

ASSIGNMENT: 4

UNIT 1: INTRODUCTION, TRANSMISSION FUNDAMENTALS:

1. Given a channel with an intended capacity of 50 Mbps, the bandwidth of the Channel is 5 MHz. What signal-to-noise ratio is required to achieve this capacity?

- By using Shannon's capacity formula,

$$C = B \log_2 (1 + S/N)$$

$$50 * 10^6 = 5 * 10^6 * \log_2 (1 + S/N)$$

$$\log_2 (1 + S/N) = 10$$

$$S/N = 1023$$

$$SNR_{dB} = 10 \log_{10}(1023) = 30dB$$

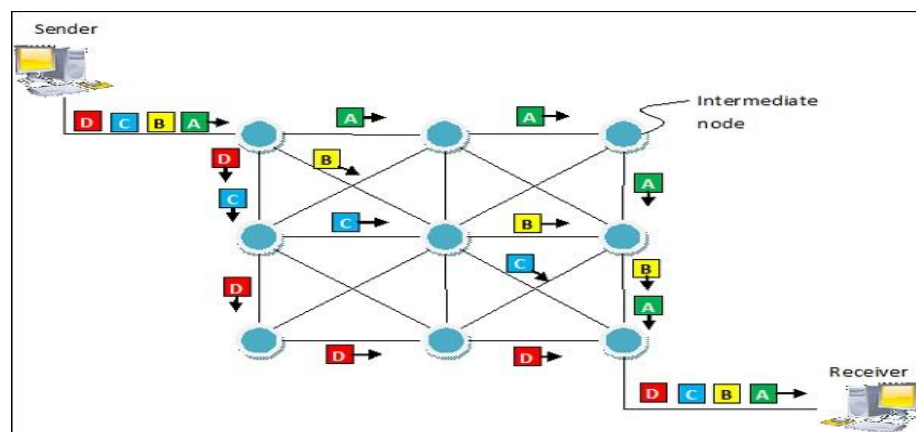
2. Explain packet switching and circuit switching with it advantages and disadvantages.

⇒ **Packet switching:**

Packet switching is a connectionless network switching technique. Here, the message is divided and grouped into a number of units called packets that are individually routed from the source to the destination. There is no need to establish a dedicated circuit for communication.

Process:

- Each packet in a packet switching technique has two parts: a header and a payload. The header contains the addressing information of the packet and is used by the intermediate routers to direct it towards its destination. The payload carries the actual data.
- A packet is transmitted as soon as it is available in a node, based upon its header information. The packets of a message are not routed via the same path. So, the packets in the message arrive in the destination out of order. It is the responsibility of the destination to reorder the packets in order to retrieve the original message.
- The process is diagrammatically represented in the following figure. Here the message comprises of four packets, A, B, C and D, which may follow different routes from the sender to the receiver.



- Packet switching is a connectionless network switching technique. Here, the message is divided and grouped into a number of units called packets that are individually routed from the source to the destination. There is no need to establish a dedicated circuit for communication.

Advantages:

- Delay in delivery of packets is less, since packets are sent as soon as they are available.
- Switching devices don't require massive storage, since they don't have to store the entire messages before forwarding them to the next node.
- Data delivery can continue even if some parts of the network face link failure. Packets can be routed via other paths.
- It allows simultaneous usage of the same channel by multiple users.
- It ensures better bandwidth usage as a number of packets from multiple sources can be transferred via the same link.

Disadvantages:

- They are unsuitable for applications that cannot afford delays in communication like high quality voice calls.
- Packet switching high installation costs.
- They require complex protocols for delivery.
- Network problems may introduce errors in packets, delay in delivery of packets or loss of packets. If not properly handled, this may lead to loss of critical information.

Circuit Switching:

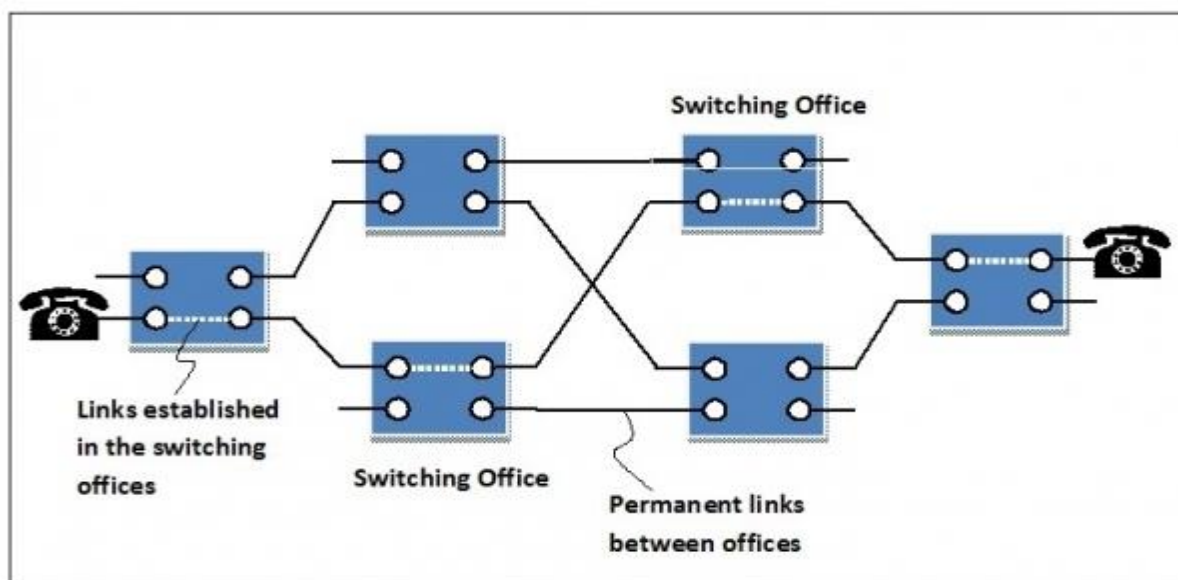
Circuit switching is a connection-oriented network switching technique. Here, a dedicated route is established between the source and the destination and the entire message is transferred through it.

Phases of Circuit Switch Connection:

- **Circuit Establishment :** In this phase, a dedicated circuit is established from the source to the destination through a number of intermediate switching centres. The sender and receiver transmits communication signals to request and acknowledge establishment of circuits.
- **Data Transfer :** Once the circuit has been established, data and voice are transferred from the source to the destination. The dedicated connection remains as long as the end parties communicate.
- **Circuit Disconnection :** When data transfer is complete, the connection is relinquished. The disconnection is initiated by any one of the user. Disconnection involves removal of all intermediate links from the sender to the receiver.

Diagrammatic Representation of Circuit Switching in Telephone:

The following diagram represents circuit established between two telephones connected by circuit switched connection. The blue boxes represent the switching offices and their connection with other switching offices. The black lines connecting the switching offices represent the permanent link between the offices. When a connection is requested, links are established within the switching offices as denoted by white dotted lines, in a manner so that a dedicated circuit is established between the communicating parties. The links remain as long as communication continues.



Advantages:

- It is suitable for long continuous transmission, since a continuous transmission route is established, that remains throughout the conversation.
- The dedicated path ensures a steady data rate of communication.
- No intermediate delays are found once the circuit is established. So, they are suitable for real time communication of both voice and data transmission.

Disadvantages:

- Circuit switching establishes a dedicated connection between the end parties. This dedicated connection cannot be used for transmitting any other data, even if the data load is very low.
- Bandwidth requirement is high even in cases of low data volume.
- There is underutilization of system resources. Once resources are allocated to a particular connection, they cannot be used for other connections.
- Time required to establish connection may be high.

3. Explain Nyquist theorem? Find the relationship among the following terms Channel Capacity (C), Bandwidth (B) and Signal-to-Noise Ratio (SNR).

- A theorem, developed by H. Nyquist, which states that an analog signal waveform may be uniquely reconstructed, without error, from samples taken at equal time intervals. The sampling rate must be equal to, or greater than, twice the highest frequency component in the analog signal. Nyquist's work states that an analog signal waveform can be converted into digital by sampling the analog signal at equal time intervals. The Nyquist Theorem, also known as the sampling theorem, is a principle that engineers follow in the digitization of analog signals.

- Channel Capacity (C)

The maximum rate at which data can be transmitted over a given communication path, or channel, under given conditions, is referred to as the channel capacity

- Bandwidth (B)

The bandwidth of the transmitted signal as constrained by the transmitter and the nature of the transmission medium, expressed in cycles per second, or hertz

- Signal-to-Noise Ratio (SNR)

In analog and digital communications, signal-to-noise ratio, often written S/N or SNR, is a measure of signal strength relative to background noise. The ratio is usually measured in decibels (dB) using a signal-to-noise ratio formula.

- Signal to noise ratio formula

The signal to noise ratio is the ratio between the wanted signal and the unwanted background noise.

$$\text{SNR} = P_{\text{signal}}/P_{\text{noise}}$$

It is more usual to see a signal to noise ratio expressed in a logarithmic basis using decibels:

$$\text{SNR}_{\text{dB}} = 10 \log_{10} (P_{\text{signal}}/P_{\text{noise}})$$

If all levels are expressed in decibels, then the formula can be simplified to:

$$\text{SNR}_{\text{dB}} = P_{\text{signal dB}} - P_{\text{noise dB}}$$

The power levels may be expressed in levels such as dBm (decibels relative to a milliwatt, or to some other standard by which the levels can be compared.

UNIT 2: CELLULAR WIRELESS NETWORKS

1. Write a short note on selective repeat ARQ.

- Selective repeat protocol, also called Selective Repeat ARQ (Automatic Repeat reQuest), is a data link layer protocol that uses sliding window method for reliable delivery of data frames. Here, only the erroneous or lost frames are retransmitted, while the good frames are received and buffered.
- It uses two windows of equal size: a sending window that stores the frames to be sent and a receiving window that stores the frames received by the receiver. The size is half the maximum sequence number of the frame. For example, if the sequence number is from 0 – 15, the window size will be 8.
- Working Principle:
 - Selective Repeat protocol provides for sending multiple frames depending upon the availability of frames in the sending window, even if it does not receive acknowledgement for any frame in the interim. The maximum number of frames that can be sent depends upon the size of the sending window.
 - The receiver records the sequence number of the earliest incorrect or un-received frame. It then fills the receiving window with the subsequent frames that it has received. It sends the sequence number of the missing frame along with every acknowledgement frame.
 - The sender continues to send frames that are in its sending window. Once, it has sent all the frames in the window, it retransmits the frame whose sequence number is given by the acknowledgements. It then continues sending the other frames.

2. What is Multi-path propagation and fading?

Multi-path propagation:

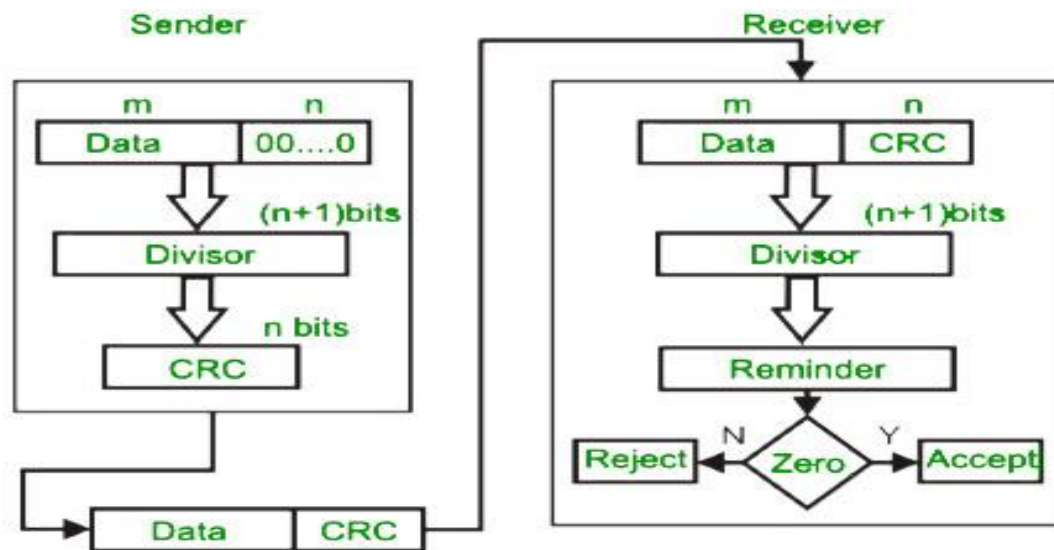
- Multipath propagation is a fact of life in any terrestrial radio scenario. While the direct or line of sight path is often the main wanted signal, a radio receiver will receive different versions of the same signal that have travelled from the transmitter via many different paths.
- The vast number of different signal paths arises from the fact that signals are reflections from buildings, mountains or other reflective surfaces including water, etc. that may be adjacent to the main path. Additionally other effects such as ionospheric reflections give rise to multipath propagation as does tropospheric ducting.
- The antennas used for transmission and reception have an effect on the number of paths that the signal can take. Non-directive antennas will radiate the signal in all directions, whereas directive ones will focus the power in one direction reducing the strength of reflected signals away from the main beam.
- The multipath propagation resulting from the variety of signal paths that may exist between the transmitter and receiver can give rise to interference in a variety of ways including distortion of the signal, loss of data and multipath fading.
- At other times, the variety of signal paths arising from the multipath propagation can be used to advantage. Schemes such as MIMO use multipath propagation to increase the capacity of the channels they use or seek to improve the signal to noise ratio.

Multipath fading:

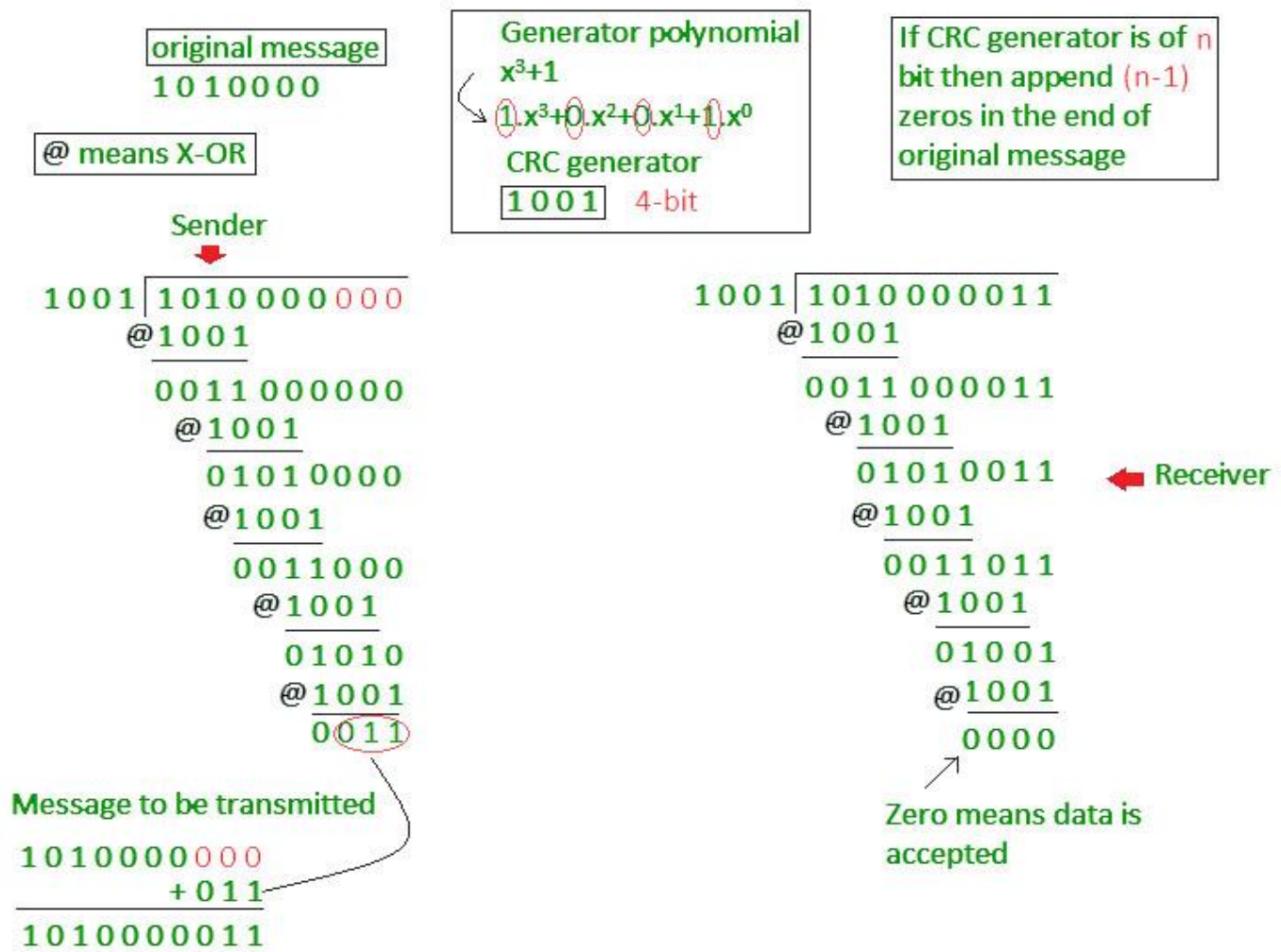
- Signals are received in a terrestrial environment, i.e. where reflections are present and signals arrive at the receiver from the transmitter via a variety of paths. The overall signal received is the sum of all the signals appearing at the antenna. Sometimes these will be in phase with the main signal and will add to it, increasing its strength. At other times they will interfere with each other. This will result in the overall signal strength being reduced.

3. Describe any one error detection technique with suitable example.**Cyclic redundancy check (CRC):**

- Unlike checksum scheme, which is based on addition, CRC is based on binary division.
- In CRC, a sequence of redundant bits, called cyclic redundancy check bits, are appended to the end of data unit so that the resulting data unit becomes exactly divisible by a second, predetermined binary number.
- At the destination, the incoming data unit is divided by the same number. If at this step there is no remainder, the data unit is assumed to be correct and is therefore accepted.
- A remainder indicates that the data unit has been damaged in transit and therefore must be rejected.



➤ Example:



4. What is the bandwidth efficiency for FSK, ASK, PSK and QPSK for a bit error rate of 10^{-7} on a channel with an SNR of 12 dB?

$$(E_b/N_0)_{dB} = 12 \text{ dB} - (R/B_T)_{dB}$$

For FSK and ASK,

$$(E_b/N_0)_{dB} = 14.2 \text{ dB}$$

$$(R/B_T)_{dB} = -2.2 \text{ dB}$$

$$R/B_T = 0.6$$

For PSK,

$$(E_b/N_0)_{dB} = 11.2 \text{ dB}$$

$$(R/B_T)_{dB} = 0.8 \text{ dB}$$

$$R/B_T = 1.2$$

The result of QPSK must be taken into account that the baud rate $D=R/2$.

$$\text{Thus } R/B_T = 2.4$$

5. What is fading? Differentiate

I. Fast and slow fading

Fast Fading	Slow Fading
It varies quickly with the frequency	It does not vary quickly with the frequency.
High Doppler Spread	Low Doppler Spread
Coherence Time < Symbol Period	Low Doppler Spread
Channel impulse response changes rapidly within the symbol duration.	Impulse response changes much slower than the transmitted signal.
It occurs Occurs if $T_s > T_c$, $B_s < B_D$	It occurs if $T_s \ll T_c$, $B_s \gg B_D$

II. Flat and selective fading.

Flat Fading	Selective Fading
BW of signal is less than the BW of channel	BW of signal is greater than BW of channel.
delay spread is less than symbol period	delay spread is greater than symbol period.

range of frequencies in a frequency spectrum are equally faded	in one part of frequency spectrum is faded more than the other part of frequency spectrum
--	---

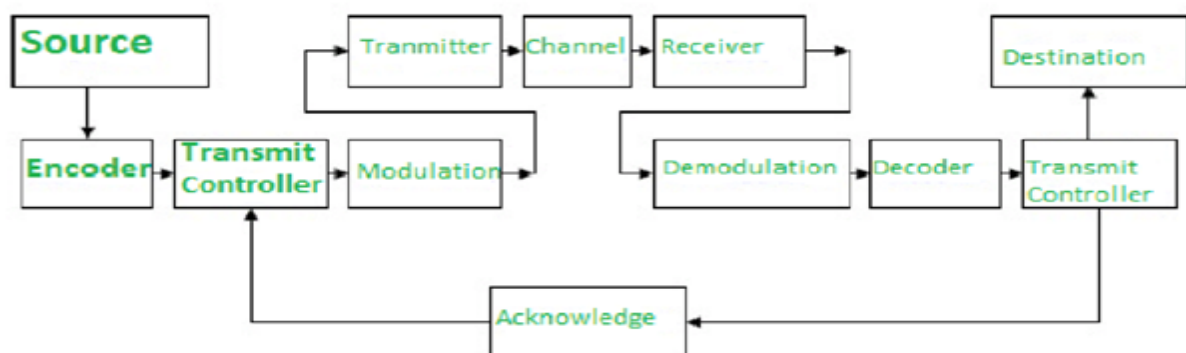
6. What is the need of ARQ? Explain Automatic Repeat Request (ARQ) in details?

Automatic Repeat Request (ARQ)

ARQ stands for Automatic Repeat Request also known as Automatic Repeat Query. ARQ is an error-control strategy used in a two-way communication system. It is a group of error-control protocols to achieve reliable data transmission over an unreliable source or service. These protocols reside in Transport Layer and Data Link Layer of the OSI(Open System Interconnection) model . These protocols are responsible for automatic retransmission of packets that are found to be corrupted or lost during the transmission process.

➤ Working Principle of ARQ:

- The main function of these protocols is, the sender receives an acknowledgment from the receiver end implying that the frame or packet is received correctly before a timeout occurs, timeout is a specific time period within which the acknowledgment has to be sent by the receiver to the sender.
- If a timeout occurs: the sender does not receive the acknowledgment before the specified time, it is implied that the frame or packet has been corrupt or lost during the transmission.
- Accordingly, the sender retransmits the packet and these protocols ensure that this process is repeated until the correct packet is transmitted.



➤ Types of ARQ Protocols

There are three ARQ protocols in the data link layer.

- Stop – and – Wait ARQ – Stop – and – wait ARQ provides unidirectional data transmission with flow control and error control mechanisms, appropriate for noisy channels. The sender keeps a copy of the sent frame. It then waits for a finite time to receive a positive acknowledgement from receiver. If the timer expires, the frame is retransmitted. If a positive acknowledgement is received then the next frame is sent.

- Go – Back – N ARQ – Go – Back – N ARQ provides for sending multiple frames before receiving the acknowledgement for the first frame. It uses the concept of sliding window, and so is also called sliding window protocol. The frames are sequentially numbered and a finite number of frames are sent. If the acknowledgement of a frame is not received within the time period, all frames starting from that frame are retransmitted.
- Selective Repeat ARQ – This protocol also provides for sending multiple frames before receiving the acknowledgement for the first frame. However, here only the erroneous or lost frames are retransmitted, while the good frames are received and buffered.

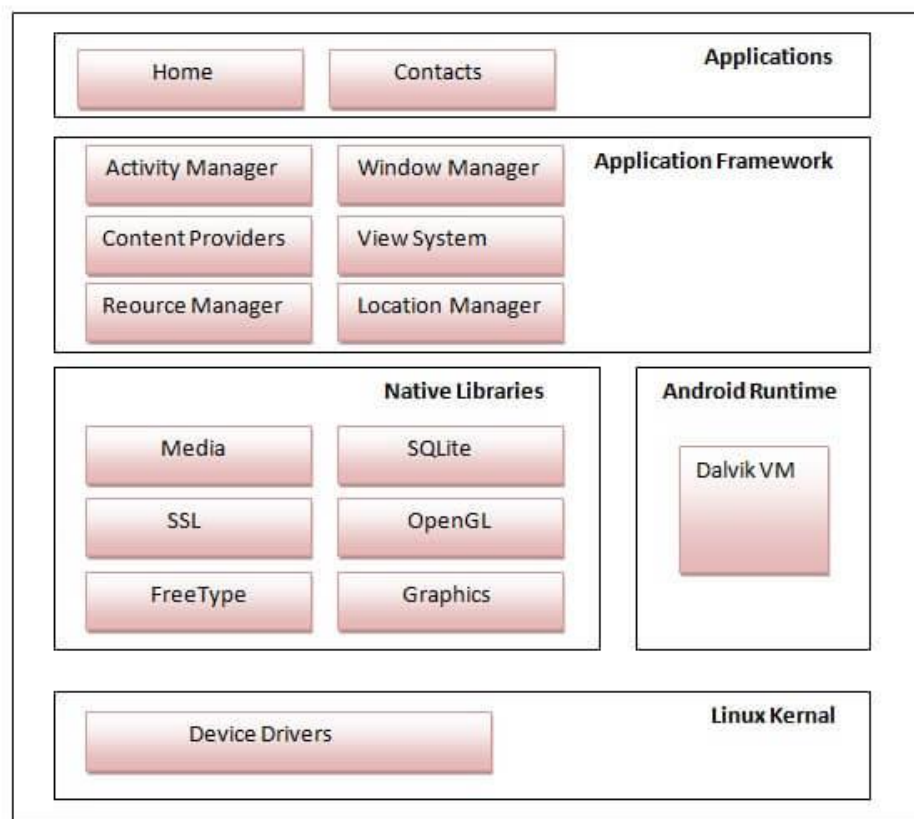
UNIT 6: ANDROID APIs

1. Draw & Explain Android application framework with their components.

Android architecture or Android software stack is categorized into five parts:

1. linux kernel
2. native libraries (middleware),
3. Android Runtime
4. Application Framework
5. Applications

Let's see the android architecture first.



1) Linux kernel:

- It is the heart of android architecture that exists at the root of android architecture. Linux kernel is responsible for device drivers, power management, memory management, device management and resource access.

2) Native Libraries:

- On the top of linux kernel, there are Native libraries such as WebKit, OpenGL, FreeType, SQLite, Media, C runtime library (libc) etc.
- The WebKit library is responsible for browser support, SQLite is for database, FreeType for font support, Media for playing and recording audio and video formats.

3) Android Runtime:

- In android runtime, there are core libraries and DVM (Dalvik Virtual Machine) which is responsible to run android application. DVM is like JVM but it is optimized for mobile devices. It consumes less memory and provides fast performance.

4) Android Framework:

- On the top of Native libraries and android runtime, there is android framework. Android framework includes Android API's such as UI (User Interface), telephony, resources, locations, Content Providers (data) and package managers. It provides a lot of classes and interfaces for android application development.

5) Applications:

- On the top of android framework, there are applications. All applications such as home, contact, settings, games, browsers are using android framework that uses android runtime and libraries. Android runtime and native libraries are using linux kernel.

2. Explain Android EditText and TextView control with an example.

Textview:

➤ Definition

A TextView displays text to the user and optionally allows them to edit it. A TextView is a complete text editor however, the basic class is configured to not allow editing.

Main Attributes of Text view

- **android:gravity:** Specifies how to align the text with the view's x- and/or y-axis when the text is smaller than the view.
- **android:hint:** Hint text to display when the text is empty.
- **android:id:** This is the ID which uniquely identifies the control.
- **android:text:** Text to display.
- **android:textColor:** Text color. May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".
- **android:textSize:** Size of the text. Recommended dimension type for text is "sp" for scaled pixels (example: 15sp).
- **Example of TextView**

```
<RelativeLayout
xmlns:android=http://schemas.android.c
```

```

        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:context=".MainActivity">
        <TextView
        android:id="@+id/text_id"
        android:layout_width="300dp"
        android:layout_height="200dp"
        android:text="hello_world"
        android:textColor="@android:color/holo_
        android:textSize="50dp"/>
    </RelativeLayout>

```

EditText

➤ Definition

An EditText is an overlay over TextView that configures itself to be editable. It is the predefined subclass of TextView that includes rich editing capabilities.

Edit text Attributes:

Following are the important attributes related to EditText control. You can check Android official documentation for complete list of attributes and related methods which you can use to change these attributes are run time.

- **android:text:** If set, specifies that this TextView has a textual input method and automatically corrects some common spelling errors.
- **android:background:** This is drawable to use as the background.
- **android:id:** This supplies an identifier name for this view.
- **android:visibility:** This controls the initial visibility of the view
- **android:visibility:** This controls the initial visibility of the view
- **Example of EditText:**

```

<RelativeLayout
    xmlns:android="http://schemas.android.c
    xmlns:tools="http://schemas.android.com
    android:layout_width="match_parent
    android:layout_height="match_parent"
    tools:context=" . MainActivity">
    EditText
    android:id="@+id/edittext"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/button"
    android:layout_below="@+id/textView1"
    android:layout_marginTop="61dp"
    android:ems="10"
    android:text="@string/enter_text"
    android:inputType="text"/>
</RelativeLayout>

```

3. Define Android layout. Explain various Android layouts.

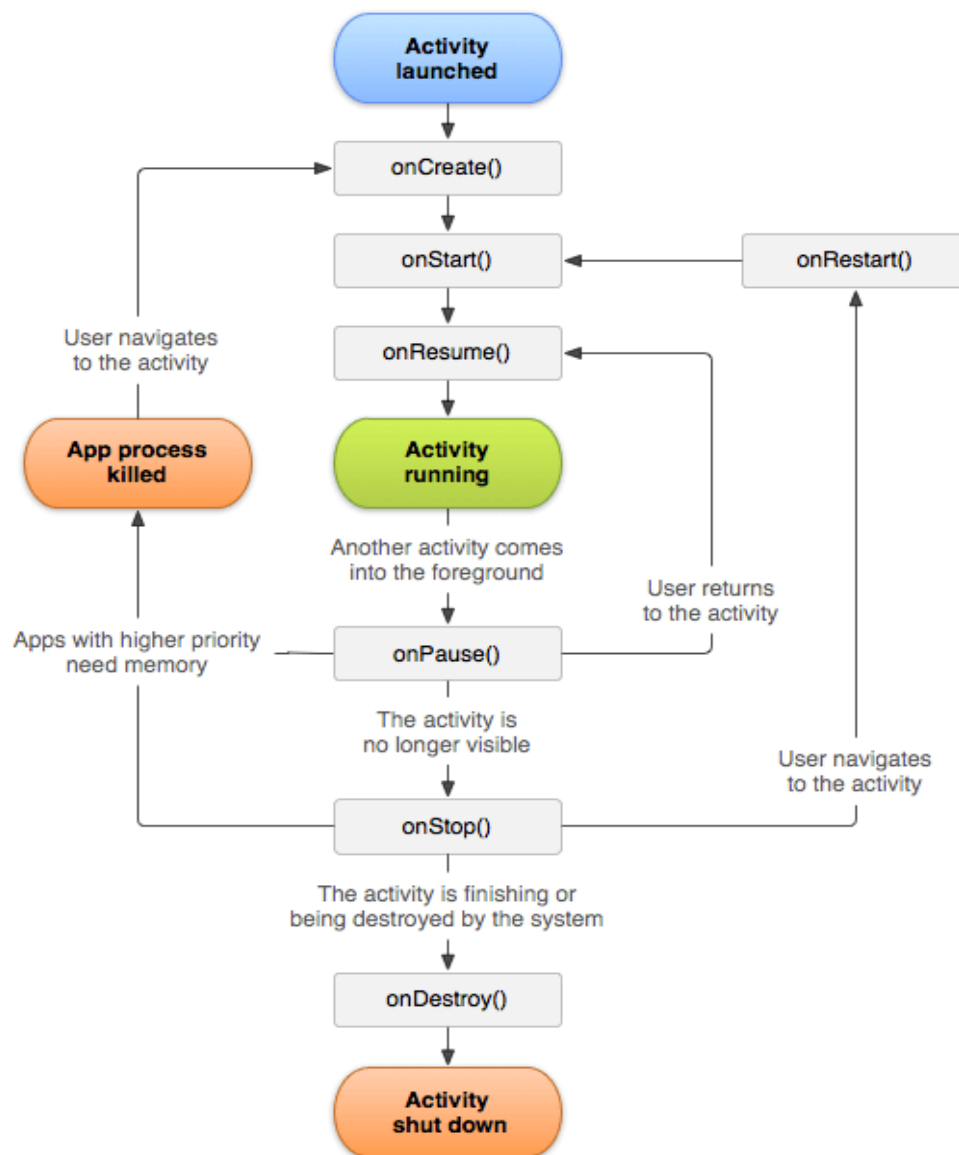
- Android Layout is used to define the user interface which holds the UI controls or widgets that will appear on the screen of an android application or activity.

Types of Android Layout

- Android Linear Layout: LinearLayout is a ViewGroup subclass, used to provide child View elements one by one either in a particular direction either horizontally or vertically based on the orientation property.
- Android Relative Layout: RelativeLayout is a ViewGroup subclass, used to specify the position of child View elements relative to each other like (A to the right of B) or relative to the parent (fix to the top of parent).
- Android Constraint Layout: ConstraintLayout is a ViewGroup subclass, used to specify the position of a layout constraints for every child View relative to other views present. A ConstraintLayout is similar to a RelativeLayout, but having more power.
- Android Frame Layout: FrameLayout is a ViewGroup subclass, used to specify the position of View elements it contains on the top of each other to display only single View inside the FrameLayout.
- Android Table Layout: TableLayout is a ViewGroup subclass, used to display the child View elements in rows and columns.
- Android Web View: WebView is a browser which is used to display the web pages in our activity layout.
- Android List View: ListView is a ViewGroup, used to display scrollable list of items in single column.
- Android Grid View: GridView is a ViewGroup which is used to display scrollable list of items in grid View of rows and columns.

4. Discuss Activity life cycle in Android.

- In Android, an activity is referred to one screen in an application. It is very similar to a single-window of any desktop application. An Android app consists of one or more screens or activities.
- Each activity goes through various stages or a lifecycle and is managed by activity stacks. So when a new activity starts, the previous one always remains below it. There are four stages of an activity.
 - I. If an activity is in the foreground of the screen i.e at the top of the stack, then it is said to be active or running. This is usually the activity that the user is currently interacting with.
 - II. If an activity has lost focus and a non-full-sized or transparent activity has focused on top of your activity. In such a case either another activity has a higher position in multi-window mode or the activity itself is not focusable in current window mode. Such activity is completely alive..
 - III. If an activity is completely hidden by another activity, it is stopped or hidden. It still retains all the information, and as its window is hidden thus it will often be killed by the system when memory is needed elsewhere.
 - IV. The system can destroy the activity from memory by either asking it to finish or simply killing its process. When it is displayed again to the user, it must be completely restarted and restored to its previous state.
- For each stage, android provides us with a set of 7 methods which have their own significance for each stage in the life cycle. The image shows a path of migration whenever an app switches from one state to another.



➤ Android Activity Lifecycle methods

Let's see the 7 lifecycle methods of android activity.

Method	Description
onCreate	called when activity is first created.
onStart	called when activity is becoming visible to the user.
onResume	called when activity will start interacting with the user.
onPause	called when activity is not visible to the user.
onStop	called when activity is no longer visible to the user.
onRestart	called after your activity is stopped, prior to start.
onDestroy	called before the activity is destroyed.

➤ Android Activity Lifecycle Example

//File: MainActivity.java

```
package example.javatpoint.com.activitylifecycle;

import android.app.Activity;
import android.os.Bundle;
import android.util.Log;

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Log.d("lifecycle", "onCreate invoked");
    }
    @Override
    protected void onStart() {
        super.onStart();
        Log.d("lifecycle", "onStart invoked");
    }
    @Override
    protected void onResume() {
        super.onResume();
        Log.d("lifecycle", "onResume invoked");
    }
    @Override
    protected void onPause() {
        super.onPause();
        Log.d("lifecycle", "onPause invoked");
    }
    @Override
    protected void onStop() {
        super.onStop();
        Log.d("lifecycle", "onStop invoked");
    }
    @Override
    protected void onRestart() {
        super.onRestart();
        Log.d("lifecycle", "onRestart invoked");
    }
    @Override
    protected void onDestroy() {
        super.onDestroy();
        Log.d("lifecycle", "onDestroy invoked");
    }
}
```

➤ Output:



5. Explain types of Intents.

- There are two types of intents in android: implicit and explicit.

1) Implicit Intent:

- Implicit Intent doesn't specify the component. In such case, intent provides information of available components provided by the system that is to be invoked.
- For example, you may write the following code to view the webpage.

```
Intent intent=new Intent(Intent.ACTION_VIEW);  
intent.setData(Uri.parse("http://www.javatpoint.com"));  
startActivity(intent);
```

2) Explicit Intent:

- Explicit Intent specifies the component. In such case, intent provides the external class to be invoked.
- For example,

```
Intent i = new Intent(getApplicationContext(), ActivityTwo.class);  
startActivity(i);
```

To get the full code of explicit intent, visit the next page.

6. Discuss the manifest file with example.

- Every project in Android includes a manifest file, which is AndroidManifest.xml, stored in the root directory of its project hierarchy. The manifest file is an important part of our app because it defines the structure and metadata of our application, its components, and its requirements.

- This file includes nodes for each of the Activities, Services, Content Providers and Broadcast Receiver that make the application and using Intent Filters and Permissions, determines how they co-ordinate with each other and other applications.
- The manifest file also specifies the application metadata, which includes its icon, version number, themes etc. and additional top- level nodes can specify any required permissions, unit tests and define hardware, screen, or platform requirements.
- The manifest comprises of a root manifest tag with a package attribute set to the project's package. It should also include an `xmlns:android` attribute that will supply several system attributes used within the file.
- We use `versionCode` attribute is used to define the current application version in the form of an integer that increments itself with the iteration of version due to updation. Also, `versionName` attribute is used to specify a public version that will be displayed to the users.
- The elements used in the above xml file are described below.

- `<manifest>`

`manifest` is the root element of the `AndroidManifest.xml` file. It has `package` attribute that describes the package name of the activity class.

- `<application>`

`application` is the subelement of the manifest. It includes the namespace declaration. This element contains several subelements that declares the application component such as activity etc.

The commonly used attributes are of this element are icon, label, theme etc.

`android:icon` represents the icon for all the android application components.

`android:label` works as the default label for all the application components.

`android:theme` represents a common theme for all the android activities.

- `<activity>`

`activity` is the subelement of `application` and represents an activity that must be defined in the `AndroidManifest.xml` file. It has many attributes such as label, name, theme, `launchMode` etc.

`android:label` represents a label i.e. displayed on the screen.

`android:name` represents a name for the activity class. It is required attribute.

- `<intent-filter>`

`intent-filter` is the sub-element of activity that describes the type of intent to which activity, service or broadcast receiver can respond to.

- <action>

It adds an action for the intent-filter. The intent-filter must have at least one action element.

- <category>

It adds a category name to an intent-filter.

➤ Example:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.javatpoint.hello"
    android:versionCode="1"
    android:versionName="1.0" >
    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="15" />
    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".MainActivity"
            android:label="@string/title_activity_main" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```