UNIT-4th

Mobile Transaction

A transaction is defined as a collection of database operations that form a single, logical unit of work. It also considered as a set of operations that translate a database from one consistent state to another consistent state. That is a computation processing is considered as a transaction or conventional transaction if it satisfies ACID properties. The properties are Atomicity, Consistency, Isolation and Durability. The meaning of each is described below :—

Atomicity All operations occur or none at all. An executable program, assumed that this program will finally terminate, has one initial state and one final state. If the program achieves its final state it is said to be committed. Otherwise if it is at initial state after some execution steps then it is aborted or rollback.

Consistency Relations and constraints among data elements and preserve consistency of data.

Isolation Each transaction must appear to execute as if no other transaction is executing at the same time.

Durability The effect of the transaction must be permanent and persist even if the system failure. Thus failure can't change the result.

The start of a transaction is an implicit BEGIN. In the normal operation, a transaction ends when COMMIT or ROLLBACK is issued. The COMMIT statement ends the current transaction by making all pending data changes permanent and automatically begins a new one. The ROLLBACK statement ends the current transaction by discarding all pending data changes. If a transaction terminates due to system crash, the state of database depends upon DBMS implementation.

Transaction Operations

⇒ **Begin Transaction** This marks the beginning of transaction execution.

⇒ **Read or Write** These specify read or write operations on the database items that are executed as a part of transaction.

⇒ **Commit Transaction** This signals a successful end of the transaction so that any changes executed by the transaction can be safely committed to the database.

⇒ **Rollback** This signals that the transaction has ended unsuccessfully, so that any changes or effects that the transaction may have applied to the database must be undone.

⇒ **End Transaction** This specifies that read and write transaction operations have ended and marks the end of transaction execution.

```
Begin_transaction()
    Execution of transaction program
If (reach_final_state) then
        Commit_Work (final_state)
Else
    Rollback_Work (initial_state)
END_transaction.
```

Programming Model of Transaction

Mobile Transaction Characteristics

The access of the information systems through mobile computers will be performed with the help of mobile transactions. However, a transaction in this environment is different from the transactions in centralized or distributed database in the following ways :—

- ⇒ The mobile transactions might have to split their computations into sets of operations, some of which execute on mobile host while others on stationary host. A mobile transaction share their states and partial results with other transaction due to disconnection & mobility.
- ⇒ The mobile transactions require computations and communications to be supported by stationary hosts.
- ⇒ As the mobile hosts move from one cell to the another, the states of transaction, states of accessed data objects and location information also move.
- ⇒ The mobile transactions are long-lived transactions due to the mobility of both the data and users, and due to the frequent disconnection.
- ⇒ The mobile transactions should support and handle concurrency, recovery, disconnection and mutual consistency of replicated data objects.

System Log

To able to recover from failures that affect transactions, the system maintains a log to keep track of all transaction operations that affect the values of database items. This information may be needed to permit recovery from failures. The log is kept on disk, so any type of failure except for the disk. In addition, the log is periodically backed up to storage to gaurd against the failures.

Mobile Transaction Models

Kangaroo Transaction Model

The architecture of the Kangaroo Transaction Model consists of three tier. An important addition to the common mobile computing architecture is the inclusion of DAA (Data Access Agent) in the middle tier. It is assumed that each MSC is capable of hosting a DAA. Major function performed by the services in the middle tier is Mobile Transaction Management (MTM). MTM involves the tracking of execution status of all transactions, logging recovery information, forwarding mobile host transaction requests to underlying DBMS and participating network handoff as MH moves from one station to another.

Reporting and Co-Transactions

Chrysanthis has proposed a transaction view for mobile computing. Like Kangaroo model, they view MTs as being built using concept of multidatabase transactions. To manage the mobile transactions, they assume that a GDBS exists at each base station to control the execution of the mobile transaction. They do assume that the subtransactions of mobile transaction will commit or abort independently and that if a subtransactions aborts, all others which are yet to be committed are also abort. However, they also have two additional types of subtransactions.

⇒ Reporting Transaction

⇒ Co-Transaction

Clustering Model

This model is a flexible two-level consistency model which deals with the frequent, predictable and varying disconnections. With the clustering model, the database is divided into the clusters. A cluster defines a set of mutually consistent data. Data are stored or cached at a mobile host (MH) to support its autonomous operations during disconnections. Transactions in this model may involve both remote data and data stored locally at the user's device. The items of a database are partitioned into clusters and they are the units of consistency in that all data items inside a cluster are required to be fully consistent, while data items at different clusters may exhibit bounded inconsistencies.

Clustering may be constructed depending on the physical location of the data: By using this locality definition, data located at the same, neighbor or strongly connected hosts may be considered to belong to same cluster, while data residing at disconnected or long distances may belongs to seperate cluster. In this way, a dynamic cluster configuration will be created when a user enters into a new cell, it can change its cluster too. Therefore, clusters of data may be explicitly created or merged by a probable disconnection or connection of the associated mobile host. It may be possible to differentiate users based on the requirements of their data and applications. For example, the data that are most often accessed by some user or data that are private to user can be considered to belong to the same cluster independent of their physical locations.

Integrity constraints specify or express the relationship of data items that a database state must satisfy in the form of restrictions. Integrity constraints among data items inside the same cluster are called <u>intra-cluster constraint</u> and constraints among data items at different clusters are called <u>inter-cluster constraints</u>. There are two types of transaction in the cluster model :—

⇒ <u>Weak Transaction</u> Consists only weak read and weak write operations and they only access data copies that belong to same cluster and can be considered as local to that cluster.

⇒ <u>Strict Transaction</u> This type of transaction involves strict read and strict write operations.

Weak Read operation on a data item reads a locally available copy, which is the value written by weak write operation in that cluster. A weak write operation writes a local copy and is not permanent unless it is committed in the network. A Strict Read operation is defined as one that reads the value of the data item which is written by last strict write operation where a strict write operation means writing one or more copies of the data item. Weak operations only support disconnected operation since a mobile device can operate disconnected as long as applications are satisfied with local copies. Users can use Weak Transaction to update mostly private data and Strict Transactions used to update highly used common data.

Semantic-Based Model

The semantics-based mobile transaction-processing scheme views mobile transaction processing as a concurrency problem. The model assumes a mobile transaction to be a long lived characterized by delays and unpredictable disconnections. This approach utilizes the object organization to split large and complex objects into smaller fragments. A stationary database server checks the fragments of a object on a request from Mobile Unit. On the completion of the transaction, the mobile hosts return the fragments to the server. These fragments are put together again by merge operation at the server. If the fragments can be recombined in any order then the objects are known as reordable objects.

Multidatabase Transaction Processing Manager

A multidatabase system (MDS) is defined as an integrated distributed database system consisting of a number of autonomous component database management systems. Each of component database system is responsible for the management of transactions locally. To facilitate the execution of the global transactions, an additional layer of software must be implemented which permits the scheduling and the co-ordination of transactions. The MDSTPM consist of the following components :—

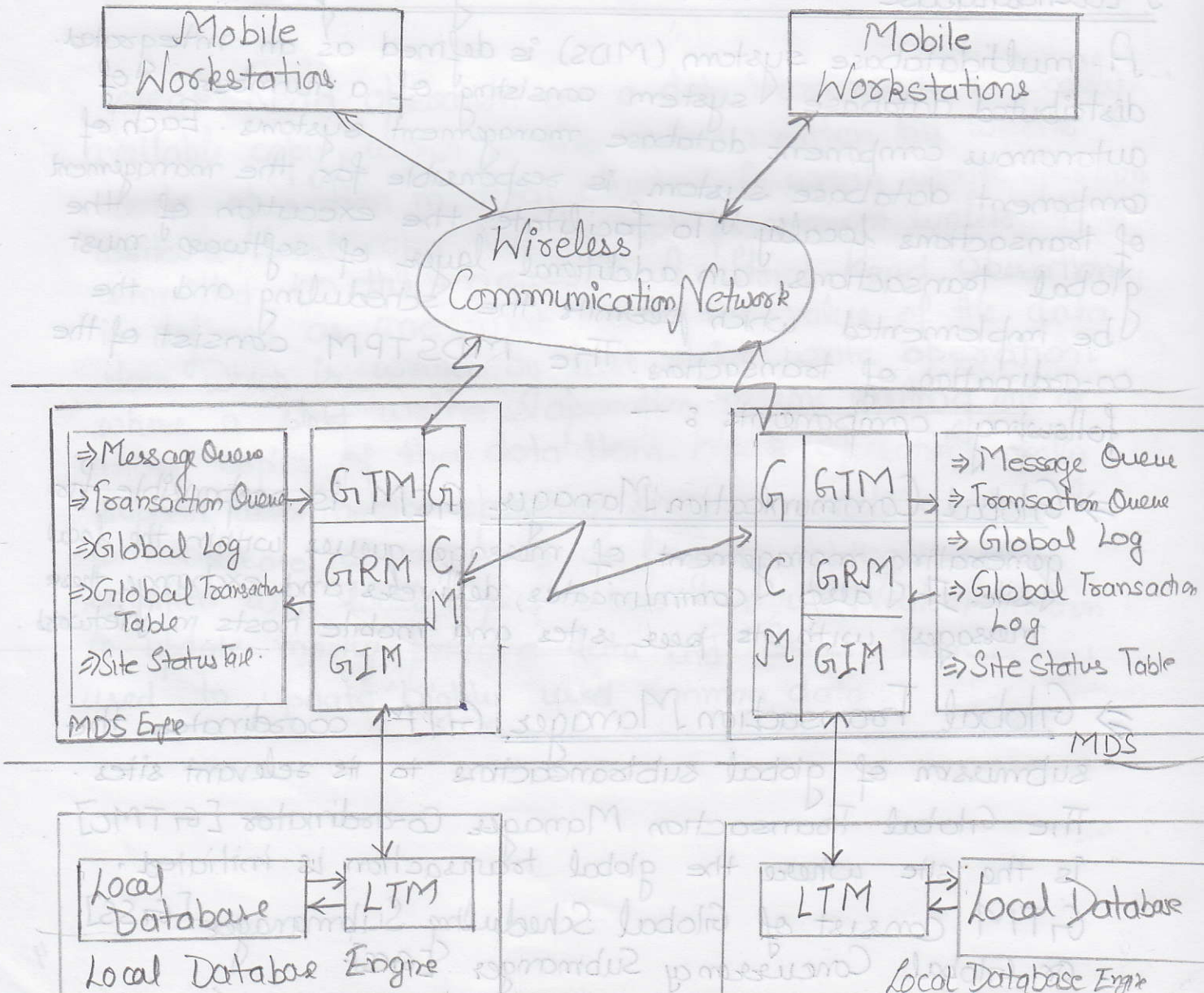
⇒ Global Communication Manager GCM is responsible for generating management of message queues within the local site. It also communicates, delivers and exchange these messages with its peer sites and mobile hosts in Network.

⇒ Global Transaction Manager GTM coordinates the submission of global subtransactions to its relevant sites. The Global Transaction Manager Co-ordinator [GTMC] is the site where the global transaction is initiated. GTM consist of Global Scheduling Submanager [GSS]

The GCS is responsible for acquisition of necessary control techniques needed for successful execution of global transaction & subtransaction.

⇒ <u>Global Recovery Manager</u>. GRM coordinates the commitment and recovery of global transaction & subtransactions after a failure. It ensures that the effects of committed global subtransaction are written to local database. It also uses write-ahead logging protocol so that the effect to the database are written immediately without having to wait for global transaction to complete or commit

⇒ <u>Global Interface Manager</u> GIM coordinates the submission of request/reply between MDSTPM and local database manager.



Mobile Transaction Models.

Models	Database System Model	Additional Infrastructure	Execution In
Reporting and Co-transactions	Multidatabase	Transaction Manager	MU or fixed network
Kangaroo Model	Heterogeneous multidatabase	Data Access Agent	Fixed Network
Clustering Model	Fully Distributed Database	Strict and Weak Transaction	MU or Fixed Network
Semantics based Model	Distributed Database	Fragmentation based object	Restricted Server MU
MDSTPM	Heterogeneous multidatabase systems	MDSTPN Layer	MU or fixed Network.

Kangaroo & Joey Transaction

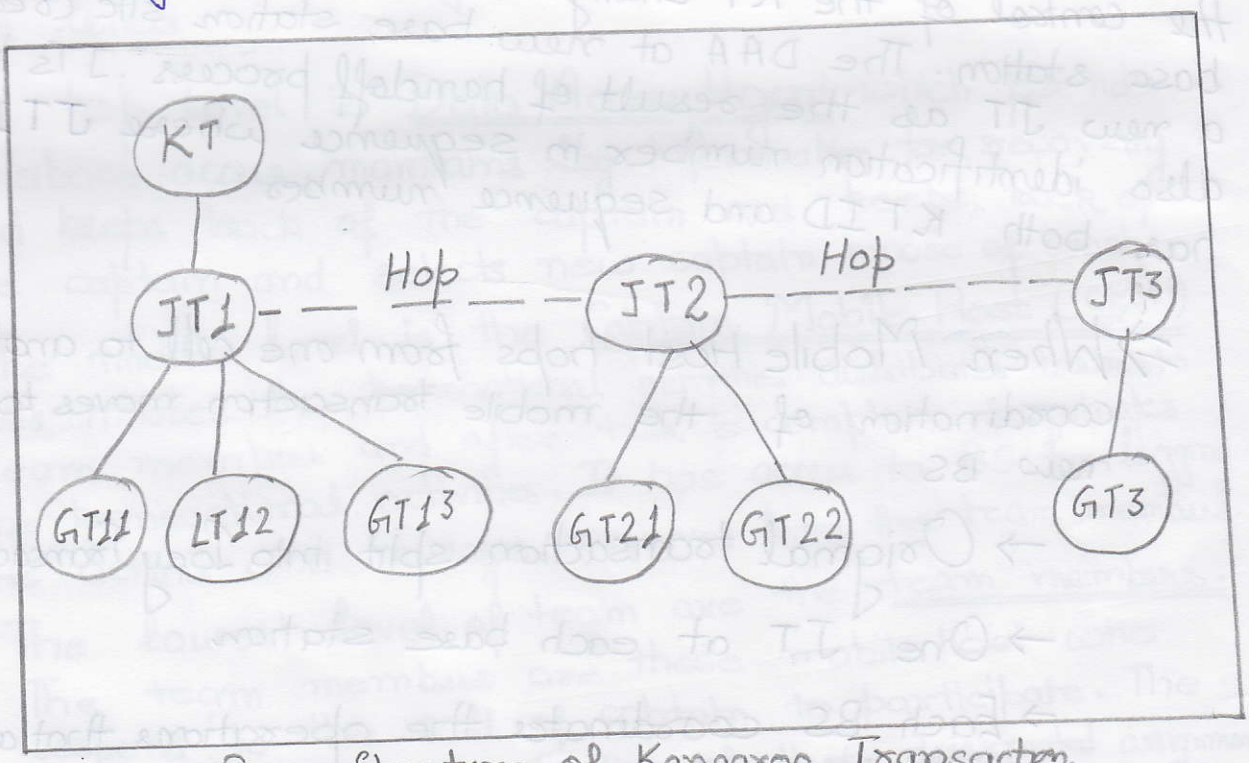
A mobile transaction model named as Kangaroo Transactions, which incorporate the property that the transactions in a mobile environment hop from one base station to another as the mobile unit moves. The model captures this movement behavior and the data behavior reflecting the access to data located in databases through the network. The reference model assumed that model has Data Access Agent, which is used for accessing data in the database and each base station hosts a DAA. When it receives a transaction request from a mobile user, the DAA forwards it to the specific base stations or fixed hosts that contain the required data. DAA acts as a Mobile Transaction Manager and data access co-ordinator for the site. It is built on the top of an existing Global Database System (GDBS). A GDBS assumes that the local DBMS systems perform the required transaction processing functions including recovery and concurrency. A DAA's view of GDBS is similar to that seen by user at a fixed terminal and GDBS is not aware of the mobile nature of some nodes in the network.

When a mobile transaction moves to new cell, the control of transaction may move or may retain at the originating site. If it remains at the originating site, message would have to be sent from the originating site to the current base station any time the mobile unit requests information. If the transaction management functions moves with the mobile unit, the overhead of these messages can be avoided. For the logging side of this movement, each DAA will have the log information for its corresponding portion of the executed transaction.

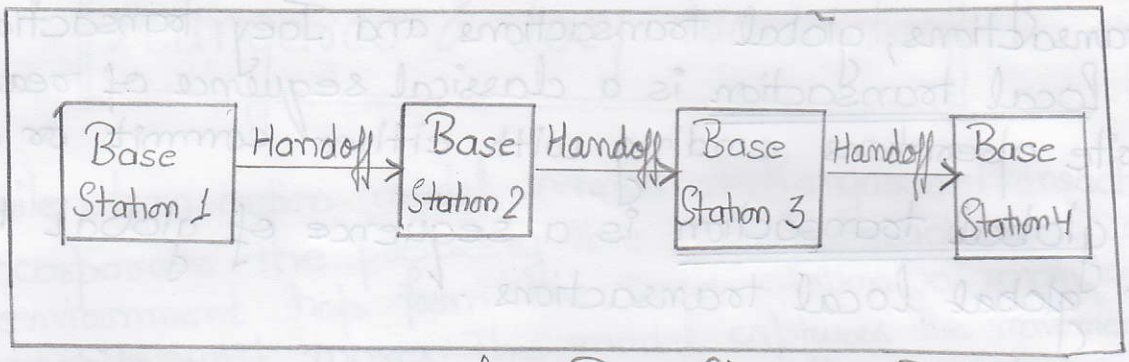
Kangaroo transactions introduce the concept of local transactions, global transactions and Joey transactions.
A Local transaction is a classical sequence of read and write operations ending with either commit or abort.
A global transaction is a sequence of global transactions or global local transactions.

Joey transaction is a sequence of global transactions and local transactions ended by either commit, abort or split.

The model is based on traditional transaction concept which is a sequence of operations including read, write, begin transaction, end transaction, commit and abort. The basic structure is mainly a Local Transaction. On the other hand, Global Transactions (GT) can consist of either sub-transactions viewed as LTs to some DBMS or sub-transactions viewed as sequence of operations which can be global themselves. A hoping property is also added to the model due to mobility of transactions.



Basic Structure of Kangaroo Transaction.



Hopping from Base Station to Base Station

Each sub-transaction represents the unit of execution at one base station and is called Joey-Transaction (JT). The sequence of global and local transactions which are executed under a given KT is defined as <u>Pouch</u>. The origin of base station initially creates a JT for its execution. A GT and JT are different from each other only JT is a part of KT and it must be coordinated by DAA at some base station site. A KT has a unique identification number within the base station. When the mobile unit moves from one cell to another, the control of the KT changes to a new DAA at another base station. The DAA at new base station site creates a new JT as the result of handoff process. JTs have also identification number in sequence where JT ID has both KT ID and sequence number.

⇒ When Mobile Host hops from one cell to another, coordination of the mobile transaction moves to the new BS.

→ Original transaction split into Joey Transaction

→ One JT at each base station

→ Each BS coordinates the operations that are executed while MH was in its cell.

Team Transaction

UNIT - 4th

Adhoc Networks are the most important term in wireless technologies. Mobile Adhoc NETwork (MANET) is a set of wireless mobile hosts forming a dynamic autonomous network through a fully mobile infrastructure without any centralized access points or base stations. In such a network, each node acts both as a router and has a host to facilitate mobility. Due to limited range of wireless network, multiple hops may be needed to exchange data between nodes in the network. MANET generally used in the situations such as natural disasters as earthquakes, floods or frontier operation by defense personnel some time. In these situations it is necessary that operations need to be carried out, the data objects need to be updated.

Team Transaction model is proposed for the cell based systems and MANET. It represent transactional activities carried out by a team. This model consist of three levels.

⇒ The top Level is <u>Data Access Agent</u> which provides database access, maintains log information for recovery and keeps track of the captain and keeps track of the captain and selects new captain incase of crash.

⇒ The middle level is the <u>Captain Mobile Host (CMH)</u> who intiates the transactional activities distributes tasks to team members and after task is complete terminates the transactional activities. It has access to BS for logging its actions and messages received from the team members.

⇒ The lowest level of team are the <u>team members.</u> The team members are those mobile host who respond to the call of captain to participate. The team members on completion of their designated assignment, delegate the responsibility to captain or to another team member.

Team Transaction consists of three ~~act~~ entities :—

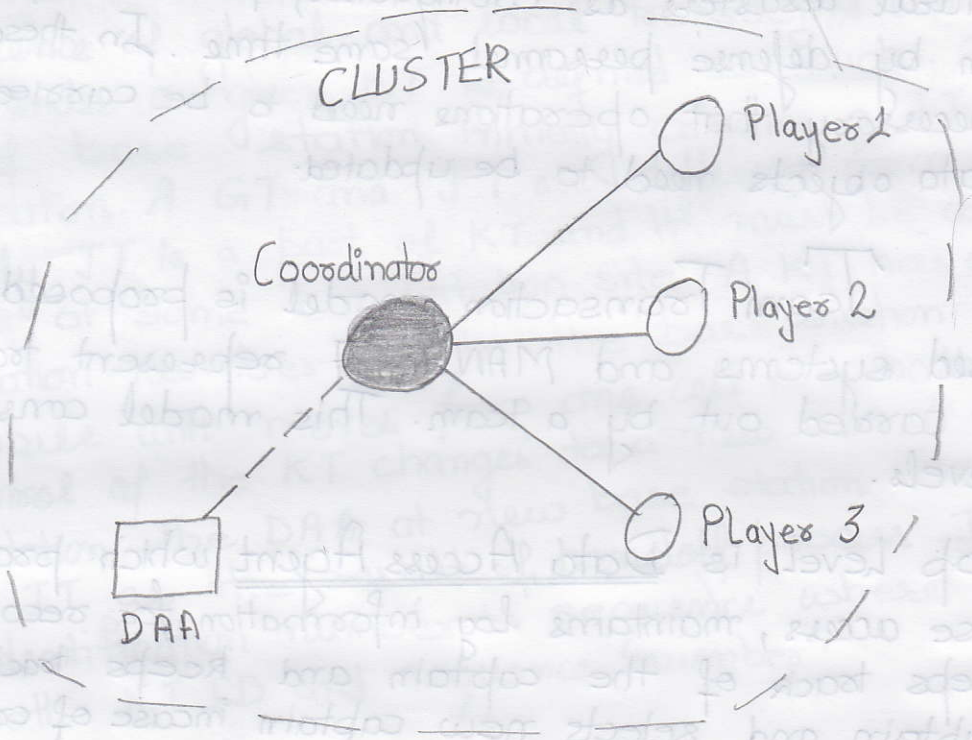
1) Coordinator 2) Players 3) Data Access Agent

These entities form a cluster.

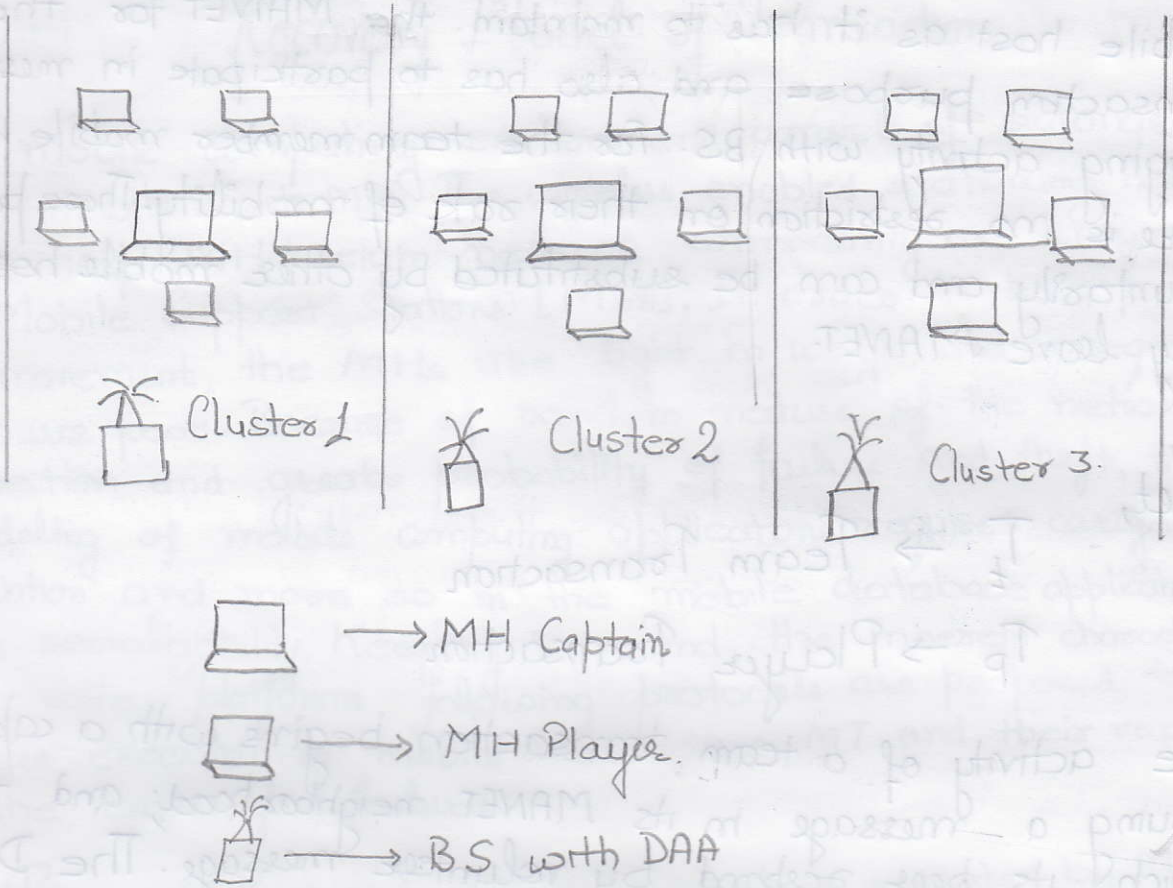
Coordinator is captain of the team and responsible for coordinating the operations of transactions.

Players carry the operations of sub-transaction assigned by coordinator.

DAA provides database access, maintains log information for recovery, and keeps track of the coordinator and selects new coordinator incase of crash.



The captain mobile host initiates the formation of MANET for the purpose of team transaction. The entire team activity is carried through messages which are logged at the BaseStation (BS).



MH Player

- ⇒ Recovery Message
- ⇒ Rollback Message
- ⇒ Delegate Message

MH Captain

- ⇒ Team initiation and member selection
- ⇒ Task distribution
- ⇒ Task substitution from one Team Member to Another

BS with DAA

- ⇒ Message Logging
- ⇒ Message Processing
- ⇒ Buffer Management
- ⇒ Rollback Management

In the team transaction, the captain is the most crucial mobile host as it has to maintain the MANET for the transaction purpose and also has to participate in message logging activity with BS. For the team member mobile hosts, there is no restriction on their rate of mobility. These participate voluntarily and can be substituted by other mobile hosts if they leave MANET.

Let,

$T_t \rightarrow$ Team Transaction

$T_p \rightarrow$ Player Transaction

The activity of a team transaction begins with a captain issuing a message in its MANET neighborhood; and to which its peer respond by volunteer message. The DAA at BS also logs these message thereby creating a message history using the message logs. The history H_{msg} basically comprises of the message logs of the messages exchanged between team members and the captain. The Begin transaction message originating from the captain marks the beginning of the team transaction. This message distinguishes between the volunteers and the actual team members and allow the team members to be in ready state to accept the next message containing the task which initiates the activity at the team members in the form of player transaction. Thus there are two transactions running at the captain one the overall team transaction and the other being the player transaction. The events of the transaction are carried by messages.

Recovery Model of Transaction

The mobile computing environment consists of stationary and mobile hosts MH. The wireless enabled stationary hosts which help MH to retain network connectivity are known as Mobile Support Stations [MSS]. In order to have freedom in movement, the MHs are light in weight and consequently resource poor. Because of random nature of the network connection and greater probability of failure and theft, the modeling of mobile computing applications requires careful attention and more so in the mobile database applications have serializability. Keeping in mind the inherent characteristic of mobile platforms following protocols are proposed to ensure execution of mobile transactions MT and their recovery in the case of failures:-

⇒ <u>Timeout Protocol</u> This protocol is executed by MSS. A timeout parameter is set by mutual agreement of the MH and MSS before commencing the transaction. After expiry of timeout the MSS is free to initiate the rollback activity for the transactions of MH. The protocol is designed to recover from inconsistencies due to unusually long inactivity of the MH. Since MH can interact with several MSS in the course of execution of MT, the time constraint parameter need to be set with each of these MSS.

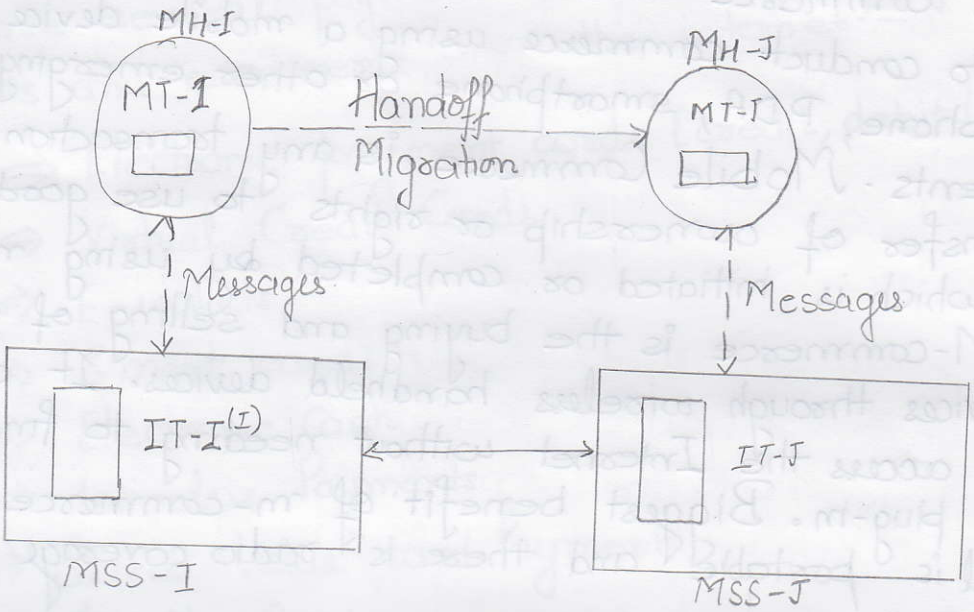
⇒ <u>Disconnect Protocol</u> This protocol is executed by MH. It is used to redefine the time constraints. It may be required that MH has to disconnect itself from MSS. The disconnection may be warranted due to fading power of battery at MH or fading strength wireless signals. As in case timeout protocol, the rollback for the transactions of MH is initiated if disconnection extends beyond the stipulated time limit.

⇒ **Hand-off Protocol** This protocol is executed by MH. MH asks its current MSS to keep its corresponding image transaction in prepared state. Then MH informs current MSS the address of its subsequent new MSS. After the connection is established with new MSS, MH conveys it the address of its previous MSS. This way the complete information of MH and therefore MT is available to both MSS.

⇒ **Migration Protocol** Unlike the Hand-off, in this protocol new settings of MH are communicated to previous MSS by the current MSS, as wireless link is no longer possible between MH and previous MSS. This migration information should reach to previous MSS before it executes the timeout or disconnect protocol.

These four protocols efficiently model the mobile computing scenario for transaction processing. The interactions among the MH and MSS are triggered through messages. Besides message data, messages also carry message identifier, issuing transaction id, mobile host id. The whole scenario can be explained with the help of following figure. The mobile transaction MT-1 is being executed at MH. The actions of this transaction are communicated throug messages to MSS-I. These form image transaction of MT-1 as $IT\text{-}1^{(I)}$. The messages also carry control information for different protocols detailed above. After some time MH moves on to MSS-J either through hand-off or through migration. The image transaction at MSS-I is put to commit and new image transaction $IT\text{-}1^{(J)}$ is formed at MSS-J

System Model for Mobile Transaction Recovery



⇒ There are N MSS. Each MSS has a transaction manager, a LOCK manager, Storage Manager, Recovery manager and Communication manager. ALL MSS poses stable storage. The database is distributed among these MSS. Logging is performed at all MSS.

⇒ MT have their image transactions at one or more MSS. Each MSS maintains its own portion of log.

⇒ Recovery managers co-ordinate among themselves for multiple site recovery, though, it is possible that only a single site has failed.

M-Commerce

Mobile Commerce also known as M-Commerce, is the ability to conduct commerce using a mobile device such as mobile phone, PDA, smartphone or other emerging mobile equipments. Mobile Commerce is any transaction involving the transfer of ownership or rights to use goods and services which is initiated or completed by using mobile devices. M-commerce is the buying and selling of goods and services through wireless handheld devices. It enables users to access the Internet without needing to find a place to plug-in. Biggest benefit of m-commerce is, that terminal is portable and there is radio coverage in major cities.

Customers typically make purchases in m-commerce by sending an advertised "short code" a form of text message to the vendor via mobile phone. The charges may appear on the consumer's mobile service bill, or the customer could arrange for charges to be billed to a credit card or debited from a bank account.

M-commerce Services.

1) Mobile Ticketing
2) Mobile Vouchers
3) Mobile Purchase
4) Mobile Marketing.
5) Location Based Services
6) Information Services

Electronic Payments

There are several payment methods supporting electronic payments and ecommerce over the internet:-

- ⇒ Electronic payment cards [credit, debit].
- ⇒ Virtual Credit Cards.
- ⇒ E-wallets
- ⇒ Smart cards
- ⇒ Electronic Cash
- ⇒ Wireless Payments
- ⇒ Stored-value card Payments
- ⇒ Loyalty Cards
- ⇒ Person to Person Payment Methods
- ⇒ Payments made electronically.

There are several issues to be taken into the account when creating a electronic payments system like:—

<u>Authentication</u> which identifies buyers and also makes sure that person is who he/she claims to be. Used methods are digital signature, finger prints, passwords or smartcards etc.

<u>Data Integrity</u> means that there must be a way to verify that data is not changed during transactions.

<u>Confidentially</u> must be preserved so that information concerning the actions are need to know the transactions.

<u>Non Repudiation</u> means that person who did payment is not able afterwards deny doing so.

Models for M-commerce

There are several protocols defined for secure ecommerce transactions and most famous are SSL & SET.

Security Socket Layer protocol [SSL]

SSL is the protocol that encodes the whole session among computers and provides the safe communication services on Internet. It is widely used. SSL protocol developed by Netscape Communications Corporation. The protocol is composed of two layers. At the lowest level, developed on top of some reliable transport protocols like TCP, there is the SSL Record protocol which receives non interpreted data from higher layers in non-empty blocks of arbitrary size. The SSL Record protocol is used for encapsulation of various higher level protocols like SSL Handshake Protocol. Handshake protocol allows the client and servers to authenticate each other and to negotiate an encryption algorithm with its associated cryptographic keys before the application protocol transmits or receives its first byte of data.

Advantage of SSL is that it is independent of an application protocol. A higher level protocol can be built on top of SSL protocol transparently. SSL allows traffic between web server and client to be strongly encrypted using public key technology. But SSL can't prevent personal information being stolen.

Secure Electronic Transaction

SET was incorporated by Visa and MasterCard with the participation from several technology companies including IBM and Microsoft. This system means that your entire credit card number is never travelling across the net, rather pieces of it are and no one sees the entire card number. SET supports electronic commerce security based on Certificate Authority (CA).

SET protocol includes a payment section which is able to deal with different credit cards, and it applies an acquirer payment gateway which is able to authorize the usage of existing networks. In the authorization request sent by the merchant to the acquirer, the purchase instruction of the customer enables the acquirer to verify that merchant and buyer agree as to what is purchased and how much is authorized. SET is a common secure electronic payment protocols with main five parties :-

i) Customer
ii) Seller
iii) Payment Gateway
iv) Certificate Authority
v) Issuer.