, running, or waiting states. The current status of a running business can also be managed by a control system, and each of the possible sales cases (can be specified by using sales forms) can be associated with a current state, such as initial, in review, waiting for approval, or final. The following is a state transition diagram of processes in an operating system, and it is actually the requirement for the implementation of the handling program:




According to the diagram, a direct program with a lot of conditional statements is obtainable. However, we know that conditional branches and expressions in conditional statements are with high risk to introduce errors in a program. Furthermore, when the customer or the stakeholder once raised a change, the effort of modifying the code will be with even higher risks of new errors.

An advanced approach is to use the state transition table, which is equivalent to the state transition diagram but is much more precise for programming:

| event / state | my turn | time up | I/O request | I/O complete |
|---|---|---|---|---|
| ready | run | | | |
| run | | run | wait | |
| wait | | | | ready |

- A newly entered process is in the ready state
- A finished process will leave

The controlling of state change is now with no conditional statement coded, since the change can be determined by checking with row index (the current state) and column index (the input). The cell with proper indices in the table tells the next state. In addition, the size of the table doesn't need to be fixed. Or, the number of rows and that of columns can be assigned outside of the code, hence again, like we have mentioned in the previous example, they and the whole contents of the table can be specified in a customizable profile.

We experienced so many cases that students' (and employees') programs are designed with long and deep conditional statements, but it seemed that some of them did enjoy this style. Actually, a

professional state transition handling mechanism (software) is with only limited size (lines of code), and it is almost always reusable. The programming books should guide the readers how to design the contents in a state transition table but not how to code according to a diagram.

## 4. THE SORTING EXAMPLE

To solve the sorting problem is a typical exercise to beginning programmers, especially to computer science students whose teacher wanted them to get into algorithms as early as possible. Usually, after students have tried the bubble sort and quick sort skills, the job is done. However, in practical programming concerns, there are much more. The following is the list of weekdays in their conventional sequence:

Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday

What is the result if we let a common sorting program to run with this list as input? If no further information is provided, the output is this:

Friday
Monday
Saturday
Sunday
Thursday
Tuesday
Wednesday

It is somewhat funny since common users will get confused with the result, but the computer science students will support the result with explanations in technical terms. A tentative conclusion may be that the computer is not as clever as human beings, because it does not understand what weekdays really are. However, the actual problem is the sorting program but not the computer. The truth is that the program is lack of the thoughtful considerations of user friendly.

A proper sorting program should be able to handle different data types, formats, languages, and even semantics. Data definitions, data dictionaries, and even guidelines can be associated with the program as changeable supports, but the code can be kept unchanged at all. Although we have not seen a universal sorting program developed by the students and employees until now, it is encouraged for them to think this way. It is the embedding of software engineering disciplines in entry-level programming.

## 5. CONVERGENCE

The three demonstrated examples should have indicated some of the software engineering disciplines such as: no literal in the code, simplifying the code whenever it is possible, and real information hiding. There may be debates, although they are just examples for better understanding. In the following, we are to prove that they are helpful for quality and productive software development from the point of being with “no source code modification” when some requirements are changed, and it is easy to do so.

**Fact 1.** Some requirement change can be handled by changing the contents in a provided user-customizable profile.

A typical example of this is the use of the win.ini file in the popular MS Windows systems. When a user is to change his/her desktop's background color, a setting in the profile satisfied the request. Either a direct text editing or a mouse clicking through graphical user interface completes the setting.

**Fact 2.** A program is composed of data sections and instruction sections. No source code modification of a program means that there is no change, neither in the data sections, nor in the instruction sections.

Data sections are declared as variables and literals (such as specific numerals and strings), and instruction sections are specified as statements. The instructions refer to literals and current values of variables, and then new data values of variables as computation results are generated.

**Fact 3.** The total number of errors in a program will not increase unless the program (source code) has been modified.

There is the number as the total number of errors in a program, although it is very hard to find the number. (Usually we use testing record to estimate the number.) However, if the program has been modified, then the relationship between the new number of errors and the original one is unclear (although both of the two numbers are unknown). What we are sure is this: If there is no change in the code, then everything is still fixed as it was before.

**Theorem 1.** For some of the requirement change, no source code modification can be achieved by replacing literals with contents in a user profile.
[Proof] Let R be the requirement. A source code C is programmed to first read from the user profile F for a value, and then assign the value to variable V in C as the requirement. R is originally in F, and C can get R for V. According to Fact 1, when there is a requirement change from R to R', we first replace R by R' in F, and we then have C perform the same process for V. R' is now the value assigned to V. However, according to Fact 2, there is no source code modification in C.

**Theorem 2.** No source code modification is helpful for quality and productive software development.
[Proof] According to Fact 3, the total number of errors will not increase if there is no source code modification in a program. For quality software development, this shows a convergence (or at least, non-divergence) in the programming versus requirement change processes, since the trend of introducing new errors does not exist. For productive software development, it is clear that the language translation procedures such as the recompilation of the modified program, the make of a new load module, and the adjusting for a new run environment, can be avoided. Hence, it is with higher productivity.

**Theorem 3.** It is doable to embed software engineering disciplines in entry-level programming.
[Proof] There is no special programming skill for students or junior employees to embed software engineering disciplines in their works. According to the three examples demonstrated, it is easily doable.

## 6. REFACTORING

Refactoring is the process to modify a program but with no functional change. It was introduced recently for advanced software development approaches such as extreme programming techniques. However, for entry-level programmers, the concept is applicable. In short, it is not the work for requirement change from the customers and the stakeholders, but it is the work for a

student or junior programmer to enhance his/her current code. The purpose, from a software engineering point of view, is also for quality and productive software development.

Most of the cases for students and junior employees in their programming experience indicated that, after a supposed to be correct output has been obtained, the programs will not be modified. Actually, the teachers or managers should ask or guide the students or junior employees to continue their works for no source code modification to some of the requirement change, as was discussed before. The theorems say that the literals in their programs can be easily replaced by contents in a user customizable profile. If this has been the discipline naturally embedded in the programming works, then they are on the right software engineering way.

Automated skills but not labor-intensive works are suggested for refactoring. For example, a first exercise to entry-level programmers may be the scanning of literals in a program, and then the second step is to properly replace the code. The refactoring process is to modify the code where a literal is used by function calls (reading an external user profile) that can return a value equivalent to the original literal. Actually, the use of literals in a specific programming language can be handled in compilers as an optional feature. Hence, the automated skills for refactoring can be applied.

## 7. CONCLUSIONS

Software engineering disciplines can be easily embedded in entry-level programming exercises as requirements. We have tried to use three examples to demonstrate how software engineering disciplines can be embedded, and we proved that they are helpful for quality and productive software development. No source code modification when some requirements were changed is the theme. The theoretical background of the disciplines is preventive maintenance, or a convergence of a software process, which makes sure that consecutive process steps can really get approach to the target. In fact, convergence can be confirmed even there have been requirement changes, since the code is not changed. If the entry-level programming works are with software engineering disciplines, then the quality of software development should be with well control, and the total software development effort should be decreased. To be more aggressive, there are simple refactoring skills that can be further applied to those programs already coded.

Our results are logically significant, and they are also practical. For example, there have been the studies on software maintenance and software reusability, such as (1) the handling of maintainability, especially the changeability [7], and (2) the metrics of reusable code [8]. Our results may indicate that (1) the handling effort of changeability can be easier, no matter the design is aspect oriented or not, and (2) the CRL LOC (i.e., the Component Reusability Level based on Lines of Code) is actually 100%. The reason is obvious since there is no code modification. (However, the proof is beyond the scope here.) The other example is with the discussion of emphasizing on preventive maintenance in the introducing of a software process to an organization [9]. The experience also indicated that the case of individual programmers with good software engineering disciplines is a key successful factor .

The future work of this study is on the requirement analysis of a practical universal sorting program. No matter what the requirement change is, the code of the sorting program is not to be changed. Although we have collected many requirements from groups of end users, it is still on going. One typical requirement is to provide a sorting program for a list of postal addresses in Chinese.

## 8. REFERENCES

1. Brian W. Kernighan, *"Programming in C: A Tutorial,"* Bell Laboratories, Murray Hills, N.J., USA, 1974

2. Abraham Silberschatz, Peter Baer Galvin, and Greg Gagne, *"Operating Systems Concepts,"* Seventh Edition, John Wiley and Sons, Inc., Ch. 5, 2005

3. Donald. E. Knuth, *"The Art of Computer Programming, Volume 3: Sorting and Searching,"* Second Edition, Addison Wesley, Ch. 5, 1998

4. Roger S. Pressman, *"Software Engineering: A Practitioner's Approach,"* Sixth Edition, McGraw-Hill, Ch. 31, International Edition, 2005

5. Martin Fowler, Kent Beck, John Brant, William Opdyke, and Don Roberts, *"Refactoring: Improving the Design of Existing Code,"* Addison Wesley, Ch. 8, 2000

6. Watts Humphrey, *"PSP, A Self-Improvement for Software Engineers,"* Addison Wesley, Ch. 1, 2005

7. Avadhesh Kumar, Rajesh Kumar, and P S Grover, *"An Evaluation of Maintainability of Aspect-Oriented Systems: a Practical Approach,"* in International Journal of Computer Science and Security, Volume 1: Issue (2), pp. 1~9, July/August 2007

8. Arun Shamar, Rajesh Kumar, and P S Grover, *"Managing Component-Based Systems with Reusable Components,"* in International Journal of Computer Science and Security, Volume 1: Issue (2), pp, 60~65, July/August 2007

9. Lung-Lung Liu, *"Software Maintenance and CMMI for Development: A Practitioner's Point of View,"* in Journal of Software Engineering Studies, Volume 1, No. 2, pp. 68~77, December 2006

# A Binary Replication Strategy for Large-scale Mobile Environments

**Ashraf A Fadelelmoula** ashrafafadel@hotmail.com
*Department of Computer and Information Sciences*
*Universiti Teknologi PETRONAS*
*31750 Tronoh, Perak, Malaysia*

**P.D.D.Dominic** dhanapal_d@petronas.com.my
*Department of Computer and Information Sciences*
*Universiti Teknologi PETRONAS*
*31750 Tronoh, Perak, Malaysia*

**Azween Abdullah** Azweenabdullah@petronas.com.my
*Department of Computer and Information Sciences*
*Universiti Teknologi PETRONAS*
*31750 Tronoh, Perak, Malaysia*

**Hamidah Ibrahim** hamidah@fsktm.upm.edu.my
*Faculty of Computer Science and Information Technology*
*Universiti Putra Malaysia*

**ABSTRACT**

An important challenge to database researchers in mobile computing environments is to provide a data replication solution that maintains the consistency and improves the availability of replicated data. This paper addresses this problem for large scale mobile environments. Our solution represents a new binary hybrid replication strategy in terms of its components and approach. The new strategy encompasses two components: replication architecture to provide a solid infrastructure for improving data availability and a multi-agent based replication method to propagate recent updates between the components of the replication architecture in a manner that improves availability of last updates and achieves the consistency of data. The new strategy is a hybrid of both pessimistic and optimistic replication approaches in order to exploit the features of each. These features are supporting higher availability of recent updates and lower rate of inconsistencies as well as supporting the mobility of users. To model and analyze the stochastic behavior of the replicated system using our strategy, the research developed Stochastic Petri net (SPN) model. Then the Continuous Time Markov Chain (CTMC) is derived from the developed SPN and the Markov chain theory is used to obtain the steady state probabilities.

**Keywords:** pessimistic replication, optimistic replication, availability, consistency, Stochastic Petri net.

## 1. INTRODUCTION

Rapid advancements in wireless technologies and mobile devices have made mobile computing enjoying considerable attention in the past few years as a fertile area of work for researchers in the areas of database and data management. As mobile computing devices become more and more common, mobile databases are also becoming popular. Mobile database has been defined as database that is portable and physically separate from a centralized database server but is capable of communicating with server from remote sites allowing the sharing of corporate data [1].

Mobility of users and portability of devices pose new problems in the management of data [2, 3], including transaction management, query processing, and data replication. Therefore, mobile computing environments require data management approaches that are able to provide complete and highly available access to shared data at any time from any where. One way to achieve such goal is through data replication techniques. The importance of such techniques is increasing as collaboration through wide-area and mobile networks becomes popular [4]. However, maintaining the consistency of replicated data among all replicas represents a challenge in mobile computing environments when updates are allowed at any replica.

This paper addresses the problem of maintaining consistency and improving availability of replicated data for large scale distributed database systems that operate in mobile environments. This type of systems is characterized by a large number of replicas (i.e. hundreds of replicas) and a large number of updates (i.e. tens of updates per data items are expected at any period of time) are performed in these replicas. Examples of such systems include mobile health care, mobile data warehousing, news gathering, and traffic control management systems.

In such type of mobile environments, the concurrent updates of large number of replicas during the disconnection time influences consistency and availability of the replicated data, by leading to divergence in the database states (i.e. the data in the database at a particular moment in time). As a solution to the aforementioned problems, this paper proposes a new replication strategy that acts in accordance with the characteristics of large scale mobile environments.

This paper is organized as follows. The next section provides the background and related work. Section 3 describes the proposed replication strategy. Section 4 gives the details of the behavior modeling. Section 5 presents the contribution and discussions. Section 6 concludes the paper.

## 2. BACKGROUND AND REALTED WORK

Data replication strategies are divided into optimistic and pessimistic approaches [5, 6, 7]. Pessimistic replication avoids update conflicts by restricting updates to a single replica based on the pessimistic presumption that update conflicts are likely to occur. This ensures data consistency because only one copy of the data can be changed. Pessimistic replication performs well in local-area networks in which latencies are small and failures uncommon. Primary-copy algorithms [8] are an example of pessimistic approaches. However, pessimistic approaches are not suitable for mobile environments, because they are built for environments in which the communication is stable and hosts have well known locations.

An optimistic replication, in contrast, allows multiple replicas to be concurrently updatable based on the optimistic presumption that update conflicts are rare. Conflicting updates are detected and resolved after they occurred. Therefore, this schema allows the users to access any replica at any time, which means higher write availability to the various sites. However, optimistic replication can lead to update conflicts and inconsistencies in the replicated data.

Using optimistic replication in mobile environments has been studied in several research efforts. ROAM [9] is an optimistic replication system that provides a scalable replication solution for the mobile user. ROAM is based on the Ward Model [10]. The authors group replicas into wards

(wide area replication domains). All ward members are peers, allowing any pair of ward members to directly synchronize and communicate.

A multi-master scheme is used in [11], that is, read-any/write-any. The servers allow access (read and write) to the replicated data even when they are disconnected. To reach an eventual consistency in which the servers converge to an identical copy, an adaptation in the primary commit scheme is used.

A hybrid replication strategy is presented in [12] that have different ways of replicating and managing data on fixed and mobile networks. In the fixed network, the data object is replicated to all sites, while in the mobile network, the data object is replicated asynchronously at only one site based on the most frequently visited site.

Cedar [13] uses a simple client-server design in which a central server holds the master copy of the database. At infrequent intervals when a client has excellent connectivity to the server (which may occur hours or days apart), its replica is refreshed from the master copy.

However, aforementioned strategies have not explicitly addressed the issues of consistency and availability of data in large scale distributed information systems that operate in mobile environments. Therefore, this paper comes to a conclusion that additional research toward a new replication strategy is needed to investigate and address above mentioned issues.

### 2.1 SPN Background

Petri Nets (PNs) are an important graphical and mathematical tool used to study the behavior of many systems. They are very well-suited for describing and studying systems that are characterized as being concurrent, asynchronous, distributed, and stochastic [17, 19]. A PN is a directed bipartite graph that consists of two types of nodes called places (represented by circles) and transitions (represented by bars). Directed arcs connect places to transitions and transitions to places. Places may contain tokens (represented by dots).

The state of a PN is defined by the number of tokens contained in each place and is denoted by a vector $M$, whose $i^{th}$ component represents the number of tokens in the $i^{th}$ place. The PN state is usually called the PN marking. The definition of a PN requires the specification of the initial marking $M'$. A place is an input to a transition if an arc exists from the place to the transition. A place is an output from a transition if an arc exists from the transition to the place. A transition is said to be enabled at a marking $M$ when all of its input places contain at least one token. A transition may fire if it is enabled. The firing of a transition t at marking $M$ removes one token from each input place and placing one token in each output place. Each firing of a transition modifies the distribution of tokens on places and thus produces a new marking for the PN.

In a PN with a given initial marking $M'$, the reachability set ($RS$) is defined as the set of all markings that can be "reached" from $M'$ by means of a sequence of transition firings. The $RS$ does not contain information about the transition sequences fired to reach each marking. This information is contained in the reachability graph, where each node represents a reachable state, and there is an arc from $M_1$ to $M_2$ if the marking $M_2$ *is* directly reachable from $M_1$. If the firing of $t$ led to changing $M_1$ to $M_2$, the arc is labeled with $t$. Note that more than one arc can connect two nodes (it is indeed possible for two transitions to be enabled in the same marking and to produce the same state change), so that the reachability graph is actually a multigraph.

SPNs are derived from standard Petri nets by associating with each transition in a PN an exponentially distributed firing time [16, 18]. These nets are isomorphic to continuous-time Markov chains (CTMCs) due to the memoryless property of exponential distributions. This property allows for the analysis of SPNs and the derivation of useful performance measures. The states of the CTMC are the markings of the reachability graph, and the state transition rates are

the exponential firing rates of the transitions in the SPN. The steady-state solution of the equivalent finite CTMC can be obtained by solving a set of algebraic equations.

## 3. REPLICATION STRATEGY

The proposed replication strategy encompasses two components: replication architecture and replication method. The purpose of the replication architecture is to provide a comprehensive infrastructure for improving data availability and supporting large number of replicas in mobile environments by determining the required components that are involved in the replication process. The purpose of the replication method is to transfer data updates between the components of the replication architecture in a manner that achieves the consistency of data and improves availability of recent updates to interested hosts.

The new strategy is a hybrid of both pessimistic and optimistic replication approaches. The pessimistic approach is used for restricting updates of infrequently changed data to a single replica. The reason behind this restriction is that if the modifications of these data are allowed on several sites, it will influence data consistency by having multiple values for the same data item (such as multiple codes for the same disease or multiple codes for the same drug). On the other hand, the optimistic replication is used for allowing updates of frequently changed data to be performed in multiple replicas. The classification into frequently and infrequently changed data is specified according to the semantic and usage of the data items during the design phase of the database.

### 3.1. System Model

This research considers a large-scale environment consists of fixed hosts, mobile hosts, a replica manager on each host, and a replicated database on each host. A replicated database is called mobile database when it is stored in a mobile host. A part of fixed hosts represent servers with more storage and processing capabilities than the rest. The replicated database contains a set of objects stored on the set of hosts. The database is fully replicated on the servers, while it is partially replicated on both fixed and mobile hosts.

**Definition 3.1.1** An object $O$ is the smallest unit of replication and it represents a tuple $O = <D, R, S>$, where $D = \{d_1, d_2, \ldots, d_n\}$ is a set of data items of the object $O$, $R = \{r_1, r_2, \ldots, r_m\}$ is a set of replicas of $O$, and S is the state of the object $O$.

**Definition 3.1.2** The state $S$ of an object $O$ is a set consisting of states that identifies current values for each data item $d_i \in D$, i.e., $S = \{s_1, s_2, \ldots, s_n\}$.

**Definition 3.1.3** A replica $R$ is a copy of an object stored in a different host and is defined as a function as follows. For a set of updates $U$ that is performed on a set of objects $\bar{O}$, the function $R : U \times \bar{O} \rightarrow S$ identifies a new separate state $s_i \in S$ for an object $O \in \bar{O}$ as a result of performing update $u \in U$ on an object $O$ in a different host.

**Definition 3.1.4** For a replica $R$ in a host $H$, Interested Data Items is a subset $I$ of the set of all data items, which is required for $H$ to perform its duties, i.e., $I \subseteq \{ \bigcup_{i=1}^{n} O_i \}$ , where $n$ is the number of objects in the system.

**Definition 3.1.5** A replicated data item $d_i \in D$ is consistent if and only if its values are identical and in same order as the values of the similar data item in the replica that is stored in the master server, which exists in the fixed network

**Definition 3.1.6** A replica $R$ is consistent if and only if each interested data item $d_i \in \{D \cap I\}$ is consistent.

**Definition 3.1.7** A replica $R$ in a mobile host is in Available-State for a data item $d_i \in D$ if and only if all updates that are performed on $d_i$ in other replicas (either in fixed hosts or mobile hosts) are merged with the updates that are performed on $d_i$ in $R$.

**Definition 3.1.8** Consistent-Available State (CA State) for a replica $R$ that is stored in a mobile host is the state in which:

1. $R$ is consistent
2. $R$ in Available-State for each interested data item $d_i$ in $R$.

### 3.2 Replication Architecture

The proposed replication architecture considers a total geographic area called the master area, divided into a set $Z = \{z_1, \ldots, z_n\}$ of zones. Each zone consists of a set $C = \{c_1, \ldots, c_m\}$ of smaller areas called cells (see figure 1). Each cell represents an area, where the mobile users can perform their duties at a particular period of time before moving to another cell. In this architecture, the network is divided into fixed network and mobile network. The fixed network consists of Fixed Hosts (FH) and wired local area network to connect the fixed hosts in the master area, and also include wide area network to connect fixed hosts in the master and zone areas, and the servers of the cell area. The cell server is augmented with a wireless interface and acts as a mobile support station for connecting mobile hosts to the fixed network. On the other hand, the mobile network consists of wireless network and Mobile Hosts (MH) in the cell area.

To provide more flexibility and application areas for this architecture, replicas are divided into three levels:

**Master Level:** This level contains the master replica, which must be synchronized with the replicas from the zone level. The server in this level is responsible for synchronizing all changes that have been performed on infrequently changed data with the lower level.

**Zone Level:** In this level, each replica must be synchronized with replicas from the lower level. The zone server is responsible for synchronizing all intra-level data with the master server.

**Cell Level:** Each replica in this level is updated frequently, and then synchronized with the cell server's replica and in turn the cell server synchronizes all intra-level data with the zone server.

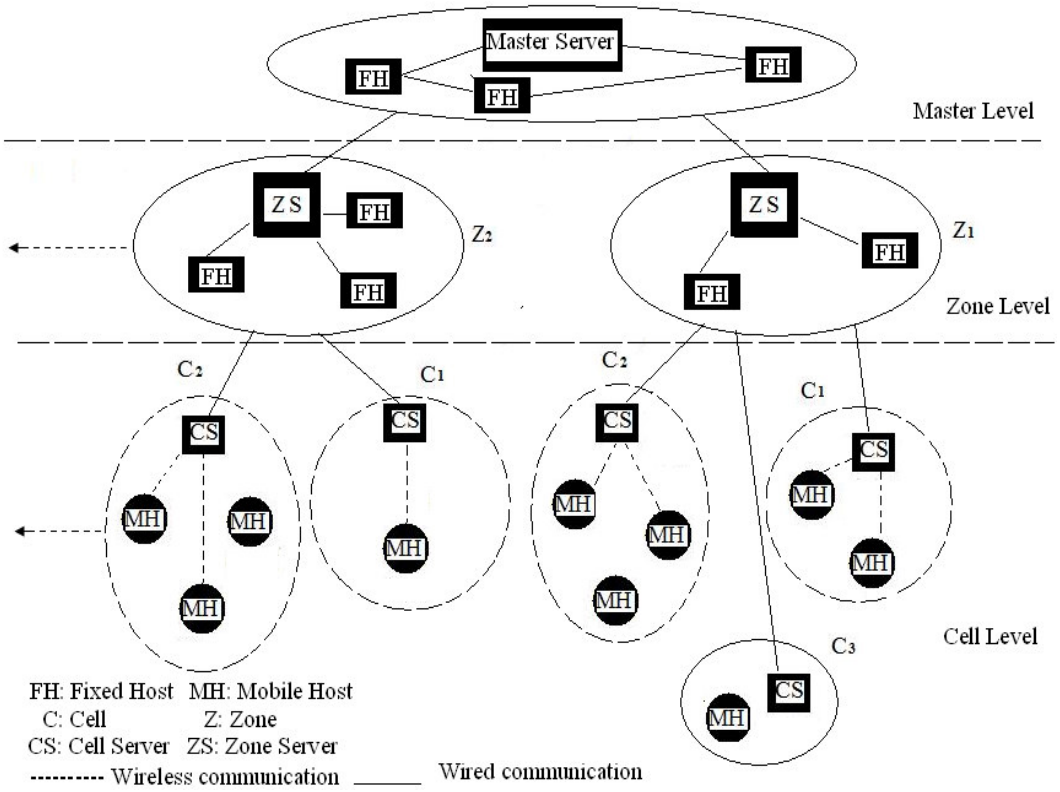



**FIGURE 1:** The Replication Architecture for Mobile Environments

### 3.3. Replication Method

The replication method is based on a multi-agent system called IIRA-dependant multi-agent system. This system is proposed based on a new type of software agent called Instance-Immigration-Removed Agent (IIRA) that is introduced in this research. The research chose this name, according to IIRA working nature, since it creates an instance of itself and this instance migrates to another host and performs its task before removing itself. The purpose of the instance is to propagate recent updates that occurred or are collected in one level to other level in the proposed replication architecture. The following definition will formally define IIRA.

**Definition 3.3.1** IIRA is a 5-tuple $<T, S, D, I, U>$, where:

$T= \{t_1, t_2, t_3, t_4\}$ is a finite set of IIRA types. A type $t_i$ maps each IIRA to a certain level (i.e. the type determines the location of IIRA).

$S = \{s_1, s_2, s_3, s_4, s_5, s_6\}$ is a finite set of IIRA states. Each state represents the current activity that is carried out by IIRA.

$D = \{d_1, \ldots, d_n\}$ is a finite set of data items that are required to store recent updates that are performed on the similar data items, which are stored in the database.

$I = \{i_1, \ldots, i_n\}$ is a finite set of primitives/instructions that are required to perform IIRA activities and the transitions.

$U: T \rightarrow \{1, 2, 3, \ldots, k\}$ is a function for assigning a unique identifier for IIRA in the system.

According to the abovementioned formal definition, the IIRA consists of code and database and it has type, state, and unique identifier.

**IIRA Types:** The research divides IIRA into four types according to the level in which the IIRA carries out its activities. The four types share common structure and behavior, but they inhabit different levels. These types are:

**MH-Resident IIRA (MHR-IIRA):** Every MH has IIRA, and it is responsible for propagating recent updates that are performed in the MH to other hosts in the mobile network or to the cell server that covers the cell where the MH is located at the time of connection.

**Cell Server-Resident IIRA (CSR-IIRA):** This type acts as a broker for propagating recent updates between the fixed network and the mobile network. It propagates all updates that are received from MHR-IIRA to the zone server and vice versa.

**Zone Server-Resident IIRA (ZSR-IIRA):** This type receives all updates that are performed in the fixed network and the mobile network, which are associated with specific zone level, and resolves update conflicts on this level. Then it propagates these updates directly to the master server.

**Master Server-Resident IIRA (MSR-IIRA):** This type receives all updates that are performed in the zone level and resolves update conflicts. Then it propagates these updates directly to each zone server, which in turn propagates these updates to underlying levels.

**IIRA States:** The possible states of the IIRA in the proposed replication strategy are:

**Monitoring:** In this state, the IIRA monitors the connection with the other devices through interacting with its environment (i.e. hosted device) via message passing.

**Retrieving:** The IIRA retrieves the set of recent updates from the hosted device. The IIRA enters this state when the monitoring state results in a connection that is realized with the other host.

**Creating Instance:** The IIRA creates an instance of it and stores the set of recent updates on this instance.

**Migration:** The IIRA instance migrates from the hosted device to other device that the connection is realized with it.

**Insertion:** In this state, the IIRA instance inserts its stored recent updates in the database of the IIRA in the other device.

**Removing:** The migrated instance of IIRA removes itself after completion of the insertion process.

### 3.3.1 IIRA-Dependant Multi-Agent System

The IIRA-dependent multi-agent system (see figure 2) is composed of the four types of IIRA. Each type interacts with others through a synchronization process, in which two types exchange their recent updates via their created instances. In this system, each type can exchange updates directly with the same type or a different type in the server of the level where it inhabits or in a server from underlying level (e.g. the MHR-IIRA can exchange updates directly with CSR-IIRA or other MHR-IIRA).

The exchanging of updates between two types occurs only during the connection period between their owner hosts. The time period that MH waits for the connection with a cell server in the fixed network is not deterministic and it depends on its own conditions, such as connection availability, battery, etc. On the other hand, the research assumes that the time period that a server in the fixed network must wait to connect with the server in the higher level is deterministic and its value depends on the availability and consistency requirements of the replicated system. To decide this value, the research views the connection timing system for connecting servers in the fixed network should mimics the shifting behavior, where the shift time is exploited in collecting recent updates from underlying level (e.g. the cell server should connect to the zone server every one hour to propagate the collected updates from underlying mobile hosts during the last past hour).

When the connection takes place between any two IIRA types, an instance of IIRA type that inhabits the lower level will propagate a set of recent updates called Updates-Result (UR), which is produced by an application, in addition to updates that may have been collected from the underlying level to the database of IIRA type that inhabits the higher level. Then, an instance of the type that inhabits the higher level propagates a set of updates called Recent-Resolved-Updates (ReRU) that is received from the higher level to the replicated database in the lower level. According to this fact, the research defines three types of propagation, as follows:
**Bottom-Up Propagation:** In this type, each MHR-IIRA propagates the set of recent updates (MH-UR) that occurred in its host to the database of the type that inhabits a cell server. Also, each server's IIRA collects the received Updates-Result from underlying level in addition to its server's updates-result in a set called Set of Updates Result (SUR) and resolves updates conflict in this set and then propagates this set to the database of IIRA in the higher level.

As previously mentioned, the time period that the server in the current level should wait for receiving recent updates (i.e. updates collection) from underlying level is deterministic. The value of this period in a server in the higher level is greater than its value in a server in the lower level (e.g. waiting period for the zone server > waiting period for the cell server). This is because the number of hosts, where updates occurred increases as we move to the higher levels. After elapsing of this deterministic period, the IIRA carries out the required operations for this type of propagation.

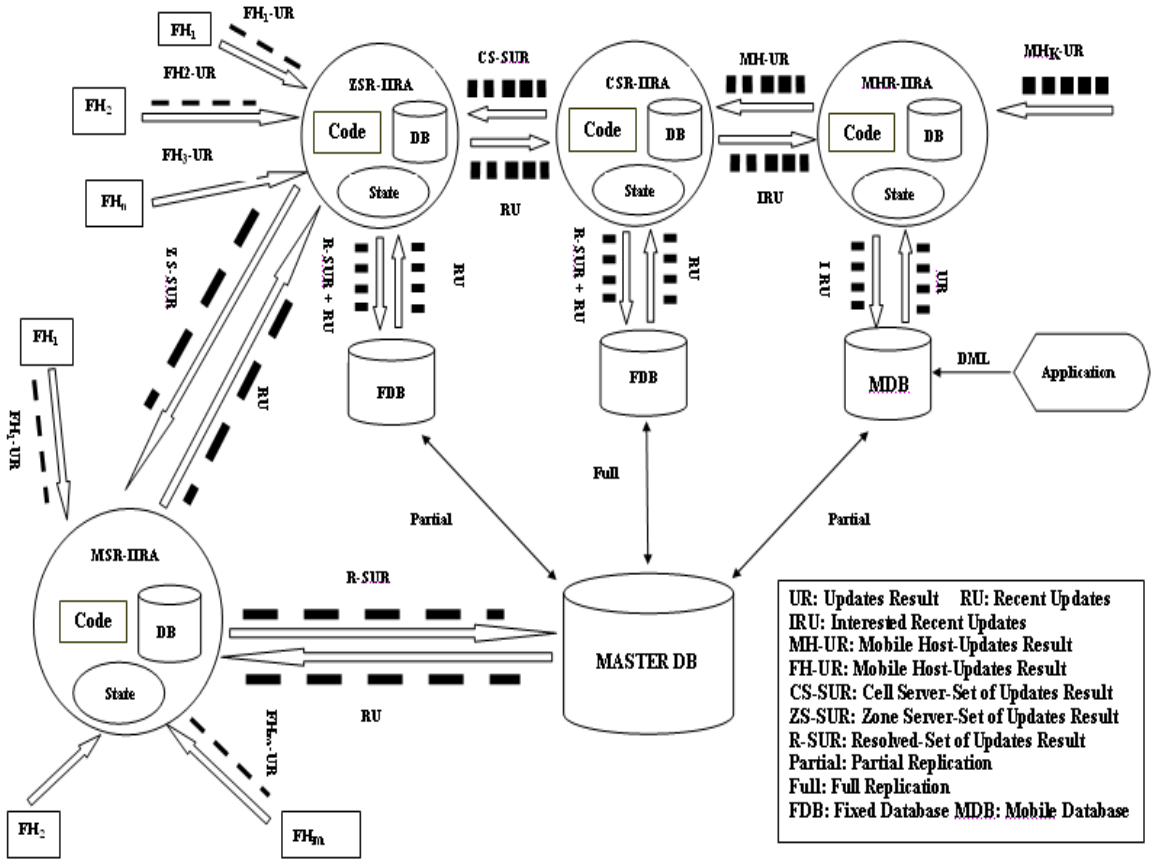



**FIGURE 2:** IIRA-Dependant Multi-Agent System

The typical operations that are involved in this propagation, which are performed by IIRA are:

- Resolving updates conflict through ordering the collected updates on the database of the IIRA type that inhabits the host in the current level and assigning the value of the update timestamp data item (in case of the update is generated on the same host).
- Creating the instance and storing the ordered updates on it.
- Migration of the instance to the host that exists in the higher level and assigning the value of the send timestamp data item to each update before the migration start and after migration request is accepted.
- Execution of the instance in the destination host through insertion of recent updates in its IIRA type's database.

**Top-Down Propagation:** In this type, each server's IIRA propagates the set of recent resolved updates (ReRU) that is received from the higher level to the lower level. For example, each ZSR-IIRA propagates the ReRU that is received from the master server to the underlying cell servers and in turn each cell server propagates a subset of this set called Interested-Recent Resolved-Updates (IReRU) to underlying mobile hosts.

The typical operations that are performed by IIRA for carrying out this propagation are:

- Creating the instance and storing the set of recent resolved updates on it.
- Migration of the instance to the host that exists in the lower level.
- Execution of the instance in the destination host.

**Peer-to-Peer Propagation:** In this type, two IIRA of the same type may exchange their updates. The typical operations that are performed by IIRA in this type are same as in Bottom-Up propagation with a distinction that both hosts, which are involved in the synchronization process, are of the same level.

## 4. BEHAVIOR MODELING

The paper models the dynamic behavior of the proposed multi-agent system with respect to the synchronization process between its components by using SPNs. The purposes are to trace how the mobile database will reach the CA state and to calculate the steady-state probabilities.

The reason of using SPNs is that the research views the synchronization process between the different levels of the replication architecture as a discrete-event stochastic system that encompasses states and events. Each state captures either a snapshot of a set of recent updates or a snapshot of a set of tuples currently is stored in the database. Each event represents either execution of IIRA instance in another level or retrieving of a subset of tuples. The system is stochastic due to its stochastic state transitions since it is evolving over continuous time and making state transitions when events associated with states occur.

The behavior modeling approach using SPN follows a systematic approach described in [14, 15], which incorporates the following steps.

- Modeling of the behavior of the IIRA-dependant multi-agent system using a stochastic Petri net.
- Transforming the developed SPN into its equivalent Markov chain for calculation of the steady state probabilities of marking occurrences. This step requires generating the reachability graph. The Markov chain is obtained by assigning each arc with the rate of the corresponding transition.
- Analyze the Markov chain to obtain steady state probabities.

The research interests in a state in which the mobile database in the mobile host receives a set of recent updates that occurred on the other hosts in both fixed and mobile networks.

### 4.1 The SynchSPN

The following definition will formally define the developed SPN that is used to model the behavior.
**Definition 4.1.** The SynchSPN is a six-tuple $<P, T, A, W, m_0, \Lambda>$ where:
1. $P = \{p_1, p_2, \ldots, p_{11}\}$ is a finite set of places, each place represents either a synchronization state for IIRA database (IIRA-Synch-State**)** or a synchronization state for the replicated database (DB-Synch-State**)** in each level. The former contains the set of recent updates, while the latter contains the set of tuples currently stored in the database.

2. $T= \{t_1, t_2, \ldots, t_{14}\}$ is a finite set of transitions, each transition represents an operation carried out by the IIRA in different levels. These operations are:
   i. Retrieving the set of recent updates.
   ii. Execution of the IIRA instance in the other level to transfer the recent updates.

3. $A \subseteq (P \times T) \cup (T \times P)$ is a finite set of arcs that represents the number of updates, which have to be transferred after the execution process or the number of updates that have to be retrieved after the retrieving process.

4. $W: A \rightarrow \{1, 2, \ldots\}$ is the weight function attached to the arcs. This function maps each arc to the number of updates that are to be propagated in each synchronization process between the different levels.

5. $m_0: P \rightarrow \{0, 0, 0, 0, |MH\text{-}DBS|i, |CS\text{-}DBS|j, |ZS\text{-}DBS|k, |MS\text{-}DBS|, 0, 0, 0\}$ is the initial marking, which represents the number of recent updates at the synchronization state for each IIRA ($p_1$, $p_2$,

$p_3$, $p_4$, $p_9$, $p_{10}$, $p_{11}$) and the number of tuples that are currently stored in each database ($p_5$, $p_6$, $p_7$ , $p_8$). (In the context of discrete-event systems, the marking of the PN corresponds to the state of the system).

6. $\Lambda = \{\lambda_1, \lambda_2, ..., \lambda_{14}\}$ is the set of firing rates associated with the SPN transitions.

SynchSPN is developed based on the following theorem.
**Assertion 4.1.** The synchronization process has a Markov property.

**Proof.** Let $U(t)$ denotes the number of recent updates that should be propagated by IIRA instance from a host to another during their synchronization at time instant $t$, where $t$ varies over a parameter set $T$. The value of $U(t)$ is not deterministic because it depends on the number of generated updates, which means $U(t)$ is a random variable. Accordingly, the synchronization process can be defined as a family of random variables $\{U(t)|t \in T\}$, where the values of $U(t)$ represent the states of the synchronization process. Thus, the synchronization process represents a stochastic process. By introducing a flag data item to mark the updates that are propagated at the current synchronization instant $n$, we find that the number of the recent updates that should be propagated on the next instant $n+1$ equals to the number of unmarked updates at n, which represent the recent updates that occur after $n$. Therefore, the value of $U(n+1)$ depends only the value of $U(n)$ and not on any past states.

By using SPN to model the synchronization system (see figure 3 and tables 1-3), the system is composed of eleven places and fourteen transitions.

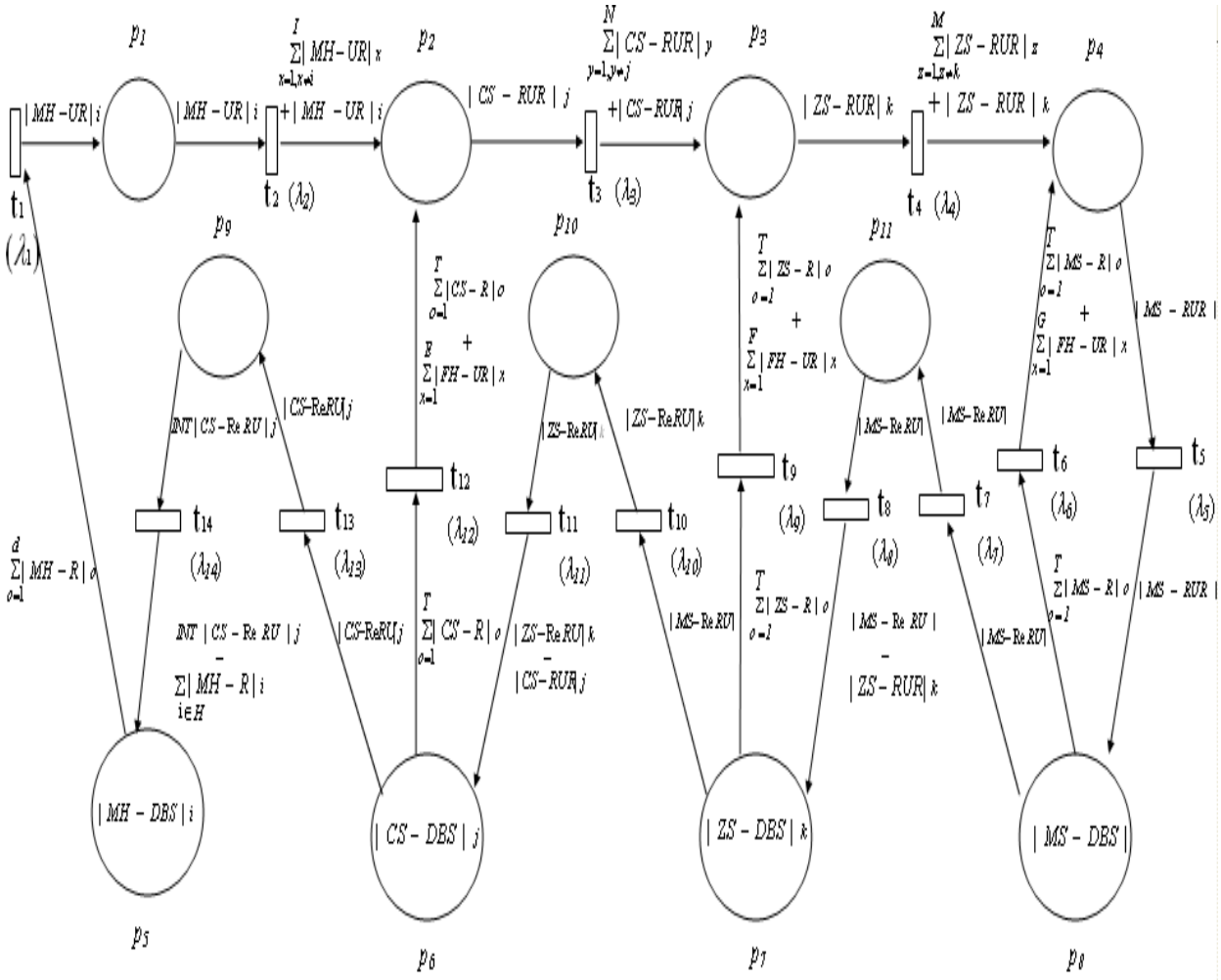



**FIGURE 3:** The SynchSPN

| Place | Description |
|---|---|
| $p_1$ | IIRA-Synch-State in $MH_i$ for execution in $CS_j$ |
| $p_2$ | IIRA-Synch-State in $CS_j$ for execution in $ZS_k$ |
| $p_3$ | IIRA-Synch-State in $ZS_k$ for execution in $MS$ |
| $p_4$ | IIRA-Synch-State in $MS$ for execution in $MS$ |
| $p_5$ | DB-Synch-State in $MH_i$ |
| $p_6$ | DB-Synch-State in $CS_j$ |
| $p_7$ | DB-Synch-State in $ZS_k$ |
| $p_8$ | DB-Synch-State in $MS$ |
| $p_9$ | IIRA-Synch-State in $CS_j$ for execution in $MH_i$ |
| $p_{10}$ | IIRA-Synch-State in $ZS_k$ for execution in $CS_j$ |
| $p_{11}$ | IIRA-Synch-State in $MS$ for execution in $ZS_k$ |

**TABLE 1:** Description of Places

**Places**. They are called synchronization states. The research looks abstractly at the synchronization state as a state/place that contains a set of updates. There are three types of places:

- **IIRA-Synch-State for execution in the higher level:** It contains the set of recent updates that occurred in its host in addition to the set of collected updates from underlying level. For example, the synchronization state $p_2$ represents the set of all recent updates that issued on cell server *j* in addition to the set of all updates that are transferred from mobile hosts that are synchronized with this server in the last synchronization period. This type includes $p_1$, $p_2$, $p_3$, and $p_4$. Note that the set of all recent updates that occurred in the servers that exist in the fixed network are assumed that they include also the recent updates that are received from the fixed hosts in the level of those severs.
- **IIRA-Synch-State for execution in the lower level:** It contains the set of all recent resolved updates that are received from the higher level. For example, the synchronization state $p_{10}$ for zone server k represents the set of all recent resolved updates that are received from the master server. This type includes $p_9$, $p_{10}$, and $p_{11}$.
- **DB-Synch-State:** This type stores the set of all tuples that are currently stored in the replicated database. It includes $p_5$, $p_6$, $p_7$, and $p_8$.

| Transition | Description |
|---|---|
| $t_1$ | Retrieving recent updates from the replicated database in $MH_i$ |
| $t_2$ | Execution of MHR-IIRA instance on $CS_j$ |
| $t_3$ | Execution of CSR-IIRA instance on $ZS_k$ |
| $t_4$ | Execution of ZSR-IIRA instance on $MS$ |
| $t_5$ | Execution of MSR-IIRA instance on $MS$ |
| $t_6$ | Retrieving recent updates from the replicated database in $MS$ |
| $t_7$ | Retrieving recent resolved updates from the replicated database in $MS$ |
| $t_8$ | Execution of MSR-IIRA instance on $ZS_k$ |
| $t_9$ | Retrieving recent updates from the replicated database in $ZS_k$ |
| $t_{10}$ | Retrieving recent resolved updates from the replicated database in $ZS_k$ |
| $t_{11}$ | Execution of ZSR-IIRA instance on $CS_j$ |
| $t_{12}$ | Retrieving recent updates from the replicated database in $CS_j$ |
| $t_{13}$ | Retrieving recent resolved updates from the replicated database in $CS_j$ |
| $t_{14}$ | Execution of CSR-IIRA instance on $MH_i$ |

**TABLE 2:** Description of Transitions

**Transitions.** Also, the research looks abstractly at the transition as an event that leads to either retrieving or propagating the set of recent updates. There are three types of transitions: execution of the IIRA instance on the higher level, execution of the IIRA instance on the lower level, and retrieving of recent updates.

**Execution of the IIRA instance on the higher level.** This type inserts the contents of synchronization state for IIRA for execution in the higher level in the database of the IIRA type that inhabits the higher level. Some conditions must be satisfied in order to fire this type of transitions. These conditions are:

1. Enabling condition. It is composed of two conditions as follows.
    i. There is at least one recent update in the input place of the execution transition.
    ii. The connection with the other host should happen.

The waiting time for the occurrence of the connection is not considered in the period that is required for firing transitions. This is because as previously mentioned, the waiting process for the connection is not considered as a required IIRA's operation for updates propagation. Moreover, the waiting time has a random value for MHs and a deterministic value for the

servers in the fixed network. Therefore, the research interests on the occurrence of the connection as a required condition for firing.

2. Completion of the updates conflicts resolution through ordering process for the collected updates.
3. Migration of the IIRA instance to the other host for execution of its parent synchronization state in that host.
4. Getting the permission for execution in the other host.

| Symbol | Meaning |
|---|---|
| *MH*, *FH*, *CS*, *ZS, and MS* | Mobile Host, Fixed Host, Cell Server, Zone Server, and Master Server, respectively |
| *T* | Total number of database objects |
| *d* | Total number of database objects that are replicated in $MH_i$ (d<T) |
| $\mid X \mid$ | The number of updates in the set *X* |
| *MH-R*, *CS-R*, *ZS-R*, and *MS-R* | Set of recent updates for object *O* in $MH_i$, $CS_j$, $ZS_k$, and *MS*, respectively |
| *MH-UR*, *FH-UR* | Set of recent updates for all replicated objects in $MH_i$ and $FH_l$, respectively |
| *CS-RUR*, *ZS-RUR*, and *MS-RUR* | Set of resolved recent updates at $CS_j$, $ZS_k$, and *MS*, respectively |
| *MS-ReRU*, *ZS-ReRU*, and *CS-ReRU* | Set of recent resolved updates that are propagated to underlying level from *MS*, $ZS_k$, and $CS_j$, respectively |
| *MS-DBS*, *ZS-DBS*, *CS-DBS*, and *MH-DBS* | Replicated database state in *MS*, $ZS_k$, $CS_j$, and $MH_i$, respectively |
| *I* | Total number of mobile hosts that have synchronized with $CS_j$ before the synchronization of $MH_i$ during the $CS_j$ updates collection period |
| *N* | Total number of cell servers that have synchronized with $ZS_k$ before synchronization of $CS_j$ during the $ZS_k$ updates collection period |
| *M* | Total number of zone servers that have synchronized with *MS* before synchronization of $ZS_k$ during the *MS* updates collection period |
| *E, F, G* | Total number of fixed hosts that have synchronized with $CS_j$, $ZS_k$, and *MS*, respectively, during their updates collection period |
| *H* | Set of recent updates that are propagated from $MH_i$ to $CS_j$ in the last synchronization period |

**TABLE 3:** Notations and Their Meanings

Each transition from this type can fire in a time instance equals to $T^+$ that is reached after elapsing of a time period of length $T^{up}$. The value of $T^{up}$ consists of the time period that is required for ordering the collected updates ($Ot^+$), the time period that is required for IIRA instance to migrate to the higher level host ($Mt^+$), and the time period that IIRA instance takes for waiting to get the permission for execution in the higher level host ($Wt^+$). Since, each one of these values is not deterministic; this means that $T^{up}$ is a random variable. Here, we omit the time that is required for creating the instance, because it is performed locally.

**Execution of the IIRA instance on the lower level.** This type inserts the contents of synchronization state for IIRA for execution on the lower level in the replicated database of the host that inhabits the lower level. For firing this type, the same conditions that should be satisfied for the first type are applied here, excluding the completion of the updates ordering process.

Each transition from this type can fire in a time instance equals to $T^{-}$ that is reached after elapsing of a time period of length $T^{down}$. The value of $T^{down}$ consists of the time period that is required for IIRA instance to migrate to the lower level host ($Mt^{-}$) and the value of the period that IIRA instance takes for waiting to get the permission for execution in the lower level host ($Wt^{-}$). Also, the time that is required for creating the instance is omitted, since it is performed locally in the same host.

**Retrieving recent updates.** This type involves either retrieving recent updates for synchronization with the higher level (propagating it to the higher level) or retrieving recent resolved updates for synchronization with the lower level. For firing this type, the enabling condition of the first type should be satisfied.
Each transition from this type can fire in a time instance equals to $T^{r}$ that is reached after elapsing of a time period of length $T^{ret}$. The value of $T^{ret}$ represents the time period that is required for IIRA to retrieve either recent updates or recent resolved updates from the replica. This value depends on the number of updates that should be retrieved in each synchronization process. Therefore, $T^{ret}$ is a random variable. The periods $T^{up}$, $T^{down}$, and $T^{ret}$ represent random variables because their values are obtained depending on non deterministic values.

Note that instead of adding a transition for representing the migration and its associated input place for representing the migrated state, we incorporate them into the execution transition and in the synchronization state, because the migrated state represents the synchronization state itself and the execution of the migration transition encompasses that the synchronization state is already migrated to the other host. Thus, incorporation is performed to prevent the complexity of SynchSPN.

**Firing rates.** Each transition in SynchSPN is associated with a firing rate (i.e. the parameter $\lambda$). This is because the periods $T^{up}$, $T^{down}$, and $T^{ret}$ that represent the firing delays after correspondence transitions are enabled are random variables and are assumed exponentially distributed.

**The initial marking.** In this marking (i.e. $m_0$), the replicated databases that are stored in the servers in the fixed network (i.e. $CS_j$, $ZS_k$, and $MS$) are assumed identical. Also, the mobile database that is stored in the $MH_i$ is assumed that has received a set of last updates. This received set may represent either all or a subset of the last updates, which occurred or propagated to the fixed network in the period that precedes the time of the last synchronization of $MH_i$ with the fixed network (i.e. during the disconnection time before the time of the last synchronization), which equals to $MH_i$-$SynchT_{n-1}$ - $MH_i$-$SynchT_{n-2}$, where $MH_i$-$SynchT_{n-1}$ is the time of the last synchronization of $MH_i$ with the fixed network and $MH_i$-$SynchT_{n-2}$ is the time of the synchronization that precedes the last synchronization. Thus, according to the time of the last synchronization (i.e. $MH_i$-$SynchT_{n-1}$), the mobile database in $MH_i$ is assumed to be in CA state in the marking $m_0$ if it contains all recent resolved updates. And if for each marking $m^{-}$ reachable from $m_0$, the mobile database contains a set of recent updates, this means that $m^{-}$ is equivalent to $m_0$.

### 4.2 System Behavior

The system behavior (i.e. evolution in time or dynamic changing of markings) is simulated by firing of transitions. The mechanism of firing in SynchSPN is based on the type of transition as follows.

- If $t$ represents the execution of IIRA instance on the higher level, then $t$ will remove the updates that exist in the input place and add them to the previously accumulated recent updates on the synchronization state of the IIRA in the higher level. The accumulated updates represent the updates received from other underlying hosts before the execution of IIRA instance on the higher level in the same time period for updates collection.
- If $t$ represents the execution of IIRA instance on the lower level, then $t$ will remove the recent resolved updates that exist in the input place and add them to the synchronization state of the

replicated database of the host in the lower level and this happens after removing the set of updates that are propagated from the host in the lower level and are included in the set of the recent resolved updates. This is to avoid storing same updates once again.

- If *t* represents the retrieving of recent updates, then *t* will take a snapshot of the recent updates from the input place and add them to the synchronization state for IIRA for execution in either the lower or higher level. This transition does not remove the recent updates from the input source, which represent the synchronization state of the replicated database according to the fact that the retrieving process does not change the state of the database.

Note that when the connection takes place between any two hosts, the firing of the transition that represents the retrieving of recent updates always occurs before the firing of the other two types. This is because updates should be retrieved first before propagation them to the other host.

Based on the initial marking and the firing mechanism, the changing of the markings of SynchSPN is tracked starting from the time of the current synchronization of MHi with the fixed network, which is denoted by $MH_i$-$SynchT_n$ and ending with the time of the next synchronization, which is denoted by $MH_i$-$SynchT_{n+1}$. For obtaining a set of last updates, $MH_i$-$SynchT_{n+1}$ should occur in this tracking after firing of all transitions. The tracking of marking changing is also depends on the fact that the $MH_i$ will obtain the last updates that are performed on both fixed and mobile networks only after propagating these updates from their sources to the higher levels , where these updates are resolved. Therefore, bottom-up propagation is considered first then the top-down propagation. Thus, the firing sequence of the transitions is divided into two sequences as shown in table 4.

| Propagation type | Firing sequence |
|---|---|
| Bottom-Up | $t_1 \rightarrow t_2 \rightarrow t_{12} \rightarrow t_3 \rightarrow t_9 \rightarrow t_4 \rightarrow t_6 \rightarrow t_5$ |
| Top-Down | $t_7 \rightarrow t_8 \rightarrow t_{10} \rightarrow t_{11} \rightarrow t_{13} \rightarrow t_{14}$ |

**TABLE 4:** The Firing Sequence of the Transitions

According to the specified firing sequence, the set of all reachable markings from $m_0$ are shown in table 5. This set represents the evolution of the system in the period $MH_i$-$SynchT_{n+1}$ - $MH_i$-$SynchT_n$. The marking that reachable from firing of $t_{14}$ represents the marking in which the replicated database in $MH_i$ should obtain a set of recent updates that occurred or propagated to the fixed network during that period. This means that this marking is equivalent to $m_0$.

| | $p_1$ | $p_2$ | $p_3$ | $p_4$ | $p_5$ | $p_6$ | $p_7$ | $p_8$ | $p_9$ | $p_{10}$ | $p_{11}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $m_0$ | 0 | 0 | 0 | 0 | $\|MH\text{-}DBS\|_i$ | $\|CS\text{-}DBS\|_j$ | $\|ZS\text{-}DBS\|_k$ | $\|MS\text{-}DBS\|$ | 0 | 0 | 0 |
| $m^-_0$ | $\|MH\text{-}UR\|_i$ | 0 | 0 | 0 | $\|MH\text{-}DBS\|_i$ | $\|CS\text{-}DBS\|_j$ | $\|ZS\text{-}DBS\|_k$ | $\|MS\text{-}DBS\|$ | 0 | 0 | 0 |
| $m_1$ | 0 | $T_{MH\text{-}CS}$ | 0 | 0 | $\|MH\text{-}DBS\|_i$ | $\|CS\text{-}DBS\|_j$ | $\|ZS\text{-}DBS\|_k$ | $\|MS\text{-}DBS\|$ | 0 | 0 | 0 |
| $m^-_1$ | 0 | $\|CS\text{-}RUR\|_j$ | 0 | 0 | $\|MH\text{-}DBS\|_i$ | $\|CS\text{-}DBS\|_j$ | $\|ZS\text{-}DBS\|_k$ | $\|MS\text{-}DBS\|$ | 0 | 0 | 0 |
| $m_2$ | 0 | 0 | $T_{CS\text{-}ZS}$ | 0 | $\|MH\text{-}DBS\|_i$ | $\|CS\text{-}DBS\|_j$ | $\|ZS\text{-}DBS\|_k$ | $\|MS\text{-}DBS\|$ | 0 | 0 | 0 |
| $m^-_2$ | 0 | 0 | $\|ZS\text{-}RUR\|_k$ | 0 | $\|MH\text{-}DBS\|_i$ | $\|CS\text{-}DBS\|_j$ | $\|ZS\text{-}DBS\|_k$ | $\|MS\text{-}DBS\|$ | 0 | 0 | 0 |
| $m_3$ | 0 | 0 | 0 | $T_{ZS\text{-}MS}$ | $\|MH\text{-}DBS\|_i$ | $\|CS\text{-}DBS\|_j$ | $\|ZS\text{-}DBS\|_k$ | $\|MS\text{-}DBS\|$ | 0 | 0 | 0 |
| $m^-_3$ | 0 | 0 | 0 | $\|MS\text{-}RUR\|$ | $\|MH\text{-}DBS\|_i$ | $\|CS\text{-}DBS\|_j$ | $\|ZS\text{-}DBS\|_k$ | $\|MS\text{-}DBS\|$ | 0 | 0 | 0 |
| $m_4$ | 0 | 0 | 0 | 0 | $\|MH\text{-}DBS\|_i$ | $\|CS\text{-}DBS\|_j$ | $\|ZS\text{-}DBS\|_k$ | $\|MS\text{-}DBS\|+$ $\|MS\text{-}RUR\|$ | 0 | 0 | 0 |
| $m^-_4$ | 0 | 0 | 0 | 0 | $\|MH\text{-}DBS\|_i$ | $\|CS\text{-}DBS\|_j$ | $\|ZS\text{-}DBS\|_k$ | $\|MS\text{-}DBS\|+$ $\|MS\text{-}RUR\|$ | 0 | 0 | $\|MS\text{-}ReRU\|$ |
| $m_5$ | 0 | 0 | 0 | 0 | $\|MH\text{-}DBS\|_i$ | $\|CS\text{-}DBS\|_j$ | $\|ZS\text{-}DBS\|_k +$ $T_{MS\text{-}ZS}$ | $\|MS\text{-}DBS\|+$ $\|MS\text{-}RUR\|$ | 0 | 0 | 0 |
| $m^-_5$ | 0 | 0 | 0 | 0 | $\|MH\text{-}DBS\|_i$ | $\|CS\text{-}DBS\|_j$ | $\|ZS\text{-}DBS\|_k +$ $T_{MS\text{-}ZS}$ | $\|MS\text{-}DBS\|+$ $\|MS\text{-}RUR\|$ | 0 | $\|ZS\text{-}ReRU\|_k$ | 0 |
| $m_6$ | 0 | 0 | 0 | 0 | $\|MH\text{-}DBS\|_i$ | $\|CS\text{-}DBS\|_j +$ $T_{ZS\text{-}CS}$ | $\|ZS\text{-}DBS\|_k +$ $T_{MS\text{-}ZS}$ | $\|MS\text{-}DBS\|+$ $\|MS\text{-}RUR\|$ | 0 | 0 | 0 |
| $m^-_6$ | 0 | 0 | 0 | 0 | $\|MH\text{-}DBS\|_i$ | $\|CS\text{-}DBS\|_j +$ $T_{ZS\text{-}CS}$ | $\|ZS\text{-}DBS\|_k +$ $T_{MS\text{-}ZS}$ | $\|MS\text{-}DBS\|+$ $\|MS\text{-}RUR\|$ | $\|CS\text{-}ReRU\|_j$ | 0 | 0 |
| $m_7$ | 0 | 0 | 0 | 0 | $\|MH\text{-}DBS\|_i +$ $T_{CS\text{-}MH}$ | $\|CS\text{-}DBS\|_j +$ $T_{ZS\text{-}CS}$ | $\|ZS\text{-}DBS\|_k +$ $T_{MS\text{-}ZS}$ | $\|MS\text{-}DBS\|+$ $\|MS\text{-}RUR\|$ | 0 | 0 | 0 |

**TABLE 5:** The Marking Table

In the marking table, the marking $m_7$ is equivalent to $m_0$ because the former represents the state in which the mobile database in MHi receives a set of recent updates that occurred or propagated to the fixed network during the period: $MH\text{-}Synch T_{n+1}$ - $MH\text{-}Synch T_n$

The marking table includes the following Equations:

$$T_{MH\text{-}CS} = \sum_{x=I, x\neq i}^{I} |MH - UR|_x + |MH\text{-}UR|_i \quad (1)$$

Where $T_{MH\text{-}CS}$ is the total number of updates that will be propagated to *CSj* during its updates collection period form *I* mobile hosts.

$$|CS\text{-}RUR|_j = T_{MH\text{-}CS} + |CS\text{-}UR|_j \quad (2)$$

This equation represents the total number of resolved updates that will be propagated from CSj to ZSk

$$T_{CS\text{-}ZS} = \sum_{y=I, y\neq j}^{N} |CS - RUR|_y + |CS\text{-}RUR|_j \quad (3)$$

Where $T_{CS\text{-}ZS}$ is the total number of updates, which will be propagated to *ZSk* during its updates collection period form *N* cell servers.

$$|ZS\text{-}RUR|_k = T_{CS\text{-}ZS} + |ZS\text{-}UR|_k \quad (4)$$

This equation represents the total number of resolved updates that will be propagated from ZSk to MS.

$$T_{ZS\text{-}MS} = \sum_{z=1, z\neq j}^{M} |ZS - RUR|_z + |ZS\text{-}RUR|_k \quad (5)$$

Where $T_{ZS\text{-}MS}$ is the total number of updates that will be propagated to *MS* during its updates collection period form M zone servers.

$$|MS\text{-}RUR| = T_{ZS\text{-}MS} + |MS\text{-}UR| \quad (6)$$

This equation represents the total number of resolved updates that will be stored in the database that exists in MS.

$$T_{MS\text{-}ZS} = |MS\text{-}ReRU| - |ZS\text{-}RUR|_k \quad (7)$$

Where $T_{MS\text{-}ZS}$ is the total number of resolved updates that will be propagated to $ZS_k$ database from *MS* excluding updates that previously propagated from $ZS_k$

$$T_{ZS\text{-}CS} = |ZS\text{-}ReRU|_k - |CS\text{-}RUR|_j \quad (8)$$

Where $T_{ZS\text{-}CS}$ is the total number of resolved updates that will be propagated to CSj database from ZSk excluding updates that previously propagated from CSj.

$$T_{CS\text{-}MH} = |CS\text{-}ReRU|_k - |MH\text{-}UR|_j \quad (9)$$

Where $T_{CS\text{-}MH}$ is the total number of resolved updates that will be propagated to MHi database from CSj excluding updates that previously propagated from MHi.

**Reachability graph.** This graph is described in figure 4. The markings $m_i^-$ , where i=0,1,…,6 that reachable from firing of retrieve transitions are not included in the reachability graph because this type of transitions affects only the local synchronization state for IIRA database. However, these transitions are included, since their firing precedes the firing of the execution transitions that leads to the markings $m_j$, where j=1, 2,…, 7.

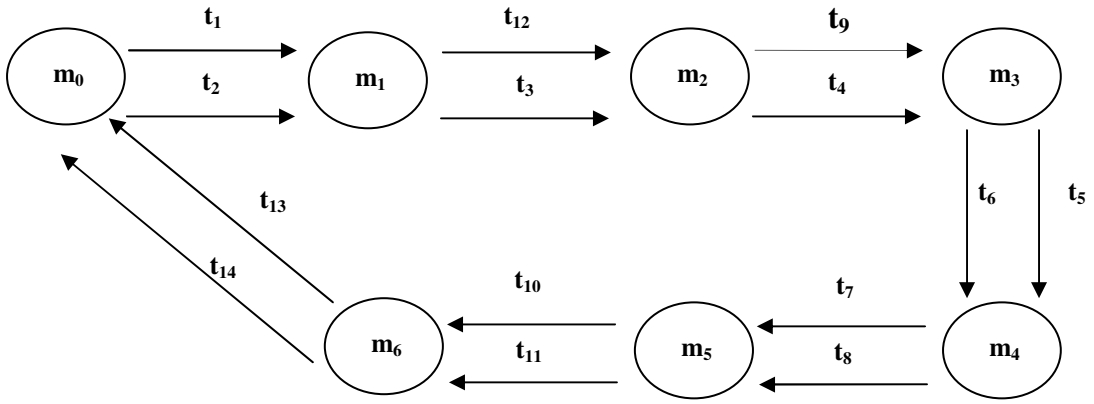



**FIGURE 4:** Reachability Graph

**Equivalent Markov chain.** In the derived CTMS (see figure 5), the firing rates of the retrieve transitions are not considered because as mentioned previously, the retrieve operation is performed locally from the replicated database on a given host.

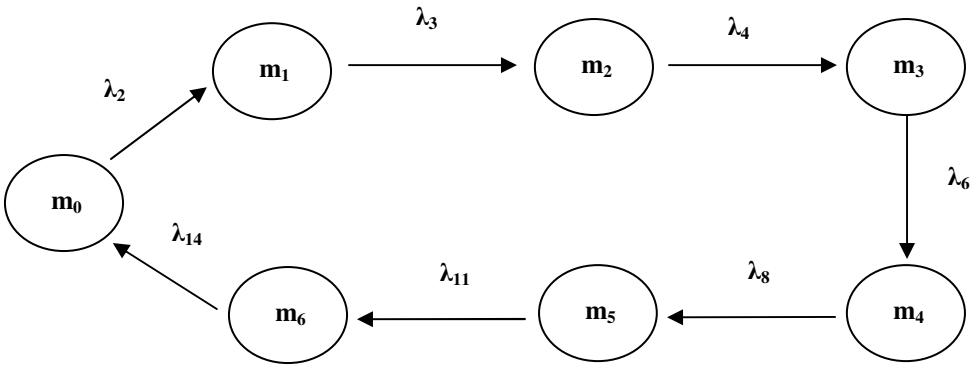



**FIGURE 5:** Derived Markov Chain

**Analysis of the Markov chain**. The steady-state probabilities, denoted by $\prod = (\pi_0, \pi_1, \pi_2, \ldots, \pi_6)$ are obtained by solving the following equations:

$$\prod A = 0 \qquad (10)$$

$$\sum_{i=0}^{6} \pi_i = 1 \qquad (11)$$

Where $A$ is the transition rate matrix. $\pi_i$ is the steady-state probability of marking that is equivalent to $m_i$.

The obtained matrix for the derived Markov chain is shown in figure 6.

| | $m_0$ | $m_1$ | $m_2$ | $m_3$ | $m_4$ | $m_5$ | $m_6$ |
|---|---|---|---|---|---|---|---|
| $m_0$ | $-\lambda_2$ | $\lambda_2$ | 0 | 0 | 0 | 0 | 0 |
| $m_1$ | 0 | $-\lambda_3$ | $\lambda_3$ | 0 | 0 | 0 | 0 |
| $m_2$ | 0 | 0 | $-\lambda_4$ | $\lambda_4$ | 0 | 0 | 0 |
| $m_3$ | 0 | 0 | 0 | $-\lambda_6$ | $\lambda_6$ | 0 | 0 |
| $m_4$ | 0 | 0 | 0 | 0 | $-\lambda_8$ | $\lambda_8$ | 0 |
| $m_5$ | 0 | 0 | 0 | 0 | 0 | $-\lambda_{11}$ | $\lambda_{11}$ |
| $m_6$ | $\lambda_{14}$ | 0 | 0 | 0 | 0 | 0 | $-\lambda_{14}$ |

**FIGURE 6:** Transition rate matrix

By solving Eq. 1 and Eq. 2, the obtained steady-state probabilities as follows:

$$\Pi = \begin{pmatrix} \pi_0 \\ \pi_1 \\ \pi_2 \\ \pi_3 \\ \pi_4 \\ \pi_5 \\ \pi_6 \end{pmatrix} = \begin{pmatrix} 1/(1+\lambda_2\omega_0) \\ 1/(1+\lambda_3\omega_1) \\ 1/(1+\lambda_4\omega_2) \\ 1/(1+\lambda_6\omega_3) \\ 1/(1+\lambda_8\omega_4) \\ 1/(1+\lambda_{11}\omega_5) \\ 1/(1+\lambda_{14}\omega_6) \end{pmatrix}$$

Where $\omega_0 = 1/\lambda_3+1/\lambda_4+1/\lambda_6+1/\lambda_8+1/\lambda_{11}+1/\lambda_{14}$, $\omega_1 = \omega_0 -(1/\lambda_3+1/\lambda_2)$, $\omega_2 = \omega_0 -(1/\lambda_4+1/\lambda_2)$, $\omega_3 = \omega_0 -(1/\lambda_6+1/\lambda_2)$, $\omega_4 = \omega_0 -(1/\lambda_8+1/\lambda_2)$, $\omega_5 = \omega_0 -(1/\lambda_{11}+1/\lambda_2)$, $\omega_6 = \omega_0 -(1/\lambda_{14}+1/\lambda_2)$
As previously mentioned, the research interests in the state in which the mobile database in the mobile host contains a set of recent updates. Therefore, the value of $\pi_0$ represents the probability of the marking that is equivalent to $m_0$.

**Assertion 4.2.1.** There is only a subset $C \subseteq R(m_0)$, such that the mobile database in specific mobile host in CA State for each $m \in C$.
**Proof**. Let $t_j$ ($j=1,\ldots,m$) denotes the transition that represents the execution state of the CSR-IIRA in $MH_i$, and $t_i$ ($i=1,\ldots,n$) denotes the transition that represents the execution state of the MHR-IIRA in the fixed network. We show that firing of $t_j$ will result in CA State for $R$ that is hosted in $MHi \Leftrightarrow$ each $t_i$ is fired during the time period that precedes the current synchronization time of $MH_i$. Since the latter condition is not realized at all synchronization times for $MH_i$ due to existing of many MHs are not connected before the synchronization of MHi with the cell server. Therefore, the firing of $t_j$ leads to CA State $\Leftrightarrow$ all updates that are performed in the mobile network are propagated to the fixed network before the synchronization of $MH_i$. This means that if the latter condition is realized, the firing of $t_j$ will results in a marking that represents an element of $C$.

**Assertion 4.2.2.** The probability that the mobile database in CA state is:

$$P(\text{CA}) = n\pi_0 \qquad (12)$$

Where $n$ is the number of synchronization times for $MH_i$ with the fixed network that led to the CA state.
**Proof.** Let $C$ be the subset of $R(m_0)$ satisfying the condition that the place $p_5$ has received all recent updates that occurred and resolved before the synchronization time of $MH_i$ with the fixed network, which led to each marking in $C$. Then the probability of this condition is: $P(C) = \sum_{i \in C} \pi_i$ ,
Since the probability that $MH_i$ receives a set of recent updates is $\pi_0$. Then $\pi_i = \pi_0$ for each marking $m_i \in$ C. This means that $P(C) = \sum_{i=1}^{n} \pi_i = n\pi_0$ , where $n$ is the number of markings in $C$. This number is equivalent to the number of the synchronization times with the fixed network that led to storing all recent updates in $p_5$.
**Assertion 4.2.3.** The marking $m_0$, where $p_5$ receives a set of recent resolved updates is recurrent after a time period equals to: $MH_i\text{-}SynchT_{n+1} - MHi\text{-}SynchT_n$
, where $MHi\text{-}SynchT_{n+1} - MHi\text{-}SynchT_n > \lambda_5 + \lambda_8 + \lambda_{11}$, $MHi\text{-}SynchT_n$ is the time instant of the current synchronization with the fixed network, $MH_i\text{-}SynchT_{n+1}$ is the time instant of the next synchronization with the fixed network.
**Proof.** $m_0$ is recurrent if the following equation is satisfied.

$$\sum_{i=MHi\text{-}SynchTn}^{MHi\text{-}SynchTn+1} P^{i}_{m_0 m_0} = 1 \qquad (13)$$

Where $P^{i}_{m0m0}$ is the probability that the system returns to state $m_0$ starting from $m_0$. Suppose that the markings that are reachable from $m_0$ occur at the following time instants:
$MHi\text{-}SynchT_n, T_1, T_2, \ldots, T_n, MHi\text{-}SynchT_{n+1}$, where $MHi\text{-}SynchT_n < T_1 < T_2 < \ldots < T_n < MHi\text{-}SynchT_{n+1}$. Obviously, the $MH_i$ can obtain the last updates that occurred or are propagated to the fixed network during the period: $MHi\text{-}SynchT_{n+1} - MHi\text{-}SynchT_n$ after a time instant > = $MHi\text{-}SynchT_{n+1}$, which means that:
$P^{MHi\text{-}SynchTn}{}_{m0m0} = 0$, $P^{MHi\text{-}SynchTn+1}{}_{m0m0} = 1$, $P^{T1}{}_{m0m0} = 0$, $P^{T2}{}_{m0m0} = 0$, …, $P^{T0}{}_{m0m0} = 0$
To obtain the latest updates, the following condition should hold:

$$MH_i\text{-}SynchT_{n+1} - MH_i\text{-}SynchT_n > \lambda_3 + (\lambda_4 - \lambda_3) + (\lambda_5 - \lambda_4) + \lambda_8 + \lambda_{11} = \lambda_5 + \lambda_8 + \lambda_{11}$$

Where $\lambda_3 < \lambda_4 < \lambda_5$. This is because the server in the master level receives updates from all underlying levels, while the servers in the zone and cell levels receive updates from the hosts that are located in their areas.

**Assertion 4.2.4.** The SynchSPN is deadlock free.
**Proof**. We prove that $\forall\ m \in R(m_0)$, $\exists\ m^{-} \in R(m)$ such that $\exists\ t$ is enabled for $m^{-}$, where $R(m_0)$ is the set of markings reachable from $m_0$ by firing a sequence of transitions. Recall that our assumptions regarding that tens of updates per each data item are expected at any period of time and the connection is reliable and fixed between the servers of the fixed network, this means that the following conditions are true:

1. At any period of time, $\exists\ DB \in RDB$ such that *DB* is updated recently (has UR) where *RDB* is the set of the replicated databases in either fixed or mobile hosts. This condition ensures enabling of selection transactions.
2. There is at least one host (either fixed or mobile) is synchronized with other host (e.g. MH with a cell server or other MH, FH with a server, CS with ZS…etc). This condition ensures that there is at least one of the execution transitions is enabled after satisfying condition one.

## 5. CONTRIBUTION AND DISCUSSIONS

The contributions of this paper can be summarized as follows: firstly, a new replication strategy is presented for replicating data by considering a logical three levels architecture and a multi-agent based replication method. The strategy supports frequent disconnections and mobility of hosts by enabling the users to perform their updates in a disconnected mode and then synchronizing their updates with the higher levels. Second, the proposed architecture supports scalability by allowing large numbers of updateable replicas in the cell level and higher levels, since the large scale distributed database systems require such a feature. Third, the paper combines both optimistic and pessimistic replication approaches, in a hybrid manner that exploits the pertinent features of each in mobile environments.

The IIRA-dependant multi-agent system achieves load balance in both propagation and ordering processes. This is because these processes are shared by multiple hosts, where each host propagates a set of the recent updates to another in either lower or higher level and each host participates in ordering the updates that are issued in its replicated database or collected from underlying levels.

To decrease the communication cost and provide a better utilization of the connection time in environments prone to more frequent disconnections and failures, the proposed strategy relies on instance immigration instead of the migration of the agent itself as in mobile agents' communities. When the connection takes place, the instance holding the updates-result will migrate to the other host, and performs its task. Then it removes itself without needing to return back as in mobile agents' communities. This minimizes the connection cost to only cost that is needed to transfer one instance per connection time.

The availability of all recent updates that occurred in both fixed and mobile networks to the mobile hosts depends directly on the propagation of these updates from their sources to the fixed network. This propagation should takes place before the synchronization of the mobile host with the fixed network.

The replication method ensures achieving the consistency of data in the mobile network according to the time of last connection happened. Therefore, data inconsistency here will depend on the difference between the connection times.

To minimize the rate of update conflicts, and taking a step on ensuring eventual consistency, data are divided into two types: infrequently changed data that are updated only in the master level and frequently changed data that are updated on the underlying levels.

## 6. CONCLUSTIONS

In this paper, our research has focused on proposing a new replication strategy to maintain consistency and improve availability of data in large scale mobile environments. The replication strategy encompassed replication architecture and replication method as a binary combination that is needed to achieve such a goal. To exploit the features of both optimistic and pessimistic replication, the new strategy is based on a hybrid approach that divides data into frequently changed data and infrequently changed data, and then updates are restricted or allowed according to these types. Stochastic Petri Net is developed to model the dynamic behavior of the replicated system in regard to reaching the Consistent-Available state for the replicated database at the mobile host.

As a part of our future research, a plan will be provided to develop the required tools and interfaces to implement the proposed strategy in mobile healthcare environments to provide healthcare practitioners with an efficient access to healthcare data.

## 7. REFERENCES

1. T. Connolly and C. E. Begg. "*Database Systems: A Practical Approach to Design, Implementation and Management*". 4th edition, Addison-Wesley, (2004)

2. S. Madria and S. Bhowdrick. "*Mobile data management*". Potentials, IEEE, 20(4):11 – 15, 2001

3. T. Imielinksi and B. Badrinath. "*Wireless mobile computing: challenges in data management*". Communications of ACM, 37(10):18-28, 1994

4. Y. Saito and M. Shapiro. "*Optimistic replication*". ACM Computing Surveys (CSUR), 37(1):42–81, 2005

5. J. Gray, P. Helland , and P. O'Neil. "*The dangers of replication and a solution*". In Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 173–182, 1996

6. M. Wiesmann, A. Pedone, B. Schiper, G. Kemme and Alonso. "*Understanding replication in databases and distributed systems*". In Proceedings of the 20th International Conference on Distributed Computing Systems ( ICDCS 2000), pp. 464, 2000

7. P. Bernstein. "*Principles of Transaction Processing*". Morgan Kaufmann Publishers Inc, (1997)

8. A. Helal, A. Heddaya and B. Bhargava. "*Replication Techniques in Distributed Systems*". Kluwer Academic Publishers, 1996

9. D. Ratner, P. Reiher and G. Popek. "*Roam: a scalable replication system for mobility*". Mobile Network and Applications, 9(5):537-544, 2004

10. D. Ratner, P. Reiher, G. Popek and G.Kuenning. "*Replication requirements in mobile environments*". Mobile Networks and Applications, 6(6): 525–533, 2001

11. J. Monteiro, A. Brayner and S. Lifschitz. "*A mechanism for replicated data consistency in mobile computing environments*". In Proceedings of the ACM symposium on Applied computing, Seoul, Korea, pp. 914 – 919, 2007

12. J. Abawajy, M. Deris and M. Omer. "*A novel data replication and management protocol for mobile computing systems".* Mobile Information Systems, 2(1):3-19, IOS Press, Netherlands, 2006

13. N. Tolia, M. Satyanarayanan and A. Wolbach. "*Improving mobile database access over wide-area networks without degrading consistency*". In Proceedings of the 5th international conference on Mobile systems, applications and services, San Juan, Puerto Rico, pp.71 – 84, 2007

14. G. Balbo. "*Introduction to generalized stochastic Petri Nets*". Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 4486/2007: 83-131, 2007

15. G. Balbo., "*Introduction to stochastic Petri Nets*" Lecture Notes in Computer Science, Springer Berlin / Heidelberg, Volume 2090/2001, pp. 84-155, 2001

16. M. Ajmone, G. Balbo, G. Conte, S. Donatelli and G. Franceschinis. "*Modeling with Generalized Stochastic Petri Nets*". J. Wiley, Chichester. (1995)

17. M. Ajmone, G. Balbo and G. Conte. "*Performance models of multiprocessor systems*". MIT Press, Cambridge, 1986

18. F. Bause and P. Kritzinger. "*Stochastic Petri Nets -- An Introduction to the Theory*," 2nd edition, Springer Verlag, Germany, 2002

19. T. Murata. "*Petri Nets: properties, analysis and applications*". In Proceedings of the IEEE, pp.541–580, 1989

# Data Quality Mining using Genetic Algorithm

**Sufal Das** sufal.das@gmail.com
*Lecturer, Department of Information Technology*
*Sikkim Manipal Institute of Technology*
*Rangpo-737136, India*

**Banani Saha** bsaha_29@yahoo.com
*Reader. Department of Computer Science & Engineering*
*Calcutta University*
*Kolkata, Pin 700073, India*

**ABSTRACT**

Data quality mining (DQM) is a new and promising data mining approach from the academic and the business point of view. Data quality is important to organizations. People use information attributes as a tool for assessing data quality. The goal of DQM is to employ data mining methods in order to detect, quantify, explain and correct data quality deficiencies in very large databases. Data quality is crucial for many applications of knowledge discovery in databases (KDD). In this work, we have considered four data qualities like accuracy, comprehensibility, interestingness and completeness. We have tried to develop Multi-objective Genetic Algorithm (GA) based approach utilizing linkage between feature selection and association rule. The main motivation for using GA in the discovery of high-level prediction rules is that they perform a global search and cope better with attribute interaction that the greedy rule induction algorithms often used in data mining.

**Keywords:** Data Quality, Genetic Algorithm, Multi-objective Optimization, Association Rule Mining.

## 1. INTRODUCTION

The main contribution of this paper is to give a first impression of how data mining techniques can be employed in order to improve data quality with regard to both improved KDD results and improved data quality as a result of its own. That is, we describe a first approach to employ association rules for the purpose of data quality mining [1]. Data mining is about extracting interesting patterns from raw data. There is some agreement in the literature on what qualifies as a "pattern", but only disjointed discussion of what "interesting" means. Problems that hamper effective statistical data analysis stem from many source of error introduction. Data mining algorithms like "Association Rule Mining" (ARM) [2,3] perform an exhaustive search to find all rules satisfying some constraints. Hence, the number of discovered rules from database can be very large. Typically the owner of the data is not fully aware of data quality deficiencies. The system might have been doing a good job for years and the owner probably has its initial status in mind. Doubts concerning data quality may raise astonishment or even disaffection. We often have been facing exactly this situation. By trying to make the best of it we employed our skills – data mining techniques – as a patched-up solution to measure, explain, and improve data quality. Based on the earlier works, it is clear that it is difficult to identify the most effective rule. Therefore, in many applications, the size of the dataset is so large that learning might not work well. The

generated rule may have a large number of attributes involved in the rule thereby making it difficult to understand. If the generated rules are not understandable to the user, the user will never use them. Again, since more importance is given to those rules, satisfying number of records, these algorithms may extract some rules from the data that can be easily predicted by the user. It would have been better for the user, if the algorithms can generate some of those rules that are actually hidden inside the data. Also, the algorithm should capture all attributes which are useful. By introducing data quality mining (DQM) we hope to stimulate research to reflect the importance and potentials of this new application field. In this paper the authors have considered Association Rule Mining and tried to improve this technique by applying Multi-objective Genetic Algorithms (MOGA) [9] on the rules generated by Association Rule Mining based on four data qualities (objectives): accuracy, comprehensibility, interestingness and completeness. A brief introduction about Association Rule Mining and GA is given in the following sub-sections, followed by methodology, which will describe the basic implementation details of Association Rule Mining and GAs. The authors will discuss the results followed by conclusion in the last section.

## 2. RELATED WORKS

### 2.1 Association Rule Mining

Let $I = \{I_1, I_2 \ldots\ldots, I_m\}$ be a set of m distinct attributes, also called literals. Ai = r is an item, where r is a domain value is attribute, Ai in a relation, R ( A1, …, An). A is an itemset if it is a subset of I. D =. { ti , $t_{i+1}$ ,……, $t_n$) is a set of transactions, called the transaction (tid, t-itemset). A transaction t contains an itemset A if and only if, for all items $i \in A$, i is in t-itemset. An itemset A in a transaction database D has a support, denoted as Supp (A) (we also use p(A) to stand for Supp (A)), that is the ratio of transactions in D contain A. Supp(A) = |A(t)| / |D|, Where A(t) = {t in D/t contains A}. An itemset A in a transaction database D is called a large (frequent) itemset if its support is equal to, or greater than, a threshold of minimal support (minsupp), which is given by users or experts. An association rule is an expression of the form IF A THEN C(or A -> C), $A \cap C = \phi$, where A and C are sets of items. The meaning of this expression is that transactions of the databases, which contain A, tend to contain C. Each association rule has two quality measurements: support and confidence, defined as:

1) The support of a rule A->C is the support of $A \cup C$, where $A \cup C$ means both A and C occur at the same time.
2) The confidence or predictive accuracy [2] of a rule A->C is conf (A->C) as the ratio: $|(A \cup C)(t)| / |A(t)$ or $Supp(A \cup C) / Supp(A)$.

That is, support = frequencies of occurring patterns; confidence = strength of implication. Support-confidence framework (Agrawal et al. 1993): Let I be the set of items in database D, A, $C \subseteq I$ be itemset, $A \cap C = \phi$, p(A) is not zero and p(C) is not zero. Minimal support minsupp) and minimal confidence (minconf) are given by users or exparts. Then A->C is a valid rule if

1. Supp ($A \cup C$) is greater or equal to minsupp,
2. Conf (A->C) is greater or equal to minconf.

Mining association rules can be broken down into the following two sub-problems:

1. Generating all itemsets that have support greater than, or equal to, the user specified minimal support. That is, generating all large itemsets.
2. Generating all the rules that have minimum confidence.

### 2.2 Genetic Algorithm

Genetic Algorithm (GA) [8] was developed by Holland in 1970. This incorporates Darwinian evolutionary theory with sexual reproduction. GA is stochastic search algorithm modeled on the process of natural selection, which underlines biological evolution. GA has been successfully applied in many search, optimization, and machine learning problems. GA process in an iteration manner by generating new populations of strings from old ones. Every string is the encoded binary, real etc., version of a candidate solution. An evaluation function associates a fitness measure to every string indicating its fitness for the problem. Standard GA apply genetic operators such selection, crossover and mutation on an initially random population in order to compute a whole generation of new strings.

- **Selection** deals with the probabilistic survival of the fittest, in that more fit chromosomes are chosen to survive. Where fitness is a comparable measure of how well a chromosome solves the problem at hand.
- **Crossove**r takes individual chromosomes from P combines them to form new ones.
- **Mutation** alters the new solutions so as to add stochasticity in the search for better solutions.

In general the main motivation for using GAs in the discovery of high-level prediction rules is that they perform a global search and cope better with attribute interaction than the greedy rule induction algorithms often used in data mining. This section of the paper discusses several aspects of GAs for rule discovery.

## 3. METHODOLOGY

Representation of rules plays a major role in GAs, broadly there are two approaches based on how rules are encoded in the population of individuals ("Chromosomes") as discussed in Michigan and Pittsburgh Approach [12]; The pros and cons as discussed in [12] is as follows, Pittsburgh approach leads to syntactically-longer individuals, which tends to make fitness computation more computationally expensive. In addition, it may require some modifications to standard genetic operators to cope with relatively complex individuals. By contrast, in the Michigan approach the individuals are simpler and syntactically shorter. This tends to reduce the time taken to compute the fitness function and to simplify the design of genetic operators. However, this advantage comes with a cost. First of all, since the fitness function evaluates the quality of each rule separately, now it is not easy to compute the quality of the rule set as a whole - i.e. taking rule interactions into account. In this paper Michigan's approach is opted i.e. each individual encodes single rule. The encoding can be done in a number of ways like, binary encoding or expression encoding etc. For example let's consider a rule "If a customer buys milk and bread then he will also buy butter", which can be simply written as

**If** milk **and** bread **then** butter

Now, in the Michigan approach where each chromosome represents a separate rule. In the original Michigan approach we have to encode the antecedent and consequent parts separately; and thus this may be an efficient way from the point of space utilization since we have to store the empty conditions as we do not known a priori which attributes will appear in which part. So we will follow a new approach that is better than this approach from the point of storage requirement. With each attribute we associate two extra tag bits. If these two bits are 00 then the attribute next to these two bits appears in the antecedent part and if it is 11 then the attribute appears in the consequent part. And the other two combinations, 01 and 10 will indicate the absence of the attribute in either of these parts. So the rule AEF->BC will look like 00A 11B 11C 01D 00E 00F. In this way we can handle variable length rules with more storage efficiency, adding only an overhead of 2k bits, where k is the number of attributes in the database. The next step is to find a suitable scheme for encoding/decoding the rules to/from binary chromosomes. Since the positions of attributes are fixed, we need not store the name of the attributes. We have to encode the values of different attribute in the chromosome only.

### 3.1 Multi-objective Optimization

Although it is known that genetic algorithm is good at searching for undetermined solutions, it is still rare to see that genetic algorithm is used to mine association rules. We are going to further investigate the possibility of applying genetic algorithm to the association rules mining in the following sections. As genetic algorithm is used to mine association rule, among all measurements, one measurement is accuracy or confidence factor. In the present work we have used another three measures of the rules like comprehensibility [9], interestingness [10] and completeness, in addition to predictive accuracy. Using these four measures, some previously unknown, easily understandable and compact rules can be generated. It is very difficult to quantify understandability or comprehensibility. A careful study of an association rule will infer that if the number of conditions involved in the antecedent part is less, the rule is more comprehensible. To reflect this behavior, an expression was derived as comp = N – (Number of conditions in the antecedent part). This expression serves well for the classification rule generation where the number of attributes in the consequent part is always one. Since, in the association rules, the consequent part may contain more than one attribute; this expression is not suitable for the association rule mining. We require an expression where the number of attributes involved in both the parts of the rule has some effect. The following expression can be used to quantify the comprehensibility of an association rule,

$$\text{Comprehensibility} = \log (1 + |C| / (|D| - |A|)) * (1 / |A|)$$

Here, |C| and |A| are the number of attributes involved in the consequent part and the antecedent part, respectively and |D| is the total number of records in the database.

It is very important that whatever rule will be selected for useful one this rule should represent all useful attributes or components. For that we have to select compact association rule with all useful features. So, we have to find out the frequent itemset with maximum length. The antecedent part and consequent for an association rule should cover all useful features as well as the two parts should be frequent. The following expression can be used to quantify the completeness of an association rule,

$$\text{Completeness} = (\log (1 + |C| + |A|) / |D|)* \text{Supp}(A) * \text{Supp}(C)$$

Here, |C| and |A| are the number of attributes involved in the consequent part and the antecedent part, respectively and |D| is the total number of records in the database. Supp (A) and Supp(C) are the occurrences of Antecedent part and consequent part respectively.

Since association rule mining is a part of data mining process that extracts some hidden information, it should extract only those rules that have a comparatively less occurrence in the entire database. Such a surprising rule may be more interesting to the users; which again is difficult to quantify. For classification rules it can be defined by information gain theoretic measures. This way of measuring interestingness for the association rules will become computationally inefficient. For finding interestingness the data set is to be divided based on each attribute present in the consequent part. Since a number of attributes can appear in the consequent part and they are not pre-defined, this approach may not be feasible for association rule mining. The following expression can be used to define as interestingness of an association rule,

$$\text{Interestingness} = \text{Supp}(A)* [(1- \text{Supp}(C)) /(1-\text{Supp}(AUC))]* [\text{Supp}(AU\ C) / \text{Supp}(A)] *[\text{Supp}(AUC) /\text{Supp}(C)].$$

This expression contains two parts. The first part, Supp (A)* [(1– Supp(C)) /(1– Supp(AUC))], compares the probability that A appears without C if they were dependent with the actual frequency of the appearance of A. The remaining part measures the difference of A and C appearing together in the data set and what would be expected if A and C where statistically dependent.

### 3.2 Genetic Algorithm with Modifications

• **Individual Representation** can be performed using Michigan's approach, i.e. each individual encodes single rule, as discussed in previous section.

• **Selection** is performed as the chromosomes are selected (using standard selection scheme, e.g. roulette wheel selection) using the fitness value. Fitness value is calculated using their ranks,

which are calculated from the non-dominance property of the chromosomes. A solution, say a, is said to be dominated by another solution, say b, if and only if the solution b is better or equal with respect to all the corresponding objectives of the solution a, and b is strictly better in at least one objective. Here the solution b is called a non-dominated solution. The ranking step tries to find the non-dominated solutions, and those solutions are ranked as one. Among the rest of the chromosomes, if $p_i$ individuals dominate a chromosome then its rank is assigned as $1 + p_i$. This process continues till all the chromosomes are ranked. Then fitness is assigned to the chromosomes such that the chromosomes having the smallest rank gets the highest fitness and the chromosomes having the same rank gets the same fitness. After assigning the fitness to the chromosomes, selection, replacement, crossover and mutation operators are applied to get a new set of chromosomes, as in standard GA.

### 3.3 Our Approach

Our approach works as follows:
1. Load a sample of records from the database that fits in the memory.
2. Generate N chromosomes randomly.
3. Decode them to get the values of the different attributes.
4. Scan the loaded sample to find the support of antecedent part, consequent part and the rule.
5. Find the confidence, comprehensibility, completeness and interestingness values.
6. Rank the chromosomes depending on the non-dominance property.
7. Assign fitness to the chromosomes using the ranks, as mentioned earlier.
8. Select the chromosomes, for next generation, by roulette wheel selection scheme using the fitness calculated in Step 7.
9. Replace all chromosomes of the old population by the chromosomes selected in Step 8.
10. Perform crossover and mutation on these new individuals.
11. If the desired number of generations is not completed, then go to Step 3.
12. Decode the chromosomes in the final stored population, and get the generated rules.
13.Select chromosomes based on accuracy, comprehensibility, completeness and interestingness.

## 4. INPLEMENTATION & RESULT

The proposed technique has been implemented on different data sets with satisfactory results. Here we present the results on one such data set having 47 attributes and 5338 records. Crossover and mutation probabilities were taken respectively as 0.87 and 0.0195; the population size was kept fixed as 50. Number of generations was fixed as 300. Best four rules which were selected based on accuracy, comprehensibility completeness and interestingness, are put in the following table.

| Rule No | Antecedent part | Consequent Part | A value | C value | I value | Co value |
|---|---|---|---|---|---|---|
| 1 | {4->3}, {18->0}, {28->3}, {38->0}, {39->2} | {1->1}, {5->2}, {9->0}, {12->3}, {15->0}, {17->1}, {21->1}, {24->2}, {26>2}, {29->0}, {31->2}, {33->2}, {34->3}, {37->1} | 0.2151 | 0.0601 | 0.0128 | 0.0168 |
| 2 | {5->3}, {12->2}, {13->2}, {15->0}, {24->1}, {26->3} | {1->2}, {4->2}, {9->0}, {16->3}, {17->1}, {20->1}, {21->2}, {28->3}, {30->1}, {31->1}, {33->3}, {34->2}, {35->0}, {37->2}, {38->1} | 0.1921 | 0.0519 | 0.0148 | 0.01279 |
| 3 | {1->2}, {3->2}, {5->2}, {7->1}, {12->3}, {16->2}, {21->3} | {2->0}, {4->2}, {9->1}, {13->1}, {15->0}, {17->2}, {20->3}, {24->2}, {26->3}, {28->1}, {30->0}, {31->0}, {32->0}, {34->2}, {38->1}, {39->3} | 0.2129 | 0.0418 | 0.0173 | 0.01085 |
| 4 | {5->2}, {12->2}, {15->0}, {24->1}, {31->1}, {38->2}, {39->0} | {3->0}, {4->3}, {8->3}, {9->2}, {17->1}, {21->2}, {26->2}, {28->0}, {30->3}, {31->2}, {34->0}, {36->3} | 0.1874 | 0.0329 | 0.0164 | 0.00983 |

N.B.: [A, C, Co and I stand for accuracy, comprehensibility, completeness and interestingness respectively and {1->0} stands for attribute no.1 having value 0.]

If we consider the first rule, then 19 attributes are involved. Now, we can say that out of 47 attributes, these 19 attributes are useful.
Here, we have got some association rules which optimal according their accuracy, comprehensibility, interestingness and completeness. If we study the result then, we can select 19 to 22 attributes among 47 attributes. Using completeness measurement we can say that these 19 to 22 attributes are only the useful, no more is left out. If completeness measurement is not considered then we have seen that 14 to 17 attributes were selected and all were useful according the other three measurements.

## 5. CONSLUSION & FUTURE WORK

The use of a multi-objective evolutionary framework for association rule mining offers a tremendous flexibility to exploit in further work. In this present work, we have used a Pareto based genetic algorithm to solve the multi-objective rule mining problem using four measures—completeness, comprehensibility, interestingness and the predictive accuracy. We adopted a variant of the Michigan approach to represent the rules as chromosomes, where each chromosome represents a separate rule. The approach can be worked with numerical valued attributes as well as categorical attributes.
This work is able to select all useful attributes for any sort of dataset. One big advantage of this approach is that user or expert do not need to have any knowledge the dataset. No threshold value is not used here.
This approach may not work properly is the given dataset is not homogeneous as this is applied on a sample of dataset. Sample of any dataset does not represent the whole dataset completely. In future we can work to remove this disadvantage.

## 6. REFERENCES

1. Jochen Hipp,Ulrich G¨untzer and Udo Grimmer, *"Data Quality Mining - Making a Virtue of Necessity",* In Proceedings of the 6th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD) 2001.

2. R. Agrawal, R. Srikant, *"Fast algorithms for mining association rules",* in Proceeding of the 20th Int'l Conference on Very Large Databases, Chile, 1994.

3. Imielinski, T., R. Agrawal and A. Swami, "*Mining association rules between sets of items in large databases"*. Proc. ACM SIGMOD Conf. Management of Data, pp: 207–216.

4. K.M. Faraoun, A. Rabhi, "*Data dimensionality reduction based on genetic selection of feature subsets",* EEDIS UDL University- SBA, (2002).

5. Cheng-Hong Yang, Chung-Jui Tu, Jun-Yang Chang Hsiou-Hsiang Liu Po-Chang Ko, "*Dimensionality Reduction using GA-PSO"(2001).*

6. P_adraig, "*Dimension Reduction",* Cunningham University College Dublin Technical Report UCDCSI-2007-7 August 8th, 2007

7. Erick Cantu-Paz, *"Feature Subset Selection, Class Separability, and Genetic Algorithms",* Center for Applied Scientic Computing Lawrence Livermore National Laboratory Livermore, CA, (1994).

8. M. Pei, E. D. Goodman, F. Punch, "*Feature Extraction using genetic algorithm", Case Center for Computer-Aided Engineering and Manufacturing* W. *Department of Computer Science,(2000).*

9. Sufal Das, Bhabesh Nath, "*Dimensionality Reduction using Association Rule Mining"*, IEEE Region 10 Colloquium and Third International Conference on Industrial and Information Systems (ICIIS 2008) December 8-10, 2008, IIT Kharagpur, India

10. Hsu, W., B. Liu and S. Chen, *"Ggeneral impressions to analyze discovered classificationrules",.* Proc. Of 3rd Intl. Conf. On Knowledge Discovery & Data Mining (KDD-97), pp: 31–36.AAAI Press.(1997)

11. Freitas, A.A., E. Noda and H.S. Lopes,. *"Discovering interesting prediction rules with a genetic algorithm"'*. Proc. Conf. Evolutionary Computation, (CEC-99), pp: 1322–1329.(1999)

12. Cristiano Pitangui, Gerson Zaverucha, "*Genetic Based Machine Learning:Merging Pittsburgh and Michigan, an Implicit Feature Selection Mechanism and a New Crossover Operator"*, Proceedings of the Sixth International Conference on Hybrid Intelligent Systems (HIS'06).(2006).

# Analysis of Natural Language Steganography

**Shaifizat Mansor** shaifizat@kedah.uitm.edu.my
*School of Computer Science*
*Universiti Sains Malaysia (USM)*
*Minden, Pulau Pinang, 11800, MALAYSIA*

**Roshidi Din** roshidi@uum.edu.my
*School of Computer Science*
*Universiti Sains Malaysia (USM)*
*Minden, Pulau Pinang, 11800, MALAYSIA*

**Azman Samsudin** azman@cs.usm.my
*School of Computer Science*
*Universiti Sains Malaysia (USM)*
*Minden, Pulau Pinang, 11800, MALAYSIA*

**ABSTRACT**

The technology of information hiding through an open network has developed rapidly in recent years. One of the reasons why we need tools to hide message, is to keep secret message concealed from unauthorized party. Steganography is one of the techniques in sending secret message. In this paper, several software metrics were used to analyze the common criteria in steganographic tools and measure the complexity of the tools in hiding message. Several criterias have been chosen: Percent Lines with Comments (PLwC); Average Statements per Function (ASpF) and Average Block Depth (ABD) to measure the tools complexity. The analysis process has been implemented using a single Linux platform.

**Keywords**: Steganography, Text Steganography, Secret Message, Software Metric.

## 1. INTRODUCTION

Recently, millions of documents are produced and easily accessed in the Internet [1]. Thus, the information of these documents needs to be secured and protected because the activities of document analysis [2]. Steganography is one of the popular areas in information protection. The purpose of steganography is to establish communication between two parties whose existence is unknown to a possible attacker [3]. If this is done correctly, the exchanged messages should not arouse any suspicion since the communicated information has an innocent looking and the communication itself does not require any secret key as part of its information hiding process. In text, this can be done in many ways such as inclusion of line break characters, and multiple spacing that represents a hidden message.

Steganography technique is not a new technique [4]. They are some older practices in message hiding such as invisible ink, tiny pin punctures on selected characters and pencil mark on

typewritten characters. In term of the key management, steganography is more secure than cryptography [5]. If The goal of steganography is to hide secret message in such a way that it does not arouse any eavesdropper's suspicion. Steganography is characterized as a process of hiding message in cover signal so that the message can be extracted unknowingly to the public at the receiving end. Steganography can be divided into two broad categories namely technical steganography and natural language steganography. Technical steganography is a technique of hiding information onto another medium such as image, audio, video or other digitally represented code invisibly [6]. On the other hand, natural language steganography is the art of using the natural language to conceal secret message [7]. Natural language steganography focuses on hiding information in text steganography, linguistic steganography and its hybrid as shown in FIGURE 1.

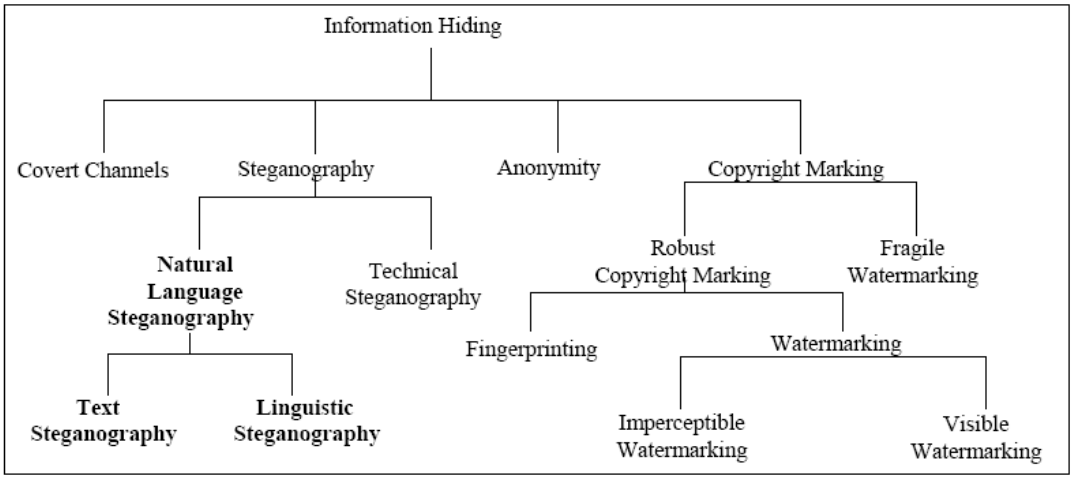



**FIGURE 1:** Field of Information Hiding and Its Classification (Adopted from [8]).

One of the linguistic steganography strength is the ability to hide a secret message during sending process [9]. The efficiency and the level of complexity to encode the hidden message is one of the linguistic steganography issues. Another important issue is the speed performance of the tools in executing the encoding process [10]. This issue raises a question on how to examine the execution time and tools complexity.

In order to improve the productivity and development of steganographic tools, the evaluation of existing steganographic tools must be carried out and use as a yardstick to improve the quality of software industry [11] especially in natural language steganography. Equally important, is the attacks analysis on steganographic tools which is deemed essential in evaluating the steganographic tools performance in order to improve the steganography algorithm [12]. Therefore, the natural language steganographic tools should be examined both from the software metric perspective as well as their robustness against attack. Thus, our main objective of this study is to analysis the performance of natural language steganographic tools based on these two perspectives.

The rest of the paper is organized as follows: In Section 2 we introduce text steganography tools that are currently being used. Section 3 discusses the parameter measurement of the selected steganographic tools in message hiding. In Section 4 we discuss the measurement of the software metric including Percent Lines with Comments *(PLwC)*, Average Statements per Function *(ASpF)*, and Average Block Depth *(ABD)*. Section 4 also discusses execution time of the steganographic tools. Section 5 provides a discussion on the evaluation of the text steganographic. Section 6 is the conclusion of this paper.

## 2. TEXT STEGANOGRAPHY

Natural language steganographic tools and techniques are becoming more widespread and need to be evaluated [13]. One of the methods in evaluating natural language steganographic tools is by employing software metric [14], which is important in determining the tools efficiency. Among the components of software metrics are cost and effect estimation, productivity measures, quality measures, reliability tools, structural complexity metrics and execution time.

This study used text steganographic tools in order to analyze the field of natural language steganography. Numerous tools have been identified as text steganographic tools [15-22]. They are: Texto, Stego, SNOW, Stegparty, Steganos, Snowdrop, PGPn123, and FFEncode. TABLE 1 shows the description of each steganography tools.

| No. | Tools | Platform | Description of encoding process |
|---|---|---|---|
| 1. | Texto | DOS:WIN(Unix Linux) | Transform uuencode or PGP-armoured ascii data |
| 2. | Steganosaurus (Stego) | C: DOS | Encode binary to text based on the dictionary from a source document |
| 3. | SNOW | C/C++: DOS WIN | Append tabs and white space to end of text lines |
| 4. | Stegparty | Unix/Linux | Alter the text on spelling and punctuation |
| 5. | Steganos | DOS | Looks at a list of words on text to find and match words |
| 6. | Snowdrop | C | Embed the text in the least significant portions of some binary output |
| 7. | PGPn123 | Window front-end to PGP | Using PGP shell tools to hide text |
| 8. | FFEncode | DOS | Using Morse code in null characters to encode message |

**TABLE 1**: The Description of Text Steganography Tools.

Among the identified text steganoography tools, only Texto, Stego, SNOW, and Stegparty are being examined. The selection is based on similarity criterion such as the steganography approach (line-shift coding, word-shift coding and feature coding) and manipulation of dictionary/corpora. In addition, the four identified tools also share common standard source code (C/C++) which are accessible from the open source. The accessibility of the codes aid tremendously in the evaluation process.

### 2.1 Texto

Texto is a rudimentary text steganography program to facilitate the exchange of binary data. It is using a simple substitution cipher which transforms *uuencoded* or *pgp ascii-armoured ascii* data, especially encrypted data into English sentences so that the text will look apparently reasonable during data transmission. FIGURE 2 shows the flow of Texto program.

Each symbol is replaced by nouns, verbs, adjectives, and adverbs in the preset sentence structures without punctuation or "connecting" words through English sentences. However, not all of the words in the resulting English are significant to the Texto program. Usually, the output of Texto is close enough to normal English text that it will slip by any kind of automated scanning.

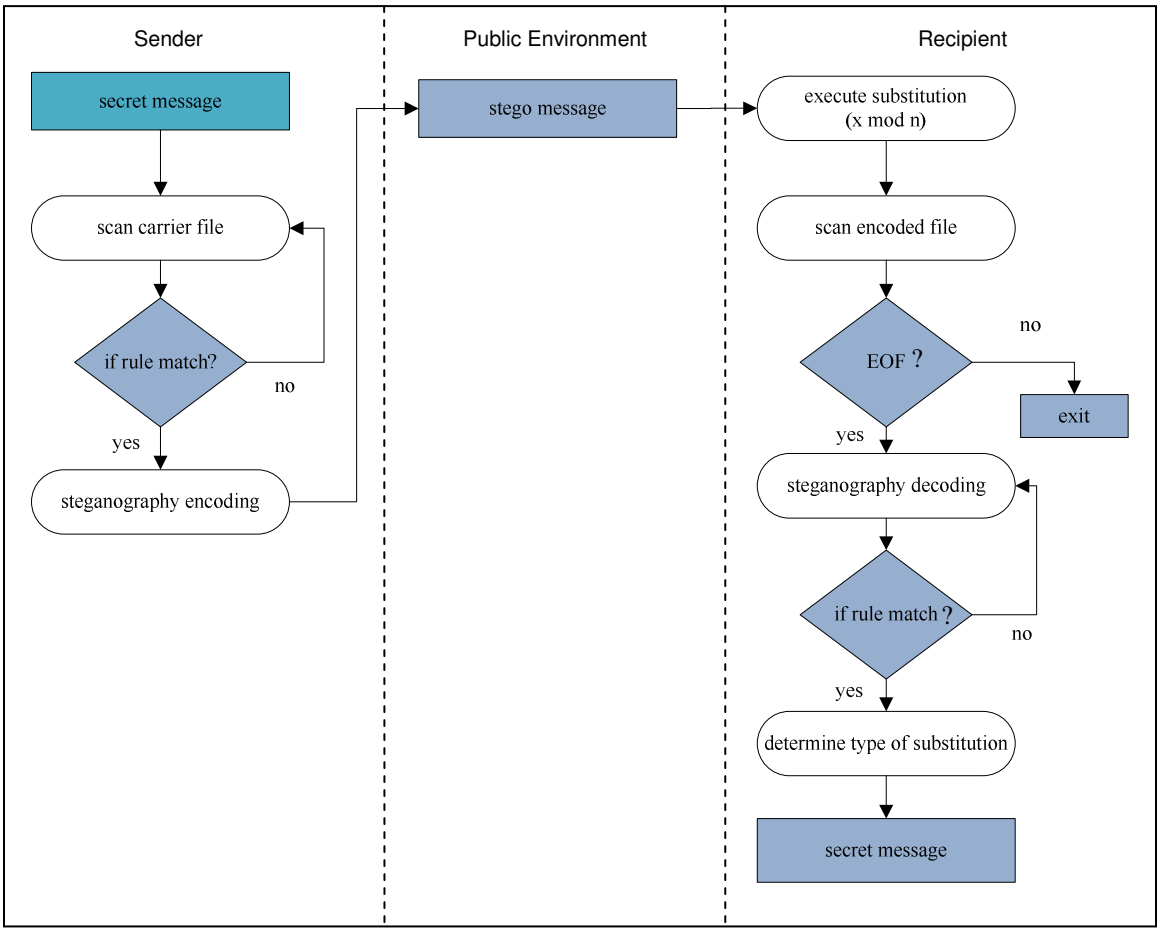



**FIGURE 2:** The Process Flow of Texto Steganography Tool.

### 2.2 Steganosaurus (Stego)

Steganosaurus also known as Stego uses a line-shift coding method which is a plain text steganography utility which encodes a binary file into a gibberish text based on either a spelling dictionary or words taken from a text document. The output of Stego converts any binary file into nonsense text based on a dictionary from a source document. The output of stego is nonsense but statistically resembles text in the language of the dictionary supplied. A human reader will instantly recognize it as gibberish while to eavesdroppers; the encrypted messages may consider it to be unremarkable, especially if a relatively small amount of such text appears within a large document. Stego makes no attempt, on its own, to prevent the message from being read. It is the equivalents of a code book with unique words as large as the dictionary.

Based on FIGURE 3, text created by stego uses only characters in the source dictionary or document. It means that during encoding process, the message will be converted into an output text file using the specified (or default) dictionary. The specified file called '*dictfile'* is used as the dictionary to encode the file. The dictionary is assumed to be a text file with one word per line, containing no extraneous white space, duplicate words, or punctuation within the words. All duplicate words and words containing punctuation characters are deleted, and each word appears by itself on a separate line. The Stego text will look less obviously gibberish if the output is based upon template sentence structures filled in by dictionaries. The efficiency of encoding a file as words depends upon the size of the dictionary used and the average length of the words in

the dictionary. Preprocessing a text file into dictionary format allows it to be loaded much faster in subsequent runs of stego. Another file namely '*textdict'*, is created to build the dictionary for input file used during encoding or decoding process. The '*textdict'* is scanned and words, consisting of alphanumeric characters, are extracted. Duplicate words are automatically discarded to prevent errors in encoding and decoding processes.

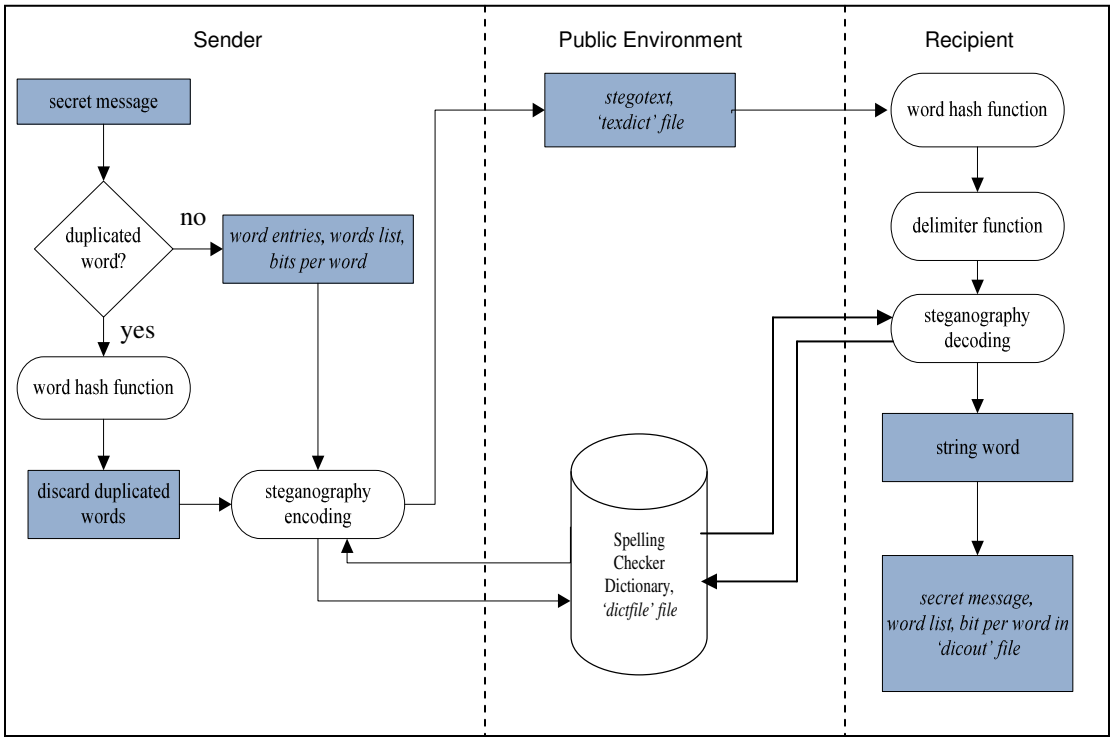



**FIGURE 3:** The Process Flow of Stego Steganography Tool.

Stego can then be applied to the encrypted output, transforming it into seemingly innocuous text for transmission, so it can be sent by sender through media, such as electronic mail, which cannot transmit binary information. If the medium used to transmit the output of stego, *'textdict'* cannot correctly deliver such data; the recipient will be unable to reconstruct the original message. To avoid this problem, the sender can either encode the data before transmission or use a dictionary which contains only characters which can be transmitted without loss. The decoding process by receiver to recover the original message, *'dictout'*, must be carried out using the same dictionary as encoding process because the ability to recognize gibberish in text is highly language dependent. Usually, the default dictionary is the system spelling checker dictionary. However, this dictionary is not standard across all systems.

### 2.3 SNOW

**S**teganographic **N**ature **O**f **W**hitespace or SNOW, is a program for conceling messages and extracting messages in ASCII text file. This feature coding method conceals messages by appending tabs and spaces (known as whitespace) at the end of lines. Tabs and spaces are invisible to most text viewers, hence the steganographic nature of this encoding scheme.

This allows messages to be hidden in the ASCII text without affecting the text visual presentation. Since trailing spaces and tabs occasionally occur naturally, their existence should not be deemed sufficient to immediately alert an observer who stumbles across them.

The data is concealed in the text file by appending sequences of up to 7 spaces, interspersed with tabs. This usually allows 3 bits to be stored in every 8 columns. The SNOW program runs in two modes which are message concealment and message extraction as shown in FIGURE 4.

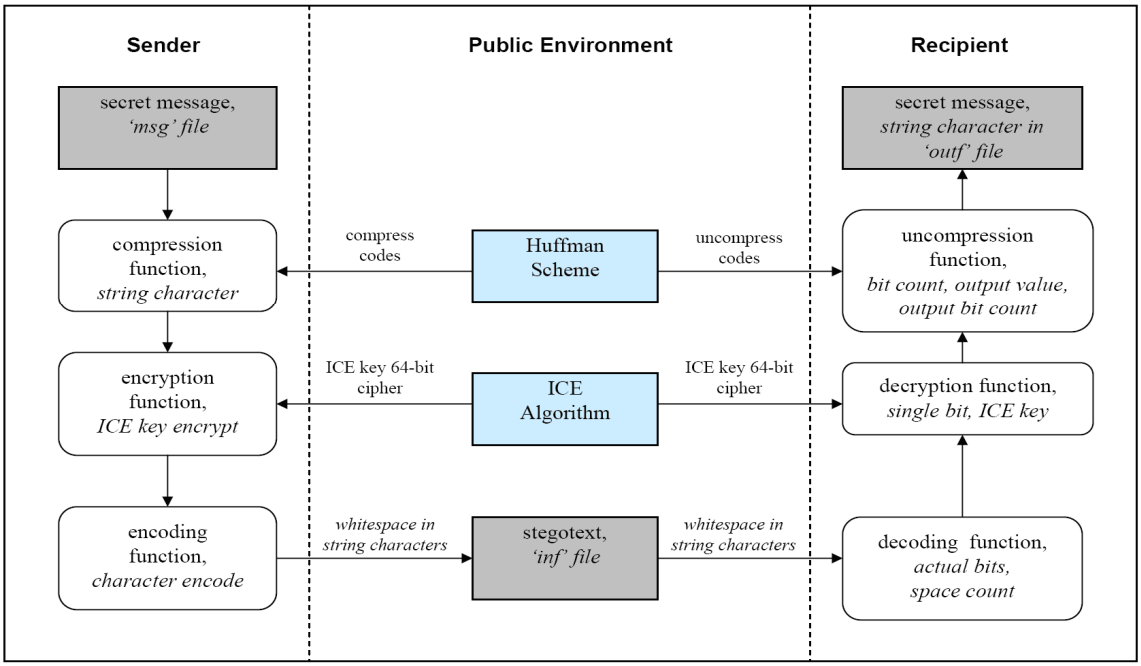



**FIGURE 4:** The Process Flow of SNOW Steganography Tool.

There are three important steps involve in the concealing process which are;

*i.* *Compression* - used a rudimentary Huffman encoding scheme where the tables are optimized for English text. This was chosen because the *whitespace* encoding scheme provides very limited storage space in some situations, and a compression algorithm with low overhead was needed.

*ii.* *Encryption* - used an *ICE* encryption algorithm [17] with 64-bit block cipher. It runs on a 1-bit *cipher-feedback* (CFB) mode, which is quite inefficient (requiring a full 64-bit encryption for each bit of output).

*iii.* *Encoding scheme* – at the beginning of a message, a tab is added immediately after the text on the first line where it will fit. Tabs are used to separate the blocks of spaces. A tab is not appended to the end of a line unless the last 3 bits coded to zero spaces, in which case it is needed to show some bits are actually there.

While in extracting process, there are also three steps involved which are decoding, decryption and decompression. All of these steps are running on sequential during extraction process. After extraction process is completed, an extracted message is transferred to output text called *outf*.

### 2.4 Stegparty

Stegparty is a hiding information system that hides data inside a text file by using a set of rules defining various flexible components within the English language. Stegparty can hide small alterations to the message by matching the text and replacing it with small typos, grammatical errors, or equivalent expressions such as spelling and punctuation changes as shown in FIGURE 5. It is a unique data hiding method by creating misspellings inside original text files.

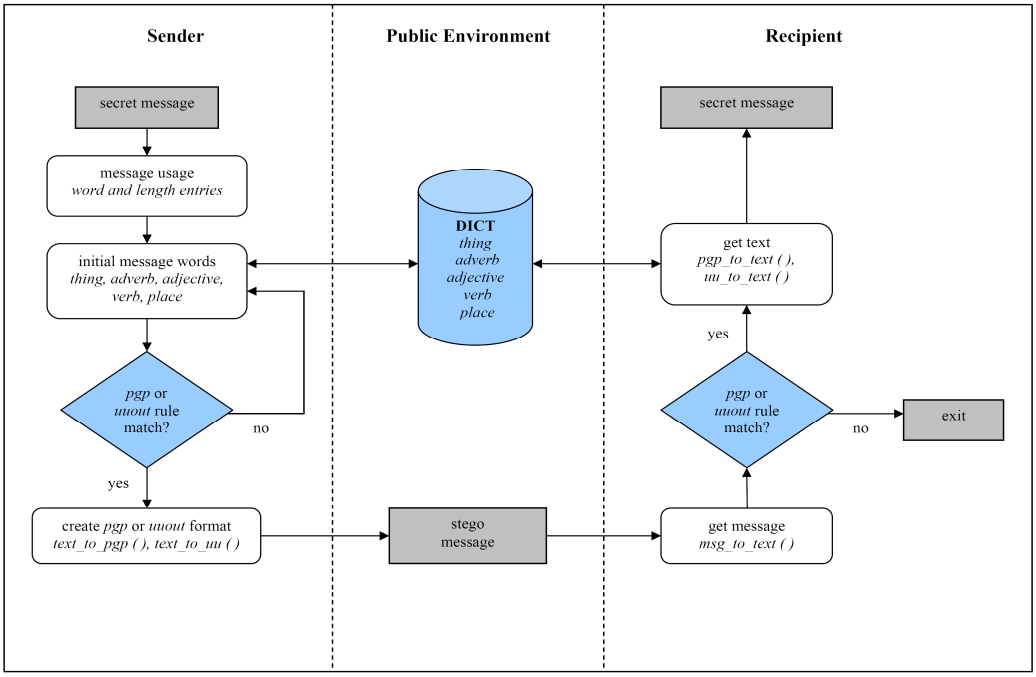



**FIGURE 5:** The Process Flow of Stegparty Steganography Tool.

## 3. EXPERIMENTAL WORK

This section discusses the parameter measurement of the selected steganographic tools during the hiding of a message. This study use software metric, such as dataset selection, execution time, and algorithm segment in order to analyse the performance of steganographic tools.

### 3.1 Software Metrics

In this analysis, three attributes of the software metric parameter have been used which are *Percent Lines with Comments (PLwC)*, *Average Statements per Function (ASpF)*, and *Average Block Depth (ABD)* [23-25].

a) *Percent Lines with Comments*

Percent Lines with Comments (*PLwC*) is used to determine the documentation level of selected tools. This analysis is very important for developers to understand the inner workings of each tool.

$$PLwC = \left( \frac{Total\ Number of\ Lines}{Total\ Number of\ Comments} \right) \times 100\%$$

$$Percent\ Lines\ (\%) = \frac{\sum_{i=1}^{1<m<2000} a_i}{\sum_{i=1}^{1<n<100} b_i} \times 100\% \qquad (1)$$

where

*a* = *Total Number of Lines*
*b* = *Total Number of Comments*

b) *Average Statements per Function (ASpF)*

ASpF calculates the average number of statements per function. Thus, *ASpF* can be used to determine the complexity of the each selected tool.

$$Average\ Statements\ per\ Function = \left( \frac{Total\ Number\ of\ Statements}{Number\ of\ Functions} \right)$$

$$\bar{x} = \frac{\sum_{i=1}^{1<n<100} c}{d} \qquad (2)$$

*where*

*c = Total Number of Statement*
*d = Number of functions*

c) *Average Block Depth (ABD)*

This analysis is used to determine the average depth of the available block in each of the selected tools. Higher value of *ABD* may lead to higher usage of memory space of the tools.

$$AverageBlockDepth = \left( \frac{Total\ Number\ of\ Nested\ Block\ Depth}{Block\ Depth} \right)$$

$$\bar{x} = \frac{\sum_{i=1}^{1<n<100} e}{f} \qquad (3)$$

*where*

$\bar{x}$ *= Average Block Depth*
*e = Total Number of Nested Block*
*f = Block Depth*

### 3.2 Data Set Selection

Text chosen dataset is one of the important components in benchmarking steganographic techniques [26]. Our study used a dataset of text which includes a variety of textures and sizes. In order to evaluate the text steganographic techniques, various file sizes have been categorized in four phases during evaluation process from phase I to phase IV, respectively. Phase I started with 10 bytes, followed by phase II with 100 bytes, phase III with 1000 bytes and end up with 20 kilobytes of plaintext files size in phase IV as shown in TABLE 2. For every size category, hundreds of files are created to ascertain the relation of time taken based on the different type of file size. Then, the result of execution time of each data files in the selected text steganographic tools during evaluation process is recorded.

| Testing Phase | Size (bytes) |
|---|---|
| Phase I | 10 |
| Phase II | 100 |
| Phase III | 1000 |
| Phase IV | 20 000 |

**TABLE 2**: Distribution of File Size Category.

### 3.3 Execution Time

To obtain the execution time of the whole process, different parameters are selected from all four steganography tools. To get a secret message to be encoded, different tools will transform this message in a four different ways. Stegparty will use code segment *secretfile* to encode message and later the message is stored in *codedfile* code segment. Stego secret message stored text in *secretmessage* code and result will be stored in code segments *cf.* Texto used *msgfile* code segment to store secret message and *engfile* code message for encoded message. In addition, SNOW tools use *–f* code in encoding secret message and *crf* code segment as an encoded message. From the encoding process, the execution time is recorded. *Start time, stop time* and *time taken* to encode message are recorded for every file sizes. The result of the time taken is discussed further in Section 4.2. TABLE 3 shows the various types of codes segment of hidden message and output file on retrieving execution time in each tool.

| Tools | Encoded File | Coded File |
|---|---|---|
| **Stegparty** | *secretfile* | *codedfile* |
| **Stego** | *secretmessage* | *cf* |
| **Texto** | *msgfile* | *engfile* |
| **SNOW** | *-f* | *crf* |

**TABLE 3**: Various Types of Codes Segment in Selected Text Steganographic Tools.

In determining the efficiency of execution time, a small segments of code is created in every steganographic tools. This code segment is activated when a *run* command is executed.

```
#include <time.h>
clock_t start, end;
double elapsed;
start = clock();
... /* Do the work. */
end = clock();
elapsed = ((double) (end - start)) / CLOCKS_PER_SEC;
```

## 4. EXPERIMENTAL RESULT

This section discusses the software metric measurement taken for Percent Lines with Comments *(PLwC)*, Average Statements per Function *(ASpF)*, and Average Block Depth *(ABD)*. This section also discusses execution time of the steganographic tools.

### 4.1 Software Metric Measurement

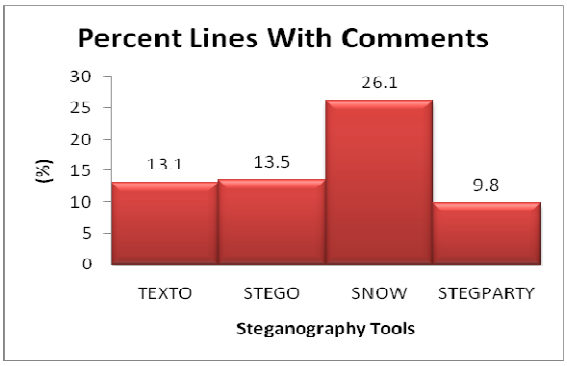



**FIGURE 6:** Percent Lines with Comments of The Chosen Text Steganographic Tools.

Based on FIGURE 6, the *PLwC* of the selected tools have been identified. A *PLwC* value of Texto is 13.1%, quite close to Stego *PLWC* value which is 13.5%. Another selected tool called Stegparty has recorded a percentage of 9.8% while the SNOW recorded a 26.1% which is the highest value among all tools. This analysis result indicates that SNOW has more documentation in its source code, compared to Texto, Stego and Stegparty.

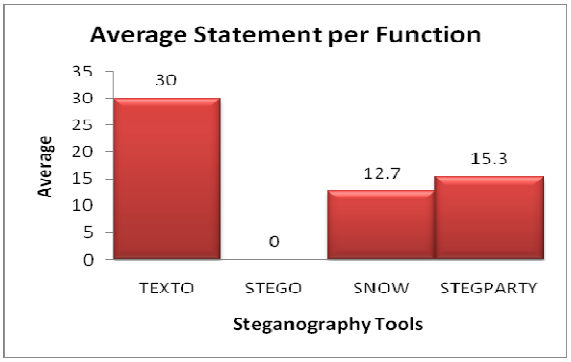



**FIGURE 7:** Average Statement per Function of The Chosen Text Steganographic Tools.

In analyzing the *ASpF* (see FIGURE 7), this study found that Texto has the highest *ASpF* value with 30 statements compared to Stego with its *ASpF* value of almost 0. While only 12.7 *ASpF* value for SNOW, Stegparty has recorded 15.3 for its *ASpF* value. Thus, SNOW and Stegparty have a comparable value for *ASpF.* The amount of code per function seems to be about the same for both tools.

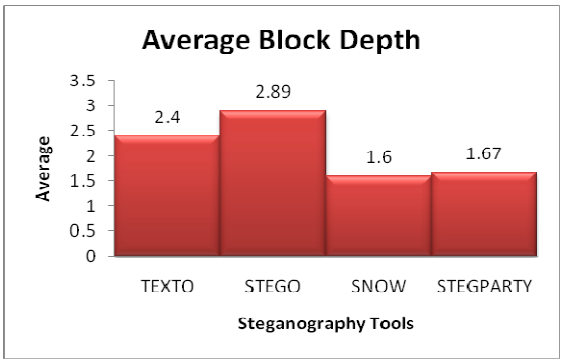



**FIGURE 8:** Average Block Depth (*ABD*) of The Chosen Text Steganographic Tools.

The value of *ABD* can be obtained by dividing the number of nested block depth with depth of block for every function. Based on FIGURE 8, the *ABD* for Texto and Stego are 2.4 and 2.89, respectively while *ABD* for SNOW is 1.6 and Stegparty is 1.67, consecutively. In term of block depth complexity, Stego and Texto are somewhat in the same league while another league comprises SNOW and Stegparty. Thus, it can be said that Texto and Stego are having about the same complexity while SNOW and Stegparty share almost the same complexity.

### 4.2 Execution Time Measurement

TABLE 4 has shown the analysis of execution time for the selected tools based on different file size, which are: 10 bytes, 100 bytes, 1000 bytes and 20 000 bytes. This study required LINUX platform and running on the Solaris 7 operating system under SunOS 5.7 version with 64-bit UltraSPARC microprocessor.

| TOOLS / BYTES | Texto | Stego | SNOW | Stegparty |
|---|---|---|---|---|
| 10 | 0.86 | 0.85 | 13.35 | 0.85 |
| 100 | 0.12 | 0.12 | 0.15 | 0.12 |
| 1000 | 0.01 | 0.01 | 0.07 | 0.01 |
| 20000 | 0.04 | 0.04 | 0.11 | 0.03 |

**TABLE 4:** Execution Time of The Chosen Text Steganographic Tools

It is found that Steparty requires 0.85 to 0.86 seconds to encode file size of 10 bytes to 1000 bytes. The running time increases significantly to 13.35 seconds when encoding file size of 20000 bytes. Stego requires much less time for encoding files where it requires 0.12 seconds to encode file size between 10 bytes to 100 bytes. Encoding file size of 20000 bytes does not change the running time very much as Stego takes only 0.15 seconds to encode the data. While Texto requires 0.01 seconds to encode file size of 10 bytes to 1000 bytes and takes an extra 0.06 seconds to encode file size of 20000 bytes. Likewise, SNOW is also in the same category as Stego and Texto. SNOW encodes file size 10 to 1000 bytes at between 0.03 and 0.04 seconds and taking only 0.11 seconds to encode file size of 20000 bytes.

## 5. DISCUSSION

This study provides the evaluation of text steganographic tools based on the criteria that has been selected. The result of the software metric and execution time have shown the level of complexity and speed performance, respectively.

In analyzing the software metric, this study has considered the Percent Lines with Comments (*PLwC*), Average Statement per Function (*ASpF*), and Average Block Depth (*ABD*) in order to measure productivity of software development. TABLE 5 shows a summarization of software metric for all four text steganographic tools.

| METRIC / TOOLS | Texto | Stego | SNOW | Stegparty |
|---|---|---|---|---|
| Percent Lines With Comments (*PLwC*) | 13.1 | 13.5 | 26.1 | 9.8 |
| Average Statement per Functions (*ASpF*) | 30 | 15.3 | 12.7 | 0 |
| Average Block Depth (*ABD*) | 2.4 | 2.89 | 1.6 | 1.7 |

**TABLE 5:** Software Metric of Text Steganographic Tools.

Result on Percent Lines with Comments (*PLwC*) indicates that SNOW has more documentation in its source code, a characteristic which is good in order to understand the flow of the tool. Compared to SNOW, Stegparty does not have much comment lines in its code. Thus, a Stegparty's developer may face some difficulties in modifying the source code due to lack of description on what each of the code does. Closer examination of Average Statement per Function (*ASpF*) for all source code reveals that Stego has small amount of statements with many functions. As such, its ratio between number of statements and number of functions seems to be quite imbalance with small number of statements and large amount of functions. On the other hand, Texto does not have many functions. Most of its statement is included in the main function. Having a non-function source code makes it difficult for future alteration if need arises. In term of modular programming [27], this study found that the coding style in Texto is weak. However, it contributes to the shortened development time because the modules can be

implemented separately, thus increasing the flexibility and comprehensibility of the program [19]. For Average Block Depth (*ABD*), Stego is more complex than the other tools since its blocks traverse deeper. There is also a probability that it uses more stack memory since traversing deeper inside the block will make the parent variables to be pushed onto stack.

In analyzing the execution time, Texto is the fastest in term of encoding secret message inside a file. It does not depend on any carrier file like Stegparty. As such, encoding file size of 10 bytes will not be too much different from encoding file size of 20000 bytes. Stego and SNOW exhibit the same behavior as Texto. These two tools are also not dependent on carrier file size. As such the time required to encode files does not differ very much. It is also discovered that Stegparty took the longest time to encode files.

## 6. CONCLUSION

In conclusion, this study found that Texto has the highest complexity level and speed performance among the text steganographic tools followed by Stegparty, SNOW and Stegano. The *PLwC* and *AspF* metrics have shown to be an important system parameter to measure productivity of software development. The envisaged future work will involve performance evaluation on the efficiency of pre-encrypt and post-encrypt process. In addition, the software productivity need to be measured more precisely by taking into account not only the four attributes but also will involve more parameters. Finally this study is only running on a single Linux platform. Taking into account more platform and different machine used to compare software performance is considered as a future work.

## 7. REFERENCES

1. M. A. A. Murad, and T. Martin. *"Similarity-based estimation for document summarization using Fuzzy sets"*. International Journal of Computer Science and Security (IJCSS), Computer Science Journal Press, Kuala Lumpur, 1(4): 1 - 12, Nov/Dec 2007

2. B. V. Dhandra, and M. Hangarge. *"Morphological reconstruction for word level script identification"*. International Journal of Computer Science and Security (IJCSS), Computer Science Journal Press, 1(1): 41 - 51, May/June 2007

3. J. J. Eggers, R. Bäuml, and B. Girod. *"A communications approach to image steganography"*. In Proceedings of SPIE: Electronic Imaging, Security and Watermarking of Multimedia Contents IV, 4675:26-37, San Jose, CA, USA, January 2002

4. J. Zollner, H. Federrath, H. Klimant, A. Pritzmann, R. Piotraschke, A. Westfeld, G. Wicke, and G. Wolf. *"Modeling the security of steganographic systems"*. In 2nd International Workshop Information Hiding. Springer, Berlin/Heidelberg, German, vol. 1525: 345 - 255, 1998

5. C. Kant, R. Nath, and S. Chaudhary. "*Biometrics security using steganography"*. International Journal of Security (IJS), Computer Science Journal Press, 2(1): 1 - 5, Jan/Feb 2008

6. N. F. Johnson, S. Jajodia. *"Exploring steganography: seeing the unseen"*. IEEE Computer Magazine, 31(2):26 – 34, 1998

7. M. Chapman, G. I. Davida, and M. Rennhard. *"A practical and effective approach to large-scale automated linguistic steganography"*. In Proceedings of the Information Security Conference (ISC '01), Malaga, Spain, 156 -165, October 2001

8. F. A. P. Petitcolas, R. J. Anderson and M. G. Kuhn. *"Information hiding: A survey"*. In Proceedings of the IEEE on Protection of Multimedia Content, 87(7):1062 - 1078, July 1999

9. Z. Oplatkova, J. Holoska, I. Zelinka, and R. Senkerik. *"Steganography Detection by Means of Neural Networks".* 19th International Conference on Database and Expert Systems Application (DEXA '08), 2008

10. K. Ochiawai, H. Iwasaki, J. Naganuma, M. Endo, and T. Ogura. "*High Speed Software-based Platform for Embedded Softwar of A Single-Chip MPEG-2 Video Encoder LSI with HDTV Scalibility*". In Proceeding of the Conference on Design, Automation and Test in Europe: 1-6, 1999

11. D. Welzel, H. L. Hausen. *"A five steps method for metric-based software evaluation: effective software metrication with respect to quality standard"*. Journal of ACM, 39(2 –5):273 – 276, 1993

12. A. Westfield, A. Pfitzmann. *"Attacks on steganographic systems"*. In Proceedings of 3rd International Workshop Computer Science (IH '99) Germany, 1999

13. S. R. Baragada, M. S. Rao, S. Purushothaman, and S. Ramakrishna. *"Implementation of radial basis function neural network for image steganalysis".* International Journal of Computer Science and Security (IJCSS), Computer Science Journal Press, Kuala Lumpur, 1(4): 12 - 22, Jan/Feb 2008

14. S. Nystedt, C. Sandros. *"Software Complexity and Project Performance*". Master Thesis and Bachelor Thesis, University of Gothenburg, 1999

15. K. Maher. *"Texto"*. Underware Software Production Ltd. Inc.,1995, http://linkbeat.com/files/

16. J. Walker. *"Steganosaurus"*. 1997, http://www.fourmilab.ch/ or http://www.fourmilab.to/stego/

17. M. Kwan. *"SNOW"*. Darkside Technologies Pty Ltd ACN 082 444 246 Australia, 1998 http://www.darkside.com.au/snow/index.html

18. S. E. Hugg. *"StegParty"*. Hamco Software (COMETBUSTERS-DOM) 1249 Turkey Point Rd Edgewater, MD 21037 US, 1999 http://www.cometbusters.com/hugg/projects/stegparty.html

19. *Steganos* can be accessed at; http://zerblatt.forex.ee/~ftp/crypto/code/STEGANOS.ZIP

20. *Snowdrop* can be accessed at; http://linux.softpedia.com/get/Programming/Version-Control/snowdrop-23917.shtml

21. *PGPn123* can be accessed at; ftp://ftp.dei.uc.pt/pub/pgp/pc/windows/

22. *FFEncode* can be accessed at; http://www.rugeley.demon.co.uk/security/ffencode.zip

23. *METRIC DATA PROGRAM* can be accessed at http://mdp.ivv.nasa.gov/loc_metrics.html#PERCENT_COMMENTS

24. C. Johns. *Applied Software Measurement (3rd Edition). USA, 2008*

25. *Metrics Complexity* can be accessed at http://download.instantiatons.com/CodeProDoc

26. M. Kharrazi, H. T. Sencar, Memon. *"Benchmarking steganographic and steganalysis techniques"* Security, Steganography, and Watermarking of Multimedia Contents, 2005

27. R. P. Cook. *"An introduction to modular programming"*, 1995 from http://www.docdubya.com/belvedere/cpp/modular.html

# Deriving Value in Digital Media Networks

**Miguel Morales-Arroyo** mangel@ntu.edu.sg
**Ravi S. Sharma** asrsharma@ntu.edu.sg
*Institute for Media Innovation/SIGIDE/SCI*
*Nanyang Technological University*
*Singapore, 637718, Singapore*

**ABSTRACT**

This paper presents a framework for the analyzing revenue distribution in the delivery of digital content such as music, movies, games, books, and news. In such content delivery networks, there are various roles played by producers, consumers, syndicators, aggregators and distributors in the marketplace. We outline a framework for business modeling known as VISOR and adapt some ideas from game theory in order to investigate notions of efficiency and fairness in the digital media eco-system. This framework suggests that the revenue distribution within a business model is determined by the range between producers' cost of production and consumers' willingness to pay. The allocation of these revenues among the players is in turn determined by their value capacity which is a function of the interface for content, service platform, organizing model and revenue streams. The game-theoretic notions of fairness and efficiency are introduced as a strategy for stability in the workplace. The paper concludes that stability is the key to derivative value in a digital media network.

**Keywords:** Digital economics, New Media Analysis, Game Theoretic Modeling.

## 1. INTRODUCTION

The Interactive Digital Media (IDM) marketplace is not entirely new. Even before the current fad of Massively Multiplayer and Online Role Playing Games (MMORPG), online music stores, Internet Protocol TV (IPTV) and a host of mobile applications, the digitization of content took off with the convergence of the world-wide web with modern telecommunications networks and devices (cf. [1], [2], [3], [4], [5], and [6]). Today's broadband Internet is at the centre of how much of this content is produced, consumed, repackaged and traded [7]. Whereas networks, content or services, and regulatory regimes have made progress through media, network and industry convergence, business models are only beginning to re-engineer themselves to the current realities of (dis-) intermediation [8], [9]. Much of this is due to the legacy cost- plus pricing of Telco's, licensing of broadcasters and subscription based revenue streams of the media industry. Content owners, on the other hand, are understandably concerned with digital rights management (DRM) and how business models and pricing strategies might cannibalize current revenues [10]. Advertising revenue streams are lucrative but work differently in the new media sectors and there is not yet an accepted split between vendors who own the content and those that own the customers [7].

Typically, the IDM marketplace comprises of 3 groups of intermediaries who come between the producers and consumers of digital media products and services [11].

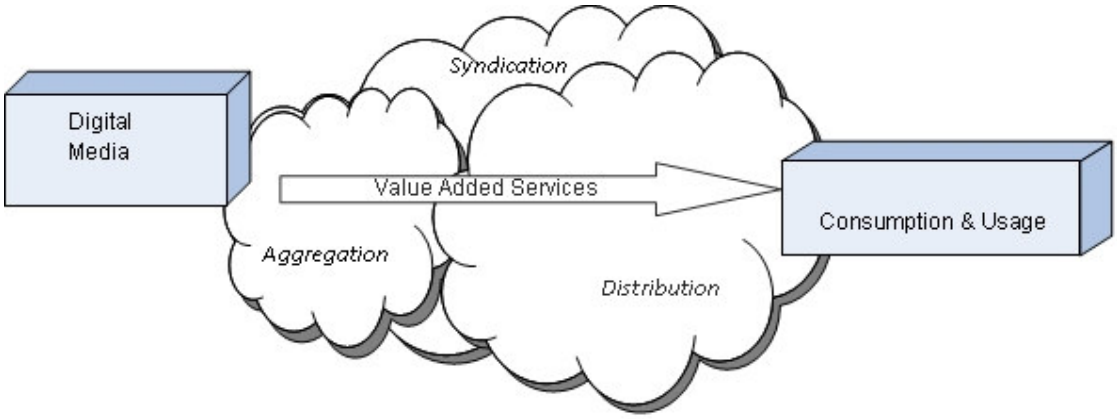



**FIGURE 1:** The Digital Media Business Eco-System.

Figure 1 depicts a simple flow chart of the IDM value cycle. Digital media is produced (or packaged) by a number of sources: online games developers, movie and animation studios, music producers, publishers of books and magazines, and a host of digital paraphernalia such as ringtones, screensavers and images. These are typically passed on to consumers in 3 stages. Aggregation is the collection of content from a variety of sources. Often content is repackaged or archived to facilitate catalog search and browsing [12]. Syndication is the proactive streaming of such content (especially after a window launch period) to alternate and repeat consumer segments [13]. Distribution is the conveying of digital content to the devices of consumers, including the billing and collection aspects [14], [15]. The roles of syndicators, aggregators and distributors are complex and often overlap. In some scenarios, some or all of these roles may even be redundant.

There are hence many such intermediary issues among the players of the IDM marketplace that remain unresolved. In this paper, we use a framework for analyzing the Digital Media business. We address the fundamental role of a business model in driving strategy and its accompanying features of alliances, pricing, and revenue sharing. One fundamental challenge would be the revenue sharing formula among the various participants of the IDM eco-system – content producers, aggregators, syndicators, distributors, other intermediaries (such as advertisers and payment brokers) and consumers. More specifically, we review some of the complex research issues that confront the continued development of the digital marketplace using the VISOR framework introduced by [16]. Some of the fundamental questions to be explored would include identifying the value brought to the market by the intermediaries of the IDM marketplace in fulfilling the syndication, aggregation and distribution roles. We will also introduce the notion of efficiency and fairness in an investigation that attempts to determine a strategic arrangement that may be seen to be stable and optimal within the IDM eco-system.

## 2. PRELIMINARIES

In this section, we briefly review the literature on media economics with respect to how business models for digital media may be formulated and roles of intermediaries may be analyzed. Strategy is an appropriation of value from the marketplace [17]. Hence a business model serves to implement strategy [18], [19]. Picard [6] has overviewed business models for online content services and how they have changed during the past two decades as technology changes and audience demand have affected operations. His work explored how the current business models emerged, how new developments are affecting those models. He also examines the implications of the changes to producers of multimedia and other content producers.
Business models are hence important in understanding the context and strategies of the major online content service providers, and how producers of content are and hope to be able to co-ordinate or integrate their operations to gain economic rents from the strengths and opportunities

provided by broadband operators [18]. Such players are necessary for the development of independent producers of digital content because they can help to provide access to the distribution systems and entry points that are necessary for commercially viable operations.

One important aspect of business models is pricing strategy. While conventional wisdom has it that value pricing is good and competitive pricing is detrimental. Several studies (cf. [5], [20], [21]) suggest that in cyber space, the potential revenue growth generated by syndicated content is far greater than mass produced and physically distributed content. In part, this could be due to the referral and distribution aspects of social networks which implicitly introduce notions of relevance, trust and branding [22]. Notwithstanding this social phenomenon, novel business models and revenue sharing arrangements are fast emerging in the space, popularly known as Triple Play (meaning voice, Internet and video) among operators and intermediaries such as content owners, portals, advertising firms and the advertisers (merchants) themselves [23].

Varian [24] suggests that information goods such as books, journals, computer software, music and videos may be copied, shared, resold, or rented in order to provide revenues. When such opportunities for sharing are present, the content producer will generally sell a smaller quantity at a higher price which may increase or decrease profits. Three circumstances where profits increase may be identified: (i) when the transactions cost of sharing is less than the marginal cost of production; (ii) when content is viewed only a few times and transactions costs of sharing are low; and (iii) when a sharing market provides a way to segment high-value and low value users.
Swatman and her associates [10] have identified the following five possible revenue sources when businesses (either collectively or separately) target individual consumers: (1) subscription fees; (2) pay per item or view; (3) bundling and selling of related items; (4) selling marketing messages for the purpose of branding in traditional as well as new media versions; and (5) selling specific advertising messages (in the form of banner ads or text) for the purpose of influencing an immediate purchasing decision. In the “walled garden” scenario (closed to non-partners), there is hence an explicit requirement for the participants in the eco-system to derive benefits from the arrangement (business model, pricing, protection of digital or customer assets etc.) in an efficient and fair relationship.

Pricing therefore becomes a key factor of revenue. Conventional micro-economic theory suggests that the higher the price, the lower the demand. Kim & Xu [25] state four factors that could lead to lower price sensitivity: (i) reputation (reputation of a vendor enables customers to perceive fewer risks and lower cost of a disappointing purchase when buying from the vendor); (ii) familiarity (focus on transaction-related understanding about the use of the vendor’s Web site, and of the whole procedure of transaction with the vendor based on previous experience); (iii) switching cost (procedural switching costs (setup costs and learning costs), financial switching costs (monetary loss costs), and relational switching costs (psychological or emotional discomfort); and (iv) pleasure (customer’s emotional response or feelings to previous transactions with an online vendor).

Technology creates a paradox in the IDM marketplace. Prices for e-services go down with time and new revenue streams are needed. Moreover, Brynjolfsson [26], [27] concluded that while there is lower friction in many dimensions of Internet competition, branding, awareness, and trust remain important sources of heterogeneity among Internet retailers. So, in the IDM marketplace, price is set as a function of value more than as a mark-up on costs.

Related to the issue of pricing, Clemons [28] suggest two theories of competition: (i) resource based value-retention and (ii) newly vulnerable markets. A market is called vulnerable when it is: (a) newly easy to enter, (b) attractive to attack, and (c) difficult to defend. With the low entry-barrier afforded by the Internet, hence the position of key players is clearly vulnerable in the IDM market place.

To sum up, the literature suggests that the IDM marketplace is indeed complex and competitive. It is very much dependent on branding, pricing, partnerships and market positioning. Suffice to

say that the current practices of the music, movie and games industries do not as yet factor the disruption brought on by the Internet and hence much work remain to be done in business strategy and modeling.

## 3. Framework

In this section we present a framework for analyzing the stability of a business model in the IDM marketplace. More specifically, we seek a framework for identifying the added value of various intermediaries and their respective derived payoffs. We posit that there is both a transitive as well as symmetric relationship between intermediaries that is necessary and sufficient to bring stability to the marketplace. We also surmise that this stable state is reached when the value is directly proportional to the payoff derived by each and every intermediary as well as producers and consumers.

A business model is defined as a group of basic interdependent systems that give life and nourish a competitive organisation, which addresses costumer needs, discriminates its offerings, and defines the tasks it should perform [19]. A business model is not a static representation. It captures the essential elements, dynamic relationships, and the basic logic strategic alternatives for the decision making process. It's major interest is the creation of value and appropriation of profit for the value created in a concrete reality and its environment [17]. The profit obtained and the core logic of their strategic alternatives provide businesses sustainability over time. Differentiation for completion based on these categories creates a competitive position [18].

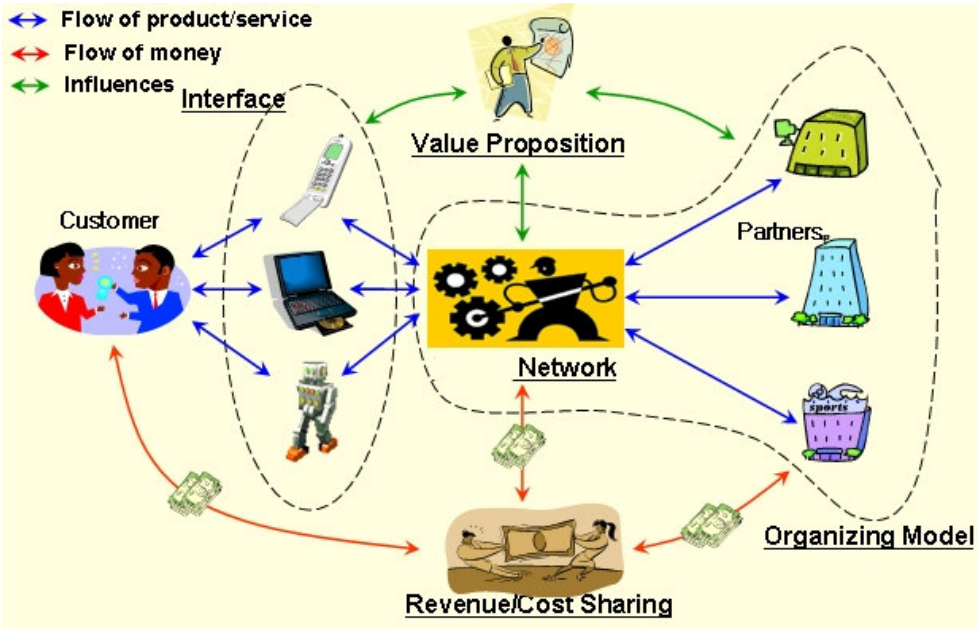



**FIGURE 2:** The Visor Model in the Digital Media Marketplace (Source: [16]).

The VISOR framework [16], first described by Professor Omar El Sawy and his co-workers at the Marshall School of Business (University of Southern California), is a business model framework developed to articulate how companies in the IDM space may react, evaluate and capitalize on the emergence of new technology or service offering. It consists of five variables that need to be considered by players: **V**alue Proposition, **I**nterface, **S**ervice Platform, **O**rganizing Model and **R**evenue/Cost Sharing. This is shown in Figure 2. More specifically, these parameters are defined as follows:

*Value Proposition* – The model should define the value that a player provides that will make customers willing to pay a premium price for them.
*Interface* – How the product or service offered by a player is going to be delivered to the customers? An easy to use, simple, convenient, user interface is required for a successful delivery of a product or service.

*Service Platforms* – The IT platform used to offer the service need to be clearly defined. Business processes and relationships needed to deliver the products and services must be supported by the IT platform.

*Organizing Model* – The distribution of product or service can be done in many ways involving many different parties. The relationships among the parties involved in the business processes and the nature of partnerships required to deliver the product or service must be clearly understood.

*Revenue/Cost Sharing* – In a good business model, the combination of the value proposition, the way that offerings are delivered, and the investments in IT platforms are such that revenues exceed costs and become attractive for all partners.

Hence the VISOR Framework posits that the value that is added to the marketplace is a function of the Interface to the Customer, Service Platform, Organising Model and Revenue Streams. Value creation is generated within a business network, which includes suppliers, consumers, substitutors, complementors, and the firm. The manner in which a firm decides to participate within its value network is a central ingredient of its business model [19].

Using the VISOR framework, we may analyze value creation within a marketplace described previously, which includes suppliers, consumers, substitutors, complementors, and the firm. These are synonymous with S, A, D. Value net analysis proposed by Brandenburger [29], [30] explores interdependencies in such a business network.

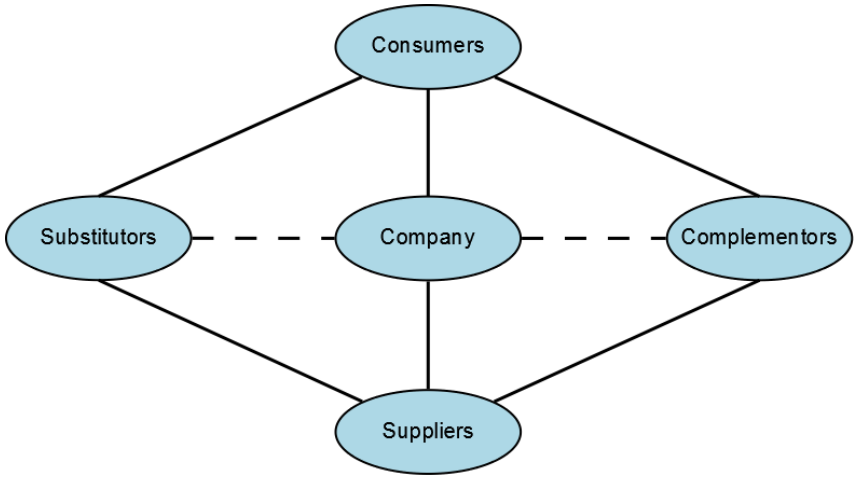



**FIGURE 3:** Game-Theoretic Value Net Analysis (Source: [29]).

The different players in the IDM marketplace and their interactions can be described using Value Net – a schematic map representing all the players in a marketplace and the interdependencies among them [29] as shown in Figure 3. Using the VISOR framework to define each player's unique value to the market, a fair and stable revenue share may be determined.

Producers, syndicators, aggregators, distributors, and even consumers (for referrals and pro-sumer content) may play different roles, but are united in seeking fair return to their contribution in stable market relationships. Hence, by using a Value Net analysis, we may examine the value

brought by each player in the chain. Since the emergence of internet and the fast growth of technology, the process of value creation and value adding has increased in complexity as well as velocity [29]. Normann [31] argued that one of the keys of creating value is to co-produce offerings that mobilize consumers to create value for themselves. For example, Amazon.com uses its database of what consumers had bought to recommend what other products may be of interest to consumers and it also provides space for consumers to give review of products for potential consumers' consideration. Here, we see how Amazon co-creates values with consumers, who then also play the role of complementors.

Such a framework, we posit, is useful to understand the interactions among the different players in the market. But, in order to understand the extent of efficiency and fairness in the interaction among the different players, we need to define the criteria for efficiency and fairness. While it is clear that pricing and revenue sharing functions are key parameters in ascertaining the value relationship in the IDM market. There is no analytical framework that allows us to articulate how values accrue to players in the network. In order to do this, we introduce two simple notions from Game Theory in the following section.

## 4. Game Theory

The IDM marketplace is structured such that numerous players take on different roles (syndication, aggregation, distribution, production, consumption) at the same time. It may make no sense to limit the quantity of information goods as the Marginal Cost of Production is zero and Distribution Cost close to zero [32]. Consequently, the quantity of digital content that may be downloaded is potentially infinite. However, the syndication (placement), aggregation (marketing) and distribution (provisioned service platform) efforts – and hence the number of channels to consumers – are not infinite [33]. And a key challenge is how to allocate resources (such as revenues, profits, customer recognition and control) fairly and efficiently. From the input-output criteria, we can derive the notion of added value which is proposed by Brandenburger [30] and Brynjolfsson [34].

It has to be noted that the changing of elements in a game are meant to create an advantage over other players in the game. In other words, when a player changes an element of a game, it should increase its bargaining power in the game. This also means that the significance of a player in a game can be measured by the value that player adds to the game. It is not necessarily the value that player brings to the market (cf. [35]). A player should not receive more than the value it contributes to the game. Brandenburger [30] formalized the method to measure added values of a player as follows:

| | | | | |
|---|---|---|---|---|
| Added value of a player | := | value created by all players | – | value created by all *other* players |

Brandenburger and his associates (op. cit.) have suggested a mechanism that may be used for the strategic analysis of the current marketplace for music, movies and games could proceed from an examination of the relationship that a player (along the SAD chain) has with suppliers, customers, substitutors and complementors. We conjecture that if the relationship is transitive (e.g. a content provider adds value to an aggregator who in turn adds value to a distributor), it will be stable. This stability would also hold in a symmetric relationship (i.e. win-win deal where players mutually benefit each others). This may seem counter-intuitive in a business that is known for its fickle and low-margin competition, but the point is that stability assures some semblance of profiting and contributing proportionately. Stability has to do with both fairness as well as efficiency.

### Efficiency

Adam Smith's notion of an invisible hand argued that competition is the best way to achieve economic efficiency or welfare maximization. Competition leads to a Pareto Optimal Point which

is technical way of saying it is not possible to reallocate resources to improve the well-being of at least one firm without harming at least one other firm in a given value network. The idea is that if we could change the current allocation of the market's resources so as to make at least one firm better off without making another worse off, and then the current allocation of resources cannot be efficient: we could do better by effecting a reallocation of resources (and revenues) [36].

In the IDM marketplace, the concept of market efficiency is based on the economic theory of price equilibrium determined by the interactions between supply and demand. In an efficient market, the price equilibrium reflects the availability of information on products to all players in the marketplace at the same time, which means that there is no information asymmetry among the different players in the marketplace. The emergence of broadband Internet markets has dramatically reduced the cost of making information available and accessible instantly, which will in turn lead to a more perfect market competition (i.e. market efficiency). The perfect market competition has been defined as the convergence of price to the level of sellers' marginal cost. In other words, consumers will not face price dispersion (cf. [37]).

Efficiency in the online market has been created by intensive competition, content variety, availability, personalization (the so-called long tail effect), and information technology infrastructure [34]. Given the nature of information goods, they can, in turn, be divided into segments that allow fragmented distribution. These can, in turn, be standardized including business rules, such as, usage rights that can be passed between firms along with the content [38]. The previous elements permit limitless virtual inventory and convenient access, reduced search and transaction costs, greater hit rates or finding relevant content, and the elimination of manufacturing and shelf space costs. Hence the distribution of content over digital networks have an extremely reduced marginal cost and have made possible disintermediation [28], serve disperse audiences, and satisfy the needs of multiple niche markets. The criteria of profitability include not only popular content, but also those less appealing to the mass market [34] and have affected market size and the price that consumers are willing to pay [30].

Grover [39] argued that the dramatic reduction of information asymmetry has not led to price convergence. Some of the causes which allow the existence of price dispersions are brand loyalty, popularity of products, and consumers trust. Conversely, brand loyalty, popularity of products, and consumer trust are caused by the success of players in creating and adding value which differentiate themselves from other companies with the same types of offerings.

**Fairness**

The notion of fairness is usually meant to convey to all players whether specific interactions, agreements, or situations are in line with the principle of justice. Grandori [40] defined fairness as rules or criteria used to divide shares of valuable resources which will be allocated to different actors. However, it has to be noted that since the goal of fairness rules are to ensure that everyone gets a fair treat, it may conflict with the notion of efficiency.
Grandori [40] further suggested four rules that can be used to identify what should be the fair interactions and agreements in inter-firm relationship, with each rule capable of generating different outcomes.

1. Input-output criteria: this is among the most widely applied and analyzed fairness criteria. This rule defines fair as the correspondence between the pay-off received by each actor and the contribution it gives to the achievement of the total output. In other words, a player can only take as revenues what it limits in proportion as resources. It can be formalized as:

$$\frac{(Output - Input)}{Input} \text{ (for the actor)} = \frac{(Output - Input)}{Input} \text{ (all other players).}$$

2. Outcome-based criteria: this is a rule proposed by game theorists and economists. This rule, unlike other rules, does not conflict with the principle of Pareto efficiency. It

is based on outcomes according to actors' utility rather than on the value that they bring. The most used mechanism to exercise this rule is the Nash Bargaining Solution (NBS), which basically determines fairness based on calculation of the maximum product of the actors' utility. In short, a player take what the market determines is its value.

3. Need-based forms: this rule focuses on the actors' necessities and aspirations instead of maximizing the product of the actors' utility. The notion of necessities is constructed socially; hence resource allocation based on necessities is independent of the contribution and utility of each actor. However, this rule can only be applied when there is mutual acknowledgement from the various actors on what is essential for each actor so that the relationship and ecosystem (i.e. marketplace) may be sustained. This seem analogous to the Marxian from each according to abilities, to each according to needs or buyers relationships to create stability.

4. Fairness heuristics: this is usually applied in ambiguous and complex situations where the calculations for resources allocation are too costly or complex to be done feasibly. When this happens, fairness is perceived as absolute equality among partners regardless of their contributions, necessities, or utilities. Hence, fairness engages players to an eco-system.

In a Nash Equilibrium, no firm wants to change its strategy because none can obtain a higher payoff. NBS has been extended to three main derivations: the Cournot, Bertrand, and Stackelberg Leader-Follower models (cf. [37]). Cournot presumes that each firm operates autonomously and tries to maximize its profits by setting quantities (or exposure in terms of channels to consumers) available in the market. In the Cournot equilibrium, each firm puts to the market the quantity that maximizes its profits provided its (correct) convictions about its competitor's alternatives. In Bertrand's model, firms define prices rather than quantity, and the Bertrand equilibrium is the marginal cost. In the Stackelberg Leader-Follower, the leader firm chooses its production level and then the competitors are free to set their optimal quantities.

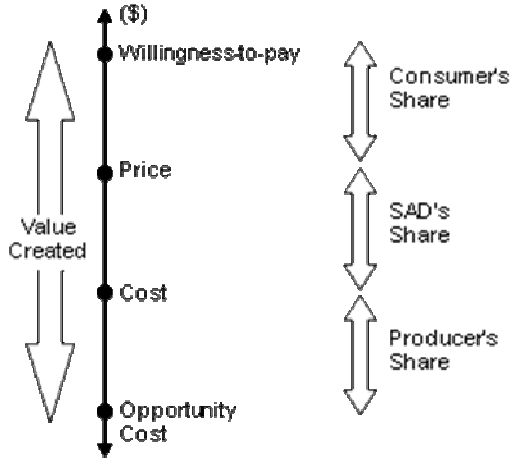



**FIGURE 4:** Division of Value in a Marketplace (Source: [30]).

The payoffs (such as revenues or brand recognition) should then be allocated based on the value brought to the marketplace by each actor [30]. Recalling some of the above rules that these actors or players in the IDM marketplace can be categorized into: content producers, consumers, and syndicators, aggregators, distributors (SAD) [9]. Syndicators, aggregators, and distributors include aggregator sites (such as Amazon, iTunes), e-malls (such as eBay, Wal-Mart) or even B2B portals. For example, in one pricing and revenue allocation scenario, the benefits that could be allocated to content consumers are the difference between the willingness to pay price and target selling price, while syndicators, aggregators, and distributors (SAD) could accrue the

difference between selling price and production cost or in some cases get commissions based on the difference. Finally, the content creators could be allocated the difference between production cost and opportunity cost. Such a scenario is summarized in Figure 4.

In the next section, we shall conclude how the concepts of value network, fairness, and efficiency may be applied to the IDM marketplace in order to investigate the interaction and dependability among the different players. We conjecture at this point that in a Nash Bargaining Solution (NBS) which promotes fairness over efficiency, the relationship among producers, consumers will be more stable. Making the perfect information assumption, a clear Value Network will emerge that convinces these players of their contributions and payoffs. Making the rationality assumption, players will therefore avoid value destroying moves and the IDM marketplace will grow for the benefit of its participants.

## 5. CONCLUSION & FUTURE WORK

The VISOR model has synthesized the basic elements required to conduct business in the IDM market: value proposition, service platform, interface interaction (customer), organizational model, and revenue/cost sharing. The traditional fee-based business models for the media industry have been disrupted [6], [14]. The business focus has shifted from the media industry to the consumer. Technology has empowered the user who increasingly demands personalized and diverse content [41]. The patterns of consumption are difficult to predict, and the techniques used in the past to predict them are useless in the new context [22]. Consumers desire to consume media content in pervasive platforms and devices - anywhere and anytime [3]. They want convenience, simplicity, a great experience, and transparency and trustworthiness in their transactions when they acquire digital media [4]. It takes little to conjecture that the business that could address the above issues will have a very attractive value proposition.

It is also well known that the convergence of service platforms and interfaces to consumers has enabled mass digital content to become ubiquitous and has further blurred the divisions among content providers, syndicators, aggregators, distributors and consumers [2], [4]. As well, it has detached content from a physical object into an omniscient commodity whereas content was traditionally associated with a specific physical object such as a song or movie linked to a CD or article to a magazine. When content is digital, its distribution has multiple channels, and both the reproduction and distribution costs are almost zero, creating a disruption in the value chain. This has created the need to better understand business model, and their value propositions.

We have attempted to understand the components of business models in the IDM marketplace and its structure. The IDM marketplace is an emerging space where it is not easy to serve audiences and customize products. In addition, this emerging marketplace is also characterized by easy duplication of content, the low costs of inventory of digital products, which will lead to multiple niche markets of products, such as the long-tail.

We have argued that in the IDM marketplace firms are no longer creating value as a part of a sequential process, but as a part of a value network [14], [31]. Firms are increasingly required to co-create value with other players in the marketplace through organizing models [42]. These relationships among different players in the marketplace may take place not only in the form of economic exchanges but also in the form of social exchanges where the relationships are based on reciprocity and trust [43], [44]. A firm typically forms partnerships with other companies to co-create value [31].

To get a more holistic picture of the significance of each actor, its relationships and its value proposition, it is important to understand the structure of the underlying business strategy. Some of the questions that are necessary to examine are: what does a firm have to offer so that consumers are willing to pay a premium price; what interface would a company have to use to deliver its products or services; and where is a firm in a value network [16], [18], [19], [45]. We

may hence conclude that the identification of value is the central step in allocating revenue and hence ensuring stability in the IDM market.

At this stage of our understanding, we conjecture that the usefulness of analyzing business models as a function of the value proposition, interface, service platform, organizing models and revenue streams. A key strategic insight is therefore how to improve efficiency, fairness, and find the equilibrium in the market place. It is also expected that the notions of fairness and efficiency are not mere theoretical constructs but also preconditions of stability for the online market ecosystem. Hence the message is clear. New media will require a level of added value like never before in the traditional markets. As such, producers, consumers, syndicators, aggregators and distributors have to be relentless in seeking out opportunities and being dynamic in their business relationships. In this paper, we have attempted to demonstrate a framework that may be used for such a purpose. In digital media networks, key value drivers that deserve attention are attractive content and its quality, consumer's appeal, convenience factors such as portability. Essentially, media providers have to identify the right content for the right consumer.

## Acknowledgements

*This work was supported by the Singapore National Research Foundation Interactive Digital Media R&D Program, under research Grant NRF.2007IDM-IDM00276 The authors of this paper also gratefully acknowledge the contribution of their international collaborators – Professors Steve Wildman, Omar El-Sawy, Francis Pereira, and Ted Tschang.*

## 6. REFERENCES

1. M. Alam, N.R. Prasad. "*Convergence Transforms Digital Home: Techno-Economic Impact*". Wireless Personal Communications, 44(1):75-93, 2008

2. S. J. Berman, S. Abraham, B. Battino, L. Shipnuck, A. Neus. *"Navigating the media divide: Innovating and enabling new business models*". Executive Brief, IBM Institute for Business Value, 2007

3. G. Eastwood. "*The Future of Convergence*". Business Insights Ltd., 2006

4. I. Gantz, D. Reinsel, C. Chute, W. Schlichting, J. McArthur, S. Minton, I. Xheneti, A. Toncheva, A. Manfrediz. "*The Expanding Digital Universe–A Forecast of Worldwide Information Growth Through 2010*". An IDC White Paper–sponsored by EMC, 2007

5. J. Meisel. "*The emergence of the internet to deliver video programming: economic and regulatory issues*". Info, 9(1):52-64, 2007

6. R. Picard. "*Changing Business Models of Online Content Services - Their Implications for Multimedia and other Content Producers*". The International Journal on Media Management, 2(2):60-68, 2000

7. D. Garcia. "*Disruptive Technologies boast Internet Advertising*". Gartner Report, 2006

8. C. Abrams. "*Key Factors Affecting Web Business Models Through 2010*". Gartner, 2007

9. R. S. Sharma, M. Morales-Arroyo, M. Tan, S. Sangwan. "*A Business Network Model for Delivering Online Content and Services on Mobile Platforms*". In The Global Mobility Roundtable Conference. Auckland, NZ, 2008

10. P. M. C. Swatman, C. Krueger, K. van der Beek. "*The changing digital content landscape*". Internet Research, 16(1):53-80, 2006

11. M. Dharmawirya, M. Morales-Arroyo, R. S. Sharma. "*Adding Value in Digital Media Networks*". In 17th AMIC Annual Conference, Changing Media, Changing Societies: Media and the Millennium Development Goals, Manila, Philippines, 2008

12. J. Hjelm. "*Why IPTV?: interactivity, technologies and services*", Telecoms explained, Chichester, U.K.: Wiley, (2008)

13. J. v. Saasten. "*Domestic and International Syndication*", in Media programming: strategies and practices, S. T. Eastman, D. A. Ferguson, Editors, Thomson, (2009)

14. G. Graham, B. Burnes, G. J. Lewis, J. Langer. "*The transformation of the music industry supply chain*". International Journal of Operations & Production Management, 24(11):1087-1103, 2004

15. S. S. Wildman. "*Interactive Channels and the Challenge of Content Budgeting*". The International Journal on Media Management, 10(3):91 - 101, 2008

16. O. El Sawy, F. Pereira, E. Fife. "*The VISOR Framework: Business Model Definition for New Marketspaces in the Networked Digital Industry*". Personal Communication, (2008)

17. R. Amit, C. Zott, "*Value Creation in E-Business*". Strategic Management Journal, 22(6/7):493-520, 2001

18. M. Morris, M. Schindehutte, J. Allen. "*The entrepreneur's business model: toward a unified perspective*". Journal of Business Research, 58(6):726-735, 2005

19. S. M. Shafer, H. J. Smith, J. C. Linder. "*The power of business models*". Business Horizons, 48(3):199-207, 2005

20. P. Barros, Kind, H. Nilssen, T. Sørgard, L.. "*Media Competition on the Internet*". Topics in Economic Analysis & Policy, 4(1):1343-1343, 2005

21. A. Jonason, G. Eliasson. "*Mobile Internet revenues: an empirical study of the I-mode portal*". Internet Research: Electronic Networking Applications and Policy, 11(4):341-348, 2001

22. H. Wang. "*New Advertising Platforms and Technologies*". Parks Associates, 2008

23. G. Eastwood. "*The Future of TV - The evolving landscape of HDTV, IPTV and mobile TV*". Business Insights Ltd., 2007

24. H. R. Varian. "*Buying, sharing and renting information goods*". Journal of Industrial Economics, 48(4):473-488, 2000

25. H. W. Kim, Y. Xu. "*Drivers of price Premium in e-markets*". Communications of the ACM, 50(11):91-95, 2007

26. E. Brynjolfsson, M. D. Smith. "*Frictionless Commerce? A Comparison of Internet and Conventional Retailers*". Management Science, 46(4):563-585, 2000

27. M. D.Smith, E. Brynjolfsson. "*Consumer Decision-making at an Internet Shopbot: Brand Still Matters*". Journal of Industrial Economics, 49(4):541-558, 2001

28. E. K. Clemons, B. Gu, K. R. Lang. "*Newly vulnerable markets in an age of pure information products: an analysis of online music and online news*". In Proceedings of the 35th Annual Hawaii International Conference on, Hi, USA, 2002

29. A. M. Brandenburger, B.J. Nalebuff, "*The Right Game: Use Game Theory to Shape Strategy*". Harvard Business Review, 73(4):57-71, 1995

30. A. M. Brandenburger, H. W. Stuart, "*Value-based Business Strategy*". Journal of Economics & Management Strategy, 5(1):5-24, 1996

31. R. Normann, R. Ramirez. "*From value chain to value constellation: designing interactive strategy*". Harvard Business Review, 71(4):65-77, 1993

32. C. Anderson. "The Long Tail", Wired Magazine. 171-177, Oct. 2004

33. S. S. Wildman. "*The new economics of value creation for media enterprises*". (forthcoming)

34. E. Brynjolfsson, Y. J. Hu, M. D. Smith. "*Consumer Surplus in the Digital Economy: Estimating the Value of Increased Product Variety at Online Booksellers*". Management Science, 49(11):1580-1596, 2003

35. R. B. Myerson. "*Game Theory – Analysis of Conflict*", Harvard University Press, (1991)

36. C. Grandy. "*Through a Glass Darkly: An Economic View of Fairness, Globalization, and States*", in Fairness, Globalization, and Public Institutions: East Asia and Beyond, J. Dator, D. Pratt, Y. Seo, Editors, University of Hawaii Press, (2006)

37. D. W. Carlton, J. M. Perloff. "*Modern Industrial Organization*", 4th ed. Pearson Addison-Wesley, (2005)

38. K. Werbach. "*Syndication: The Emerging Model for Business in the Internet Era*". Harvard Business Review, 78(3):85-93, 2000

39. V. Grover, J. Lim, R. Ayyagari, "*The Dark Side of Information and Market Efficiency in E-Markets*". Decision Sciences, 37(3):297-324, 2006

40. A. Grandori, M. Neri. "*The fairness properties of interfirm networks*", in Interfirm Networks: Organization and Industrial Competitiveness, A. Grandori, M. Neri. Editors, Routledge: London (1999)

41. Y. Benkler. "*The Wealth of Networks*", Yale University Press (2006)

42. C. K. Prahlad, M. S. Krishnan. "*The New Age of Innovation*", McGraw Hill, (2008)

43. R. Axelrod, W. Hamilton. "*The evolution of cooperation*". Science,. 211(4489):1390–1396, 1981

44. P. M. Blau. "*Exchange and Power in Social Life*", New York: John Wiley & Sons, (1964)

45. A. Osterwalder. "*The Business Model Ontology: A Proposition in a Design Science Approach*". PhD Thesis, Universite de Lausanne, 2004

# Modified One Time Pad Data Security Scheme: Random Key Generation Approach

**Sharad Patil** sd_patil057@rediffmail.com

*Research Student,*
*Bharti Vidyapeeth,*
*Pune, India*

**Manoj Devare** deore.manoj@gmail.com

*Vidya Pratishthan's*
*Institute of Information Technology,*
Baramati(MS), India

**Ajay Kumar** ajay19_61@rediffmail.com
*Jaywant institute ,*
*Pune(MS), India*

**ABSTRACT**

In this articles we show how the random key stream can be used to create lifetime supply of keys for one time pads. Here we provided the practical approach that you can use to set up your own one-time pad encryption. For simplicity let's study how randomized key can be achieved. Random key generation can simply be obtained via use of permutation. Permutation techniques can be used in conjunction with other technique includes substitution, encryption function etc. for effective performance. The goal of this article to show how the one-time pad encryption technique can be achieved by a combining of these techniques.

**Keywords :-** Cryptography, Cryptosystem, One-time pad, encryption, auto-key , Key enhancement, Digital Signature.

## 1. INTRODUCTION

Today's networks are seriously threatened by network attacks. Besides the rapid improvement of attacking technologies powered by profits, there are three reasons that cause the present serious status of network security, including internet itself having a weak basis, the current security technologies having respective drawbacks and limitations and the dilemma between security performance and according cost. By considering this three , in this article , we try to put secure one time pad scheme with random key generation approach.

One well known realization of perfect secrecy is the One-time Pad, which was first decscibed by Gillbert Vernam in 1917 for use in automatic encryption and decryption of telegraph messages. It is interesting that the One-time Pad was thought for many years to be an " unbreakable" cryptosystem, but there was no mathematical proof of this until shannon developed the concept of perfect secrecy over 30 year later.

**1.1 Cryptosystem for One time Pad :** Let n ≥1 be an interger and take Þ =e = k = (Z2)n .

For K ε (Z2)n , define eK (x) to be the vector sum modulo 2 of K and x ( or equivalently , the exclusive –or of the two associated bitstrings ) So, If x=(x1….xn) and K=(K1…..Kn) then

eK (x)= (x1 +K1….. ,xn + Kn) mod 2

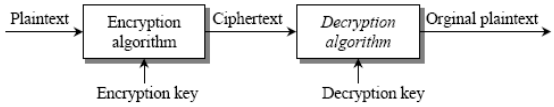
Decryption is identical to encryption.

If y=(y1….yn) , then , dK (y)= (y1 +K1….. ,yn + Kn) mod 2 [3]

Vernam patented his idea in the hope that it would have widespread commeracial use . but due to unconditionally secure cryptosystem like one time pad , the amount of key that must be communicated securely is at least as large as the amount of plaintext. The one-time pad is vulnerable to a known-plaintext attack. If the key is used once for every plaintext , it creates the severe key management

From the experimet it is easily seen that the One-time Pad provides perfect secrecy and not breakable becacuse of the two facts , encryption key is random number and the key is used once only. The system is also more attractive because of easily encryption and decryption. One time pad has been empolyed where unconditional security may be of great importance includes military and diplomatic context.
It should be clear that the one-time pad is discarded after a one time use, so this technique is highly secure and suitable for small message only and impractical for large message.

**1.2 Cryptography**




Cryptography may be used at different levels of a security model. The algorithms used, the quality of their software implementation and the key length used are the main factors determining the strength of a cryptography application.
Cryptography can reformat and transform your data, making it suffer on its trip between computers. The technology is based on the essentials of secret codes arguments by modern mathematics and protects your data in powerful ways
Secret code= Encryption – digital Signature

**1.3 Biometric security**

For many organizations, implementing the right biometric user identification technique can improve data security and lead to significant cost savings by reducing help desk calls Biometric user authentication techniques can be used to protect PCs and networks from unauthorized access by authenticating users based on a physical feature such as a fingerprint, retina, iris, hand, or face. Although voice and signature identification do not involve physical characteristics, they are usually included with biometric user authentication techniques. For some of these biometric techniques, the cost of the equipment needed to identify a user is too high for widespread use. For example, the cameras and equipment needed for iris scanning can cost thousands of dollars. However, iris scanning is one of the most accurate biometric user authentication techniques, and it is expected to be used extensively in applications such as user identification for automatic teller machines (ATMs).

## 2. PRELIMINARIES

Basically there are two type of random number generators used in cryptography, the true random number generator[TRNG] and the pseudorandom number generators[PRNG]. The aim of a TRNG is to generate individual bits, with uniform probability and without any correlation between those bits. Consequently , the knowledge of some bits does not give any information about the other generated bits. However , achieving this task in real practice appears to be a difficult possibility . Consequently the cryptographic designers and implementers often do resort to pseudorandom bit generators in many applications. [3]
A long random [ or pseudo –random] string used to encrypt a message with a simple EX-OR operation is known as a one tine pad. A stream cipher generates a one time pad and applies it to a stream of plaintext with EX-OR.

**2.1 History**

A little bit history of cryptography as a science or art , was originally developed by the Arabs. The year 1412 saw the publication of Subh-al-a “sha, a, a 14-volume encyclopedia written by shihab al-Din. The text explain transposition and substitution.
The one-time pad system itself was perfected in 1917 during the first World war. Random keys were written on paper that were glued together to form the pad. In this encryption technique the key used once hence refer as one-time pad. An army Signal Corp. officer, Joseph Mauborgne , proposed an improvement to Vernam cipher that give concrete ultimate in security. A long random[ or pseudo random] string used to encrypt a message with a simple Ex-OR operation known as a one-time-pad. The key stream for a one-time pad must be a true random stream, mean that every key byte can take any value in between 1 to 256 octets. The practical difficulty of this encryption scheme is that the key bytes cannot be used again.

**Law PAD1**: The one-time pad is a method of key transmission, not message transmission. [Blakeley] . The One-Time Pad is just a simple variation on the Beale Cipher.

**2.2 One time pad system**

The one-time pad encryption scheme is define as any method of encryption where each byte of the plaintext is encrypted using one byte of the key stream and each key byte is used once only. Hence name as One time pad.

One time pad encryption algorithm can be known from the following equation

C i = E(P i ,K i) for I= 1,2,3,……n

Where : E = the encryption parameter

P I= the ith character of the plaintext

Ki = the ith bytes of the key used for massage

C i =the ith character of the cipher text

n = length of the key stream.

Both the encryption parameter and Key stream must be kept secret .

For practical application , the key used for one time pad cipher is a string of random bits, usually generated by a Cryptographically Strong Pseudo-Random Number Generator. However for ultimate security , it is suggested to generate the key by using the natural randomness of quantum mechanical events, since quantum events are believed scientifically to be the only source of truly random information in the universe.

If the key is truly random an XOR operation based one –time pad encryption scheme is perfectly secure against cipher text-only cryptanalysis.

We come to the point that if the hackers does not know the sender or receiver key, then the one time pad encryption scheme is 100 % secure. We can only talk about OTP if four important rules are followed. If these rules are applied correctly, the one-time pad can be proven to be unbreakable (see Claude Shannon's "Communication Theory of Secrecy Systems"). However, if only one of these rules is disregarded, the cipher is no longer unbreakable.

1. The key is as long as the plaintext.
2. The key is truly random (not generated by simple computer Rnd functions or whatever!)
3. There should only be two copies of the key: one for the sender and one for the receiver (some exceptions exist for multiple receivers)
4. The keys are used only once, and both sender and receiver must destroy their key after use.

**In Short One Time pad has the characteristics**

* if a truly random key as long as the message is used, the cipher will be secure
• called a One-Time pad
• is unbreakable since cipher text bears no statistical relationship to the plaintext

• since for **any plaintext & any cipher text** there exists a key mapping one to other
• can only use the key once though
• have problem of safe distribution of key

### 2.3 Advantages and disadvantage of one time pad

Message encrypted by using One Time Pad cannot be broken because of, the fact that , encryption key is a random number and the key is used only once. However the practical difficult of using One-Time Pad is that the key bytes cannot be reused.

What we are doing to enhance the security ?

Today, more than ever, computer networks are utilized for sharing services and resources. Information traveling across a shared IP-based network, such as the Internet, could be exposed to many devious acts such as eavesdropping, forgery and manipulation. Fortunately, there are several mechanisms that can protect any information that needs to be sent over a network. This paper introduces One Time pad security mechanism and explains available security mechanisms to effectively prevent such threats from happening.

No one wants his or her confidential or classified information revealed. Confidential information that you do not want to share with others is the easiest to protect, but ever so often there is a need to share this type of information. Whenever this happens, you need to be able to send the information in a secure manner to your trusted receiver. This issue is particularly important when network communication is involved, since network communication has become the cornerstone for organizational effectiveness and today's digital communication often includes sensitive information such as control and corporate financial data. Consequently, we need security mechanisms whenever sensitive information is to be exchanged over the network. And hence it need tobe improve the security upto optimize level.

## 3. METHODOLOGY

We used simulation methodology to check the encrypted text for alphabets. Here we first create the ASCII chart for small alphabet that shown in Table-1 and then generate the random number key after that enter the plain text

which we want to sent to the recipient as most secure one, then next access the equivalent ASCII number of given plain text from Table-1 hereafter we add the first ASCII character equivalent number and the first random key number i.e. we add key to the plain text. then implement the equivalent ASCII character of addition repeat the process still end of string , then write the encrypted text. While sending the encrypted text to the recipient we must add the random number key at the end of encrypted text , key factor is that here at the begin of encrypted message we add the first number that treat as the length of plain text which can be helpful for the recipient to identify the actual string or message.

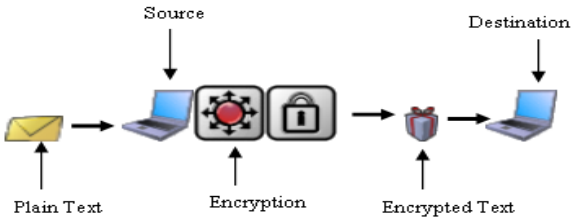



Fig.1-Encryption Process by Random Key generation (At source)


Fig.2-Decryption Process at destination

Here the practical information is used that can be used to setup one time pad encryption system. For easy adoption the steps is given 1] create the key 2] format the massage 3] Encrypt the massage 4] Decrypt the massage . The Figure 1 and Figure 2 shows the easiest map for the process. The proposed method of encryption is shown in Figure -1 . This process is done at source host. The part of decryption process will be performed at the destination end. that shown in Figure -2

## 4. ALGORITHM

* Random Key Generation ( );
Step-1 Create Ascii Chart ( ) ;
Step-2 Create random_array();
Setp-3 Create Get_Plain_Text( )
Step 4 Write _file for Plain text()
Step 5 Create encrypted Text ()
Step 6 Write file for encrypted text()
Step 7 Send the encrypted file to the recipient
Step 8 Perform decryption process
Step 9 get the Plain text at the destination
Step 10 end

Table-1 Sample Ascii chart

| Alphabet | ASCII | Alphabet | ASCII |
|---|---|---|---|
| a | 97 | n | 110 |
| b | 98 | o | 111 |
| c | 99 | p | 112 |

| | | | |
|---|---|---|---|
| d | 100 | q | 113 |
| e | 101 | r | 114 |
| f | 102 | s | 115 |
| g | 103 | t | 116 |
| h | 104 | u | 117 |
| i | 105 | v | 118 |
| j | 106 | w | 119 |
| k | 107 | x | 120 |
| l | 108 | y | 121 |
| m | 109 | z | 122 |

## 5. IMPLEMENTATION AND ANALYSIS AND RESULTS

```
#include <stdlib.h>
#include <stdio.h>
FILE *fp;
int cntr,i,var,index;
int len,other;
int random_array[10];
char plain_text[10];
char encrypt_text[20];
char decrypt_text[20];
char alphabet[26];
int ascii[26];
void get_plain_text();
void create_ascii_chart();
void create_random_array();
void write_file();
void create_encrypt_text();
void create_decrypt_text();

void main()
{
clrscr();
randomize();
alphabet[0]='a';
create_ascii_chart();
create_random_array();
get_plain_text();
write_file("p.txt",plain_text);
create_encrypt_text();
write_file("e.txt",encrypt_text);
create_decrypt_text();
write_file("d.txt",decrypt_text);
//Decide whether to ADD/substract/multiply/divide by
//available random values to the original text/table
getch();
}


void write_file(char *file_name,char *str)
{
        fp=fopen(file_name,"w");
        fputs(str,fp);
        fclose(fp);
}
```

```
void create_ascii_chart()
{
	printf("Contents of alphabet Array\n");
	for(cntr=0;cntr<26;cntr++)
	{
		alphabet[cntr]=alphabet[0]+cntr;
		ascii[cntr]=alphabet[cntr];
		printf("%c\t%d\n",alphabet[cntr],ascii[cntr]);
	}

}

void create_random_array()
{
	printf("The ten random numbers are=\n");
	for(i=0;i<=10;i++)
	{
	random_array[i]=rand()%10;
	printf("number is[%d] =%d\n",i,random_array[i]);
	}
}

void create_encrypt_text()
{
var=0;
index=0;
other=0;
len=strlen(plain_text);
for(index=0;index<len;index++)
{
	for(other=0;other<26;other++)
	{
		if(plain_text[index]==alphabet[other])
		{
		encrypt_text[index]=ascii[other]+random_array[index];
		break;
		}
	}
}
//append radom_array/KEY to the encrypted format
for(i=0;i<10;i++)
{
encrypt_text[index]=random_array[i];
index++;
}
printf("\n\nEncrypted Text=");
printf("%s",encrypt_text);
}
void get_plain_text()
{
printf("\nPlease enter plain Text.[no space, max. 10 character] .\n");
	gets(plain_text);
	printf("Your Plain text is=");
	puts(plain_text);
}
void create_decrypt_text()
{
```

```
var=0;
index=0;
other=0;
len=strlen(encrypt_text);
for(index=0;index<len;index++)
{
        for(other=0;other<26;other++)
        {
                if(encrypt_text[index]==alphabet[other])
                {
                decrypt_text[index]=ascii[other]-random_array[index];
                break;
                }
        }
}
//append radom_array/KEY to the encrypted format
for(i=0;i<10;i++)
{
decrypt_text[index]=random_array[i];
index++;
}
printf("\n\nDecrypted [Plain text] Text=");
printf("%s",decrypt_text);
}
```

In this type of encryption system ,we took only small lower case alphabets and their ASCII value , so by using ASCII code of the alphabets and random generated key , the encrypted text is more complex and it's analysis is difficult for attackers. Hence I come to conclusion that by designing different encryption method as a onetime pad is more difficult for crack. In further research ,we would like to design the algorithm on modular arithmetic base with complements concepts.

## 6. CONCLUSION & FUTURE WORK :

This algorithm has a lot of scope to enhance the security by using combining the different approaches such as binary addition, multiplication and modular arithmetic function are also common. instead of using ASCII . We have outlined a number of defense strategies, many of which demand much further research. The algorithm become more dynamic if we choose the above approaches randomly. In further research we would like to design the algorithm on modular arithmetic base with complements concepts.

## 7. REFERENCES

1. Larry l. Peterson et al. "Computer Networks –A Sysytem Approach ", Third Edition , Morgan Kaufmann Publishers ISBN:0-55860-833- 8.

2. Behrouz A. Forouzan et al., " Data Communication and Networking " Third Edition , TATA McGRAW –HILL EDITION ISBN-0-07-058408- 7.

3. Douglas R, Stinson " CRYPTOGRPHY Theory and Practice " Second Edition .

4. Charlie Kaufman st al. " Network Security " PRIVATE Communication in a PUBLIC World. , Prentice Hall of India Private Limited. 2003

5. Information Technology Journal 4(3) : 204-221, 2005

6. Claude Shannon's " Communication Theory of Secrecy Systems" .
7. Neal R. Wagner "The Laws of Cryptography: Perfect Cryptography: The One-Time Pad "
8. Ritter, Terry 1991. The Efficient Generation of Cryptographic Confusion Sequences. Cryptologia "15: 81-139.
9. www.EFYMAG.com - February-2007

10. www.zdnetindia.com

11 .www.sans.org

# Towards a Flow-based Internet Traffic Classification for Bandwidth Optimization

**Abuagla Babiker Mohd** Bmbabuagla2@siswa.utm.my

*Faculty of electrical engineering*
*University Technology Malaysia UTM*
*Johore bahru, 813100 , Malaysia*

**Dr. Sulaiman bin Mohd Nor** sulaiman@fke.utm.my
*Faculty of electrical engineering*
*University Technology Malaysia UTM*
*Johore bahru, 813100 , Malaysia*

**ABSTRACT**

The evolution of the Internet into a large complex service-based network has posed tremendous challenges for network monitoring and control in terms of how to collect the large amount of data in addition to the accurate classification of new emerging applications such as peer to peer, video streaming and online gaming. These applications consume bandwidth and affect the performance of the network especially in a limited bandwidth networks such as university campuses causing performance deterioration of mission critical applications.
Some of these new emerging applications are designed to avoid detection by using dynamic port numbers (port hopping), port masquerading (use http port 80) and sometimes encrypted payload. Traditional identification methodologies such as port-based signature-based are not efficient for today's traffic.
In this work machine learning algorithms are used for the classification of traffic to their corresponding applications. Furthermore this paper uses our own customized made training data set collected from the campus, The effect on the amount of training data set has been considered before examining, the accuracy of various classification algorithms and selecting the best.
Our findings show that random tree, IBI, IBK, random forest respectively provide the top 4 highest accuracy in classifying flow based network traffic to their corresponding application among thirty algorithms with accuracy not less than 99.33%.

**Keywords:** NetFlow, machine learning, classification, accuracy, decision tree, video streaming, peer to peer.

## 1. INTRODUCTION

Network management is a service that employs a variety of tools, applications and devices to assist human network managers in monitoring and maintaining networks.
In recent times, ISPs are facing great challenges due to the domination of certain unregulated applications which are affecting their revenue.

Classifying traffic according to the application or application class that produces it is an essential task for network design and planning or monitoring the trends of the Internet applications. Since the early 2000s, application detection has become a difficult task because some applications try to hide their traffic from traditional detection mechanism like port based or payload based [1]
In this work, we focus on selecting the best machine learning classification algorithm for identifying flow -based traffic to their originating applications. That can be done by testing the accuracy of 30 algorithms then selecting the best 15, then doing deeper checking to reduce the best set (e.g the best 4 algorithm from an accuracy point of view).
Since our goal is to classify traffic for traffic control purpose, the extended work will focus in testing the effect of time for those four best algorithms so as to build a near real-time system for traffic control.
The remainder of this article is structured according to the following topics: Related work, methodology, results and discussion, and finally, conclusion and future work.

## 2. Related Work

A lot of research work has been done in the area of network traffic classification by application types and several classifiers have been suggested. However the majority of them are based on transport layer port-numbers (currently lack of efficiency because of port hopping and port tunneling), signature-base (which fails to identify encrypted payloads), heuristics or behavioral-based (which is not efficient for real time or online classification). Recently machine learning is widely used in this field. The following subsections explore these approaches in more details.

### 2.1 Port Number Based Classification

This method classifies the application type using the official Internet Assigned Numbers Authority (IANA) [2] list. Initially it was considered to be simple and easy to implement port-based inline in real time. However, nowadays it has lower accuracies to around about between 50% to 70% [3]. Many other studies [4, 5, 6, and 7] claimed that mapping traffic to applications based on port numbers is now ineffective. Network games, peer to peer applications, multimedia streaming uses dynamically assigned ports (port hopping) for their sub transactions, so it is difficult to classify them using this method. Also, the above mentioned applications can disguise their traffic as other known classes (such as http, and ftp).

### 2.2 Payload Based Classification

In this approach packet payloads are examined to search for exact signatures of known applications. Studies show that these approaches work very well for the current Internet traffic including many of P2P traffic. So that this approach is accepted by some commercial packet shaping tools (e.g. Packeteer. Several payload-based analysis techniques have been proposed [3, 7, 8, 9].
Payload-based classification still has many disadvantages. First, these techniques only identify traffic for which signatures are available and are unable to classify any other traffic. Second, payload analysis requires substantial computationally power [1], [10] and storage capacity [11] since it analyzes the full payload. Finally, the privacy laws [10] may not allow administrators to inspect the payload and this technique will fail if payload is encrypted.
Alok Madhukar et al. [7] focus on network traffic measurement of peer to peer applications on the Internet, The paper compared three methods to classify P2P applications: port-based analysis, application-layer signature and transport layer heuristics. Their results show that classic port-based analysis is ineffective and has been so for quite some time. The proportion of "unknown" traffic increased from 10-30% in 2003 to 30-70% in 2004-2005. While application-layer signatures are accurate, it requires the examination of user-payload, which may not always be possible

### 2.3 Protocol Behavior or Heuristics Based Classification

Transport-layer heuristics offer a novel method that classifies traffic to their application types based on connection-level patterns or protocol behavior. This approach is based on observing and identifying patterns of host behavior at the transport layer. The main advantage of this method is that there is no need for packet payload access.

BLINK [12] proposed a novel approach to classify traffic in the dark. It attempts to capture the inherent behaviors of a host at three different levels, first, social level, which examines the popularity of the host. Second, the functional level which means whether the intended host provides or consumes the service. Finally, the application level that is intended to identify the application of the origin. The authors claimed that their approach classified approximately 80% - 90% of the total number of flows in each trace with 95% accuracy. However this method cannot be applied for inline near real time classification because of the classification speed limitation.

Fivos, et al [13] presented a new approach for P2P traffic identification that uses fundamental characteristics of P2P protocols such as a large diameter and the presence of many hosts acting both as servers and clients. The authors do not use any application-specific information and thus they expect to be able to identify both known and unknown P2P protocols in a simple and efficient way. Again, from the study done, it is anticipated that they will face a problem due to port tunneling.

### 2.4 Statistical Analysis Based Classification

This approach treats the problem of application classification as a statistical problem. Their discriminating criterion is based on various statistical features of the flow of packets e.g. number of packets, packet size, inter arrival time. Machine learning is used for classification. The advantage of this approach is that there is no packet payload inspection involved.

Nigel Williams et al. [14] compared five-widely used machine learning classification algorithms to classify Internet traffic. Their work is a good attempt to create discussion and inspire future research in the implementation of machine learning techniques for Internet traffic classification. The authors evaluated the classification accuracy and computational performance of C4.5, Bayes Network, Naïve Bayes and Naïve Bayes Tree algorithms using feature sets. They found that C4.5 is able to identify network flows faster than the remaining algorithms. Also they found that NBK has the slowest classification speed followed by NBTree, Bayes Net, NBD and C4.5.

Jiang, et al. [15 ] showed by experiments that NetFlow records can be usefully employed for application classification. The machine learning used in their study was able to provide identification accuracy of around 91%. The authors used data collected by the high performance monitor (full packet capturing system) where NetFlow record was generated by utilizing nProbe (a software implementation of Cisco NetFlow).

Wang1, et al. [16] discovered a new method based on the support vector machines (SVM) to solve the P2P traffic identification and application classification problem. Experimental results show that their method can achieve high accuracy if they carefully tune the parameters, and it also has promising identification speed.

## 3. Methodology

In this work, the approach used here exploits the statistical flow features of the application during the communication phase. Machine Learning (ML) is used to classify the traffic. Weka toolkit [17], MYSQL-database and other supporting programs has been used to achieve our goals. Figure 3.1 represents the flow diagram of the methodology.

This paper concentrates on an accurate classification of the bandwidth consumer applications e.g. video streaming for traffic profiling purposes. This can be used later for traffic control, capacity planning or usage based billing that can indirectly contribute on enhancing the performance of the network. The following subsections explain the flow diagram in more details.

### 3.1 Data Collection

Figure 3.2 shows the test bed setup which is used in data collection for this work. The setup is a typical faculty environment which is generating mixed Internet traffic of different applications. UDP traffic represents a small amount of data compared to TCP traffic (74.3% of the UDP flows consist only one packet per .

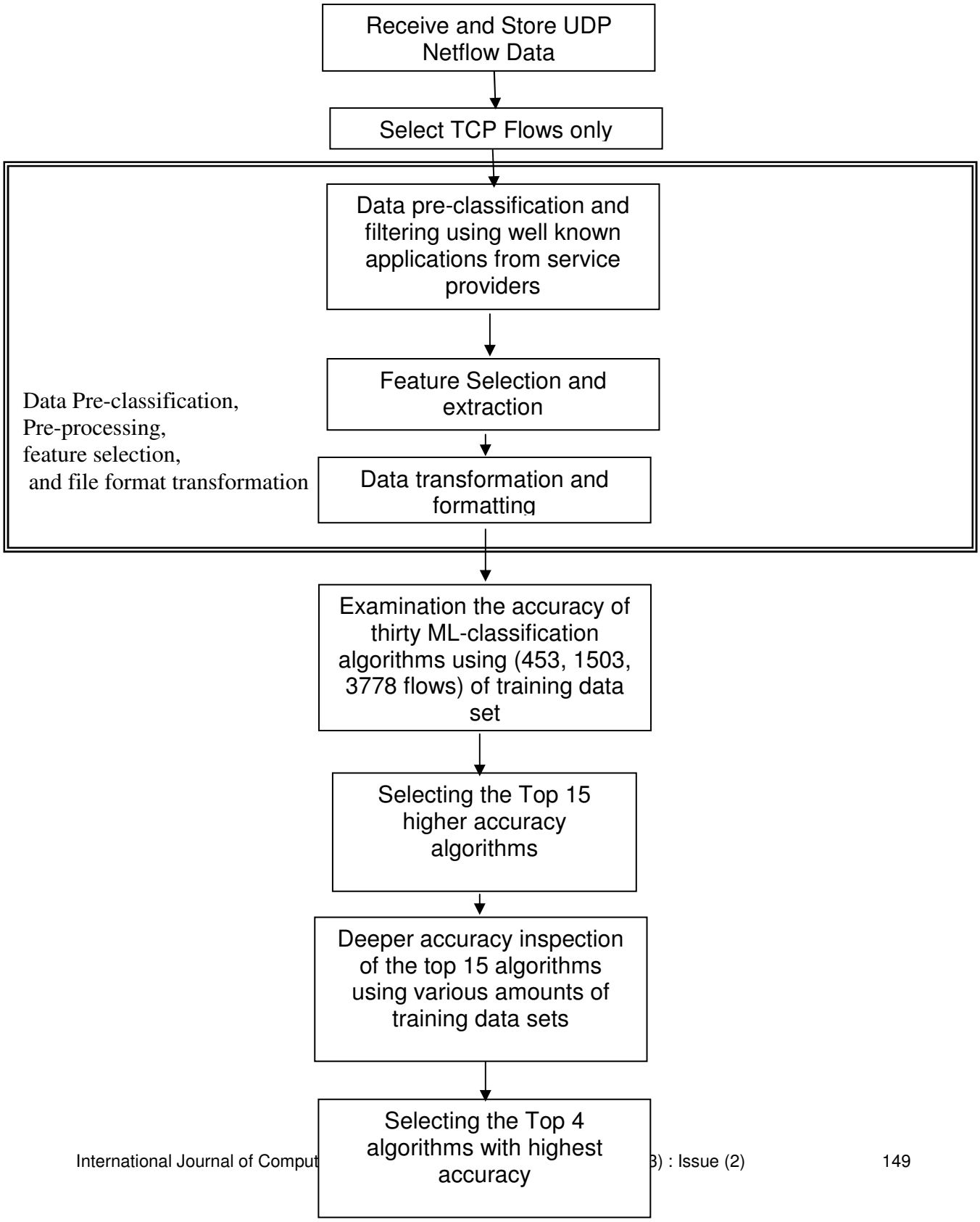


**FIGURE 3-1:** Methodology Flow Diagram

flow). For the purpose of evaluating algorithms in classifying the network traffic, only TCP flows have been taken into consideration.
Here, all users are connected to access switches which collapses to a distribution switch. These distribution switches in turn is connected to core routers centrally located. NetFlow data was collected from the distribution and exported to a NetFlow collector server. The collected NetFlow data were stored into Mysql database for further preprocessing, profiling and off-line classification.

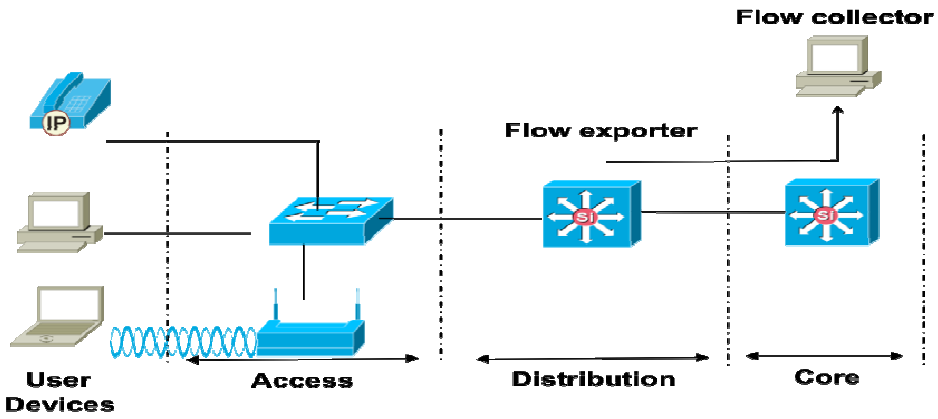



**FIGURE 3-2:** typical setup in a faculty with NetFlow exporter and collector

### 3.2 Data Pre-classification and Filtering

Using well known providers that always provide well known services, such as google.com for http, gmail and yahoo for mail service and youtube.com for video streaming, traffic pre-classification has been implemented. The pre-classified training data has been converted from table format to text file so as to train and tests Weka's classification algorithms.

### 3.3 Data Preparations, Preprocessing, and Accuracy Testing

From the NetFlow records, useful statistical features such as number of bytes, numbers of packets, start time and last time have been extracted. Derived features have also been produced which includes duration, mean packet size, mean inter arrival time and class.
The NetFlow data were prepared and processed in an acceptable format for further file conversion process to be compatible with Weka toolkit [18].
These pre-classified data has been used to train different classification algorithms, initially thirty machine learning classification algorithm with different amount of datasets {453, 1503, 3778 flows}. From them, the 15 best accuracy algorithms have been taken for deeper accuracy examinations using various amounts of datasets.
To accurately select the best classification algorithms that give more accuracy in our current situation, different amount of training datasets starting from (453 to 3778 flows) have been applied to each of the 15 classification algorithm so as to examine their accuracy. Also to obtain the best accuracy, according to our previous work [18] we choose flows from servers to client's direction. Finally the 4 top accuracy algorithms have been determined.

## 4. Results and Discussion:

In our work we neglect UDP flows because most of the applications use TCP as a transport layer protocol. Furthermore the number of UDP flows that consist of less than or equal to 2 packets per flow equals to 84.15 %

As stated in the methodology our initial testing domain started with examining the accuracy of thirty (30) machine learning classification algorithms to identify the originating application of the flow based network traffic. The 15 top (with accuracy.>70%) classification algorithms were chosen for further accuracy investigation. This has been shown in and figures 4-1 and table 4-1.

| Data Series → | | 1 | 2 | 3 | 4 | 5 | | Average Of Accuracy |
|---|---|---|---|---|---|---|---|---|
| Algorithm No ↓ | Data/ algorithm | 453 | 978 | 1503 | 2553 | 3078 | 3778 | |
| 1 | Bayesnet | 73.7 | 74.53 | 77.31 | 77.43 | 78.16 | 79.11 | 76.71 |
| 2 | IBI | 100 | 99.69 | 99.73 | 99.49 | 99.54 | 99.52 | 99.66 |
| 3 | IBK | 100 | 99.69 | 99.73 | 99.64 | 99.67 | 99.68 | 99.74 |
| 4 | Kstar | 95.36 | 94.06 | 94.67 | 93.57 | 93.3 | 93.19 | 94.03 |
| 5 | BFTree | 89.85 | 87.12 | 89.68 | 88.09 | 90.44 | 90.92 | 89.35 |
| 6 | J 48 | 88.78 | 88.03 | 86.36 | 86.44 | 86.48 | 85.86 | 86.99 |
| 7 | lmt | 91.17 | 90.28 | 88.56 | 83.3 | 87.26 | 87.18 | 87.96 |
| 8 | nbtree | 75.74 | 80.87 | 82.96 | 83.5 | 81.54 | 83.77 | 81.4 |
| 9 | Random forest | 99.56 | 99.08 | 99.334 | 99.13 | 99.38 | 99.47 | 99.33 |
| 10 | Random tree | 100 | 99.69 | 99.73 | 99.64 | 99.67 | 99.68 | 99.74 |
| 11 | REP tree | 80.79 | 77.33 | 82.96 | 82.02 | 82.32 | 85.7 | 81.85 |
| 12 | simpleCart | 90.29 | 88.75 | 91.28 | 91.89 | 92.59 | 92.61 | 91.24 |
| 13 | rules.jrisp | 70.4 | 74.23 | 78.04 | 79.7 | 79.98 | 77.42 | 76.63 |
| 14 | rules.part | 86.7 | 85.37 | 83.83 | 85.82 | 80.83 | 84.83 | 84.56 |
| 15 | rules.rider | 80.13 | 81.08 | 81.9 | 81.66 | 82.97 | 83.24 | 81.83 |

**Table 4-1 Accuracy matrix of algorithms (column) corresponding to the various training data sets (row)**

As can be seen from table 4-1 , and figure4-1 it is clearly reported that random tree , IBI, IBK, random forest respectively provide the top 4 highest accuracy in classifying flow based network traffic to their corresponding application type among the 15 selected algorithms as an overall average accuracy. Furthermore the results also show that theses algorithms give high accuracy regardless the amount of training data sets (average accuracy of more than 99.33%).

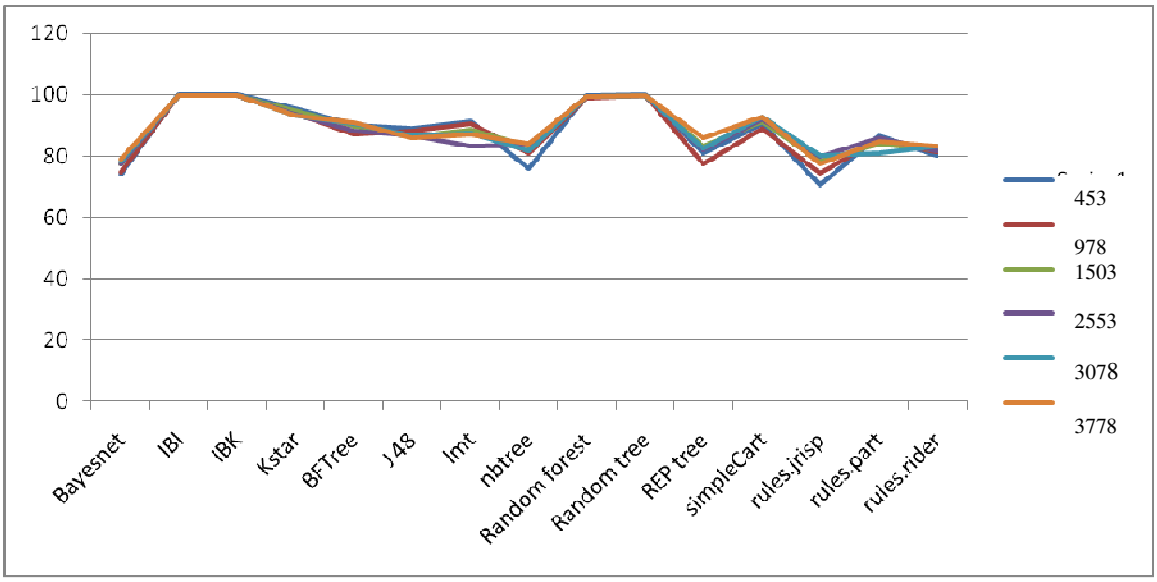



**FIGURE 4-1:** algorithms and their corresponding accuracy with various training data sets

## 5. Conclusion and Future Work

This paper evaluates the accuracy of 30 Machine learning classification algorithms as one of the important performance metrics using custom made datasets.

Our new findings show that random tree, IBI, IBK, random forest scored the top 4 highest accuracy classification algorithms among others for identifying flow–based network traffic to their corresponding application type.

However we started with offline classification and selected the best algorithms based on accuracy. This chosen algorithm will also be tested for processing time as a future work and the best algorithm according to time and accuracy will be used for real time inline detection.

Since our primary goal is to regulate the network traffic and to optimize the bandwidth, our future work must consider the time factor of the classification model.

## 6. REFERENCES

[1] Daniel Roman Koller, Application Detection and Modeling using Network Traces, master thesis “swiss federal institute of technology,2007

[2] http://www.iana.org/assignments /port-numbers

[3] A.W.Moore and D.papagiannaki, “Toward the accurate Identification of network applications”, in poc. 6th passive active measurement. Workshop (PAM), mar 2005,vol. 3431, pp 41-54

[4] Williamson, A. M. C. (2006). A Longitudinal Study of P2P Traffic Classification. Proceedings of the 2th IEEE International Symposium on (MASCOTS '06), Los Alamitos, California, IEEE. Pp 179 - 188

[5] T. Karagiannis, A. B., and N. Brownlee (2004). Is P2P Dying or Just Hiding? . GLOBECOM '04. Dallas, USA, IEEE: pp:1532 - 1538 Vol.3.

[6] Thomas, K., B. Andre, et al. (2004). Transport layer identification of P2P traffic. Proceedings of the 4th ACM SIGCOMM conference on Internet measurement. Taormina, Sicily, Italy, ACM: pp: 121 - 134

[7] Subhabrata, S., S. Oliver, et al. (2004). Accurate, scalable in-network identification of p2p traffic using application signatures. Proceedings of the 13th international conference on World Wide Web. New York, NY, USA, ACM: pp: 512 - 521

[8] Christian, D., W. Arne, et al. (2003). An analysis of Internet chat systems. Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement. Miami Beach, FL, USA, ACM: pp: 51 - 64

[9] Patrick, H., S. Subhabrata, et al. (2005). ACAS: automated construction of application signatures. Proceedings of the 2005 ACM SIGCOMM workshop on Mining network data. Philadelphia, Pennsylvania, USA, ACM: pp: 197 - 202

[10] Feldman, R. S. a. A. "An IDS Using NetFlow Data." Retrieved march 2008.

[11] Jeffrey, E., A. Martin, et al. (2006). Traffic classification using clustering algorithms. Proceedings of the 2006 SIGCOMM workshop on Mining network data. Pisa, Italy, ACM: pp: 281 - 286

[12] Thomas, K., P. Konstantina, et al. (2005). BLINC: multilevel traffic classification in the dark. Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications. Philadelphia, Pennsylvania, USA, ACM: pp: 229 - 240

[13] Fivos Constantinou, Panayiotis Mavrommatis, “Identifying Known and Unknown Peer-to-Peer Traffic “, Fifth IEEE International Symposium on Network Computing and Applications (NCA'06) 0-7695-2640-3/06 $20.00 © 2006 IEEE

[14] Nigel, W., Z. Sebastian, et al. (2006). "A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification." SIGCOMM Comput. Commun. Rev. 36(5): 5-16.

[15] Hongbo, J., W. M. Andrew, et al. (2007). Lightweight application classification for network management. Proceedings of the 2007 SIGCOMM workshop on Internet network management. Kyoto, Japan, ACM: pp: 299 - 304

[16] Rui Wang1, Y. L., Yuexiang Yang3, Xiaoyong Zhou4 (16-18 October 2006). Solving the App-Level Classification Problem of P2P Traffic via Optimized Support Vector Machines. Proceedings of the Sixth International Conference on Intelligent Systems