



## ENVIRONMENT FOR AGENT-BASED MODEL IN MOBILE DATABASE TRANSACTION: A REVIEW

Suhaila Mohd Zain<sup>1</sup>, Sharmilla Sulong<sup>1</sup>, Noor Muzlinda Mat Hashim<sup>1</sup> and Zairi Ismael Rizman<sup>2</sup>

<sup>1</sup>Department of Information Technology and Communication, Kuala Terengganu Polytechnic, Kuala Terengganu, Terengganu, Malaysia

<sup>2</sup>Faculty of Electrical Engineering, Universiti Teknologi MARA, Dungun, Terengganu, Malaysia

E-Mail: [suhaila\\_mdzain@pkkt.edu.my](mailto:suhaila_mdzain@pkkt.edu.my)

### ABSTRACT

This paper described the concepts of mobile environment that are vastly used and discussed in the database transaction application. In order to understand the management of databases in a mobile environment, special terms and concepts need to be clarified. This includes of the setting for elements involved and how they are organized. Many technologies are applied to deliver the accessibility of data needed in the mobile database transaction. Features identified in transaction management of mobile database came from the characteristic of the network structure in mobile topology, database operation nature and also limited resources. In this review, we try to cater disconnection issue for planned and unplanned disconnection of mobile database transaction in Mobile Ad-Hoc Network (MANET) architecture. The study also involved various types of database in two types of architectures which are general environment and MANET.

**Keywords:** mobile network architecture, network features, database transaction, agent model in networking.

### INTRODUCTION

Emerging technologies for wireless is useful not just for sharing information and communication among different users in statics. Conversely, varies of device can now be used when a user wants to access information in mobile while the information changes according to the user's movement anywhere and at any time. The mobility has affected the way data is disseminated, the query is processed and the transaction is managed. The current phenomena that we are in are called mobile environment.

#### Concepts clarification

Some people are confused with many terms and definition that being used to refer to the mobile computing area. Mobile computing involves mobility of hardware, data and software in computer applications. It can be viewed as a specialized class of distributed systems which consisted of either client-server model or peer-to-peer model.

Mobile computing is a special case of distributed system where connections between hosts are dynamically changed [1]. It involves computations and data transfer over distributed and mobile nodes, not just voice communication such in mobile communications and point-to-point or person-to-person as in telecommunications, e.g. from the mobile node to databases, thus mobile computing applications can be applied over telecommunication networks which comprised of voice and data. It involves mobility, but not necessarily with wireless computing that is wireless which does not imply mobility [2]. In mobile computing, there can be stationary nodes which connected wirelessly such as wireless local loop from "backbone" to home. To simplify, it can be said as wireless computing aids mobile computing. There are many issues such as

mobile computing without wireless networks. Nomadic computing which often means that user mobility without device mobility. Three mobility paradigm which viewed from user side can be identified as static, nomadic and continuous mobility. Nomadic definition which according to [3], "the ability to move easily from place to place and retain access to information and communication services while moving, at intermediate stop and at the destination", but the nomadic currently refer to portability. In [4] stated that nomadic computing refers to mobile computers in integrated part of distributed computing environment. Meanwhile, continuous mobility can be defined where users are always reachable and network provides reasonable communication capabilities for moving users [3]. Further explained that a mobile motion of objects can be in two approaches, user movement (personal mobility) or movement of the user's device (device mobility).

#### Components involved in mobile computing

There are many similar terms that used to refer to the same component. According to previous research, mobile computing environment basically includes [1, 2, 5-8]:

- i. A fixed network which is a distributed system.
- ii. Fixed Host (FH) typically is a computer with high speed fixed network, but has no connection with Mobile Host (MH).
- iii. Mobile Unit (MU) or MH or sometime called Mobile Terminal (MT) is a mobile computer connected to the fixed network via a wireless link. In [2] prefer Mobile Station (MS) to refer to same thing. MH in some respect may also refer to sites.
- iv. Base Station (BS) or Mobile Support Stations (MSS) by using a wireless interface to connect to an MU.



- v. Database Server (DBS) has database processing capability.
- vi. Cell where each BS/MSS is responsible for all of the MHs in a given logical area. A cell can be wireless local area network (WLAN), ad-hoc network or a cell phone network.

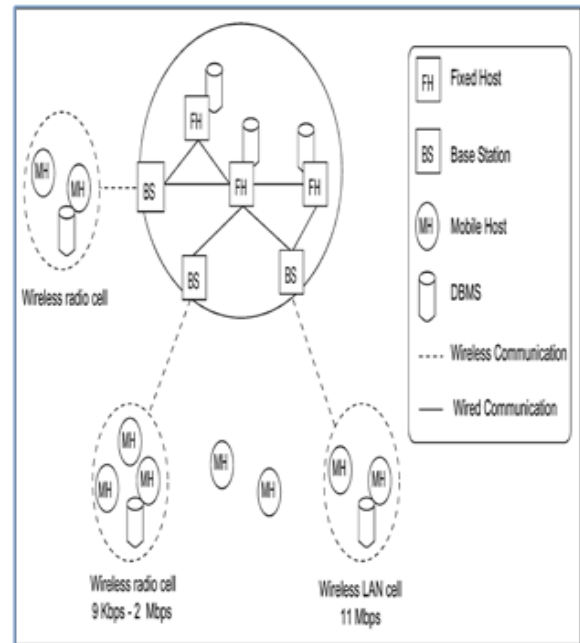
Mobile and fixed hosts can be either clients or servers and MHs can be of different natures, from a personal digital assistant (PDAs) to personal computers. An MU may change its location and network connections while computations are being processed. In cellular wireless networks, MTs only communicate with the BS but not with each other.

Here, no specific assumption are made about the database model (relational, object) and the centralized or distributed nature of the database management system (DBMS). The same applies to models in a distributed system (peer-to-peer/client-server). To understand the movement of MU, handoff procedure is used as a process of entering a new cell where it can be called as hand-off or hand-over [6]. Information of current MH state should be transferred to new BS while crossing the boundaries of a cell [1].

#### Architecture for mobile computing system

Basically, mobile computing architectures can be categorized into two types [7].

- a) General Architecture consists of fixed BSs/MSSs that maintain all MHs in its cell. Control handover done by new MSS when an MH moves to a new cell. This architecture concerned the mobility of users as MH. Figure-1 shows a topology of mobile computing from prior research.
- b) Mobile Ad-Hoc Network (MANET) Architecture consists of MHs that roamed in the network. They are connected to a wireless network that frequently changed its topology, so there is no fixed infrastructure and fixed MSSs. This architecture has databases which are distributed in heterogeneous community [9]. It is dynamically configurable structure suits with its name as ad-hoc. It also as several mobile nodes which sharing one or more wireless channels without centralized control or an established infrastructure.



**Figure-1.** General architecture used by [6] for survey in mobile transactions.

In [10, 11] agreed that general architecture of the mobile environment consists of MUs and FHs. However, in [12] added that each MH includes several applications and one small DBMS. The function is to retain database consistency of the local application transaction. Replication control tools are then used to maintain replicated data in mobile database environments for data synchronization by multiple MHs.

In other paper, in [1] added that both BS and FH need DBS help to perform transaction management. The DBSs can be set either at the BS or FH. It suggests that most commonly used applications are provided by BS, thus gives advantage to MH which has power and storage constraints.

In [6] considered that MHs can store and run DBMS modules. The communication capabilities between MHs and BSs are asymmetric. It was assumed that no direct communication between MHs, so this limits the communication just by BS. Another unique features which stated by [13] that MH can initiate a transaction to disseminate it directly among a set of MH or FH. However, there must be a specific protocol to support movement of these MH while accessing data from MH through the nearest BS.



In [7] stated many previous researchers which employed the first architecture for mobile database transaction management. However, they raise issues in MANET, an architecture which concerns about the movement of database stored in MH. The added feature to the architecture is that two MHs of different cell can communicate by using intermediate MHs between them. This has made it a dynamic and effective architecture that will enhance the transaction processing resource [7-9, 14].

### Data management in mobile computing

Diverse data possibly from database systems, data warehouses, information services and other Internet source [15] which can be accessed by user in mobile environment. There are heterogeneous database systems which available in mobile database computing such as object oriented database, relational database, extensible markup language (XML) type or others. Regarding this scenario, two types of database system are reside in the mobile environment, one is located in fixed host and another located in mobile host. The latter represent mobile database [14].

A database which store locations of users will often change due to the user's location varies at times. According to [16], the database is structured to local databases and distributed at different BSs. It consists of temporal and non-temporal data objects. The difference between both is that temporal data objects are used to record object status in the external environment. The timestamp is associated with each temporal data object to trace the data object life. The timestamp is given when a transaction is initiated and set upon it if the transaction is successfully committed before its deadline.

There are two operations which regarding data broadcast into the environment namely push-based and pull-based. Distributing information to clients without requests is a push-based, whereas clients request data from server is pull-based operation [10]. The fact is then expanded by [17] stated that there are three communication methods in MANET, broadcast, data query and peer to peer messages. However, not all types have been used in an application altogether. The broadcast is the most practical approach for conserving energy in which the other MUs just need to listen and forward relevant message or query.

Management of data in mobile and distributed environment differs in such a way where the distributed application focused on "location transparency" whereby in mobile application focused on "location awareness" [5]. According to [1, 14, 18], data management in mobile computing can be described as global and local data management. Global data management relates with network aspect such as location and addressing to find routes and send packets, replication such in copying data as local and broadcasting to discovers and exchange messages.

On the other hand, local data management refers to the end user point of view. For example, energy utilization of data access operation, disconnection management possibly from handoff and query processing regarding atomic transaction.

Characteristic of mobile database in mobile computing environment have been analyze by many researchers [1, 2, 10, 18, 19] from various aspect such as it impact to data management, it features to user choices and it constraints to devices. Below is a list of the characteristics such as:

- i. Asymmetry in the communication
- ii. Energy constraint/power limitations
- iii. Size of screen
- iv. Communication in a cell by using broadcast
- v. Limited resources such as storage capacity
- vi. Varies locations at times (dynamic topologies)
- vii. Scalability or coverage
- viii. Limited security (vulnerable to intruders)
- ix. Frequent disconnections (and predictable disconnections)
- x. Greater mobility implies a weaker network connection
- xi. Bandwidth availability (effect quality of connections, more nodes will reduce network capacity)

In [20] consider that querying processing are basic task in mobile database systems. In discusses an approach to optimize query regarding fast changing location in mobile database systems. A query will depend on the state of the MU (connected or disconnected) thus executed differently.

Some query in the database may refer to location dependent data. In [21] stated that moving object changed position continuously as a function of time. Query can also be identified as remote access to data in heterogeneous MDBC. The MDBC need to be integrated where it includes many mobile databases to process query [14]. These queries can refer to both mobile nodes and fixed nodes accessing data through a network connection characterized by constraints as listed.

These nodes made connections to the network to access data either through land-based lines (a modem and telephone lines) or through a wireless medium. User interaction in a mobile database is accomplished through transactions [14, 15, 22].

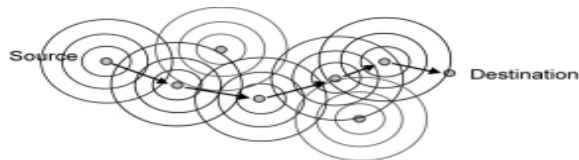
### MANET architecture

Specifically, mobility covers movement of object (users or devices) between geographical location, different network, changes in communication devices and different application [1]. In this research, it considers physical mobility of devices and logical mobility of database queries or transactions. Whatever the aspect of mobility is, continuous connectivity of wireless access must be supported especially in ad-hoc networks.



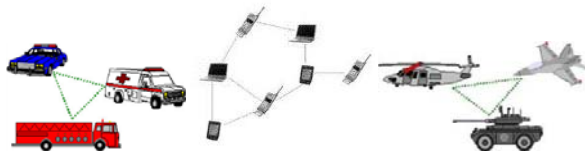
There is existed two different ways of configuring a mobile network, infrastructure based and ad-hoc based. The first used fixed network access points (base stations) such in WLAN (IEEE 802.11) and global wireless networks (global system for mobile communications (GSM), general packet radio services (GPRS), universal mobile telecommunication system (UMTS)).

An ad-hoc network existed for the duration of the communication only. This refers to the nature of the network being transitory, and only coming together for a short time when it is needed without the need for permanent infrastructure [23]. Due to the lack of infrastructure, MU act as routers and forward packets to others. Each MU periodically broadcasts its power level and location information to each other.



**Figure-2.** Dynamic topology and the communication in MANET [24].

The infrastructure has dynamic topology, in which the MUs will get in and out from the network without fixed setup. MANET applications have common characteristic such that they have same logical group of data accessed from the network. It moved randomly but still within a finite distance, for example 802.11b wireless distance is 100 meters. In the mobile computing environment, each MU has its own database system [25]. It changes its location while the computation is being processed. Many applications apply this feature in tactical networks, emergency services, entertainment, location-aware services, etc. Figure-3 are examples of them.



**Figure-3.** Sample application showing the communication between nodes [24].

However, the popularity of MANET application comes with restriction in resources. Due to this problem, many research discussed on how to manage the resources effectively. Some works have tried to reduce energy consumption by assigning each MU with different modes. Their modes identified are Active, Doze and Sleep as shown follows [1, 7, 8]:

- **Active Mode:** The MU performs its usual activities. Its central processing unit (CPU) is working and its communication device can transmit and receive signals.
- **Doze Mode:** The CPU of the MU will be working at a lower rate. It can examine messages from other MUs. The communication device can receive signals. So, the MU can be awoken by a message from other MUs.
- **Sleep Mode:** Both CPU and the communication device of the MU are suspended due to the communication lost [26]. This is equivalent to a system failure considered in classical transaction processing work.

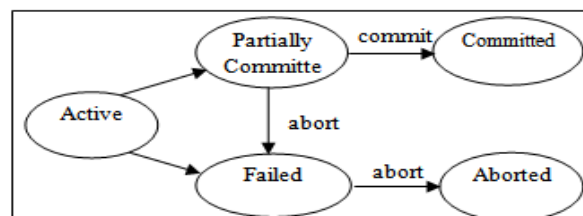
In database management, it finds that there are two types of data namely frequently accessed data and less frequently accessed data. Data accessibility in MANET is lower compared to fixed networks. In order to overcome this problem, data items are replicated at MU in which the MU may not be the original data host. However, this solution is difficult to realize due to the poor resources. According to [27], it is impossible for MU to replicate all data item in the network. So, one possible way is that replication needs to be made in selective way. For example to replicate data to MU that have enough resource or to replicate popular data item.

## METHODOLOGY

Defining disconnection requires one to understand the transaction of database and apply suitable available techniques. The following section discussed the state of transaction in centralized and mobile environment, types of transaction and agent application to handle transaction.

### Mobile transaction and query processing

A transaction is one of the fundamental concepts in a DBMS. A transaction is a program which consists of user access operations; read set such as select statement to retrieves the dataset and write set where this includes all insert, update and delete statements. The write set includes begin and end transaction statements. The transaction effect is made directly on the database resulting commit or abort (rollback). Figure-4 illustrates the flow of transaction states.



**Figure-4.** State transition diagram of a database transaction [28].



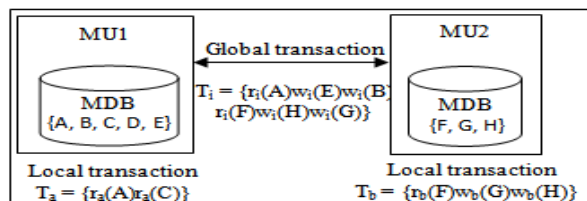
In executing transactions many events happened with or without prior notice. In order to associate the situation, status of transaction operations can be identify as one of the following [1]:

- fully connected (normal connection);
- totally disconnected (e.g., not regarded as a failure of MU);
- weak connection (a terminal is connected to the rest of the network via low bandwidth).

There are many distinctions between a centralized or distributed database transactions compared to transactions in mobile environment. In [1] reports the characteristics as:

- a. The mobile transactions might have to split their computations into sets of operations, some of which execute on a mobile host while others execute on stationary host.
- b. A mobile transaction shares its states and partial results with other transactions due to disconnection and mobility.
- c. The mobile transactions require computations and communications to be supported by stationary hosts.
- d. When the mobile user moves during the execution of a transaction, it continues its execution in the new cell. The partially executed transaction may be continued at the fixed local host according to the instruction given by the mobile user. Different mechanisms are required if the user wants to continue the transaction at a new destination.
- e. As the mobile hosts move from one cell to another, the states of transaction, states of accessed data objects and the location information also move.
- f. The mobile transactions are long-lived transactions due to the mobility of both data and users, and due to the frequent disconnections.
- g. The mobile transactions should support and handle concurrency, recovery, disconnection and mutual consistency of the replicated data objects.

Study from [14] categorized transaction as local and global transaction. Figure-5 illustrates multiple transactions on mobile databases (MDB) with A, B, C, D, E, F, G and H as data item.  $T_1$  is an example of global transaction which consists of sub-transactions  $T_a$  and  $T_b$ .



**Figure-5.** Simple model of local and global transactions in mobile databases.

Therefore, this model generalizes the concepts of distributed transactions by using global transactions.

In mobile environment, at least one MH is involved to call it a mobile transaction [6]. There are suggestions on where mobile transaction management should be relied to. First idea was to put it as MH's responsibility, second suggestion said BSs should handle it and third option suggested that the management moved with the MH which requested the transaction [13].

Many data sets can be executed through multiple BSs in a single mobile transaction. This is due to the MUs movement. The transaction result is atomic in that every operation in the transaction either completes, or does not complete. In other words, transactions are comprised of read/write operations with commit or abort operation at the end [15].

Example of transaction that can be seen in mobile commerce [29] is where a traveller has request information for nearest hotel. They paid a hotel room and made transaction. To succeed in the transaction, the properties of atomicity, consistency, isolation and durability (ACID) need to be fulfilled. By executing all operations of the transaction successfully, the atomicity can be achieved.

In previous researches [9, 14, 15, 30] stated that there are two kinds of transactions to be executed concurrently in MDBC such as mobile transactions and local transactions. The difference of both transactions is that mobile transactions do not have one computer site which serves as the transaction management site [15].

To ensure the correctness of these transactions, serializability needed to adhere. It means that all concurrent of local and mobile transactions should be executed in schedule which effected as a synchronized execution of the transactions. In other words, it depicted that the transaction is executed in isolation without interference from other transaction steps.

From multi-database system (MDBS) perspective, in [11] viewed a mobile transaction not as a global transaction (global transaction are transaction submitted through a global interface which may consist of sub-transactions at multiple local sites [15]). The reason is due to the Data Access Agent (DAA) that act as mobile transaction only which performs limited transaction management functions. In addition, the DAA have to handle a dynamic architecture and must be able to change transaction structure based on MUs movement.

In other research, in [31] classify two basic transactions as online transaction and offline transaction by referring to owner of transaction. Mobile transaction manager directly interact with online transaction at the DBSs attached to FH. However, in offline transaction, the transaction manager will execute and manage the disconnected MH.

Problems during transaction operation can be due to frequent aborts that will resulted delays to the user



request. It then effect the database consistency of facing deadlock [22].

The nature of transactions that moves through the network thus makes a difficult task that is satisfying ACID properties. ACID properties have been regarded as the indicator to satisfy the successful of data communication in mobile computing. These properties impact to the transaction management in aspects as concluded in Figure-5.

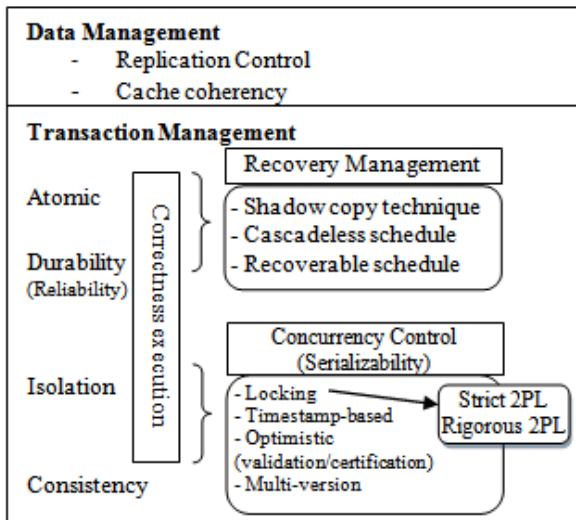


Figure-6. Conclusion of data management and transaction processing.

### Agent model in networking

The use of agent has shown significant success in current applications, especially regarding web services. An agent is autonomous program that can act and react on its' own. They adapt to the environment in which they resides with capability to achieve series of user designed goals, especially in mobile computing [25]. In other words, the mobile codes can do it jobs as prescribed by the system [14].

Generally, the purpose of using agent was to control and handle transaction management. The usage of agent application in the networks mainly where it can move freely and can be customized into the environment. Thus, further gives efficiency and flexibility to the transaction management [25].

Characteristics of an agent model which according to [32] are listed below:

- There are no global transaction manager since agent based computation is decentralized.
- Agent is dynamic in which can be modified at runtime, it decomposition depends on previous actions.

- The interface consists of primitive method from each database system involved.

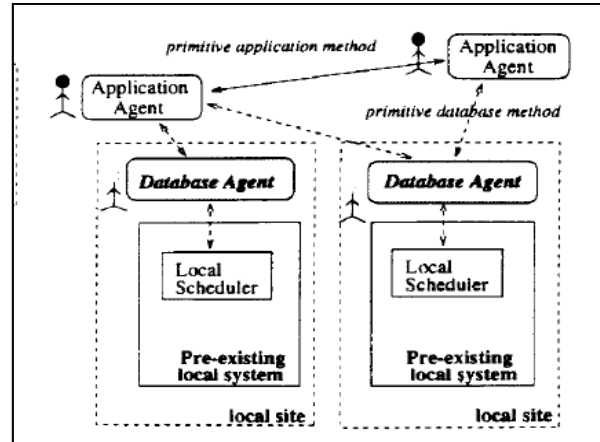


Figure-7. Sample agent based model extracted from [32].

### Methodology and finding

Currently, there are many research which regarding to the mobile transaction processing where most of them in client server for general architecture with fixed network. The research topics include mobile location data management, transaction processing and broadcast, cache management and replication and query processing [1]. In [18], data management problems are identified as management of location dependent data, wireless data broadcasting, disconnection management and energy efficient data access.

Some works promote caching as the key technology for mobile database transactions [1, 26]. Therefore, it will try to focus on recovery by using timestamp and cache consistency. Other aspect of transaction such as concurrency and serializability will also be considered in reviewing the techniques applied.

### Challenges in mobile database transaction

Significant distinction between mobile transaction management and distributed transaction management is transaction movement through the network. This is due to its behavior to access data that may change location or change according to changes in one of the component in the architecture.

Another distinct feature can be seen in the transaction model where a mobile transaction is only identified by the collection of cells it passes through. The execution of the mobile transaction is thus not fully coordinated by the system.

Transaction control may moved along with MH or remain at originating cell when it hops to a new cell. Messages must be sent to new BS if transaction control remains at the originating site where MU needs previous



information. To handle messages overload, the transaction control must moves with the MU [11].

Mobile transaction management system includes problem related to replication control, mobility control and failure handling [33]. Ensuring reliability of shared data becomes difficult due to complexities in mobile computing structures added with limitations and problems in communication channels. Possible problems during handoff may increase the rate of transaction failures such as changing signal strength which then disconnect MH from the network. Partitioning of the network due to MU failures can complicate process to update and to find routes.

Disconnection can happen when a connection link of a mobile host is broken accidentally or intentionally. However, it must be handled as "normal" situations where MHs may disconnect to save battery consumption [6]. A MH should be set to automatic operation even in disconnection mode. Special action must be taken to cater predicted disconnection over active transaction, where several identified steps are [1]:

- Migrating transaction process to FH if the transaction needed no further user interaction.
- Data needed in transaction can be downloaded in advance before disconnection, so the transaction can continue executing locally on the MH after disconnection.
- Transferring log records from MH to FH, a precaution for instability of storage.
- The MH can announce itself upon predicted disconnection by disabling any connection. This is to give alert for distributed protocol to handle the disconnection.

To ensure proper support for mobile transactions (example: transactions initiated on a MH), local autonomy must be given to allow transactions to be processed and committed on the MH regardless of constraint such in disconnection.

One suggestion in [1, 34] for prevention of transaction abort is by using cache in MUs. This further handle frequent disconnection of relocation and low bandwidth problem. In doing it, the availability of wireless connection can be optimized. Caching mechanisms are characterized by its granularity, consistency strategy and replacement policy.

In [35] rise issues of maintaining transactional cache consistency for mobile computing due to frequent disconnection. Narrow bandwidth would be clogged if massive numbers of mobile clients attempted to query a server to validate cache data. It discusses about broadcast facility to support transaction in mobile database system.

Other mechanism to manage disconnected operation is by using hoarding. This technique preload data on mobile

device to allow continued operation, then to serialize log update and integrate the database upon reconnection [36].

Another research introduced transaction processing model as replication method, which created algorithms that would release strict consistency in replicas. It is an alternative to handle site failures in a mobile environment [33].

Techniques in caching schemes and data replication has been rebuilt and re-evaluated to adapt new model and architectures. It synchronously emerge with technology advancement, thus create ideas for problem solutions [5].

### Recovery and concurrency control management

Replication scheme and caching scheme are data duplication techniques that be applied in mobile transactions. According to [22], both schemes differ in scope of the data access. In the first scheme, other outside system can access the replicated data, while cached data only possible for the system in which the data reside.

In [37] proposed Multi-Checkout Timestamp Order (MTCO). This idea is to allow replicated data of MU to access and update with low cost. By applying peer to peer and replication concept MCTO in planned disconnection mode, nodes have to declare their intention and take the object for writing. This is done by obtaining lock on the data item before disconnection, and the use of timestamp to serialize the transactions after the mobile node reconnect to the network.

The approach by using timestamp has similarities with framework by [13], a method to manage data which called Fixed End Timeout Technique (FETOT). The method will managed planned disconnection in mobile database over fixed network with weighted data distribution. In discussed about connection time accessed by user, where the time will determine transaction management. Its function was to reduce delay in total commit time for time-out, then to allocate data to MH and FH as requested by using timestamp order.

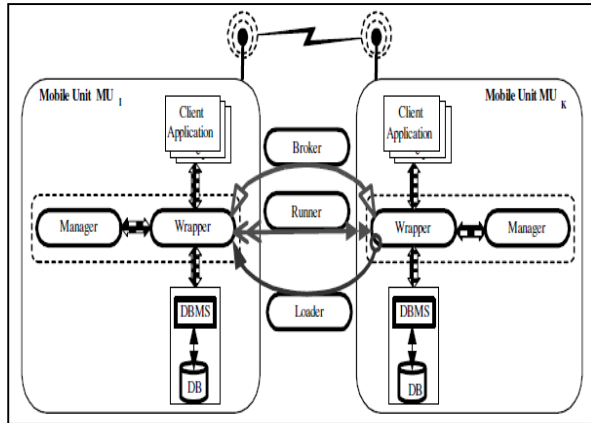
In [19] proposed PRO-MOTION, a transaction processing model to support disconnected database operations between database server and MU. The technique is to use agent named compact, which derived to process transaction request executing at MU. The basic idea is to permit global constraints to be transformed into local constraint in achieving database consistency. Therefore, the model can support dynamic replication as well as caching. However, the interaction of compact agent and compact manager is processed by MSS. So, the architecture relies on MSS to handle the transaction.

### Agent application in transaction processing

In [14] proposed an agent based model in MANET called Accessing Mobile Database (AMBD). The model proposed to support heterogeneous mobile database and query processing concerning concurrency control and



recovery for mobile and distributed transaction. They assume two-phase commit (2PC) protocol is enforced to ensure atomicity in global transactions since all existing database implement the protocol. Mobile agent named wrapper created runner agent to manage global transaction from user, then it will create another agent called broker as a backup for coordinating 2PC process. The interaction of agents involved in the architecture is depicted in Figure-7.



**Figure-8.** Abstract model of AMDB.

The model replaced central coordinator role by creating sequence of agents therefore supports physical and logical mobility. However, too many agents in the architecture will complicate the task and consume more energy.

In [9] proposed a new transaction model based on semantic knowledge with existing mobile agent technology. The model enforced serializability by using two concurrency protocols namely semantic locking protocol and  $S_e$  SGC checking protocol.

Two types of transaction in mobile transaction are considered as local transaction and mobile transaction. The mobile transaction may contain sequence of sub-transactions, which are local transactions executed by mobile database. They simplified the proposed of efficient transaction model for transaction processing by following the relationship:

$$\text{atomic unit} \leq \text{recovery unit} < \text{transaction} \quad (1)$$

The atomic unit represents an execution of sub-transaction without interference from other operations, meanwhile the recovery unit is a definition for a transaction that the execution is interrupted and cannot be completed. By referring to Figure-4, (A, B and E are objects from one mobile database and F, H and G from another mobile database) the model defined a transaction from different database as sequence of modules that is two modules:

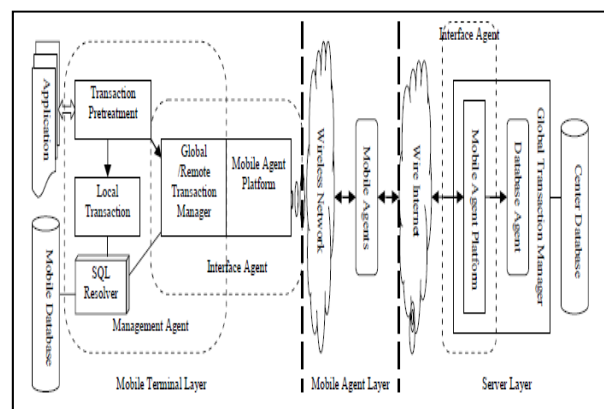
$$T_i = [r_i(A)w_i(E)w_i(B)] + [r_i(F)w_i(H)w_i(G)] \quad (2)$$

Nevertheless, the overall goal to implement self-adaptable scheduler in MDBC still has constraint by creating transaction abort.

In [32] proposed and agent application with timestamp approach. It works with each application agent receives a timestamp defined to be a combination of the clock value and the user's identification (ID). This timestamp corresponds to a global serialization order, in which each application agent serializes all conflicting methods on its local data based on the timestamp order. An operation on its local data issued by another agent is executed only after ensuring that the two agents are allowed to "meet" at a break or relocation point.

MDB model by [38] considered m-commerce transaction with agent support. Management of mobile transaction is done by three different agents such as planning, execution and recovery agents. These agents must keep record of a mobile transaction and gather necessary information. They use timeout techniques to deal with disconnection, in which agents are free to act if the expected duration is already expired. However, mutual agreement needs to be specified before implementing the transaction.

In [25] proposed an architecture based on agent to improve processing efficiency of mobile database system (Figure-8). The model was only applicable on client server architecture. The major functionalities are done by local interface agent who creates four other agents in mobile agent layer. However, their work did not specify any action to handle disconnection and ways to cater limited communication bandwidth which are part of their goal in proposing the new architecture.



**Figure-9.** Mobile database architecture based on agent.

In [34] focused on using dynamic agent to handle cache in MU to prevent transaction abort for disconnection problem. A cache is maintained to keep up with data so that the transaction is not lost due to connection failure.



Every agent created by MSS will have an ID to indicate its boundary and its associated MUs. It works by predicting information to be sent to MU for cache operation. However, MSS does the prediction by using record number between current MU and other MUs with weighted tuples in fuzzy database.

In [39] proposed Agent-based Transaction Management for Mobile Multidatabase (AT3M), an agent based transactions management for mobile multidatabase environment. The agent algorithm allows global sub-transactions to process in parallel and support user mobility while in disconnection. However, the scheme is in client server structure.

To conclude, constraints in mobile database transaction comes with the network structure, mobility features, resource limitation, types of database and various techniques to manage transaction failures.

### Summary of research regarding disconnection issue

Table-1 details are referring to the categorization which made as the following:

- Types of database (Column: DB) which discussed: (1) Distributed Database Systems, (2) Multi-Database Systems, (3) Federated Database Systems and (4) Mobile Databases.
- Techniques which mentioned or discussed (Column: Techniques): (R) recovery, (C) concurrency, (T) timestamps, data replication, caching, (B) broadcast, routing, Global and Local transaction (GLT).
- Architecture type is (General) and (MANET).
- Aspects which not mentioned in the paper are denoted as (-).

**Table-1.** Summary of previous research which regarding to the disconnection problems.

No.	Research by	Year	Architecture Type [17]:		DB	Mobile Agent	Techniques
			Gen	M			
1							
2	T. Imieinski and B. R. Badrinath [18]	1993	√	-	4,1	√	P, H, B
3	E. Pitoura and B. Bhargava [32]	1995	√	-	4,2	√	R, C, T
4	M. H. Dunham <i>et al.</i> [11]	1997	√	-	4,2	√	GLT, R
5	G. D. Walborn and P. K. Chrysanthos [19]	1997	√	-	-	√	R, H
6	S. Lee <i>et al.</i> [35]	1999	√	-	-	-	B, H
7	K.-Y. Lam <i>et al.</i> [16]	2000	√		4,1	-	C, T
8	L. Gruenwald and S. M. Banik [7]	2001		√	4,2	√	-
9	J. B. Lim and A. R. Hurson [22]	2002	√	-	4,2	√	C, P, GLT
10	J. Eder <i>et al.</i> [14]	2003	-	√	4,3	√	GLT
11	S. Han <i>et al.</i> [40]	2003	-	-	-	√	B, O
12	P. Serrano-Alvarado <i>et al.</i> [6]	2004	-	√	4,3,2	-	C, P, H
13	A. Brayner and J. De Aguiar Moraes Filho [9]	2006	-	√	4,2	√	C, P
14	L. Gruenwald <i>et al.</i> [41]	2006	-	√	4	-	C
15	L. Gruenwald <i>et al.</i> [8]	2007		√	4,2	-	C, GLT
16	K. Chang <i>et al.</i> [31]	2007	√	-	4	√	H
17	M. Vijayalakshmi and A. Kannan [42]	2008	√	-	4	√	H
18	Y. Wong Cheow <i>et al.</i> [43]	2008	√	-	4,2	-	B
19	R. Alshorman and W. Hussak [30]	2008	-	√	4,2	-	C, T
20	M. Ongtang <i>et al.</i> [39]	2009	√	-	4,2	√	GLT
21	L. Jing and W. Jianhua [25]	2009	√	-	4	√	GLT
22	A. Rahbar [17]	2009	-	√	4,1,3	-	B
23	W. Jinghua <i>et al.</i> [26]	2010	√	-	4	√	T, B, H
24	W. Romsaiyud and W. Premchaiswadi [44]	2010	-	√	4	-	H



## CONCLUSIONS

This study is not to evaluate the effectiveness of techniques chose to solve problem. Each research has a unique and different approach to cater many types of problems which depends on the environment and scope. These include the type of database, database transaction, techniques to solve problem and methods of disconnection. However, this study has defined and clarified in many terms and concepts that be used in the environment of mobile application especially for database in general architecture and MANET structure.

## REFERENCES

- [1] S. K. Madria, M. Mohania, S. S. Bhowmick and B. Bhargava. 2002. Mobile data and transaction management. *Information Sciences*. 141(3): 279-309.
- [2] Ahmad A. 2005. *Wireless and mobile data networks*. John Wiley and Sons. New Jersey, USA.
- [3] Uskela S. 2002. Mobility management in mobile Internet. In: *Third International Conference on 3G Mobile Communication Technologies*. pp. 91-95.
- [4] R. Alonso and H. F. Korth. 1993. Database system issues in nomadic computing. *ACM SIGMOD International Conference on Management of Data*. 22(2): 388-392.
- [5] Dunham M. H. and Helal A. 1995. Mobile computing and databases: Anything new? In: *ACM SIGMOD Record*. 24(4): 5-9.
- [6] P. Serrano-Alvarado, C. Roncancio and M. Adiba. 2004. A survey of mobile transactions. *Distributed and Parallel Databases*. 16(2): 193-230.
- [7] Gruenwald L. and Banik S. M. 2001. A power-aware technique to manage real-time database transactions in mobile ad-hoc networks. In: *12<sup>th</sup> International Workshop on Database and Expert Systems Applications*. pp. 570-574.
- [8] L. Gruenwald, S. M. Banik and C. N. Lau. 2007. Managing real-time database transactions in mobile ad-hoc networks. *Distributed and Parallel Databases*. 22(1):27-54.
- [9] A. Brayner and J. D. A. M. Filho. 2006. On mobile transaction processing in dynamically configurable mobile database communities. *International Journal of Parallel, Emergent and Distributed Systems*. 21(3): 199-213.
- [10] D. Barbara. 1999. Mobile computing and databases-a survey. *IEEE Transactions on Knowledge and Data Engineering*. 11(1): 108-117.
- [11] M. H. Dunham, A. Helal and S. Balakrishnan. 1997. A mobile transaction model that captures both the data and movement behavior. *Mobile Networks and Applications*. 2(2): 149-162.
- [12] Kim H. W., Park D. S., Rhee H. K. and Kim U. M. 2001. Advanced transaction scheduling protocol for multilevel secure database in wireless mobile network environment. In: *Joint 4<sup>th</sup> IEEE International Conference on ATM and High Speed Intelligent Internet Symposium*. pp. 240-244.
- [13] Khder M. A., Zin A. M., Zulaiha H. and Othman A. 2008. Fixed end timeout technique (FETOT) for planned disconnection in mobile database over fixed network with weighted data distribution. In: *3<sup>rd</sup> International Conference on Information and Communication Technologies: From Theory to Applications*. pp. 1-6.
- [14] A. Brayner and J. A. M. Filho. 2003. Sharing mobile databases in dynamically configurable environments. *Advanced Information Systems Engineering*. 2681: 724-737.
- [15] J. B. Lim and A. R. Hurson. 2001. Transaction processing in a mobile, multi-database environment. *Multimedia Tools and Applications*. 15(2): 161-185.
- [16] K. Y. Lam, T. W. Kuo, W. H. Tsang and G. C. Law. 2000. Concurrency control in mobile distributed real-time database systems. *Information Systems*. 25(4): 261-286.
- [17] Rahbar A., Mohsenzadeh M. and Rahmani A. M. 2009. HDD3M: A new data communication protocol for heterogeneous distributed mobile databases in mobile ad-hoc networks. In: *International Conference on Education Technology and Computer*. pp. 93-98.
- [18] T. Imieinski and B. R. Badrinath. 1993. Data management for mobile computing. *SIGMOD Record*. 22(1): 34-39.



- [19] Walborn G. D. and Chrysanthis P. K. 1997. PROMOTION: Management of mobile transactions. In: ACM Symposium on Applied Computing. pp. 101-108.
- [20] Kottkamp H. E. and Zukunft O. 1998. Location-aware query processing in mobile database systems. In: ACM Symposium on Applied Computing. pp. 416-423.
- [21] Gök H. G. and Ulusoy O. 2000. Transmission of continuous query results in mobile computing systems. *Information Sciences*. 125(1): 37-63.
- [22] J. B. Lim and A. R. Hurson. 2002. Transaction processing in mobile, heterogeneous database systems. *IEEE Transactions on Knowledge and Data Engineering*. 14(6): 1330-1346.
- [23] Golding P. 2005. Next generation wireless applications. John Wiley & Sons. London, England.
- [24] Advanced Network Technologies Division. 2011. Mobile Ad-Hoc Networks (MANETs). [http://www.antd.nist.gov/wahn\\_mahn.shtml](http://www.antd.nist.gov/wahn_mahn.shtml).
- [25] Jing L. and Jianhua W. 2009. A new architecture model of mobile database based on agent. In: First International Workshop on Database Technology and Applications. pp. 341-344.
- [26] Wang J., Zhou Z. and Liu M. 2010. An improved cache invalidation schema based on mobile agent. In: 2<sup>nd</sup> International Conference on Future Computer and Communication. 1: V1-7-V1-9.
- [27] Hara T. and Madria S. K. 2004. Dynamic data replication using aperiodic updates in mobile adhoc networks. In: Database Systems for Advanced Applications. Y. J. Lee, J. Li, K.-Y. Whang and D. Lee (Eds.). 2973: 111-136.
- [28] Silberschatz A., Korth H. F. and Sudarshan S. 2010. Database system concepts. 6<sup>th</sup> Ed. McGraw Hill. Singapore.
- [29] A. Tsalgatidou and E. Pitoura. 2001. Business models and transactions in mobile electronic commerce: requirements and properties. *Computer Networks*. 37(2):221-236.
- [30] Alshorman R. and Hussak W. 2008. Multi-step transactions specification and verification in a mobile database community. In: 3<sup>rd</sup> International Conference on Information and Communication Technologies: From Theory to Applications. pp. 1-6.
- [31] Le H. N. and Nygård M. 2007. A mobile database sharing protocol to increase data availability in mobile environments. In: Advances in Web and Network Technologies, and Information Management. K. C.-C. Chang, W. Wang, L. Chen, C. A. Ellis, C.-H. Hsu, A. C. Tsoi and H. Wang (Eds.). 4537: 50-61.
- [32] E. Pitoura and B. Bhargava. 1995. A framework for providing consistent and recoverable agent-based access to heterogeneous mobile databases. *ACM SIGMOD Record*. 24(3): 44-49.
- [33] Jianting Z. and Wanlei Z. 1998. A web-based design for the mobile transaction management of a distributed database system. In: IEEE Technology of Object-Oriented Languages. pp. 372-380.
- [34] Sagayeshi F. M., Shiri M. E. and Faraahi A. 2009. Using fuzzy theory to information predicate for improvement of cache operation in mobile databases. In: IEEE International Conference on Future Networks. pp. 104-107.
- [35] Lee S., Hwang C. S. and Yu H. 1999. Supporting transactional cache consistency in mobile database systems. In: 1<sup>st</sup> ACM International Workshop on Data Engineering for Wireless and Mobile Access. pp. 6-13.
- [36] Satyanarayanan M. 1996. Fundamental challenges in mobile computing. In: 15<sup>th</sup> Annual ACM Symposium on Principles of Distributed Computing. pp. 1-7.
- [37] Abdul-Mehdi Z. T., Mamat A., Ibrahim H. and Dirs M. M. 2006. Multi-check-out timestamp order technique (MCTO) for planned disconnections in mobile database. In: 2<sup>nd</sup> International Conference on Information and Communication Technologies. 1: 491-498.
- [38] Younas M., Chao K. M. and Anane R. 2003. M-commerce transaction management with multi-agent support. In: 17<sup>th</sup> International Conference on Advanced Information Networking and Applications. pp. 284-287.
- [39] Ongtang M., Hurson A. R., Jiao Y. and Potok T. E. 2007. Agent-based transactions management for



mobile multidatabase. In: 3<sup>rd</sup> IEEE International Conference on Wireless and Mobile Computing, Networking and Communications. pp. 67.

- [40] Han S., Kim K. J. and Kim Y. S. 2003. Efficient migration algorithm for mobile agents in wired/wireless Internets. In: Computational Science. P. M. A. Sloot, D. Abramson, A. V. Bogdanov, Y. E. Gorbachev, J. J. Dongarra and A. Y. Zomaya (Eds.). 2660: 626-634.
- [41] Gruenwald L., Bernedo P. and Padmanabhan P. 2006. PETRANET: A power efficient transaction management technique for real-time mobile ad-hoc network databases. In: 22<sup>nd</sup> IEEE International Conference on Data Engineering. pp. 172-172.
- [42] Vijayalakshmi M. and Kannan A. 2008. Processing location dependent continuous queries in distributed mobile databases using mobile agents. In: International Conference on Signal Processing, Communications and Networking. pp. 119-125.
- [43] Yuen W. C., Ibrahim H. and Udzir N. I. 2008. Distributed real-time processing of range-monitoring queries in heterogeneous mobile databases. In: 3<sup>rd</sup> International Conference on Convergence and Hybrid Information Technology. 1: 74-81.
- [44] Romsaiyud W. and Premchaiswadi W. 2010. Predict the query transaction using fuzzy logic for MANETs data caching. In: 11<sup>th</sup> ACIS International Conference on Software Engineering Artificial Intelligence Networking and Parallel/Distributed Computing. pp. 139-142.