Enhancement of Bluetooth and Bluetooth Smart stack for Android based devices

BY

PANCHAL AKASHKUMAR G.

14MCEC17

तमसो मा ज्योतिर्गमय

NIRMA UNIVERSITY

INSTITUTE OF TECHNOLOGY

NAAC ACCREDITED ‘A’ GRADE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

AHMEDABAD-382481

Enhancement of Bluetooth and Bluetooth Smart stack for Android based devices

Major Project

Submitted in partial fulfillment of the requirements

For the degree of

Master of Technology in Computer science & Engineering

PANCHAL AKASHKUMAR G.

14MCEC17

Guided By

Prof. Manish Chaturvedi

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Mr. Ravi Nagarajan

BLUETOOTH SOFTWARE ENGINEERING, BROADCOM

तमसो मा ज्योतिर्गमय

NIRMA
UNIVERSITY

INSTITUTE OF TECHNOLOGY

NAAC ACCREDITED 'A' GRADE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

AHMEDABAD-382481

Certificate

This is to certify that Mr. Akash G. Panchal (14MCEC17), a student of M. Tech. Computer Science(CSE) , Institute of Technology, Nirma University is working with us since 15th June, 2015 and carried out his thesis work titled ”Enhancement of Bluetooth and Bluetooth Smart stack for Android based devices”. He is working as a Software Development Intern under the supervision of Mr. Ravi Nagarajan, Sr. Manager, Bluetooth SW Development. He has successfully completed the assigned work and is allowed to submit his dissertation report. We wish him all the success in future.

Mr. Ravi Nagarajan
Sr. Manager
Bluetooth Software Development,
Broadcom Pvt. Ltd.
Bengaluru

Certificate

This is to certify that the project entitled ”ENHANCEMENT OF BLUETOOTH and BLUETOOTH SMART STACK FOR ANDROID BASED DEVICES” submitted by PANCHAL AKASHKUMAR G. (14MCEC17), towards the partial fulfillment of the requirements for the degree of Master of Technology in Computer Science & Engineering of Nirma University, Ahmedabad is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this Seminar, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Prof. Manish Chaturvedi
Associate Professor,
Dept. of Computer Science and and Engg.,
Institute of Technology,
Nirma University, Ahmedabad

Dr. Priyanka Sharma
Coordinator M.tech CSE & Professor,
Dept. of Computer Science & Engg.,
Institute of Technology,
Nirma University, Ahmedabad

Dr. Sanjay Garg
Professor and Head,
Dept. of Computer Science and and Engg.,
Institute of Technology,
Nirma University, Ahmedabad

Dr. P.N. Tekwani
Director,
Institute of Technology,
Nirma University, Ahmedabad

Statement of Originality

I, **Panchal AkashKumar G.**, **14MCEC17**, give undertaking that the Major Project entitled "**Enhancement of Bluetooth and BluetoothSmart stack for Android based devices.**" submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Computer Science & Engineering** of Institute of Technology, Nirma University, Ahmedabad, contains no material that has been awarded for any degree or diploma in any university or school in any territory to the best of my knowledge. It is the original work carried out by me and I give assurance that no attempt of plagiarism has been made.It contains no material that is previously published or written, except where reference has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

Signature of Student
Date:
Place:

Endorsed by
Prof. Manish Chaturvedi

Abstract

Bluetooth technology is a standard for exchanging data over short distances enabling simple and secure connectivity for the wide range of devices like Phones, TV, Headset, etc. AVRCP is designed to provide a standard interface to control TVs, hi-fi equipment, or others to allow a single remote control (or other device) to control all the A/V equipment to which a user has access.

Contents

Chapter 1

Literature Survey

1.1 Bluetooth Technology

Bluetooth is a global wireless communication standard that connects devices together over a certain distance. Bluetooth Smart technology with its new updatable platform and low-power consumption, creates new application development opportunities for the cell phones, electronics, Computers, automotive, health and fitness, smart home systems and retail industries.

The Bluetooth core system consists of an protocol stack, base-band, and RF transceiver. It offers different services enabling the connection of devices and the exchange of a variety of data. Many features of the core bluetooth specification are optional to use allowing for the flexible product implementations for firmwares and softwares.

- **Client/Server Architecture:** All the profile of bluetooth have been designed with the use of traditional client/server architecture concept. Server are the devices that usually characterized as having data to be sent to some other device, examples are Temperature sensor, iBeacon. A "client" devices are those devices that need the data or the information from a server devices, examples are Mobile Phone, Tablet or Smart TV.

- **Device Discovery:** Device Discovery is possible because of the inquiry process. In inquiry client will send some message that will interpreted by near devices that is called server devices.After that server device will respond to message to client device. This process allows the client device to create a list of potential server devices to connect with.

  In Bluetooth low energy server devices continuously broadcast their presence using advertisement messages. And Client listens for advertisements to create a list of interested partners using scanning.

- **Connection Establishment:** Below are the steps to establish the connection :

  - How often data or control message is exchanged between client and server devices.
  - Using Security Manager Protocol establish secure connection.
  - Verifying of client and server if they are compatible.

- **Data Exchange:** Once a connection has been established, the client devices may request for the information from the server devices, send the information for the server to acquire, or command to the server devices to take an action.It can also client and server will exchange their roles [There is role switch request command].Here Server or client device will send needed information that is needed to peer device. Here, both devices can simply switch their roles: the server becomes the client and the client becomes the server.

- **Connection Termination:** If communication between both devices has been completed, one of can terminate the connection in orderly manner. In the termination process, the devices may decide to connect again at some point of time in future. This will help to reduce the power consumption. Next time they don't have to follow full process of establishing connection if they are paired previously.

1.2 Bluetooth Low Energy

Bluetooth low energy (BLE) known as Bluetooth Smart is a wireless personal area network technology designed for the applications specially in the health-care, fitness, security And home improvement and entertainment industries.

Compared to Classic Bluetooth, Bluetooth LE is intended to provide considerably reduced power consumption while maintaining a similar communication range with reduced cost.

Bluetooth Low Energy provides:

- Very low peak, power consumption reduction
- Can use standare coin cell as power source
- Implementation costs low as bluetooth
- Support Multi-vendor
- Improved range of connection

Enhancement to the Bluetooth Core Specification allows mainly two types of implementation, single-mode and dual-mode.

- **Dual-mode implementation:** A dual-mode device is an IC that includes both a conventional Bluetooth radio chip and a BLE radio chip. In the dual-mode implementation IC, Bluetooth low energy radio is integrated into an conventional Bluetooth controller chip. The resulting architecture shares both of Classic Bluetooth technology's existing radio and functionality resulting in a minimal cost increase compared to conventional Bluetooth technology.

  Moreover, manufacturers can use current Bluetooth technology with the new low energy stack of BLE, by enhancing the development of Bluetooth enabled devices with new capabilities.

- **Single-mode implementation:** Single-mode chips, which enables highly compact but highly integrated devices, also features a lightweight Link Layer by providing very-low power idle mode operations which includes device discovery, point-to-multipoint transfer of data with power-save and secure connections at the lowest cost.

  The Link Layer in enables the Internet connected sensors to schedule traffic between Bluetooth transmissions.

1.2.1 Technical details

- **Data Transfer -** Bluetooth Smart (Bluetooth LE) transfers data packets having less size that are transferred upto 1 Mbps. All BLE connections use advanced sniff-sub rating to achieve low duty cycles.

- **Frequency Hopping -** Bluetooth Smart (Bluetooth low energy) uses the adaptive frequency hopping common to all versions of Bluetooth technology to minimize the interference from the other similar technologies in the same 2.4 GHz ISM Band.

- **Host Control -** Bluetooth Smart (Bluetooth low energy) places a significant amount of knowledge about when to go to sleep to save the power. Mostly the device is sleeping only, it will wake up by host or some event occure. This is the most big advantage of bluetooth LE.

- **Latency -** Bluetooth Smart (Bluetooth low energy) can support connection setup, data transfer and connection termination as low as 3ms, allowing bluetooth applications to form a bluetooth connection to transfer data in few milliseconds for a short communication burst followed by quickly finish the connection.

- **Robustness -** Bluetooth Smart (Bluetooth low energy) uses a 24 bit CRC on all packets ensuring the maximum robustness against radio interference.

- **Strong Security -** Same as bluetooth it also supports full AES-128 encryption using CCM which helps to provide very strong encryption and authentication of data packets.
- **Topology -** Bluetooth Smart (Bluetooth low energy) uses a 32 bit access address on every packet for each slave to be connected. Mostly it is made with one to one connection but to support multiple connection it is using star topology.

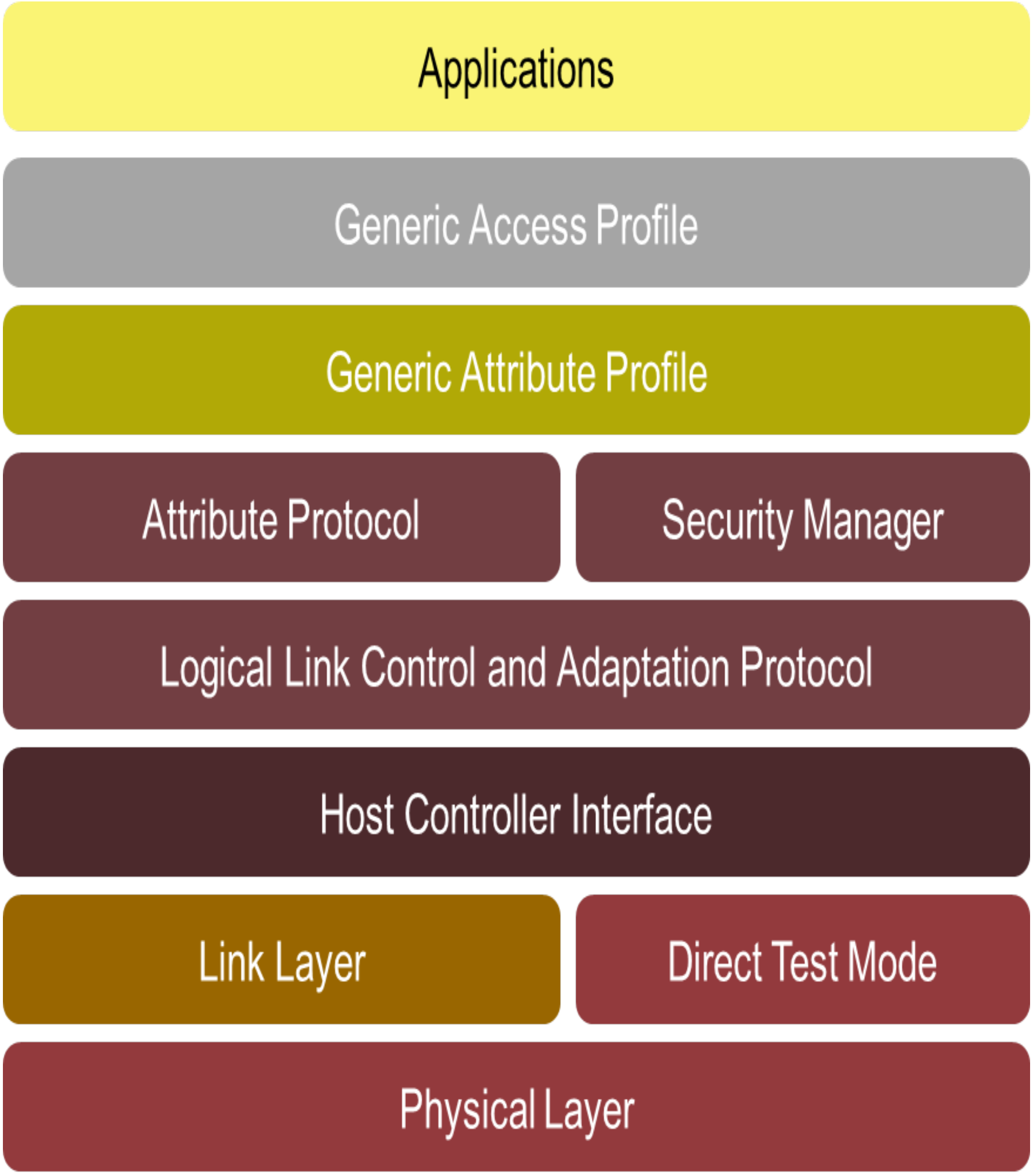


Figure 1.1: Bluetooth Low Energy stack

GAP (Generic Attribute Profile) : Using ATT (attribute protocol) bluetooth 4.0 onwards are using service base model. GAP provide data communication for all LE data exchange. It is using proper way so one application or profile can exchange information with others.

The server contains a number of attributes, and the GATT Profile defines how to use the Attribute Protocol to discover, read, write and obtain indications. These features support a service-based architecture. The services are used as defined in the profile specifications. GATT enables you to expose service and characteristics defined in the profile specification.

The GATT architecture makes it easy to both create and implement new profiles, as an example :

- Alert Notification Profile
- Device Information Service
- Health Thermometer Profile
- Heart Rate Profile
- Phone Alert Status Profile
- Proximity Profile

Generic Attribute Profile : The Generic Attribute Profile (GATT) establishes the in detail of how to exchange user data over a BLE connection. In contrast with GAP, which defines the low-level interactions with Bluetooth Low Energy devices, GATT deals only with the actual data transfer procedures and the formats.

All the standard BLE profiles are therefore based on GATT profile and must comply with it to operate correctly. This makes GATT a key section of the BLE specification, Here every single data item relevant to the apps and the data must be formatted, packed, and sent according to its rules.

Logical Link Control and Adaptation : The Bluetooth logical link control and adaptation protocol (L2CAP) supports higher-level protocol multiplexing, segmentation of packets and reassembly of data, and the confirming the QoS information.

L2CAP permits higher level protocols and apps to receive and transfer the upper layer data packets up to 64 KB in size. It also permits per-channel flow control and retransmission via the Flow Control and Retransmission.

Host controller interface : A host controller interface (HCI) is a register-level interface that enables a host controller for FireWire hardware to communicate with a host controller driver in software. The driver software is typically provided with an OS of a personal computer or mobile device , and can also be implemented by application-specific devices such as a wearables(microcontroller).

Link layer : The link layer much provides idle mode low power operation, device discovery and reliable data transfer and termination of connection with advanced power-saving functionality.

Physical layer : The radio uses the 2.4 GHz ISM band for communication. It divides this band into 40 channels each of 2 MHz from 2.4000 GHz to 2.4835 GHz. Out of these 40 channels 0-36 are used for connection data and the last three channels (37, 38, and 39) are used as advertising and scanning channels to set up connections and broadcast data.

Chapter 2

Analysis

Bluetooth Low Energy operates in 2.4Ghz frequency and that is divided into 40 channels. This channels are created using GFSK modulation and by using this we can get maximum 1Mbps speed. Data access can be of two type of 1) TDMA (Time Division Multiple Access and 2) FDMA (Frequency Division Multiple access.

In FDMA technique from 2.4GHz frequency 2Mhz of total 40 channels are hoping by frequency hopping algorithm. From 40 channel 37 channels are used as the data or information channel and 3 channel are used as advertising channel.

In TDMA technique one device send data at predetermined time and other device scan device responded to first device after specific predetermined time. In BLE there is major two type of events 1) Advertising Event and 2) Connection Event.

Using advertising channel we can put the device into discoverable mode. After that the other device which is in scanning that will respond to that advertising device with respond data. If scanner device is interested then it will send connection request to establish bi directional link.

While creating the connection both device will use only advertising channel only. After that after the connection they will start data transmission using data channel. In the data transfer all the configuration parameter is set while the connection only. The configuration parameter includes 1) Connection interval : that define time after that both device communicate 2) Connection latency : if it is non zero than that many attempts other device can ignored 3) Supervision Timeout : that define the time of creating connection, If the connection creation is not completed within that time the connection will be timeout.

After the connection the scanned becomes the master and advertiser becomes the slave device. If they want to exchange the role then they can switch the role by permission of both devices. The connection configuration parameter can also be modified after the connection using connection update parameter event. For the termination of the connection any of device can be initiate the disconnect request and close the connection.

2.1 Bluetooth v/s Bluetooth Smart

Parameters	**Bluetooth**	**Bluetooth Smart**
Data Throughput	3 Mbps	upto 100 Kbps
Robustness	Good	Good
Connection Range	Up to 50m	More than 100m
Large scale network	Weak	Good
Connection set-up	slow	fast
Power	High	Low
Cost	minimal	extra

2.2 Saving Energy

For ensuring the low energy in the BLE major two modification take a big role 1) Different type of technical technique and 2) different set of Radio. In addition in BLE we also have low energy sleep mode that is very useful to save the energy. The sleep parameter is depends on the use case it is not specified by the SIG.

Major energy is saved using the limited channel for initiating the connection. Mean we have only 3 advertising channel so the device which ever interested only have to scan only 3 channels. So most of energy is saved because in the BE/EDR total 32 different channel need to be scan for that.

BLE takes 0.6 to 1.2 ms for scanning while BR/EDR takes 22.5 ms as it has to scan 32 channels. Approximately 10 to 20 times of power is saved using this technique.

Chapter 3

AVRCP profile on Android

The Audio/Video Remote Control Profile is a Bluetooth profile that allows Bluetooth devices to control media playback on remote devices. It is typically used with A2DP devices for next/previous track selection and pause/play functions.

AVRC is a Bluetooth profile that allows remote control of media playback on other devices. Supported functions are play, pause, stop, next, and previous. It is usually coupled with A2DP so that, for example, wireless headphones use A2DP to stream the music and AVRC to control playback.

AVRCP is intended to give a standard interface to control TVs, Wifi hardware, or others to permit a solitary remote control (or other gadget) to control all the A/V gears to which a client has admittance. It might be utilized as a part of show with A2DP or VDP.

The AVRCP defines two roles, that of a controller and target device.

- **Controller:** A Remote control device is called the controller.
- **Target:** Device which act upon the commands is caled the Target.

3.1 Design Specification

In AVRCP, there are two device come in a picture, a controller and a target. controller sends the action to target device via Bluetooth. The remote control which is controller here is designed specifically for A/V control only.

AVCTP defines different message and procedures to be sent across for controlling the A/V devices. Service discovery protocol helps to check if the same service is available or not on both the device. AV control is AV/C command-based entity.

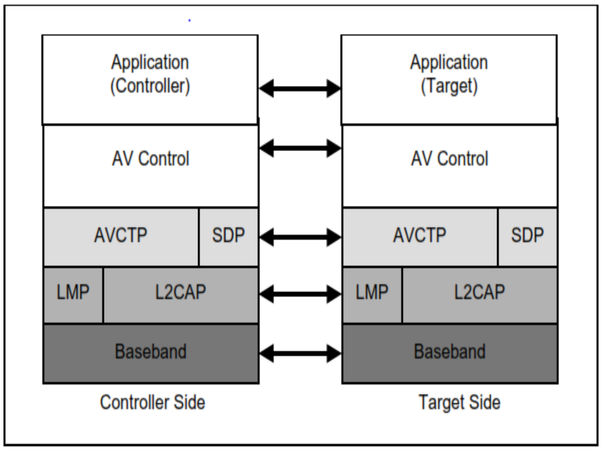


Figure 3.1: Profile Stack

3.1.1 Configuration and Roles

- The **controller (CT):** Controller or CT is a device sends commands to the target. For the examples Handsfree, Smart watch or smart car or music player, music recorder or display monitors.
- The **target (TG):** Target or TG can be device which receives commands from the CT and acts accordingly following which generating a response. For the examples Mobile phone, music player/recorder, a smart TV, an amplifier, pocket player.

3.1.2 Connection for Control

Connection establishment of L2CAP for an AVCTP instance can be instantiated by the TG or can be instantiated by the CT. Any event either internal or the external event(user generated), like power off/on or initiating the connection establishment manually.

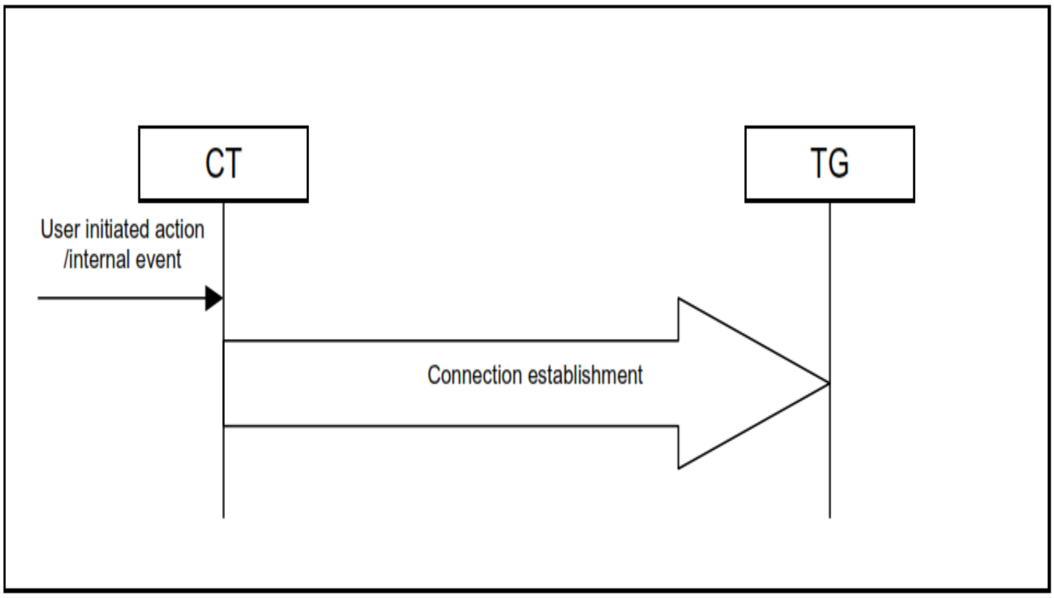


Figure 3.2: CT : Connection Initiated by CT

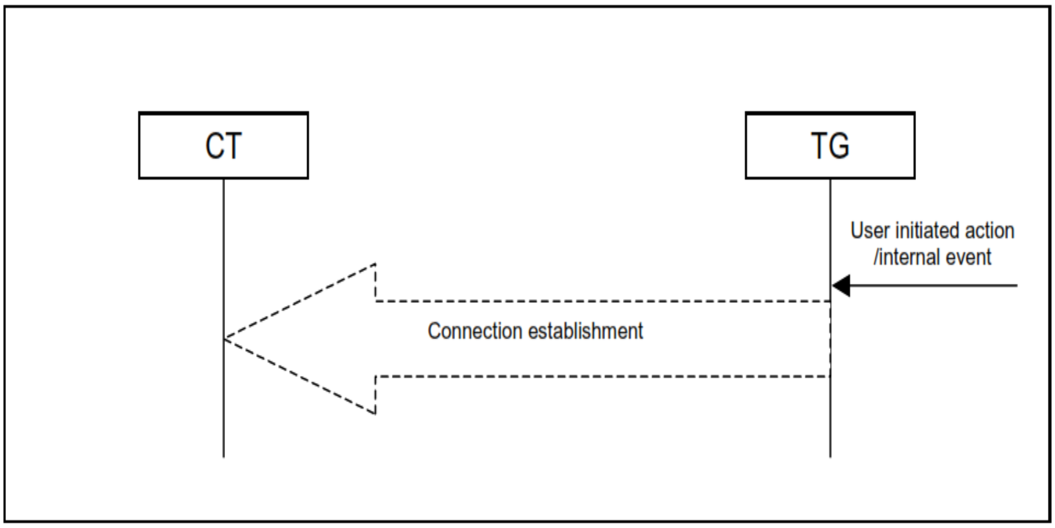


Figure 3.3: TG : Connection Initiated by TG

3.1.3 Release Connection for Control

Release of the connection falls the same.The Release of current L2CAP connection can be triggered by the CT or by the TG. Any event either internal event or external (user generated) even like turning the power off/on or initiating the connection manually.

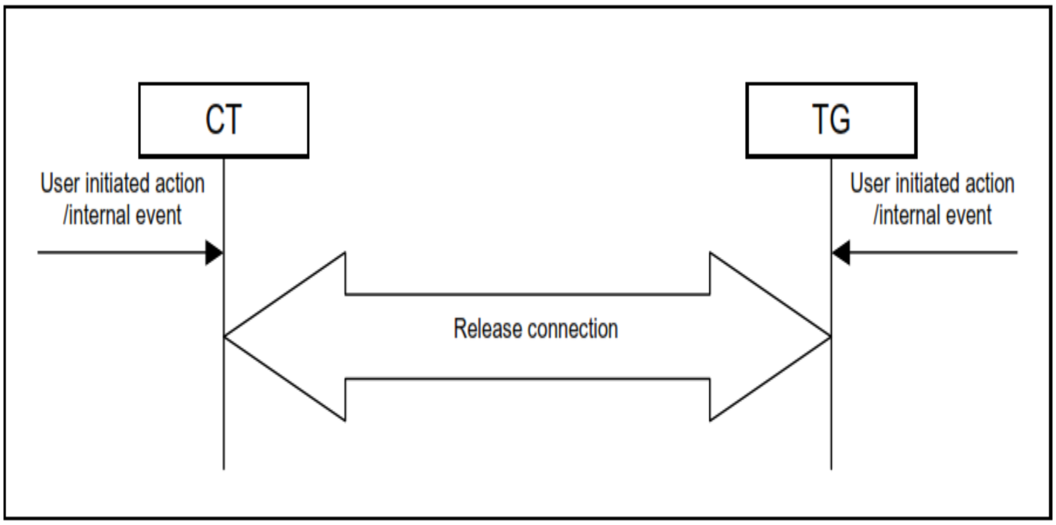


Figure 3.4: Connection Release Initiated by CT or TG

3.2 Operation

3.2.1 Procedure of AV/C Command

In case of any internal or any external, the CT initiates the connection(if it is not there already) After the connection establishment, it can send AV/C command for controlling.

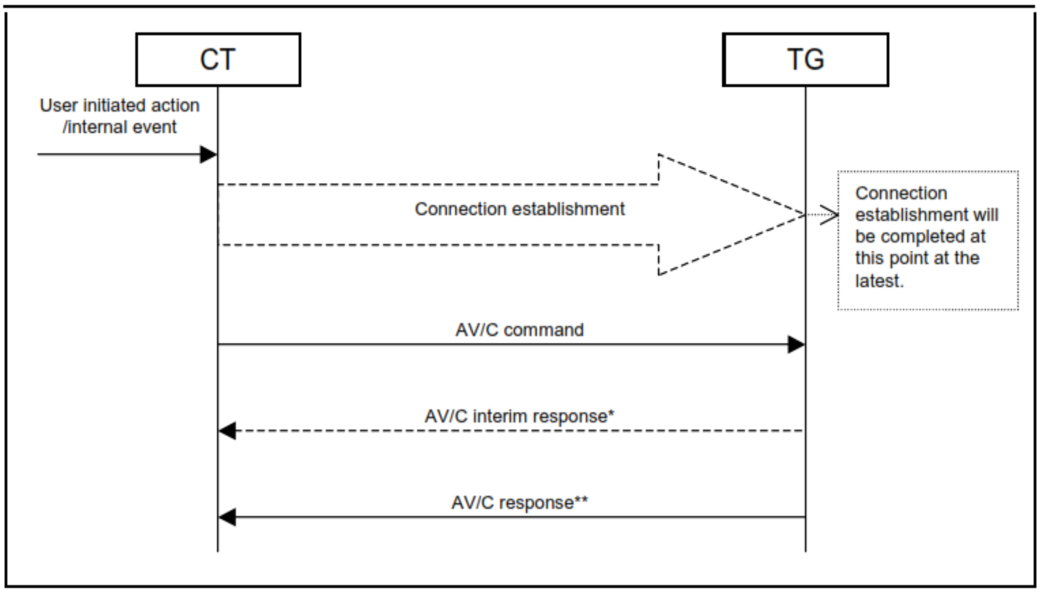

CT
TG
User initiated action /internal event
Connection establishment
Connection establishment will be completed at this point at the latest.
AV/C command
AV/C interim response*
AV/C response**

Figure 3.5: Procedure of AV/C Command

3.2.2 AV/C Command Operation

The general determination of AV/C indicates general model, unit/subunit model and standard unit/subunit charges. An AV/C subunit instantiation is only a substance distinguished inside an AV/C unit. It has set of capacities the gadget can gives. Distinctive capacities are characterized for every classification of gadgets in the determination of the subunit.

AV/C order comprises of the general determination and subunit summon for each. Unit information and Subunit data are obligatory for the general particular. It additionally relies on upon the gadget usage that which subunit to support.

Unit data is utilized to get the data about the AV/C unit in general, While the Subunit summon is utilized to acquire data about the subunit of the unit.

Principle highlight of this profile is only controlling the remote gadget by the PASS THROUGH summon. The subunit gives a client driven model to activating the controls on a gadget. At the point when the client attempt to change a progressing operation utilizing client interface,the controller sends charge to the subunit.

In return the TG performs the desired action(S).

Regardless of the fact that having numerous subunits in a TG, the TG should have stand out board subunit. The principle motivation behind the board subunit is to permit it to decipher the approaching client activity charges into interior activities, which may influence other subunit.The affect can be TG-independent events, results may have an effect on media streams. A controller conveys these user operations to the Target device (TG) using the PASS THROUGH command.

3.2.3 Rules for the AV/C Transaction

AV/C transaction is set of messages with commands (cmd frame) for TG and zero or message having a response(response frame) within the specified time periods.

All the transaction should comply within the time period as described as follow. TG shall respond with $\mathrm{T}_{\mathrm{RCP}}(100)$ starting from the instance a cmd is received.

If the TG is not able to respond within time, it can response with INTERIM meaning that final response may follow later.

For the Metadata transfer, following time period is defined. TG shall response within the time period of $\mathrm{T_{MTC}}$(200) once receiving the command.

$\mathrm{T_{MTP}}$(1000) time is the period for Target device to generate required response the INTERIM for the NOTIFY cmds and FIANAL for STATUS cmds.

3.2.4 Command Frame

Following is the structure diagram of AV/C command frame which contains 512 bytes data. It is part of AVCTP command/response message information field.

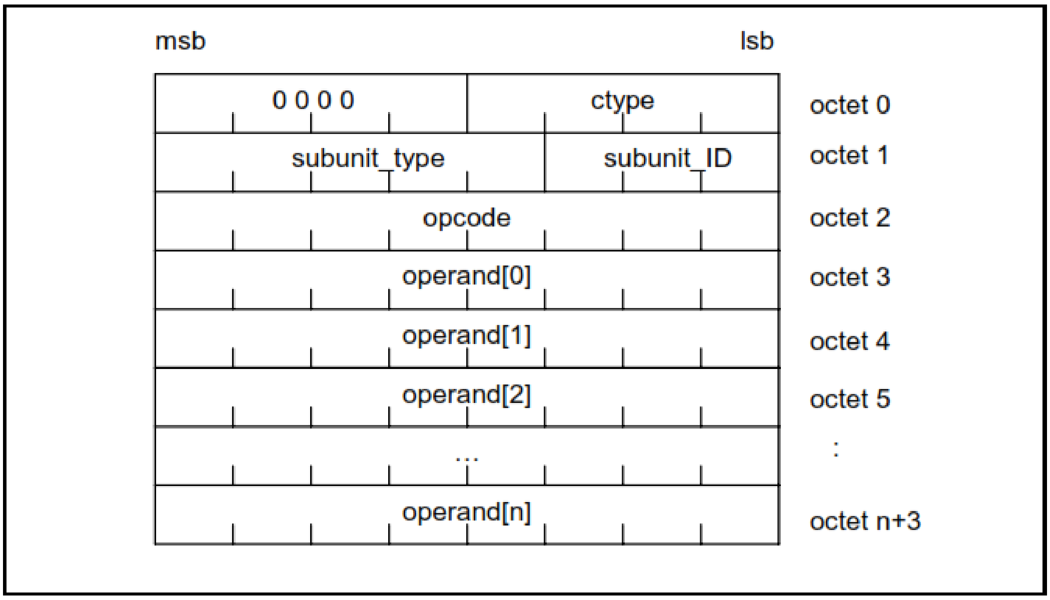


Figure 3.6: Command Frame

Here, Operands are optional and depends on the submit type and opcode.

3.2.5 Response Frame

Following is the structure diagram of AV/C response frame which contains 512 bytes data. It is part of AVCTP command/response message information field.

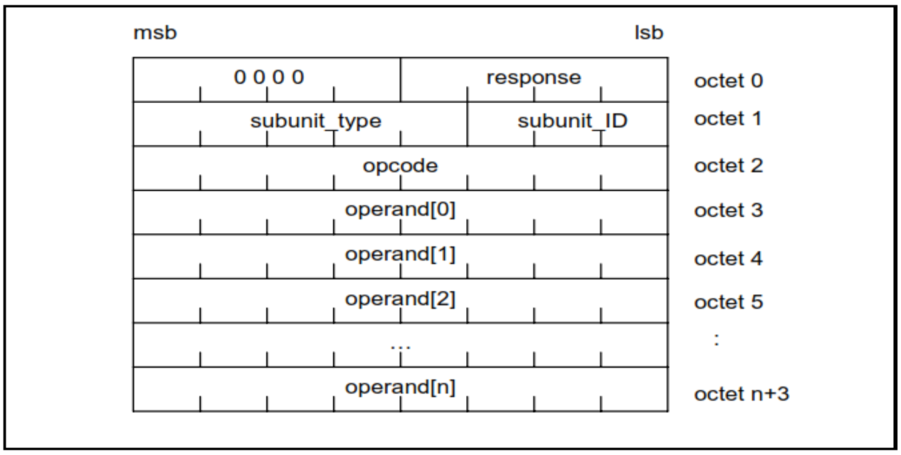


Figure 3.7: Response Frame

Here, Operands are optional and depends on the submit type and opcode.

3.2.6 Event notifications from target device

Followings are the functionalities provided by the target device as a event notification.

1 CT can access current play position, status and duration of song.

2 Events that can be monitored on the target are,

- Play
- Pause
- Stop
- Forward Seeking
- Backward Seeking
- Position change

- Change of track
- Start of song
- End of song

3 CT can provide NOTIFY command to TG

4 TG can issue INTERIM response for every NOTIFY command.

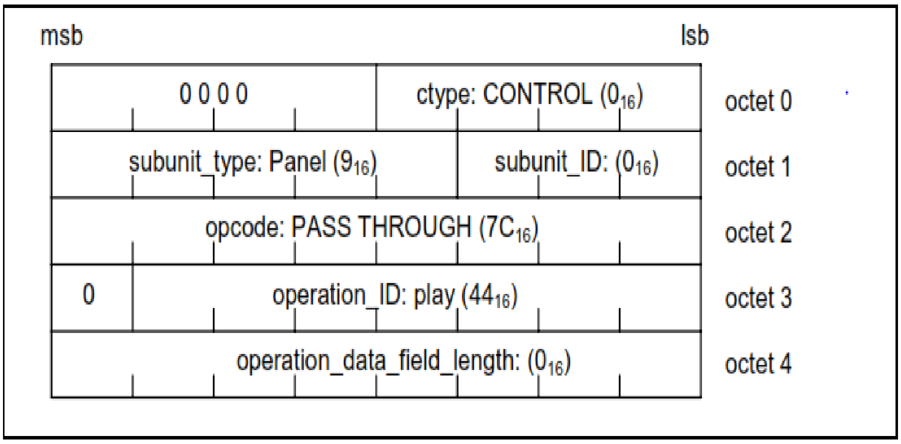
5 After getting any event the Target device shal notify with a RESPONSE the CT device with real time status.

6 There shall be atmost 16 labels of the transaction at any time in a session.

3.3 Working

3.3.1 AV/C Commands and Responses

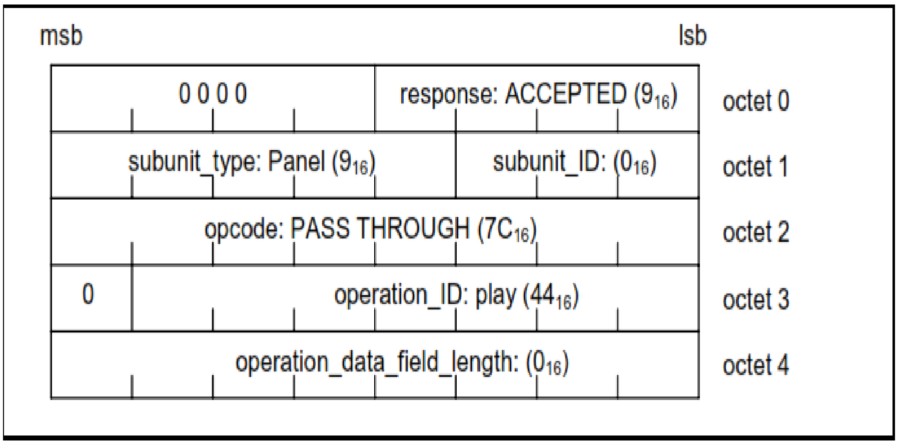
When ”PLAY” or ”PAUSE” button is paused on the CT, the PASS THROUGH command is sent. Frame structure of the PASS THROUGH is as shown below.



msb lsb

0 0 0 0	ctype: CONTROL (0_{16})	octet 0
subunit_type: Panel (9_{16})	subunit_ID: (0_{16})	octet 1
opcode: PASS THROUGH ($7C_{16}$)		octet 2
0	operation_ID: play (44_{16})	octet 3
operation_data_field_length: (0_{16})		octet 4

Figure 3.8: Command Frame for PASS THROUGH



msb lsb

0 0 0 0	response: ACCEPTED (9_{16})	octet 0
subunit_type: Panel (9_{16})	subunit_ID: (0_{16})	octet 1
opcode: PASS THROUGH ($7C_{16}$)		octet 2
0	operation_ID: play (44_{16})	octet 3
operation_data_field_length: (0_{16})		octet 4

Figure 3.9: Response Frame fro PASS THROUGH

Following is the example of Message Sequence Chart showing TG accessing track information, changing track using PASS THROUGH command from CT.

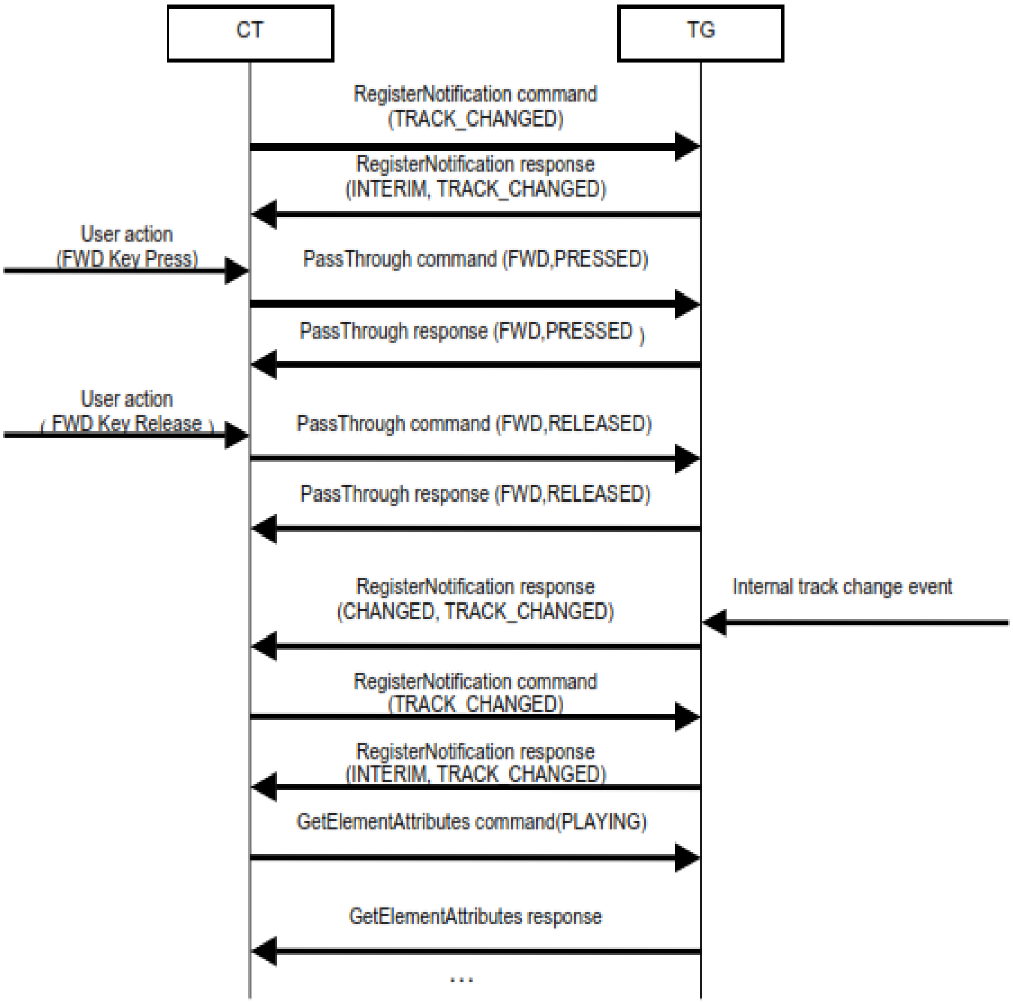

CT
TG
RegisterNotification command (TRACK_CHANGED)
RegisterNotification response (INTERIM, TRACK_CHANGED)
User action (FWD Key Press)
PassThrough command (FWD,PRESSED)
PassThrough response (FWD,PRESSED)
User action (FWD Key Release)
PassThrough command (FWD,RELEASED)
PassThrough response (FWD,RELEASED)
RegisterNotification response (CHANGED, TRACK_CHANGED)
Internal track change event
RegisterNotification command (TRACK_CHANGED)
RegisterNotification response (INTERIM, TRACK_CHANGED)
GetElementAttributes command(PLAYING)
GetElementAttributes response
...

Figure 3.10: Message Sequence Chart

Chapter 4

Android Automation Testing Tool

Android applications keep running on the gadgets with little CPU force, memory and battery supply. The conduct of these applications likewise relies on upon some different things like data connection , wifi network, and so on.

Thus, its critical to test, troubleshoot and enhance Android App. Having a decent test scope for Android application improves the application.

It is impossible to test the Android applications on all the mobile or devices, it is a good practice to performe Android tests on the device with the typical configurations. One ought to test the application on the mobile device with the most reduced and most astounding accessible design to guarantee that it works fine on these gadgets.

What to test on Android applications

- 75-85 % unit tests : To make sure the stability of the code
- 25-35 % functional tests : To make sure that the app works as expected
- Cross functional tests to check if the application integrates with other Application components such as gallery, contacts, etc

4.0.2 Unit Testing for Android

Unit testing for Android is classified as follows:

- Unit tests - tests that runs on the java virtual machine
- Instrumented tests - tests which runs on the actual Android system (Dalvik Virtual Machine on device or on mobile

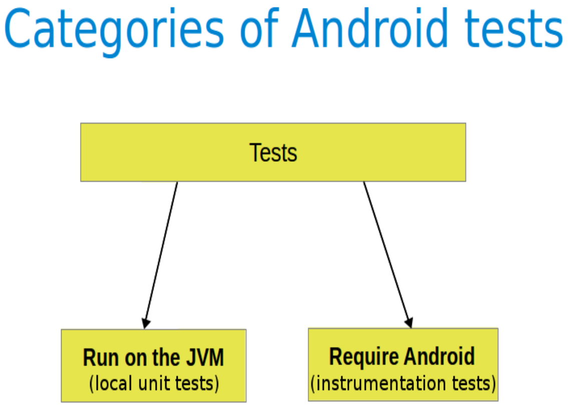


Figure 4.1: Android Testing Categories

4.1 Unit Tests

Gradle plug-in supports in Android helps to run Android unit tests on the java JVM. Gradle builds a new version of the android.jar. This java archive file is provided to the unit test thus all classes and methods are available to JVM. But any call to the this JAR results in an exception by default.

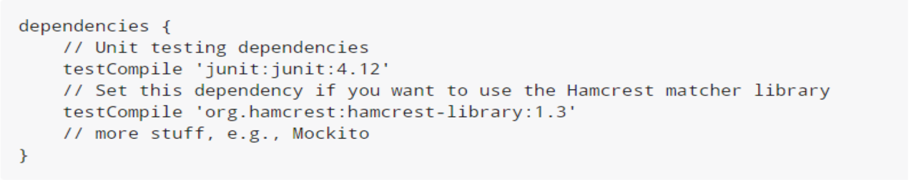
Thus, if classes from the App doesn't call any Android API in the Application life cycle, the JUnit testing framework (JUnit3/JUnit4)can call without any restrictions. If there is any dependencies to the Android API, these dependencies in the code should be replaced for the unit tests.

Unit test on JVM executes very fast compared to running on Android. Unit test are those which can be run on the JVM of the local machine.

A unit test verifies the functional behaviour of a certain component or method in isolation. For the example, if the ImageButton from the app is used to toast the message. A unit test should check that if the intent is sent or not when the button is clicked, not that the message was toasted or not.

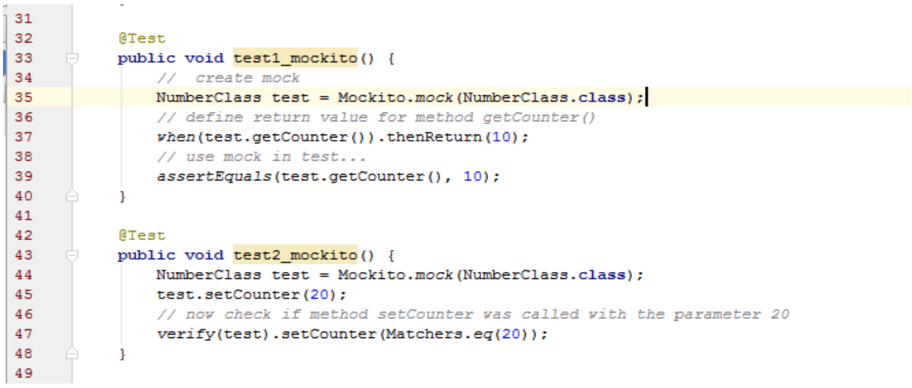
The unit tests executes on a modified version of the Android library of the Application where all final modifiers from the methods and classes have been removed. That allows to mocking libraries. Methods in the used android.jar file of the Android library throws exceptions. This make sure that the unit tests only verifies the code instead of depending on the Android platform.

Required dependencies in the Gradle build file:



```
dependencies {
    // Unit testing dependencies
    testCompile 'junit:junit:4.12'
    // Set this dependency if you want to use the Hamcrest matcher library
    testCompile 'org.hamcrest:hamcrest-library:1.3'
    // more stuff, e.g., Mockito
}
```

Figure 4.2: Gradle build file



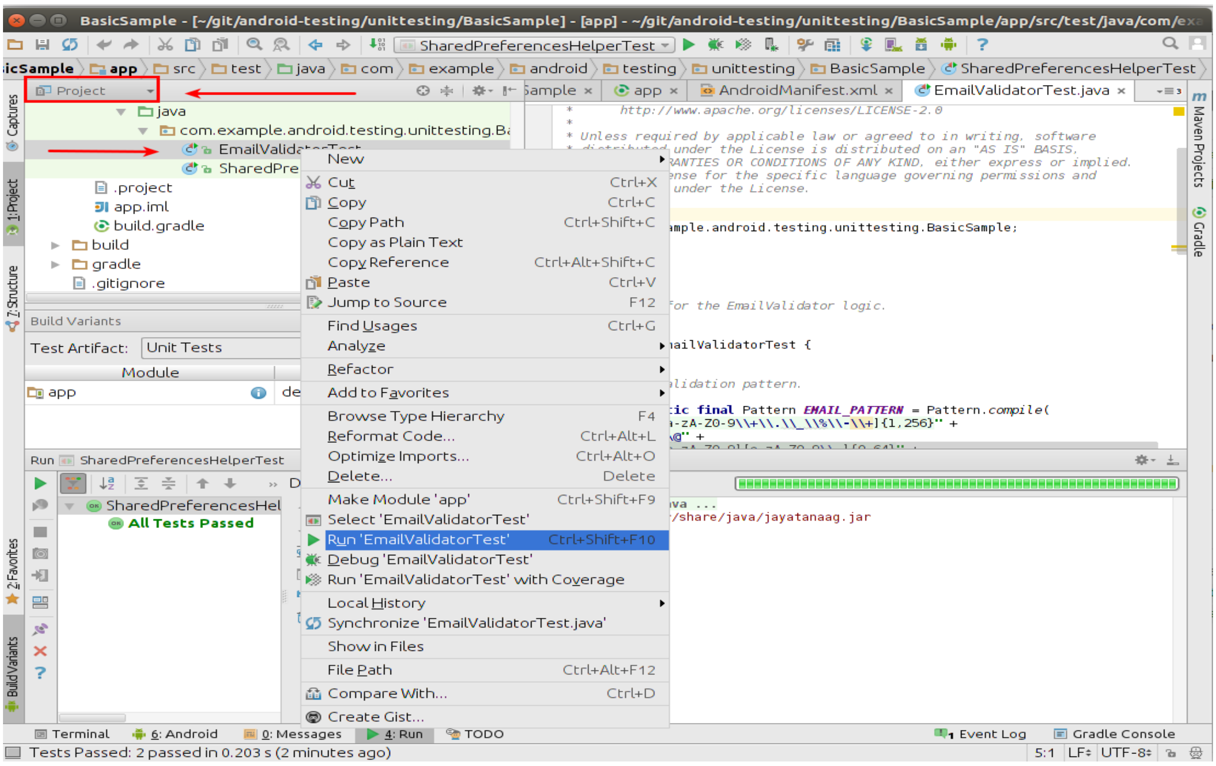
```

@Test
public void test1_mockito() {
    //  create mock
    NumberClass test = Mockito.mock(NumberClass.class);
    // define return value for method getCounter()
    when(test.getCounter()).thenReturn(10);
    // use mock in test...
    assertEquals(test.getCounter(), 10);
}

@Test
public void test2_mockito() {
    NumberClass test = Mockito.mock(NumberClass.class);
    test.setCounter(20);
    // now check if method setCounter was called with the parameter 20
    verify(test).setCounter(Matchers.eq(20));
}

```

Figure 4.3: Example Mockito test

Run the unit tests from Gradle: execute **gradlew test** command to run unit test using gradle.

Run the unit tests from Android Studio

select the test class and select run.



BasicSample - [~/git/android-testing/unittesting/BasicSample] - [app] - ~/git/android-testing/unittesting/BasicSample/app/src/test/java/com/exa

```
 *      http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
RANTIES OR CONDITIONS OF ANY KIND, either express or implied.
ense for the specific language governing permissions and
 under the License.

ample.android.testing.unittesting.BasicSample;

For the EmailValidator logic.

nailValidatorTest {

alidation pattern.
tic final Pattern EMAIL_PATTERN = Pattern.compile(
a-zA-Z0-9\\+\\.\\_\\%\\-\\+]{1,256}" +
\@" +
```

```
va ...
/share/java/jayatanaag.jar
```

SharedPreferencesHel
All Tests Passed

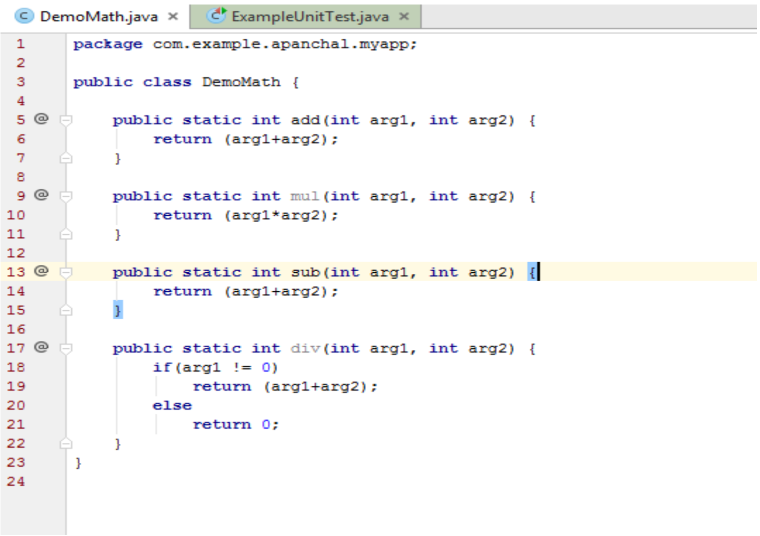
Tests Passed: 2 passed in 0.203 s (2 minutes ago)

Figure 4.4: Run Android Test

Test Reports:

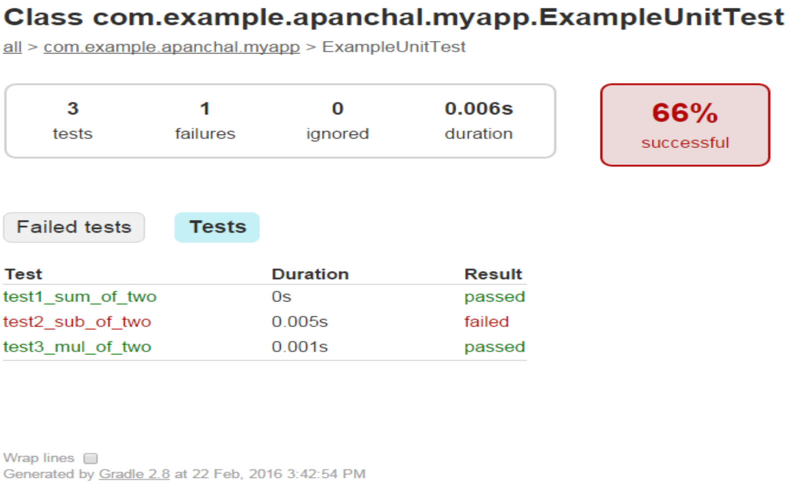
Reports are created in the app/build/reports/tests/debug/‘directory of the Application source. The ‘index.html provide the abstract and links to the individual test pages which can be viewed using browsers like Internet Explorer or Chrome.

Example of Unit Test in Android



```java
package com.example.apanchal.myapp;

public class DemoMath {

    public static int add(int arg1, int arg2) {
        return (arg1+arg2);
    }

    public static int mul(int arg1, int arg2) {
        return (arg1*arg2);
    }

    public static int sub(int arg1, int arg2) {
        return (arg1+arg2);
    }

    public static int div(int arg1, int arg2) {
        if(arg1 != 0)
            return (arg1+arg2);
        else
            return 0;
    }
}
```

Figure 4.5: Unit Test code



Class com.example.apanchal.myapp.ExampleUnitTest

all > com.example.apanchal.myapp > ExampleUnitTest

3	1	0	0.006s	66%
tests	failures	ignored	duration	successful

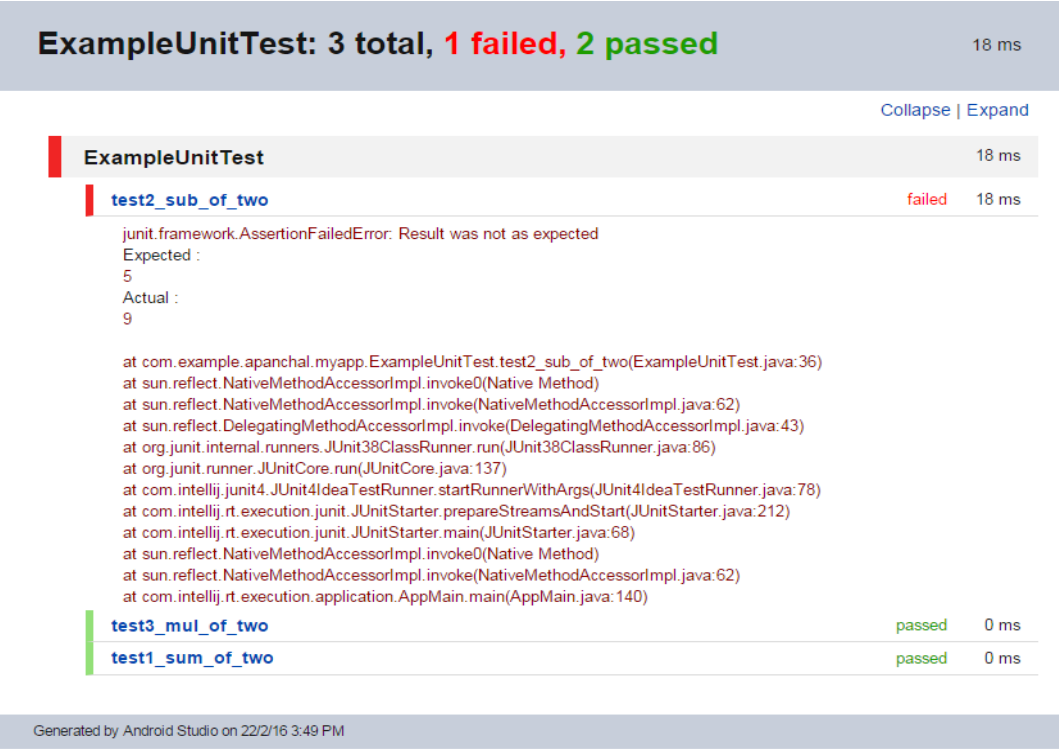
Test	Duration	Result
test1_sum_of_two	0s	passed
test2_sub_of_two	0.005s	failed
test3_mul_of_two	0.001s	passed

Wrap lines □

Generated by Gradle 2.8 at 22 Feb, 2016 3:42:54 PM

Figure 4.6: Test Report

Example of Unit Test in Android



ExampleUnitTest: 3 total, 1 failed, 2 passed 18 ms

ExampleUnitTest 18 ms

test2_sub_of_two failed 18 ms

```
junit.framework.AssertionFailedError: Result was not as expected
Expected :
5
Actual :
9

at com.example.apanchal.myapp.ExampleUnitTest.test2_sub_of_two(ExampleUnitTest.java:36)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at org.junit.internal.runners.JUnit38ClassRunner.run(JUnit38ClassRunner.java:86)
at org.junit.runner.JUnitCore.run(JUnitCore.java:137)
at com.intellij.junit4.JUnit4IdeaTestRunner.startRunnerWithArgs(JUnit4IdeaTestRunner.java:78)
at com.intellij.rt.execution.junit.JUnitStarter.prepareStreamsAndStart(JUnitStarter.java:212)
at com.intellij.rt.execution.junit.JUnitStarter.main(JUnitStarter.java:68)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
at com.intellij.rt.execution.application.AppMain.main(AppMain.java:140)
```

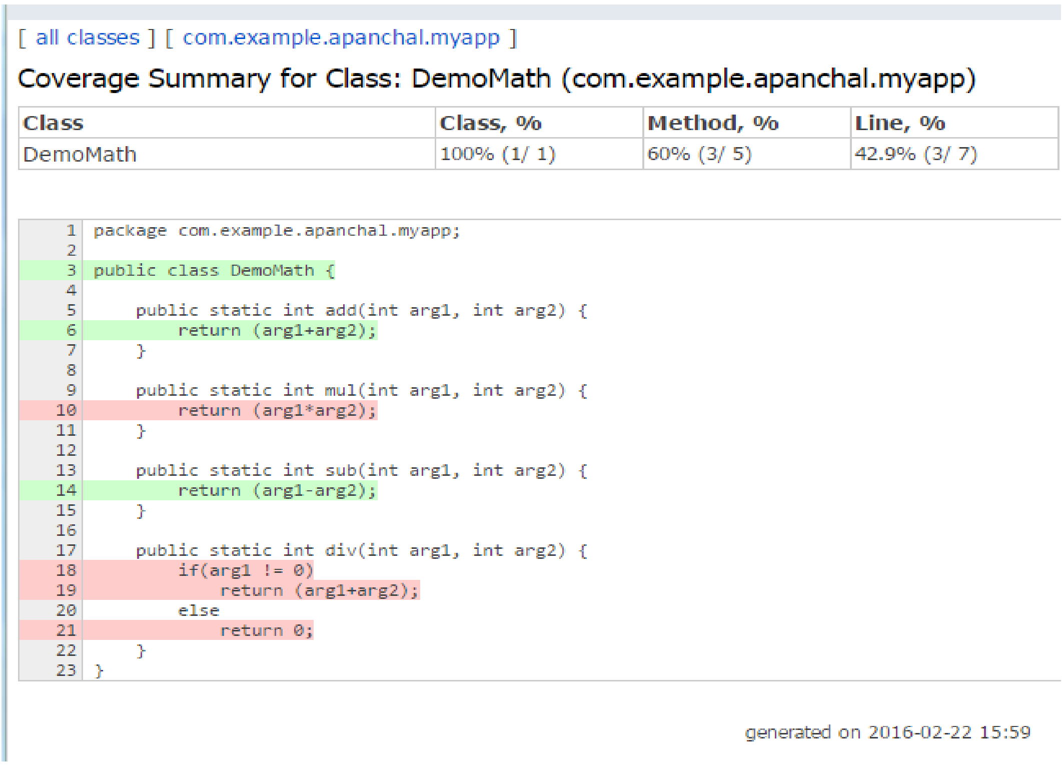
test3_mul_of_two passed 0 ms

test1_sum_of_two passed 0 ms

Generated by Android Studio on 22/2/16 3:49 PM

Figure 4.7: Test report with error

Example of Unit Test in Android



[all classes] [com.example.apanchal.myapp]

Coverage Summary for Class: DemoMath (com.example.apanchal.myapp)

Class	Class, %	Method, %	Line, %
DemoMath	100% (1/ 1)	60% (3/ 5)	42.9% (3/ 7)

```
package com.example.apanchal.myapp;

public class DemoMath {

    public static int add(int arg1, int arg2) {
        return (arg1+arg2);
    }

    public static int mul(int arg1, int arg2) {
        return (arg1*arg2);
    }

    public static int sub(int arg1, int arg2) {
        return (arg1-arg2);
    }

    public static int div(int arg1, int arg2) {
        if(arg1 != 0)
            return (arg1+arg2);
        else
            return 0;
    }
}
```

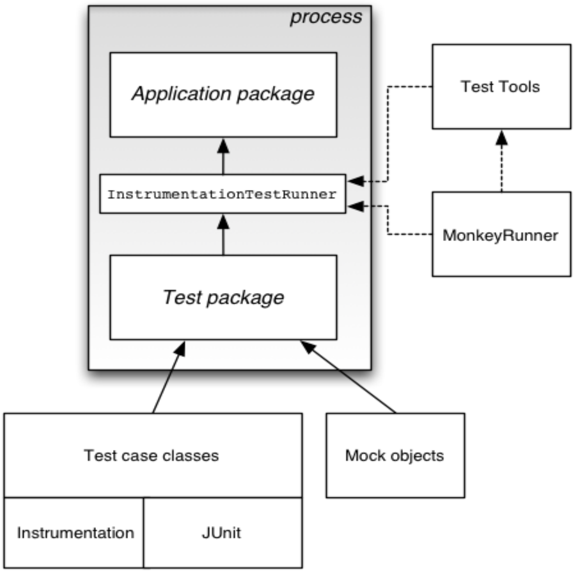
generated on 2016-02-22 15:59

Figure 4.8: Code Coverage during testing

4.2 Instrumentation Test

The Android testing API gives screen guides into the Android segment and application life cycle. Which is instrumentation APIs that screens the android application from the outside the application.

Instrumentation framework provides the ability to control the life cycle events of the application which a user can not control while using the app. As an example, the user can not control when the onCreate() method called.



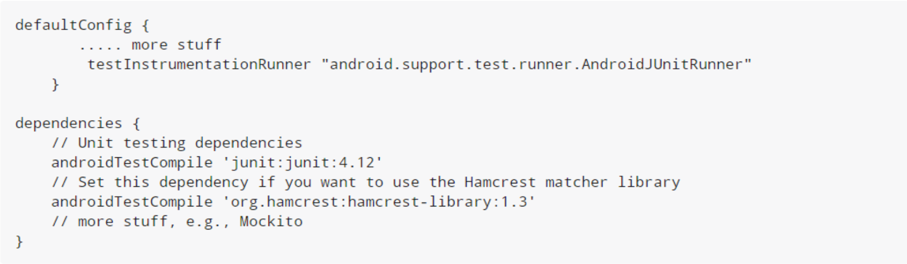
Instrumentation tests

Instrumented tests runa on Android devices or on emulators instead of local Java virtual machine. Also These it has full access to the android phone and components such as Camera, Radio, etc. An example is a test which validates turning ON of the Bluetooth Radio.

An instrumentation-based test class also provides the ability to send UI events like button pressed(or touch events) to the application under testing which can be very useful to test the usability of the application.

For example, your instrumentation test can start the activity. Afterwards, it can call the finish() and restart the activity to test if the instance state of the activitiy is correctly restored. This way you can execute whole life cycle of the Application from the outside of the Application.

Define dependencies, testInstrumentationRunner in the Gradle build



```
defaultConfig {
       ..... more stuff
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }

dependencies {
    // Unit testing dependencies
    androidTestCompile 'junit:junit:4.12'
    // Set this dependency if you want to use the Hamcrest matcher library
    androidTestCompile 'org.hamcrest:hamcrest-library:1.3'
    // more stuff, e.g., Mockito
}
```

Run tests using Gradle: Run ”gradlew connectedCheck” command on gradle console.

Run the unit tests from Android Studio: select the test class and select run.

Reports are generated in the app/build/reports/androidTests/connected/ directory.

The browsing file would provide the links to the individual test pages which can be viewed using browsers like Internet Explorer or Chrome.

Chapter 5

Summary

Bluetooth enables devices to share files, music, data using low power technology compared to other wireless technology. AVRCP enables the device to be controlled by CT devices like SmartCar, Headset, SmartWatch, etc. Android Automation testing tool is useful to test underlying Android APIs.

References

1 Specification of the Bluetooth System v4.2, Bluetooth SIG

2 Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology, Carles Gomez, Joaquim Oller and Josep Paradells

3 http://developer.android.com

4 http://Androidxref.com

5 Bluetooth AVRCP Spec v1.3