iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 2 **Issue:** I **Month of publication:** January 2014

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

Vol. 2 Issue I, January 2014
ISSN: 2321-9653

INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE AND ENGINEERING TECHNOLOGY (IJRASET)

Wi-Fi Direct in Android Using Peer to Peer Communication

Neshat Karim Shaukat
Faculty of Management Studies and Information Technology

Jamia Hamdard (Hamdard University)

New Delhi India

Abstract: Open Handset Alliance (OHA) a confederation of 50 Telecoms, mobile hardware, and software companies, headed by Google, were found on 5th of November, 2007 and unveiled a new platform that is called Android. At present become an important player in the mobile arena. This research paper evaluates and introduces the application "Wi-Fi Direct in Android using Peer to peer communication". The purpose of this research is to design and implement a android application that uses Wi-Fi direct protocol in P2P (Peer-to-Peer) as a means of communication between Smartphone's with Android at no internet cost.

Keywords: Android; Wi-Fi Direct; JAVA; P2P; JXTA; API

INTRODUCTION

In starting time Peer to peer communication is used only for file sharing but at present p2p is used in many type of application for example VOIP, Skype, Facebook, gtalk by using the internet bandwidth.

But after introducing the WI-FI Direct Protocol [1] presented the opportunity for supporting P2P in such a way that can be effectively relieve the overload on some mobile networks. Both Internet Service Providers (ISPs) and consumers could benefit from this. Wi-Fi Direct will allow a mobile to connect directly with another mobile in its range that is also running the protocol, with no hubs or routers are involved. Data rates are expected to be over 250 Mbps with a coverage range of with 200 meters [1]. It can be connect the just a few meters but also across home. It means that making a Wi-Fi Direct group connection will be convenient, even when devices aren't in immediate proximity to one another [1].The main objective of this paper is how peer to peer communication can be effectively used in Android by using the Wi-Fi direct protocol.

Why WI-Fi Direct Protocol.

Wi-Fi Direct is a standard that permits Wi-Fi devices to communicate with each other without any requirement for wireless access points (hot spots). The Wi-Fi Alliance has announced a new path for W-Fi enabled devices or Smartphone's to connect to one another, even in the absence of a Wi-Fi base station. This new protocol, called as "Wi-Fi Direct," that will allow any device that implements the standard to connect directly to another device to send and receive data.

A side benefit of Wi-Fi Direct is that it can operate at higher speeds and greater distances than Bluetooth, though Bluetooth typically uses far less power than Wi-Fi. Furthermore, the standard is aimed at enterprise use as well as consumer use, with the inclusion of enterprise management features and WPA2 security.

The Wi-Fi Direct protocol uses physical P2P communication in that it allows Wi-Fi devices in range to talk to each other without the need of intermediate wireless access points or routers [2].

Vol. 2 Issue I, January 2014
ISSN: 2321-9653

Wi-Fi performance, range, and demonstrated interoperability delivering the excellent user experience of any peer-to-peer technology available [1].

Wi-Fi Direct has the good range, speed and security features to easily handle even the most demanding applications and content [1].

Why Choose Android Operating System

Android is a mobile operating system that is based on a modified version of Linux. It was originally developed by a startup of the same name, Android, Inc. In 2005, as part of its strategy to enter the mobile space, Google purchased Android and took over its development work (as well as its development team).

The main advantage of choosing android is the unified approach of application development.

Developers need only develop for Android, and their applications should be able to run on several different devices, as long as the devices are powered using Android.

From android application developer point of view, the advantages are listed below:

Open :

Android allows us to access core mobile device functionality through standard API calls. It's really amazing for developers that just two line of code can do everything as sending the mobile SMS[3].

Geolocation:

Android OS supports geographical location detection through various different implementations, including GPS (Global Positioning System) and cell-tower triangulation, and additionally Internet services that use techniques such as IP sniffing to determine location. Also combine information from the web with original data on the phone such as contacts, geographic location to create new experience for users[4].This provides the way to building new innovative application.

All applications are equal :

Android OS doesn't categories the basic and third parties applications even the dialer or home page can be replaced.

Easy and fast deployment :

To develop the rich and useful applications with android is provide the SDK which includes the true device emulator and advanced debugging tools [4].

General features:

Storage — Uses SQLite, a lightweight relational database, for data storage.

Connectivity — Supports GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth (includes A2DP and AVRCP), WiFi, LTE, and WiMAX.

Messaging — Supports both SMS and MMS.

Web browser — Based on the open-source WebKit, together with Chrome's V8 JavaScript engine.

Media support — Includes support for the following media: H.263, H.264 (in 3GP or MP4 container), MPEG-4 SP, AMR, AMR-WB (in 3GP container), AAC, HE-AAC (in MP4 or 3GP container), MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF, and BMP

Hardware support — Accelerometer Sensor, Camera, Digital Compass, Proximity Sensor, and GPS

Multi-touch — Supports multi-touch screens

Multi-tasking — Supports multi-tasking applications

Flash support — Android 2.3 supports Flash 10.1.

Histology & Architechture of Android

All of the android application runs in its own process with its own instances of the Dalvik virtual machine[5].The android application are built in JAVA ,so we need to have the basic understanding of java. Dalvik is not exactly a java machine since it is unable to read the java code but it includes own byte code known as "dex" and so the executable files compacted using Dalvik holds the file type name '.dex'. [4] which is optimized for minimal memory

footprint.

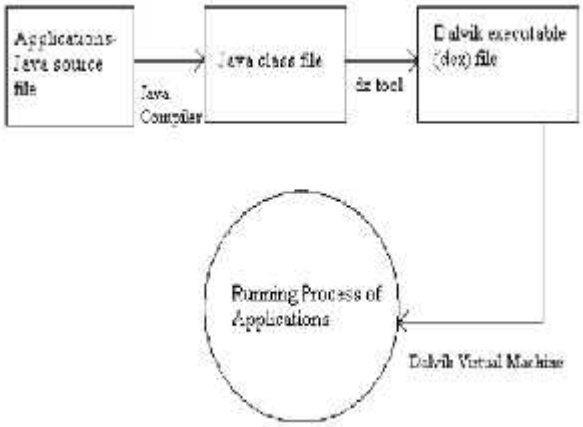


FIG : HISTOLOGY OF ANDROID.

The Android OS is roughly divided into five sections in four main layers:

Linux kernel — This is the kernel.Android relies linux version 2.6 for core system services such as security, memory management, process management, network stack, and driver model. It also act as an abstraction layer between the hardware and rest of the software stack [5] . This layer contains all the low level device drivers for the various hardware components of an Android device [4].

Libraries — These contain all the code that provides the main features of an Android OS.it is the C/C++ libraries used by android system and through application framework it can be used by developers.

Android runtime — At the same layer as the libraries, the Android runtime provides a set of core libraries that enable developers to write Android apps using the Java programming language.

Application framework — Exposes the various capabilities of the Android OS to application developers so that they can make use of them in their applications [6].

Applications — At this top layer, you will find applications that ship with the Android device (such as Phone, Contacts, Browser, etc.), as well as applications that you download and install from the Android Market. Any applications that you write are located at this layer [6].

CURRENT STATE OF THE ART FOR P2P

Mobile and wireless systems can especially benefit from the self-organizing characteristics of P2P systems, e.g., with respect to robustness, however the heterogeneity and dynamicity of those systems also poses severe challenges to P2P concepts.

In the present mobile world, although thousands of different P2P applications have been installed in the fixed internet, utilizing hundreds of peer-to-peer protocols, there have not been many peer-to-peer applications or protocols available in the mobile domain. Some projects have utilizes the peer-to-peer protocols on mobile platforms, such as SymTorrent, Symella but none of these projects have really considered the special needs and constraints of mobile domain. Also, most of this work has focused on one service, i.e., on P2P file-sharing [3].

Now we present the state of the art in P2P system by discussing three prominent P2P system.

P2P system was designed to use in stationary host but now a days it available with mobile internet services or broadband Many individuals use their mobile broadband through Data card or USB sticks connected to their laptops and PCs. These users mostly using the P2P applications which are framed for wired connections, often expecting comparable performance with the same application running on a computer connected to a wired network [2]. This may be highly inefficient since classical P2P protocols are not optimized for mobile networks, which clearly have different parameters and capabilities. Additionally, this approach does not take advantage in any way of the added features and capabilities that mobile devices can have compared to stationary hosts, such as mobility, ubiquity, and sensing [10].

P2P protocols where mobile devices are considered weaker nodes and thus have to rely on stationary hosts. Mobile devices generally have queries to and from other peers mediated by a stationary node, which behaves as a proxy. In present scenario, most of the mobile devices have more better computational strength than PCs had ten years ago and are suitable for properly designed P2P technologies [10].

3. Ad-hoc Network based and application –Oriented P2P systems.Mobile Peer to peer system in Ad-Hoc network is a great novel application system. Personal Area Network and Ad hoc Network is combined to produced a concept of mobile ad hoc information system which is dynamic distributed with high speed self motivated network which

consist of independent mobile devices [12]. Tools, applications, and services implement application specific functionality, which tends to certain Peer to peer applications running on top of the underlying P2P infrastructure [11]. In Addition, since till this time here is no any widely established middleware framework (except JXTA) is available, every new or day to day developing P2P application needs to be developed from scratch.

OBJECTIVE AND GOALS

The main objective of this project is to test of Android platform and to present how peer-to-peer based services can be efficiently realized in next-generation protocol named as WI-FI Direct Protocol by reusing their existing protocols as much as possible. It is also evaluated what kind of special requirements mobile environment poses for peer-to-peer applications and consider those requirements in our application design.

Contribution

Our contribution in this project is to comparing the available Peer to peer application and we present a new way of implementation using the Wi-Fi Direct protocol in Android platform provided built-in PEER TO PEER messaging capabilities, file and content sharing etc.

IMPLEMENTATION

Android P2P using Wi-Fi direct

The Android SDK provides a set of APIs for retrieving information about the Wi-Fi networks available to the device and Wi-Fi network connection details.

The Wi-Fi Direct APIs consist of the following main parts:

Methods that allow us to discover, request, and connect to peers are defined in the WifiP2pManager class.

Listeners that allow us to be notified of the success or failure of WifiP2pManager method calls. When calling WifiP2pManager methods, each method can receive a specific listener passed in as a parameter [13].

Intents that notify you of specific events detected by the Wi-Fi Direct framework, such as a dropped connection or a newly discovered peer [13].

The primary class we need to work with is WifiP2pManager, which we can acquire by calling getSystemService(WIFI_P2P_SERVICE).
The WifiP2pManager includes APIs that allow us to:

Initialize our application for P2P connections by calling initialize()

Discover nearby devices by calling discoverPeers()

Start a P2P connection by calling connect()

In order to use the Wi-Fi P2P APIs, our app must request the following user permissions in the AndroidManifest.xml file:

ACCESS_WIFI_STATE This permission is required for accessing the information regarding Wi-Fi networks.

CHANGE_WIFI_STATE This permission is required to request any information about Wi-Fi Device.

Steps to create the application as follow:

1. Request permission to use the Wi-Fi hardware on the device [13].

```
<uses-sdk android:minSdkVersion="14" />

<uses-permission
android:name="android.permission.ACCESS_WIFI_STATE"
/>

"android.permission.CHANGE_WIFI_STATE"

"android.permission.CHANGE_NETWORK_STATE"

"android.permission.INTERNET"

"android.permission.ACCESS_NETWORK_STATE"
```

2. Check to see if Wi-Fi Direct is on and supported.

Now the application required is an instance of the WifiManager object. It is a system service, so the getSystemService() method works.

```
WifiManager wifi = (WifiManager)

getSystemService(Context.WIFI_SERVICE);

@Override
public void onReceive(Context context, Intent intent) {
String action = intent.getAction();

if
(WifiP2pManager.WIFI_P2P_STATE_CHANGED_ACTION
.equals(action)) { int state =
intent.getIntExtra(WifiP2pManager.EXTRA_WIFI_STATE, -
1);
if (state ==
WifiP2pManager.WIFI_P2P_STATE_ENABLED) {
// Wifi Direct is enabled
} else {
// Wi-Fi Direct is not enabled
}
}
...
}
```

3. In your activity's onCreate() method, obtain an instance of WifiP2pManager and register application with the Wi-Fi Direct framework by calling initialize(). This method returns a WifiP2pManager.Channel, which is used to connect application to the Wi-Fi Direct framework. we should also create an instance of our Application with the WifiP2pManager and WifiP2pManager.Channel objects along with a reference to activity. This allows our application to notify activity of interesting events and update it accordingly. It also lets you manipulate the device's Wi-Fi state if necessary

4. Create an intent filter and add the same intents that your Application checks for.

5.Register the Application in the onResume() method of our activity and unregister it in the onPause() method of activity.

6. Discover nearby devices by calling discoverPeers().

PEER TO PEER JXTA

JXTA is open source Peer to peer middleware platform developed by Sun Microsystem in 2001. JXTA platform is a continue series of classes and methods for managing and transmitting application and control data between JXTA compatible peer platforms. These core services are used to create peer-to-peer applications [14]. The goal of JXTA is not too associated with Java everywhere, whereas peer-to-peer networking everywhere. The Java application of JXTA should be completely compatible with any other version, whether written in C, Pearl, or other popular language [14].

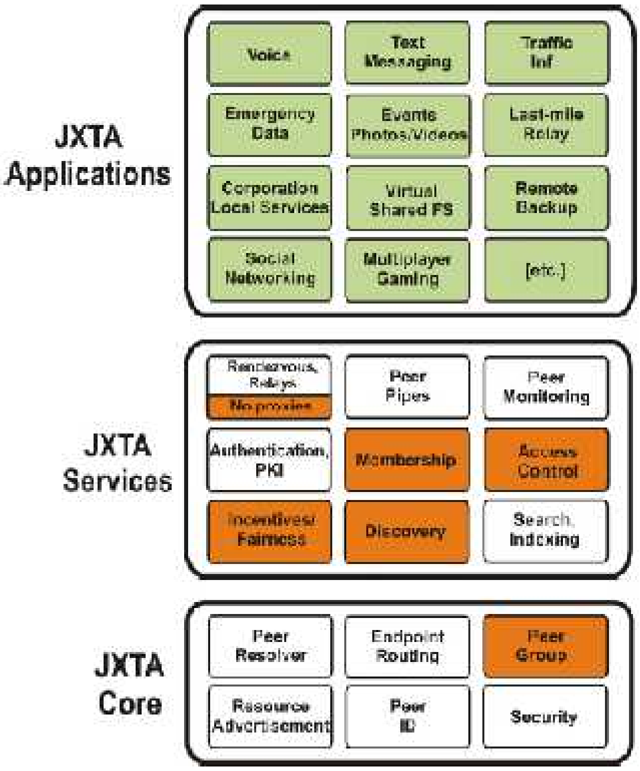


Fig :The three layers in our mobile-optimized JXTA [2].

JXTA provides the many peer-to-peer communication services and protocols for mobile users and developers to used in development of different types f P2P application.

Through Peer Resolver Protocol (PRP) user can have the access to send a search query to another peer [3] .

Peer Discovery Protocol used to discover other Peers.

Endpoint Router Protocol (ERP) used for routing information.

Peer Information Protocol (PIP) to know the present status of another peer.

conclusion

In this research project, I have highlighted how P2P with wi-fi direct is implemented in wireless network. Before this mostly P2P is used in wired network but we present a unique way of integrating a peer-to-peer network model with Wi-Fi Direct protocol over Android platform. Wi-Fi Direct can be beneficial for both consumers and service providers. The main objective of this thesis is to present how peer-to-peer based services can be efficiently realized in next-generation networks and emerging 3g or 4 technologies by reusing their existing protocols with new as much as possible, and to present some enhancements to these protocols.

REFERENCES

[1]http://www.wi-fi.org/files/20091019 Wi-Fi Direct FAQ.pdf.

[2] *AP2PS 2010 : The Second International Conference on Advances in P2P Systems:*Wireless P2P: Problem or Opportunity by Rossana Motta and Joseph Pasquale Department of Computer Science and Engineering University of California, San Diego La Jolla, USA

[3]http://ieeepkhi.org/studentseminar2011/doc/Android%20wiFi%20p2p.pdf

[4]http://www.dotcominfoway.com/attachments/268_White-paper-Android-by-2012.pdf

[5]www.ijcaonline.org/journal/number7/pxc387277.pdf

[6]http://www.wrox.com/WileyCDA/WroxTitle/Professional-Android-Application-Development.productCd-0470344717.html

[7]http://www.dbgroup.unimo.it/nep4b/risorse/d2.1.1.pdf

[8]citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.122
[9]citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.29...rep.

[10]http://gigaom.com/2009/07/21/will-p2p-soon-be-the-scourgeof-mobile-networks.

[11]Data in P2P Systems citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.116

[12] http://www.scientific.net/AMR.171-172.575

[13]http://developer.android.com/guide/topics/wireless/wifip2p.html

[14]http://java.sun.com/developer/Books/networking/jxta/jxtap2pch03.pdf

Visited Website :

[1] http://www.amazon.com/Android-Wireless-Application-Development-Conder/dp/0321627091

[2]http://ebookstore.sony.com/ebook/shaneconder/android-wireless-application-development/

[3] http://android-developers.blogspot.com

[4]http://it.toolbox.com/wiki/index.php/Dalvik_virtual_machine

[5]http://java.sun.com/developer/Books/networking/jxta/jxtap2pch03.pdf

[6]http://www.informit.com/articles/article.aspx?p=27302&seqNum=3

[7]http://www.tmcnet.com/sip/1106/sip-speaking-sip-1106.htm

[8]http://www.broadnets.org/2004/workshoppapers/Broadwim/broadwim2004-03-banerjeeinvite.pdf

[9]http://developer.android.com/guide/topics/network/sip.html

10] http://developer.android.com/

[11] en.wikipedia.org/wiki/Wi-Fi

[12] en.wikipedia.org/wiki/Java_(programming_language)

[13] en.wikipedia.org/wiki/Peer-to-peer

[14] www.wifinotes.com/

[15] www.skifactz.com/wifi/wifi_sip_phone.htm

[16] www.globalcache.com/support/wifitutorial/

[17]www.wi-fiplanet.com/tutorials/.../Wi-Fi-Hotspot-Security-The-Issues.htm

Vol. 2 Issue I, January 2014
ISSN: 2321-9653

[18] www.securitytube.net/WiFi-MAC-header-(Part-1)-video.aspx

[19]forums.comodo.com/.../nintendo-wifi-connectiontutorial-wii-ds-t4429.0.html

[20] www.p2pnet.net/categories/p2p

[21] www.p2p-blog.com/

[22] www.p2psip.org

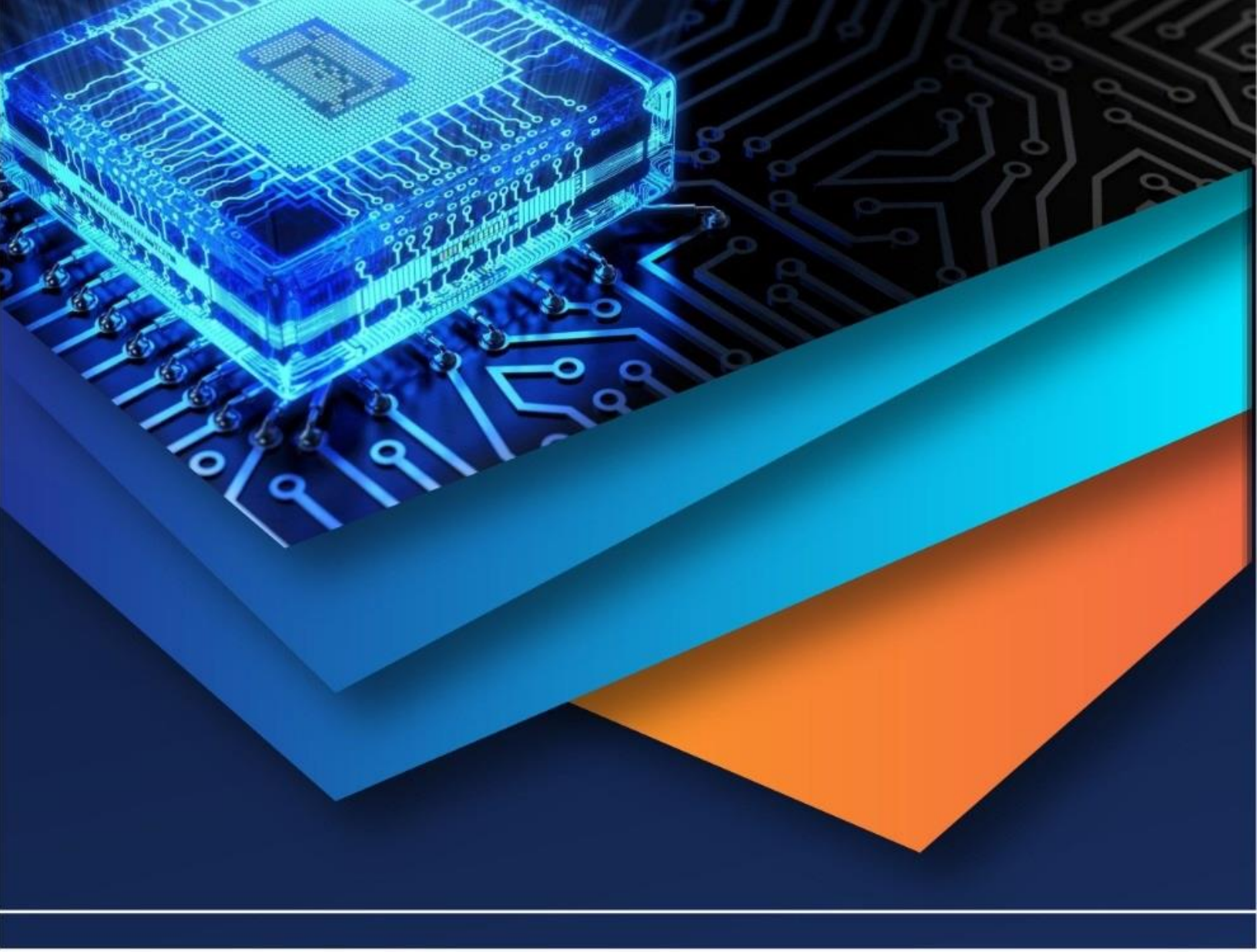
[23]www.oracle.com/us/technologies/java/index.h

[24]www.ies.co.jp/math/java/download.oracle.com/javase/tutorial/

[25]www.java2s.com/Tutorial/Java/CatalogJava.htm

[26] www.ibiblio.org/java/javatutorial.html

[27]www.java2s.com/Tutorial/Java/...PDF/Catalog0419__PDF.htm



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY