SEMESTER-IV

18CAP402 MOBILE COMPUTING 4H - 4C

Instruction Hours / week: L: 4 T: 0 P: 0 Marks: Internal: 40 External: 60 Total: 100

End Semester Exam: 3Hours

COURSE OBJECTIVES

- To introduce an advanced element of learning in the field of wireless communication and the concepts of wireless devices and mobile computing.
- To introduce wireless communication and networking principles, that support connectivity to cellular networks, wireless internet and sensor devices.
- To understand the use of transaction and e-commerce principles over devices to support mobile business concepts and to learn the basic concepts, aware of the GSM, SMS, GPRS Architecture.
- To understand the mobile operating system development environment such as android.

COURSE OUTCOMES(COs)

- Gain the knowledge about various types of Wireless Data Networks and Wireless Voice Networks.
- Analyse the architectures and the challenges of Wireless Networks.
- Analyse the role of Wireless Protocols in Wireless Networks.
- Know about different types of Wireless Communication Networks and their functionalities.
- Develop Mobile Applications Using Android

UNIT I - INTRODUCTION

Mobile Computing- Middleware and Gateways-Developing Mobile Computing Applications-Security in Mobile Computing – Architecture of Mobile Computing-Three-Tier Architecture-Design Consideration for Mobile Computing-Mobile Computing through Internet- Mobile Computing through Telephone-Developing an IVR Applications

UNIT II – BLUETOOTH AND GSM

Bluetooth- Features and working of RFID - Wireless Broadband (WiMax)- Mobile IP – IPV6-Java Card –Global System for Mobile Communications – GSM Architecture – Call Routing in GSM – GSM Addresses and Identifiers – Network Aspects in GSM – GSM Frequency Allocation – Authentication and Security- Mobile Computing Over SMS – SMS-Value Added Services through SMS.

UNIT III – GPRS, 3G AND 4G NETWORKS

GPRS and Packet Data Network – GPRS Network Architecture – GPRS Network Operations –Data Services in GPRS- Applications for GPRS – Limitations of GPRS- Spread Spectrum Technology- CDMA Versus GSM – – Features of 3G Networks –Architecture of 3G-

Applications of 3G - Features of 4G- Architecture of 4G - Wireless Technologies Used in 4G-Merits and Demerits of 4G

UNIT IV – MOBILE AD-HOC NETWORKS

MOBILE Ad-Hoc Basic Concepts – Characteristics – Applications – Design Issues – Routing – Essential of Traditional Routing Protocols –Popular Routing Protocols – Vehicular Ad Hoc Networks (VANET) – MANET Vs VANET – Security

UNIT V – ANDROID OPERATING SYSTEM

History of Android -Introduction to Android Operating Systems -Android Architecture - Android Virtual Device Manager - Features of Eclipse and Android Studio-Comparison of Kotlin Language to Java- User Interface Architecture of Android: Application context, intents, Activity life cycle, User Interface Design of Android –Features of Android SQLite Database

SUGGESTED READINGS

1. Asoke K. Talukder, Roopa R. (2011). Mobile Computing: technology, applications, and service creation, New Delhi ,Tata McGraw Hill.
2. R.Roger, J Lombarddo, Z Mednieks and B. Meike (2010). Android Applications Development, O'Reilly, Shroft Publishers & Distributors Pvt Ltd, New Delhi.
3. JochenSchiller (2000). Mobile Communication, Addison Wesley.
4. Brian Fling (2009). Mobile Design and Deevelopment, O'Reilly Media, Inc
5. William C.Y.Lee (1993). Mobile Communication Design Fundamentals, John Wiley.
6. Ivan Stojmenovic, (2002). Handbook Of wireless Networks And Mobile Computing, A Wiley-Interscience Publication.
7. Charles E.Perkins, (2008).Ad Hoc Networking, Addison-Wesley Publications

WEBSITES:

1. en.wikipedia.org/wiki/Mobile_computing
2. www.cse.iitk.ac.in/users/rkg/Talks/mobile_main.pdf
3. www.tutorialspoint.com/android/
4. pl.cs.jhu.edu/oose/resources/android/Android-Tutorial.pdf

KARPAGAM ACADEMY OF HIGHER EDUCATION

(Deemed to be University Established Under Section 3 of UGC Act 1956)

Coimbatore – 641 021.

LECTURE PLAN
DEPARTMENT OF CS,CA&IT

STAFF NAME: Dr.E.J.THOMSON FREDRIK
SUBJECT NAME: MOBILE COMPUTING SUB.CODE:18CAP402
SEMESTER: IV CLASS: II MCA

S.No	Lecture Duration Period	Topics to be Covered	Support Material/Page Nos
		UNIT-I	
1.	1	Introduction to Mobile Computing and Middleware and Gateways	T1: 5-8
2.	1	Developing Mobile Computing Applications, Security in Mobile Computing	T1: 15-16,W1
3.	1	Architecture of Mobile Computing and Three-Tier Architecture	T1: 31-34,
4.	1	Design Consideration for Mobile Computing	T1: 41- 53
5.	1	Mobile Computing through Internet	T1: 54-55
6.	1	Mobile Computing through Telephone, Developing an IVR Applications	T1: 55-56, J1
7.	1	Recapitulation and Discussion of important questions	
	Total No of Hours Planned For Unit I=7		

		UNIT-II	
1.	1	Bluetooth, RFID , Wireless Broadband (WiMax)	T1:84-88, W2
2.	1	Mobile IP, IPV6, Java Card ,	T1: 95-103
3.	1	Global System for Mobile Communications, GSM Architecture	T1:111-117
4.	1	Call Routing in GSM , GSM Addresses and Identifiers	T1: 129-130
5.	1	Network Aspects in GSM, GSM Frequency Allocation	T1: 130-135,W2
6.	1	Authentication and Security, Mobile Computing Over SMS	T1: 136-144,J2
7.	1	SMS Value Added Services through SMS	T1: 145-150, W1
8.	1	Recapitulation and Discussion of important questions	
	Total No of Hours Planned For Unit II = 8		
		UNIT-III	
1.	1	GPRS and Packet Data Network, GPRS Network Architecture	T1: 174-176
2.	1	GPRS Network Operations, Data Services in GPRS, Applications for GPRS	T1: 177-182
3.	1	Limitations of GPRS ,Spread Spectrum Technology, CDMA Versus GSM	T1: 218-235
4.	1	Wireless Data, Third Generation Networks	T1: 236-238
5.	1	Applications on 3G, Wireless LAN Advantages	T1: 243-245
6.	1	Wireless LAN Architecture, Mobility in Wireless LAN	T1: 251-257, R3:94-100
7.	1	Deploying Wireless LAN, Wireless LAN Security	T1:268-274
8.	1	Recapitulation and Discussion of important questions	
	Total No of Hours Planned For Unit III = 8		

		UNIT-IV	
1.	1	Mobile Phones, PDA, Design Constraints in Applications for Handheld Devices,	T1: 316-320
2.	1	Palm OS Architecture ,Communications in Palm OS, Introduction to Symbian	T1: 344-348
3.	1	Symbian OS Architecture, Applications for Symbian	T1: 360-362
4.	1	Security on the Symbian OS	T1:382-383
5.	1	JAVA in the Handset	T1: 388-389
6.	1	Java 2 Micro Edition Technology	T1: 392-396
7.	1	Different Flavours of Windows CE, Windows CE Architecture	T1: 465-468
8.	1	Recapitulation and Discussion of important questions	
	Total No of Hours Planned For Unit IV=8		
		UNIT-V	
1.	1	Android : Getting to know Android	T2: 3-5
2.	1	Android development environment	T2: 13-18
3.	1	Android development environment for Real Applications	T2: 27-30
4.	1	Startup code, MJ Android Applications	T2: 43-52
5.	1	Debugging Android Applications	T2:57-74
6.	1	Recapitulation and Discussion of important questions	
7.	1	Discussion on previous ESE question papers	
8.	1	Discussion on previous ESE question papers	
9.	1	Discussion on previous ESE question papers	
	Total No of Hours Planned for Unit V= 9		
Total Planned Hours	**40**		

TEXTBOOKS:

T1: Ashok K Talukder and Roopa R Yuuvagal, 2005 "Mobile Computing", Tata McGraw Hill Publishing Company Limited.

T2: R.Roger, J Lombarddo, Z Mednieks and B. Meike, 2010, Android Applications Development, O"Reilly, Shroft Publishers & Distributors Pvt Ltd, New Delhi.

REFERENCE BOOKS:

R1: JochenSchiller, Mobile Communication, Addison Wesley, 2000.

R2: William C.Y.Lee, Mobile Communication Design Fundamentals, John Wiley, 1993.

R3: Brian Fling , Mobile Design and Development, O'Reilly Media, Inc 2009

R4: Ashok K. Talukder, Roopa R., Mobile Computing: technology, applications, and service creation, New Delhi : Tata McGraw Hill,2011

Websites:

W1: en.wikipedia.org/wiki/mobile_computing
W2: www.cse.iitk.ac.in/users/rkg/talks/mobile_main.pdf
W3: www.tutorialspoint.com/android
W4: pl.cs.jhu.edu/oose/resources/android/Android-Tutorial.pdf

JOURNALS:

J1: "Mobile Computing Application Design and Development Issues",International Journal of Scientific Research, Vol 2,2013.

J2:"IEEE 802.11 Wireless LAN Security Issues and Solutions", IEEE Journal on Computer Networks, Vol 15,2014

J3:"Interface Design for Mobile Commerce", ACM Journal of Communications, Vol 46, No 12,2013.

SYLLABUS OF UNIT I - INTRODUCTION

Mobile Computing- Middleware and Gateways-Developing Mobile Computing Applications-Security in Mobile Computing – Architecture of Mobile Computing-Three-Tier Architecture-Design Consideration for Mobile Computing-Mobile Computing through Internet- Mobile Computing through Telephone-Developing an IVR Applications

Introduction to Mobile Computing

Mobile Computing is a technology that allows transmission of data, voice and video via a computer or any other wireless enabled device without having to be connected to a fixed physical link.
Mobile computing is all about portable and small computers, which includes PDAs (Personal Digital Assistants) like mobile phones, palmtops, laptops etc. In this growing technological world, people are much bound to work on computers and Internet.

Mobile computing can be defined as the ability to use technology that is not physically c connected to any static network. Nowadays, most laptops and personal digital assistants (PDA) all have wireless cards or Bluetooth interfaces built into them for very Good mobile internet access. Mobile computing is "taking a computer and all necessary files and software to the next Level. They are wireless devices. Mobile devices are portable that enables easy to carry and work with while you are on the move. It has attractive user interface. Provide many features like wireless LAN to access Internet from any part of the world. Enables voice and typical data transmission. Enables one to one contact to have conversations. Though the mobile computing devices have drawbacks such as low bandwidth, lack of security, loss of connectivity and battery backup issue, people still prefer mobile computing devices to desktops.

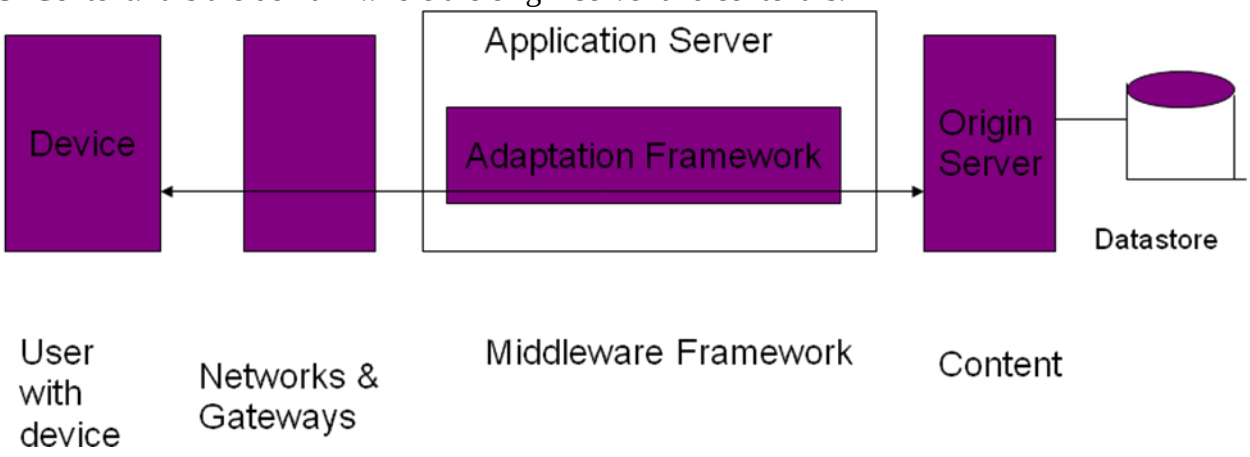
Mobile Computing Functions

A computing environment is defined as mobile if it supports one or more of these characteristics:

- User mobility: User should be able to move from one physical location to another location and use same service
- Network mobility: User should be able to move from one network to another network and use same service
- Device mobility: User should be able to move from one device to another and use same service
- Session mobility: A user session should be able to move from one user-agent environment to another.
- Service mobility: User should be able to move from one service to another
- Host mobility: The user should can be either a client or server.
- Mobile computing functions can be logically divided into the major segments:

1- User with device: fixed, portable
2- Network: different networks: GSM, CDMA, Ethernet, Wireless LAN, …etc.

3- Gateway: Interfacing different transport bearers
4- Middleware: handling the presentation and rendering of the content on a particular device.
5- Content: it is the domain where the origin server and content is.



Networks:

Mobile computing will use different types of networks: fixed telephone network, GSM, GPRS, ATM, …etc.

- 1- Wireline Networks: designed over wire. It is called fixed network. Copper or fiber optic cables.
- 2- Wireless Networks: mobile networks
- 3- Ad-hoc Networks: for this purpose only.
- 4- Bearers: transport bearers: TCP/IP, http, protocols for dialup connection.

Middleware and Gateways:

- Middleware: A software layered between a user application and operating system.
- Examples: communication middleware, object oriented middleware, message oriented middleware, database middleware, …etc.

In mobile computing we need different types of middleware components and gateways at different layers of the architecture. These are Types of Middleware

- 1- Communication middleware

The application will communicate with different nodes and services through different communication middleware. Examples could be NT3270 for IBM mainframe or Javamail connector

- 2- Transaction processing middleware

In many cases a service will offer session oriented dialogue (SoD). For a session to maintain over the stateless Internet. This is done through an application server. The user may be using a

device, which demands a short transaction whereas the service at the backend offers a SoD. In such cases a separate middleware component will be required to convert a SoD to a short transaction. Management of the Web components will be handled by this middleware as well.

- 3- Behavior management middleware

For different devices we need different types of rendering. We can have applications which are developed specially for different types of rendering. For example, we can have one application for Web, another for WAP, and a different for SMS.

- 4- Communication gateways

Between the device and the middleware there will be network of networks. Gateways are deployed when there are different transport bearers or network with dissimilar protocols. For example, we need an IVR gateway to interface voice with a computer, or a WAP gateway to access internet over a mobile phone.

Security in Mobile Computing:

Mobile security is the protection of smartphones, tablets, laptops and other portable **computing** devices, and the networks they connect to, from threats and vulnerabilities associated with wireless **computing**. **Mobile security** is also known as wireless **security**.

Mobililty is clearly the future of computing. Smartphones and tablets are more powerful and bring-your-own device is an accepted reality.

This raises major security issues, as mobile computing can be readily compromised at the device, network and wireless connectivity levels. The mobile device itself—whether a portable computer, personal digital assistant, laptop, smartphone, tablet computer, or a wearable computer—can be damaged, lost, or stolen. Mobile communications elements, properties, protocols, data formats, and technologies can be targeted and result is lost or damaged data.

Smartphones pose special risks due to the inherent weaknesses in SMS, MMS, Wi-Fi networks, and GSM. They are also vulnerable to attacks against apps, web browsers, and operating systems. Most importantly, typical mobile device users tend to be unaware of or too casual about security measures, commonly leaving their devices open to malicious software and individualized attacks.

It doesn’t help that portable devices frequently depend on public networks for normal operations. Many people use mobile devices to collect, access, and process large amounts of valuable and sensitive personal, corporate, and financial information while on insecure networks.

Fortunately, countermeasures currently exist and and more are being developed.

Mobile computing security can be implemented in various layers of mobile software, operating systems, and downloadable apps. In addition, end users can be sensitized to the dangers and educated as to best practices, greatly increasing their devices' security.

Guidelines for Mobile Computing Security

Encryption: Mobile devices that do store sensitive data can be protected by means of encryption systems. Automatic encryption/decryption systems exist, but are less secure than systems which require the user to enter a password at the beginning of every session. Both Android and Apple iOS devices can be set up to utilize encryption capabilities.

External Identification: End users should label their mobile devices with their name and telephone contact information so lost devices can be returned to them, even after their battery has gone dead.

Limiting Data Storage: One of the best ways to prevent the compromise or loss of sensitive data is not to store it on a mobile device. Such data can be stored in the cloud or accessed from a proprietary server. Naturally, means of access must be thoroughly secured, or there is no advantage to be gained from keeping sensitive data off a mobile device.

Lost Device Locator and Data Eraser Systems: Depending on the mobile device and its operating systems, there are various technologies that enable end users to locate a lost device (even if it's just between the couch cushions). Failing that, there are ways to remotely erase sensitive data. Encourage end users to enroll their devices in a good system, and to learn how to use it.

Passwords and Timeouts: End users should set a password and a relatively brief timer to shut down and lock their mobile devices when left idle for even a few minutes. Passwords and timeouts prevent—or at least delay—unauthorized users from gaining access to sensitive data not only on lost or stolen devices, but also on devices left unattended in homes and offices.

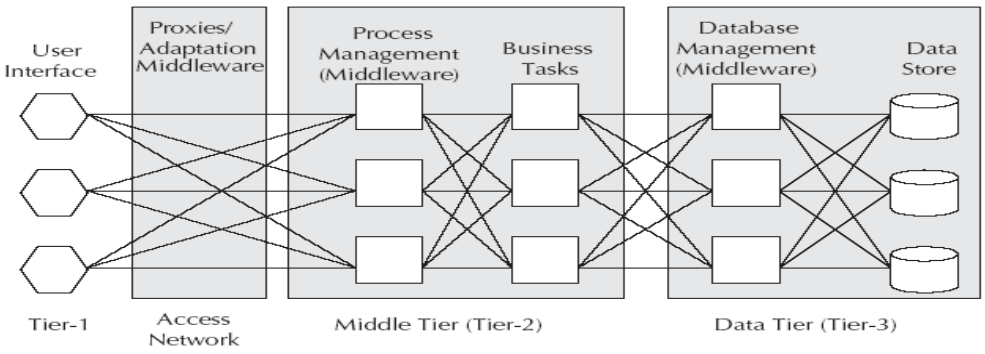
Trusted Sources: Mobile devices can add software from a variety of sources, but end users should rely only on trusted sources, such as the Apple iTunes Store, Google Play, or the Amazon App Store for Android. Other sources are less likely to thoroughly search for and prevent software contaminated by viruses or other malware.

Updates: Hackers and defensive software are engaged in running battles for superiority, so any delay in updating operating systems and/or security systems leaves mobile devices particularly vulnerable. Systems should be set to check automatically for updates, and users should get in the habit of performing manual updates at regular intervals.

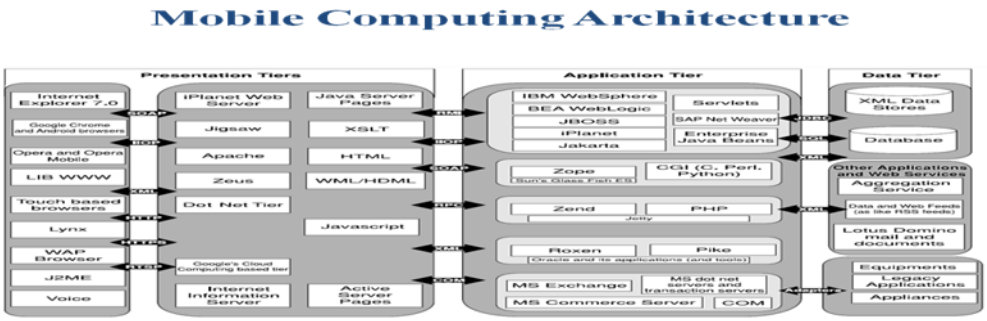
While it may seem like no data is safe in this technological age, users can greatly decrease the likelihood of a security breach on their devices by adhering to these mobile computing security guidelines.

Architecture of Mobile Computing
Three Tier Architecture:

The three tier architecture contains the user interface or the presentation tier, the process management or the application tier and the



Data management tier



Presentation Tier

- Responsible for presenting the information to the end user
- Run on the client device and offer all the user interfaces
- Includes web browsers, WAP browsers and client programs

Application Tier

- Independent of presentation and database management
- Handles functions related to middleware
- Middleware – layer of software sitting between the operating system and user facing software

- Many types of middleware – Message Oriented Middleware, Transaction Processing Middleware, Communication Middleware, Distributed Objects and Components, Transcoding Middleware, Web Services, etc.

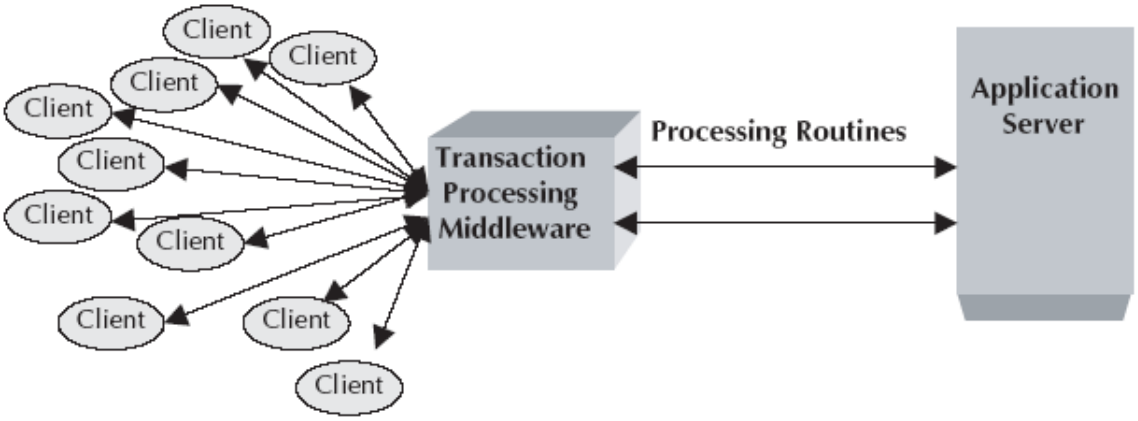
Message Oriented Middleware

- Loosely connects different applications through asynchronous exchange of messages
- Works independent of platform or processor configuration
- Generally asynchronous and peer to peer
- Works in publish / subscribe fashion
- Examples – MQ series from IBM, JMS, etc.

Transaction Processing Middleware

- Provides tools and environment for developing transaction based distributed applications
- Capable of providing services to thousands of clients in a distributed client – server environment
- Independent of database architecture

Example – CICS from IBM



Communication Middleware

- Used to connect one application to another
- Quite useful in the telecommunications world
- Uses mediation server to automate the telnet protocol to communicate to nodes in the network
- Example – Using telnet to connect one application to another

Mobile Computing through Telephony:

Evolution of Telephony

- 1876 – First telephone system by Alexander Graham Bell
- 1892 – Strowger switch installed as first automatic telephone exchange
- 1960 – Electronic Switching System (ESS) developed by AT&T
- 1962 – Carrier system went digital
- 1960's and 1970's – Stored program telephone exchanges
- 1980's – Wireless telephony emerged
- 1990's – Mobile telephony flourished commercially across the world

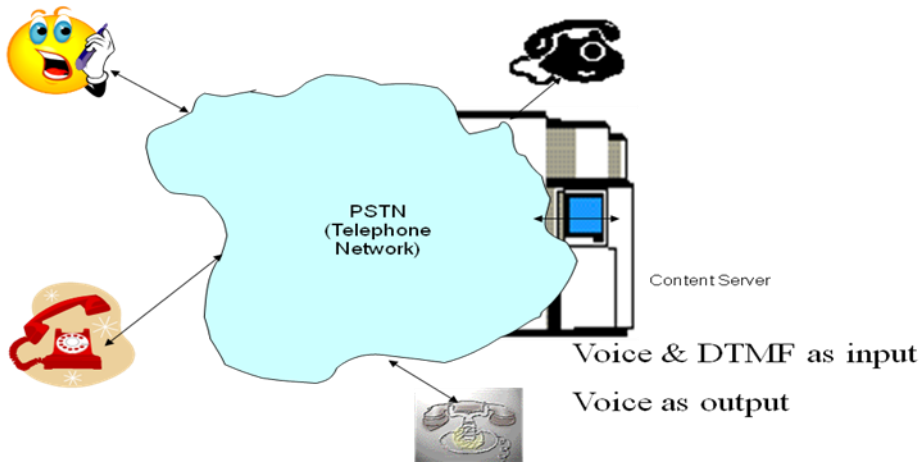
Public Switched Telephone Network

- Three categories of nodes – local exchanges, transit exchanges, and international exchanges.
- Local exchanges are used for subscriber connection.
- Transit exchanges switch traffic within and between different geographical areas.
- International exchanges switch traffic to telecommunication networks in foreign countries and other distant networks.
- Last mile – a physical wire (also known as local loop) that is laid from local exchange to the device at subscriber premises.
- Last mile is absent in case of GSM or Will network.

Mobile Computing through Telephone:

- Computer Telephony Interface (CTI) – accessing applications through voice interface
- Intelligent Networks (IN) – solving problem of multiple and geographically bounded numbers
- CTI achieved through Interactive Voice Response Service (IVRS)
- IVRS – known as Voice Response Unit in USA
- Audiotex systems – systems based on IVRS and VRU
- Dual Tone Multi Frequency (DTMF) – extensively used in IVRS

Architecture for CTI



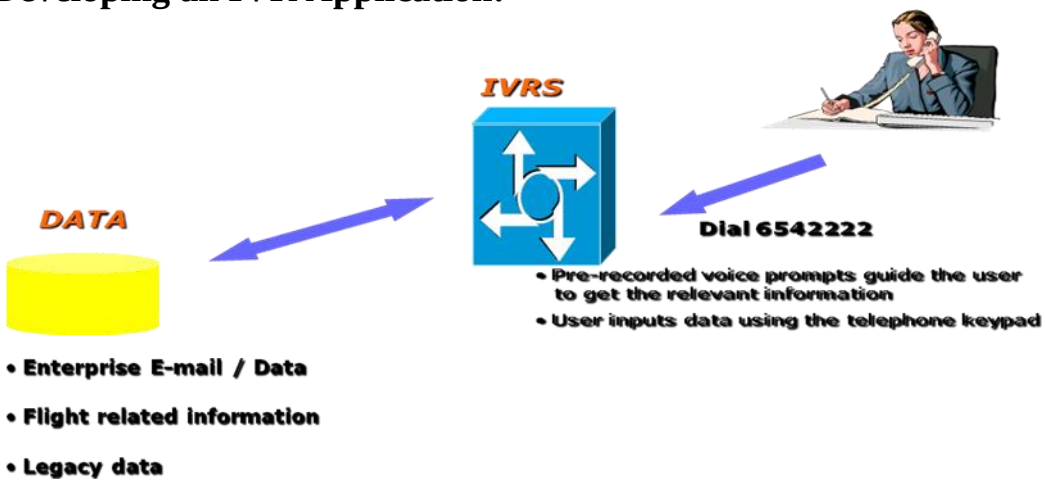
Benefits of CTI:

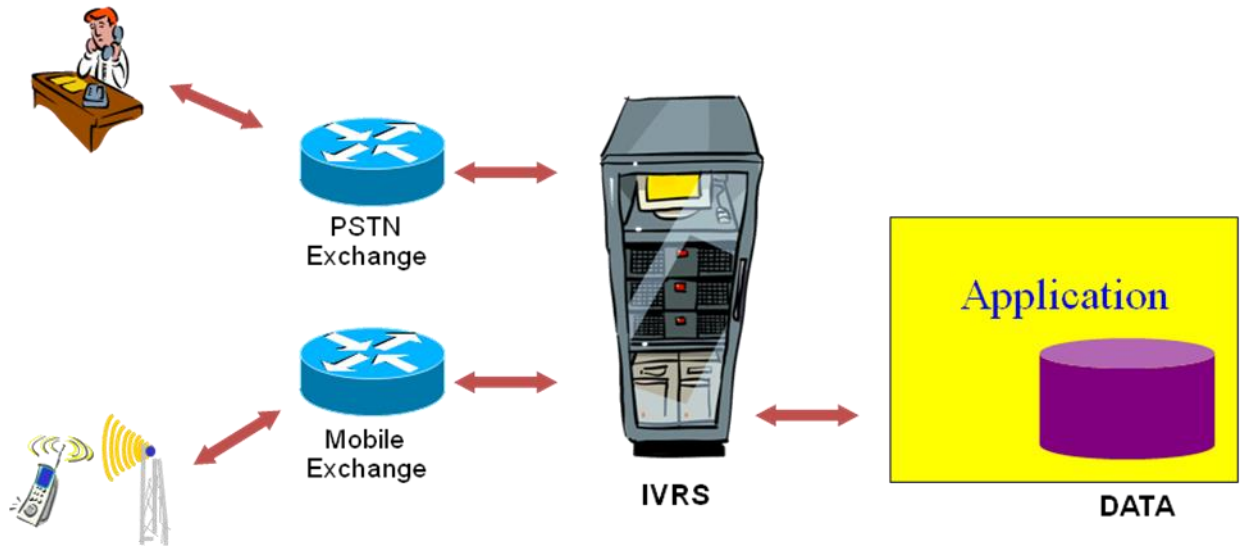
- Easy integration with existing telephone and computer systems irrespective of whether they are state – of – the – art databases or legacy systems
- Independent of the locale and culture as the same application can be adapted for other languages very quickly
- Very effective for rural population and senior citizens who are not computer savvy
- Comprehensive voice – processing capabilities designed specifically for small, middle and large – size businesses
- Provide customized application development tools that are ideal for different size and types of businesses

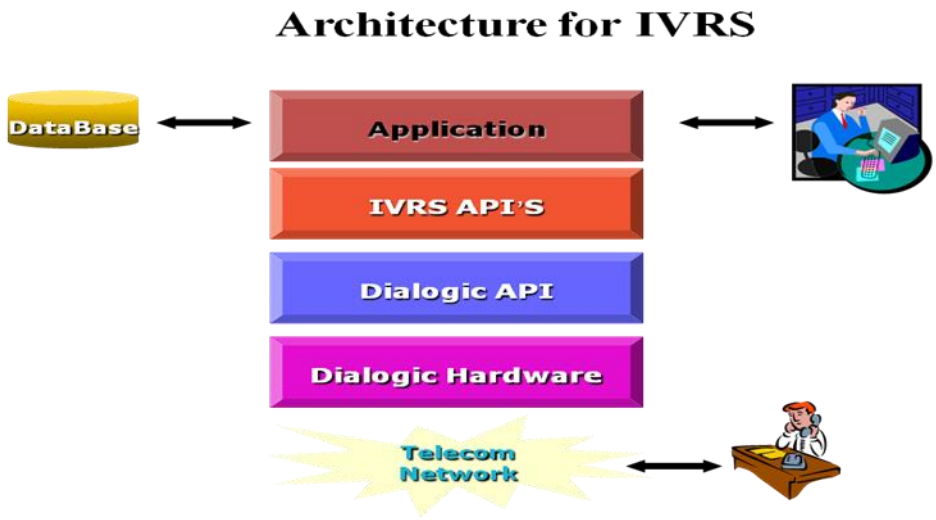
How CTI is different?

- The only difference is User Interface
- Input through DTMF telephone keypad
- Input through rotary telephone
- Input through voice
- Output is always audible sound

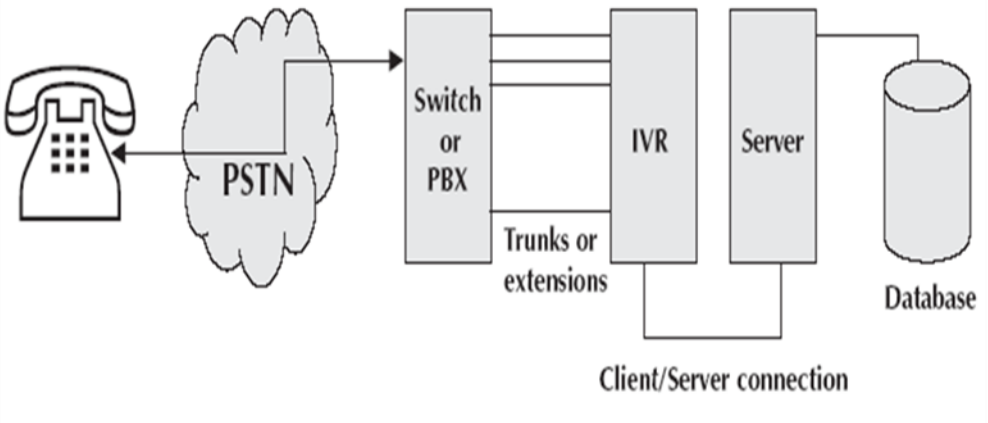
Developing an IVR Application:



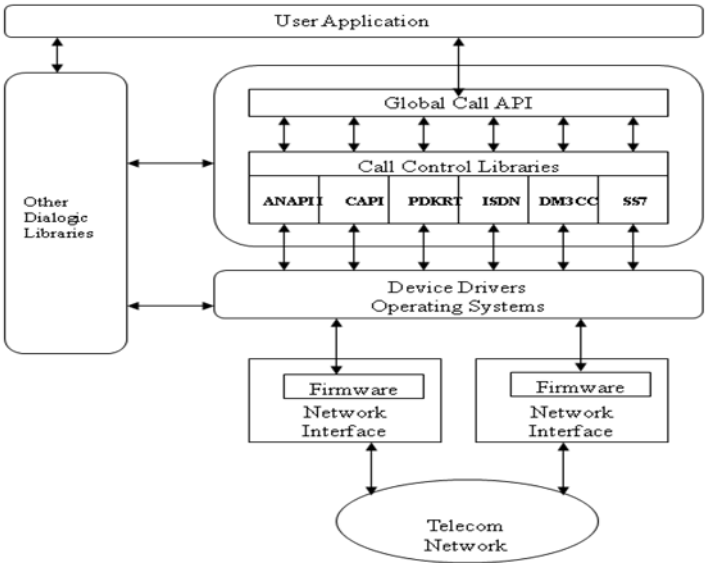




Architecture for IVRS – in general



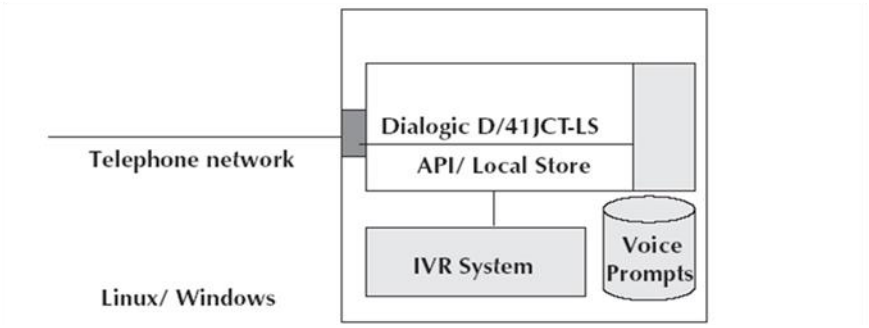
Dialogic architecture for IVRS



IVRS enables solutions like …

- Text to speech conversion
- Voice / music messaging
- Inbound and outbound call processing
- Audiotex premium rate services
- Intelligent call directing and routing
- Voice mail
- Billing information and fax on demand.
- Reminder service

Inside an IVRS



IVRS – Interface to exchange

- **Physical Interface :**
  Analog : RJ 11
  Digital : 2Mbps E1 interface (75 ohm / 120 ohm)
- **Digital Signaling :**

ISDN-Primary Rate Interface
R2-MFC
SS#7

Features of IVRS – I

- Touch tone input (DTMF)
- Caller identification detection
- Multiple application hosting
- DNI based application routing
- Out dialing
- Call transfer
- Audio recording
- Text – to – speech conversion

Features of IVRS – II

- CDR
- GUI based remote monitoring
- Call statistics (Web Interface)
- Database access
- Data access through Internet
- Support of standard voice file format (.vox, .wav, etc)

Features of IVRS – III

- ❑ Input through

- Telephone keypad
- Rotary dial
- Voice recognition

- ❑ Numeric input (avoid alphabetic input)

- Convert numeric input to other types of input through menu

- ❑ Alphabetic input can be OK if played back
- ❑ Output through

- Synthesized voice
- Text – to – speech

Using IVRS through telephone keys

- ❑ Alphabets A, B and C on key 2
- ❑ Alphabets C, E and F on key 3
- ❑ Alphabets G, H and I on key 4
- ❑ Alphabets J, K and L on key 5
- ❑ Alphabets M, N and O on key 6
- ❑ Alphabets P, Q, R and S on key 7
- ❑ Alphabets T, U and V on key 8
- ❑ Alphabets W, X, Y and Z on key 9

Using IVRS – DTMF frequencies

	1209 Hz	1336Hz	1477Hz
697 Hz	1	2/ABC	3/DEF
770Hz	4/GHI	5/JKL	6/MNO
852 Hz	7/PQRS	8/TUV	9/WXYZ
941 Hz	*	0	#

Synthesized Voice

- ❑ Prerecorded syllables or part of the response recorded into separate files and stored in persistent storage

- As and when necessary, these files will be played through programming logic the way we do convert numeric to text in a bank cheque
- Example: 123 = one.vox + hundred.vox + twenty.vox + three.vox

where .vox is pre – recorded voice

Voice driver and API's

- Used to communicate and control voice hardware on IVRS system
- Can make calls, answer calls, identify caller, play and record sound from phone line and detect DTMF signals
- Can tear down a call detect that caller has hung up
- Offers API's to record transaction details

IVRS Programming

- Voice libraries are supported by Dialogic to interface with voice driver
- Voice libraries exist for single and multithreaded applications
- C function libraries exist for a number of purposes
- Standard run time library provides a set of common functions independent of device and applicable to Dialogic devices

Single threaded programming model

- Enables a program to control multiple voice channels within a single thread
- Allows development of applications where multiple tasks need to be coordinated simultaneously
- Supports both polled and call back event management

Multithreaded programming model

- Uses functions that block application execution until the function completes
- Applications control each channel from a separate thread or a process
- Enables IVRS system to assign distinct applications to different channels dynamically in real time
- Dialogic provides API's to use voice board
- API's are available for:

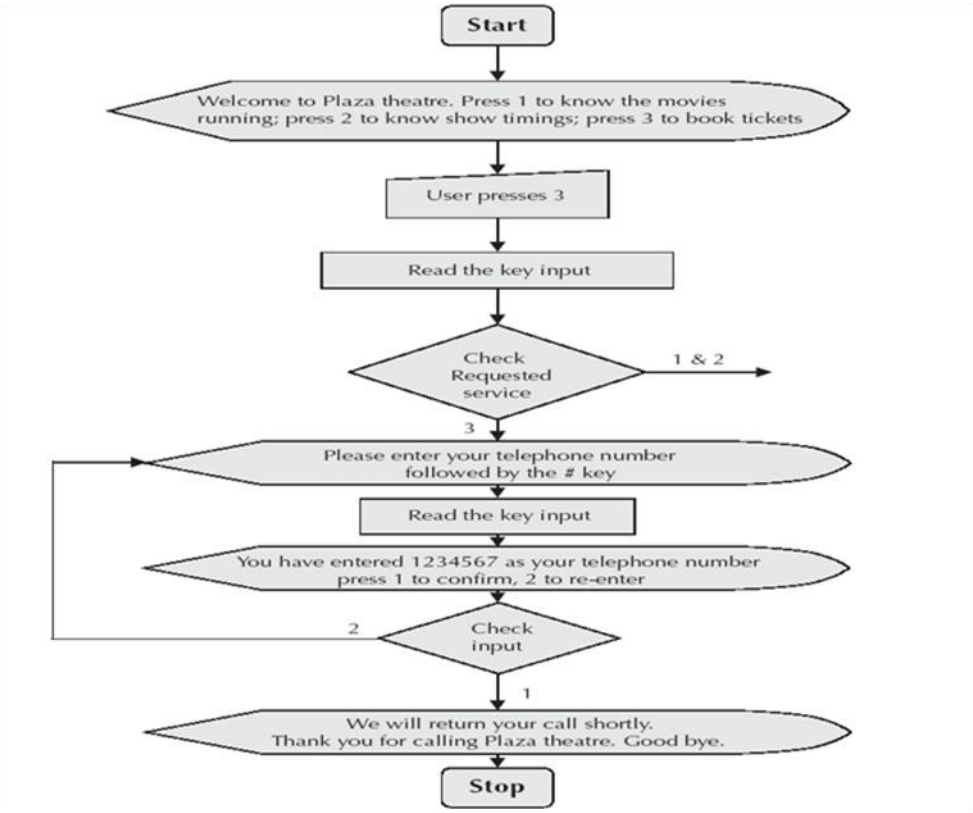
- Device management
- Configuration functions
- I/O functions
- Play and record functions
- Tone detection and generation functions
- Call control functions
-

Dialogic IVRS functions

- **dx_open( )** - opens a voice channel
- **dx_close( )** - closes a voice channel

- **dx_wtcallid( )** - waits for rings and reports Caller ID
- **dx_getdig( )** - gets digits from channel digit buffer
- **dx_play( )** - plays recorded voice data
- **dx_playvox( )** - plays a single .vox file
- **dx_playwav( )** - plays a single .wave file
- **dx_rec( )** - records voice data
- **dx_recvox( )** - records voice data to a single .vox file
- **dx_recwav( )** - records voice data to a single .wave file
- **dx_dial( )** - dials an ASCII string of digits

A typical IVRS call flow



UNIT I - POSSIBLE QUESTIONS

PART – B

(Each question carries six marks)

1. Explain in detail about Middlewares and Gateways.
2. Explain the three-tier architecture for mobile computing with suitable diagram.
3. Write notes on Mobile Computing and its functions using suitable diagram.
4. Explain the applications and services of mobile computing.
5. Explain the application and data -tier architecture of mobile computing with suitable diagram.
6. Write about the IVR architecture in mobile computing with suitable example
7. Explain the multiple access procedure in mobile computing with suitable diagram.
8. Explain the voice interfaces and software in mobile computing with suitable example
9. Explain the three-tier architecture of mobile computing with suitable diagram.
10. Write the development of mobile computing through telephone with suitable example.

PART – C

Compulsory Question

(Each question carries ten marks)

1. What should be the characteristics of Mobile Computing Devices?
2. Describe the security aspects in mobile computing.
3. Explain the functionality of Mobile Computing
4. How would you broadly classify the mobile computing applications?
5. Who are all the stakeholders of wireless network?
6. What is WiMax? How it is differ from WiFi?
7. Describe 3G networks.
8. Explain the Process of GSM Call Routing.
9. Explain the Process of Interactive Voice Response.
10. Briefly describe the Eclipse IDE for the development of android application.

MOBILE COMPUTING UNIT- I

S.No	Question	Option1	Option2	Option3	Option4	Answer
1	Mobile Computing can be defined as a computing environment of _____mobility	physical	logical	ethical	universal	physical
2	It is defined as an environment in a foreign network	AnytimeAnywhere	Wearable Computers	GSM	VHE	VHE
3	It is made available in any environment	pervasive computing	ubiquitous computing	Nomadic computing	Cloud computing	pervasive Computing
4	In_____mobility the user should be able to move from one physical location to another and use the same service	Bearer mobility	User mobility	Network mobility	Device Mobility	User mobility
5	In_____each node in the network is a combination of a host and a router.	MANET	GSM	GPRS	WAP	MANET
6	_____is the mobility,in that the user should be able to move from one bearer to another and use the same service	User Mobility	Bearer Mobility	Network mobility	Device Mobility	Bearer Mobility
7	_____is an example for a sales representatives using their desktop computer in their home office	Device Mobility	User mobility	Agent Mobility	Session Mobility	Device Mobility
8	_____is an example could be a user using his service through CDMA	Host Mobility	User mobility	Session Mobility	Agent Mobility	Session Mobility
9	In_____mobility,virus software that moves from one node to another	Host Mobility	Session Mobility	Bearer Mobility	Agent Mobility	Agent Mobility
10	Software agents in_____will constantly be mobile	Agent Mobility	Cloud computing	Session Mobility	HostMobility	Cloud computing
11	In_____mobility the user device can be a client or a server	Session Mobility	Host Mobility	Agent Mobility	Cloud computing	Host Mobility
12	In case of host mobility,mobility of the_____needs to be taken care of.	Device	Harware	Software	IP	IP
13	Middleware components and gateways are divided into _____ types	2	3	5	4	4
14	Any software layered between a user application and operating system is _____	Hardware	Software	Middleware	Firmware	Middleware
15	Remote Procedure call is an example for _____	Operating System	Application	Hardware	Middleware	Middleware
16	In_____middleware,the application will communicate with different nodes	Communication	Transaction	Behavior	Communication Gateways	Communication
17	A Separate middleware component will be required tocovert	SID to SOD	SOD to SID	SMS to MMS	MMS to SMS	SOD to SID

18	WAP gateway to access Internet over a mobile phone is an example for _____	CommunicationMiddleware	Communication Gateways	Transaction Processing Middleware	Behavior Management Middleware	Communication Gateways
19	_____technology is operator centric.	MMS	SMS	GPRS	WAP	SMS
20	In Mobile computing all infrastructure and technology is designed by _____	GPS	CDMA	GSM	IETF	GSM
21	In Mainframe Computers many mission critical systems use a _____	TP	WebTV	VDU	POS	TP
22	A TP Monitor coordinates with the user to pick up the right processing to service _____	User Requests	Client Requests	Business Transactions	Banking Transactions	Business Transactions
23	A TP Monitor provides functions such as _____	Stacking	Scheduling	Analysing	Queuing	Queuing
24	The network centric mobile computing architecture uses _____ architecture	3-tier	2-tier	single tier	4-tier	3-tier
25	Data tier architecture is also called as_____Management	Business	DataBase	Server	Change	DataBase
26	The 3-tier architecture hides the complexity of _____ processing from the user	Centralized	Distributed	Transaction Processing Middleware	Data	Distributed
27	Touch screen is also referred as _____	Screen	Pen	Kiosks	Light screen	Kiosks
28	_____ presentation will relate to rendering on Screen	Powerpoint	Visual	OHP	Animation	Visual
29	In a Web scraper ,the agent embeds functionality of the _____ browser	IE	Netscape	HTTP	Chrome	HTTP
30	_____ tier is the engine of the ubiquitous computing.	Application	Presentation	Session	Physical	Application
31	_____provides a message queue between any two interoperating applications	TP	MOM	JMS	SMS	MOM
32	MOM support_____message passing	Asyncronous	Synchronous	Short	Multimedia	Synchronous
33	An example of MOM is message queue from_____computer known as MQ Series	Micro	Macro	Mini	IBM	IBM
34	JMS stands for _____	Java Muitimedia Service	Java Message Service	Java Middleware Service	Java Mail Service	Java Message Service
35	The first automatic telephone exchange was called the _____	Electronic Switching system	Strowger switch	Cross bar switch	Context switch	Strowger switch
36	_____is a signal applied to the line after the calling party	Ring Tone	Waiting Tone	Dial Tone	Busy Tone	Dial Tone
37	_____indicated either that the called subscriber already busy or subscriber is congested	Busy Tone	Dial Tone	Ring Tone	Waiting Tone	Busy Tone
38	When a circuit between A party and the B party is established,the telephone rings at B party's end and a_____is generated for the A party	Dial Tone	Busy Tone	Ring Tone	Waiting Tone	Ring Tone
39	A normal telephone system is called _____	WiLL	GSM	PSTN	AMPS	PSTN

40	_____is traditionally known as the last mile	Local Loop	Wireless in Local Loop	GSM	PSTN	Local Loop
41	A physical wire is also known as _____	WiLL	Local Loop	PSTN	AMPS	Local Loop
42	_____is one of the most common multiplexing procedures	CDMA	TDMA	SDMA	FDMA	FDMA
43	The first generation analog mobile networks includes _____	AMPS	SDMA	PSTN	TDMA	AMPS
44	In_____several users shares the same frequency channel	PRMA	TDMA	SDMA	CDMA	TDMA
45	_____is an example of digital transmission	IS-136	TACS	WiLL	AMPS	IS-136
46	Mobile Computing work through voice interfaces are called as _____	IN	CT	PSTN	IVR	IVR
47	A telephone keyboard has_____keys	10	12	11	15	12
48	The key inouts received by the voice card in the telephone as _____	IVR	Business card	DTMF	TTS	DTMF
49	_____interface convert text into speech	DTMF	TTS	IVR	API	TTS
50	DTMF signals are otherwise referred as _____	Ring Tones	Beep Tones	Alert Tones	Touch Tones	Touch Tones

UNIT II NOTES

UNIT II – BLUETOOTH AND GSM

Bluetooth- Features and working of RFID - Wireless Broadband (WiMax)- Mobile IP – IPV6-Java Card –Global System for Mobile Communications – GSM Architecture – Call Routing in GSM – GSM Addresses and Identifiers – Network Aspects in GSM – GSM Frequency Allocation – Authentication and Security- Mobile Computing Over SMS – SMS-Value Added Services through SMS.

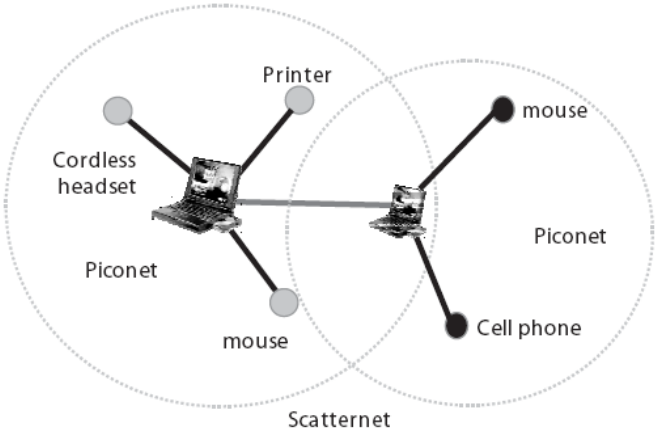
Bluetooth

- Name comes from nickname of Danish king Harald Blåtand
- Allows users to make ad hoc wireless connections between devices like mobile phones, desktop or notebook computers wirelessly
- Data transfer at a speed of about 720 Kbps within 50 meters (150 feet) of range or beyond through walls, clothing and even luggage bags
- Built into a small microchip
- Operates in a globally available frequency band ensuring worldwide interoperability
- Managed and maintained by Bluetooth Special Interest Group

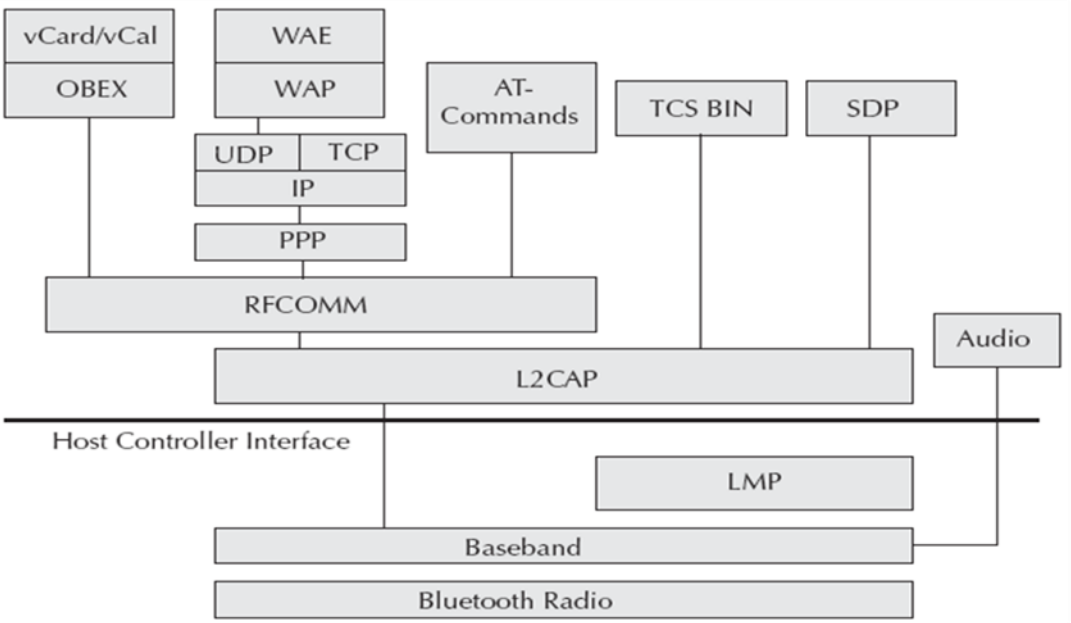
Bluetooth Protocol

- Uses the unlicensed 2.4 GHz ISM (Industrial Scientific and Medical) frequency band
- 79 available channels spaced 1 MHz apart from 2.402 GHz to 2.480 GHz
- Allows power levels starting from 1mW (covering 10 centimetres) to 100mW (covering upto 100 meters) suitable for short device zone to personal area networks within a home
- Supports both unicast (point-to-point) and multicast (point-to-multipoint) connections
- Bluetooth protocols are a collection of many inter-related protocols
- Uses the master and slave relationship
- Master and slaves together form a Piconet when master allows slaves to talk
- Up to seven ‘slave’ devices can be set to communicate with a ‘master’ in a Piconet.
- Scatternet is formed when several of piconets are linked together to form a larger network in an ad hoc manner

Scatternet is a topology where a device from one piconet also acts as a member of another piconet wherein a device being a master in one piconet can simultaneously be a slave in the other one.



- Bluetooth Core protocols plus Bluetooth radio protocols are required by most of Bluetooth devices
- Uses spread spectrum technologies at the Physical Layer while using both direct sequence and frequency hopping spread spectrum technologies
- Uses connectionless (ACL–Asynchronous Connectionless Link) and connection-oriented (SCO–Synchronous Connection-oriented Link) links
- Cable Replacement layer, Telephony Control layer and Adopted protocol layer form application-oriented protocols



OBEX – Object Exchange Protocol
WAE – Wireless Application Environment
WAP – Wireless Application Protocol
LMP – Link Manager Protocol
TCS BIN – Telephony Control Specification Binary
SDP – Service Discovery Protocol
RFCOMM – Radio Frequency Communication
L2CAP – Logical Link Control and Adaptation Protocol

Bluetooth Protocol Stack:

1. Bluetooth Core Protocols
2. Cable Replacement Protocol
3. Telephony Control Protocols
4. Adopted Protocols

- Baseband – enables physical RF link
- Link Manager Protocol (LMP) – manages devices in range, power modes, connections, duty cycles, etc.
- Logical Link Control and Adaptation Protocol (L2CAP) – segmentation and re-assembly of fragmented packets with their multiplexing
- Service Discovery Protocol (SDP) – Enables a device to join a piconet

Telephony Control Protocols

- Telephony Control Specification Binary (TCS BIN) - defines the call control signaling protocol and handles mobility management for groups of Bluetooth TCS devices

- Attention (AT) Commands - defines a set of commands by which a mobile phone can be used and controlled as a modem for fax and data transfers

Adopted Protocols:

- Point-to-Point Protocol (PPP) - means of taking IP packets to/from the PPP layer and placing them onto the LAN
- Transmission Control Protocol/Internet Protocol (TCP/IP) - used for communication across the Internet
- Object Exchange (OBEX) Protocol - session protocol to exchange objects and used to browse the contents of folders on remote devices
- Content Formats - used to exchange messages and notes and synchronize data amongst various devices
-

Bluetooth Security

- Offers security infrastructure starting from authentication, key exchange to encryption
- Uses the publicly available cipher algorithm known as SAFER+ to authenticate a device's identity

Bluetooth Application Models

Each application model in Bluetooth is realized through a Profile. Profiles define the protocols and protocol features supporting a particular usage model. Some common profiles are:

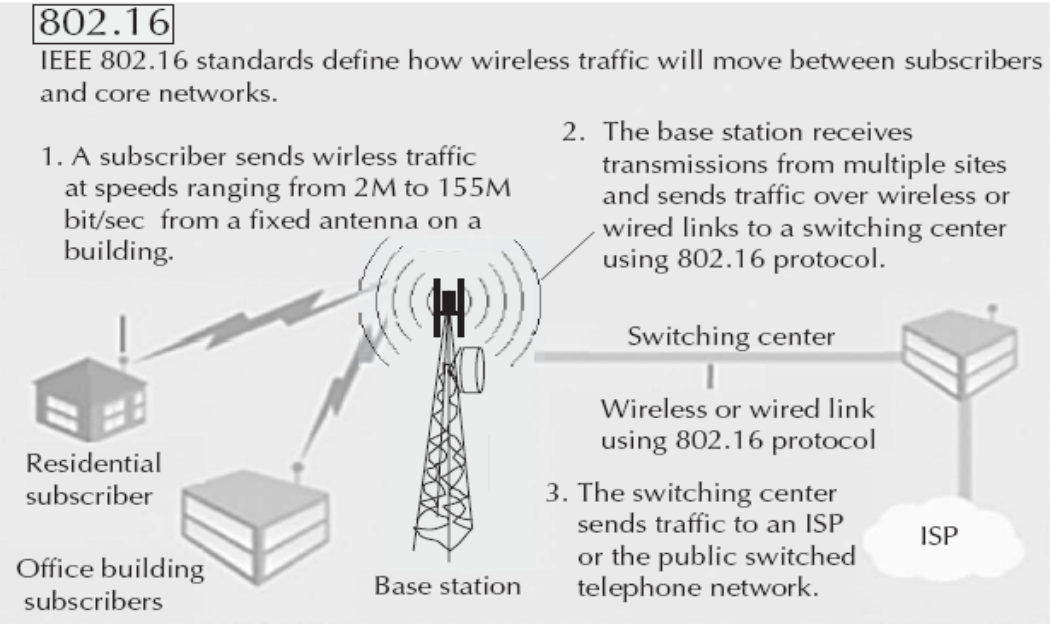
- File Transfer
- Internet Bridge
- LAN Access
- Synchronization
- Headset

Wireless Broadband:

- Also known as Wireless Metropolitan Area Network (Wireless MAN) and Wireless Microwave Access (WiMAX)
- IEEE 802.16 standard released in April 2002
- Offers an alternative to high bandwidth wired access networks like fiber optic, cable modems and DSL
- Provides network access to buildings through exterior antennas communicating with radio base stations

- Networks can be created in just weeks by deploying a small number of base stations on buildings or poles to create high capacity wireless access systems.

Overview of IEEE 802.16



Sub-standards of IEEE 802.16

- IEEE 802.16.1 - Air interface for 10 to 66 GHz
- IEEE 802.16.2 - Coexistence of broadband wireless access systems
- IEEE 802.16.3 - Air interface for licensed frequencies, 2 to 11 GHz

Basics of IEEE 802.16

IEEE 802.16 standards are concerned with the air interface between a subscriber's transceiver station and a base transceiver station

- The Physical Layer
- MAC Layer
- Convergence Layer

Physical Layer

- Specifies the frequency band, the modulation scheme, error-correction techniques, synchronization between transmitter and receiver, data rate and the multiplexing structure
- Both TDD and FDD alternatives support adaptive burst profiles in which modulation and coding options may be dynamically assigned on a burst-by-burst basis

- Three physical layer for services: Wireless MAN-SC2, Wireless MAN-OFDM and Wireless MAN-OFDMA

Medium Access Control Layer

- Designed for point-to-multipoint broadband wireless access
- Addresses the need for very high bit rates, both uplink (to the base station) and downlink (from the base station)
- Services like multimedia and voice can run as 802.16 MAC is equipped to accommodate both continuous and bursty traffic

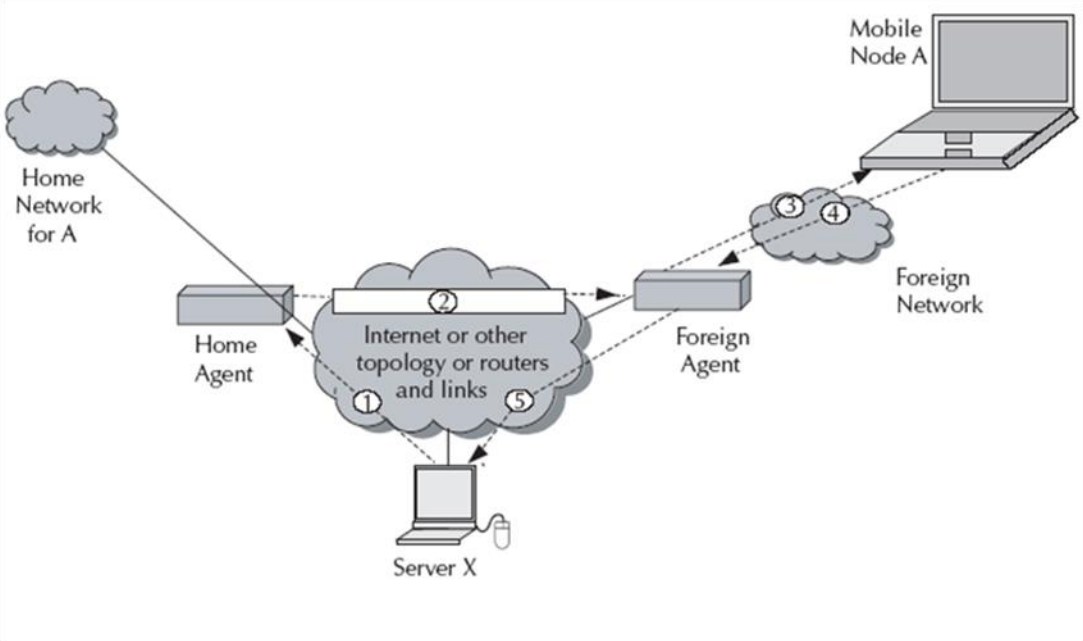
Convergence Layer

- Provides functions specific to the service being provided
- Bearer services include digital audio/video multicast, digital telephony, ATM, Internet access, wireless trunks in telephone networks and frame relay

Mobile IP:

- 'Mobile IP' signifies that, while a user is connected to applications across the Internet and the user's point of attachment changes dynamically, all connections are maintained despite the change in underlying network properties
- Similar to the handoff/roaming situation in cellular network
- Mobile IP allows the mobile node to use two IP addresses called home address and care of address
- The home address is static and known to everybody as the identity of the host
- The care of address changes at each new point of attachment and can be thought of as the mobile node's location specific address

Working of Mobile IP

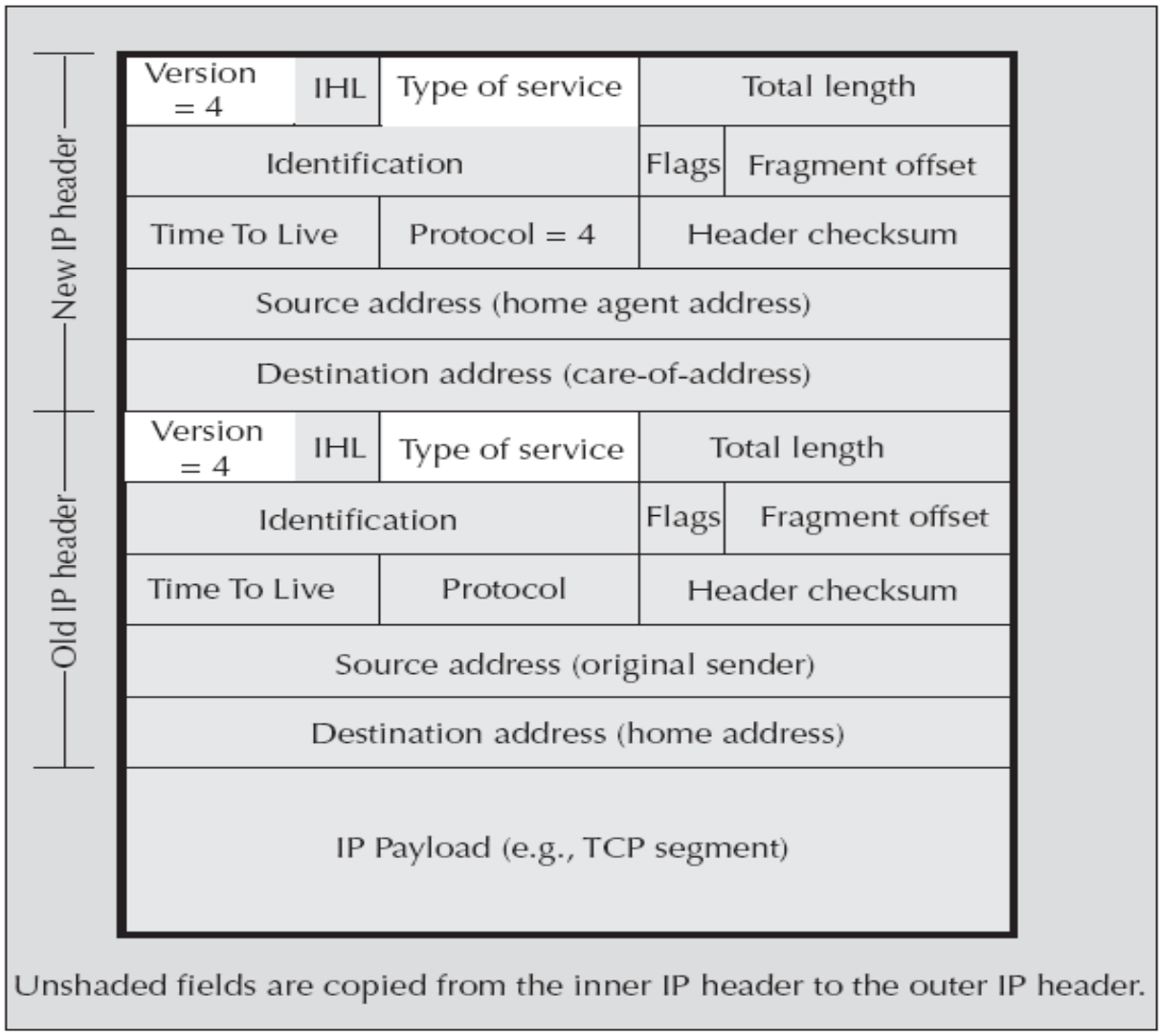


Let's take the case of mobile node (A) and another host (server X). The following steps take place:

- Server X wants to transmit an IP datagram to node A. The home address of A is advertised and known to X. X does not know whether A is in the home network or somewhere else. Therefore, X sends the packet to A with A's home address as the destination IP address in the IP header. The IP datagram is routed to A's home network.
- At the A's home network, the incoming IP datagram is intercepted by the home agent. The home agent discovers that A is in a foreign network. A care of address has been allocated to A by this foreign network and available with the home agent. The home agent encapsulates the entire datagram inside a new IP datagram, with A's care of address in the IP header. This new datagram with the care of address as the destination address is retransmitted by the home agent.
- At the foreign network, the incoming IP datagram is intercepted by the foreign agent. The foreign agent is the counterpart of the home agent in the foreign network. The foreign agent strips off the outer IP header, and delivers the original datagram to A.
- A intends to respond to this message and sends traffic to X. In this example, X is not mobile; therefore X has a fixed IP address. For routing A's IP datagram to X, each datagram is sent to some router in the foreign network. Typically, this router is the foreign agent. A uses X's IP static address as the destination address in the IP header.
- The IP datagram from A to X travels directly across the network, using X's IP address as the destination address.

- Discovery - A mobile node uses a discovery procedure to identify prospective home agents and foreign agents.
- Registration - A mobile node uses a registration procedure to inform its home agent of its care-of address.
- Tunneling - Tunneling procedure is used to forward IP datagrams from a home address to a care of address.

IP headers in Mobile IP



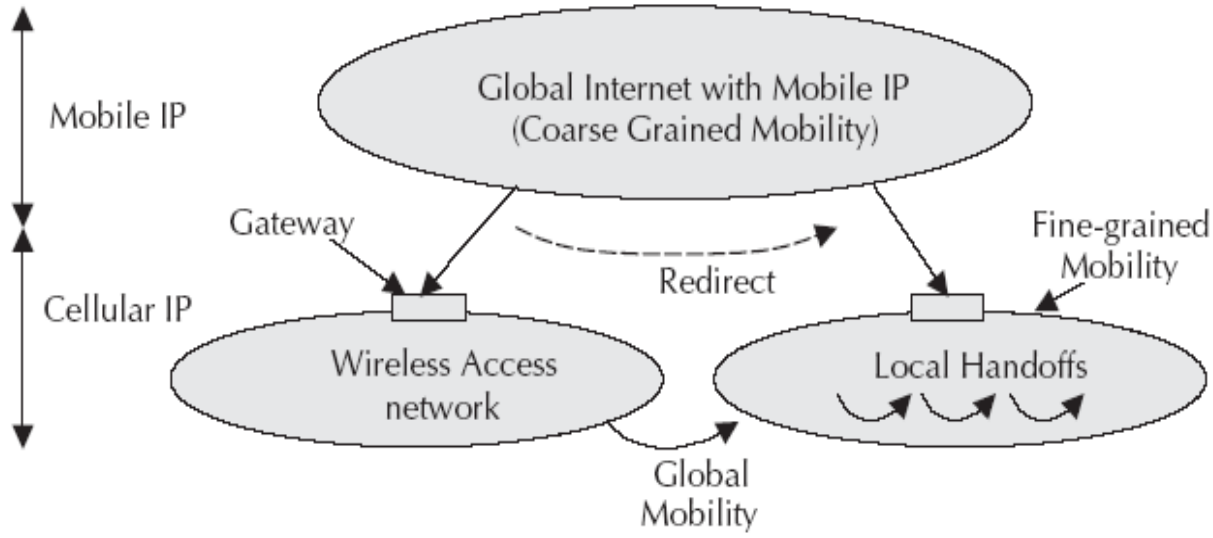
Cellular IP

None of the nodes know the exact location of a mobile host. Packets addressed to a mobile host are routed to its current base station on a hop-by-hop basis where each node only needs to know on which of its outgoing ports to forward packets. This limited routing information (referred as mapping) is local to the node and does not assume that nodes have any knowledge of the

wireless network topology. Mappings are created and updated based on the packets transmitted by mobile hosts.

- Uses two parallel structures of mappings through Paging Caches (PC) and Routing Caches (RC)
- PCs maintain mappings for stationary and idle (not in data communication state) hosts
- RC maintains mappings for mobile hosts
- Mapping entries in PC have a large timeout interval, in the order of seconds or minutes. RCs maintain mappings for mobile hosts currently receiving data or expecting to receive data

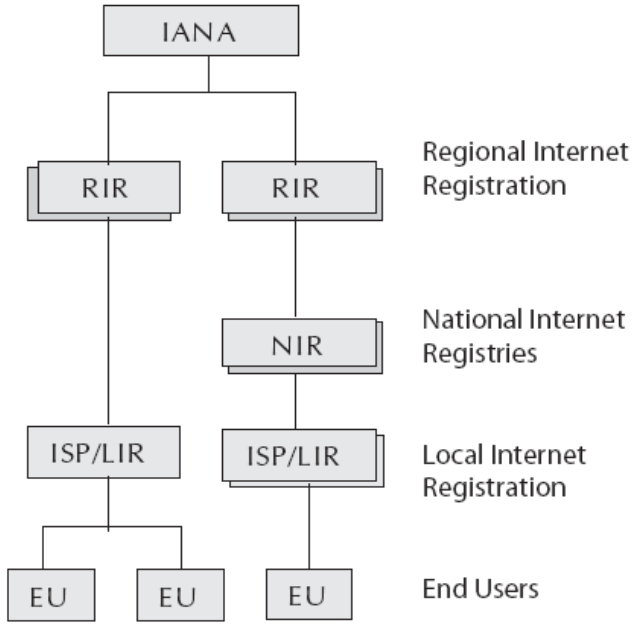
Relationship between Mobile IP and Cellular IP



Internet Protocol version

- Successor to today's IP version 4 protocol (IPv4)
- Internet Engineering Task Force (IETF) has produced a comprehensive set of specifications (RFC 1287, 1752, 1886, 1971, 1993, 2292, 2373, 2460, 2473, etc.) that define the next-generation IP protocol originally known as 'IPNg'
- Uses 128 bit addresses for each packet creating a virtually infinite number of IP addresses (approximately 3.4*10**38 IP addresses) as opposed to 3758096384 IPv4 addresses
- There are global addresses and local addresses
- Global addresses are used for routing of global Internet
- Link local addresses are available within a subnet
- IPv6 uses hierarchical addressing with three level of addresses
- Includes a Public Topology (the 48 bit external routing prefix)
- Site Topology (typically a 16 bit subnet number)
- Interface Identifier (typically an automatically generated 64 bit number unique on the local LAN segment)

Hierarchical addressing of IPv6



IPv6 Security

- Comes native with a security protocol called IP Security (IPSec)
- IPSec protocol is a standards-based method of providing privacy, integrity and authenticity to information transferred across IP networks.

Features of IPSec

- Diffie-Hellman key exchange mechanism for deriving key between peers on a public network
- Public key cryptography to guarantee the identity of the two parties and avoid man-in-the-middle attacks
- Bulk encryption algorithms, such as 3DES, for encrypting the data
- Keyed hash algorithms, such as HMAC, combined with traditional hash algorithms such as MD5 or SHA for providing packet authentication
- Digital certificates signed by a certificate authority to act as digital ID cards
- IPSec provides IP network layer encryption

Migrating from IPv4 to IPv6

- Migration of the network components to be able to support IPv6 packets. Using IP tunneling, IPv6 packets can propagate over an IPv4 envelope. Existing routers can support IP tunneling.
- Migration of the computing nodes in the network. This will need the operating system upgrades so that they support IPv6 along with IPv4. Upgraded systems will have both IPv4 and IPv6 stacks.

Migration of networking applications in both client and server systems. This requires porting of the applications from IPv4 to IPv6 environment.

Interconnecting IPv6 networks

- Tunneling is one of the key deployment strategies for both service providers as well as enterprises during the period of IPv4 and IPv6 coexistence.
- Tunneling service providers can offer an end-to-end IPv6 service without major upgrades to the infrastructure and without impacting current IPv4 services.

Tunneling Mechanisms

- Manually created tunnels such as IPv6 manually configured tunnels (RFC 2893)
- IPv6 over IPv4 tunnels
- Semiautomatic tunnel mechanisms such as that employed by tunnel broker services
- Fully automatic tunnel mechanisms such as IPv4 compatible

Mobile IP with IPv6

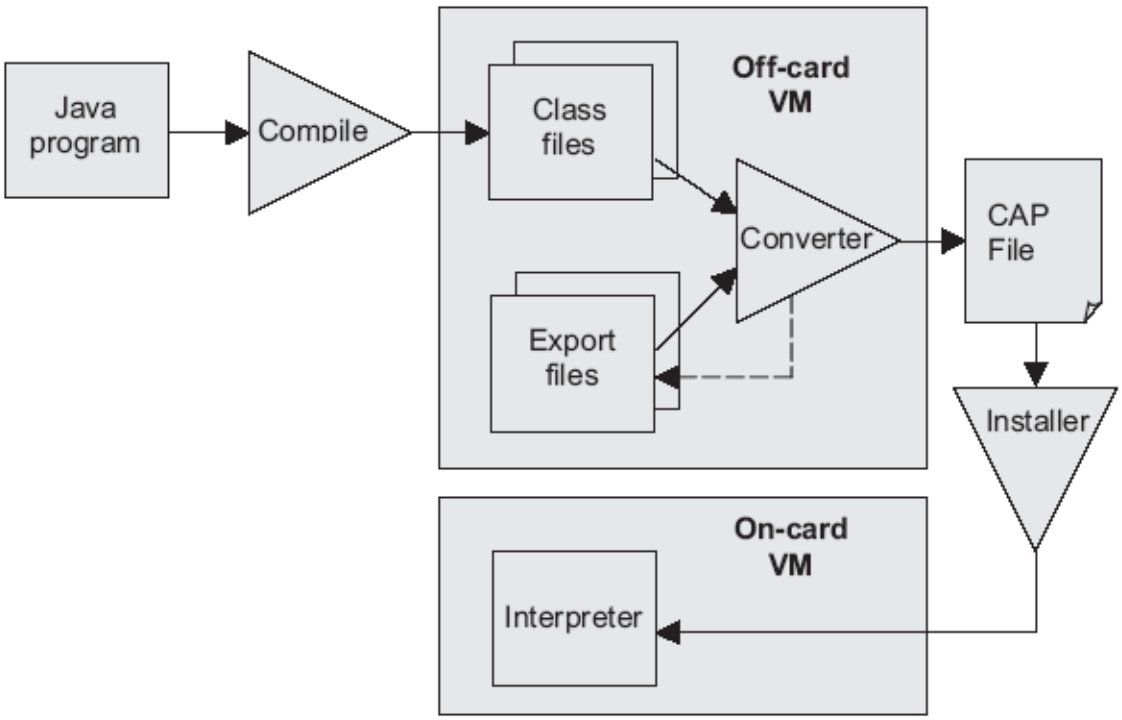
- IPv6 with hierarchical addressing scheme can manage IP mobility much efficiently.
- IPv6 also attempts to simplify the process of renumbering which could be critical to the future routing of the Internet traffic.
- Mobility Support in IPv6, as proposed by the Mobile IP working group, follows the design for Mobile IPv4. It retains the ideas of a home network, home agent and the use of encapsulation to deliver packets from the home network to the mobile node's current point of attachment.
- While discovery of a care of address is still required, a mobile node can configure its a care of address by using Stateless Address Auto-configuration and Neighbor Discovery. Thus, foreign agents are not required to support mobility in IPv6.

Java Card

- Smart card with Java framework
- Smart card is a plastic card with intelligence and memory
- A smart card is embedded with either (i) a microprocessor and a memory chip or (ii) only a memory chip with non-programmable logic
- A microprocessor card can have an intelligent program resident within the card which can add, delete, and otherwise manipulate information on the card
- A memory card can store some information for some pre-defined operation
- Java was chosen as the vehicle for interoperability
- All the microprocessor based smart cards now offer Java API framework on the smart card

- Java Card technology preserves many of the benefits of the Java programming languages such as productivity, security, robustness, tools, and portability
- For Java card, the Java Virtual Machine (JVM), the language definition, and the core packages have been made more compact to bring Java technology to the resource constrained smart cards

Architecture of Java Card



Functioning of Java Card

- Java card technology supports OTA (Over The Air) downloads
- Applications written for the Java Card platform are referred to as applets
- Challenge of Java Card technology on smart card is to fit Java system software in a resource constraint smart card while conserving enough space for applications

- The Java Card virtual machine on a smart card is split into two parts; one that runs off-card and the other that runs on-card.
- Many processing tasks that are not constrained to execute at run time, such as class loading, bytecode verification, resolution and linking, and optimization, are dedicated to the virtual machine that is running off-card where resources are usually not a concern
- The on-card components of Java Card include components like the Java card virtual machine (JCVM), the Java card runtime environment (JCRE), and the Java API
- Task of the compiler is to convert a Java source into Java class files
- The converter will convert class files into a format downloadable into the smart card while ensuring the byte code validity.

The converter checks the classes off-card for

- Well formedness
- Java Card subset violations
- Static variable initialization
- Reference resolution
- Byte code optimization
- Storage Allocation.
- The java card interpreter
- Executes the applets
- Controls run-time resources
- Enforces runtime security
- Following conversion by the off-card VM into CAP (Converted APlet) format, the applet is transferred into the card using the installer
- Applet is selected for execution by the JCRE
- JCRE is made up of the on-card virtual machine and the Java Card API classes and performs additional runtime security checks through applet firewall
- Applet firewall partitions the objects stored into separate protected object spaces, called contexts and controls the access to shareable interfaces of these objects
- It is then executed on JVCM which is scaled down version of standard JVM (Java Virtual Machine)

Global System for Mobile Communications:

- Originally GSM stood for Groupe Speciale Mobile GSM to meet the following business objectives:

1. Support for international roaming
2. Good speech quality
3. Ability to support handheld terminals
4. Low terminal and service cost
5. Spectral efficiency

6. Support for a range of new services and facilities
7. ISDN compatibility

GSM Timeline

1982	Groupe Spécial Mobile (GSM) established
1987	Essential elements of wireless transmission specified
1989	GSM become an ETSI technical committee
1990	Phase 1 GSM 900 specification (designed 1987 through 1990) frozen
1991	First GSM network launched
1993	First roaming agreement came into effect
1994	Data transmission capability launched
1995	Phase 2 launched. Fax and SMS roaming services offered
2002	SMS volume crosses 24 billions/year, 750 millions subscribers

Use of TDMA and FDMA in GSM

- Uses a combination of FDMA (Frequency Division Multiple Access) and TDMA (Time Division Multiple Access)
- Allocation of 50 MHz (890–915 MHz and 935–960 MHz) bandwidth in the 900 MHz frequency band and using FDMA further divided into 124 (125 channels, 1 not used) channels each with a carrier bandwidth of 200 KHz
- Using TDMA, each of the above mentioned channels is then further divided into 8 time slots
- So, with the combination of FDMA and TDMA, a maximum of992 channels for transmit and receive can be realized

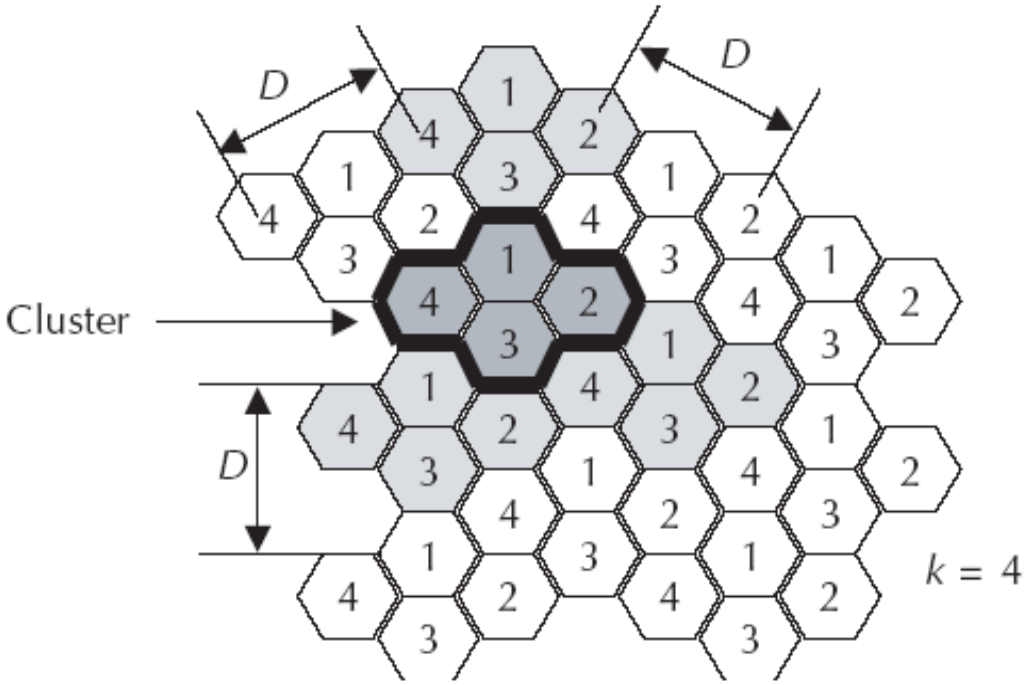
Frequency reuse in GSM

- To serve hundreds of thousands of users, the frequency must be reused and this is done through cells.
- The area to be covered is subdivided into radio zones or cells. Though in reality these cells could be of any shape, for convenient modeling purposes these are modeled as hexagons. Base stations are positioned at the center of these cells.
- Each cell *i* receives a subset of frequencies *fbi* from the total set assigned to the respective mobile network. To avoid any type of co-channel interference, two neighboring cells never use the same frequencies.

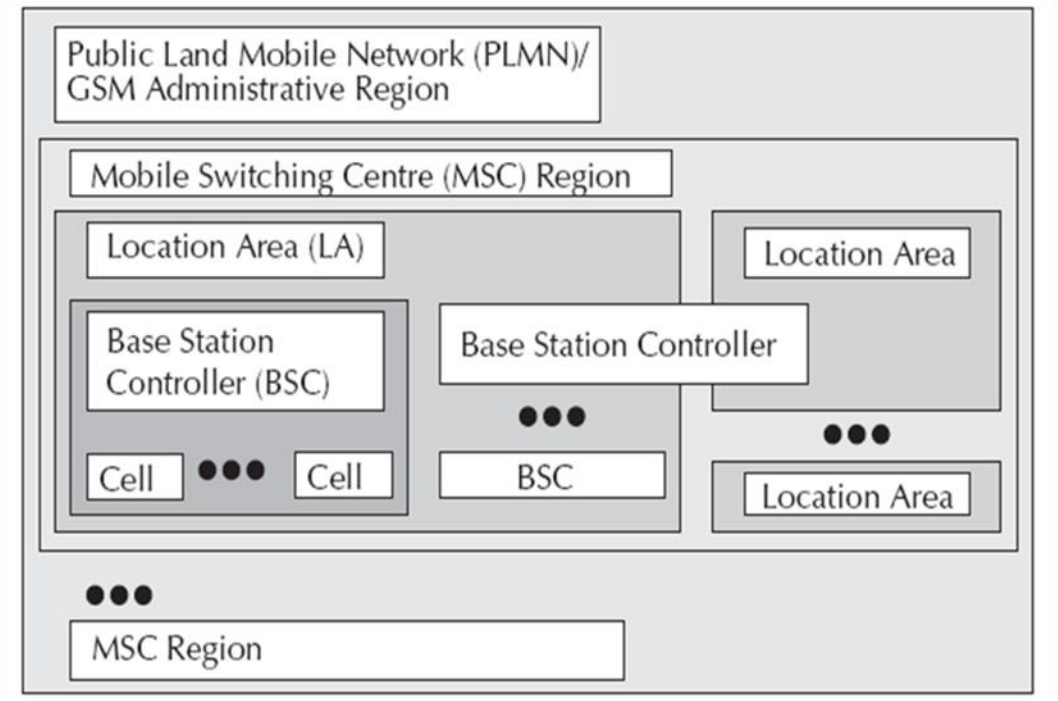
- Only at a distance of *D* (known as frequency reuse distance), the same frequency from the set *fbi* can be reused. Cells with distance *D* from cell *i*, can be assigned one or all the frequencies from the set *fbi* belonging to cell *i*.

- When moving from one cell to another during an ongoing conversation, an automatic channel change occurs. This phenomenon is called handover. Handover maintains an active speech and data connection over cell boundaries.
- The regular repetition of frequencies in cells result in a clustering of cells. The clusters generated in this way can consume the whole frequency band. The size of a cluster is defined by *k*, the number of cells in the cluster. This also defines the frequency reuse distance *D*. The figure in next slide shows an example of cluster size of 4.

Cell clusters in GSM:

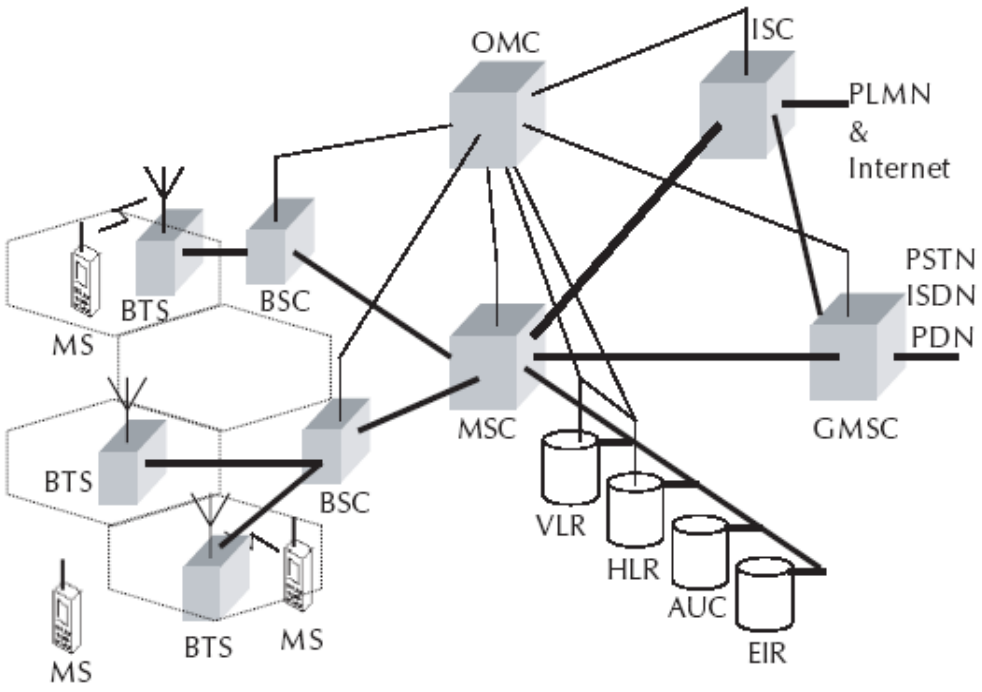


GSM System Hierarchy



- Consists at the minimum one administrative region assigned to one MSC (Mobile Switching Centre)
- Administrative region is commonly known as PLMN (Public Land Mobile Network)
- Each administrative region is subdivided into one or many Location Area (LA)
- One LA consists of many cell groups and each cell group is assigned to one BSC (Base Station Controller)
- For each LA, there will be at least one BSC while cells in one BSC can belong to different LAs

GSM Architecture

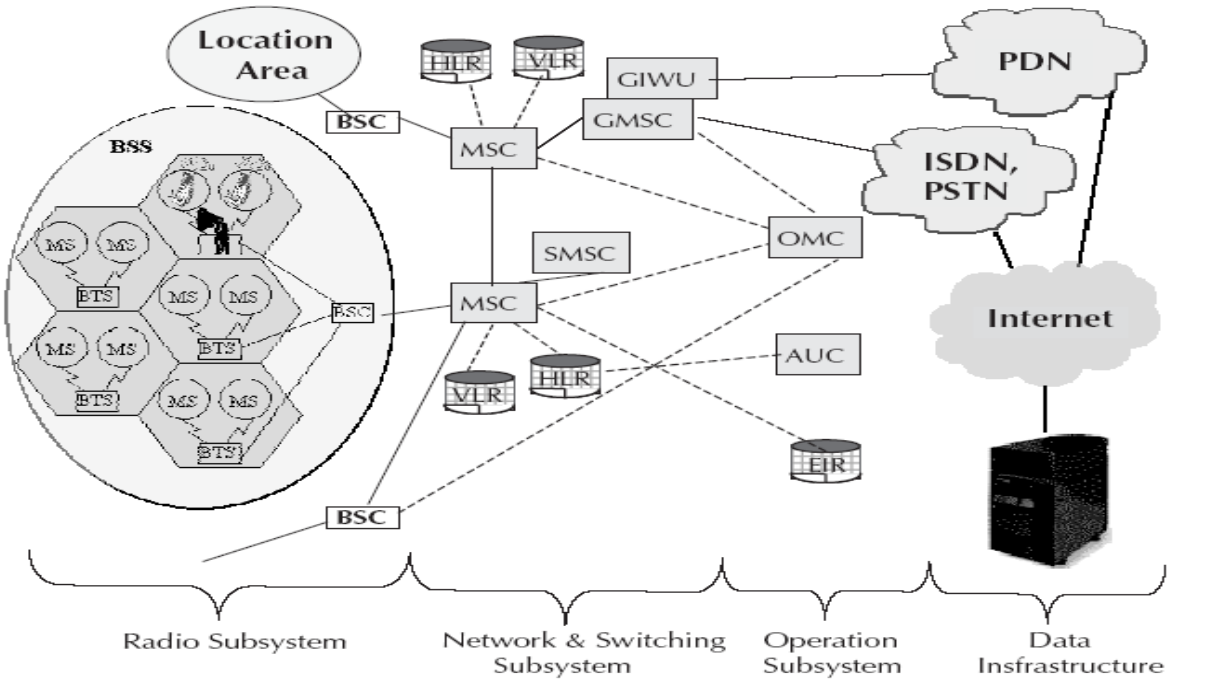


BTS	Base Transceiver Station
BSC	Base Station Controller
MSC	Mobile Switching Center
GMSC	Gateway MSC
ISC	International Switching Center

MS	Mobile Station
HLR	Home Location Register
VLR	Visitor Location Register
AUC	Authentication Center
EIR	Equipment Identity Register
OMC	Operation and Maintenance Center

- Cells are formed by the radio areas covered by a BTS (Base Transceiver Station)
- Several BTSs are controlled by one BSC
- Traffic from the MS (Mobile Station) is routed through MSC
- Calls originating from or terminating in a fixed network or other mobile networks is handled by the GMSC (Gateway MSC)

Operational Architecture of GSM:



BTS	Base Transceiver Station	ISDN	Integrated System Digital Network
BSC	Base Station Controller	HLR	Home Location Register
MSC	Mobile Switching Center	VLR	Visitor Location Register
GMSC	Gateway MSC	AUC	Authentication Center
SC	Service Center/SMS Controller	EIR	Equipment Identity Register
GIWU	Gateway Interworking Unit	OMC	Operation and Maintenance Center
PDN	Public Data Network	PSTN	Public Switched Telephone Network
MS	Mobile Station		

Home Location Register (HLR) in GSM

- It contains the following information:

1. Authentication information like International Mobile Subscriber Identity (IMSI)
2. Identification information like name, address, etc. of the subscriber
3. Identification information like Mobile Subscriber ISDN (MSISDN) etc.
4. Billing information like prepaid or postpaid
5. Operator selected denial of service to a subscriber
6. Handling of supplementary services like for CFU (Call Forwarding Unconditional), CFB (Call Forwarding Busy), CFNR (Call Forwarding Not Reachable) or CFNA (Call Forwarding Not Answered)
7. Storage of SMS Service Center (SC) number in case the mobile is not connectable so that whenever the mobile is connectable, a paging signal is sent to the SC
8. Provisioning information like whether long distance and international calls allowed or not
9. Provisioning information like whether roaming is enabled or not
10. Information related to auxiliary services like Voice mail, data, fax services, etc.
11. Information related to auxiliary services like CLI (Caller Line Identification), etc.
12. Information related to supplementary services for call routing. In GSM network, one can customize the personal profile to the extent that while the subscriber is roaming in a

foreign PLMN, incoming calls can be barred. Also, outgoing international calls can be barred, etc.

13. Some variable information like pointer to the VLR, location area of the subscriber, Power OFF status of the handset, etc.

Entities in GSM

- The Mobile Station (MS) - This includes the Mobile Equipment (ME) and the Subscriber Identity Module (SIM).
- The Base Station Subsystem (BSS) - This includes the Base Transceiver Station (BTS) and the Base Station Controller (BSC).
- The Network and Switching Subsystem (NSS) - This includes Mobile Switching Center (MSC), Home Location Register (HLR), Visitor Location Register (VLR), Equipment Identity Register (EIR), and the Authentication Center (AUC).
- The Operation and Support Subsystem (OSS) - This includes the Operation and Maintenance Center (OMC).

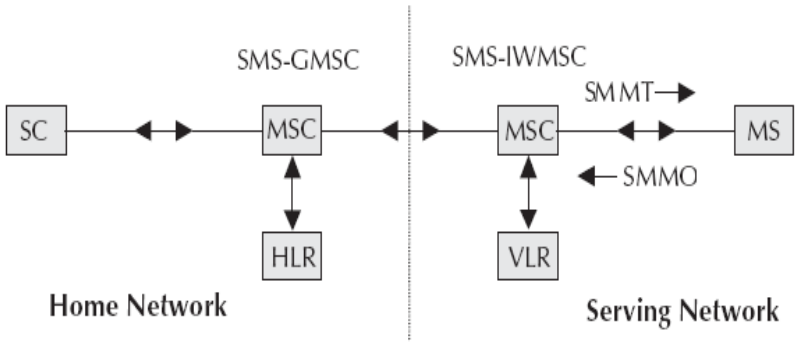
Short Message Service:

- Short Message Service (SMS) is one of the most popular services within GSM
- SMS is a data service and allows a user to enter text message up to 160 characters in length when 7 bit English characters are used
- SMS is a proactive bearer and is an 'always on' network
- Message center is referred to as Service Centre (SC) or SMS Controller (SMSC)
- SMSC is a system within the core GSM network which works as the store and forward system for SMS messages.

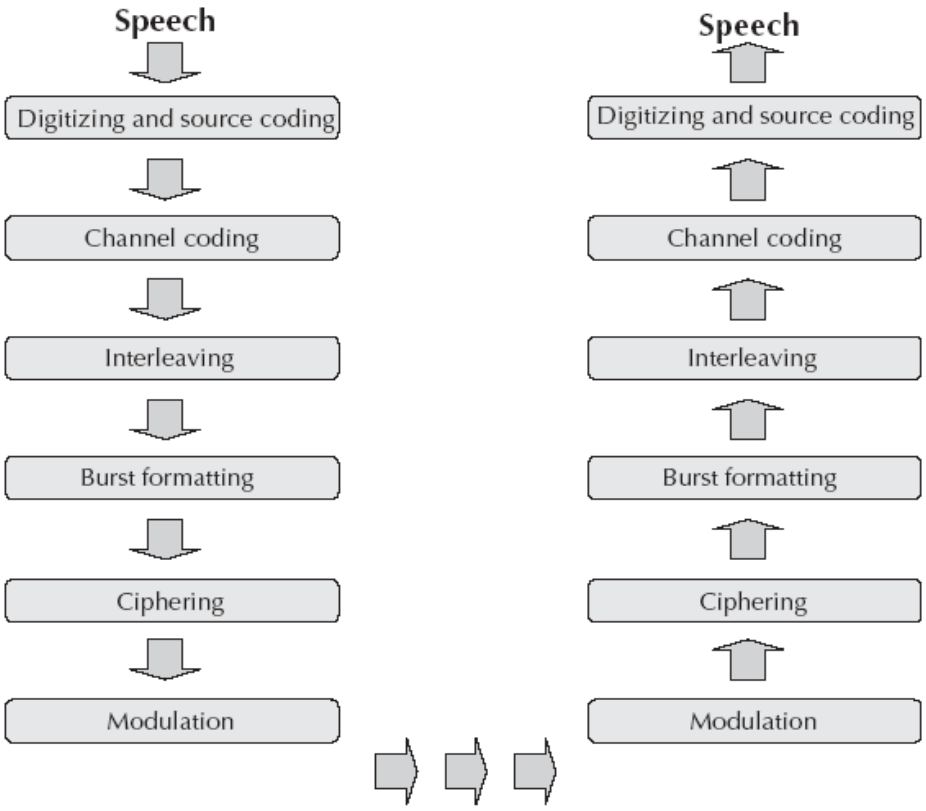
SMS

- Two types of SMS: SMMT (Short Message Mobile Terminated Point-to-Point) and SMMO (Short Message Mobile Originated Point-to-Point)
- SMMT is an incoming short message from the network and is terminated in the MS (phone or Mobile Station)
- SMMO is an outgoing message originated in the MS, and forwarded to the network for delivery
- For an outgoing message, the SMS is sent from the phone to SC via the VLR and the Inter Working MSC (IWMSC)
- For incoming message, the path is from SC to the MS via the HLR and the Gateway MSC (GMSC)

SMS Transfer :



From speech to radio waves



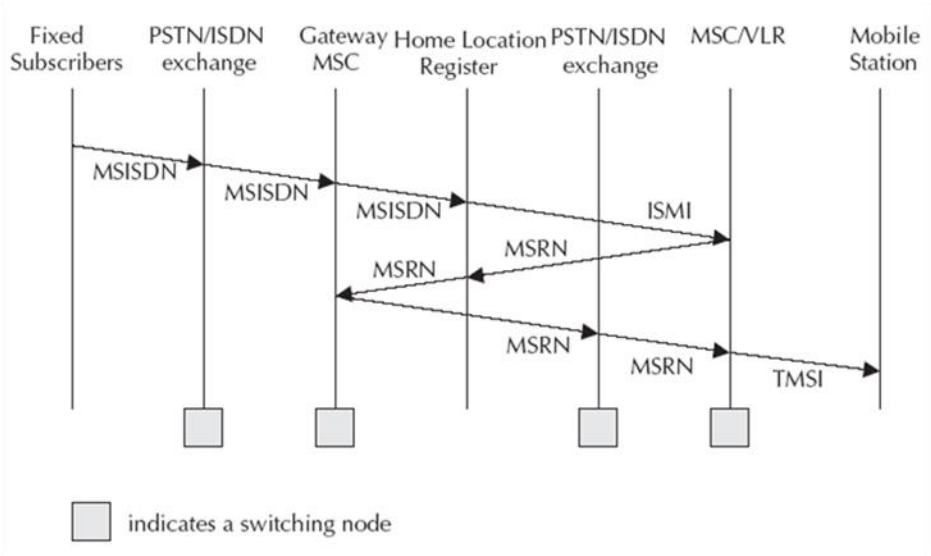
- Digitizer and source coding: The user speech is digitized at 8 KHz sampling rate using Regular Pulse Excited–Linear Predictive Coder (RPE–LPC) with a Long Term Predictor loop where information from previous samples is used to predict the current sample. Each sample is then represented in signed 13-bit linear PCM value. This digitized data is passed to the coder with frames of 160 samples where encoder compresses these 160 samples into 260-bits GSM frames resulting in one second of speech compressed into 1625 bytes and achieving a rate of 13 Kbits/sec.
- Channel coding: This introduces redundancy into the data for error detection and possible error correction where the gross bit rate after channel coding is 22.8 kbps (or 456 bits every 20 ms). These 456 bits are divided into eight 57-bit blocks and the result is interleaved amongst eight successive time slot bursts for protection against burst transmission errors.
- Interleaving: This step rearranges a group of bits in a particular way to improve the performance of the error-correction mechanisms. The interleaving decreases the possibility of losing whole bursts during the transmission by dispersing the errors.
- Ciphering: This encrypts blocks of user data using a symmetric key shared by the mobile station and the BTS.

- Burst formatting: It adds some binary information to the ciphered block for use in synchronization and equalization of the received data.
- Modulation: The modulation technique chosen for the GSM system is the Gaussian Minimum Shift Keying (GMSK) where binary data is converted back into analog signal to fit the frequency and time requirements for the multiple access rules. This signal is then radiated as radio wave over the air.
- Multipath and equalization: An equaliser is in charge of extracting the 'right' signal from the received signal while estimating the channel impulse response of the GSM system and then it constructs an inverse filter. The received signal is then passed through the inverse filter.
- Synchronization: For successful operation of a mobile radio system, time and frequency synchronization are needed. Frequency synchronization is necessary so that the transmitter and receiver frequency match (in FDMA) while Time synchronization is necessary to identify the frame boundary and the bits within the frame (in TDMA).
- To avoid collisions of burst transmitted by MS with the adjacent timeslot such collisions, the Timing Advance technique is used where frame is advanced in time so that this offsets the delay due to greater distance. Using this technique and the triangulation of the intersection cell sites, the location of a mobile station can be determined from within the network.

Call Routing:

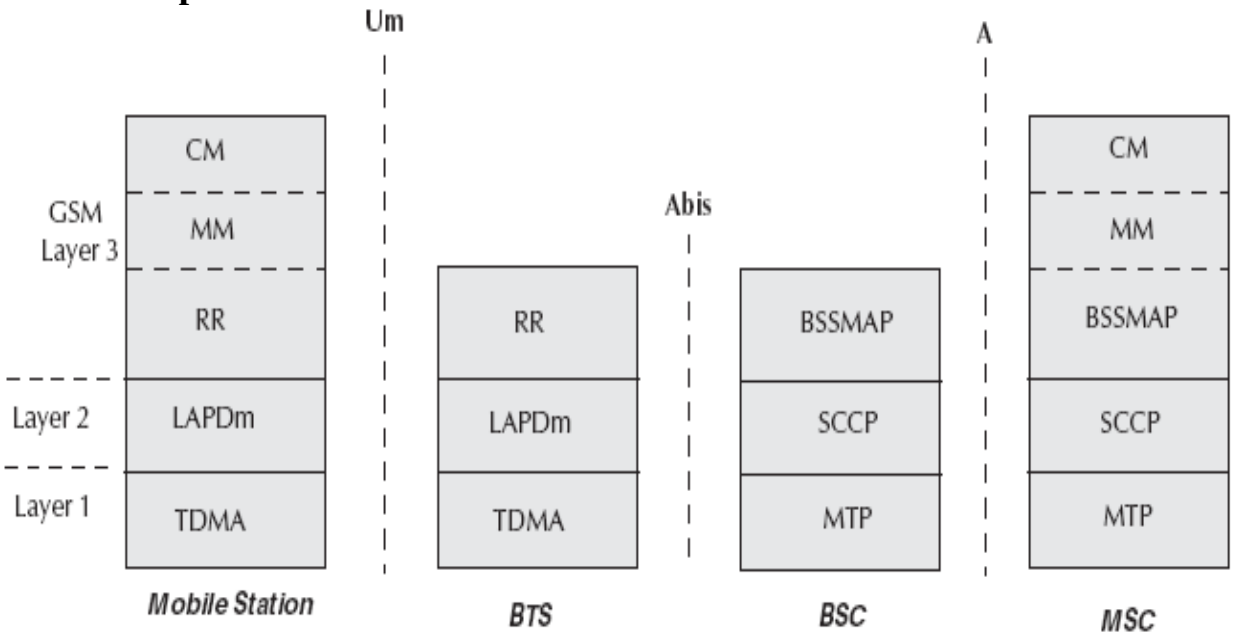
- The directory number dialed to reach a mobile subscriber is called the Mobile Subscriber ISDN (MSISDN) which is defined by the E.164 numbering plan and includes a country code and a National Destination Code, which identifies the subscriber's operator. The first few digits of the remaining subscriber number may identify the subscriber's HLR within the home PLMN
- For example, the MSISDN number of a subscriber in Bangalore associated with Airtel network is +919845XYYYYY which is a unique number and understood from anywhere in the world. Here, + means prefix for international dialing, 91 is the country code for India and 45 is the network operator's code (Airtel in this case). X is the level number managed by the network operator ranging from 0 to 9 while YYYYY is the subscriber code which , too, is managed by the operator.

Call Routing



- The call first goes to the local PSTN exchange where PSTN exchange looks at the routing table and determines that it is a call to a mobile network.
- PSTN forwards the call to the Gateway MSC (GMSC) of the mobile network.
- MSC enquires the HLR to determine the status of the subscriber. It will decide whether the call is to be routed or not. If MSC finds that the call can be processed, it will find out the address of the VLR where the mobile is expected to be present.
- If VLR is that of a different PLMN, it will forward the call to the foreign PLMN through the Gateway MSC. If the VLR is in the home network, it will determine the Location Area (LA).
- Within the LA, it will page and locate the phone and connect the call.

Network aspects in GSM



- Layer 3 of the GSM signaling protocol is itself divided into three sub-layers:

1. Radio Resources Management: It controls the set-up, maintenance and termination of radio and fixed channels, including handovers.
2. Mobility Management: It manages the location updating and registration procedures as well as security and authentication.
3. Connection Management: It handles general call control and manages Supplementary Services and the Short Message Service.

Handover

- The procedure of change of resources is called handover when the user is mobile while the call is in progress.
- There are four different types of handover in the GSM system, which involve transferring a call between:

1. Channels (time slots) in the same cell
2. Cells (Base Transceiver Stations) under the control of the same Base Station Controller (BSC)
3. Cells under the control of different BSCs but belonging to the same Mobile Switching Center (MSC)
4. Cells under the control of different MSCs.

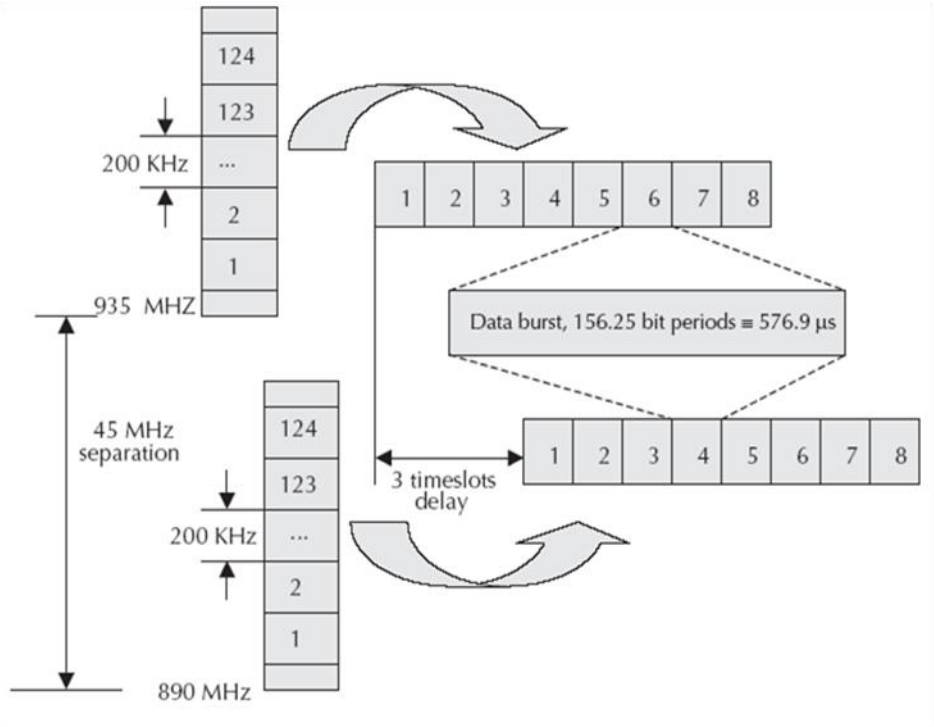
- First two types of handover, called internal handovers, involve only one Base Station Controller (BSC). To save signaling bandwidth, they are managed by the BSC without involving the Mobile services Switching Center (MSC), except to notify it at the completion of the handover.
- Last two types of handover, called external handovers, are handled by the MSC.

GSM Frequency Allocation

- Normally, GSM uses 900 MHz band wherein 890-915 MHz is allocated for the uplink (mobile station to base station) and 935–960 MHz is allocated for the downlink (base station to mobile station). Each way the bandwidth for the GSM system is 25 MHz which provides 125 carriers uplink/downlink each having a bandwidth of 200 kHz.
- ARFCN (Absolute Radio Frequency Channel Numbers) denote a forward and reverse channel pair which is separated in frequency by 45 MHz.
- Practically, a guard band of 100 kHz is provided at the upper and lower end of the GSM 900 MHz spectrum and only 124 (duplex) channels are implemented.

- GSM uses a combination of Time Division Multiple Access (TDMA) and Frequency Division Multiple Access (FDMA) encoding.
- One or more carrier frequencies are assigned to each base station and each of these carrier frequencies is then divided in time using a TDMA scheme where fundamental unit is called a burst period lasting approximately 0.577 ms.
- Eight burst periods are grouped into a TDMA frame of approximately 4.615 ms which forms the basic unit for the definition of logical channels.
- One physical channel is one burst period per TDMA frame while, normally, channels are defined by the number and position of their corresponding burst periods.

GSM Frequency Allocation



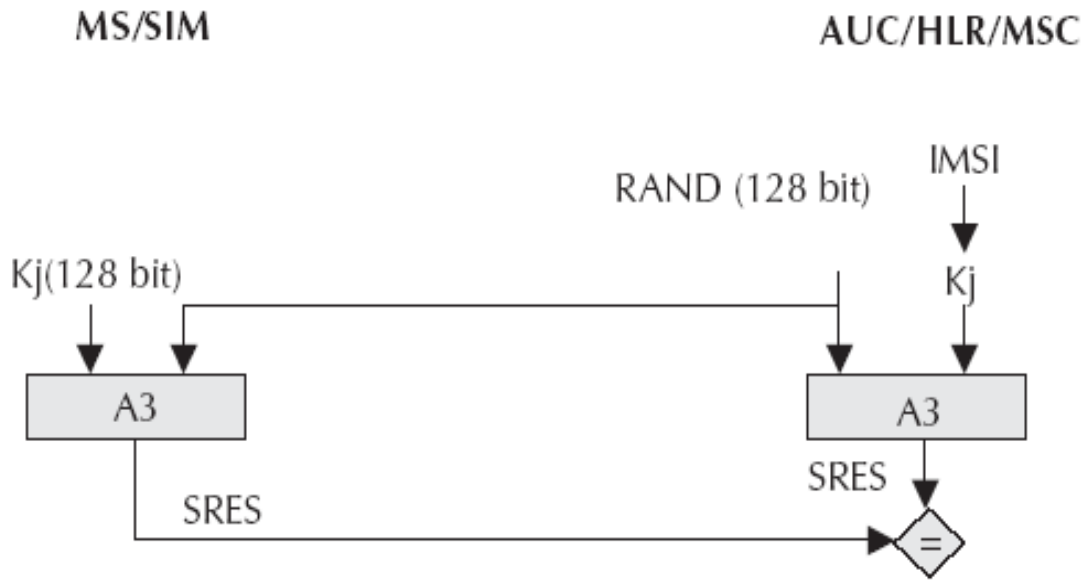
Authentication and Security

- Authentication involves two functional entities - the SIM card in the mobile phone and the Authentication Center (AUC).
- Following authentication by algorithm A3, a key is generated for encryption.
- An algorithm A8 is used to generate the key while a different algorithm called A5 is used for both ciphering and deciphering procedures for signaling, voice and data.
- So, SS7 signal, voice, data, and SMS within GSM networks are ciphered over the wireless radio interface.

A3 Algorithm

- During authentication, MSC challenges the MS with a random number (RAND).
- SIM card uses this RAND received from the MSC and a secret key Kj stored within the SIM as input. Both the RAND and the Kj secret are 128 bits long. Using the A3 algorithm with RAND and Kj as input a 32-bit output called signature response (SRES) is generated in the MS and then sent back to MSC.
- Using the same set of algorithms, the AUC also generates a SRES. The SRES from MS and the SRES generated by the AUC are compared.
- If they are the same, the MS is authenticated.

- The idea is that no keys will be transacted over the air. However, if the SRES values calculated independently by the SIM and the AUC are the same, then Kj has to be same and if Kj is same, SIM card is genuine.



Short Message Service (SMS)

- Most popular data bearer/service within GSM
- More than one billion SMS messages interchanged everyday with a growth of more than half a billion every month on an average
- Runs on SS7 signaling channels, which are always present but mostly unused, be it during an active user connection or in the idle state
- Each short message is up to 160 characters in length when 7-bit English characters are used and 140 octets when 8-bit characters are used

Strengths of SMS

- Omnibus nature of SMS: SMS uses SS7 signaling channel which is available throughout the world.
- Stateless: SMS is session-less and stateless as every SMS message is unidirectional and independent of any context. This makes SMS the best bearer for notifications, alerts and paging.
- Asynchronous: SMS is completely asynchronous. In case of SMS, even if the recipient is out of service, the transmission will not be abandoned and hence, SMS can be used as message queues. SMS can be used as a transport bearer for both synchronous (transaction oriented) and asynchronous (message queue and notification) information exchange.

- Self-configurable and last mile problem resistant: SMS is self-configurable and subscriber is always connected to the SMS bearer irrespective of the home and visiting network configurations.
- Non-repudiable: SMS message carries the Service Center (SC) and the source MSISDN as a part of the message header through which any SMS can prove beyond doubt its origin.
- Always connected: As SMS uses the SS7 signaling channel for its data traffic, the bearer media is always on. Users cannot switch OFF, BAR or DIVERT any SMS message. SMS message is delivered to the Mobile Station (MS) without any interruption to the ongoing call.

SMS Architecture

- Two types of SMS - **SM MT** (Short Message Mobile Terminated Point-to-Point) and **SM MO** (Short Message Mobile Originated Point-to-Point)
- SM MT is an incoming short message from the network and is terminated in the MS
- SM MO is an outgoing message originated in the MS and forwarded to the network for delivery
- For an outgoing message, the path is from MS to SC via the VLR and the IWMSC (Inter Working MSC) function of the serving MSC whereas for an incoming message the path is from SC to the MS via HLR and the GMSC (Gateway MSC) function of the home MSC

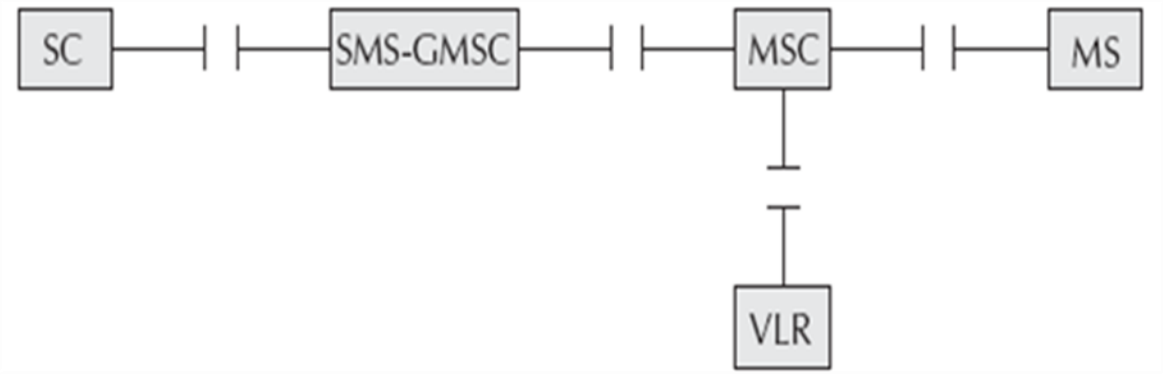
Strengths of SMS

- Omnibus nature of SMS: SMS uses SS7 signaling channel which is available throughout the world.
- Stateless: SMS is session-less and stateless as every SMS message is unidirectional and independent of any context. This makes SMS the best bearer for notifications, alerts and paging.

- Asynchronous: SMS is completely asynchronous. In case of SMS, even if the recipient is out of service, the transmission will not be abandoned and hence, SMS can be used as message queues. SMS can be used as a transport bearer for both synchronous (transaction oriented) and asynchronous (message queue and notification) information exchange.

Short Message Mobile Terminated (SMMT)

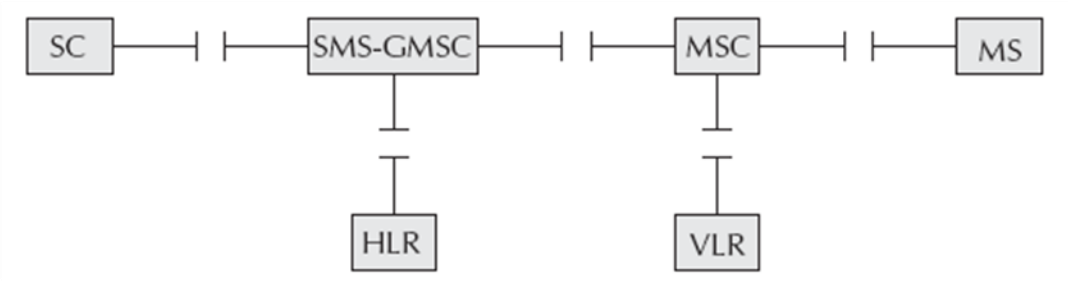
- Message is sent from SC to the MS.
- For the delivery of MT or incoming SMS messages, the SC of the serving network is never used which implies that a SMS message can be sent from any SC in any network to a GSM phone anywhere in the world.



Interfaces in SMMT

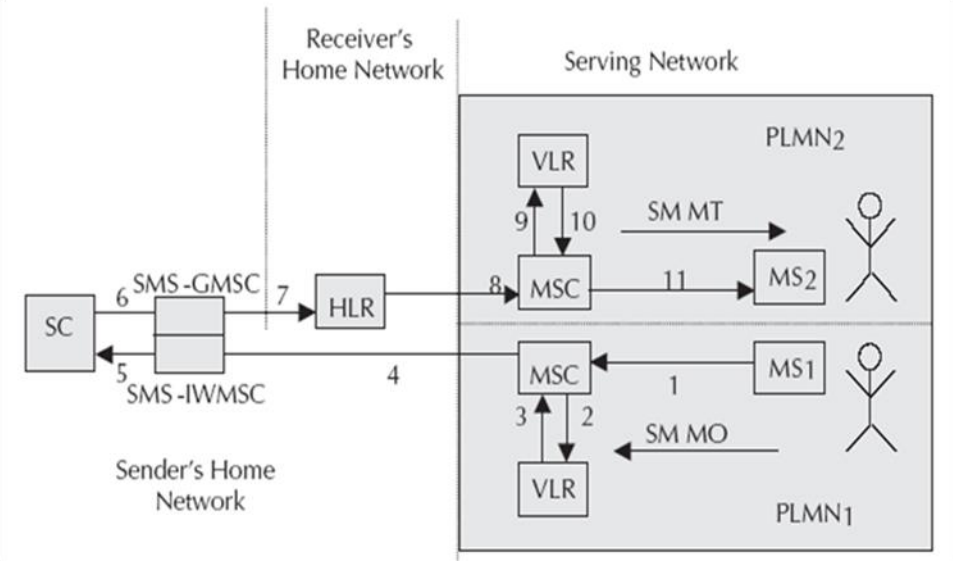
Short Message Mobile Originated

- For a MO message, the MSC forwards the message to the home SC.
- MO message works in two asynchronous phases. In the first phase, the message is sent from the MS to the home SC as a MO message. In the second phase, the message is sent from the home SC to the MS as a MT message.



Interfaces in SMMO

SMS Transfer



SMS as an Information Bearer

- For using SMS as an information bearer, we need to connect the services running on the Enterprise Origin server to the SC through an SME (Short Message Entity) or ESME (External Short Message Entity).
- SME in any network is generally a SMS gateway.
- With respect to SMS, a GSM subscriber is always in control of the SC in the home network irrespective of the serving network.
- If there is any SMS-based data service in the home network, it will be available in any foreign network.

Value Added Services through SMS

- Value Added Services (VAS) can be defined as services, which share one or more of the following characteristics:

1. Supplementary service (not a part of basic service) but adds value to total service offering
2. Stimulates incremental demand for core services offering
3. Stands alone in terms of profitability and revenue generation potential
4. Can sometimes stand-alone operationally
5. Does not cannibalize basic service unless clearly favorable
6. Can be an add-on to basic service, and as such, may be sold at a premium price
7. May provide operational and/or administrative synergy between or among other services and not merely for diversification

VAS over SMS are entertainment and information on demand which is further categorized into:

1. Static information which does not change frequently
2. Dynamic information which changes in days
3. Real-time information which changes continually

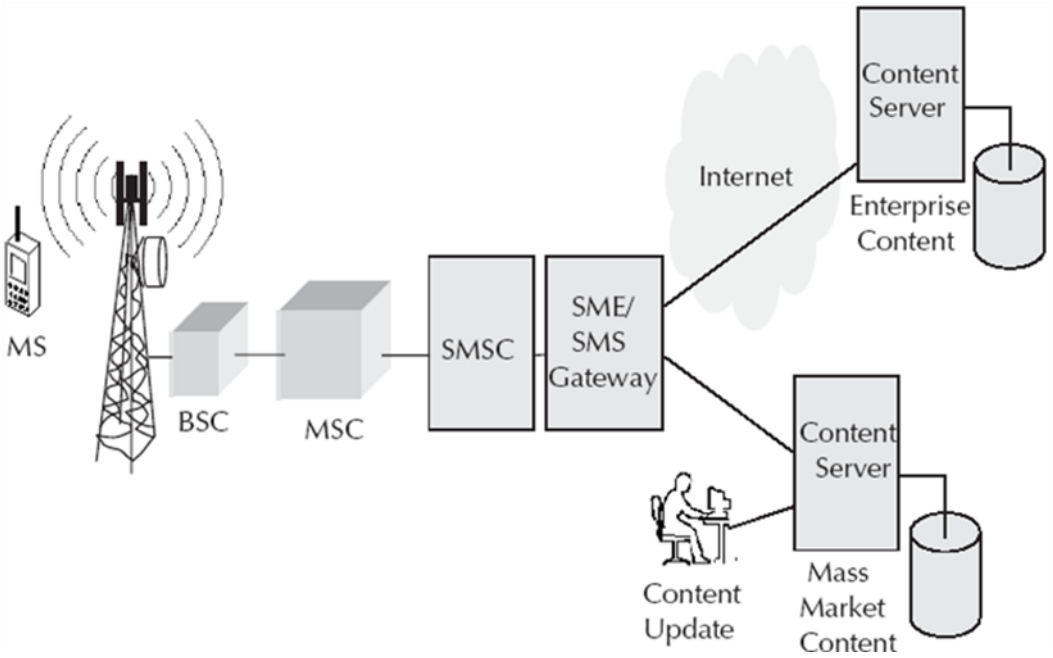
- Some of the common VAS examples are:

1. News/Stock Quotes Service
2. Session-based Chat Application
3. Email through SMS
4. Health Care Services
5. Micro-Payment Services

Alert services through VAS

- Proactive alert services can be of the two kinds – Time based and Watermark based
- Time based proactive alerts are sent to the mobile phone at a pre-assigned time of the day
- Watermark based proactive alerts are sent when some event occurs

VAS Architecture



UNIT II - POSSIBLE QUESTIONS

PART – B
(Each Question carries six marks)

1. Write about Bluetooth protocol stack with diagram.
2. Explain GSM architecture with suitable diagram.
3. Explain RFID transponders with suitable example.
4. Write a detailed notes on the value added service through SMS.
5. Write about the WIMAX architecture with suitable example.
6. Explain the Java card application with suitable diagram.
7. Explain the Mobile IP architecture with suitable example.
8. Explain the architecture of Java card with suitable diagram.
9. Explain the GSM System hierarchy with suitable diagram.
10. Explain the Short Messaging Service and architecture with suitable diagram.

PART – C
Compulsory Question
(Each question carries ten marks)

1. What should be the characteristics of Mobile Computing Devices?
2. Describe the security aspects in mobile computing.
3. Explain the functionality of Mobile Computing
4. How would you broadly classify the mobile computing applications?
5. Who are all the stakeholders of wireless network?
6. What is WiMax? How it is differ from WiFi?
7. Describe 3G networks.
8. Explain the Process of GSM Call Routing.
9. Explain the Process of Interactive Voice Response.
10. Briefly describe the Eclipse IDE for the development of android application.

MOBILE COMPUTING (18CAP402) UNIT- II

S.No	Question	Option1	Option2	Option3	Option4	Answer
1	_____is the next generation Internet protocol	IPv2	IPv6	IPv4	IPv1	IPv6
2	_____is emerging as a leading technology in the logistics,manufacturing and ret	Bluetooth	WiMax	RFID	WAP	RFID
3	_____is a technology in the Personal Area Network	RFID	Bluetooth	WiMax	GSM	Bluetooth
4	Bluetooth enabled chips can easily transfer data at a speed of about_____in ba	5 Mbps	1Kbps	1 Mbps	1.5 Mbps	1 Mbps
5	Bluetooth uses unlicensed_____frequency band	2.4 GHZ ISM	1 GHZ ISM	4 GHZ ISM	3.5 GHZ ISM	2.4 GHZ ISM
6	_____supports both unicast and multicast connections	WiMax	GSM	CDMA	Bluetooth	Bluetooth
7	_____protocol uses the concept of both master and slave	Internet	SMTP	FTP	Bluetooth	Bluetooth
8	The master and slave protocol together form a _____	Ethernet	Internet	Piconet	MANET	Piconet
9	The network of piconet is called _____	Piconet	Scatternet	WANET	MANET	Scatternet
10	Bluetooth uses publicly available cipher algorithm known as _____	RSA	ASCII	ethical	SAFER +	SAFER +
11	RFID tags are categorized into_____criteria	2	3	4	5	3
12	The earliest use of RFID was for tracking _____	Farm animals	Elephants	Tigers	Deers	Farm animals
13	_____is a costlier technology	RFID	WiMax	Bluetooth	GSM	RFID
14	In RFID_____tags are generally in low frequency	Active	Passive	Power based Gro	Active and Passive	Active
15	RFID tags areof different _____	Varieties	Shapes	Sizes	Shapes and Sizes	Shapes and Sizes
16	RFID Tags can be screw shaped to identify_____	Animal tagging	Postal tracking	trees or wooden	Retail store	trees or wooden
17	RFID components consists of_____components	1	3	5	7	3
18	In RFID the reader emits radio waves in any range from one centimeter to	35	15	25	11	25
19	In RFID a_____programmed with unique information	antenna	transponder	transceiver	coil	transponder
20	In RFID_____emits radio signals to read and write data into the tag	transponder	tranceiver	receiver	antenna	antenna
21	In RFID,standardization helped GSM and_____technology to be widely accep	CDMA	FDMA	TDMA	Barcode	Barcode
22	WLL stands for _____	Wireless Local Lo	Wireless in Local L	Wired Local Loop	Wired in Local Loop	Wireless Local Loo
23	WiLL stands for_____	Wired Local Loop	Wired in Local Loo	Wireless in Local	Wireless Local Loop	Wireless in Local L
24	_____is also known as fixed-wireless system	Wireless in Local	Wired Local Loop	Wired in Local Lo	Wireless Local Loop	Wireless Local Loo
25	The world is moving towards a convergence of _____	voice,data,video	electromagnetic sig	Digital signals	Analog signals	voice,data,video

26	_____offers an alternative to high bandwidth wired access networks like fibre op	Wireless MAN	Wireless WAN	Wireless LAN	Wireless LAN and W	Wireless MAN
27	_____is popularly known as WiMax	Wireless WAN	Wireless LAN and	Wireless LAN	Wireless MAN	Wireless MAN
28	WiMax technology provides_____bandwidth without the need for cables	5 Mbps	10 Mbps	15 Mbps	20 Mbps	10 Mbps
29	In a wireless environment the number of channels is always_____compared to	high	low	medium	very high	low
30	_____standardizes the air interfaces and related functions associated with WLL	IEEE 802.16.2	IEEE 802.16	IEEE 802.16.1	IEEE 802.16.3	IEEE 802.16
31	IEEE 802.16.1 produces air interface for 10 to_____GHz	55	33	66	77	66
32	_____standardizes the air interface for licensed frequencies 2 to 11 GHz	IEEE 802.16.3	IEEE 802.16	IEEE 802.16.2	IEEE 802.16.1	IEEE 802.16.3
33	IEEE 802.16 can serve as a backbone for_____networks	IEEE 802.16.1	IEEE 802.16.2	IEEE 802.11	IEEE 802.16.3	IEEE 802.11
34	IEEE 802.16 standards are organized into a_____layered architecture	3	2	4	5	3
35	The_____layer specifies the error correction techniques in IEEE 802.16	MAC	Physical	Network	Transport	Physical
36	_____layer is responsible for transmitting data in frames	Covergence	Network	Transport	MAC	MAC
37	Above the MAC layer is a_____layer	Transport	MAC	Covergence	Network	Covergence
38	_____deals with a situation where the user is at a vehicular state and accessing	MM	Mobile IP	Javacard	TCP/IP	MM
39	Vehicular state generally means moving at a speed of_____Kmph	50	40	60	30	60
40	_____port number is application specific and remains constant	TCP	IPv6	IPv4	MobileIP	TCP
41	_____is network specific port number	IPv2	IPv6	TCP	IP	IP
42	_____technology allow mobility while data connection is alive	Mobile computing	Mobile IP	TCP	Mobile telephony	Mobile IP
43	In_____network,when a user is mobile the point of base station changes	Mobile	IP	Cellular	MANET	Cellular
44	_____routes packets from a source endpoint to a desination endpoint	Node	Network	IP	Address	IP
45	For meaningful communication we need the_____port of the applications	Http	IP	FTP	UDP	UDP
46	MobileIP allows the mobile node to use two IP addresses called_____and	home ,care -off	office,station	station,roaming	roaming,local	home,careoff
47	_____can be thought of as the mobile node's location specific address	home	station	care-off	office	care-off
48	_____networks can be accomplished by tunneling	IPv2	IPv1	IPv4	IPv6	IPv6
49	The dual-stack routers run both_____and_____protocols	IPv2,IPv1	IPv2,IPv4	IPv4,IPv6	IPv1,IPv2	IPv4,IPv6
50	_____card is a smart card with Javaframework	sound	video	Sim	Java	Java
51	Smart card was developed in the year _____	1975	1972	1974	1970	1974

UNIT II NOTES

UNIT II – BLUETOOTH AND GSM

Bluetooth- Features and working of RFID - Wireless Broadband (WiMax)- Mobile IP – IPV6-Java Card –Global System for Mobile Communications – GSM Architecture – Call Routing in GSM – GSM Addresses and Identifiers – Network Aspects in GSM – GSM Frequency Allocation – Authentication and Security- Mobile Computing Over SMS – SMS-Value Added Services through SMS.

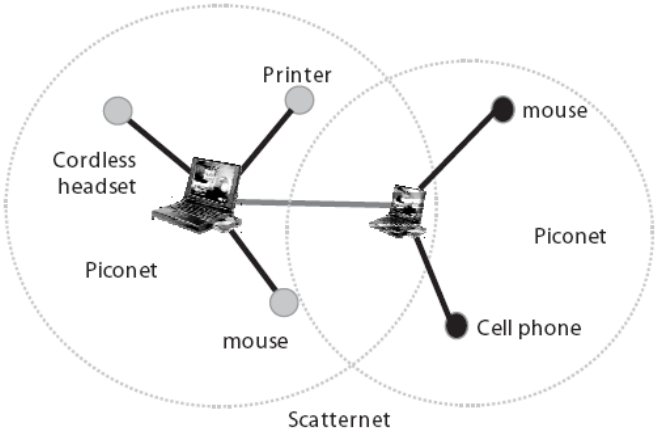
Bluetooth

- Name comes from nickname of Danish king Harald Blåtand
- Allows users to make ad hoc wireless connections between devices like mobile phones, desktop or notebook computers wirelessly
- Data transfer at a speed of about 720 Kbps within 50 meters (150 feet) of range or beyond through walls, clothing and even luggage bags
- Built into a small microchip
- Operates in a globally available frequency band ensuring worldwide interoperability
- Managed and maintained by Bluetooth Special Interest Group

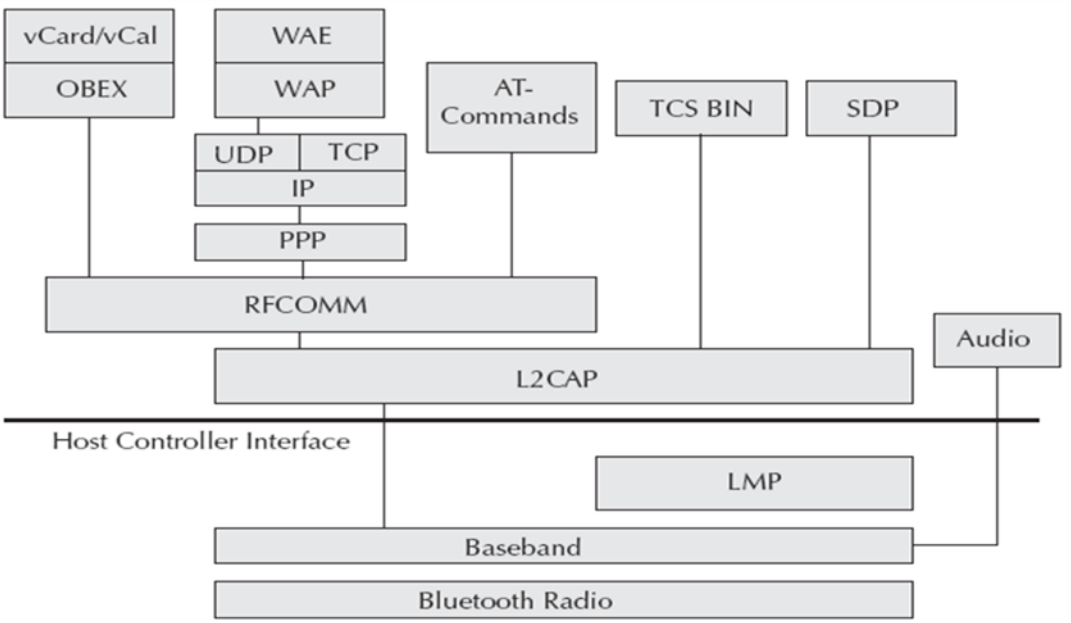
Bluetooth Protocol

- Uses the unlicensed 2.4 GHz ISM (Industrial Scientific and Medical) frequency band
- 79 available channels spaced 1 MHz apart from 2.402 GHz to 2.480 GHz
- Allows power levels starting from 1mW (covering 10 centimetres) to 100mW (covering upto 100 meters) suitable for short device zone to personal area networks within a home
- Supports both unicast (point-to-point) and multicast (point-to-multipoint) connections
- Bluetooth protocols are a collection of many inter-related protocols
- Uses the master and slave relationship
- Master and slaves together form a Piconet when master allows slaves to talk
- Up to seven ‘slave’ devices can be set to communicate with a ‘master’ in a Piconet.
- Scatternet is formed when several of piconets are linked together to form a larger network in an ad hoc manner

Scatternet is a topology where a device from one piconet also acts as a member of another piconet wherein a device being a master in one piconet can simultaneously be a slave in the other one.



- Bluetooth Core protocols plus Bluetooth radio protocols are required by most of Bluetooth devices
- Uses spread spectrum technologies at the Physical Layer while using both direct sequence and frequency hopping spread spectrum technologies
- Uses connectionless (ACL–Asynchronous Connectionless Link) and connection-oriented (SCO–Synchronous Connection-oriented Link) links
- Cable Replacement layer, Telephony Control layer and Adopted protocol layer form application-oriented protocols



OBEX – Object Exchange Protocol
WAE – Wireless Application Environment
WAP – Wireless Application Protocol
LMP – Link Manager Protocol
TCS BIN – Telephony Control Specification Binary
SDP – Service Discovery Protocol
RFCOMM – Radio Frequency Communication
L2CAP – Logical Link Control and Adaptation Protocol

Bluetooth Protocol Stack:

1. Bluetooth Core Protocols
2. Cable Replacement Protocol
3. Telephony Control Protocols
4. Adopted Protocols

- Baseband – enables physical RF link
- Link Manager Protocol (LMP) – manages devices in range, power modes, connections, duty cycles, etc.
- Logical Link Control and Adaptation Protocol (L2CAP) – segmentation and re-assembly of fragmented packets with their multiplexing
- Service Discovery Protocol (SDP) – Enables a device to join a piconet

Telephony Control Protocols

- Telephony Control Specification Binary (TCS BIN) - defines the call control signaling protocol and handles mobility management for groups of Bluetooth TCS devices

- Attention (AT) Commands - defines a set of commands by which a mobile phone can be used and controlled as a modem for fax and data transfers

Adopted Protocols:

- Point-to-Point Protocol (PPP) - means of taking IP packets to/from the PPP layer and placing them onto the LAN
- Transmission Control Protocol/Internet Protocol (TCP/IP) - used for communication across the Internet
- Object Exchange (OBEX) Protocol - session protocol to exchange objects and used to browse the contents of folders on remote devices
- Content Formats - used to exchange messages and notes and synchronize data amongst various devices
-

Bluetooth Security

- Offers security infrastructure starting from authentication, key exchange to encryption
- Uses the publicly available cipher algorithm known as SAFER+ to authenticate a device's identity

Bluetooth Application Models

Each application model in Bluetooth is realized through a Profile. Profiles define the protocols and protocol features supporting a particular usage model. Some common profiles are:

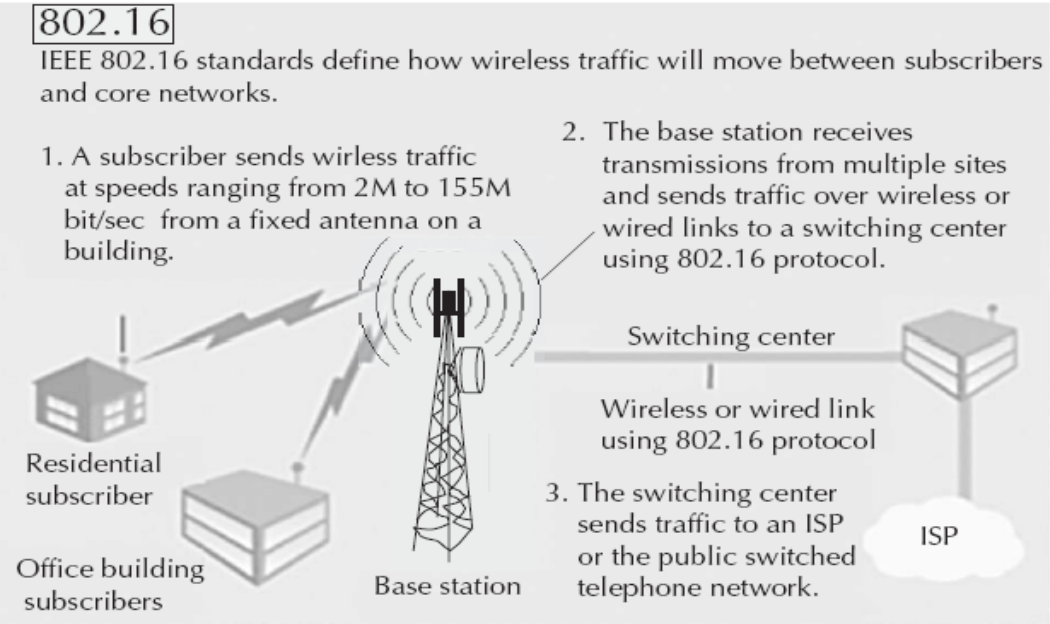
- File Transfer
- Internet Bridge
- LAN Access
- Synchronization
- Headset

Wireless Broadband:

- Also known as Wireless Metropolitan Area Network (Wireless MAN) and Wireless Microwave Access (WiMAX)
- IEEE 802.16 standard released in April 2002
- Offers an alternative to high bandwidth wired access networks like fiber optic, cable modems and DSL
- Provides network access to buildings through exterior antennas communicating with radio base stations

- Networks can be created in just weeks by deploying a small number of base stations on buildings or poles to create high capacity wireless access systems.

Overview of IEEE 802.16



Sub-standards of IEEE 802.16

- IEEE 802.16.1 - Air interface for 10 to 66 GHz
- IEEE 802.16.2 - Coexistence of broadband wireless access systems
- IEEE 802.16.3 - Air interface for licensed frequencies, 2 to 11 GHz

Basics of IEEE 802.16

IEEE 802.16 standards are concerned with the air interface between a subscriber's transceiver station and a base transceiver station

- The Physical Layer
- MAC Layer
- Convergence Layer

Physical Layer

- Specifies the frequency band, the modulation scheme, error-correction techniques, synchronization between transmitter and receiver, data rate and the multiplexing structure
- Both TDD and FDD alternatives support adaptive burst profiles in which modulation and coding options may be dynamically assigned on a burst-by-burst basis

- Three physical layer for services: Wireless MAN-SC2, Wireless MAN-OFDM and Wireless MAN-OFDMA

Medium Access Control Layer

- Designed for point-to-multipoint broadband wireless access
- Addresses the need for very high bit rates, both uplink (to the base station) and downlink (from the base station)
- Services like multimedia and voice can run as 802.16 MAC is equipped to accommodate both continuous and bursty traffic

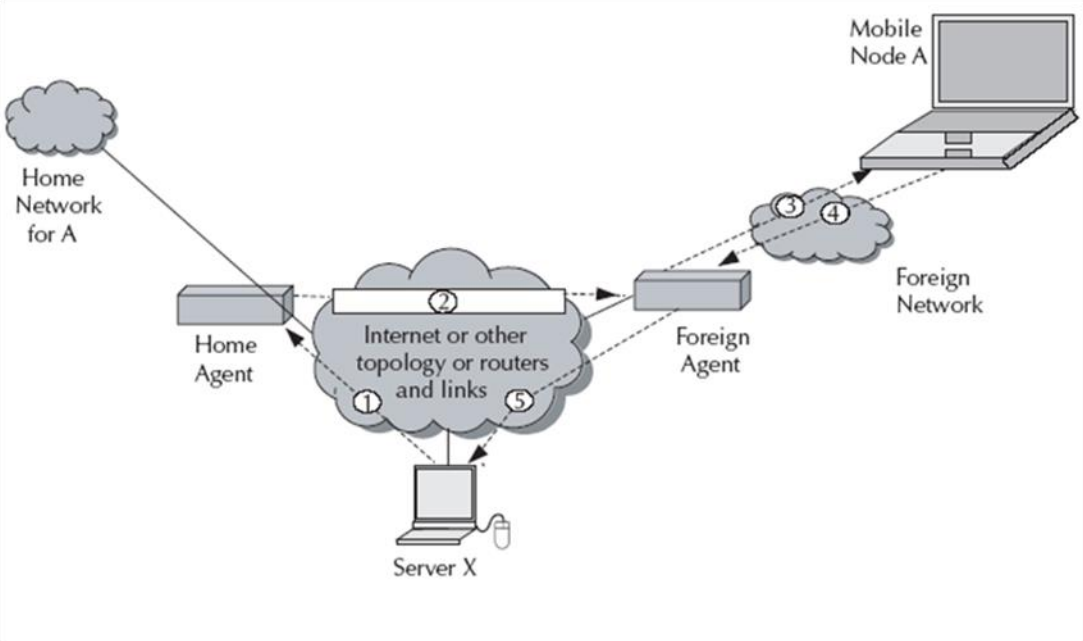
Convergence Layer

- Provides functions specific to the service being provided
- Bearer services include digital audio/video multicast, digital telephony, ATM, Internet access, wireless trunks in telephone networks and frame relay

Mobile IP:

- 'Mobile IP' signifies that, while a user is connected to applications across the Internet and the user's point of attachment changes dynamically, all connections are maintained despite the change in underlying network properties
- Similar to the handoff/roaming situation in cellular network
- Mobile IP allows the mobile node to use two IP addresses called home address and care of address
- The home address is static and known to everybody as the identity of the host
- The care of address changes at each new point of attachment and can be thought of as the mobile node's location specific address

Working of Mobile IP

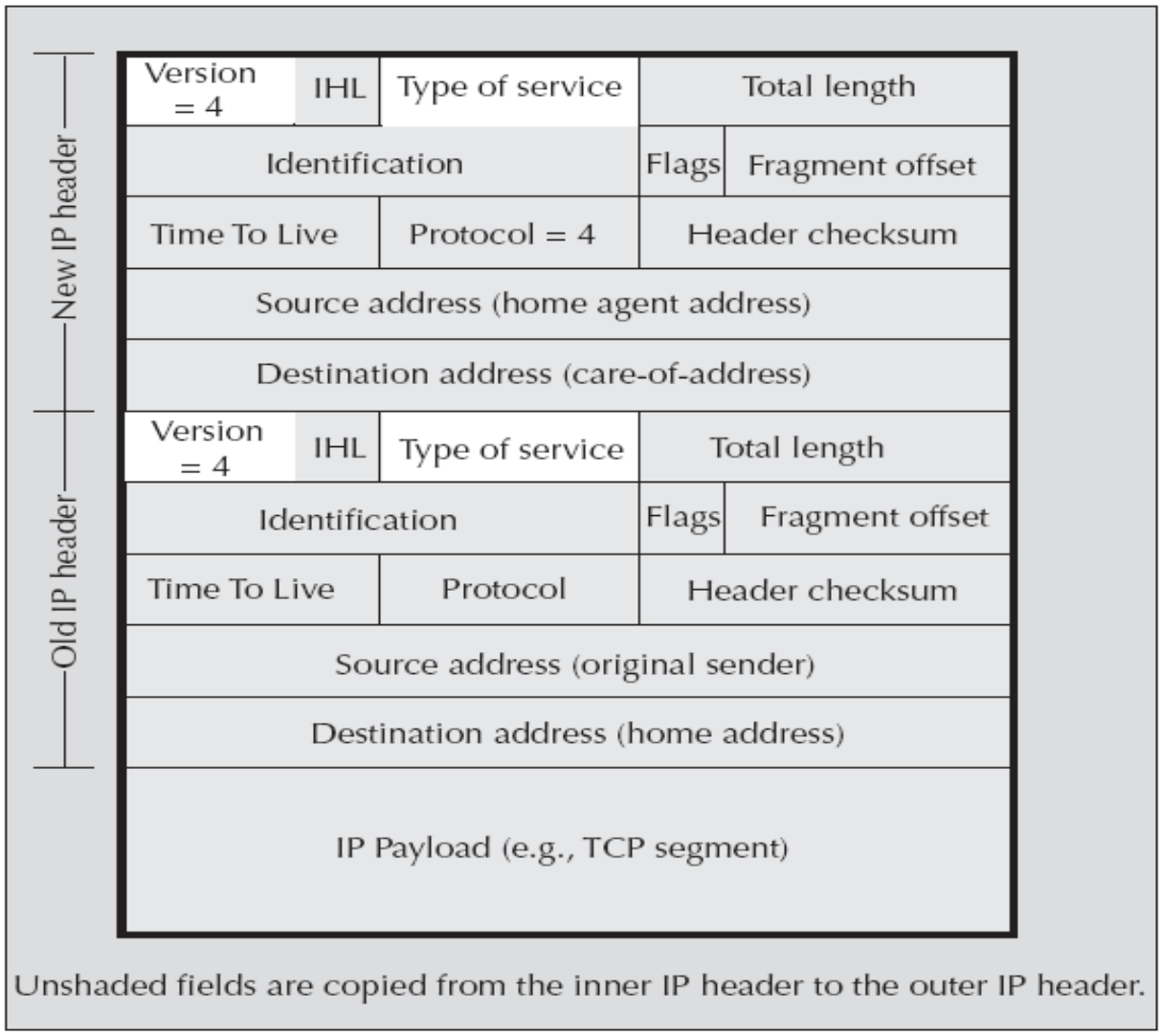


Let's take the case of mobile node (A) and another host (server X). The following steps take place:

- Server X wants to transmit an IP datagram to node A. The home address of A is advertised and known to X. X does not know whether A is in the home network or somewhere else. Therefore, X sends the packet to A with A's home address as the destination IP address in the IP header. The IP datagram is routed to A's home network.
- At the A's home network, the incoming IP datagram is intercepted by the home agent. The home agent discovers that A is in a foreign network. A care of address has been allocated to A by this foreign network and available with the home agent. The home agent encapsulates the entire datagram inside a new IP datagram, with A's care of address in the IP header. This new datagram with the care of address as the destination address is retransmitted by the home agent.
- At the foreign network, the incoming IP datagram is intercepted by the foreign agent. The foreign agent is the counterpart of the home agent in the foreign network. The foreign agent strips off the outer IP header, and delivers the original datagram to A.
- A intends to respond to this message and sends traffic to X. In this example, X is not mobile; therefore X has a fixed IP address. For routing A's IP datagram to X, each datagram is sent to some router in the foreign network. Typically, this router is the foreign agent. A uses X's IP static address as the destination address in the IP header.
- The IP datagram from A to X travels directly across the network, using X's IP address as the destination address.

- Discovery - A mobile node uses a discovery procedure to identify prospective home agents and foreign agents.
- Registration - A mobile node uses a registration procedure to inform its home agent of its care-of address.
- Tunneling - Tunneling procedure is used to forward IP datagrams from a home address to a care of address.

IP headers in Mobile IP



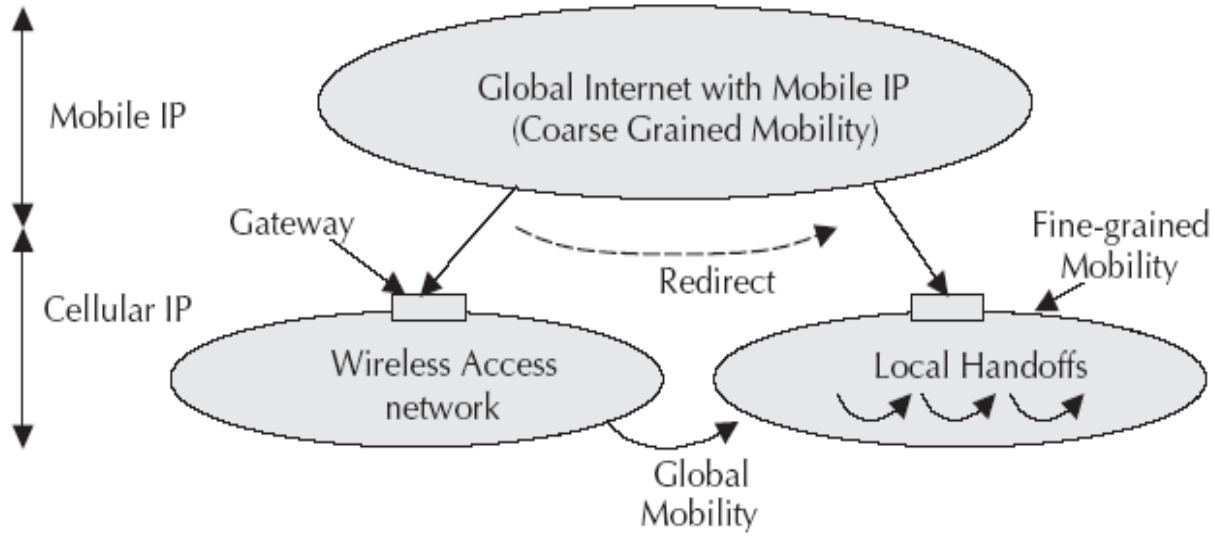
Cellular IP

None of the nodes know the exact location of a mobile host. Packets addressed to a mobile host are routed to its current base station on a hop-by-hop basis where each node only needs to know on which of its outgoing ports to forward packets. This limited routing information (referred as mapping) is local to the node and does not assume that nodes have any knowledge of the

wireless network topology. Mappings are created and updated based on the packets transmitted by mobile hosts.

- Uses two parallel structures of mappings through Paging Caches (PC) and Routing Caches (RC)
- PCs maintain mappings for stationary and idle (not in data communication state) hosts
- RC maintains mappings for mobile hosts
- Mapping entries in PC have a large timeout interval, in the order of seconds or minutes. RCs maintain mappings for mobile hosts currently receiving data or expecting to receive data

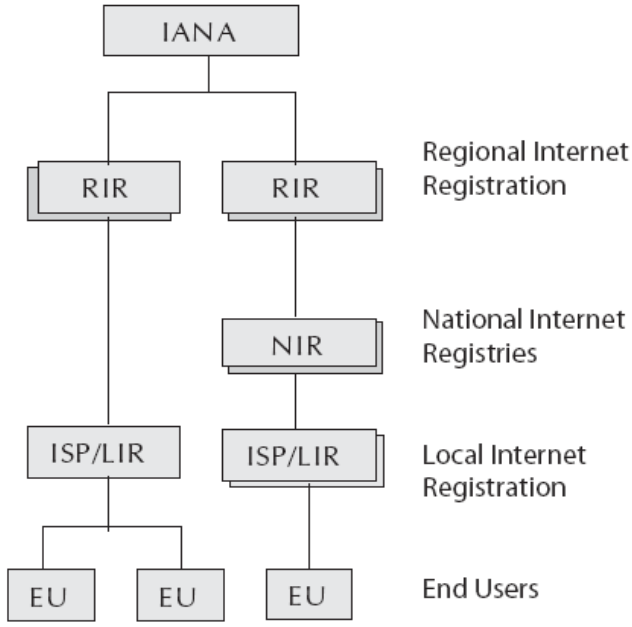
Relationship between Mobile IP and Cellular IP



Internet Protocol version

- Successor to today's IP version 4 protocol (IPv4)
- Internet Engineering Task Force (IETF) has produced a comprehensive set of specifications (RFC 1287, 1752, 1886, 1971, 1993, 2292, 2373, 2460, 2473, etc.) that define the next-generation IP protocol originally known as 'IPNg'
- Uses 128 bit addresses for each packet creating a virtually infinite number of IP addresses (approximately 3.4*10**38 IP addresses) as opposed to 3758096384 IPv4 addresses
- There are global addresses and local addresses
- Global addresses are used for routing of global Internet
- Link local addresses are available within a subnet
- IPv6 uses hierarchical addressing with three level of addresses
- Includes a Public Topology (the 48 bit external routing prefix)
- Site Topology (typically a 16 bit subnet number)
- Interface Identifier (typically an automatically generated 64 bit number unique on the local LAN segment)

Hierarchical addressing of IPv6



IPv6 Security

- Comes native with a security protocol called IP Security (IPSec)
- IPSec protocol is a standards-based method of providing privacy, integrity and authenticity to information transferred across IP networks.

Features of IPSec

- Diffie-Hellman key exchange mechanism for deriving key between peers on a public network
- Public key cryptography to guarantee the identity of the two parties and avoid man-in-the-middle attacks
- Bulk encryption algorithms, such as 3DES, for encrypting the data
- Keyed hash algorithms, such as HMAC, combined with traditional hash algorithms such as MD5 or SHA for providing packet authentication
- Digital certificates signed by a certificate authority to act as digital ID cards
- IPSec provides IP network layer encryption

Migrating from IPv4 to IPv6

- Migration of the network components to be able to support IPv6 packets. Using IP tunneling, IPv6 packets can propagate over an IPv4 envelope. Existing routers can support IP tunneling.
- Migration of the computing nodes in the network. This will need the operating system upgrades so that they support IPv6 along with IPv4. Upgraded systems will have both IPv4 and IPv6 stacks.

Migration of networking applications in both client and server systems. This requires porting of the applications from IPv4 to IPv6 environment.

Interconnecting IPv6 networks

- Tunneling is one of the key deployment strategies for both service providers as well as enterprises during the period of IPv4 and IPv6 coexistence.
- Tunneling service providers can offer an end-to-end IPv6 service without major upgrades to the infrastructure and without impacting current IPv4 services.

Tunneling Mechanisms

- Manually created tunnels such as IPv6 manually configured tunnels (RFC 2893)
- IPv6 over IPv4 tunnels
- Semiautomatic tunnel mechanisms such as that employed by tunnel broker services
- Fully automatic tunnel mechanisms such as IPv4 compatible

Mobile IP with IPv6

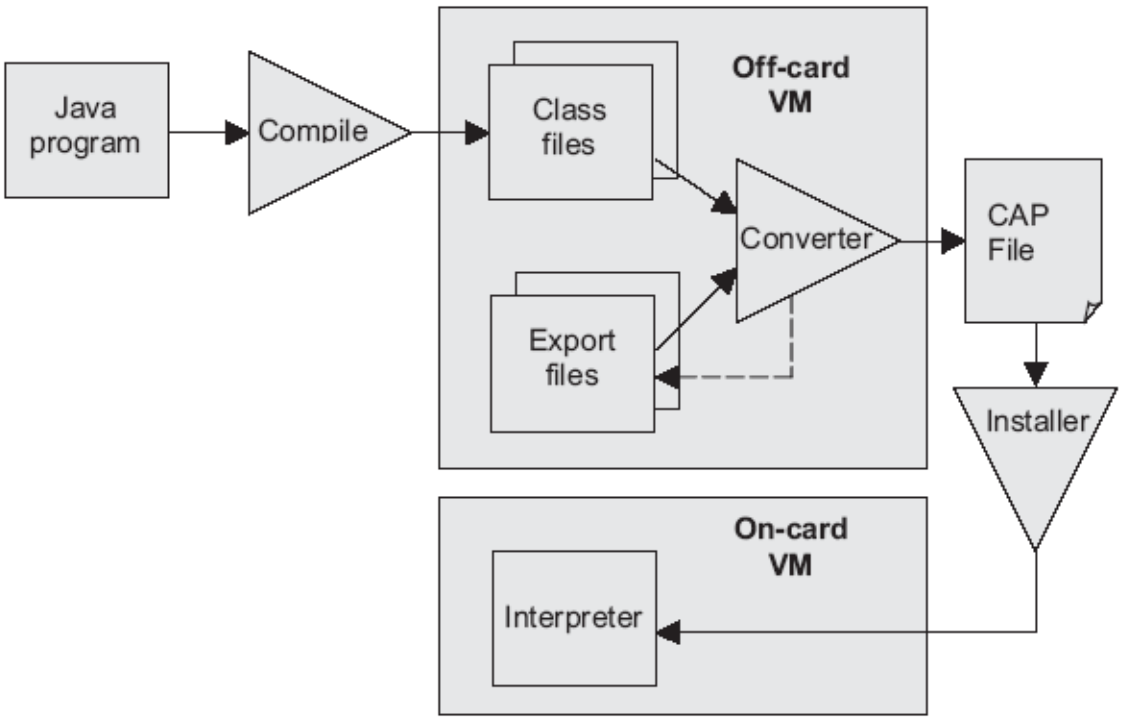
- IPv6 with hierarchical addressing scheme can manage IP mobility much efficiently.
- IPv6 also attempts to simplify the process of renumbering which could be critical to the future routing of the Internet traffic.
- Mobility Support in IPv6, as proposed by the Mobile IP working group, follows the design for Mobile IPv4. It retains the ideas of a home network, home agent and the use of encapsulation to deliver packets from the home network to the mobile node's current point of attachment.
- While discovery of a care of address is still required, a mobile node can configure its a care of address by using Stateless Address Auto-configuration and Neighbor Discovery. Thus, foreign agents are not required to support mobility in IPv6.

Java Card

- Smart card with Java framework
- Smart card is a plastic card with intelligence and memory
- A smart card is embedded with either (i) a microprocessor and a memory chip or (ii) only a memory chip with non-programmable logic
- A microprocessor card can have an intelligent program resident within the card which can add, delete, and otherwise manipulate information on the card
- A memory card can store some information for some pre-defined operation
- Java was chosen as the vehicle for interoperability
- All the microprocessor based smart cards now offer Java API framework on the smart card

- Java Card technology preserves many of the benefits of the Java programming languages such as productivity, security, robustness, tools, and portability
- For Java card, the Java Virtual Machine (JVM), the language definition, and the core packages have been made more compact to bring Java technology to the resource constrained smart cards

Architecture of Java Card



Functioning of Java Card

- Java card technology supports OTA (Over The Air) downloads
- Applications written for the Java Card platform are referred to as applets
- Challenge of Java Card technology on smart card is to fit Java system software in a resource constraint smart card while conserving enough space for applications

- The Java Card virtual machine on a smart card is split into two parts; one that runs off-card and the other that runs on-card.
- Many processing tasks that are not constrained to execute at run time, such as class loading, bytecode verification, resolution and linking, and optimization, are dedicated to the virtual machine that is running off-card where resources are usually not a concern
- The on-card components of Java Card include components like the Java card virtual machine (JCVM), the Java card runtime environment (JCRE), and the Java API
- Task of the compiler is to convert a Java source into Java class files
- The converter will convert class files into a format downloadable into the smart card while ensuring the byte code validity.

The converter checks the classes off-card for

- Well formedness
- Java Card subset violations
- Static variable initialization
- Reference resolution
- Byte code optimization
- Storage Allocation.
- The java card interpreter
- Executes the applets
- Controls run-time resources
- Enforces runtime security
- Following conversion by the off-card VM into CAP (Converted APlet) format, the applet is transferred into the card using the installer
- Applet is selected for execution by the JCRE
- JCRE is made up of the on-card virtual machine and the Java Card API classes and performs additional runtime security checks through applet firewall
- Applet firewall partitions the objects stored into separate protected object spaces, called contexts and controls the access to shareable interfaces of these objects
- It is then executed on JVCM which is scaled down version of standard JVM (Java Virtual Machine)

Global System for Mobile Communications:

- Originally GSM stood for Groupe Speciale Mobile GSM to meet the following business objectives:

1. Support for international roaming
2. Good speech quality
3. Ability to support handheld terminals
4. Low terminal and service cost
5. Spectral efficiency

6. Support for a range of new services and facilities
7. ISDN compatibility

GSM Timeline

1982	Groupe Spécial Mobile (GSM) established
1987	Essential elements of wireless transmission specified
1989	GSM become an ETSI technical committee
1990	Phase 1 GSM 900 specification (designed 1987 through 1990) frozen
1991	First GSM network launched
1993	First roaming agreement came into effect
1994	Data transmission capability launched
1995	Phase 2 launched. Fax and SMS roaming services offered
2002	SMS volume crosses 24 billions/year, 750 millions subscribers

Use of TDMA and FDMA in GSM

- Uses a combination of FDMA (Frequency Division Multiple Access) and TDMA (Time Division Multiple Access)
- Allocation of 50 MHz (890–915 MHz and 935–960 MHz) bandwidth in the 900 MHz frequency band and using FDMA further divided into 124 (125 channels, 1 not used) channels each with a carrier bandwidth of 200 KHz
- Using TDMA, each of the above mentioned channels is then further divided into 8 time slots
- So, with the combination of FDMA and TDMA, a maximum of992 channels for transmit and receive can be realized

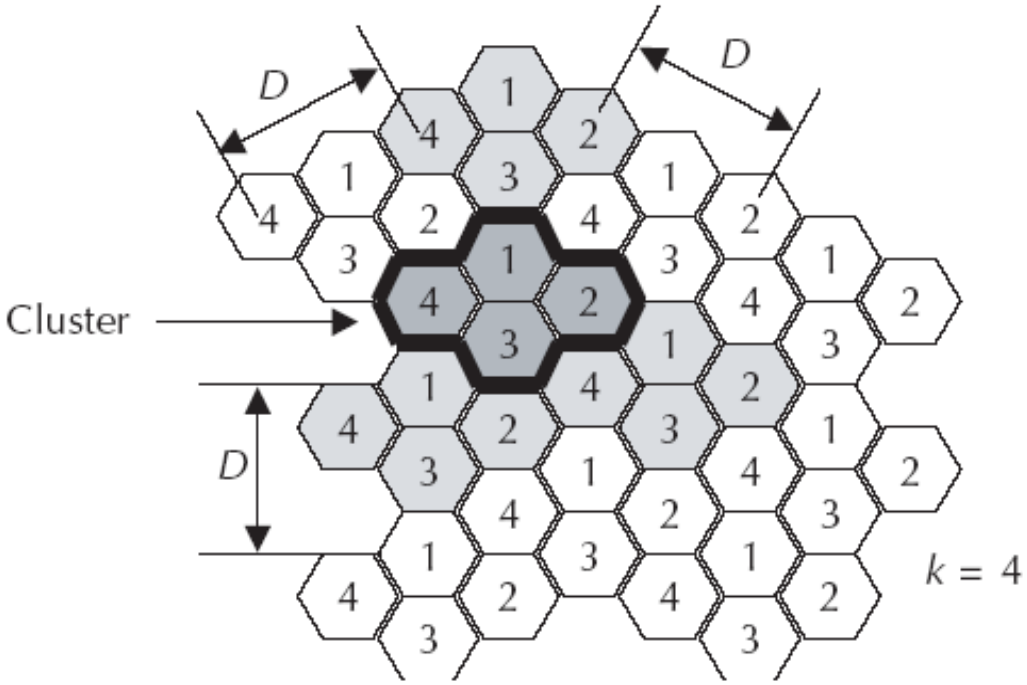
Frequency reuse in GSM

- To serve hundreds of thousands of users, the frequency must be reused and this is done through cells.
- The area to be covered is subdivided into radio zones or cells. Though in reality these cells could be of any shape, for convenient modeling purposes these are modeled as hexagons. Base stations are positioned at the center of these cells.
- Each cell *i* receives a subset of frequencies *fbi* from the total set assigned to the respective mobile network. To avoid any type of co-channel interference, two neighboring cells never use the same frequencies.

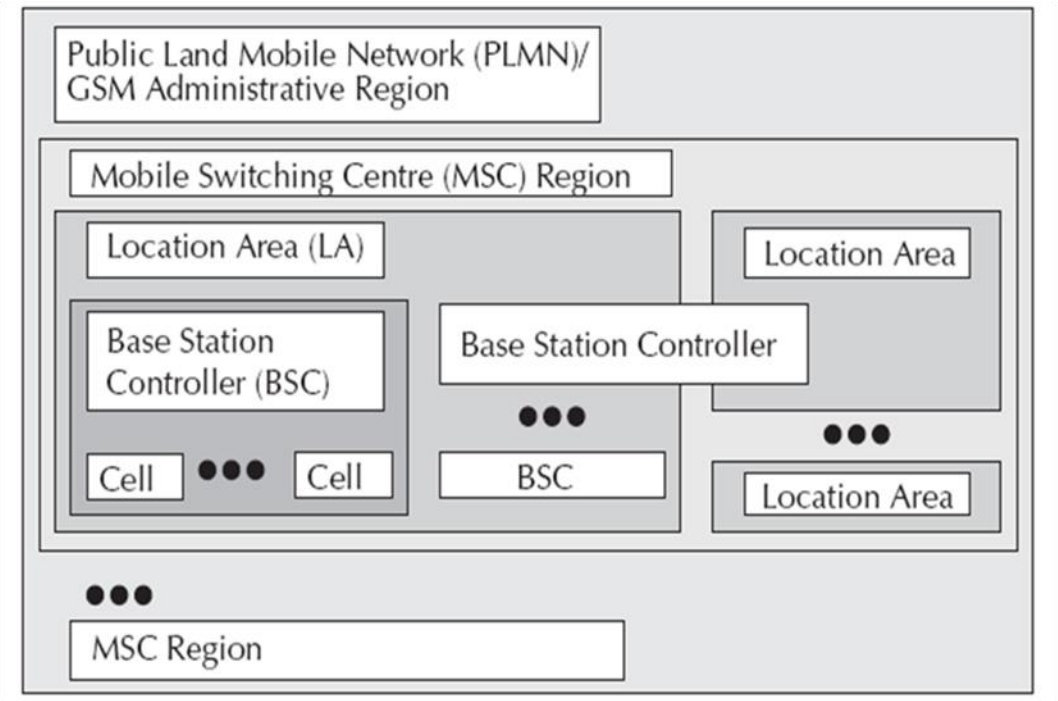
- Only at a distance of *D* (known as frequency reuse distance), the same frequency from the set *fbi* can be reused. Cells with distance *D* from cell *i*, can be assigned one or all the frequencies from the set *fbi* belonging to cell *i*.

- When moving from one cell to another during an ongoing conversation, an automatic channel change occurs. This phenomenon is called handover. Handover maintains an active speech and data connection over cell boundaries.
- The regular repetition of frequencies in cells result in a clustering of cells. The clusters generated in this way can consume the whole frequency band. The size of a cluster is defined by *k*, the number of cells in the cluster. This also defines the frequency reuse distance *D*. The figure in next slide shows an example of cluster size of 4.

Cell clusters in GSM:

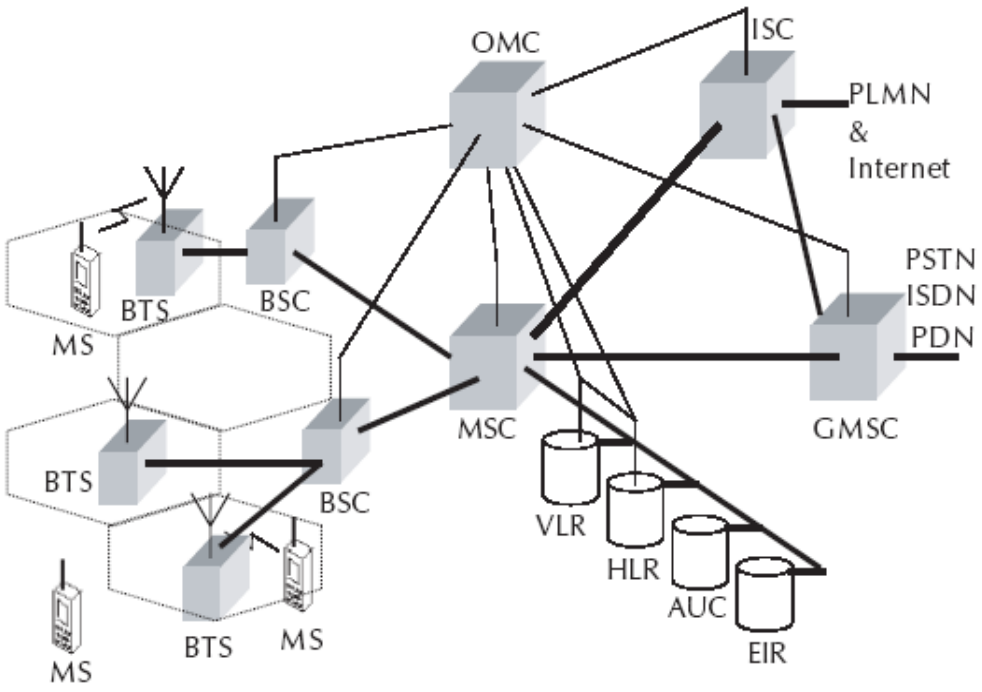


GSM System Hierarchy



- Consists at the minimum one administrative region assigned to one MSC (Mobile Switching Centre)
- Administrative region is commonly known as PLMN (Public Land Mobile Network)
- Each administrative region is subdivided into one or many Location Area (LA)
- One LA consists of many cell groups and each cell group is assigned to one BSC (Base Station Controller)
- For each LA, there will be at least one BSC while cells in one BSC can belong to different LAs

GSM Architecture

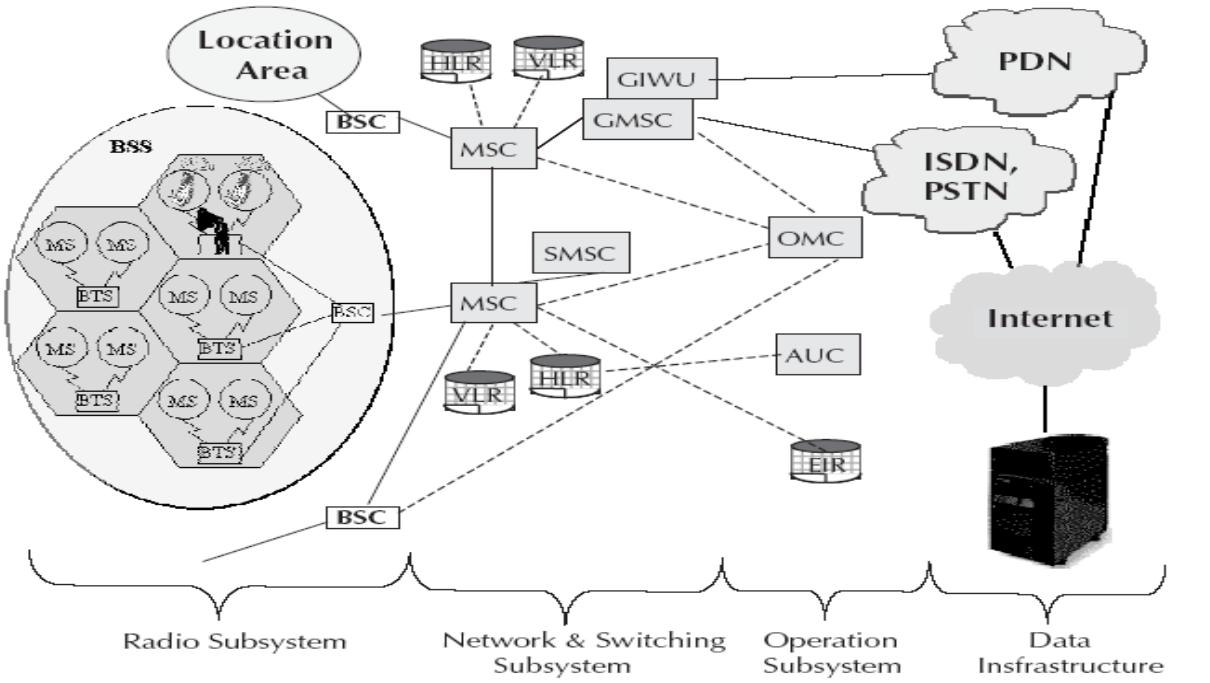


BTS	Base Transceiver Station
BSC	Base Station Controller
MSC	Mobile Switching Center
GMSC	Gateway MSC
ISC	International Switching Center

MS	Mobile Station
HLR	Home Location Register
VLR	Visitor Location Register
AUC	Authentication Center
EIR	Equipment Identity Register
OMC	Operation and Maintenance Center

- Cells are formed by the radio areas covered by a BTS (Base Transceiver Station)
- Several BTSs are controlled by one BSC
- Traffic from the MS (Mobile Station) is routed through MSC
- Calls originating from or terminating in a fixed network or other mobile networks is handled by the GMSC (Gateway MSC)

Operational Architecture of GSM:



BTS	Base Transceiver Station	ISDN	Integrated System Digital Network
BSC	Base Station Controller	HLR	Home Location Register
MSC	Mobile Switching Center	VLR	Visitor Location Register
GMSC	Gateway MSC	AUC	Authentication Center
SC	Service Center/SMS Controller	EIR	Equipment Identity Register
GIWU	Gateway Interworking Unit	OMC	Operation and Maintenance Center
PDN	Public Data Network	PSTN	Public Switched Telephone Network
MS	Mobile Station		

Home Location Register (HLR) in GSM

- It contains the following information:

1. Authentication information like International Mobile Subscriber Identity (IMSI)
2. Identification information like name, address, etc. of the subscriber
3. Identification information like Mobile Subscriber ISDN (MSISDN) etc.
4. Billing information like prepaid or postpaid
5. Operator selected denial of service to a subscriber
6. Handling of supplementary services like for CFU (Call Forwarding Unconditional), CFB (Call Forwarding Busy), CFNR (Call Forwarding Not Reachable) or CFNA (Call Forwarding Not Answered)
7. Storage of SMS Service Center (SC) number in case the mobile is not connectable so that whenever the mobile is connectable, a paging signal is sent to the SC
8. Provisioning information like whether long distance and international calls allowed or not
9. Provisioning information like whether roaming is enabled or not
10. Information related to auxiliary services like Voice mail, data, fax services, etc.
11. Information related to auxiliary services like CLI (Caller Line Identification), etc.
12. Information related to supplementary services for call routing. In GSM network, one can customize the personal profile to the extent that while the subscriber is roaming in a

foreign PLMN, incoming calls can be barred. Also, outgoing international calls can be barred, etc.

13. Some variable information like pointer to the VLR, location area of the subscriber, Power OFF status of the handset, etc.

Entities in GSM

- The Mobile Station (MS) - This includes the Mobile Equipment (ME) and the Subscriber Identity Module (SIM).
- The Base Station Subsystem (BSS) - This includes the Base Transceiver Station (BTS) and the Base Station Controller (BSC).
- The Network and Switching Subsystem (NSS) - This includes Mobile Switching Center (MSC), Home Location Register (HLR), Visitor Location Register (VLR), Equipment Identity Register (EIR), and the Authentication Center (AUC).
- The Operation and Support Subsystem (OSS) - This includes the Operation and Maintenance Center (OMC).

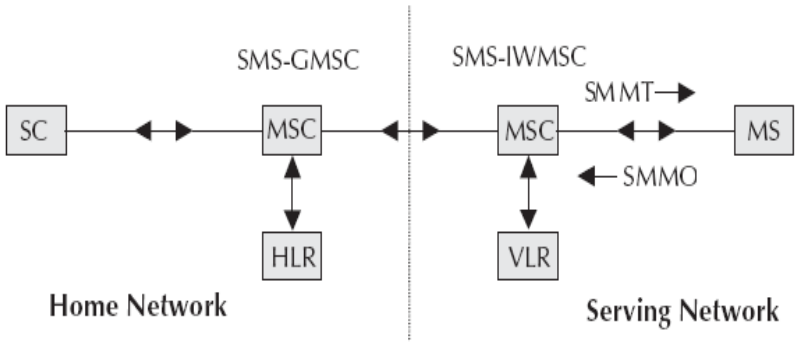
Short Message Service:

- Short Message Service (SMS) is one of the most popular services within GSM
- SMS is a data service and allows a user to enter text message up to 160 characters in length when 7 bit English characters are used
- SMS is a proactive bearer and is an 'always on' network
- Message center is referred to as Service Centre (SC) or SMS Controller (SMSC)
- SMSC is a system within the core GSM network which works as the store and forward system for SMS messages.

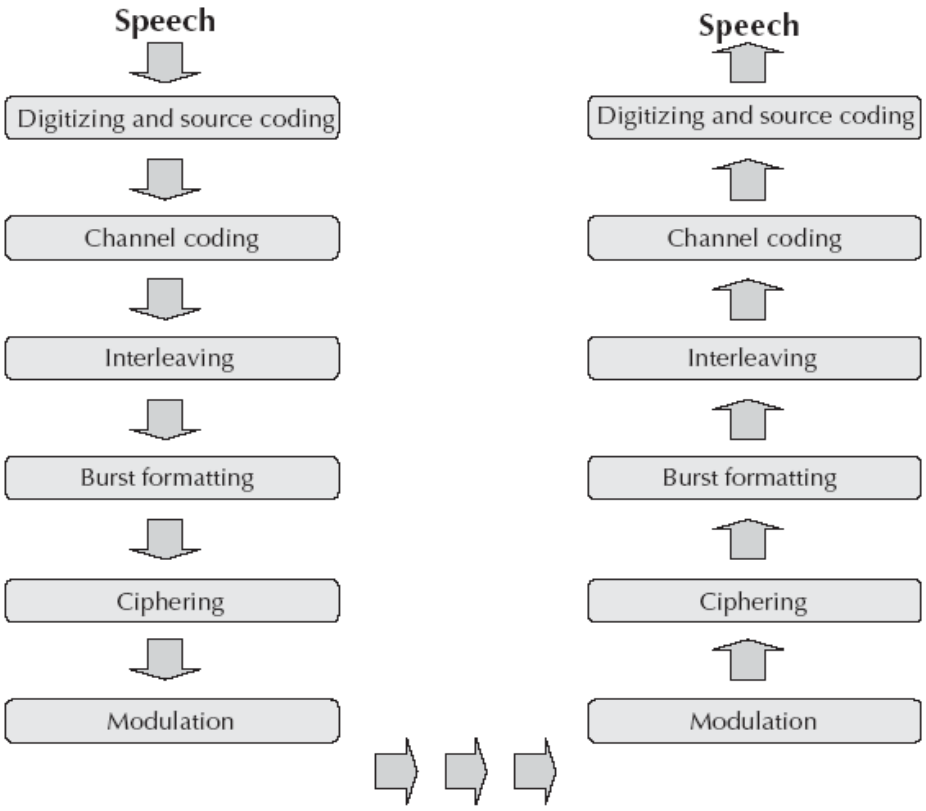
SMS

- Two types of SMS: SMMT (Short Message Mobile Terminated Point-to-Point) and SMMO (Short Message Mobile Originated Point-to-Point)
- SMMT is an incoming short message from the network and is terminated in the MS (phone or Mobile Station)
- SMMO is an outgoing message originated in the MS, and forwarded to the network for delivery
- For an outgoing message, the SMS is sent from the phone to SC via the VLR and the Inter Working MSC (IWMSC)
- For incoming message, the path is from SC to the MS via the HLR and the Gateway MSC (GMSC)

SMS Transfer :



From speech to radio waves



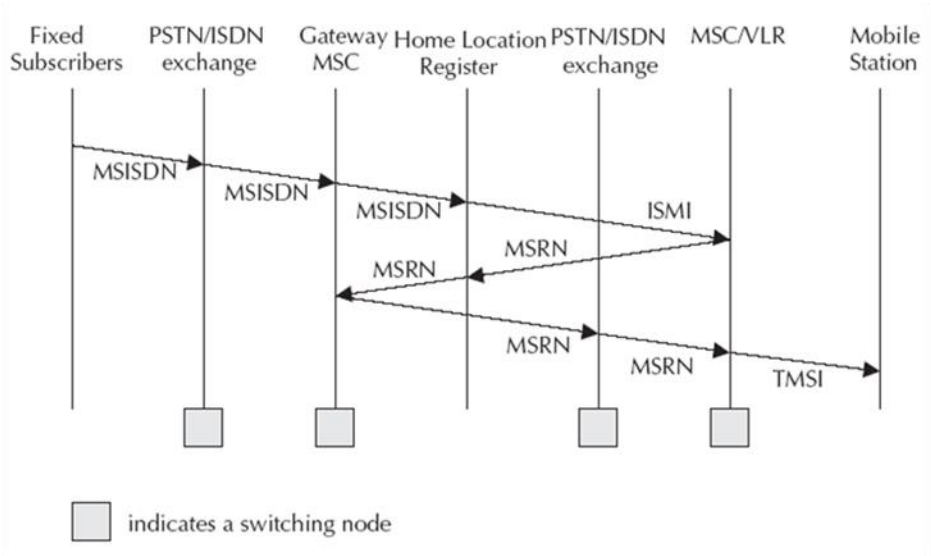
- Digitizer and source coding: The user speech is digitized at 8 KHz sampling rate using Regular Pulse Excited–Linear Predictive Coder (RPE–LPC) with a Long Term Predictor loop where information from previous samples is used to predict the current sample. Each sample is then represented in signed 13-bit linear PCM value. This digitized data is passed to the coder with frames of 160 samples where encoder compresses these 160 samples into 260-bits GSM frames resulting in one second of speech compressed into 1625 bytes and achieving a rate of 13 Kbits/sec.
- Channel coding: This introduces redundancy into the data for error detection and possible error correction where the gross bit rate after channel coding is 22.8 kbps (or 456 bits every 20 ms). These 456 bits are divided into eight 57-bit blocks and the result is interleaved amongst eight successive time slot bursts for protection against burst transmission errors.
- Interleaving: This step rearranges a group of bits in a particular way to improve the performance of the error-correction mechanisms. The interleaving decreases the possibility of losing whole bursts during the transmission by dispersing the errors.
- Ciphering: This encrypts blocks of user data using a symmetric key shared by the mobile station and the BTS.

- Burst formatting: It adds some binary information to the ciphered block for use in synchronization and equalization of the received data.
- Modulation: The modulation technique chosen for the GSM system is the Gaussian Minimum Shift Keying (GMSK) where binary data is converted back into analog signal to fit the frequency and time requirements for the multiple access rules. This signal is then radiated as radio wave over the air.
- Multipath and equalization: An equaliser is in charge of extracting the 'right' signal from the received signal while estimating the channel impulse response of the GSM system and then it constructs an inverse filter. The received signal is then passed through the inverse filter.
- Synchronization: For successful operation of a mobile radio system, time and frequency synchronization are needed. Frequency synchronization is necessary so that the transmitter and receiver frequency match (in FDMA) while Time synchronization is necessary to identify the frame boundary and the bits within the frame (in TDMA).
- To avoid collisions of burst transmitted by MS with the adjacent timeslot such collisions, the Timing Advance technique is used where frame is advanced in time so that this offsets the delay due to greater distance. Using this technique and the triangulation of the intersection cell sites, the location of a mobile station can be determined from within the network.

Call Routing:

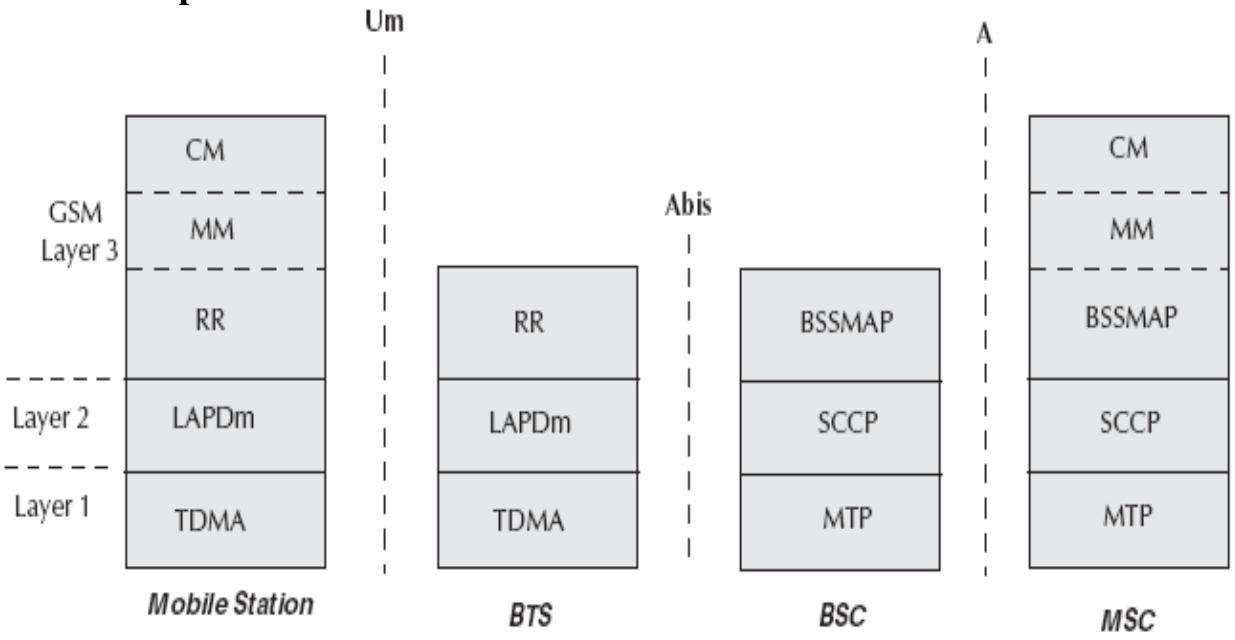
- The directory number dialed to reach a mobile subscriber is called the Mobile Subscriber ISDN (MSISDN) which is defined by the E.164 numbering plan and includes a country code and a National Destination Code, which identifies the subscriber's operator. The first few digits of the remaining subscriber number may identify the subscriber's HLR within the home PLMN
- For example, the MSISDN number of a subscriber in Bangalore associated with Airtel network is +919845XYYYYY which is a unique number and understood from anywhere in the world. Here, + means prefix for international dialing, 91 is the country code for India and 45 is the network operator's code (Airtel in this case). X is the level number managed by the network operator ranging from 0 to 9 while YYYYY is the subscriber code which , too, is managed by the operator.

Call Routing



- The call first goes to the local PSTN exchange where PSTN exchange looks at the routing table and determines that it is a call to a mobile network.
- PSTN forwards the call to the Gateway MSC (GMSC) of the mobile network.
- MSC enquires the HLR to determine the status of the subscriber. It will decide whether the call is to be routed or not. If MSC finds that the call can be processed, it will find out the address of the VLR where the mobile is expected to be present.
- If VLR is that of a different PLMN, it will forward the call to the foreign PLMN through the Gateway MSC. If the VLR is in the home network, it will determine the Location Area (LA).
- Within the LA, it will page and locate the phone and connect the call.

Network aspects in GSM



- Layer 3 of the GSM signaling protocol is itself divided into three sub-layers:

1. Radio Resources Management: It controls the set-up, maintenance and termination of radio and fixed channels, including handovers.
2. Mobility Management: It manages the location updating and registration procedures as well as security and authentication.
3. Connection Management: It handles general call control and manages Supplementary Services and the Short Message Service.

Handover

- The procedure of change of resources is called handover when the user is mobile while the call is in progress.
- There are four different types of handover in the GSM system, which involve transferring a call between:

1. Channels (time slots) in the same cell
2. Cells (Base Transceiver Stations) under the control of the same Base Station Controller (BSC)
3. Cells under the control of different BSCs but belonging to the same Mobile Switching Center (MSC)
4. Cells under the control of different MSCs.

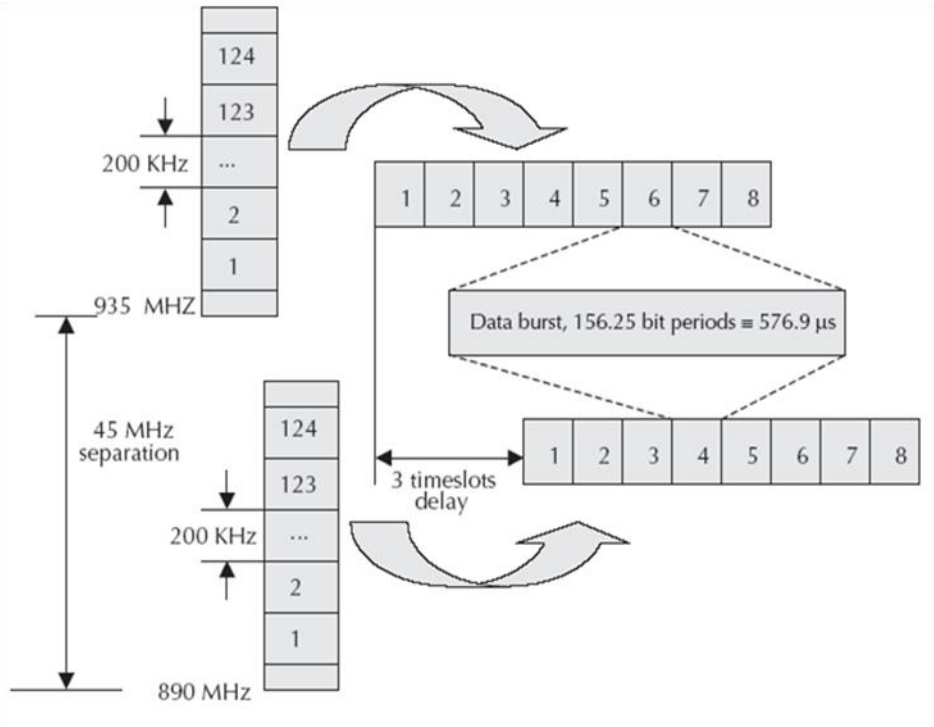
- First two types of handover, called internal handovers, involve only one Base Station Controller (BSC). To save signaling bandwidth, they are managed by the BSC without involving the Mobile services Switching Center (MSC), except to notify it at the completion of the handover.
- Last two types of handover, called external handovers, are handled by the MSC.

GSM Frequency Allocation

- Normally, GSM uses 900 MHz band wherein 890-915 MHz is allocated for the uplink (mobile station to base station) and 935–960 MHz is allocated for the downlink (base station to mobile station). Each way the bandwidth for the GSM system is 25 MHz which provides 125 carriers uplink/downlink each having a bandwidth of 200 kHz.
- ARFCN (Absolute Radio Frequency Channel Numbers) denote a forward and reverse channel pair which is separated in frequency by 45 MHz.
- Practically, a guard band of 100 kHz is provided at the upper and lower end of the GSM 900 MHz spectrum and only 124 (duplex) channels are implemented.

- GSM uses a combination of Time Division Multiple Access (TDMA) and Frequency Division Multiple Access (FDMA) encoding.
- One or more carrier frequencies are assigned to each base station and each of these carrier frequencies is then divided in time using a TDMA scheme where fundamental unit is called a burst period lasting approximately 0.577 ms.
- Eight burst periods are grouped into a TDMA frame of approximately 4.615 ms which forms the basic unit for the definition of logical channels.
- One physical channel is one burst period per TDMA frame while, normally, channels are defined by the number and position of their corresponding burst periods.

GSM Frequency Allocation



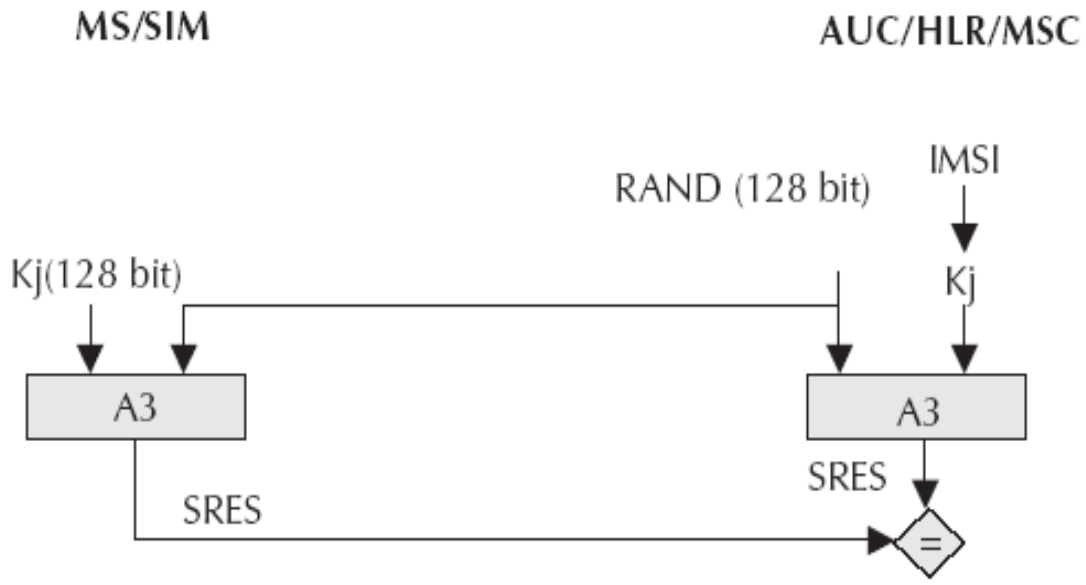
Authentication and Security

- Authentication involves two functional entities - the SIM card in the mobile phone and the Authentication Center (AUC).
- Following authentication by algorithm A3, a key is generated for encryption.
- An algorithm A8 is used to generate the key while a different algorithm called A5 is used for both ciphering and deciphering procedures for signaling, voice and data.
- So, SS7 signal, voice, data, and SMS within GSM networks are ciphered over the wireless radio interface.

A3 Algorithm

- During authentication, MSC challenges the MS with a random number (RAND).
- SIM card uses this RAND received from the MSC and a secret key Kj stored within the SIM as input. Both the RAND and the Kj secret are 128 bits long. Using the A3 algorithm with RAND and Kj as input a 32-bit output called signature response (SRES) is generated in the MS and then sent back to MSC.
- Using the same set of algorithms, the AUC also generates a SRES. The SRES from MS and the SRES generated by the AUC are compared.
- If they are the same, the MS is authenticated.

- The idea is that no keys will be transacted over the air. However, if the SRES values calculated independently by the SIM and the AUC are the same, then Kj has to be same and if Kj is same, SIM card is genuine.



Short Message Service (SMS)

- Most popular data bearer/service within GSM
- More than one billion SMS messages interchanged everyday with a growth of more than half a billion every month on an average
- Runs on SS7 signaling channels, which are always present but mostly unused, be it during an active user connection or in the idle state
- Each short message is up to 160 characters in length when 7-bit English characters are used and 140 octets when 8-bit characters are used

Strengths of SMS

- Omnibus nature of SMS: SMS uses SS7 signaling channel which is available throughout the world.
- Stateless: SMS is session-less and stateless as every SMS message is unidirectional and independent of any context. This makes SMS the best bearer for notifications, alerts and paging.
- Asynchronous: SMS is completely asynchronous. In case of SMS, even if the recipient is out of service, the transmission will not be abandoned and hence, SMS can be used as message queues. SMS can be used as a transport bearer for both synchronous (transaction oriented) and asynchronous (message queue and notification) information exchange.

- Self-configurable and last mile problem resistant: SMS is self-configurable and subscriber is always connected to the SMS bearer irrespective of the home and visiting network configurations.
- Non-repudiable: SMS message carries the Service Center (SC) and the source MSISDN as a part of the message header through which any SMS can prove beyond doubt its origin.
- Always connected: As SMS uses the SS7 signaling channel for its data traffic, the bearer media is always on. Users cannot switch OFF, BAR or DIVERT any SMS message. SMS message is delivered to the Mobile Station (MS) without any interruption to the ongoing call.

SMS Architecture

- Two types of SMS - **SM MT** (Short Message Mobile Terminated Point-to-Point) and **SM MO** (Short Message Mobile Originated Point-to-Point)
- SM MT is an incoming short message from the network and is terminated in the MS
- SM MO is an outgoing message originated in the MS and forwarded to the network for delivery
- For an outgoing message, the path is from MS to SC via the VLR and the IWMSC (Inter Working MSC) function of the serving MSC whereas for an incoming message the path is from SC to the MS via HLR and the GMSC (Gateway MSC) function of the home MSC

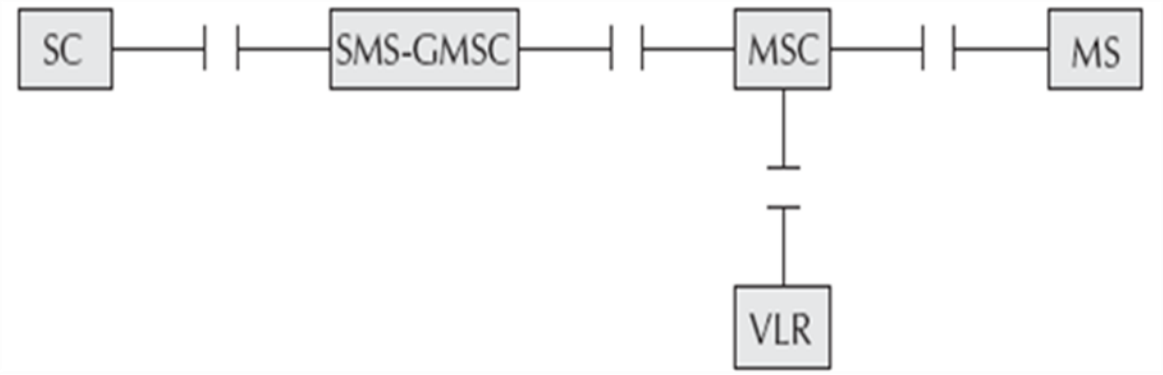
Strengths of SMS

- Omnibus nature of SMS: SMS uses SS7 signaling channel which is available throughout the world.
- Stateless: SMS is session-less and stateless as every SMS message is unidirectional and independent of any context. This makes SMS the best bearer for notifications, alerts and paging.

- Asynchronous: SMS is completely asynchronous. In case of SMS, even if the recipient is out of service, the transmission will not be abandoned and hence, SMS can be used as message queues. SMS can be used as a transport bearer for both synchronous (transaction oriented) and asynchronous (message queue and notification) information exchange.

Short Message Mobile Terminated (SMMT)

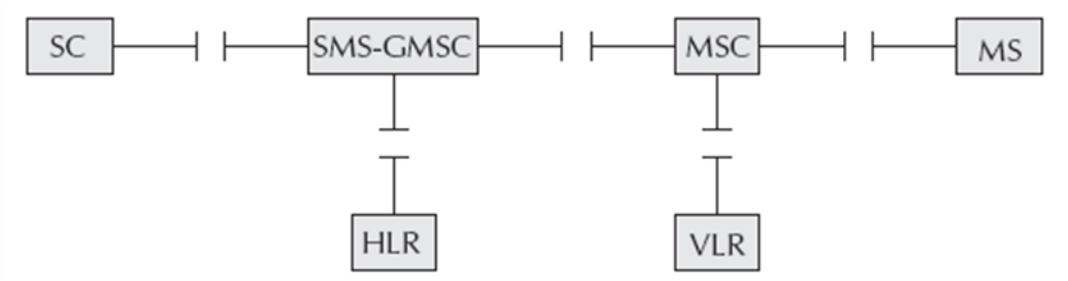
- Message is sent from SC to the MS.
- For the delivery of MT or incoming SMS messages, the SC of the serving network is never used which implies that a SMS message can be sent from any SC in any network to a GSM phone anywhere in the world.



Interfaces in SMMT

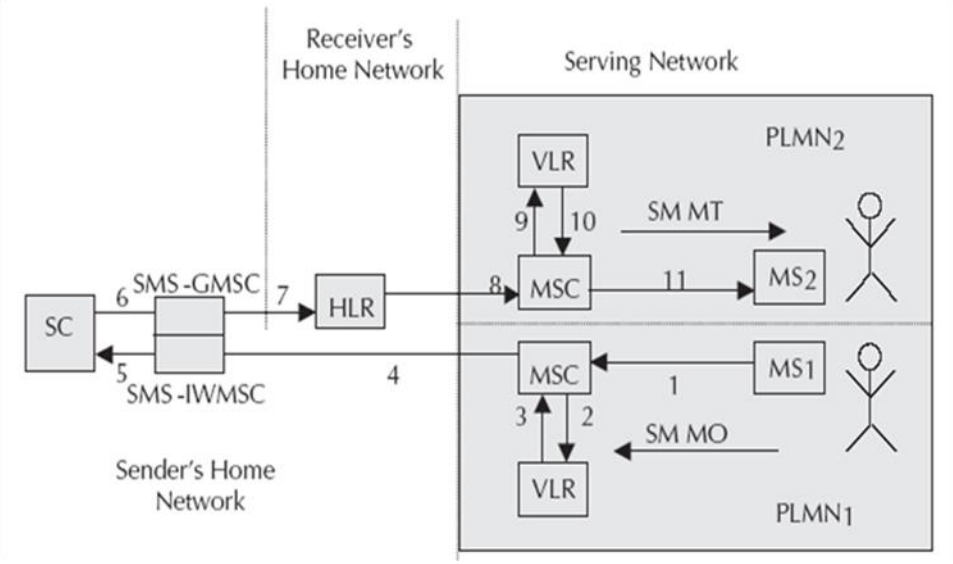
Short Message Mobile Originated

- For a MO message, the MSC forwards the message to the home SC.
- MO message works in two asynchronous phases. In the first phase, the message is sent from the MS to the home SC as a MO message. In the second phase, the message is sent from the home SC to the MS as a MT message.



Interfaces in SMMO

SMS Transfer



SMS as an Information Bearer

- For using SMS as an information bearer, we need to connect the services running on the Enterprise Origin server to the SC through an SME (Short Message Entity) or ESME (External Short Message Entity).
- SME in any network is generally a SMS gateway.
- With respect to SMS, a GSM subscriber is always in control of the SC in the home network irrespective of the serving network.
- If there is any SMS-based data service in the home network, it will be available in any foreign network.

Value Added Services through SMS

- Value Added Services (VAS) can be defined as services, which share one or more of the following characteristics:

1. Supplementary service (not a part of basic service) but adds value to total service offering
2. Stimulates incremental demand for core services offering
3. Stands alone in terms of profitability and revenue generation potential
4. Can sometimes stand-alone operationally
5. Does not cannibalize basic service unless clearly favorable
6. Can be an add-on to basic service, and as such, may be sold at a premium price
7. May provide operational and/or administrative synergy between or among other services and not merely for diversification

VAS over SMS are entertainment and information on demand which is further categorized into:

1. Static information which does not change frequently
2. Dynamic information which changes in days
3. Real-time information which changes continually

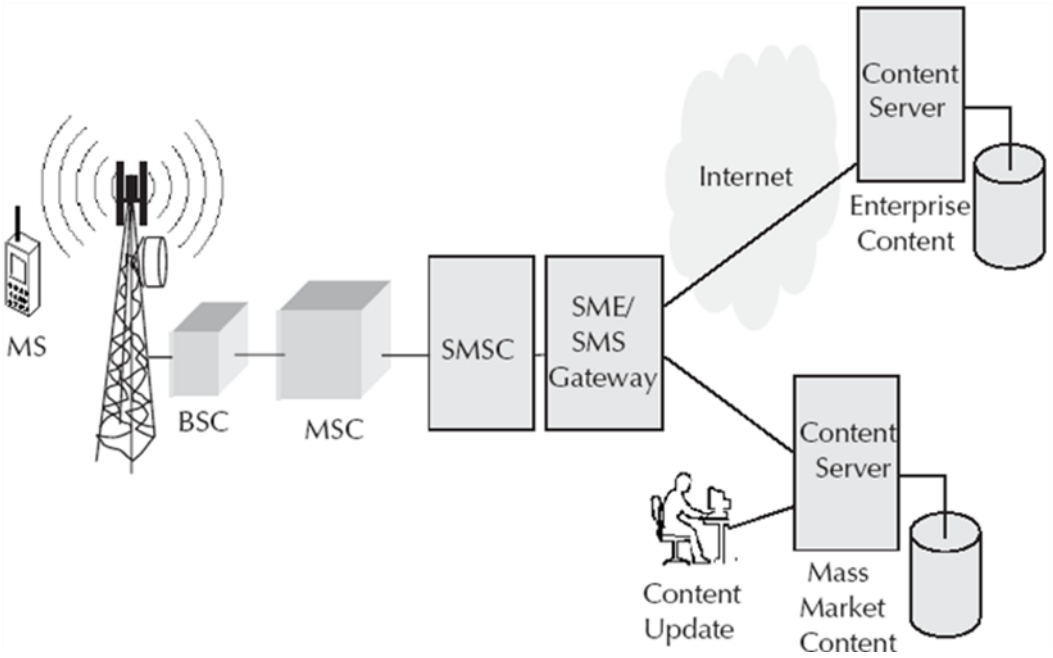
- Some of the common VAS examples are:

1. News/Stock Quotes Service
2. Session-based Chat Application
3. Email through SMS
4. Health Care Services
5. Micro-Payment Services

Alert services through VAS

- Proactive alert services can be of the two kinds – Time based and Watermark based
- Time based proactive alerts are sent to the mobile phone at a pre-assigned time of the day
- Watermark based proactive alerts are sent when some event occurs

VAS Architecture



UNIT II - POSSIBLE QUESTIONS

PART – B
(Each Question carries six marks)

1. Write about Bluetooth protocol stack with diagram.
2. Explain GSM architecture with suitable diagram.
3. Explain RFID transponders with suitable example.
4. Write a detailed notes on the value added service through SMS.
5. Write about the WIMAX architecture with suitable example.
6. Explain the Java card application with suitable diagram.
7. Explain the Mobile IP architecture with suitable example.
8. Explain the architecture of Java card with suitable diagram.
9. Explain the GSM System hierarchy with suitable diagram.
10. Explain the Short Messaging Service and architecture with suitable diagram.

PART – C
Compulsory Question
(Each question carries ten marks)

1. What should be the characteristics of Mobile Computing Devices?
2. Describe the security aspects in mobile computing.
3. Explain the functionality of Mobile Computing
4. How would you broadly classify the mobile computing applications?
5. Who are all the stakeholders of wireless network?
6. What is WiMax? How it is differ from WiFi?
7. Describe 3G networks.
8. Explain the Process of GSM Call Routing.
9. Explain the Process of Interactive Voice Response.
10. Briefly describe the Eclipse IDE for the development of android application.

MOBILE COMPUTING (18CAP402) UNIT- III

S.No	Question	Option1	Option2	Option3	Option4	Answer
1	_____is a step to efficiently transport high-speed data over the current GSM a	CDMA	GPRS	2G	3G	GPRS
2	GPRS has the ability to offer data speeds of_____to_____for comfor	14 Kbps,150 Kbps	14 .1Kbps,165 Kbps	14 .4 Kbps,171.2 K	13 .4 Kbps,161.2 Kb	14 .4 Kbps,171.2 Kbps
3	_____is a less costly mobile data service compares to SMS and Circuit Switche	2G	GSM	GPS	GPRS	GPRS
4	_____is the priority of a service in relation to another service	Service precedence	Reliability	Delay	Throughput	Service precedence
5	_____indicates the transmission characteristics required by an application	Delay	Throughput	Reliability	Service precedence	Reliability
6	_____parameters define maximum values for the mean delay	Reliability	Delay	Throughput	Service precedence	Delay
7	_____specifies the maximum bit rate and the mean bit rate	Service precedence	Reliability	Delay	Throughput	Throughput
8	GPRS is called the_____in the evolution process of wireless cellular networks	2G	2.5G	3G	4G	2.5G
9	_____processes registration of new mobile subscribers and keeps a record of	SGSN	GGSN	PDP	HLR	SGSN
10	SGSN sends queries to_____to obtain profile data of GPRS subscribers	GGSN	HLR	SGSN	PDP	HLR
11	The_____is connected to the base station system with frame relay	HLR	PDP	GGSN	SGSN	SGSN
12	_____acts as an interface between the GPRS backbone network and theexter	PDP	GGSN	HLR	SGSN	GGSN
13	GGSN maintains routing information that is necessary to tunnel the_____to the	PDUs	PDPs	HLRs	VLRs	PDUs
14	The_____also performs authentication and charging functions related to data	HLR	GGSN	SGSN	PDP	GGSN
15	_____is used to protect the transmitted data packets against errors	Conventional coding	Reliable coding	Channel Coding	Channel encoding	Channel Coding
16	In GPRS network architecture under very bad channel conditions_____schem	Channel encoding	Channel coding	Reliable coding	Conventional coding	Reliable coding
17	In reliable coding scheme many redundant bits are added to recover from _____	coding	burst	bit	omission	burst
18	In reliable coding scheme a data rate of_____is achieved per time slot	9.05 Kbps	10.05 Kbps	11.05 Kbps	12.05 Kbps	9.05 Kbps
19	Under good channel conditions,no encoding scheme is used resulting in a higher data	20.5 Kbps	21.4 Kbps	12.5 Kbps	35.5 Kbps	21.4 Kbps
20	With_____time slots,a maximum data rate of_____can be achieved	7 and 170.5 Kbps	8 and 171.2 Kbps	6 and 172.5 Kbps	7 and 162.5 Kbps	8 and 171.2 Kbps
21	A user is likely to use either of the_____modes of the GPRS network	Application or tunnelin	Silent or ringing	Vibrate or meeting	physical or logical	Application or tunneling
22	In_____mode the user will be using the GPRS mobile phone to access the appli	Tunneling	Silent	Vibrate	Application	Application
23	All GPRS phones have_____browser as an embedded application	IE	Netscape	WAP	Chrome	WAP
24	_____mode is for mobile computing where the user will use the GPRS interface	User	Silent	Application	Tunneling	Tunneling
25	In Tunneling mode the mobile phone will be connected to the device and used as a	network	Interface	modem	adapter	modem

26	The end user device will be a large foot print device like _____	Laptop	Palmtop	simputer	pager	Laptop
27	Laptop and PDA's are referred as_____devices	White box	Black box	Sand box	Glass box	Black box
28	The_____devices do not have display,keypad and voice accessories of a stan	Glass box	White box	Black box	Sand box	Black box
29	A_____terminal can be one of three classes A,B or C	SMS	GSM	GPRS	PTM	GPRS
30	A_____terminal supports GPRS,data and other GSM services such as SMS a	Class A	Class B	Class C	Class D	Class A
31	A_____terminal can make or receive calls on two service simultaneously	Class D	Class B	Class C	Class A	Class A
32	SMS is supported in_____terminal	Class B	Class A	Class C	Class A and Class B	Class A and Class B
33	_____can be received while a voice or data call is in progress	MMS	Bcard	SMS	Vcard	SMS
34	A_____terminal can monitor GSM amd GPRS channels simultaneously	Class A	Class D	Class B	Class C	Class B
35	A_____terminal supports only non-simultaneous attach	Class B	Class C	Class A	Class D	Class C
36	_____terminal can make or receive calls from only the manually selected netw	Class D	Class B	Class C	Class A	Class C
37	The support of SMS is_____for Class C terminals	Mandatory	Optional	Impossible	Possible	Optional
38	Due to_____, mobile internet browsing will be better suited to GPRS	lower bandwidth	extremely lower band	higher bandwidth	extremely higher ba	higher bandwidth
39	Access to corporate_____can add a new dimension to mobile workers	Intranets	Internets	Ethernets	Extranets	Intranets
40	_____is another generic application people may like to use while mobile	E-Commerce	E-Business	M-Commerce	M-Banking	M-Commerce
41	Indian banks are offering banking over _____	GSM/WAP	GPRS/WAP	CSD/GSM	CDMA/WAP	GPRS/WAP
42	_____is very popular service in internet and GSM over SMS	voice call	Video call	Chat	E-mail	Chats
43	_____can be sent as an electronic object or a printed one	Data	Picture	Text	Audio	Picture
44	GPRS network can be used to offer_____services	SMS	ATM	VPN	WAP	VPN
45	Many bank ATM machines use_____to connect with the ATM system with the	CSD	VPN	BPN	VSAT	VSAT
46	Many_____in india are migrating from VSAT to GPRS,as the_____in	Retail Shops and Spe	Banks and bandwidth	Mall and accuracy	Aircrafts and efficien	Banks and bandwidth
47	GPRS is expected to reduce the transaction time by _____	10%	15%	25%	30%	25%
48	GPS is a free-to-use global network of_____satellites run by US DoD	45	24	30	55	24
49	In india_____application is becoming popular in logistics industry	Vehicle Positioning	JobSheet dispatch	Unified Messaging	Telematics	Vehicle Positioning
50	All systems developed for_____system are built around GPRS and GPS tech	JobSheet dispatch	Unified Messaging	Intelligent Transpor	Vehicle Positioning	Intelligent Transportation

UNIT IV NOTES

UNIT IV – MOBILE AD-HOC NETWORKS

MOBILE Ad-Hoc Basic Concepts – Characteristics – Applications – Design Issues – Routing – Essential of Traditional Routing Protocols –Popular Routing Protocols – Vehicular Ad Hoc Networks (VANET) – MANET Vs VANET – Security

Mobile Phones:

Mobile device : A handheld **computing** device has an operating system (OS), and can run **mobile** apps. In the 2010s, most handheld devices are also equipped with Wi-Fi, Bluetooth, near field **communication** (NFC) capabilities and Global Positioning System (GPS) capabilities.

The **features of mobile phones** are the set of capabilities, services and applications that they offer to their users. mobile phones are often referred to as feature phones, and offer basic telephony.[*clarification needed*] Handsets with more advanced computing ability through the use of native soo try to differentiate their own products by implementing additional functions to make them more attractive to consumers. This has led to great innovation in mobile phone development over the past 20 years.

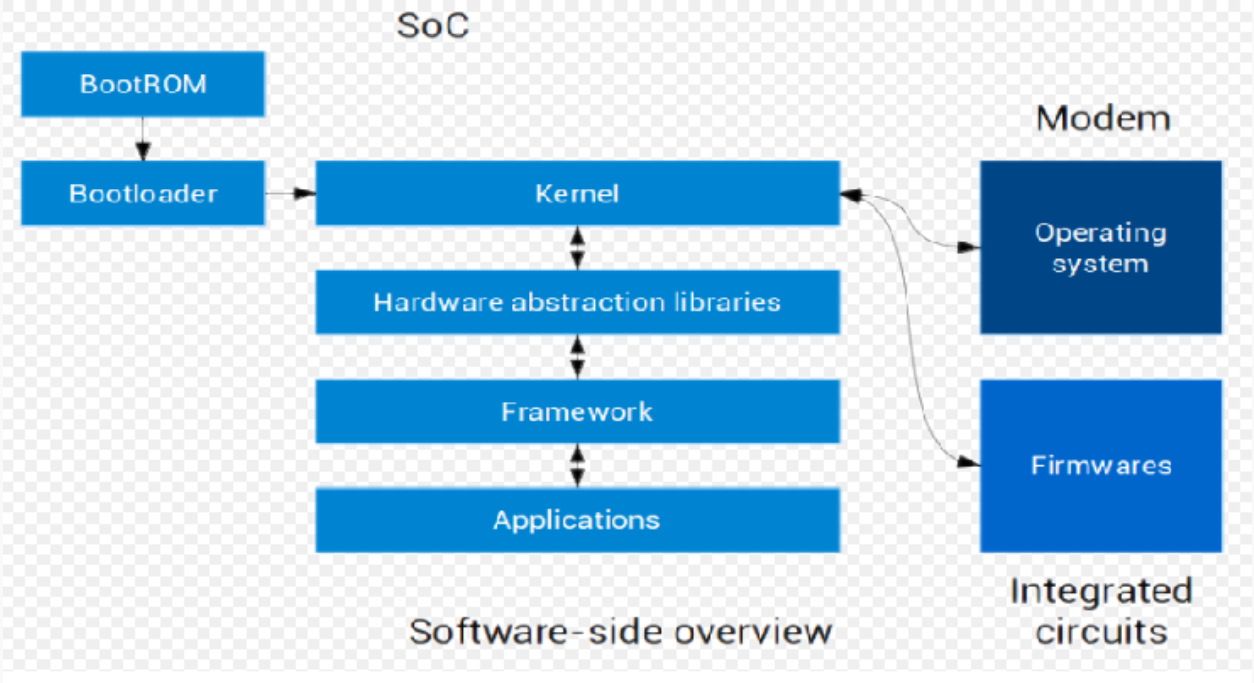
The common components found on all phones are:

- A battery, providing the power source for the phone functions.
- An input mechanism to allow the user to interact with the phone. The most common input mechanism is a keypad, but touch screensare also found in some high-end smartphones.
- Basic mobile phone services to allow users to make calls and send text messages.
- All GSM phones use a SIM card to allow an account to be swapped among devices. Some CDMA devices also have a similar card called a R-UIM.
- Individual GSM, WCDMA, iDEN and some satellite phone devices are uniquely identified by an International Mobile Equipment Identity (IMEI) number.

All mobile phones are designed to work on cellular networks and contain a standard set of services that allow phones of different types and in different countries to communicate with each other. However, they can also support other features added by various manufacturers over the years:

- roaming which permits the same phone to be used in multiple countries, providing that the operators of both countries have a roaming agreement.
- send and receive data and faxes (if a computer is attached), access WAP services, and provide full Internet access using technologies such as GPRS.
- applications like a clock, alarm, calendar and calculator and a few games.
- Sending and receiving pictures and videos (by without internet) through MMS, and for short distances with e.g. Bluetooth.
- In Multimedia phones Bluetooth is commonly but important Feature.
- GPS receivers integrated or connected (i.e. using Bluetooth) to cell phones, primarily to aid in dispatching emergency responders and road tow truck services. This feature is generally referred to as E911.

Software, applications and services:



- On early stages, every mobile phone company had its own user interface, which can be considered as "closed" operating system, since there was a minimal configurability. A limited variety of basic applications (usually games, accessories like calculator or conversion tool and so on) was usually included with the phone and those were not available otherwise. Early mobile phones included basic web browser, for reading basic WAP pages. Handhelds (Personal digital assistants like Palm, running Palm OS) were more sophisticated and also included more advanced browser and a touch screen (for use with stylus), but these were not broadly used, comparing to standard phones. Other capabilities like Pulling and Pushing Emails or working with calendar were also made more accessible but it usually required physical (and not wireless) Syncing. BlackBerry 850, an email pager, released January 19, 1999, was the first device to integrate Email.

- A major step towards a more "open" mobile OS was the symbian S60 OS, that could be expanded by downloading software (written in C++, java or phyton), and its appearance was more configurable. In July 2008, Apple introduced its App store, which made downloading mobile applications more accessible. In October 2008, the HTC Dream was the first commercially released device to use the Linux-based Android OS, which was purchased and further developed by Google and the Open Handset Alliance to create an open competitor to other major smartphone platforms of the time (Mainly Symbian operating system, BlackBerry OS, and iOS)-The operating system offered a customizable graphical user interface and a notification system showing a list of recent messages pushed from apps.

- The most commonly used data application on mobile phones is SMS text messaging. The first SMS text message was sent from a computer to a mobile phone in 1992 in the UK, while the first person-to-person SMS from phone to phone was sent in Finland in 1993.
- The first mobile news service, delivered via SMS, was launched in Finland in 2000. Mobile news services are expanding with many organizations providing "on-demand" news services by SMS. Some also provide "instant" news pushed out by SMS.
- Mobile payments were first trialled in Finland in 1998 when two Coca-Cola vending machines in Espoo were enabled to work with SMS payments. Eventually, the idea spread and in 1999 the Philippines launched the first commercial mobile payments systems, on the mobile operators Globe and Smart. Today, mobile payments ranging from mobile banking to mobile credit cards to mobile commerce are very widely used in Asia and Africa, and in selected European markets. Usually, the SMS services utilize short code.
- Some network operators have utilized USSD for information, entertainment or finance services (e.g. M-Pesa).
- Other non-SMS data services used on mobile phones include mobile music, downloadable logos and pictures, gaming, gambling, adult entertainment and advertising. The first downloadable mobile content was sold to a mobile phone in Finland in 1998, when Radiolinja (now Elisa) introduced the downloadable ringtone service. In 1999, Japanese mobile operator NTT DoCoMo introduced its mobile Internet service, i-Mode, which today is the world's largest mobile Internet service.
- Even after the appearance of smartphones, network operators have continued to offer information services, although in some places, those services have become less common.

Miscellaneous features:

- Other features that may be found on mobile phones include GPS navigation, music (MP3) and video (MP4) playback, RDS radio receiver, built-in projector, vibration and other "silent" ring options, alarms, memo recording, personal digital assistant functions, ability to watch streaming video, video download, video calling, built-in cameras (1.0+ Mpx) and camcorders (video recording), with autofocus[*dubious – discuss*] and flash, ringtones, games, PTT, memory card reader (SD), USB (2.0), dual line support, infrared, Bluetooth (2.0) and WiFi connectivity, NFC, instant messaging, Internet e-mail and browsing and serving as a wireless modem.
- The first smartphone was the Nokia 9000 Communicator in 1996 which added PDA functionality to the basic mobile phone at the time. As miniaturization and increased processing power of microchips has enabled ever more features to be added to phones, the concept of the smartphone has evolved, and what was a high-end smartphone five years ago, is a standard phone today.
- Several phone series have been introduced to address a given market segment, such as the RIM BlackBerry focusing on enterprise/corporate customer email needs; the

SonyEricsson Walkman series of musicphones and Cybershot series of cameraphones; the Nokia Nseries of multimedia phones, the Palm Pre the HTC Dream and the Apple iPhone.

- Nokia and the University of Cambridge demonstrated a bendable cell phone called the Morph. Some phones have an electromechanical transducer on the back which changes the electrical voice signal into mechanical vibrations. The vibrations flow through the cheek bones or forehead allowing the user to hear the conversation. This is useful in the noisy situations or if the user is hard of hearing.

Multi-mode and multi-band mobile phones:

- Most mobile phone networks are digital and use the GSM, CDMA or iDEN standard which operate at various radio frequencies. These phones can only be used with a service plan from the same company. For example, a Verizon phone cannot be used with a T-Mobile service, and vica versa.
- A multi-mode phone operates across different standards whereas a multi-band phone (also known more specifically as dual, tri or quad band) mobile phone is a phone which is designed to work on more than one radio frequency. Some multi-mode phones can operate on analog networks as well (for example, dual band, tri-mode: AMPS 800 / CDMA800 / CDMA 1900).
- For a GSM phone, dual-band usually means 850 / 1900 MHz in the United States and Canada, 900 / 1800 MHz in Europe and most other countries. Tri-band means 850 / 1800 / 1900 MHz or 900 / 1800 / 1900 MHz. Quad-band means 850 / 900 / 1800 / 1900 MHz, also called a world phone, since it can work on any GSM network.
- Multi-band phones have been valuable to enable roaming whereas multi-mode phones helped to introduce WCDMA features without customers having to give up the wide coverage of GSM. Almost every single true 3G phone sold is actually a WCDMA/GSM *dual-mode* mobile. This is also true of 2.75G phones such as those based on CDMA-2000 or EDGE.

Challenges in producing multi-mode phones:

- The special challenge involved in producing a multi-mode mobile is in finding ways to share the components between the different standards. Obviously, the phone keypad and display should be shared, otherwise it would be hard to treat as one phone. Beyond that, though, there are challenges at each level of integration. How difficult these challenges are depends on the differences between systems. When talking about IS-95/GSM multi-mode phones, for example, or AMPS/IS-95 phones, the base band processing is very different from system to system. This leads to real difficulties in component integration and so to larger phones.
- An interesting special case of multi-mode phones is the WCDMA/GSM phone. The radio interfaces are very different from each other, but mobile to core network messaging has strong similarities, meaning that software sharing is quite easy. Probably more importantly, the WCDMA air interface has been designed with GSM compatibility in

mind. It has a special mode of operation, known as punctured mode, in which, instead of transmitting continuously, the mobile is able to stop sending for a short period and try searching for GSM carriers in the area. This mode allows for safe inter-frequency handovers with channel measurements which can only be approximated using "pilot signals" in other CDMAbased systems.

- A final interesting case is that of mobiles covering the DS-WCDMA and MC-CDMA 3G variants of the CDMA-2000 protocol. Initially, the chip rate of these phones was incompatible. As part of the negotiations related to patents, it was agreed to use compatible chip rates. This should mean that, despite the fact that the air and system interfaces are quite different, even on a philosophical level, much of the hardware for each system inside a phone should be common with differences being mostly confined to software.

SMARTPHONES:

- A **smartphone** is a mobile phone (also known as cell phones) with an advanced mobile operating system which combines features of a personal computer operating system with other features useful for mobile or handheld use. Smartphones, which are usually pocket-sized, typically combine the features of a mobile phone, such as the abilities to place and receive voice calls and create and receive text messages, with those of other popular digital mobile devices like personal digital assistants (PDAs), such as an event calendar, media player, video games, GPS navigation, digital camera and digital video camera. Most smartphones can access the Internet and can run a variety of third-party software components ("apps"). They typically have a color display with a graphical user interface that covers 70% or more of the front surface. The display is often a touchscreen, which enables the user to use a virtual keyboard to type words and numbers and press onscreen icons to activate "app" features.
- In 1999, the Japanese firm NTT DoCoMo released the first smartphones to achieve mass adoption within a country. Smartphones became widespread in the late 2000s. Most of those produced from 2012 onward have high-speed mobile broadband 4G LTE, motion sensors, and mobile payment features. In the third quarter of 2012, one billion smartphones were in use worldwide. Global smartphone sales surpassed the sales figures for regular cell phones in early 2013. As of 2013, 65% of mobile consumers in the United States owned smartphones. By January 2016, smartphones held over 79% of the U.S. mobile market.

PERSONAL DIGITAL ASSISTANT (PDA)

A **personal digital assistant** (**PDA**), also known as a **handheld PC**, or **personal data assistant**, is a mobile device that functions as a personal information manager. The term evolved from Personal Desktop Assistant, a software term for an application that prompts or prods the user of a computer with suggestions or provides quick reference to contacts and other lists. PDAs

were largely discontinued in the early 2010s after the widespread adoption of highly capable smartphones, in particular those based on iOS and Android.

Nearly all PDAs have the ability to connect to the Internet. A PDA has an electronic visual display, enabling it to include a web browser, all models also have audio capabilities enabling use as a portable media player, and also enabling most of them to be used as mobile phones. Most PDAs can access the Internet, intranets or extranets via Wi-Fi or Wireless Wide Area Networks. Most PDAs employ touchscreen technology.

The first PDA was released in 1984 by Psion, the Organizer. Followed by Psion's Series 3, in 1991, which began to resemble the more familiar PDA style. It also had a full keyboard. The term *PDA* was first used on January 7, 1992 by Apple Computer CEO John Sculleyat the Consumer Electronics Show in Las Vegas, Nevada, referring to the Apple Newton.In 1994, IBM introduced the first PDA with full mobile phone functionality, the IBM Simon, which can also be considered the first smartphone. Then in 1996, Nokia introduced a PDA with full mobile phone functionality, the 9000 Communicator, which became the world's best-selling PDA. The Communicator spawned a new category of PDAs: the "PDA phone", now called "smartphone". Another early entrant in this market was Palm, with a line of PDA products which began in March 1996. The terms "personal digital assistant" and "PDA" apply to smartphones but are not used in marketing, media, or general conversation to refer to devices such as the BlackBerry, iPad, iPhone or Android devices.

Personal Digital Assistants

Personal digital assistants (PDAs) are small, hand-held computers. They use a form of touch screen technology, using a light pen or stylus as an input device rather than a keyboard. In addition, a detachable keyboard or a voice recorder can be used. PDAs are versatile information processing appliances. They are used frequently as personal information managers (PIMs) to record telephone numbers, addresses, appointments, and to-do lists. Also, PDAs can synchronize with microcomputers to transfer e-mail, text documents, spreadsheets, files, or databases. Other types of PDAs incorporate an **integrated modem** to connect with the Internet or to dial-up another computer in order to transfer data.

Some PDAs use wireless communications technology, providing great mobility. This use of wireless technology allows companies to provide PDAs to their employees in situations where laptops would be unworkable.

Palmtops, which are small computers that fit in the palm of one's hand, are also referred to as PDAs. The difference between a PDA and Palmtop computer is that the PDAs are pen-based, using a stylus rather than a keyboard for input, while the palmtop uses a small, integrated keyboard.

In 1993 Apple Computer introduced the first PDA, which was called the Newton MessagePad. Rather than just storing handwritten words, Newton converted them into typescript. Early versions of Newton had limited success with this difficult process. Three years later, 3Com's

Palm Computing introduced the revolutionary PalmPilot. In June 1998, Microsoft began shipping a scaled down Windows operating system for manufacturers of palm-sized PCs. Apple's Newton was later taken off the market.

PDA Features

Several factors should be considered before purchasing a PDA, including what application software the user wishes to run; which types of input devices (keyboard, light pen, touch screen, voice recorder) are available and desired by the user; whether the amount of memory, specifically **random access memory (RAM)** , and the life of the battery are sufficient for the user' needs; what size of the device is desired; and, of course, what price the user wants to pay. In 2001 prices for PDAs ranged from a low of $100 for a simple personal information manager to more than $1,000 for full-screen PDAs with integrated keyboards.

Other features available in PDAs are multimedia and audio capabilities, various screen sizes (starting with a full screen), and modems (integrated or not, wireless, or wired). Which features are most important depends on the tasks that the user wishes to perform. In addition to these standard features, mobile telephones or clip-on cameras can be combined with several PDA models.

Primary PDA Manufacturers

Two of the leading manufacturers of PDAs are Palm, Inc. and Casio, Inc. PDAs made by Palm dominate the market with two popular models: the Palm m100 and the Palm VIIx. Both devices use the Palm OS 3.5 as an operating system, HotSync Manager 2.1.0, Pocket Mirror 2.0.7 as synchronizing software, and a 20-MHz Motorola DragonBall EZ as a processor. The Palm OS is fast, simple, and compatible with both Macintosh and Windows-based computers.

The amount of memory available in these devices is important, especially the amount of RAM. The Palm m100 has 2MB of RAM and 2MB of ROM. The Palm VIIx has more RAM (8MB) and the same amount of read only memory (ROM). Both use a touch screen as their input and have a speaker as well as an infrared port. To exchange data between Palm PDAs, users can beam information from one to another using an infrared port. Users also can place the Palm PDA into a docking cradle and synchronize their PDA with their microcomputers. The Palm VIIx has a docking serial cradle for synchronizing with a microcomputer and a built-in modem that can access the Internet or other networks without wires.

The PDA models produced by Casio offered the first 64-bit screens on the market. Casio's PDAs include the Casio Cassiopeia E-125 and the Cassiopeia EM-500. Both have color displays, use Windows CE 3.0 as an operating system and ActiveSync 3.1 as synchronizing software, and use a 150-MHz NEC VR4122 as a processor. Most of these use the Windows CE operating system (WinCE). WinCE is similar to Windows 95/98. It can run set-top TV controllers or small handheld devices that can communicate with each other and synchronize with Windows-based computers.

The Casio E-125 PDA has 32MB RAM while the EM-500 PDA has 16MB RAM. Both use touch screen or voice recorders as input and have a microphone and speaker as well as a stereo headphone jack and an infrared port.

Making PDAs More Useful:

PDAs must be trained to recognize their users' handwriting. To do this, the user must write each numeric digit and letter (in uppercase) several times on the screen with the light pen. Even so, recognition is not always 100 percent. For users who really want a keyboard, text can also be entered by tapping the light pen on the appropriate letters from an on-screen virtual keyboard. There are other problems with entering data on the screen with just a light pen. How can users delete an entire word, or bring up an entire document? These problems have been solved in several ingenious ways. For example, deletion of a word occurs by crossing it out on the screen. Tapping the light pen on the name of a stored document brings it up on the screen.

As these problems are being solved, new applications for PDAs are being developed and new technologies are being combined with PDAs. For example, Symbol Technologies, Inc. has combined PDAs with **bar code** scanning equipment. Symbol is supplying the Kmart Corporation with in-store wireless and mobile computing solutions in its entire chain of more than 2,100 stores. Kmart is using Symbol's wireless **local area network (LAN)** and the company's PDT 6840 wireless devices for receiving merchandise, tracking inventory, and printing labels, and on the sales floor for price checking and employee communications. Many market watchers believe that PDAs will gain wide use in a variety of business applications, especially retailing, warehousing, and inventory control.

Palm OS:

Palm OS (also known as **Garnet OS**) is a discontinued mobile operating system initially developed by Palm, Inc., for personal digital assistants (PDAs) in 1996. Palm OS was designed for ease of use with a touchscreen-based graphical user interface. It is provided with a suite of basic applications for personal information management. Later versions of the OS have been extended to support smartphones. Several other licensees have manufactured devices powered by Palm OS.

Following Palm's purchase of the Palm trademark, the currently licensed version from ACCESS was renamed Garnet OS. In 2007, ACCESS introduced the successor to Garnet OS, called Access Linux Platform and in 2009, the main licensee of Palm OS, Palm, Inc., switched from Palm OS to webOS for their forthcoming devices.

Palm OS Device Architecture

At the highest level, the architecture of the Palm OS device, and most other PDAs, can be broken down into three layers (Figure 1): Application, Operating System, and Hardware.

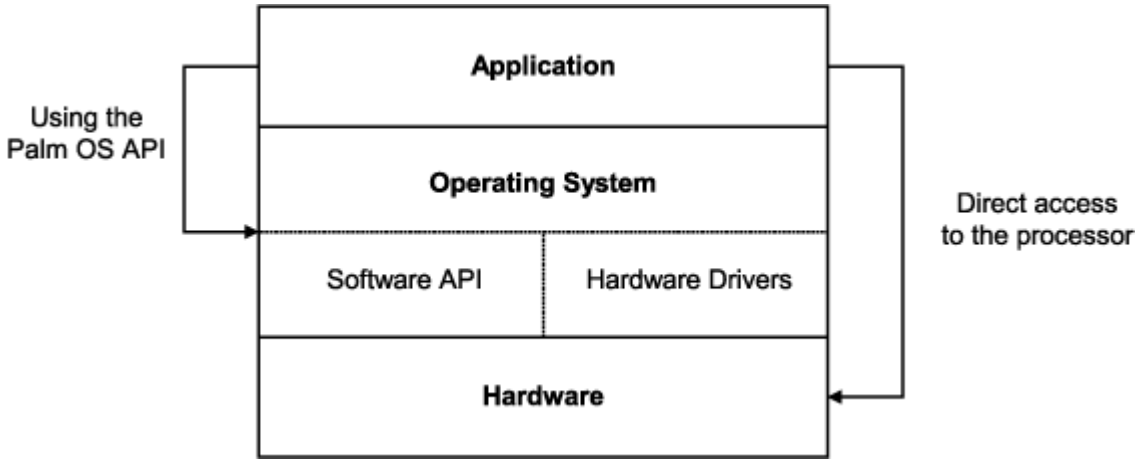


Figure 1: Typical layered architecture of a PDA

Use of the Palm OS Application Programming Interface (API) provides the application developer with a notion of hardware independence and provides a layer of abstraction. If the API is used properly, recompiling of the application is all that is necessary in order to run on Palm OS devices based on different hardware. Therefore, it is important to examine weaknesses and attack vectors that can be found at the programming interface to the operating system.

Directly accessing the processor by avoiding the interface put forward by the operating system allows the developer to have more control of the processor and its functionality. A risk of legitimate use of direct processor access is the loss of compatibility for future models. For example, older Palm OS devices did not support a grayscale LCD palette through the Palm OS API, even though the underlying hardware possessed this capability.

Palm OS supports three kinds of communication:

Serial

Serial communication occurs between the handheld and other devices using the cradle port. This is the most common form of communication on the Palm OS. For an example, we'll develop a special serial application that communicates (indirectly) with satellites.

TCP/IP

Currently, this communication standard is available only via a serial or modem connection. The future has no boundaries, however, so you may see built-in Ethernet or devices using wireless TCP/IP appear some day. To show you how to use TCP/IP, we'll create a small application that sends email to a server.

IrDA

This is an industry-standard hardware and software protocol for communicating over infrared. We won't discuss the details of communicating using IrDA in this chapter. We will, however, show you how to use the Exchange Manager to implement beaming. Beaming is a data-exchange method built on top of IrDA.

Symbian OS:

Symbian was a mobile operating system (OS) and computing platform designed for smartphones. Symbian was originally developed as a closed-source OS for PDAs in 1998 by Symbian Ltd.[] Symbian OS was a descendant of Psion's EPOC, and runs exclusively on ARM processors, although an unreleased x86 port existed. Symbian was used by many major mobile phone brands, like Samsung, Motorola, Sony Ericsson, and above all by Nokia.

Java on the Handset—J2ME:

The Plot was also available for a large population of Java-capable handset owners. The Java game was a stripped-down version of the N-Gage game: The missions were similar but smaller and simpler and did not use 3D graphics. Still, a number of missions were available for download and offered challenges that varied from tough puzzles to easy action sequences.

The emergence of Java, an application programming language and runtime environment developed by Sun Microsystems, revolutionized software development starting from the mid-1990s. The core idea behind the design of Java language was the capability to make the runtime environment available on as many different computing environments as possible and thus maximize the number of places where Java programs could be used. Now a special version of Java is aiming at the same goal in the mobile world and is widely supported by handset manufacturers around the world.

Java 2 Platform Micro Edition, J2ME, is a version of Java 2 technology that is optimized for use with small devices such as PDAs and cell phones.41 It has millions of potential developers because the larger-scale versions of the language and runtime environments are extensively used in both server and application programming.

The J2ME applications, properly called MIDlets, do not run natively in mobile devices. This means that the device must be provided with a special code interpreter, Java Virtual Machine, for executing MIDlets. Consequently, software written with Java needs more resources than the native applications that do not need an interpreter to run.

Because of the memory limitations of low-end cell phones, the permitted application size is relatively small. In some cases, the maximum application size is only a few dozen kilobytes. Newer handsets allow larger Java applications.

There are two ways to deliver mobile Java applications. The most common way is to download over the air (OTA) directly to the cell phone. Another option is to first download the MIDlet to a desktop PC and then install it to a phone via a serial cable, infrared, or Bluetooth connection.

Carriers encourage Java OTA downloads because they are able to generate revenue from the increased airtime. Additionally, they charge the third-party content providers for the billing of the transaction. In many cases, carriers sign revenue-sharing deals with game publishers and developers to generate more data traffic and to promote OTA download as a delivery channel.

Windows CE:

Windows CE

In 1995, Microsoft started the development of the new operating system called Windows CE under the code name Pegasus. The graphical user interface and usability has similarities with Windows 95 and has been designed especially for embedded devices. The application programming interface (API) contains a subset of the Win32 API used for Windows on personal computers. At the beginning, Handheld PCs and Personal Digital Assistents (PDAs) are supported with the processor architectures as MIPS 3000/4000, SuperH of Hitachi and Intel x86. The 32-bit operating system has a limit of up to 32 processes with up to 32 MB virtual memory for each process. The minimum requirements are stated as 4 MB ROM and 2 MB RAM.

Windows CE 1.0 has been presented at the COMDEX, Las Vegas in November 1996. One of the first devices for Windows CE was HP 300 LX and available at the market on 16th November 1996. The operating system is not sold separately and always tied to the device unit delivered by the original equipment manufactures. The resolution of the touch screen is 640x240 pixel and corresponds to the half-VGA resolution. Data can be synchronized between mobile device and desktop computer with the "Handheld PC Explorer" software. Altough the localization in German was planned, the first Version of Windows CE was only available in English.

At the beginning there have been speculations about what "CE" is standing for. Interprations like "Compact Edition" and also "Consumer Electronics" was spread. Microsoft declared later that "CE" is not an abbreviation but represents more the rules of orientation during the development to create an new operating system that is more "Compact", Connectable", Compatible", an "Companion," and "Efficient".

Different Flavours of Windows CE:

Windows CE 2.0 came in October 1997 with the first devices on the market. The operating system can now be designed modular by the device manufacturers itself. TrueType fonts improving now the appearance of characters on the screen with a display of 640x480 pixel full VGA resolution and 24-bit color depth. The manageable memory can now be up to 4 MB. The software "Handheld PC Explorer" was renamed to ActiveSync. The operating system is also available in German now.

The update **Windows CE 2.10** in July 1998 allows the use of TCP/IP and the file system FAT32. With the modular file wrapper can be incorporated up to 256 different file systems. The RAM can now be up to 16 MB. The new command line processor allows in this release for the first

time the use of commands without a graphical user interface. An infrared port and USB controller increases the scope.

Pocket PC 2003 or **Windows Mobile 2003** has a real-time kernel and is based on the Windows CE. NET 4.2 kernel. The features of the XScale CPU are now fully available and use the advanced commands of the ARM v5 architecture. The Pocket Media Player can now play videos in Windows Media 9 format. A WLAN stack was added and the setup of the connections was made easier.

Windows Mobile 2003 SE from March, 25. 2004 for Pocket PCs can switch over the display contents between the portrait and landscape format and displays up to 480 × 640 pixels. The start menu was changed light and the handwriting recognizer software Transcriber is controllable with Shortcuts now. The Internet Explorer converts web pages to a adapted format for better view on small displays.

Windows Mobile 5.0 was finished on May 5th, 2005. It is used in Pocket PCs, Smartphones and compact Media Players. Important innovations in this release are the support of Persistant Storage to prevent a data loss at a low battery usage and the revised mobile Office with Word, Excel and Powerpoint. The Windows Media player was updated to version 10 and the sync software ActiveSync to version 4.0. With Direct3Dmobile a new standard API was created for a simplified programming of 3D applications and games for PDAs.

Windows Embedded CE 6.0 was introduced in the year 2006. It offers a revised kernel architecture of the operating system, up to 32,000 parallel processes can be executed. A virtual addressable range of 2 gbyte is possible for every process. The multimedia capabilities have been expanded and now support HD-DVD, DVD (MPEG-2), UDF 2.5, multi-channel audio and much more. The compatibility to existing Windows CE applications and drivers are kept. Microsoft published the whole source code of the kernel within the own Shared Source Initiative.

Microsoft announced at the "CTIA Wireless" **Windows Mobile 6.1** in April 2008. This new release is to be easier for the access to messages and to configure. The integrated web browser supports Adobes Flash, Microsoft Silverlight and the video codec H.264. Functions for Smartphones have been improved.

The market research group Gartner, Inc. released a press news about the worldwide market share of PDA ventors and operating systems on 22nd May 2007. 3.1 million Windows CE units have been sold in the 1st quarter 2007. This represents a dominant position with a market share of 62.1% by operating systems. The Windows CE operating system increased his share compared to the 1st quarter 2006 with 52.8% and 1.9 million units sold.

The market share changed significantly 10 years later. On 19th May 2016, Gartner released the following figures. With 8.2 million Windows units sold in 1st quarter 2015 and 2.3 million devices in 1st quarter 2016, this is equal to a market share of 2.5% and 0.7% only. As a result the Windows operating system is on 3rd place behind iOS and Android the market leader.

UNIT IV - POSSIBLE QUESTIONS

PART – B
(Each question carries six marks)

1. Discuss about PDA with suitable example.
2. Explain Palm OS architecture with suitable example.
3. Explain the mobile phones and its features with suitable diagram.
4. Write the Symbian OS architecture with suitable diagram.
5. Explain the components of PDA with suitable diagram.
6. Write the kernel features and memory architecture of Palm OS with suitable diagram.
7. Write notes on the following components
   i)Microprocessor ii)Batteries iii) LCD Display iv)I/O ports.
8. Explain the connection manager architecture of Palm OS with suitable diagram.
9. Write notes on security architecture in Symbian with suitable example.
10. Explain the OS layer of Windows CE with suitable diagram.

PART – C
Compulsory Question
(Each question carries ten marks)

1. What should be the characteristics of Mobile Computing Devices?
2. Describe the security aspects in mobile computing.
3. Explain the functionality of Mobile Computing
4. How would you broadly classify the mobile computing applications?
5. Who are all the stakeholders of wireless network?
6. What is WiMax? How it is differ from WiFi?
7. Describe 3G networks.
8. Explain the Process of GSM Call Routing.
9. Explain the Process of Interactive Voice Response.
10. Briefly describe the Eclipse IDE for the development of android application.

MOBILE COMPUTING (18CAP503)

UNIT- IV

S.No	Question	Option1	Option2	Option3	Option4	Answer
1	In a Cell phone_____is the signal reception unit	Processor	Battery	Circuit Board	Antenna	Antenna
2	_____is the control unit of a cell phone	Circuit board	Processor	Battery	ROM	Circuit board
3	_____controls and coordinates the handset functions	Battery	memory cards	microprocessor	Circuit board	microprocessor
4	The most important of microprocessor controls are user input/output and_____intefaces	physical	network	application	transport	network
5	_____is a highly sophisticated chip that manages signal manipulations	ROM	DSP	RF	LCD	DSP
6	The_____chip manages the signal channels while the power section is responsible for po	RF	ROM	LCD	DSP	RF
7	In Cell phone, display unit is the_____unit	input	output	control	memory	output
8	Keyboard is referred as_____in mobile phone	output	exit	qwerty	entry	qwerty
9	_____is needed to facilitate speech transmission	Microphone	Headphone	Blue tooth	Speaker	Microphone
10	_____is the microphone's listener counter part	Headphone	Speaker	Microphone	Bluetooth	Speaker
11	_____is the source of electrical energy	ROM Chip	Headphone	Speaker	Battery	Battery
12	_____is unlike the phone evolved from the PCs	Desktop	Palmtop	PDA	TFT	PDA
13	_____have a similar architecture as most desktops have	PDA	Palmtop	CRT	TFT	PDA
14	_____are considered a portable extension to the desktop	Laptops	PDAs	PCs	Mobiles	PDAs
15	PDAs have an operating system and mimic the application such as_____on the desktops	chat,SMS	MMS,voice call	addressbook,email	bluetooth,camera	addressbook,email
16	PDAs use cheap low -end processors such as _____	Apple iball	Intel	Intel 8086	Motorola Dragonba	Motorola Dragonbal
17	The processor in PDA have a very basic speed capacity of about _____	20-100 MHz	15-25 MHz	16-75 MHz	55-85 MHz	16-75 MHz
18	The PDA has no concept of _____	Memory	Hard drive	Input	Output	Harddrive
19	In PDA, it is common to have about_____MB of memory	1	2	3	4	2
20	In PDA , the user data is stores in _____	RAM	ROM	Main Memory	Secondary memor	RAM
21	The Batteries used in PDA last for about_____hours on an average	2	3	4	1	2
22	Major culprits that drain batteries in PDAs are _____	Memory	Color LCD Display	RAM	ROM	Color LCD Display
23	The LCD screen of PDA have grayscale of about _____	15	12	16	18	16
24	The LCD screen of PDA have color of about _____	87,882	43,677	70,787	65,536	65,536
25	Software inside the PDA converts the_____to letters and numbers	digits	bits	characters	objects	characters

26	On_____devices ,the software that recognizes letters is called Graffiti	Mobile	PDA	Handheld	Palm	Palm
27	Palm OS consists of_____layers	1	2	3	4	3
28	Palm OS occupies_____bytes of memory	100K	300K	200K	250K	300K
29	Palm OS can run in_____bytes of RAM	32K	26K	16K	20K	32K
30	In Palm OS the_____supports both maskable and non-maskable interrupts in normal an	Memory	RAM	Kernel	Software	Kernel
31	In Palm OS the_____supports a mechanism to trap errors	Kernel	Hardware	Software	Memory	Kernel
32	In Palm OS the_____can initiate other tasks	Multitasking	Interrupts	Scheduling	Processess	Interrupts
33	Palm OS Memory uses_____bit addresses	16	32	12	8	16
34	Theoritically, the Palm memory can have_____memory cards	356	366	256	656	256
35	Palm OS memory is divided into_____logical heaps	1	2	3	4	3
36	In Palm OS the_____heap contains the operating system's global variables and data typ	Static	Storage	ROM	Dynamic	Dynamic
37	In Palm OS the_____heap holds the operating system's kernel	ROM	Dynamic	Storage	Static	ROM
38	In Symbian OS the kernel is less than_____Kb	100	200	300	350	200
39	The main objective of the Symbian OS is to provide_____and_____resources	Hardware abstraction	Application,Physical	Software abstraction	Reliable,Efficient	abstraction,Manag
40	Symbian OS supports _____	pre-emptive multitaski	non-preemptive multit	single tasking	multitasking	pre-emptive multita
41	The_____file contains the libraries and resources of the application,secured by a certifica	SIS	DLL	Exe	Obj	SIS
42	Symbian offers two development enviroments ,they are_____and _____	C ,C++	C,Java	C++,Java	Java,Java Applet	C++,Java
43	Security in symbian OS includes_____for standard cryptographic algorithms	APIs	SSL	WTL	IPSec	APIs
44	_____is needed to define a computing platform that could accommodate consumer electro	J2EE	J2ME	Java applet	J2SE	J2ME
45	J2ME is a subset of _____	J2ME	J2EE	J2SE	Java applet	J2SE
46	The_____layer is responsible for getting a Windows CE-based OS to run on a new hardw	OEM	OAL	POR	DLL	OEM
47	Windows CE to become compact, the size of ROM is less than_____KB	100	400	200	512	200
48	Windows CE maps the bottom section of memory into 33,32 Mb slices called _____	Segment	Slots	Fragements	Blocks	Slots
49	_____slots are alloted for user processes	20	35	28	38	28
50	Windows CE is a_____machine	64-bit	16-bit	48-bit	32-bit	32-bit
51	The address space of Windows CE is _____	2GB	4GB	8GB	12GB	4GB

sking

UNIT V NOTES

UNIT V – ANDROID OPERATING SYSTEM

History of Android -Introduction to Android Operating Systems -Android Architecture - Android Virtual Device Manager - Features of Eclipse and Android Studio-Comparison of Kotlin Language to Java- User Interface Architecture of Android: Application context, intents, Activity life cycle, User Interface Design of Android –Features of Android SQLite Database

Getting to know Android:

Android is an open source and Linux-based **Operating System** for mobile devices such as smartphones and tablet computers. Android was developed by the *Open Handset Alliance*, led by Google, and other companies.

Android offers a unified approach to application development for mobile devices which means developers need only develop for Android, and their applications should be able to run on different devices powered by Android.

The first beta version of the Android Software Development Kit (SDK) was released by Google in 2007 where as the first commercial version, Android 1.0, was released in September 2008.

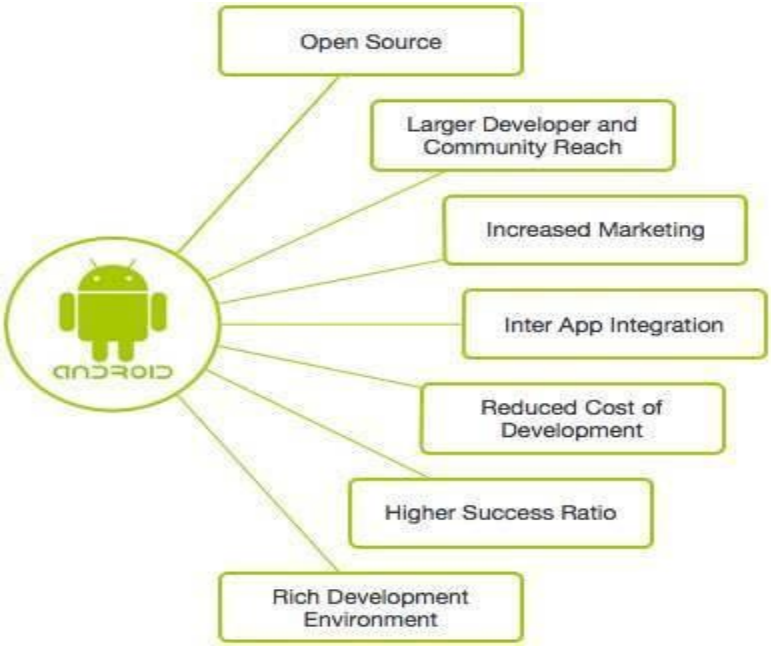
On June 27, 2012, at the Google I/O conference, Google announced the next Android version, 4.1 **Jelly Bean**. Jelly Bean is an incremental update, with the primary aim of improving the user interface, both in terms of functionality and performance.

The source code for Android is available under free and open source software licenses. Google publishes most of the code under the Apache License version 2.0 and the rest, Linux kernel changes, under the GNU General Public License version 2.

Why Android ?

Features of Android

Android is a powerful operating system competing with Apple 4GS and supports great features. Few of them are listed below



Sr.No.	Feature & Description
1	**Beautiful UI** Android OS basic screen provides a beautiful and intuitive user interface.
2	**Connectivity** GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, NFC and WiMAX.
3	**Storage** SQLite, a lightweight relational database, is used for data storage purposes.
4	**Media support** H.263, H.264, MPEG-4 SP, AMR, AMR-WB, AAC, HE-AAC,

	AAC 5.1, MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF, and BMP.
5	**Messaging** SMS and MMS
6	**Web browser** Based on the open-source WebKit layout engine, coupled with Chrome's V8 JavaScript engine supporting HTML5 and CSS3.
7	**Multi-touch** Android has native support for multi-touch which was initially made available in handsets such as the HTC Hero.
8	**Multi-tasking** User can jump from one task to another and same time various application can run simultaneously.
9	**Resizable widgets** Widgets are resizable, so users can expand them to show more content or shrink them to save space.
10	**Multi-Language** Supports single direction and bi-directional text.
11	**GCM** Google Cloud Messaging (GCM) is a service that lets developers send short message data to their users on Android devices, without needing a proprietary sync solution.

12	**Wi-Fi Direct** A technology that lets apps discover and pair directly, over a high-bandwidth peer-to-peer connection.
13	**Android Beam** A popular NFC-based technology that lets users instantly share, just by touching two NFC-enabled phones together.

Android Development Environment

Android applications are usually developed in the Java language using the Android Software Development Kit.

Once developed, Android applications can be packaged easily and sold out either through a store such as **Google Play**, **SlideME**, **Opera Mobile Store**, **Mobango**, **F-droid** and the **Amazon Appstore**.

Android powers hundreds of millions of mobile devices in more than 190 countries around the world. It's the largest installed base of any mobile platform and growing fast. Every day more than 1 million new Android devices are activated worldwide.

This tutorial has been written with an aim to teach you how to develop and package Android application. We will start from environment setup for Android application programming and then drill down to look into various aspects of Android applications.

Set up your development environment

1. Check your system requirements
2. Install the Instant Apps Development SDK
3. Set up your device or emulator
4. Next steps

This page describes how to set up a development environment for Android Instant Apps onto a Linux, macOS, or Windows development machine.

Check your system requirements

To develop an instant app, you need the following:

- A development machine running Linux, macOS, or Windows.
- JDK 1.8
- Android Studio 3.0 or higher

- An environment variable called ANDROID_HOME that points to the location of the Android SDK on your development machine.

In Android Studio, use the Android SDK Manager to install the following packages:

- Android SDK 6.0 or higher
- Android SDK Build Tools 26.x or higher
- Android SDK Tools 25.x or higher
- Android SDK Platform Tools 25.x or higher
- Android Support Library (latest)
- Android Repository (latest)

Install the Instant Apps Development SDK

1. Go to **Tools > Android > SDK Manager** to open the Android SDK Manager.
2. Click the **SDK Tools** tab.
3. From the list, select **Instant Apps Development SDK**.
4. Click **Apply**.
5. After the **Component Installer** completes the installation, click **Finish**.

For more information about using the Android SDK Manager, see Android SDK Tool Updates.

Set up your device or emulator

You can develop instant apps on the following devices and emulators:

- **Devices:** Most Android devices running Android 5.1 (API level 21) or higher.
- **Emulator:** Nexus 5X image running Android 5.1 (API level 21) or higher, x86, with Google APIs. Create the emulator as described in Create and Manage Virtual Devices. Make sure you use the Google Play Store device profile.

Note: You cannot use x86_64 architectures of the operating system for testing instant apps on an emulator.

To setup your device or emulator, follow these instructions:

1. Go to **Accounts > Google** to log on to a Google account on the device or emulator. You can use a test account for this purpose.
2. If your device or emulator is running Android 7.1.1 (API level 25) or lower, Android Studio installs or updates the following when you first run an instant app.
    - *Google Play Services* - Update your Google Play Services. To do this, click the three dots on the emulator control, open the **Google Play** tab, and then click **Update**.
    - *Google Play for Services for Instant Apps* - The instant apps runtime.
    - *Instant Apps Development Manager* - A developer tool that lets you locally install and launch instant apps from your workstation (without deploying to Google Play).

To confirm the configuration settings on devices running Android 7.1.1 (API level 25) or lower, open the Settings app, and navigate to **Apps > Google Play Services for Instant Apps**. You can also run the command adb shell pm list packages | grep "com.google.android.instantapps.supervisor".

Android 8.0 (API level 26) has instant apps support built into the framework and there is no need to install these components.

In general, creating an Android app requires the SDK (Software Development Kit), an IDE (Integrated Development Environment) like **Android Studio** or **Eclipse**, the Java Software Development Kit (JDK) and a virtual device to test on. All this takes work to set up, and that's before you've even started looking into things like Google Play Services, screen sizes, API levels…

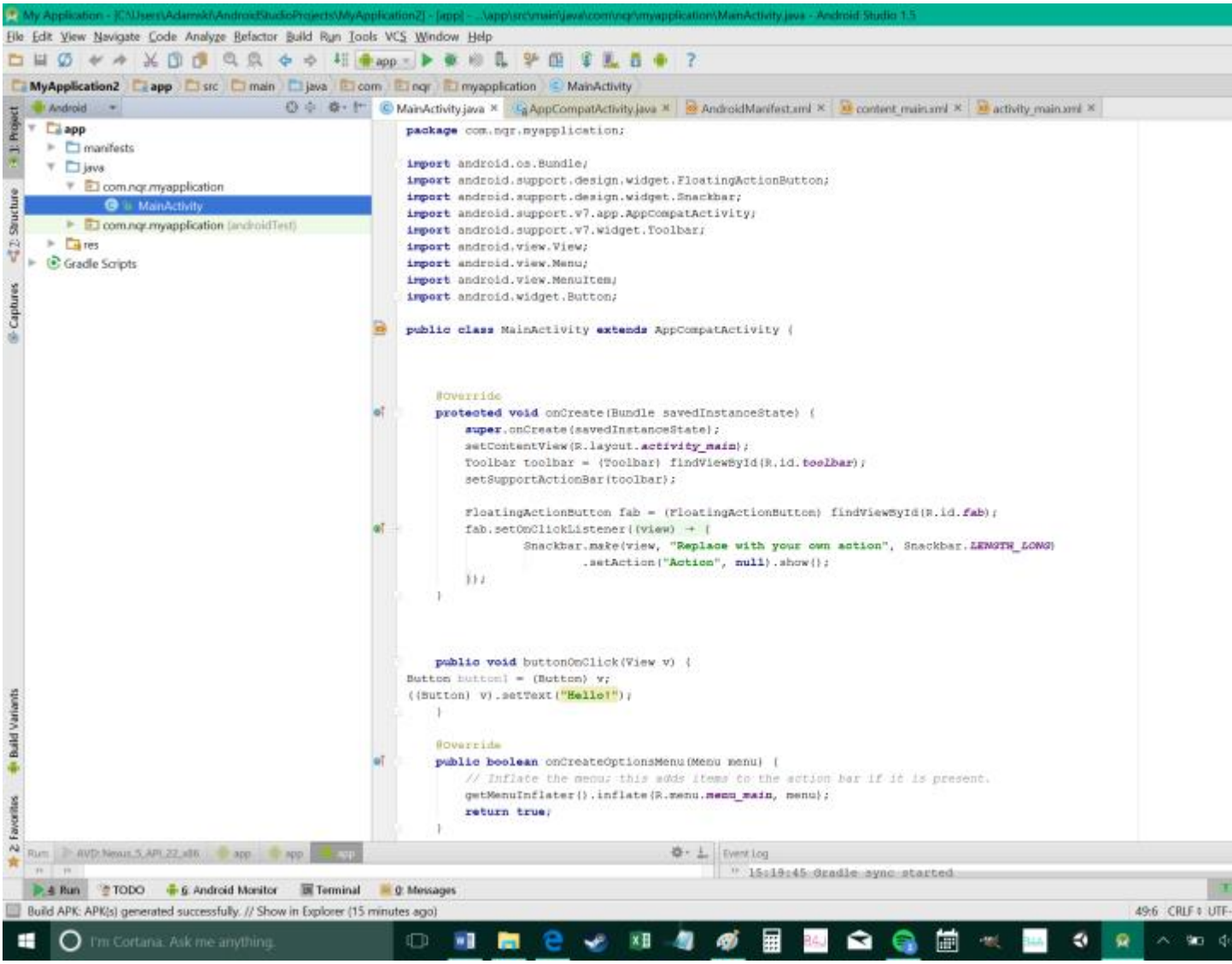
It's just such a dense amount of information and it's enough to put an awful lot of people off before they even begin. My aim with this article then, is to provide an approachable guide to try and make the whole prospect of creating an app a little less daunting… I'll explain the bits you need to know and gloss over the rest and by the end you should have a basic app that you can start iterating on and experimenting with.

Go and make yourself a cup of tea first though, this may take a while…

Step 1: Download Android Studio

To program in most languages, you need a piece of software called an IDE or 'Integrated Development Environment'. The most common IDE for Android development is **Android Studio**, which comes direct from Google itself. You can get it here.

An IDE is what gives you the main UI where you'll enter your code (you can't just start typing into notepad). It also highlights things you get wrong, offers suggestions and lets you run and test your creations conveniently. It creates the files you need, it provides basic layouts and generally it saves you a lot of time and effort.



What's great about Android Studio is that it is designed specifically for Android development (unlike the second most popular option, Eclipse). This means that when you download the software, you'll also get a lot of the other bits you need including the **Android SDK** (a selection of tools including the Android platform itself) and the **Android Virtual Device**, which is an emulator you can test your apps on. When you go through the installation, make sure you leave the boxes ticked to confirm that you want these additional components. You could manually add them later, but this will just complicate matters.

As mentioned, there are some alternatives to Android Studio. Eclipse is an older IDE that can be used for developing other things too (such as iOS apps) and that is a bit more flexible overall. It's also a much more fiddly to get started with though and not nearly as beginner-friendly. Another personal favorite of mine is Basic4Android. Basic4Android is an IDE that lets you code Android apps with the BASIC programming language. It makes things easier in a number of other ways too and is focused on 'rapid development'.

There are other options too, such as Unity3D and numerous app builders, each of which has specific strengths and weaknesses depending on what you're planning on building. For the sake of simplicity though, we're focusing on Android Studio because it has become the 'main' way to build basic apps and pretty much the industry standard. If you think you might ever sell your business, if you want to give yourself the most flexibility and control possible, or if you'd like to become a professional app developer, you'll need this tool.

That said, if you read through all this and you find it too much still, you might want to consider Basic4Android as a simpler approach and I'll be covering that in a future post.

Okay, just to recap: we now have Android Studio downloaded and installed. But, don't run it until you read step two! So far so good… What could possibly go wrong?

Step 2: Setting Up Android Studio

Now you have Android Studio installed you've taken your first, bold step toward becoming a developer! A lot of people only manage it this far and then leave the software installed on their computer for months on end, feeling guilty every time they see it in the Start Menu. Eventually they end deleting it to make space for the next AAA title on Steam and so ends the whole sorry affair… Don't end up like them – it's time for some more affirmative action!

Before you can get started, you also need to install Java on your machine to use Android Studio. Specifically, you're going to need install the Java Development Kit (JDK). Java is the programming language you're going to be using to build your apps in this instance and you need to install the JDK in order for Android Studio to be able to interpret and compile your code (compiling means turning the source into something that is understood by the CPU – machine code). You'll find the Java Development Kit here. Just download and follow the instructions to install.

Android Development Environment for Real Applications

You can click on Android Studio to launch it. Once it opens up, you'll be presented with a menu where you'll be able to get started or configure some options. The great thing is that everything is handled for you at this point, though you may want to familiarize yourself with the **SDK Manager** (Configure > SDK Manager) which is where you'll update your Android SDK to support newer versions, as well as download things like code samples or support for Google Glass. But don't worry about that now but if Android Studio says you're missing something, this is where you'll probably need to go to find it.

So really there are three main things interacting when you use Android Studio to create your apps.

- Android Studio itself, which is an IDE that provides you with a nice interface for coding.
- The code you write in Java, which you installed a moment ago…
- And the Android SDK which you'll access through your Java code in order to do Android-type things

If you find this all a bit complicated and daunting then… well, you don't know you're born. This used to be *way* worse.

Maybe that offers some consolation…

Step 3: Starting a New Project

Once you've installed your samples, you can go back to the first page you saw when you loaded up Android Studio. Now you want to choose **Start a new Android Studio Project** – it's finally happening!

Enter the name you want for your application and your 'company domain'. Together these elements will be used to create your package name with the following format:
com.companyname.appname
The package will be the compiled file or **APK** ('Android Package File') that you'll eventually upload to the Google Play Store. There are ways that people can see this, so if you're planning on making something you'll eventually release, try to stay away from using 'funny words'.

The last field to enter is the directory where you want to save all the files pertaining to your app. I like to save in DropBox to make sure I always have a backup of my code. Click Next again and guess what… More options! Huzzah! Don't worry, we're nearly there…
Next you need to decide what type of device you're going to be developing for and in this case we'll start with the **Phone and Tablet** option. Other options are TV, Wear and Glass. It's fine if you want to develop for a myriad of platforms in the future – that's one of the wonders of Android – but let's start with something a bit more straightforward to begin with, okay?
The other choice you have to make at this stage is the 'Minimum SDK'. This is the lowest version of Android you want to support. Why not just enter the latest version of Android in here? Well, because relatively few people actually *have* the latest version of Android installed on their device at any given time. You want to support phones that are still running older versions in order to reach the largest possible audience – especially overseas.
Why not just go with Android 1.1? Well, apart from this not being an option (Froyo is as low as you can go), that would also prevent you from using any of the fancy new features from the latest updates.
The best bet at this stage is to go with the default option, so just leave this field as it is. On the next page, you'll be given the option to pick the way you want your app to look at the start. This will be the look of your main '**Activity Module**' which is basically the main page of your app. Think of these like templates; do you want to have the title of your app along the top of the screen, or do you want your UI to fill the whole display? Do you want to start off with some elements ready-designed for you? Is your app primarily going to use Google Maps (don't go here for a bit, things get more complicated with Google Play Services).

Bear in mind that an app can have multiple activities that act like separate pages on a website. You might have a 'settings' activity for instance and a 'main' activity. So the activity isn't the *app* per say but rather one stand-alone page of your app.
For your first creation though, you'll probably do best to make something *really* simple that just displays a single, basic activity. Select '**Basic Activity**' to keep things as simple as possible and for all intents and purposes, this will now be your app. Click Next again you get the last few options.
Now you get to pick the name for your activity and the layout name (if you chose 'Basic Activity' you'll also have the title option and the 'menu_resource' name). The activity name is how you'll refer to your activities in your code, so call it something logical (good advice for coding generally) like 'MainActivity'. Creative, I know.
The layout name meanwhile describes a file that determines the layout of an activity. This is a separate piece of code that runs in concert with the main activity code to define where elements like images and menus go and what fonts you'll use. This is actually not Java but XML – or Extensible Markup Language if you want to impress your friends.

For anyone with a background in web development, your XML is going to work a little like HTML or a CSS style sheet. The Java code for the activity meanwhile says *what* the elements on the screen do when pressed etc. It's fine to leave the default name here as 'activity_main'. Lastly, choose a name for the menu and for the title. Pick something nice for the title, as your users will be able to see this at some points. Click next… and now you get to see your app!

Your blank, useless app… All that just to get started! You see why people give up? But really we can break it down into the following very basic steps:

- Download and install Android Studio, making sure to include the Android SDK
- Install Java SDK
- Start a new project and select the basic details

So it's really not that bad… And remember: once you've done all this once, you can forget about it forever and focus on the fun stuff: creating apps! Your tea is probably cold at this point, so the next very important step, is to get more.

Step 4: Making an Actual Thing

Once your app opens, you should see a directory tree on the left with all the different files and folders that make up your app and a picture of a phone displaying 'Hello World!' in the center. Well, hello to you as well!

(A basic app that displays 'Hello World' is what most new developers make first when they learn to program in a new language. Android Studio cheats though, because it does it for you!)

You might notice that the open tab (along the top) is 'activity_main.xml', which is what the big phone is showing on its display. You may recall that activity_main.xml is the XML code that defines the layout instructions for your main activity.

If you selected 'Basic Activity' when you started your project, then you'll see a second XML file too called 'content_main.xml'. For the most part, these two do the same thing but the 'acitvity_main.xml' contains the basic layout that Android Studio created for you when you selected 'Basic Activity'. The stuff you want to edit is in content_main.xml, so open that up and don't worry about it for now.

(If this isn't what is open to start, then use the directory on the left to open it by choosing: *app* > *res* > *content_main.xml.*)

The Layout

Android Studio is not showing the XML code itself here but rather a rendering of how the layout will appear on the screen. This is a visual editor a bit like Dreamweaver for web design and it makes life a little easier for us developers.

You also have a bunch of options called '**widgets**' down the left that you can add to your app. This is your basic app stuff; so for instance, if you want to add a button saying 'OK' to your activity, you can simply drag it over to the screen and drop it anywhere you like. Go ahead and dump an 'OK' button right underneath the 'Hello World'.

Something else you'll find is that you can click on either of these elements in order to change the text and the 'ID'. The ID is how you're refer to each element (called a '**view**') in your Java code, while the text is of course what you display to the user.

Delete the 'Hello World' widget (or view) and change the text on the button to 'Hello?'. Likewise, change the 'id' on the button to 'button1'.

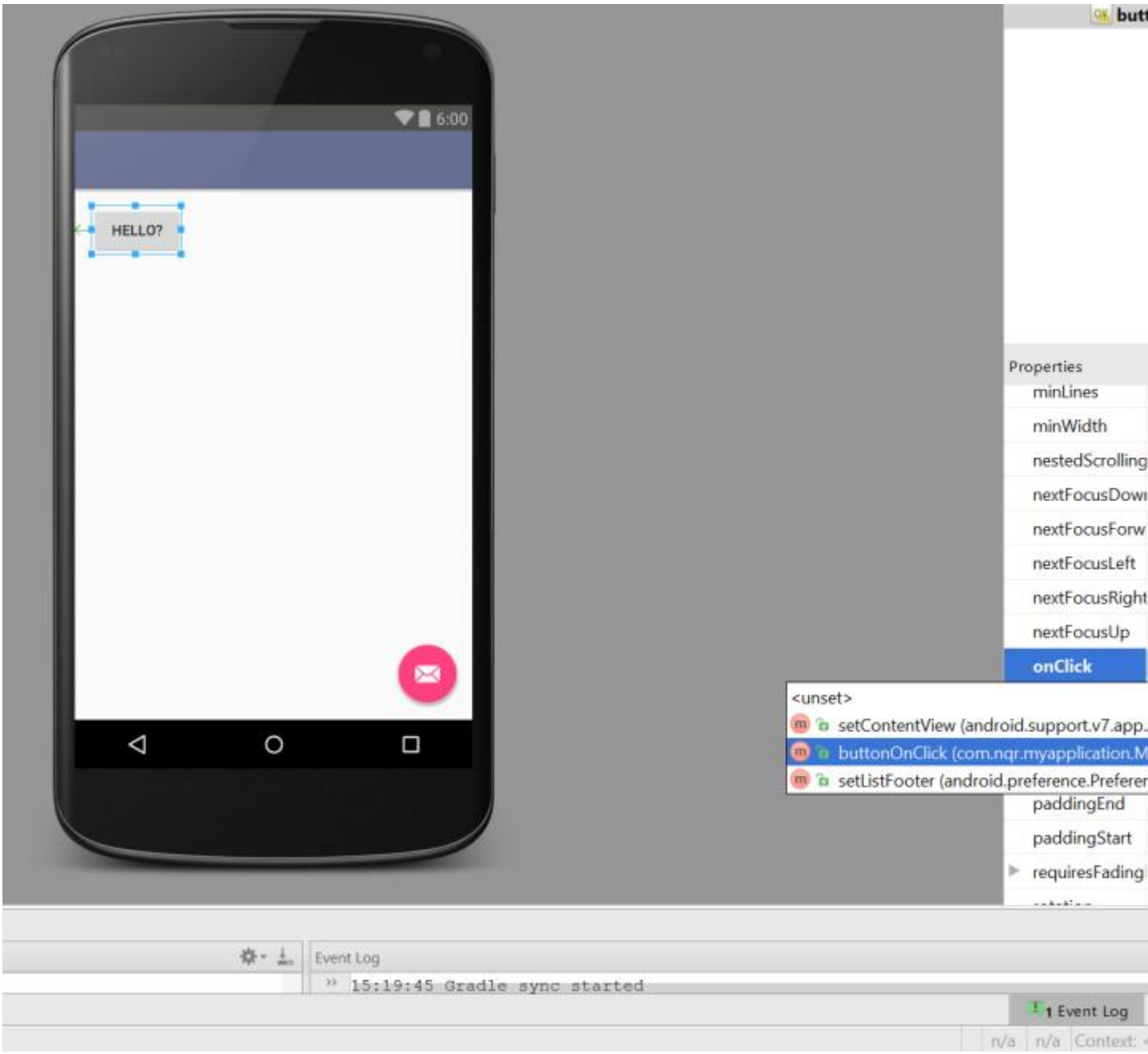
I am now stealthily getting you to write a little program… Notice as well that when you select a view, you get options in the bottom right to change the text color and size etc. You can play

around with these variables if you like to change the look of your button. We're coming back here in a minute though so make a mental note!
Now open up your MainActivity.java. The tab will be along the top but in case it isn't, find it under: *App > Java.*

What does it all mean? Well basically, anything following "**void buttonOnClick**" will be carried out when someone clicks on the button. We're then finding the button with the "**Button button1 = (Button) v;**" code and then changing the text.
Yes, there are other ways you could achieve the same thing but I feel like this keeps it nice and simple and thus easy to understand. Spend some time reading it and try to get your head around what is doing what…
At the top of the page is the word 'import…'. Click on that to expand it and make sure that somewhere there is the line: "**import android.widget.Button;**". It should have appeared on its own when you typed out the last bit (Android Studio is smart like that) but you can add it yourself if it didn't.

```
import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.View;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.Button;
```

(Notice as we type that lines end in ";". This is basic Java formatting and if you forget one, it will throw up an error. Get used to searching around for them!)
Now go back to your content_main.xml and click on the button. In the right corner, where you have your parameters for the button, you should be able to find an option called 'onClick'. Click on this and then select the 'onClick' line of code you just wrote from the drop down menu. What you've just done, is told Android Studio that you want to associate the section of code with the button you created (because you'll have lots of buttons in future).



Now all that's left to do is run the app you just made. Simple go to 'run' along the top and then select 'run app' from the drop down menu. You should already have your AVD (Android Virtual Device) installed but if not, you can go to: *tools > Android > AVD Manager > + Create Virtual Device*. Don't forget you also need to install an Android version *onto* the device.

START UP CODE

Follow the steps to launch the emulator running your app. Be patient, it can sometimes take an *age* to load up… If it *never* loads up, you can consider 'packaging' the app in order to create an APK. Drag this onto your Android device and double click on it to install and run it.

Once it's finally up and running you can have a go with this fun, fun app. What you should find is that when you click the button, the text from 'Hello?' to 'Hello!'. We're going to be rich…

(If it doesn't work… something has gone wrong. It wasn't me, my one works! Look for red text in your code and hover your mouse over it to get suggestions from Android Studio.)

Step 5: How to Get Better At App Development

Okay, so that was a lie. We're probably *not* going to be rich. At the moment the app we've made is pretty lame. You can try and sell it sure but you probably won't get that many good reviews.

The reason I talked you through this basic app creation though is because it teaches you the very fundamentals of programming. You have an action and a reaction – pressing on a button *does* something. Throw in some variables and some math, add some pretty images and a useful function and that's genuinely enough to make a very basic app.

So where do we go from here? There's so much more to learn: we haven't looked at the **Android Manifest** yet, we haven't talked about your **private keysign** (or how fun it is when you lose that) and we haven't even studied the Android app 'lifecycle' (nothing to do with *The Lion King*). There's issues with supporting different screen sizes and there's just so much more to learn.

Unfortunately, it would take an entire *book* to teach you the entirety of Android app development. So that's a good place to start: buy a book!

But more important is just to play around and try things. Don't set out to make your world-changing app on day one. Instead, focus on making something simple and straightforward and then build on that. Try changing the layout of the text and try adding in more buttons and more rules to make your app actually useful.

Eventually, you'll find there's something you want to do that you can't figure out on your own. Maybe you want a sound to play when someone clicks on your button, for example. This is where the real learning starts. Now all you need to do is search in Google: "How to play sound onClick Android"

You'll find a bunch of complicated answers but eventually someone, probably on Stack Overflow, will break down the answer simply for you. Then what you do is you copy that code and you paste it into your app, making a few changes as you go.

Likewise, try out some of the code samples available through Android studio. See how they work, try changing things and just experiment. Things will go wrong and error messages will come up but for the most part, if you just follow the instructions, it's easy enough to handle. Don't panic! And that's pretty much *how you learn to make apps*. A lot of it boils down to reverse engineering and copying and pasting. Once you have the main program in place, the rest you pick up as you go.

If you want the absolute easiest way to start, then just find some sample code that's close to what you make and change it. No one is going to be able to explain all this to you in a way that makes any sense and if you worry about not grasping everything to begin with, you'll never get anywhere.

So instead, dive in, get your hands dirty and learn on the job. It's complicated and it's frustrating but ultimately it's *highly* rewarding and more than worth the initial effort.

Debug Your App

In this document

1. Enable debugging
2. Start debugging
    1. Attach the debugger to a running app
3. Change the debugger type
4. Use the system log
    1. Write log messages in your code
    2. View the system log
    3. Create custom logcat filter
5. Work with breakpoints
    1. View and configure breakpoints
    2. Debug window frames
6. Inspect variables
    1. Add watchpoints
7. View and change resource value display format

See also

- Create and edit Run/Debug Configurations

Android Studio provides a debugger that allows you to do the following and more:

- Select a device to debug your app on.
- Set breakpoints in your Java, Kotlin, and C/C++ code.
- Examine variables and evaluate expressions at runtime.

This page includes instructions for basic debugger operations. For more documentation, also see the IntelliJ IDEA Debugging docs.

Enable debugging

Before you can begin debugging, you need to prepare as follows:

- **Install LLDB:**

If your project includes C/C++ code, you need to install **LLDB** from the SDK Manager.

- **Enable debugging on your device:**

If you're using the emulator, this is enabled by default. But for a connected device, you need to enable debugging in the device developer options.

- **Run a debuggable build variant:**

You must use a build variant that includes debuggable true in the build configuration. Usually, you can just select the default "debug" variant that's included in every Android Studio project (even though it's not visible in the build.gradle file). But if you define new build types that should be debuggable, you must add `debuggable true` to the build type:

```
android {
   buildTypes {
      customDebugType {
         debuggable true
         ...
```

```
    }
  }
}
```

This property also applies to modules with C/C++ code. (The jniDebuggable property is no longer used.)

Note: If your app depends on a library module that you also want to debug, that library must also be packaged with debuggable true so it retains its debug symbols. To ensure that the debuggable variants of your app project receive the debuggable variant of a library module, be sure that you publish non-default versions of your library.

Start debugging

You can start a debugging session as follows:

1. Set some breakpoints in the app code.
2. In the toolbar, click **Debug**  to display the **Select Deployment Target** window.

If no devices appear in the **Select Deployment Target** window after you click **Debug**, then you need to either connect a device via USB or click **Create new virtual device** to use the Android Emulator.

If, instead of the **Select Deployment Target** window, you see a dialog asking if you want to "switch from Run to Debug," that means your app is already running on the device and it will restart in order to begin debugging. If you'd rather keep the same instance of the app running, click **Cancel Debug** and instead attach the debugger to a running app.

3. Select a deployment target and click **OK**.

Android Studio builds an APK, signs it with a debug key, installs it on your selected device, and runs it. If you add C and C++ code to your project, Android Studio also runs the LLDB debugger in the **Debug** window to debug your native code.

4. If the **Debug** window is not open, select **View** > **Tool Windows** > **Debug** (or click **Debug**  in the tool window bar), and then click the **Debugger** tab, as shown in figure 1.

Figure 1. The Debugger window, showing the current thread and the object tree for a variable

Attach the debugger to a running app

If your app is already running on your device, you can start debugging without restarting your app as follows:

1. Click **Attach debugger to Android process**  .
2. In the **Choose Process** dialog, select the process you want to attach the debugger to.

If you're using an emulator or a rooted device, you can check **Show all processes** to see all processes.

From the **Debugger** drop-down menu, you can select a different debug type. By default, Android Studio uses the **Auto** debug type to select the best debugger option for you, based on whether your project includes Java or C/C++ code.

3. Click **OK**.

The **Debug** window appears.

Note: The Android Studio debugger and garbage collector are loosely integrated. The Android virtual machine guarantees that any object the debugger is aware of is not garbage collected until after the debugger disconnects. This can result in a buildup of objects over time while the debugger is connected. For example, if the debugger sees a running thread, the associated Thread object is not garbage collected until the debugger disconnects, even if the thread has terminated.

Change the debugger type

Because different debugger tools are required to debug Java/Kotlin code and C/C++ code, the Android Studio debugger allows you to select which debugger type to use. By default, Android Studio decides which debugger to use based on which languages it detects in your project (with the **Auto** debugger type). However, you can manually select the debugger in the debug configuration (click **Run > Edit Configurations**) or in the dialog that appears when you click **Run > Attach debugger to Android process**.

The debug types available include the following:

- **Auto**: Select if you want Android Studio to automatically choose the best option for the code you are debugging. For example, if you have any C or C++ code in your project, Android Studio automatically uses the **Dual** debug type. Otherwise, Android Studio uses the **Java** debug type.
- **Java**: Select if you want to debug only code written in Java or Kotlin—the Java debugger ignores any breakpoints or watches you set in your native code.
- **Native**: (Available only with C/C++ code.) Select if you want to use only LLDB to debug your code. When using this debug type, the Java debugger session view is not available. By default, LLDB inspects only your native code and ignores breakpoints in your Java code. If you want to also debug your Java code, you should switch to either the **Auto** or **Dual** debug type.
- **Dual**: (Available only with C/C++ code.) Select if you want to switch between debugging both Java and native code. Android Studio attaches both the Java debugger and LLDB to your app process, one for the Java debugger and one for LLDB, so you can inspect breakpoints in both your Java and native code without restarting your app or changing your debug configuration.

In figure 2, notice the two tabs to the right of the **Debug** window title. Because the app has both Java and C++ code, one tab is for debugging the native code, and the other for debugging Java code, as indicated by **-java**.

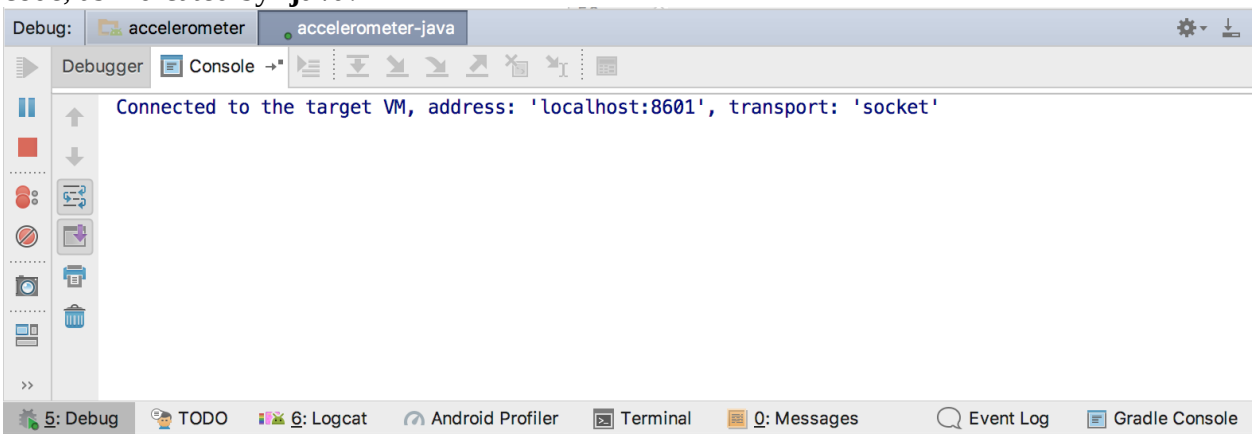


Figure 2. Tab for debugging native code and tab for debugging Java code
Native debugging on 32-bit Windows is not supported on Android Studio 3.0 and higher. If you are using 32-bit Windows and you plan to debug native code.

Note: If you are debugging native code that is optimized by the compiler, you may get the following warning message: This function was compiled with optimizations enabled. Some debugger features may not be available. When using optimization flags, such as -O flags, the compiler makes changes to your compiled code to make it run more efficiently. This can cause the debugger to report unexpected or incorrect information because it's difficult for the debugger to map the optimized compiled code back to the original source code. For this reason, you should disable compiler optimizations while debugging your native code.
Use the system log

The system log shows system messages while you debug your app. These messages include information from apps running on the device. If you want to use the system log to debug your app, make sure your code writes log messages and prints the stack trace for exceptions while your app is in the development phase.
Write log messages in your code
To write log messages in your code, use the Log class. Log messages help you understand the execution flow by collecting the system debug output while you interact with your app. Log messages can tell you what part of your application failed. For more information about logging, see Write and View Logs.
The following example shows how you might add log messages to determine if previous state information is available when your activity starts:

```
import android.util.Log;
...
public class MyActivity extends Activity {
   private static final String TAG = MyActivity.class.getSimpleName();
   ...
   @Override
   public void onCreate(Bundle savedInstanceState) {
      if (savedInstanceState != null) {
         Log.d(TAG, "onCreate() Restoring previous state");
         /* restore state */
      } else {
         Log.d(TAG, "onCreate() No saved state available");
         /* initialize app */
      }
   }
}
```

During development, your code can also catch exceptions and write the stack trace to the system log:

```
void someOtherMethod() {
   try {
      ...
```

```
    } catch (SomeException e) {
        Log.d(TAG, "someOtherMethod()", e);
    }
}
```

Note: Remove debug log messages and stack trace print calls from your code when you are ready to publish your app. You could do this by setting a DEBUG flag and placing debug log messages inside conditional statements.

View the system log

You can view and filter debug and other system messages in the **Logcat** window. For example, you can see messages when garbage collection occurs, or messages that you add to your app with the Log class.

To use logcat, start debugging and select the **Logcat** tab in the bottom toolbar as shown in figure 3.

For a description of logcat and its filtering options,Work with breakpoints

---

Android Studio supports several types of breakpoints that trigger different debugging actions. The most common type is a line breakpoint that pauses the execution of your app at a specified line of code. While paused, you can examine variables, evaluate expressions, then continue execution line by line to determine the causes of runtime errors.

To add a line breakpoint, proceed as follows:

1. Locate the line of code where you want to pause execution, then either click the left gutter along that line of code or place the caret on the line and press Control+F8 (on Mac, Command+F8).
2. If your app is already running, you don't need to update it to add the breakpoint—just click **Attach debugger to Android proccess**  . Otherwise, start debugging by clicking **Debug**  .

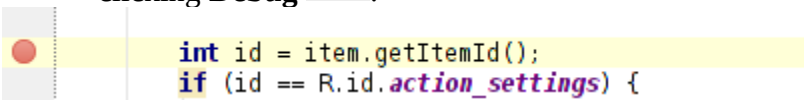


Figure 3. A red dot appears next to the line when you set a breakpoint

When your code execution reaches the breakpoint, Android Studio pauses execution of your app. You can then use the tools in the **Debugger** tab to identify the state of the app:

- To examine the object tree for a variable, expand it in the **Variables** view. If the **Variables** view is not visible, click **Restore Variables View**  .
- To evaluate an expression at the current execution point, click **Evaluate Expression**  .
- To advance to the next line in the code (without entering a method), click **Step Over**  .
- To advance to the first line inside a method call, click **Step Into**  .
- To advance to the next line outside the current method, click **Step Out**  .

- To continue running the app normally, click **Resume Program**  .

If your project uses any native code, by default the **Auto** debug type attaches both the Java debugger and LLDB to your app as two separate processes, so you can switch between inspecting Java and C/C++ breakpoints without restarting your app or changing settings.

Note: For Android Studio to detect breakpoints in your C or C++ code, you need to use a debug type that supports LLDB, such as Auto, Native, or Dual. You can change the debug type Android Studio uses by editing your debug configuration. To learn more about the different debug types, read the section about using other debug types.

M J ANDROID APPLICATIONS

When Android Studio deploys your app to your target device, the Debug window opens with a tab or debug session view for each debugger process, as shown in figure 4.

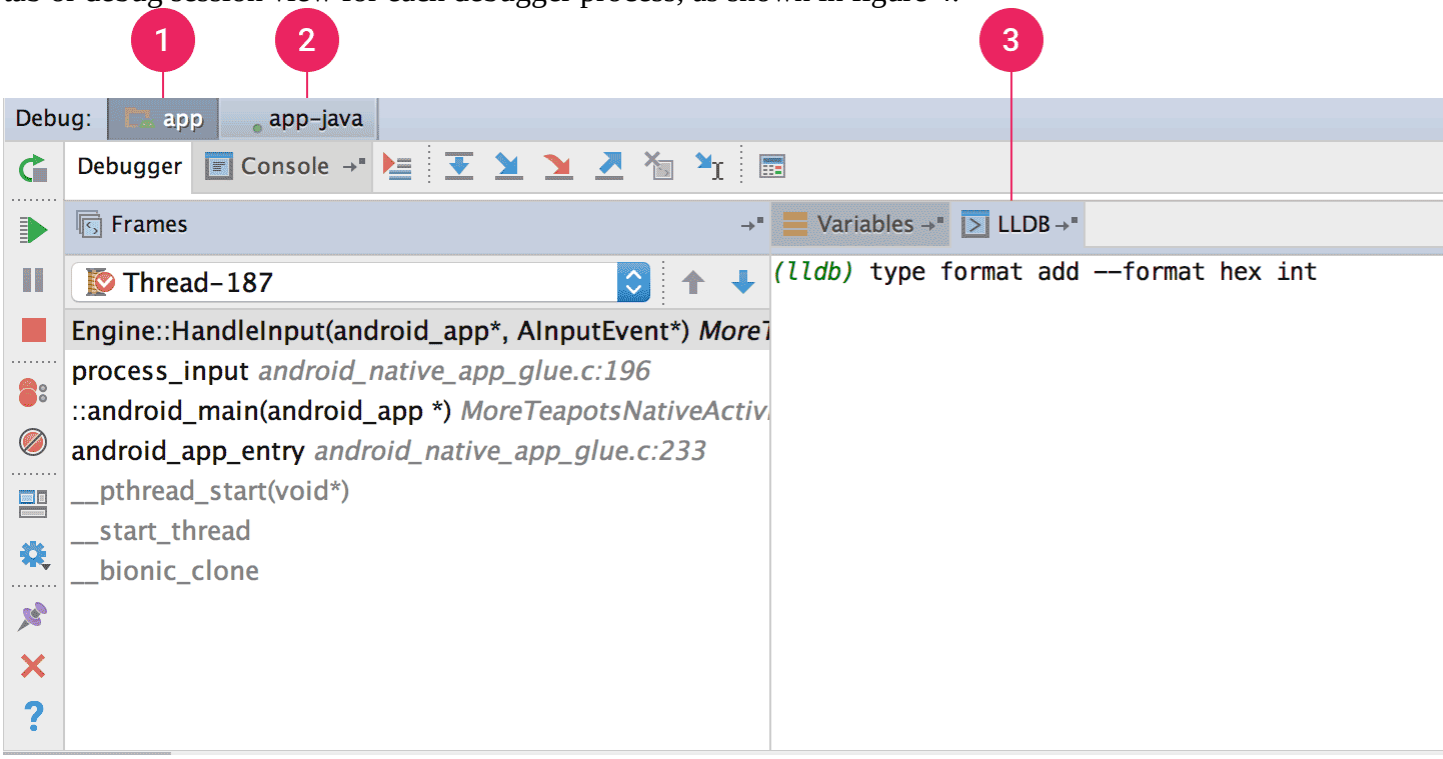


Figure 4. Debugging native code using LLDB

1. Android Studio switches to the <***your-module***> tab when LLDB debugger encounters a breakpoint in your C/C++ code. The **Frames**, **Variables**, and **Watches** panes are also available and work exactly as they would if you were debugging Java code. Although the **Threads** pane is not available in the LLDB session view, you can access your app processes using the drop-down list in the **Frames** pane. You can learn more about these panes in the sections about how to Debug Window Frames and Inspect Variables.

Note: While inspecting a breakpoint in your native code, the Android system suspends the virtual machine that runs your app's Java bytecode. This means that you are unable to interact with the Java debugger or retrieve any state information from your Java debugger session while inspecting a breakpoint in your native code.

2. Android Studio switches to the **<*your-module*>-java** tab when the Java debugger encounters a breakpoint in your Java code.
3. While debugging with LLDB, you can use the **LLDB** terminal in the LLDB session view to pass command line options to LLDB. If you have certain commands that you would like LLDB to execute each time you start debugging your app, either just before or just after the debugger attaches to your app process, you can add those commands to your debug configuration.

While debugging C/C++ code, you can also set special types of breakpoints, called *watchpoints*, that can suspend your app process when your app interacts with a particular block of memory. To learn more, read the section about how to add watch points.

View and configure breakpoints

To view all the breakpoints and configure breakpoint settings, click **View Breakpoints** on the left side of the **Debug** window. The **Breakpoints** window appears, as shown in figure 5.

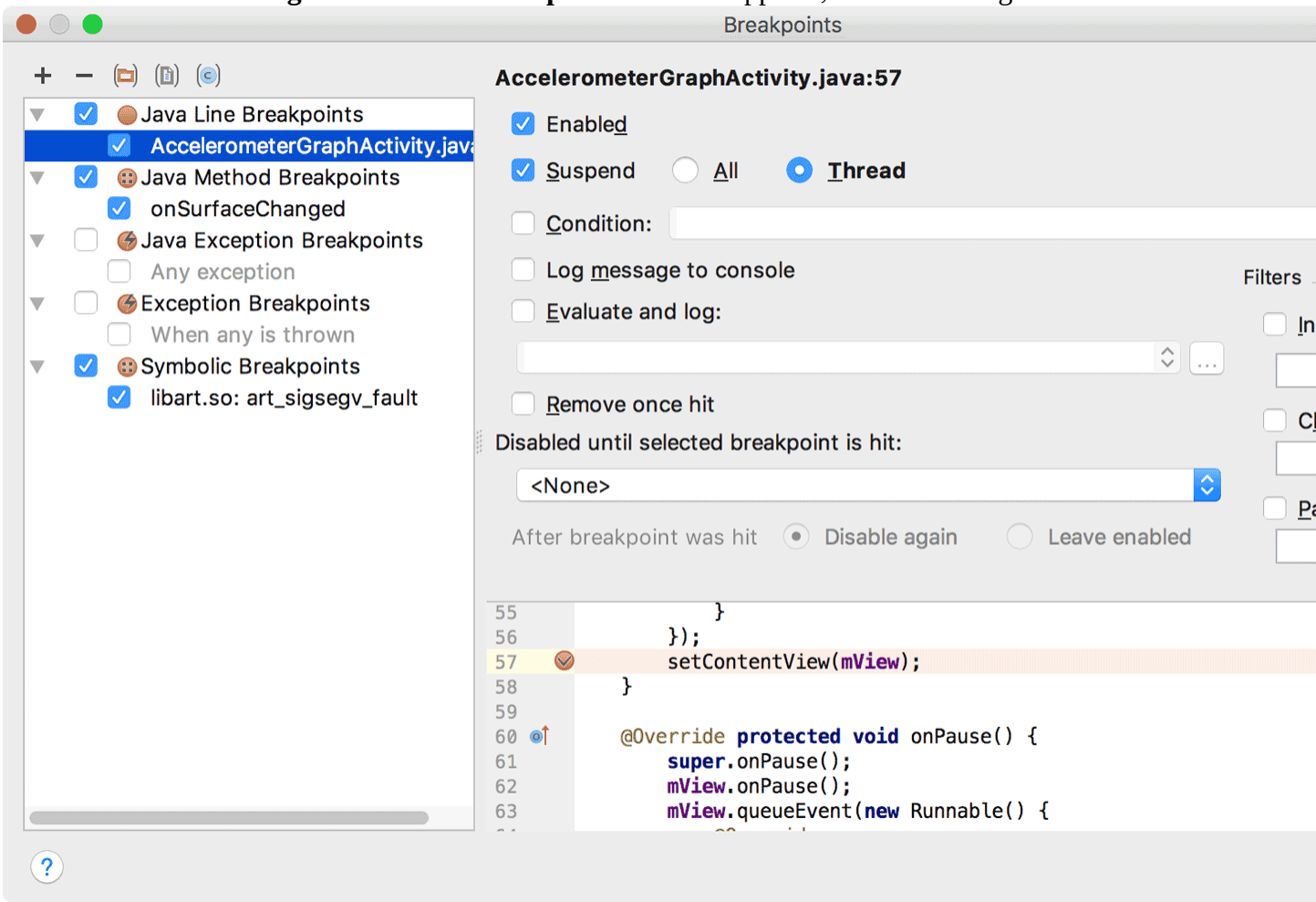


Figure 5. The Breakpoints window lists all the current breakpoints and includes behavior settings for each

The **Breakpoints** window lets you enable or disable each breakpoint from the list on the left. If a breakpoint is disabled, Android Studio does not pause your app when it hits that breakpoint. Select a breakpoint from the list to configure its settings. You can configure a breakpoint to be

disabled at first and have the system enable it after a different breakpoint is hit. You can also configure whether a breakpoint should be disabled after it is hit. To set a breakpoint for any exception, select **Exception Breakpoints** in the list of breakpoints.

Debug window frames

In the **Debugger** window, the **Frames** pane lets you inspect the stack frame that caused the current breakpoint to be hit. This enables you to navigate and examine the stack frame and also inspect the list of threads in your Android app. To select a thread, use the thread selector drop-down and view its stack frame. Clicking the elements in the frame opens the source in the editor. You can also customize the thread presentation and export the stack frame as discussed in the Window Frames guide.

Inspect variables

DEBUGGING ANDROID APPLICATIONS

In the **Debugger** window, the **Variables** pane lets you inspect variables when the system stops your app on a breakpoint and you select a frame from the **Frames** pane. The **Variables** pane also lets you evaluate ad-hoc expressions using static methods and/or variables available within the selected frame.

The **Watches** pane provides similar functionality except that expressions added to the **Watches** pane persist between debugging sessions. You should add watches for variables and fields that you access frequently or that provide state that is helpful for the current debugging session. The **Variables** and **Watches** panes appear as shown in figure 5.

To add a variable or expression to the **Watches** list, follow these steps:

1. Begin debugging.
2. In the **Watches** pane, click **Add**  .
3. In the text box that appears, type the name of the variable or expression you want to watch and then press Enter.

To remove an item from the **Watches** list, select the item and then click **Remove**  .

You can reorder the elements in the **Watches** list by selecting an item and then clicking **Up** 

or **Down**  .

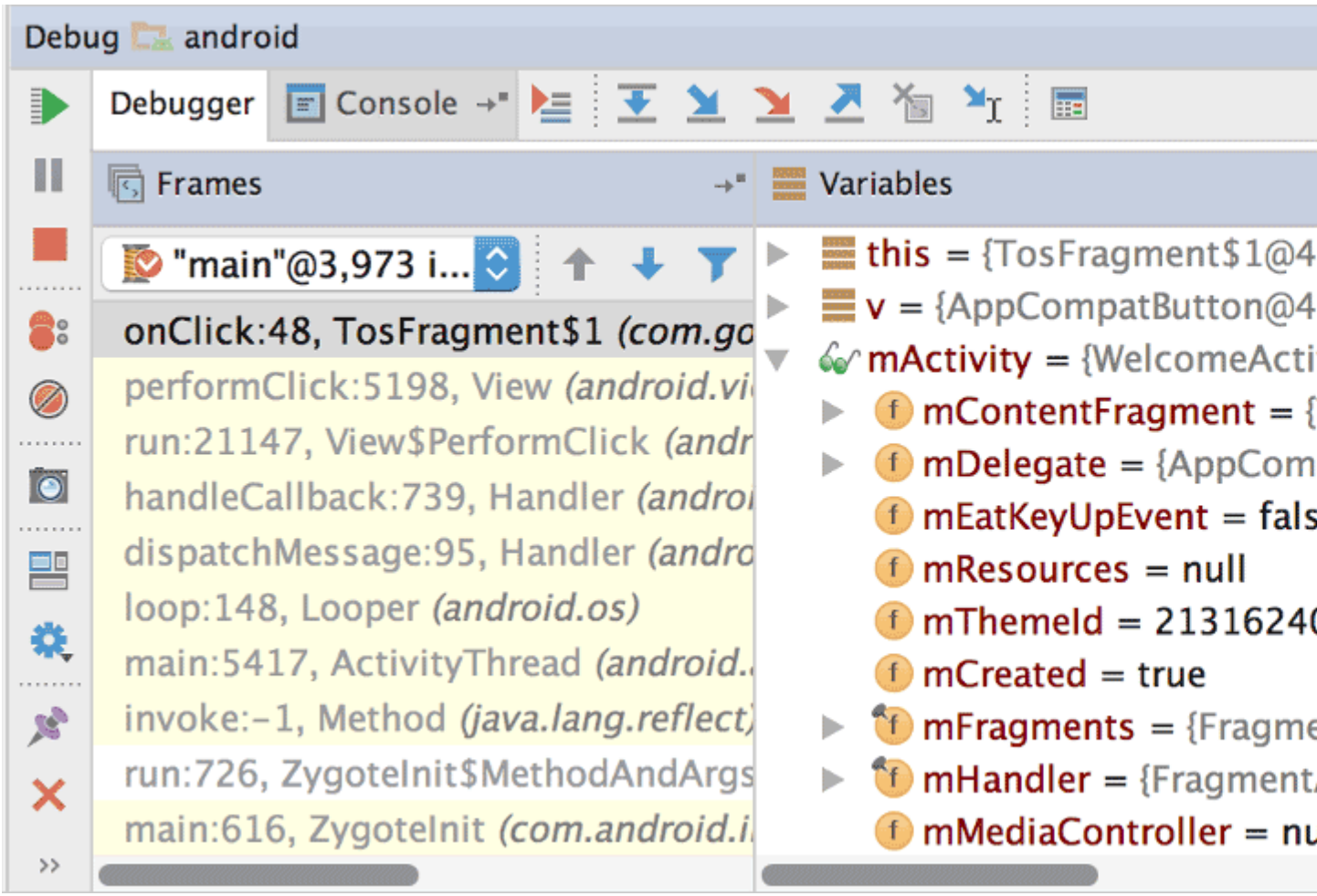


Figure 6. The Variables and Watches panes in the Debugger window

Add watchpoints

While debugging C/C++ code, you can set special types of breakpoints, called *watchpoints*, that can suspend your app process when your app interacts with a particular block of memory. For example, if you set two pointers to a block of memory and assign a watchpoint to it, using either pointer to access that block of memory triggers the watchpoint.

In Android Studio, you can create a watchpoint during runtime by selecting a specific variable, but LLDB assigns the watchpoint to only the block of memory the system allocates to that variable, not the variable itself. This is different from adding a variable to the **Watches** pane, which enables you to observe the value of a variable but doesn't allow you to suspend your app process when the system reads or changes its value in memory.

Note: When your app process exits a function and the system deallocates its local variables from memory, you need to reassign any watchpoints you created for those variables.

To set a watchpoint, you must meet the following requirements:

- Your target physical device or emulator uses an x86 or x86_64 CPU. If your device uses an ARM CPU, then you must align the boundary of your variable's address in memory to either 4 bytes for 32-bit processors, or 8 bytes for 64-bit processors. You can align a variable in your native code by specifying __attribute__((aligned(***num_bytes***))) in the variable deceleration, as shown below:

// For a 64-bit ARM processor
int my_counter __attribute__((aligned(8)));

- You have assigned three or fewer watchpoints already. Android Studio only supports up to four watchpoints on x86 or x86_64 target devices. Other devices may support fewer watchpoints.

If you meet the requirements above, you can add a watchpoint as follows:

1. While your app is suspended on a breakpoint, navigate to the **Variables** pane in your LLDB session view.
2. Right-click on a variable that occupies the block of memory you want to track and select **Add Watchpoint**. A dialog to configure your watchpoint appears, as shown in figure 7.

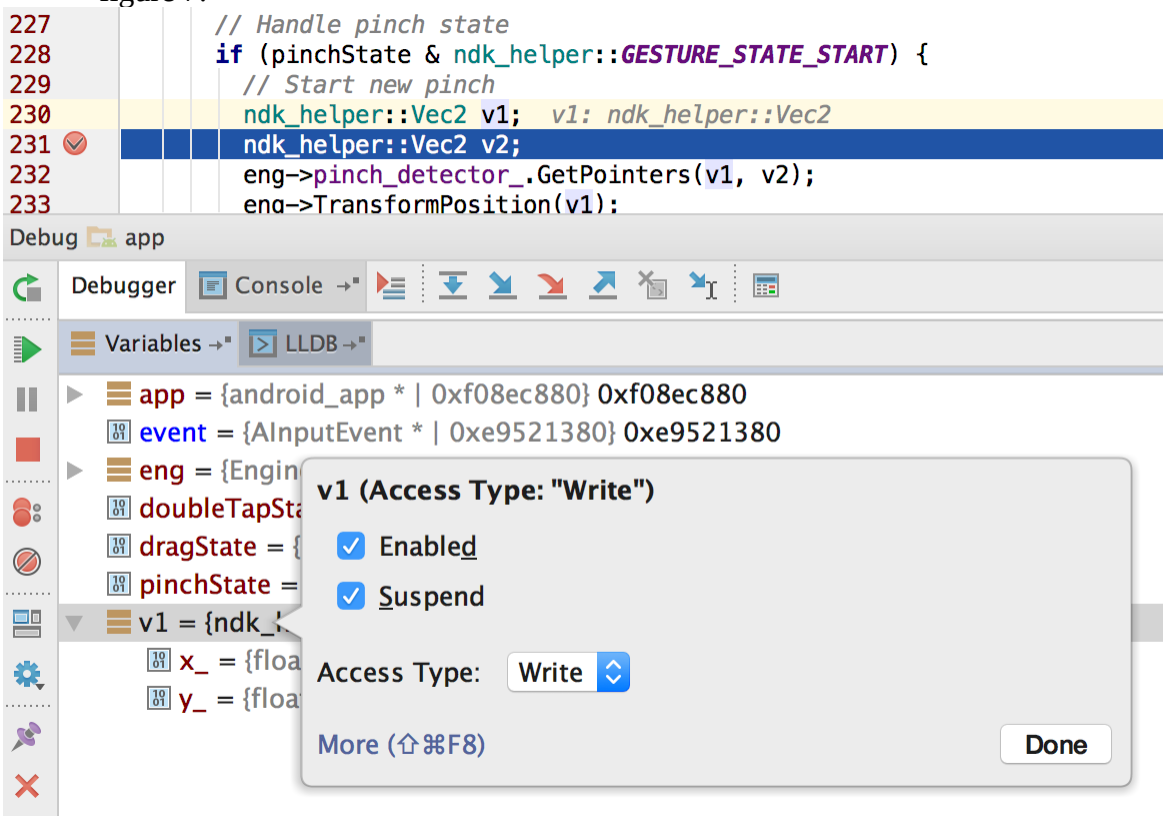


Figure 7. Adding a watchpoint to a variable in memory

3. Configure your watchpoint with the following options:
   - **Enabled:** You can deselect this option if you want to tell Android Studio to ignore the watchpoint for the time being. Android Studio still saves your watchpoint so you can access it later in your debug session.
   - **Suspend:** By default, the Android system suspends your app process when it accesses a block of memory you assign to a watchpoint. You can deselect this option if you don't want this behavior—this reveals additional options you can use to customize behavior when the system interacts with your watchpoint: **Log message to console** and **Remove [the watchpoint] when hit**.

- **Access Type:** Select whether your app should trigger your watchpoint when it tries to **Read** or **Write** to the block of memory the system allocates to the variable. To trigger your watchpoint on either a read or write, select **Any**.

4. Click **Done**.

To view all your watchpoints and configure watchpoint settings, click **View Breakpoints** on the left side of the **Debug** window. The **Breakpoints** dialog appears, as shown in figure 8.

After you add your watchpoint, click **Resume Program** on the left side of the **Debug** window to resume your app process. By default, if your app tries to access a block of memory that you have set a watchpoint to, the Android system suspends your app process and a watchpoint icon appears next to the line of code that your app executed last, as shown in figure 9.

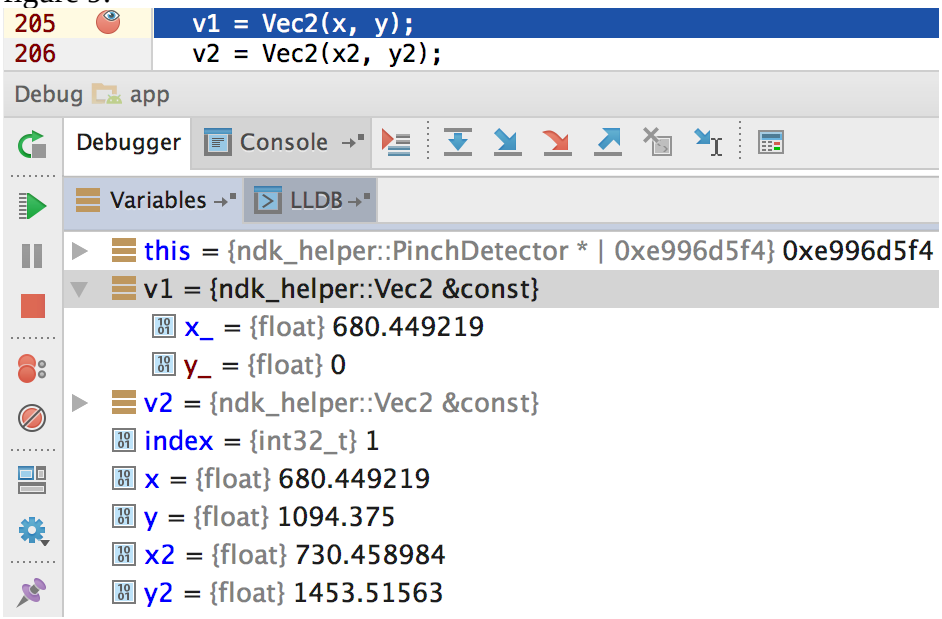


Figure 9. Android Studio indicates the line of code that your app executes just before triggering a watchpoint

View and change resource value display format

In debug mode, you can view resource values and select a different display format for variables in your Java code. With the **Variables** tab displayed and a frame selected, do the following:

1. In the **Variables** list, right-click anywhere on a resource line to display the drop-down list.
2. In the drop-down list, select **View as** and select the format you want to use.

The available formats depend on the data type of the resource you selected. You might see any one or more of the following options:

- **Class:** Display the class definition.
- **toString:** Display string format.
- **Object:** Display the object (an instance of a class) definition.
- **Array:** Display in an array format.

- **Timestamp:** Display date and time as follows: yyyy-mm-dd hh:mm:ss.
- **Auto:** Android Studio chooses the best format based on the data type.
- **Binary:** Display a binary value using zeroes and ones.
- **MeasureSpec:** The value passed from the parent to the selected child.
- **Hex:** Display as a hexadecimal value.
- **Primitive:** Display as a numeric value using a primitive data type.
- **Integer:** Display a numeric value of type Integer.

UNIT V - POSSIBLE QUESTIONS

PART – B

(Each question carries six marks)

1. Explain how to create an android development environment.
2. Write notes on Android and social networking.
3. Explain the components of an Android application and its activity life cycle.
4. Discuss the building and running of Micro Jobs applications in Android.
5. Explain the MJ Android code with suitable example.
6. Write about the primary tools used by Android developers with suitable example
7. Explain the three perspectives for developing android projects with suitable example.
8. Explain about the i) Debugger ii)Logcat tools in android development environment.
9. Explain the following procedures in creating Android development environment
    a. i)Installing JDK and Eclipse ii)Installing Android plug-in
10. Write in detail about the Micro Jobs Android code with suitable example.

PART – C
Compulsory Question
(Each question carries ten marks)

1. What should be the characteristics of Mobile Computing Devices?
2. Describe the security aspects in mobile computing.
3. Explain the functionality of Mobile Computing
4. How would you broadly classify the mobile computing applications?
5. Who are all the stakeholders of wireless network?
6. What is WiMax? How it is differ from WiFi?
7. Describe 3G networks.
8. Explain the Process of GSM Call Routing.
9. Explain the Process of Interactive Voice Response.
10. Briefly describe the Eclipse IDE for the development of android application.

MOBILE COMPUTING (18CAP402)

UNIT- V

S.No	Question	Opt1	Opt2	Opt3	Opt4	Opt5	Opt6	Answer
1	Android is licensed under_____open source licensing license	Gnu's GPL	Apache/MIT	OSS	Sourceforge			Apache/MIT
2	Android is not actually owned by Google. Who owns the Android platform?	Oracle Techno	Dalvik	Open Hand	Android is owned by Google			Open Handset
3	As an Android programmer,_____version of Android should you use as your minimum development target?	Versions 1.6 or 2.0	Versions 1.0 or 1.1	Versions 1.2 or 1.3	Versions 2.3 or 3.0			Versions 1.6 or 2.0
4	What was the first phone released that ran the Android OS?	T-Mobile G1	Google gPhone	Motorola Droid	HTC Hero			T-Mobile G1
5	What year was the Open Handset Alliance announced?	2005	2006	2007	2008			2007
6	Android tries hard to_____low-level components, such as the software stack, with interfaces so that vendor-specific code can be managed easily.	confound	abstract	modularize	compound			abstract
7	What part of the Android platform is open source?	low-level Linux	high-level Linux	packages	resource files			low-level Linux
8	When did Google purchase Android?	2007	2005	2008	2010			2005
9	Which one is not a nickname of a version of Andriod?	cupcake	Gingerbread	Honeycomb	Muffin			Muffin
10	Which Android version had the greatest share of the market as of January 2011?	1.1	1.5	2.3	3.4			1.5
11	Which piece of code used in Android is not open source?	Keypad driver	WiFi driver	Audio driver	Power management			WiFi driver

12	Android is based on Linux for the following reason.	Security	Portability	Networking	All of these			All of these
13	What operating system is used as the base of the Android stack?	Linux	Windows	Java	XML			Linux
14	What year was development on the Dalvik virtual machine started?	2003	2005	2007	2006			2005
15	When developing for the Android OS, Java byte code is compiled into _____	Java source code	Dalvik application code	Dalvik byte code	C source code			Dalvik byte code
16	What does the .apk extension stand for?	Application Package	Application Program Kit	Android Proprietary Kit	Android Package			Application Package
17	Which of these are not one of the three main components of the APK?	Dalvik Executable	Resources	Native Libraries	Webkit			Webkit
18	_____is the name of the program that converts Java byte code into Dalvik byte code?	Android Interpretive Compiler (AIC)	Dalvik Converter	Dex compiler	Mobile Interpretive Compiler (MIC)			Dex compiler
19	Android Applications must be signed _____	After they are installed	Before they are installed	Never	two weeks of installatio			Before they are installed
20	Which of the following are not a component of an APK file?	Config files	DLLs	binary files	Dalvik executable			Dalvik executable
21	An activity can be thought of as corresponding to _____	A Java project	A Java class	A method call	An object field			A Java class
22	The_____file specifies the layout of your screen.	Layout file	Manifest file	Strings XML	R file			Layout file

23	What is the driving force behind an Android application and that ultimately gets converted into a Dalvik executable?	Java source code.	R-file.	the emulator.	the SDK			ava source code.
24	While developing Android applications, developers can test their apps on _____	Modulator	Physical Android phone	Demodulator	Simulator			Physical Android phone
25	The XML file that contains all the text that your application uses _____	stack.xml	text.xml	strings.xml	string.java			strings.xml
26	Which of the following is the most "resource hungry" part of dealing with Activities on Android	Closing an app	Suspending an app	Opening a new app	Restoring the most recent app			Opening a new app
27	What runs in the background and doesn't have any UI components?	Intents	Content Providers	Services	Applications			Services
28	When an activity doesn't exist in memory it is in _____	Starting state	Running state	Loading state	Inexistent state.			Starting state
29	Which of the following is NOT a state in the lifecycle of a service?	Starting	Running	Destroyed	Paused			Paused
30	Broadcast receivers are Android's implementation of a system-wide publish/subscribe mechanism, or more precisely,_____design pattern	Observer	Facade	Mediator	Command			Observer
31	_____built-in database is Android shipped with	SQLite	Apache	MySQL	Oracle			SQLite
32	Creating a UI (User Interface) in Android requires careful use of _____	Java and SQL	XML and Java	XML and C++	Dreamweaver			XML and Java
33	Status data will be exposed to the rest of the Android system via _____	Intents	A content provider	Network receivers	Altering permissions			A content provider
34	Once installed on a device, each Android application lives in_____	device memor	external me	n security sa	None of the above			security sandbox

35	Parent class of Activity includes _____	Object	Context	ActivityGr	ContextThemeWrapper	ContextThemeWrapper
36	_____is the Parent class of Service	Object	Context	ContextW	ContextThemeWrapper	ContextWrapper
37	_____are the indirect Direct subclasses of Services	RecognitionS	RemoteVie	SpellChec	InputMethodService	InputMethodService
38	_____component is not activated by an Intent	Activity	Services	ContentPro	BroadcastReceiver	ContentProvider
39	Using_____contentProvider would be activated	Using Intent	Using SQLit	Using Con	None of these	Using ContentResolver
40	_____is the important device characteristics that you should consider as you design and develop your application	Screen size an	Density	resolution	processor speed	Screen size and density
41	_____are the screen sizes in Android	small	normal	extremely	abnormal	normal
42	_____are the screen densities in Android	low density	medium den	extremely	high density	medium density
43	You can shut down an activity by calling its_____method	onDestory()	finishActivit	finish()	None of these	finish()
44	_____is NOT related to fragment class	DialogFragme	ListFragmen	Preference	CursorFragment	CursorFragment
45	In android_____ways to start services	Started	Bound	Started and	waited	Started and Bound
46	If your service is private to your own application and runs in the same process as the client (which is common), you should create your interface by extending the_____class	Messenger	Binder	AIDL	Bounder	Binder
47	If you need your interface to work across different processes, you can create an interface for the service with a _____	Binder	Messenger	AIDL	messenger or AIDL	messenger or AIDL

48	Layouts in android includes _____	Frame Layout	Sequence L	Absolute	Single Layout			Frame Layout
49	Dialog classes in android includes _____	Wndows Dialo	Color Dialo	TimePicke	FontDialog			TimePicker Dialog