

# **CSE 320-Mobile Database Systems**

---

**Dr. R. B. Patel**

## **Mobile Database Systems**

---

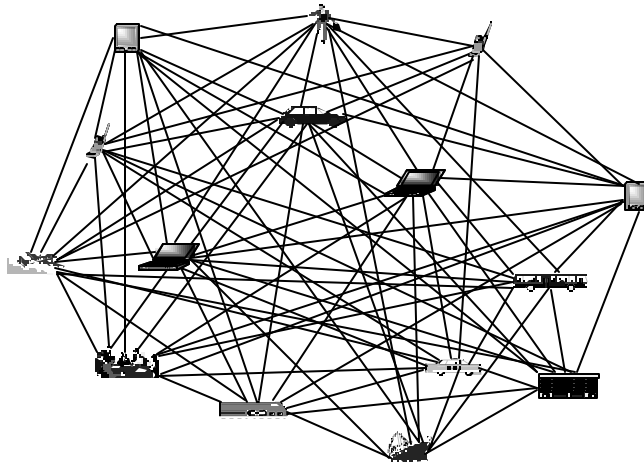
### **Outline**

- ❖ **Fully Connected Information Space**
- ❖ **Mobile Database Systems (MDS)**
- ❖ **Transaction Management**
- ❖ **Data Caching**
- ❖ **Query Processing**
- ❖ **Data Classification**
- ❖ **Conclusion**

## **Mobile Database Systems**

---

### **Fully connected information space**



## **Mobile Database Systems**

---

### **Fully connected information space**

- ❖ Each node of the information space has some communication capability.
- ❖ Some node can process information.
- ❖ Some node can communicate through voice channel.
- ❖ Some node can do both

## **Mobile Database Systems**

---

### **Fully connected information space**

Can be created and maintained by integrating legacy database systems, and wired and wireless systems (PCS, Cellular system, and GSM)

## **Mobile Database Systems**

---

### **What is a Mobile Database System (MDS)?**

A system with the following structural and functional properties

- ❖ Distributed system with *mobile connectivity*
- ❖ Full database system capability
- ❖ Complete spatial mobility
- ❖ Built on PCS/GSM platform
- ❖ Wireless and wired communication capability

## **Mobile Database Systems**

---

### **What is a mobile connectivity?**

A mode in which a client or a server can establish communication with each other whenever needed. *Intermittent connectivity* is a special case of mobile connectivity.

## **Mobile Database Systems**

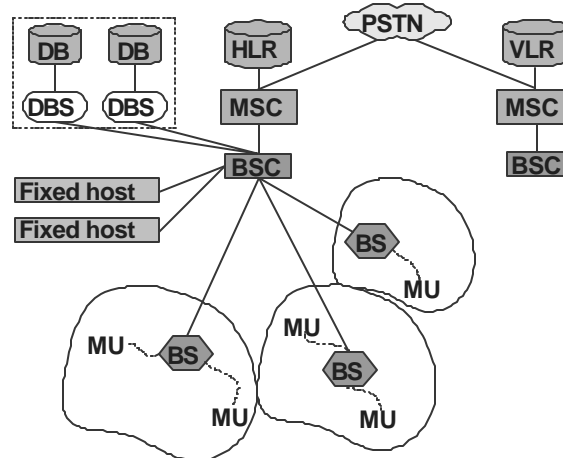
---

### **What is intermittent connectivity?**

A node in which only the client can establish communication whenever needed with the server but the server cannot do so.

## Mobile Database Systems (MDS)

### A Reference Architecture (Client-Server model)



## Mobile Database Systems (MDS)

### MDS Applications

- ❖ Insurance companies
- ❖ Emergencies services (Police, medical, etc.)
- ❖ Traffic control
- ❖ Taxi dispatch
- ❖ E-commerce
- ❖ Etc.

## **Mobile Database Systems (MDS)**

---

### **MDS Limitations**

- ❖ Limited wireless bandwidth
- ❖ Wireless communication speed
- ❖ Limited energy source (battery power)
- ❖ Less secured
- ❖ Vulnerable to physical activities
- ❖ Hard to make theft proof.

## **Mobile Database Systems (MDS)**

---

### **MDS capabilities**

- ❖ Can physically move around without affecting data availability
- ❖ Can reach to the place data is stored
- ❖ Can process special types of data efficiently
- ❖ Not subjected to connection restrictions
- ❖ Very high reachability
- ❖ Highly portable

## **Mobile Database Systems (MDS)**

---

### **Objective**

To build a truly ubiquitous information processing system by overcoming the inherent limitations of wireless architecture.

## **Mobile Database Systems (MDS)**

---

### **MDS Issues**

- ❖ **Data Management**
  - Data Caching
  - Data Broadcast (Broadcast disk)
  - Data Classification
- ❖ **Transaction Management**
  - Query processing
  - Transaction processing
  - Concurrency control
  - Database recovery

## **Mobile Database Systems (MDS)**

---

### **MDS Data Management Issues**

How to improve data availability to user queries using limited bandwidth?

#### **Possible schemes**

- ❖ **Semantic data caching:** The cache contents is decided by the results of earlier transactions or by *semantic data set*.
- ❖ **Data Broadcast on wireless channels**

## **Mobile Database Systems (MDS)**

---

### **MDS Data Management Issues**

How to improve data availability to user queries using limited bandwidth?

#### **Semantic caching**

- ❖ **Client maintains a semantic description of the data in its cache instead of maintaining a list of pages or tuples**
- ❖ **The server processes simple predicates on the database and the results are cached at the client.**



## **Mobile Database Systems (MDS)**

---

### **MDS Data Management Issues**

#### **Data Broadcast (Broadcast disk)**

A set of most frequently accessed data is made available by continuously broadcasting it on some fixed radio frequency. Mobile Units can tune to this frequency and download the desired data from the broadcast to their local cache.

A broadcast (file on the air) is similar to a disk file but located on the air.

## **Mobile Database Systems (MDS)**

---

### **MDS Data Management Issues**

#### **Data Broadcast (Broadcast disk)**

The contents of the broadcast reflects the data demands of mobile units. This can be achieved through data access history, which can be fed to the data broadcasting system.

For efficient access the broadcast file use index or some other method.

## **Mobile Database Systems (MDS)**

---

### **MDS Data Management Issues**

How MDS looks at the database data?

Data classification

- ❖ Location Dependent Data (LDD)
- ❖ Location Independent Data (LID)

## **Mobile Database Systems (MDS)**

---

### **MDS Data Management Issues**

Location Dependent Data (LDD)

The class of data whose value is functionally dependent on location. Thus, the value of the location determines the correct value of the data.

Location — Data value

Examples: City tax, City area, etc.

## **Mobile Database Systems (MDS)**

---

### **MDS Data Management Issues**

#### **Location Independent Data (LID)**

The class of data whose value is functionally independent of location. Thus, the value of the location does not determine the value of the data.

Example: Person name, account number, etc.  
The person name remains the same irrespective of place the person is residing at the time of enquiry.

## **Mobile Database Systems (MDS)**

---

### **MDS Data Management Issues**

#### **Location Dependent Data (LDD)**

Example: Hotel Taj has many branches in India. However, the room rent of this hotel will depend upon the place it is located. Any change in the room rate of one branch would not affect any other branch.

Schema: It remains the same only multiple correct values exists in the database.

## **Mobile Database Systems (MDS)**

---

### **MDS Data Management Issues**

#### **Location Dependent Data (LDD)**

LDD must be processed under the location constraints. Thus, the tax data of Pune can be processed correctly only under Pune's finance rule.

Needs *location binding* or *location mapping* function.

## **Mobile Database Systems (MDS)**

---

### **MDS Data Management Issues**

#### **Location Dependent Data (LDD)**

*Location binding* or *location mapping* can be achieved through database schema or through a location mapping table.

## Mobile Database Systems (MDS)

### MDS Data Management Issues

#### Location Dependent Data (LDD) Distribution

MDS could be a federated or a multidatabase system. The database distribution (replication, partition, etc.) must take into consideration LDD.

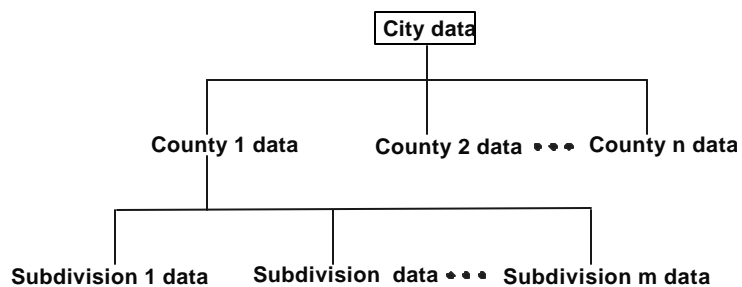
One approach is to represent a city in terms of a number of mobile cells, which is referred to as “Data region”. Thus, Pune can be represented in terms of  $N$  cells and the LDD of Pune can be replicated at these individual cells.

## Mobile Database Systems (MDS)

### MDS Data Management Issues

#### Concept Hierarchy in LDD

In a data region the entire LDD of that location can be represented in a hierarchical fashion.



## **Mobile Database Systems (MDS)**

---

### **MDS Query processing**

#### **Query types**

- ❖ **Location dependent query**
- ❖ **Location aware query**
- ❖ **Location independent query**

## **Mobile Database Systems (MDS)**

---

### **MDS Query processing**

#### **Location dependent query**

**A query whose result depends on the geographical location of the origin of the query.**

#### **Example**

**What is the distance of Pune railway station from here?**

**The result of this query is correct only for “here”.**

## **Mobile Database Systems (MDS)**

---

### **MDS Query processing**

#### **Location dependent query**

**Situation:** Person traveling in the car desires to know his progress and continuously asks the same question. However, every time the answer is different but correct.

**Requirements:** Continuous monitoring of the longitude and latitude of the origin of the query. GPS can do this.

## **Mobile Database Systems (MDS)**

---

### **MDS Transaction Management**

**Transaction properties ACID** (Atomicity, Consistency, Isolation, and Durability).

**Too rigid for MDS.** Flexibility can be introduced using workflow concept. Thus, a part of the transaction can be executed and committed independent to its other parts.

# Transaction

A computation processing is considered as a transaction or conventional transaction if it satisfies ACID (Atomicity, Consistency, Isolation, and Durability) properties:

- **Atomicity:** 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 the initial state after some execution steps then it is aborted or rollback.
- **Consistency:** if a program produces consistent result only then it satisfies the consistency property and it will be at the final state or committed. If the result is not consistent then a transaction program should be at the initial state, in other word the transaction is aborted.

## *Transaction (Contd.)*

- **Isolation:** if a program is executing and if it is only single program on the system then it satisfies the isolation property. If there are several other processes on the system, then none of the intermediate state of this program is viewable until it reaches its final state.
- **Durability:** if a program reaches to its final state and the result is made available to the outside world then this result is made permanent. Even a system failure cannot change this result. In other words, when a transaction commits its state is durable.



## ***Transaction (Contd.)***

- The programming model of a transaction will be:

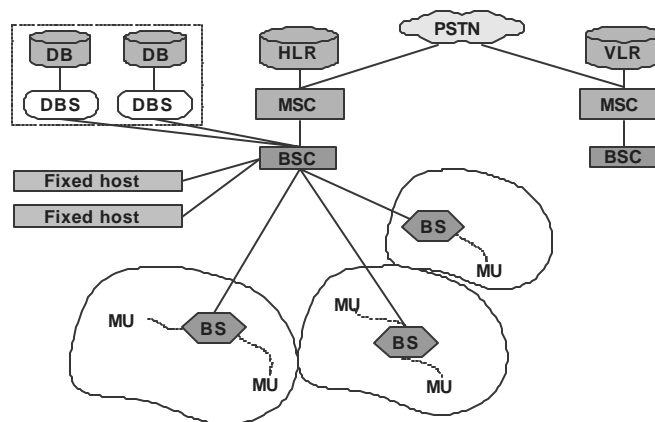
```
Begin_transaction ()  
  Execution of transaction program  
■ If (reach_final_state) then  
  Commit_Work(final_state)  
■ Else  
  Rollback_Work(initial_state)
```

The simplest form of transaction is flat transaction. A flat transaction can be considered as a sequential correctness computer program. Every execution step is after one another.

## **Mobile Database Systems (MDS)**

### **MDS Transaction Management**

Transaction fragments for distribution.



## Mobile Database Systems (MDS)

### MDS Transaction Management

Transaction fragments for distributed execution

Execution scenario: User issues transactions from his/her MU and the final results comes back to the same MU. The user transaction may not be completely executed at the MU so it is fragmented and distributed among database servers for execution. This creates a Distributed mobile execution.

## Mobile Database Systems (MDS)

### MDS Transaction Management

A mobile transaction (MT) can be defined as

$T_i$  is a triple  $\langle F, L, FLM \rangle$ ; where

$F = \{e_1, e_2, \dots, e_n\}$  is a set of execution fragments,

$L = \{l_1, l_2, \dots, l_n\}$  is a set of locations, and

$FLM = \{flm_1, flm_2, \dots, flm_n\}$  is a set of fragment location mapping where  $flm_i(e_i) = l_i$

## Mobile Database Systems (MDS)

### MDS Transaction Management

An execution fragment  $e_{ij}$  is a partial order  $e_{ij} = \{s_j, \ell_j\}$  where

- ❖  $s_i = OS_j \hat{E} \{N_i\}$  where  $OS_j = \hat{E}_k O_{jk}, O_{jk} \hat{I} \{read, write\}$ , and  $N_j \{Abort_L, Commit_L\}$ .
- ❖ For any  $O_{jk}$  and  $O_{jl}$  where  $O_{jk} = R(x)$  and  $O_{jl} = W(x)$  for data object  $x$ , then either  $O_{jk} \ell_j O_{jl}$  or  $O_{jl} \ell_j O_{jk}$ .

## Mobile Database Systems (MDS)

### MDS Transaction Management

#### Mobile Transaction Models

**Kangaroo Transaction:** It is requested at a MU but processed at DBMS on the fixed network. The management of the transaction moves with MU. Each transaction is divided into subtransactions. Two types of processing modes are allowed, one ensuring overall atomicity by requiring compensating transactions at the subtransaction level.

## **Mobile Database Systems (MDS)**

---

### **MDS Transaction Management**

#### **Mobile Transaction Models**

**Reporting and Co-Transactions:** The parent transaction (workflow) is represented in terms of reporting and co-transactions which can execute anywhere. A reporting transaction can share its partial results with the parent transaction anytime and can commit independently. A co-transaction is a special class of reporting transaction, which can be forced to wait by other transaction.

## **Mobile Database Systems (MDS)**

---

### **MDS Transaction Management**

#### **Mobile Transaction Models**

**Clustering:** A mobile transaction is decomposed into a set of weak and strict transactions. The decomposition is done based on the consistency requirement. The read and write operations are also classified as weak and strict.

## Mobile Database Systems (MDS)

### MDS Transaction Management

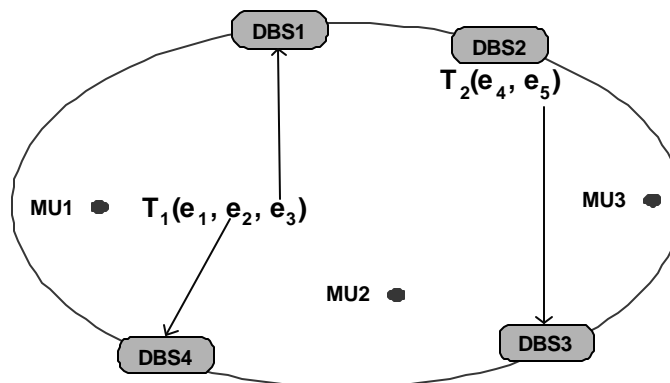
#### Mobile Transaction Models

**Semantics Based:** The model assumes a mobile transaction to be a long lived task and splits large and complex objects into smaller manageable fragments. These fragments are put together again by the merge operation at the server. If the fragments can be recombined in any order then the objects are termed *reorderable* objects.

## Mobile Database Systems (MDS)

### MDS Transaction Management

#### Mobile Transaction execution.



## **Mobile Database Systems (MDS)**

---

### **MDS Transaction Management**

Serialization of concurrent execution.

- ❖ Two-phase locking based (commonly used)
- ❖ Timestamping
- ❖ Optimistic

Reasons these methods may not work satisfactorily

- ❖ Wired and wireless message overhead.
- ❖ Hard to efficiently support disconnected operations.
- ❖ Hard to manage locking and unlocking operations.

## **Mobile Database Systems (MDS)**

---

### **MDS Transaction Management**

Serialization of concurrent execution.

New schemes based on timeout, multiversion, etc., may work. A scheme, which uses minimum number of messages, especially wireless messages is required.

## **Mobile Database Systems (MDS)**

---

### **MDS Transaction Management**

Database update to maintain global consistency.

Database update problem arises when mobile units are also allowed to modify the database. To maintain global consistency an efficient database update scheme is necessary.

## **Mobile Database Systems (MDS)**

---

### **MDS Transaction Management**

Transaction commit.

In MDS a transaction may be fragmented and may run at more than one nodes (MU and DBSs). An efficient commit protocol is necessary. 2phase commit (2PC) or 3-phase commit (3PC) is no good because of their generous messaging requirement. A scheme which uses very few messages, especially wireless, is desirable.

## **Mobile Database Systems (MDS)**

---

### **MDS Transaction Management**

**Transaction commit.**

One possible scheme is “timeout” based protocol.

Concept: MU and DBSs guarantee to complete the execution of their fragments of a mobile transaction within their predefined timeouts. Thus, during processing no communication is required. At the end of timeout, each node commit their fragment independently.

## **Mobile Database Systems (MDS)**

---

### **MDS Transaction Management**

**Transaction commit.**

**Protocol: TCOT-Transaction Commit On Timeout**

**Requirements**

**Coordinator: Coordinates transaction commit**

**Home MU: Mobile Transaction (MT) originates here**

**Commit set: Nodes that process MT (MU + DBSs)**

**Timeout: Time period for executing a fragment**



## **Mobile Database Systems (MDS)**

---

### **MDS Transaction Management**

**Protocol: TCOT-Transaction Commit On Timeout**

- ❖ MT arrives at Home MU.
- ❖ MU extract its fragment, estimates timeout, and send rest of MT to the coordinator.
- ❖ Coordinator further fragments the MT and distributes them to members of commit set.
- ❖ MU processes and commits its fragment and sends the updates to the coordinator for DBS.
- ❖ DBSs process their fragments and inform the coordinator.
- ❖ Coordinators commits or aborts MT.

## **Mobile Database Systems (MDS)**

---

### **MDS Transaction Management**

**Transaction and database recovery.**

**Complex for the following reasons**

- ❖ Some of the processing nodes are mobile
- ❖ Less resilient to physical use/abuse
- ❖ Limited wireless channels
- ❖ Limited power supply
- ❖ Disconnected processing capability

## **Mobile Database Systems (MDS)**

---

### **MDS Transaction Management**

Transaction and database recovery.

Desirable recovery features

- ❖ Independent recovery capability
- ❖ Efficient logging and checkpointing facility
- ❖ Log duplication facility

## **Mobile Database Systems (MDS)**

---

### **MDS Transaction Management**

Transaction and database recovery.

- ❖ Independent recovery capability reduces communication overhead. Thus, MUs can recover without any help from DBS
- ❖ Efficient logging and checkpointing facility conserve battery power
- ❖ Log duplication facility improves reliability of recovery scheme

## **Mobile Database Systems (MDS)**

---

### **MDS Transaction Management**

Transaction and database recovery.

#### **Possible approaches**

- ❖ Partial recovery capability
- ❖ Use of mobile agent technology

## **Mobile Database Systems (MDS)**

---

### **MDS Transaction Management**

Transaction and database recovery.

#### **Possible MU logging approaches**

- ❖ Logging at the processing node (e.g., MU)
- ❖ Logging at a centralized location (e.g., at a designated DBS)
- ❖ Logging at the place of registration (e.g., BS)
- ❖ Saving log on Zip drive or floppies.

## **Mobile Database Systems (MDS)**

---

### **Mobile Agent Technology**

A mobile agent is an independent software module capable of

- ❖ Migrating to any node on the network
- ❖ Capable of spawning and eliminating itself
- ❖ Capable of recording its own history

## **Mobile Database Systems (MDS)**

---

### **Mobile Agent Technology**

A mobile agent can be used for the following activities, which are essential for recovery.

- ❖ Centralized and distributed logging
- ❖ Log carrier. A Mobile unit may need to carry its log with it for independent recovery
- ❖ Log processing for database recovery
- ❖ Transaction commit or abort

## **Mobile Database Systems (MDS)**

---

### **Mobile Agent Technology**

#### **Possible approaches**

- ❖ Agent broadcast on a dedicated wireless channel
- ❖ Pool of agents at every processing node
- ❖ Agent migration to a required node.

## **Mobile Database Systems (MDS)**

---

### **summary**

Wireless network is becoming a commonly used communication platform. It provides a cheaper way to get connected and in some cases this is the only way to reach people. However, it has a number of easy and difficult problems and they must be solved before MDS can be built. This tutorial discussed some of these problems and identified a number of possible approaches.

## **Mobile Database Systems (MDS)**

---

### **summary**

The emerging trend is to make all service providing disciplines, such as web, E-commerce, workflow systems, etc., fully mobile so that any service can be provided from any place. Customer can surf the information space from any location at any time and do their shopping, make flight reservation, open bank account, attend lectures, and so on. This is what the wireless technology driving us to.