MOBILE COMPUTING

e-NOTES

B.Sc. Computer Science / B.C.A

Dr.P.Rizwan Ahmed
Vice Principal (Academic) & HOD

Department of Computer Applications &
PG Department of Information Technology

Mazharul Uloom College, Ambur-635802
(Managed by Ambur Muslim Educational Society (AMES),
Affiliated to Thiruvalluvar University, Vellore)

Unit – III Mobile Computing

Unit –III Important Questions(5 marks & 10 marks)

- **Requirement of Mobile IP**
- **Routing**
- **DSVR**
- **DSR**
- **Mobile TCP**
- **Dynamic Host Configuration Protocol (DHCP)**

Mobile IP

Mobile IP is a network layer solution for homogenous and heterogeneous mobility on the global Internet which is scalable, robust, and secure and which allows nodes to maintain all ongoing communications while moving.

Need for Mobile IP

The IP addresses are designed to work with stationary hosts because part of the address defines the network to which the host is attached. A host cannot change its IP address without terminating on-going sessions and restarting them after it acquires a new address. Other link layer mobility solutions exist but are not sufficient enough for the global Internet.

- ***Mobility*** is the ability of a node to change its point-of-attachment while maintaining all existing communications and using the same IP address.
- ***Nomadicity*** allows a node to move but it must terminate all existing communications and then can initiate new connections with a new address.
- It has been foreseen that mobile computing devices will become more pervasive, more useful, and more powerful in the future.
- The power and usefulness will come from being able to extend and integrate the functionality of all types of communication such as Web browsing, e -mail, phone calls, information retrieval, and perhaps even video transmission.
- For Mobile IP computing to become as pervasive as stationary IP networks of the world, a ubiquitous protocol for the integration of voice, video, and data must be developed.
- The most widely researched and developed protocol is Mobile IP.

Components of Mobile IP

Three main components of Mobile IP

1. ***Discovering the care-of address:*** mobile node uses discovery procedure to identify prospective home and foreign agents.
2. ***Registering the care-of address***: mobile node uses authenticated registration procedure to inform home agent of its care-of address.
3. ***Tunneling the care-of address***: used to forward IP datagrams from a home address to a care-of address

Requirements to Mobile IP / Characteristics of Mobile IP

There are several requirements accompanied the development of the standard:

Transparency

- Mobile end-systems keep their IP address.
- Continuation of communication after interruption of link possible.
- Point of connection to the fixed network can be changed.

Compatibility

- Support of the same layer 2 protocols as IP.
- No changes to current end-systems and routers required.
- Mobile end-systems can communicate with fixed systems

Security

- All messages used to transmit information to another node about the location of a mobile node must be authenticated to protect against remote redirection attacks.

Efficiency and scalability

- Only little additional messages to the mobile system required (connection typically via a low bandwidth radio link).
- World-Wide support of a large number of mobile systems in the whole Internet.

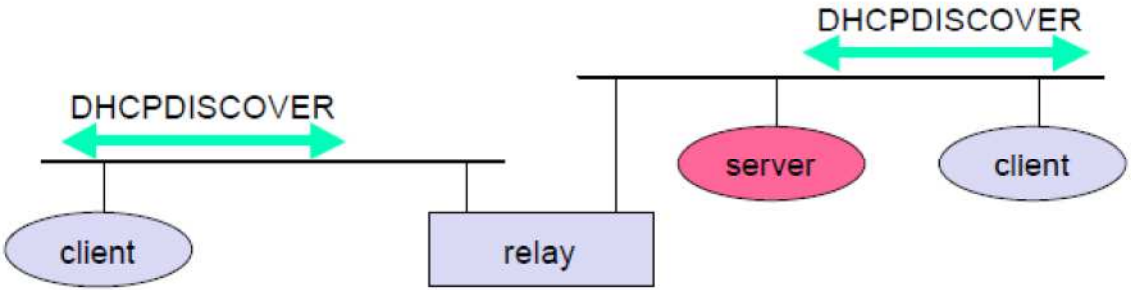
Dynamic Host Configuration Protocol (DHCP)

DIICP was developed based on bootstrap protocol (BOOTP). DIICP is an automatic configuration protocol used on IP networks. DHCP allows a computer to join an IP-based network without having a pre-configured IP address. DHCP is a protocol that assigns unique IP addresses to devices, then releases and renews these addresses as devices leave and re-join the network.

If a new computer is connected to a network, DHCP can provide it with all the necessary information for full system integration into the network, e.g., addresses of a DNS server and the default router, the subnet mask, the domain name, and an IP address. Providing an IP address makes DHCP very attractive for mobile IP as a source of care-of-addresses.

Client/Server Model

The client sends via a MAC broad a request to the DHCP server (might be via a DHCP relay)



Consider the scenario where there is one client and two servers are present. A typical initialization of a DHCP client is shown below:

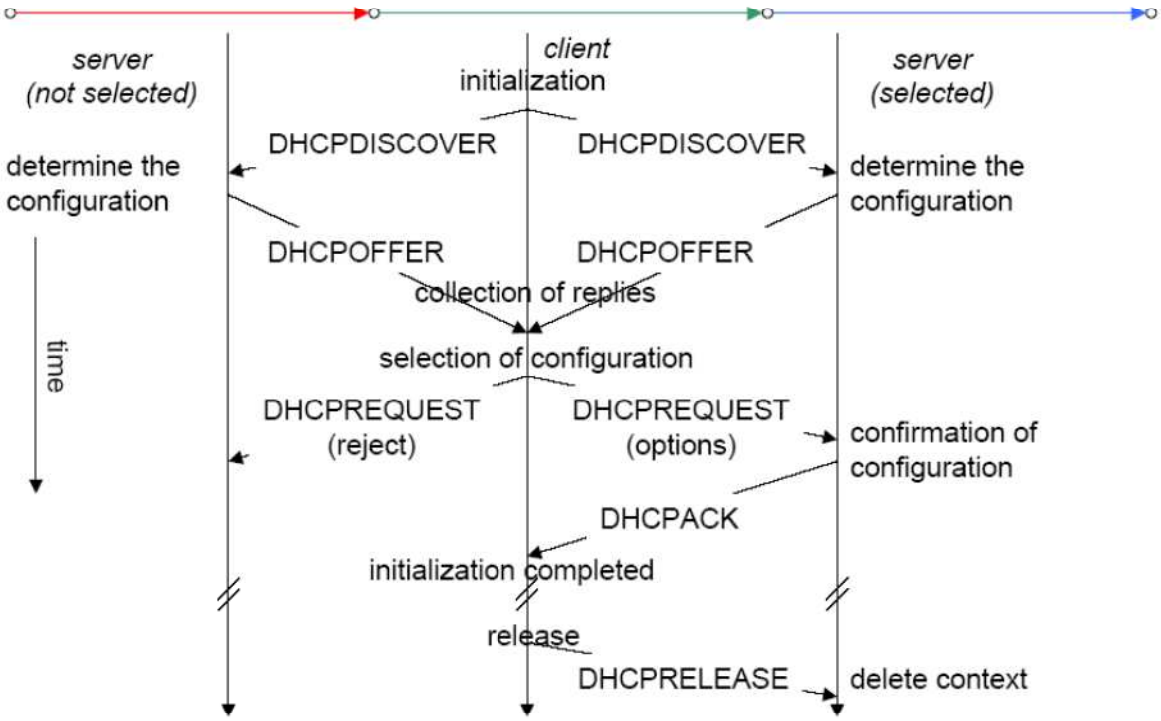


Figure – DHCP protocol mechanism

Routing

- A destination node might be out of range of a source node transmitting packets. Thus routing is needed to find a path between source and destination and to forward the packets appropriately.

- In wireless networks using an infrastructure cells have been defined and the base station can reach all mobile nodes without routing.
- In the case of ad-hoc networks each node must be able to forward data for other nodes.
- At a certain time t1 the network topology might look as shown below.
- Five nodes N1 to N5 are connected depending on the current transmission characteristics between them.
- N4 can receive N1 over a good link but N1receives N4 only via a weak link.
- Thus links do not necessarily have the same characteristics in both directions.
- In t2, N1 cannot receive N4 any longer, N4 receives N1 only via a weak link. N1 has an asymmetric but bi-directional link to N2,that did not exist before.

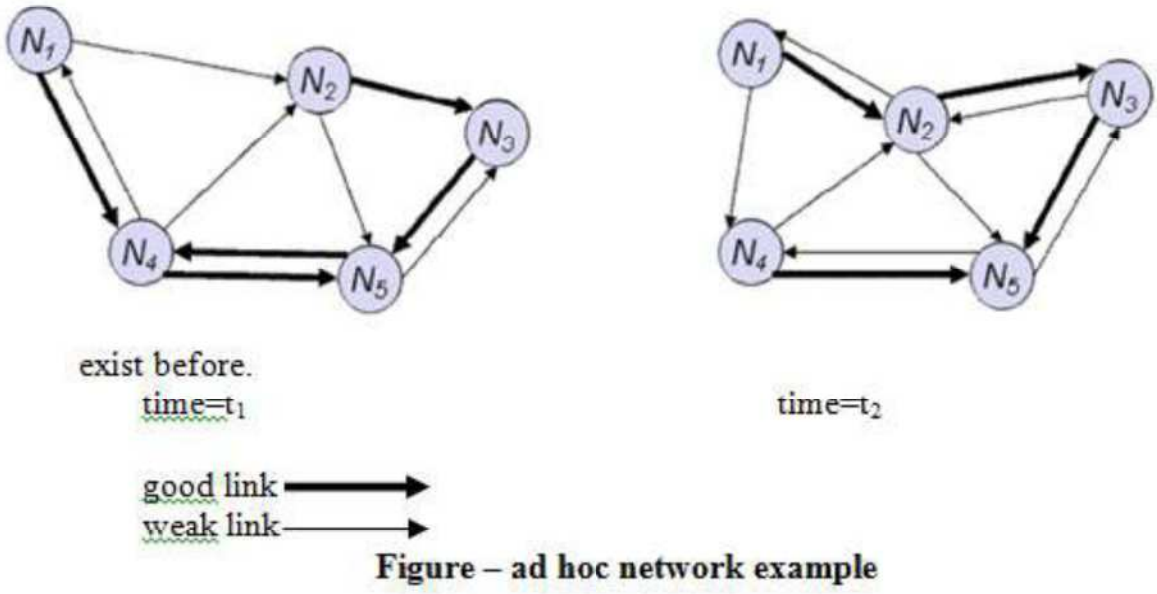


Figure – ad hoc network example

Destination Sequence Distance Vector (DSDV)

- DSDV routing is an enhancement to distance vector routing for ad-hoc networks.
- Distance vector routing performs poorly with certain network changes due to count to infinity problems.

DSDV adds two things to Distance Vector Algorithm:

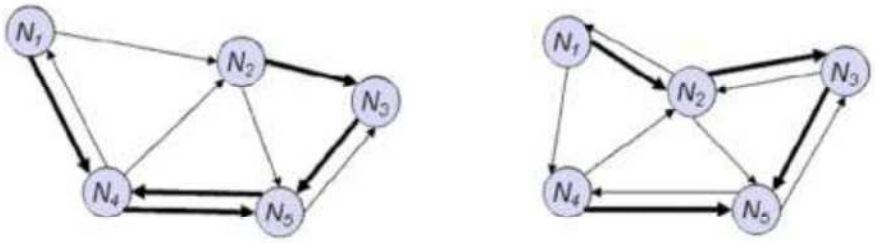
Sequence Number

- Each routing advertisement comes with a sequence number and in ad-hoc networks advertisements may propagate along many paths.
- Apply the advertisements in correct order.

Damping

- Transient changes in topology that are of short duration should not destabilize the routing mechanisms.

Example



The routing table for N1 is shown below:

Destination	Next hop	Metric	Sequence no	Install time
N_1	N_1	0	$S_1 = 321$	T_1
N_2	N_2	1	$S_2 = 218$	T_2
N_3	N_3	2	$S_3 = 043$	T_3
N_4	N_4	1	$S_4 = 092$	T_4
N_5	N_5	2	$S_5 = 163$	T_5

Dynamic Source Routing

Dynamic Source Routing **s**plit routing into discovering a path and maintaining a path. Dynamic Source routing divides the task of routing into two separate problems.

Discover a path

- Only if a path for sending packets to a certain destination is needed and no path is currently available.

Maintaining a path

- Only while the path is in use one has to make sure that it can be used continuously.

Dynamic source routing eliminates all periodic routing updates and works as follows:

- If a node needs to discover a route, it broadcasts a route request with a unique identifier and destination address as parameter. Any node that receives a route request does the following.
- If the node has already received the request it drops the request packet.
- If the node recognizes its own address as the destination the request has reached its target.
- Otherwise the node appends its own address to a list of traversed hops in the packet and broadcasts this
- Updated route request.

Applying route discovery for a route from N1 to N3 at time t1, results in following

- N1 broadcasts the request and N2 and N4 receive this request.
- N2 then broadcasts and N4 broadcasts, N3 and N5 receive N2's broadcast,
- N1,N2 and N5 receives N4's broadcast.
- N3 recognizes itself as target,N5 broadcasts and N3,N4 receives the broadcast.
- N4 drops N5 broadcast and N3 recognizes as an alternate but longer route.
- N3 now has to return the path to N1. This is simple assuming symmetric links working in both directions.

The basic algorithm for route discovery can be optimized in many ways:

- To avoid too many broadcasts, each route request could contain a counter. Every node rebroadcasting the request increments the counter by one.
- Knowing the maximum network diameter nodes can drop a request if the counter reaches this number.
- A node can cache path fragments from recent requests. These fragments can now be used to answer other route requests much faster.
- A node can also update this cache from packet headers while forwarding other packets.
- If a node overhears transmissions from other nodes, it can also use this information for shortening routes.
- After discovering a route, it has to be maintained as long as node sends packets along this route.

Alternative Metrics

Interference-based routing

Routing based on assumptions about interference between signals.

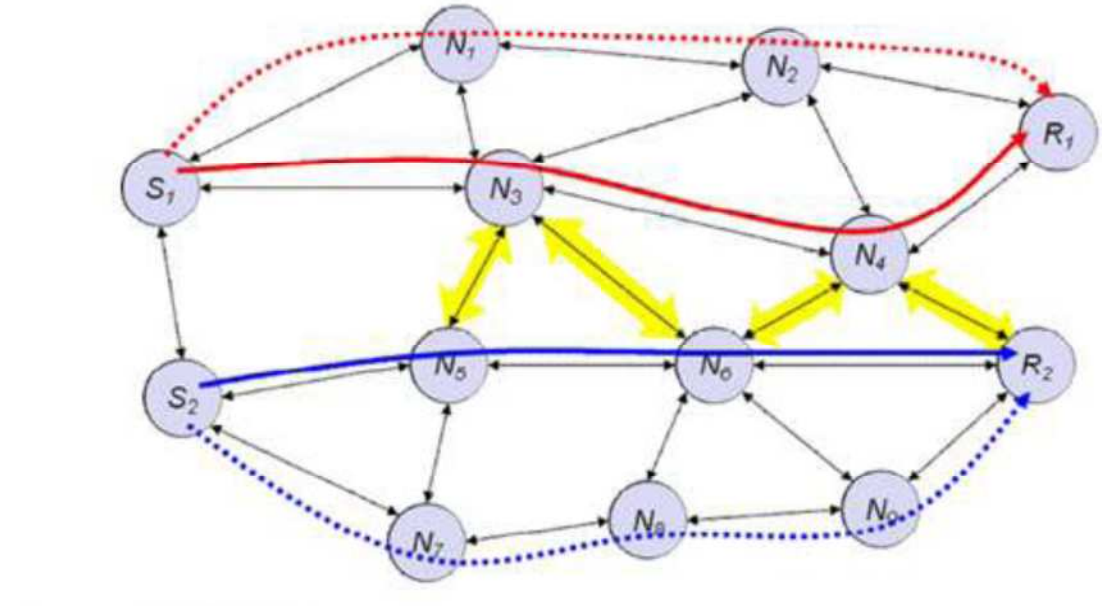


Figure – Example of LIR

Figure - Examples for interference based routing

(LIR)

- Calculate the cost of a path based on the number of stations that can receive a transmission.

Max-Min Residual Capacity Routing (MMRCR)

- Calculate the cost of a path based on a probability function of successful transmissions and interference.

Least Resistance Routing (LRR)

- Calculate the cost of a path based on interference, jamming and other transmissions.
- LIR is very simple to implement, only information from direct neighbors is necessary.

Mobile TCP

Both I-TCP and Snooping TCP does not help much, if a mobile host gets disconnected. The **M-TCP (mobile TCP)** approach has the same goals as I-TCP and snooping TCP: to prevent the sender window from shrinking if bit errors or disconnection but not congestion cause current problems. M-TCP wants to improve overall throughput, to lower the delay, to maintain end-to-end semantics of TCP, and to provide a more efficient handover. Additionally, M-TCP is especially adapted to the problems arising from lengthy or frequent disconnections. M-TCP splits the TCP connection into two parts as I-TCP does. An unmodified TCP is used on the standard host-**supervisory host (SH)** connection, while an optimized TCP is used on the SH-MH connection.

The SH monitors all packets sent to the MH and ACKs returned from the MH. If the SH does not receive an ACK for some time, it assumes that the MH is disconnected. It then chokes the sender by setting the sender's window size to 0. Setting the window size to 0 forces the sender to go into **persistent mode**, i.e., the state of the sender will not change no matter how long the receiver is disconnected. This means that the sender will not try to retransmit data. As soon as the SH (either the old SH or a new SH) detects connectivity again, it reopens the window of the sender to the old value. The sender can continue sending at full speed. This mechanism does not require changes to the sender's TCP. The wireless side uses an adapted TCP that can recover from packet loss much faster. This modified TCP does not use slow start, thus, M-TCP needs a **bandwidth manager** to implement fair sharing over the wireless link.

Advantages of M-TCP

- Maintains semantics
- Supports disconnection
- No buffer forwarding

Disadvantages of M-TCP

- Loss on wireless link propagated into fixed network.
- Adapted TCP on wireless link.

-------&&&--------

Unit IV Mobile Computing

WAP Model

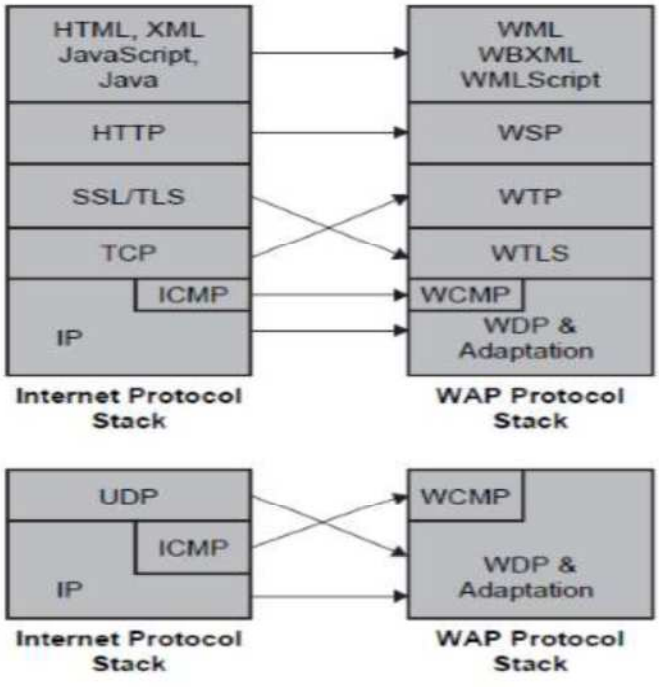
a. WAP is a set of protocols that allow wireless devices like hand-held cell phones to access the internet. But it has programming model similar to that of WWW.

b. WAP content and applications are specified in a set of well-known content formats based on WWW content formats

c. Transport of content is based on standard communication protocols which are based on that of WWW. The micro browser is analogous to standard web browser.

d. Wherever possible, existing standards are adopted, there are also some extensions to match characteristics of wireless environment.

e. This has provided benefits to application developers and ability to use existing tools (eg. XML tools).



WML: Wireless Markup Language
WSP: Wireless Session Protocol
WTP: Wireless Transport Protocol
WTLS: Wireless Transport Layer Security
WCMP: Wireless Control Management Protocol
WDP: Wireless Datagram Protocol

2. Standard components of WAP programming model

i. Standard naming model: WWW standard URLs identify WAP content on origin servers and local resources in a device (eg. Call control functions)

ii. Content typing: all WAP content has a specific type consistent with WWW content types.

iii. Standard content formats: WAP content formats are based on WWW technology and include display markup, calendar information, electronic business card objects, images and scripting language.

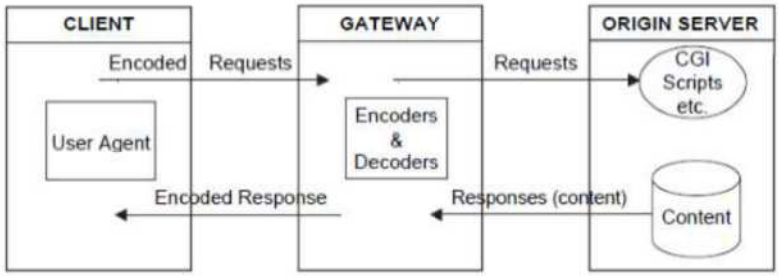
iv. Standard protocols: the WAP communication protocols and content types are optimized for mass market, hand-held wireless devices. The protocols enable the communication of browser requests from mobile terminal to network web server.

3. Functionality of WAP proxy: WAP utilizes proxy technology to connect between wireless domain and WWW.

i. Protocol gateway: it translates requests from WAP protocol stack to WWW protocol stack.

ii. Content encoders and decoders: the encoders translate the content into compact encoded formats to reduce the size of data. Thus, mobile users can browse wide variety of WAP content. Also, application author is able to build applications that run on large number of mobile devices.

iii. WAP proxy allows content and applications to be hosted on WWW servers and to be build using WWW technologies like 'Cell global identity (CGI)' scripting.



Mobile Location based services

Location-based services (LBS) use real-time geo-data from a mobile device or smartphone to provide information, entertainment or security. Some services allow consumers to "check in" at restaurants, coffee shops, stores, concerts, and other places or events. Often, businesses offer a reward — prizes, coupons or discounts — to people who check in. Google Maps, Foursquare, GetGlue, Yelp and Facebook Places are among the more popular services.

Location-based services use a smartphone's GPS technology to track a person's location, if that person has opted-in to allow the service to do that. After a smartphone user opts-in, the service can identify his or her location down to a street address without the need for manual data entry.

Uses of location-based services

Companies have found several ways to use a device's location:

- **Store locators.** Using location-based intelligence, retail customers can quickly find the nearest store location.
- **Proximity-based marketing.** Local companies can push ads only to individuals within the same geographic location. Location-based mobile marketing delivers ads to potential customers within that city who might actually act on the information.
- **Travel information.** An LBS can deliver real-time information, such as traffic updates or weather reports, to the smartphone so the user can plan accordingly.
- **Roadside assistance.** In the event of a blown tire or accident, many roadside assistance companies provide an app that allows them to track your exact location without the need for giving directions.
- **Mobile workforce management.** For logistics-dependent companies that employ individuals out in the field or at multiple locations, an LBS allows employees to check in at a location using their mobile device.
- **Fraud prevention.** An LBS creates another level of security by matching a customer's location through the smartphone to a credit card transaction. Tying the smartphone's location to a credit card allows you to flag transactions made across several geographic locations over a short time.

WAP Gateway

WAP gateway is a software system that helps WAP-enabled wireless devices to communicate to Internet Web sites and applications. Web sites deliver pages in special format called Wireless Markup Language (WML) that is compiled and forwarded by the WAP gateway. In order to access Internet resources from a WAP-enabled wireless device you need a WAP gateway service.

A WAP gateway sits between mobile devices using the Wireless Application Protocol (WAP) and the World Wide Web, passing pages from one to the other much like a proxy. This translates pages into a form suitable for the mobiles, for instance using the Wireless Markup Language (WML). This process is hidden from the phone, so it may access the page in the same way as a browser accesses HTML, using a URL (for example, http://example.com/foo.wml), provided the mobile phone operator has not specifically prevented this. WAP gateway software that encodes and decodes request and response between the smartphones, micro browser and internet. It decodes the encoded WAP requests from the micro browser and send the HTTP requests to the internet or to a local application server. It also encodes the WML and HDML data returning from the web for transmission to the micro browser in the handset.

WAP protocols

WAP protocols consist of Wireless Datagram Protocol (WDP), Wireless Transport Layer Security (WTLS), Wireless Transaction Protocol (WTP), Wireless Session Protocol (WSP) and Wireless Application Environment (WAE)

WDP

The WAP Datagram Protocol (WDP) is the transport layer that sends and receives messages
via any available bearer network, including SMS, USSD, CSD, CDPD, IS–136 packet

data, and GPRS.
The Transport layer protocol in the WAP architecture is the Wireless Datagram Protocol
(WDP). The WDP layer operates above the data capable bearer services aided by various network types.

WTLS
Wireless Transport Layer Security (WTLS), an optional security layer, has encryption facilities that provide the secure transport service required by many applications, such as e-commerce.
WTLS is a security protocol based upon the industry standard Transport Layer Security (TLS) protocol, formerly known as Secure Sockets Layer (SSL). WTLS is intended for use with the WAP transport protocol and has been optimized for use over narrow-band communications channels.

WTP
The WAP Transaction Protocol (WTP) layer provides transaction support, adding reliability to the datagram service provided by WDP.
The Wireless Transaction Protocol (WTP) runs on top of a datagram service and provides a lightweight transaction-oriented protocol that is suitable for implementation in „thin‰ clients (mobile stations). WTP operates efficiently over
secure or non-secure wireless datagram networks

WSP
The WAP Session Protocol (WSP) layer provides a lightweight session layer to allow efficient exchange of data between applications.
The Wireless Session Protocol (WSP) offers the application layer of WAP with a consistent interface for two session services.
- The first is a connection oriented service that operates above the transaction layer protocol WTP
- The second is a connectionless service that operates above a secure or nonsecure datagram service (WDP)

WAE
The **Wireless Application Environment (WAE)** is a general-purpose application environment based on a mixture of the World Wide Web (WWW) and Mobile Telephony technologies. The primary objective of the WAE effort is to establish an interoperable environment that will allow operators and service providers to build applications and services that can catch a wide variety of different wireless platforms in an efficient and useful manner.

WAP user agent profile

Concept

The schema for WAP User Agent Profiles consists of description blocks for the following key components:

HardwarePlatform: A collection of properties that adequately describe the hardware characteristics of the terminal device. This includes, the type of device, model number, display size, input and output methods, etc.

SoftwarePlatform: A collection of attributes associated with the operating environment of the device. Attributes provide information on the operating system software, video and audio encoders supported by the device, and user s preference on language .

BrowserUA: A set of attributes to describe the HTML browser application **NetworkCharacteristics:**Information about the network-related infrastructure and environment such as bearer information. These attributes can influence the resulting content, due to the variation in capabilities and characteristics of various network infrastructures in terms of bandwidth and device accessibility.

WapCharacteristics: A set of attributes pertaining to WAP capabilities supported on the device. This includes details on the capabilities and characteristics related to the WML Browser, WTA [WTA], etc.

Significance

Faster when compared to all other user agent profiles.

Caching model

Caching can reduce the bandwidth requirement in a mobile computing environment. However, due to battery power limitations, a wireless mobile computer may often be forced to operate in a doze (or even totally disconnected) mode. As a result, the mobile computer may miss some cache invalidation reports broadcast by a server, forcing it to discard the entire cache contents after waking up. In this paper, we present an energy-efficient cache invalidation method, called GCORE (Grouping with COld update-set REtention), that allows a mobile computer to operate in a disconnected mode to save the battery while still retaining most of the caching benefits after a reconnection. We present an efficient implementation of GCORE and conduct simulations to evaluate its caching effectiveness. The results show that GCORE can substantially improve mobile caching by reducing the communication bandwidth (or energy consumption) for query processing.

While a WSP session is established (whether active or suspended), the WAP gateway caches all Profile and Profile- Diff headers associated with that session. A third party host may issue a request for this CPI to, for instance, generate content that wil subsequently be pushed to the client device. The request is initiated from the third-party host and delivered to a Push Protocol Gateway (PPG). This specification defines neither a protocol for issuing this request nor a means for addressing the requested information. However, it is expected that such requests will typically be made to the WAP gateway using HTTP, as suggested by [WAP-PAP]. Upon

receiving the profile request, the gateway accesses the cached Profile and Profile-Diff headers and resolves them to form a complete CPI profile. It responds to the CPI request with the resolved profile (a CC/PP document) using a MIME type of **text/xml**.

WML

WML is an acronym for Wireless Markup Language. Wireless Markup Language (WML)is an XML(mark up language) language used to specify content **and user interface for WAP devices like PDA and Mobile Phones.** (WML), based on XML . It provides navigational support, data input, hyperlinks, text and image presentation, and forms, much like HTML(HyperText Markup Language).

While the HTML language creates web pages for the PCs, the WML creates web pages for the handheld devices. WMLÊs similarity to HTML was not random. The structure, formatting, and syntax are immediately recognizable to those familiar with HTML and XML. WML is a direct descendent of Handheld Device Markup Language (HDML). WML, however, has been optimized for the constraint wireless device. As a result, there is a prominent difference between HTML and WML. WML requires a micro browser to interpret the various commands necessary to render a document, or „deck‰ as WML documents are called. These browsers are usually embedded in the mobile device.

WML Script

WML uses WMLScript to run simple code on the client. WMLScript is a light JavaScript language. However, WML scripts are not embedded in the WML pages. WML pages only contains references to script URLs. WML scripts need to be compiled into byte code on a server before they can run in a WAP browser.

WML Script resides within the high-level applications layer of the WAP structure and adds capabilities for swifter, more advanced and interactive logical operations between the mobile device and server. WML Script code can be written in normal text files.

WML (Wireless Markup Language)is a client-side scripting language that is very similar to JavaScript. Like javascript WML script is used for user input validation, generation of error message and other dialog boxes etc A major difference between JavaScript and WMLScript is that JavaScript code can be embedded in the HTML markup, whereas WMLScript code is always placed in a file separated from the WML markup. URLs are used to refer to the actual WMLScript code in the WML document.

WML Script syntax rules

- An “executable” file in WML Script is called a function
- A function is
- An ordered set of language statements
- Modeled to perform a task
- Selected with the wmls filename extension
- Stored on the application server

- Executed on the device
- Referenced from WML, not contained in it
- Accessed by URL, like a deck or card

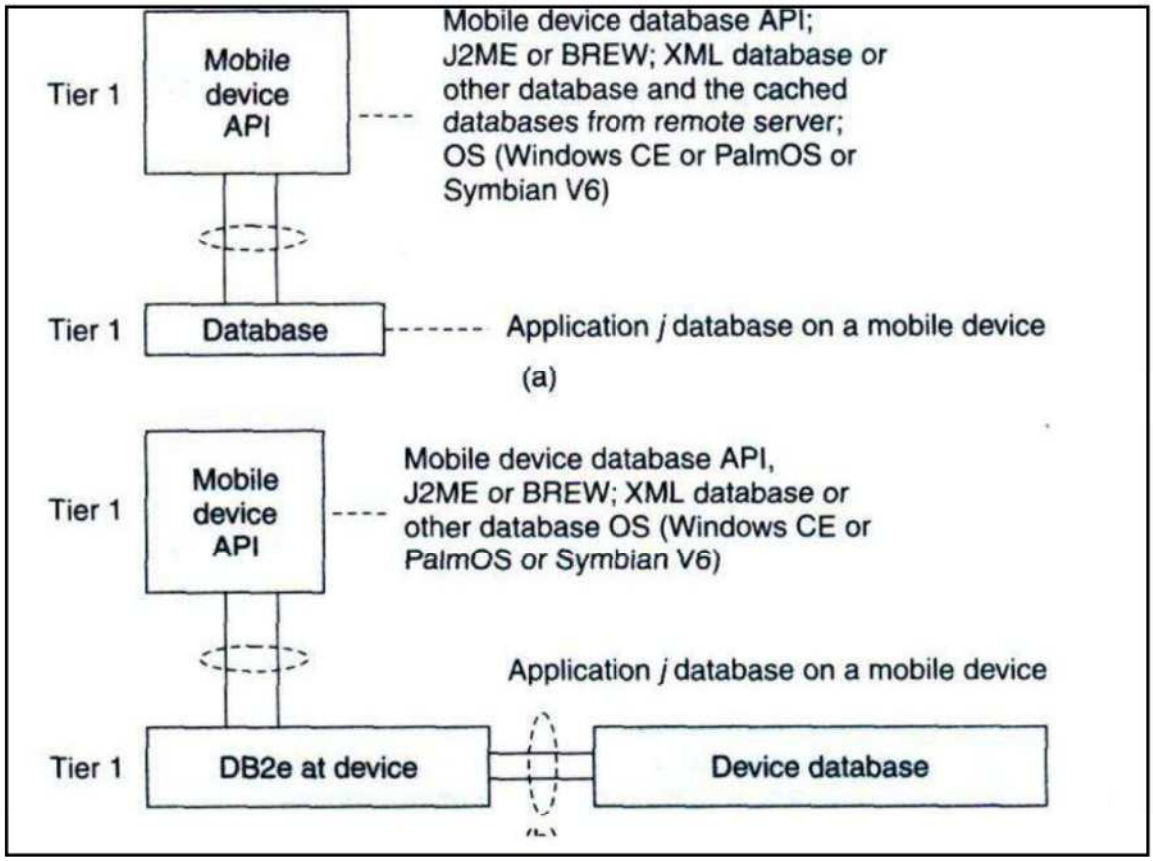
Unit – V Mobile Computing

Data Base Issues

A database is a collection of systematically stored records or information. Databases store data in a particular logical manner. A mobile device is not always connected to the server or network; neither does the device retrieve data from a server or a network for each computation. Rather, the device caches some specific data, which may be required for future computations, during the interval in which the device is connected to the server or network. Caching entails saving a copy of select data or a part of a database from a connected system with a large database. The cached data is hoarded in the mobile device database. Hoarding of the cached data in the database ensures that even when the device is not connected to the network, the data required from the database is available for computing.

Hoarding techniques

Database hoarding may be done at the application tier itself. The following figure shows a simple architecture in which a mobile device API directly retrieves the data from a database. It also shows another simple architecture in which a mobile device API directly retrieves the data from a database through a program, for ex: IBM DB2 Everyplace (DB2e).



Both the two architectures belong to the class of one-tier database architecture because the databases are specific to a mobile device, not meant to be distributed to multiple devices, not synchronized with the new updates, are stored at the device itself. Some examples are downloaded ringtones, music etc. **IBM DB2 Everyplace (DB2e)** is a relational database engine which has been designed to reside at the device. It supports J2ME and most mobile device operating systems. DB2e synchronizes with DB2 databases at the synchronization, application, or enterprise server.

The ***advantage of hoarding*** is that there is no access latency (delay in retrieving the queried record from the server over wireless mobile networks). The client device API has instantaneous data access to hoarded or cached data. After a device caches the data distributed by the server, the data is hoarded at the device. The disadvantage of hoarding is that the consistency of the cached data with the database at the server needs to be maintained.

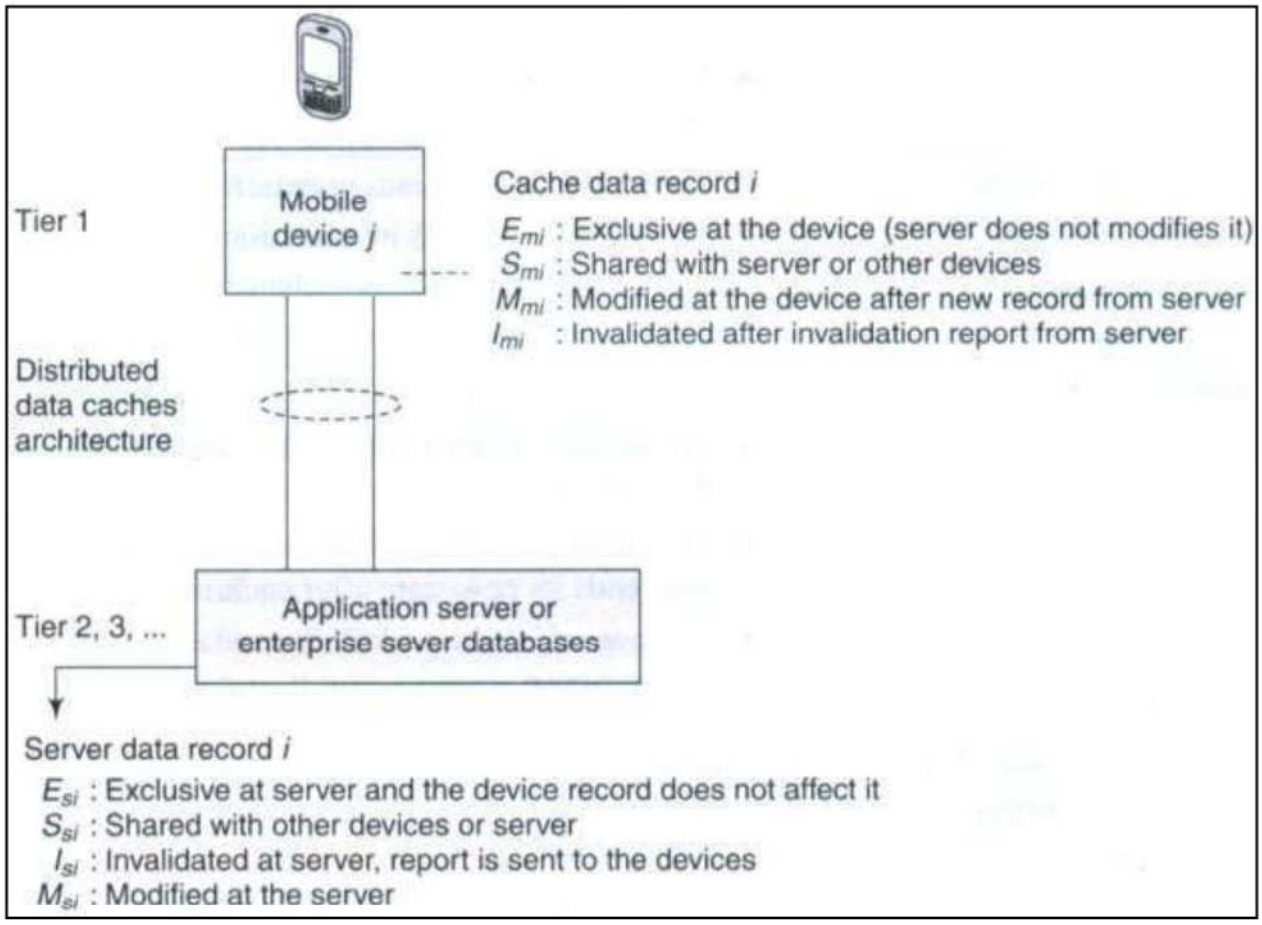
Caching Invalidation Mechanisms

A cached record at the client device may be invalidated. This may be due to expiry or modification of the record at the database server. Cache invalidation is a process by which a cached data item or record becomes invalid and thus unusable because of modification, expiry, or invalidation at another computing system or server. Cache invalidation mechanisms are used to synchronize the data at other processors whenever the cache-data is written (modified) by a processor in a multiprocessor system, cache invalidation mechanisms are also active in the case of mobile devices having distributed copies from the server.

A cache consists of several records. Each record is called a cache-line, copies of which can be stored at other devices or servers. The cache at the mobile devices or server databases at any given time can be assigned one of four possible tags indicating its state—*modified* (after rewriting), *exclusive, shared,* and *invalidated* (after expiry or when new data becomes available) at any given instance. These four states are indicated by the letters M, E, S, and I, respectively (MESI). The states indicated by the various tags are as follows:

a) The ***E*** tag indicates the *exclusive* state which means that the data record is for internal use and cannot be used by any other device or server.
b) The ***S*** tag indicates the *shared* state which indicates that the data record can be used by others.
c) The ***M*** tag indicates the *modified* state which means that the device cache
d) The ***I*** tag indicates the *invalidated state* which means that the server database no longer has a copy of the record which was shared and used for computations earlier.

The following figure shows the four possible states of a data record *i* at any instant in the server database and its copy at the cache of the mobile device *j*.



Another important factor for cache maintenance in a mobile environment is *cache consistency* (also called *cache coherence).* This requires a mechanism to ensure that a database record is identical at the server as well as at the device caches and that only the valid cache records are used for computations.

Cache invalidation mechanisms in mobile devices are triggered or initiated by the server. There are four possible invalidation mechanisms – ***Stateless asynchronous, stateless synchronous, stateful asynchronous and stateful synchronous.***

Stateless Asynchronous: A stateless mechanism entails broadcasting of the invalidation of the cache to all the clients of the server. The server does not keep track of the records stored at the device caches. It just uniformly broadcasts invalidation reports to all clients irrespective of whether the device cache holds that particular record or not. The term 'asynchronous' indicates that the invalidation information for an item is sent as soon as its value changes.

Stateless Synchronous: This is also a stateless mode, i.e., the server has no information regarding the present state of data records at the device caches and broadcasts to all client

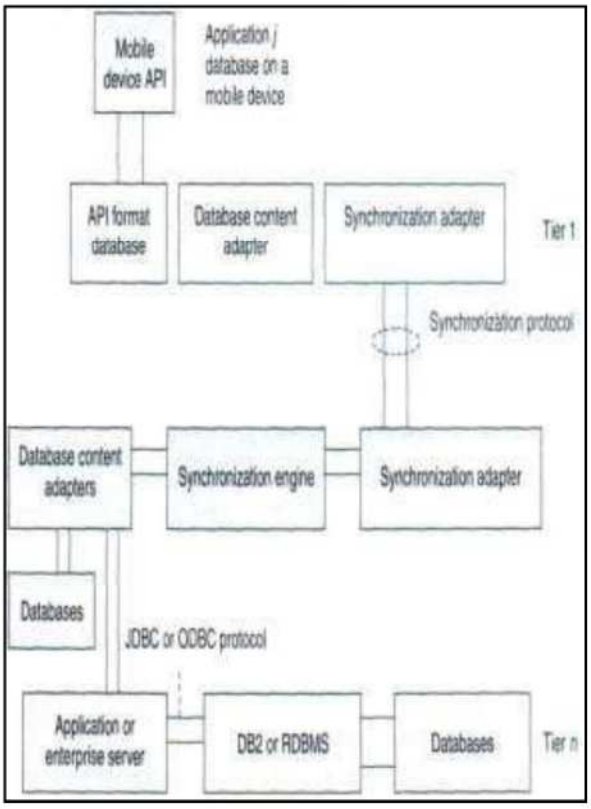
devices. However, unlike the asynchronous mechanism, here the server advertises invalidation information at periodic intervals as well as whenever the corresponding data-record at server is invalidated or modified.

Stateful Asynchronous: The stateful asynchronous mechanism is also referred to as the AS (asynchronous stateful) scheme. The term 'stateful' indicates that the cache invalidation reports are sent only to the affected client devices and not broadcasted to all. The server stores the information regarding the present state (a record I can have its state as *Emi, Mmi, Smi,* or *Imi)* of each data-record at the client device caches. This state information is stored in the home location cache (HLC) at the server.

Stateful Synchronous: The server keeps the information of the present state (Emi, Mmi, Smi, or Imi) of data-records at the client-caches. The server stores the cache record state at the home location cache (HLC) using the home agent (HA).

Client-Server Computing with Adaptation

The data formats of data transmitted from the synchronization server and those required for the device database and device APIs are different in different cases, there are two adapters at a mobile device—an adapter for standard data format for synchronization at the mobile device and another adapter for the backend database copy, which is in a different data format for the API at the mobile device. An adapter is software to get data in one format or data governed by one protocol and convert it to another format or to data governed by another protocol.



Context-aware Computing

Context-aware computing leads to application-aware computing. This is so because the APIs are part of the context (implicit or explicit contexts). For example, if context is a contact, the phone-talk application will adapt itself to use of the telephone number from the 'contact' and to the use of GSM or CDMA communication. Use of context in computing helps in reducing possibility of errors. It helps in reducing the ambiguity in the action(s). It helps in deciding the expected system response on computations. For example, if *name* is input in personal biodata context, then the *address, experience,* and *achievements,* which correspond to that name, are also required for computations. This is because all four are related and needed in biodata context. When *name* is input in telephone directory context, then the *address* and phone number, which correspond to that name, are also required for computations. This is because all three are related in context to telephone directory. The *name* in two different contexts (personal biodata and telephone directory) during computations needs computations to perform different actions.

Context Types in Context-aware Computing The five types of contexts that are important in context-aware computing are-physical context, computing context, user context, temporal context, and structural context.

Physical Context**:** The context can be that of the physical environment. The parameters for defining a physical context are service disconnection, light level, noise level, and signal strength. For example, if there is service disconnection during a conversation, the mobile device can sense the change in the physical conditions and it interleaves background noise so that the listener does not feel the effects of the disconnection.

Computing Context: The context in a context-aware computing environment may also be computing context. Computing context is defined by interrelationships and conditions of the network connectivity protocol in use (Bluetooth, ZigBee, GSM, GPRS, or CDMA), bandwidth, and available resources. Examples of resources are keypad, display unit, printer, and cradle. A cradle is the unit on which the mobile device lies in order to connect to a computer in the vicinity.

User Context: The user context is defined user location, user profiles, and persons near the user. Reza B 'Far defines user-interfaces context states as follows—'within the realm of user interfaces, we can define context as the sum of the relationships between the user interface components, the condition of the user, the primary intent of the system, and all of the other elements that allow users and computing systems to communicate.

Temporal Context: Temporal context defines the interrelation between time and the occurrence of an event or action. A group of interface components have an intrinsic or extrinsic temporal context. For example, assume that at an instant the user presses the switch for *dial* in a mobile device.

Structural Context: Structural context defines a sequence and structure formed by the elements or records. Graphic user interface (GUI) elements have structural context. Structural context may also be extrinsic for some other type of context.

Transaction Models

A transaction is the execution of interrelated instructions in a sequence for a specific operation on a database. Database transaction models must maintain data integrity and must enforce a set of rules called ACID rules. These rules are as follows:

Atomicity: All operations of a transaction must be complete. In case, a transaction cannot be completed, it must be undone (rolled back). Operations in a transaction are assumed to be one indivisible unit (atomic unit).

Consistency: A transaction must be such that it preserves the integrity constraints and follows the declared consistency rules for a given database. Consistency means the data is not in a contradictory state after the transaction.

Isolation: If two transactions are carried out simultaneously, there should not be any interference between the two. Further, any intermediate results in a transaction should be invisible to any other transaction.

Durability: After a transaction is completed, it must persist and cannot be aborted or discarded. For example, in a transaction entailing transfer of a balance from account A to account B, once the transfer is completed and finished there should be no roll back.

Consider a base class library included in Microsoft.NET. It has a set of computer software components called ADO.NET (ActiveX Data Objects in .NET). These can be used to access the data and data services including for access and modifying the data stored in relational database systems.

The ADO.NET transaction model permits three transaction commands:

1. BeginTransaction**: It is used to begin a transaction. Any operation after** BeginTransaction **is assumed to be a part of the transaction till the** CommitTransaction **command or the** RollbackTransaction **command. An example of a command is as follows:** connectionA.open(); transA = connectionA.BeginTransaction(); **Here** connectionA **and** transA **are two distinct objects.**

2. Commit: **It is used to commit the transaction operations that were carried out after the** BeginTransaction **command and up to this command. An example of this is** transA.Commit(); **All statements between** BeginTransaction **and** commit **must execute automatically.**

3. Rollback: **It is used to rollback the transaction in case an exception is generated after the** BeginTransaction **command is executed.**

A DBMS may provide for auto-commit mode. *Auto-commit mode* means the transaction finished

automatically even if an error occurs in between.

Query Processing

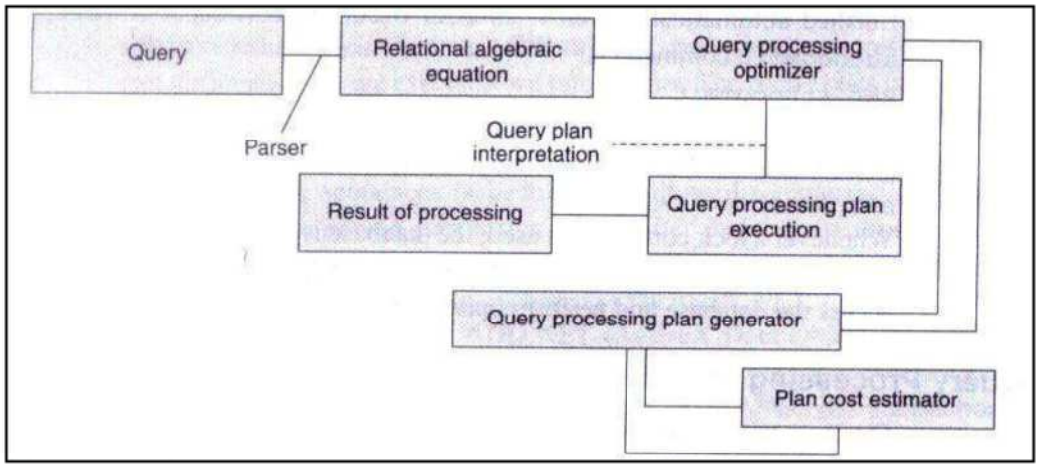
Query processing means making a correct as well as efficient execution strategy by *query decomposition* and *query-optimization*. A relational-algebraic equation defines a set of operations needed during query processing. Either of the two equivalent relational-algebraic equations given below can be used.

$$\pi_{cName,\ cTelNum}\ (\sigma_{Contacts.firstChar\ =\ "R"}\ (\sigma_{Contacts.cTelNum\ =\ DialledNumbers.dTelNum}\ (Contacts)\ \times\ DialledNumbers))$$

This means first select a column Contacts.cTelNum in a row in Contacts in which Contacts.cTelNum column equals a column DialledNumbers.dTelNum by crosschecking and matching the records of a column in Contacts with all the rows of DialledNumbers. Then in the second step select the row in which Contacts. firstChar = –R‖ and the selected cTelNum exists. Then in the third step project cName and CTelNum.

$$\pi_{cName,\ cTelNum}\ (\sigma_{Contacts.firstChar\ =\ "R"\ \wedge\ Contacts.cTelNum\ =\ DialledNumbers.dTelNum}\ (Contacts\ \times\ DialledNumbers))$$

This means that in first series of step, crosscheck all rows of Contacts and DialledNumbers and select, after AND operation, the rows in which Contacts.firstchar = –R‖ and Contacts.cTelNum = DialledNumbers.dTelNum. Then in the next step project cName and cTelNum form the selected records.



Query processing architecture

Π represents the projection operation, σ the *selection* operation, and Λ, the AND operation. It is clear that the second set of operations in query processing is less efficient than the first. Query

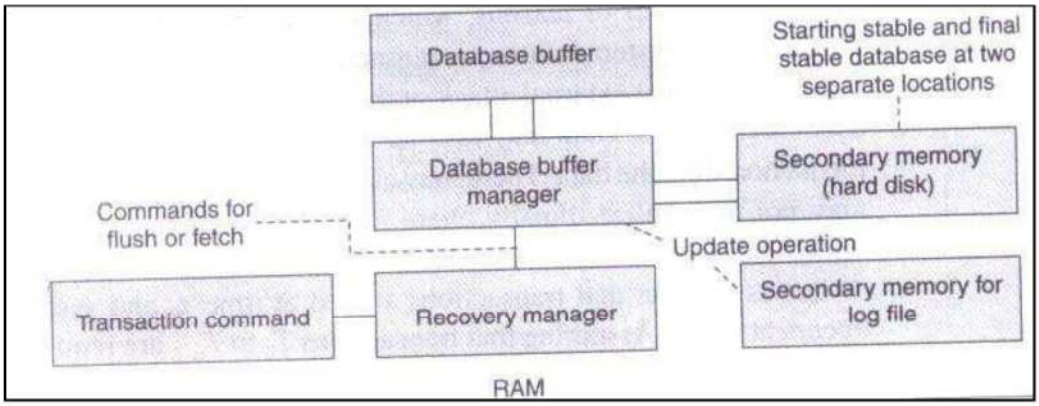
decomposition of the first set gives efficiency. Decomposition is done by (i) analysis, (ii) conjunctive and disjunctive normalization, and (iii) semantic analysis.

Efficient processing of queries needs optimization of steps for query processing. Optimization can be based on cost (number of micro-operations in processing) by evaluating the costs of sets of equivalent expressions. Optimization can also be based on a heuristic approach consisting of the following steps: perform the selection steps and projection steps as early as possible and eliminate duplicate operations.

Recovery and quality of service issues

Data is non-recoverable in case of media failure, intentional attack on the database and transactions logging data, or physical media destruction. However, data recovery is possible in other cases. Figure below shows recovery management architecture. It uses a recovery manager, which ensures atomicity and durability. Atomicity ensures that an uncommitted but started transaction aborts on failure and aborted transactions are logged in log file. Durability ensures that a committed transaction is not affected by failure and is recovered. Stable state databases at the start and at the end of transactions reside in secondary storage. Transaction commands are sent to the recovery manager, which sends fetch commands to the database manager.

The database manager processes the queries during the transaction and uses a database buffer. The recovery manager also sends the flush commands to transfer the committed transactions and database buffer data to the secondary storage. The recovery manager detects the results of operations. It recovers lost operations from the secondary storage. Recovery is by detecting the data lost during the transaction.



Recovery Management Architecture

The recovery manager uses a log file, which logs actions in the following manner:

1. Each instruction for a transaction for update (insertion, deletion, replacement, and addition) must be logged.
2. Database read instructions are not logged
3. Log files are stored at a different storage medium.
4. Log entries are flushed out after the final stable state database is stored.

Each logged entry contains the following fields.

- transaction type (begin, commit, or rollback transaction)
- transaction ID
- operation-type
- object on which the operation is performed
- Pre-operation and post-operation values of the object.

A procedure called the Aries algorithm is also used for recovering lost data. The basic steps of the algorithm are:

i. Analyse from last checkpoint and identify all dirty records (written again after operation restarted) in the buffer.
ii. Redo all buffered operations logged in the update log to finish and make final page.
iii. Undo all write operations and restore pre-transaction values.

The recovery models used in data recovery processes are as follows:

i. The *full recovery model* creates back up of the database and incremental backup of the changes. All transactions are logged from the last backup taken for the database.
ii. The *bulk logged recovery model* entails logging and taking backup of bulk data record operations but not the full logging and backup. Size of bulk logging is kept to the minimum required. This improves performance. We can recover the database to the point of failure by restoring the database with the bulk transaction log file backup. This is unlike the full recovery model in which all operations are logged.
iii. The *simple recovery model* prepares full backups but the incremental changes are not logged. We can recover the database to the most recent backup of the given database.

_____&&&_____