

UPnP and Google Chromecast based Mobile Media Streaming Application for TV

¹Priyank Mandalia, ¹Yanyan Yang

¹School of Engineering, University of Portsmouth, United Kingdom
priyank.mandalia@myport.ac.uk; linda.yang@port.ac.uk

Abstract

Streaming media content like movies and musical compositions which are stored locally in phone or PC directly to TV wirelessly sanctions us to disseminate information to wider audience in a convenient way. This field has magnetized people from both academia and industry as this can lead to whole incipient possibilities in the field of regalement and broadcasting. In this paper, we discuss the current situation in media streaming from mobile phone to TV and incipient possibilities in this field. A mobile application that enables people stream media which is stored in the phone directly to TV utilizing coalescence of UPnP technology and Google Chromecast [9]. To achieve that, we have developed M2TV1 algorithm. We utilize different scenarios to illustrate the possibilities and usefulness of the application predicated on M2TV architecture. The application is now available on Google Play store with the name "Vbio" for anyone to download and in addition to that, the application has been downloaded and relished by thousands of users around the world with very good reviews.

Keywords: *Mobile to TV, PC to TV, Video Streaming, Audio Streaming, Home Networked Devices, Wireless Technology, Chromecast, UPnP.*

1. Introduction

It was reported that 81% of the people in the world owned a mobile phone [18]. Even more surprisingly the recent figures says there are more number of mobile phones than there are people. According to the recent survey 80% of the smartphone users have Android smartphone [19]. Another survey tells that 73% of the household in UK have Wi-Fi or Broadband connection [6]. Internet connectivity has become basic need in this era of Information technology where Information is at the fingertips of the people. Streaming media content from mobile device directly to TV enable people to disseminate information to wider audience in a convenient way. This field has attracted people from both academia and industry as this can lead to whole incipient possibilities in the field of regalement and broadcasting..

Technologies like DLNA [7] which is developed by Sony [20], is progressing in this field. The only issue is that less number of TVs and smartphone incorporate this technology. Google has recently released a photos application where people can stream their photos to TV using Chromecast. But these photos should be in cloud server [5] for them to be streamed. So, this is not the solution we are looking for. Apple Airplay [1] allows you to stream online media content.

Streaming online content can now be achieved through Google Chromecast, but what if user wants to stream the media content stored in the device? Also, Phone should be able to connect to TV seamlessly and make it phone's secondary display, so the user can see the notification and read them on the TV itself. So even the phone's display is off it should be displaying all the important information on TV screen.

To achieve the ability of streaming local media content we have developed an android application. This application works on the combination of UPnP [23] technology and Chromecast. So, now the user can stream videos and songs to TV seamlessly. One more advantage of the application is that phone's display need not to be on, so the user can turn-off the display and streaming will still continue. Not only user can stream media content stored on the phone, but can even stream the media content stored on PC. User can control the playback of the media through application. One more feature of the

application is the ability to control playback of the media through Google Android Wear smartwatch [2]. So, once the video is playing, user can pause and play the video or song from the watch itself.

The rest of the paper is structured as follows. Section 2 presents related work. Section 3 gives an overview of the system. Section 4 explains M2TV architecture and design. Section 5 discusses M2TV algorithm. Section 7 presents system evaluation. Section 8 concludes the paper.

2. Related Work:

In September 2012, a solution based on Smart TVs was released [21]. It described about the connectivity challenges associated with Smart TV and incipient technologies which can be incorporated in it. But the proposed system fell short of connectivity between sundry smartphone applications and TV and it does not sanction utilizer to stream local media content from phone or laptop to TV or HD projector.

In May 2013, a Media sharing system was released [22]. It designs and implements an integrated architecture that fortifies the alfresco remote control to home contrivances and the sharing of digital media among indoor and alfresco contrivances. Though it does provide facility to apportion media on DLNA fortified contrivances, but its key quandary is that it does not provide firm solution for streaming local media content to TVs and HD projectors which are not DLNA certified.

Chandra et al. [12] presents home media center and media clients for multi-room audio and video regalement and describes the remote user-interface protocol, which is predicated on remote frame buffer (RFB) protocol and remote utilizer interface technology. Users in another room can visually examine live digital TV programs, relish multimedia contents, and access the Internet with full GUI information utilizing low-cost media client connected to home media center via home network, and these media clients can be facilely embedded into exhibit contrivances. Home media center and media clients are UPnP compatible contrivances. This system does provide centralized solution for sharing media in the home, but it does not provide facility of streaming media content stored on user's phone via application that is more utilizer cordial.

Venkitaraman et al. [26] presents a mobile content sharing mechanism predicated on UPnP framework and wireless technology. As a result, users could conveniently share media contents stored in mobile contrivances and play them by other home networked contrivances. The developed system does provide the most proximate solution to what we are endeavoring to achieve in our system, but support less number of contrivances compared to our system.

Fasbender et al. [8] and Wei-Shun et al. [25] presents a media streaming solution based on DLNA and UPnP. This system allows users to stream media using DLNA compliant devices in a Wireless Network. This system does meets all the requirement for media streaming, but does not work when TV or HD projector is not DLNA certified.

Lorenzo et al. [15] discusses the scenario where group of users are visually examining same video utilizing mobile Internet concurrently. They have designed a system called MicroCast which utilizer resources on all smartphones of the group in a cooperative way to ameliorate the streaming experience. Key quandary of this system is that it does not sanction users to visually examine different videos individually and it does not address the issue of streaming media content from phone or laptop to TV or HD projector.

The problem with all the media streaming application, is that they does not provide ideal solution for streaming local media content stored in phones or tablets to TV.

In this paper, we are interested in the scenario where a smartphone user or group of smartphone users are interested in watching a video or play music which is stored in the phone on TV wirelessly. Consider, for example, that a user wants to show video clip stored in his/her phone to a group of friends; user wants to listen song stored in the phone on TV; or someone wants to broadcast video to wider audience. The default option is to connect the phone to TV via USB port or HDMI port and view the media files. However, this is not the ideal solution as it limits the user to stay in the bounds of the length of cable. One more scenario where in the business meeting during presentation, the presenter wants of show video clip stored in his/her phone or laptop to the members via HD projector wirelessly.

3. System Overview

The aim of the M2TV system is to engender an Android application that streams local media content to TV, utilizing Java programming language. The application uses Wi-Fi to connect with Chromecast contrivance connected to TV and act as a transmission medium to transmit media files.

The M2TV Android application can be installed on all smartphones and Tablet running Android 4.0 or later versions of the Operating System. The application has implemented UPnP technology which engenders media server on the phone. This will then show list of videos and musical compositions on the phone and then utilizer can connect to Chromecast device and stream media files to TV.

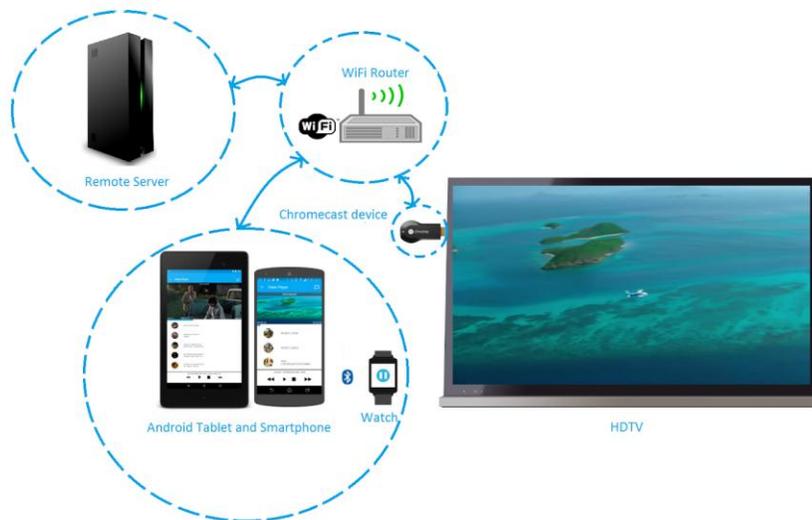


Figure 1. System Overview

As all the laptops support UPnP technology, the application can connect to Laptop and show list of videos stored on the laptop on phone screen, provided the laptop is connected to same Wi-Fi network. The utilizer can play the videos and control the playback through phone.

From the users' perspective, they can download the application from Google Play store and install on their phone. The application is self-explanatory and facile to understand. The utilizer just need to cull the device they optate to connect to TV, that is either their phone or laptop. Then the application will show list of media content which can be played on the TV when utilizer connects the application with Chromecast device by just pressing Cast icon. Utilizer can then utilize the phone as remote to control the playback of the media files playing on TV. In this way, M2TV system provides service which no other Android application does.

4. M2TV System Architecture and Design

The M2TV system architecture is predicated on UPnP (Universal Plug and Play) and Google Chromecast. UPnP is a networking protocol which sanctions devices such as mobile phone and PC discover each other if they are in the same network and provides network connection for services like data sharing and media streaming.

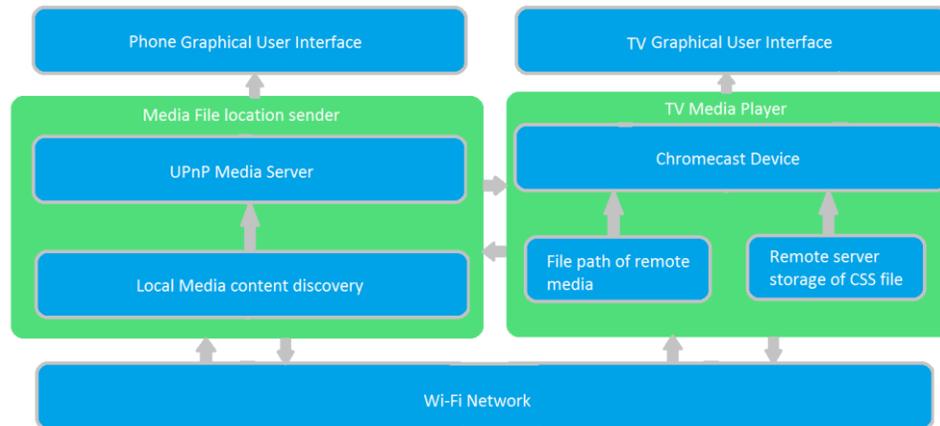


Figure 2. M2TV System Architecture

As shown in the above architecture UPnP media server is engendered with avail of Wi-Fi. The media server will then get the media content either videos or musical compositions predicated on the utilizer input from the smartphone's local storage. Then it will show the media content as list view on the phone. The utilizer then needs to connect the application to Chromecast utilizing Cast icon, which will sanction utilizer to play media on TV. When user play any media content, the application sends http address of the file stored on media server to Chromecast and commences playing it.

On the TV side when application connects to Chromecast, it loads receiver application along with its user-interface. The receiver application is stored on Google server where it is registered. The user-interface of TV is stored on a separate remote server as CSS file and sundry image assets.

4.1 Wi-Fi Network

Android Wi-Fi API is utilized to engender media server. This sanctions the transmission of media files in the shared network among UPnP enabled devices. This API sanctions application to connect to Chromecast device. After matching the unique application id which is assigned to the application when the application is registered on Google Cast developer website, the receiver application on Chromecast contrivance downloads GUI files for TV from the separate remote server and renders it on TV screen.

4.2 Media File Path Sender

Media file path sender as the name suggested is used to send media file path from the application to Chromecast device. It constitutes of two main components: 1) UPnP Media Server and 2) Local Media content discovery.

4.2.1 UPnP Media Server

Android UPnP library [3] sanctions you to engender media server on the device. This server can connect to other UPnP devices connected on the same network and can transmit and receive media file. To get the list of videos and musical compositions on the application we require to get the list of all device, which will be shown as device name given by the manufacturer. Clicking on it will pass device id to the media server, which will retrieve media content stored on the phone. When the media server engendered, it will retrieve data predicated on the utilizer predilection such as Videos, musical compositions and images as shown in figure 3.

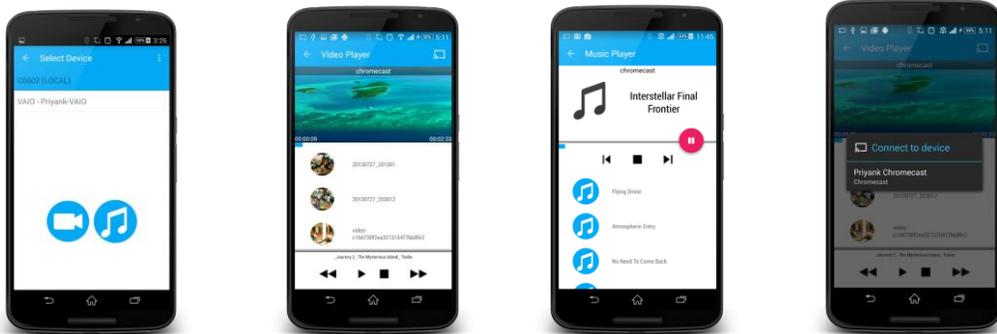


Figure 3. Device Selection **Figure 4.** Video Player **Figure 5.** Music Player **Figure 6.** Connect to TV

While engendering media server one of the most consequential thing to do is to assign http address to the file path of each media file, as Chromecast accepts http address of the media content. For that we had to implement a code which assigns http address while retrieving list of media files.

4.2.2 Local Media Content Discovery

The local storage is where the media files such as videos, songs and Images are stored. Media server will get list of files along with its address from each and every folder in the storage. Suppose, if user selects videos, then it will get list of all the videos stored in the phone's storage and show it on list view long with its thumbnail.

4.3 Phone's Graphical User Interface:

Graphical user interface provides an interface to utilizer to interact with application to stream media to TV. It additionally sanctions utilizer to control playback of the media files by providing features as play/pause, stop, expeditious forward, rewind, next song, precedent song, etc. The code of GUI is written in XML according to the Android's Material Design guidelines. This application can run on any Android powered device either smartphone or tablets running Android 4.0 or later. For application to have consistent GUI over wide range of screen sizes, I have implemented separate GUI for contrivances of 4 to 6 inch screen size and devices such as tablets of 7 to 10 inch screen size. The application will automatically picks correct GUI predicated on screen size, hence providing consistent UI across contrivances. The design and implementation of GUI follows Google's latest guidelines of Material design, which makes application's GUI comely and smooth.

4.3.1 Video Player

Video player allows user to select videos which are to be streamed on TV. In addition to that it will also play videos on phone if application is not connected to TV. This is where user can

connect to Chromecast device when Cast icon is pressed. We have provided screenshot of video player as shown in Figure 4.

4.3.2 Music Player

Music player allows user to select songs which are to be streamed to TV. In addition to that it will also play songs on phone if application is not connected to TV. We have provided screenshot of music player as shown in Figure 5.

4.4 TV's Graphical User Interface

The Graphical User Interface of TV is stored on separate remote server as CSS file along with all the image resources such as application icon and background theme and color. This will give consistent GUI to TV while playing media files.



Figure 7. TV's GUI

4.5 TV Media Player

TV media player plays media files on TV. It constitutes of three main components: 1) Chromecast device, 2) Media file path acceptor and 3) Remote server storage of CSS file and image assets for user-interface.

4.5.1 Chromecast Device

Chromecast API [11] sanctions application to connect to Chromecast device and send http address of the media file and stream it. This will additionally sanctions us to control playback of the media files on TV. Utilizer can even increase or decrease volume of the media playback from phone by pressing hardware volume button. For that to achieve, we have overridden the volume button functionality, which connects it to media player running on TV.

Figure 6 shows screenshot of the application which gives option to connect to Chromecast device connected to TV and shared network.

4.5.2 Remote Server Storage of CSS files and Image assets

Remote server stores CSS file and image assets for the user-interface of the media player on TV. These files are linked with the mobile application, hence when the application connects to Chromecast device it gets those file through Chromecast API call and it renders on TV.

4.5.2 Media File Path Acceptor

Media file path acceptor as the name suggest, accepts media file paths which are stored in mobile phone or tablet, which is achieved using Chromecast API call.

4.6 Playback Control through Android Wear watch

We have implemented the feature of controlling the playback of the media content through Android wear smartwatch. Android Wear smartwatch shows all the notification exhibited on the phone on watch's display. So, to achieve that we came up with an algorithm which will engender notification each time whenever utilizer play incipient video or song, which will be withal shown on Android Wear smartwatch [2]. We have integrated the same functionality of controlling media playback shown on the above screen, on the notification utilizing notification manager class. This will sanction us to have same controls on smartwatch and as smartwatch is always connected to phone via Bluetooth, it will send pause and play signals to the application as a background process even when the phone's screen is off.

5. M2TV Algorithms

In this section, we present the algorithms that we developed to implement M2TV system.

5.1 UPnP Device Discovery Algorithm

```
if Wi-Fi is on and connected then  
search local and remote devices in the network  
if local device found then  
append name with (LOCAL)  
else if remote device found then  
append name with (REMOTE)
```

This is where we faced major challenge as we have to distinguish between local and remote device. For that we compare name of the device given by the manufacturer with the device name on which application is running and if the name matches then it is considered as local device and if not then it considered as remote device. Here, it will check whether Wifi is on or not and is connected. If it is connected UPnP will search for local device that is the device on which it is running and remote devices that are connected in the network. Then if it finds local device it will append device name with word "LOCAL" and remote device name with "Remote", so that user can easily understand it and connect to it.

5.2 UPnP Media Server Algorithm

```
if local or remote device selected then  
create media server  
if video selected then  
get list of video files  
else if music selected then  
get list of music files
```

This is one of the all major challenges we faced while developing the application, as we have search all the folders in memory storage of phone for getting list of videos and songs, so that it can be displayed in the list. Here, when the user selects local or remote device, it will create media server on the phone and when user selects videos option, it will get list of movies and videos stored locally in the phone and if user selects music option, it will get list of songs stored locally in the phone.

5.3. Chromecast Connection Algorithm

```
if Chromecast device available then  
    connect and get media player status  
else if Chromecast device not available then  
    display device not available
```

This is a very crucial part for controlling media playback, because if user tries to pause the video and application has been disconnected from Chromecast then it cannot be paused and will result into failure of the whole application. To overcome that issue, we check that if Chromecast device is available in the network, user can connect to it. When the user connects to it, it will get current media player status of TV to check if it's already connected and playing media on TV. If device is not available in the network it will display that no device is available.

5.4 Media Files Playback

```
if connected to Chromecast device then  
    send media file location to it and get media player status of TV for playback control  
else if not connected to Chromecast device then  
    play media files on application's video player or music player
```

Here, we allow user to play videos and songs on phone itself if it is not connected to Chromecast device. To achieve that we check regularly check if application is connected to Chromecast device and if the application is connected to TV then it will send media file location to the Chromecast device and will start playing it. If application is not connected to Chromecast device, it will start playing media files either movies or songs on the application itself. This feature will make application robust and more useful as media can be played even if the Chromecast device is not available.

5.5 Playback Control Algorithm

```
if application connected to Chromecast and playing on TV then  
    if pause button pressed then  
        pause the video playback  
    else if play button is pressed and media is paused then  
        Start media playback on TV  
else if application not connected to Chromecast then  
    if pause button pressed and media is playing then  
        pause the video playback  
    else if play button is pressed and media is paused then  
        Start media playback on phone
```

This part is very crucial and challenging as we have to check media player status every time, we try to either pause or play the media file. Here, when application is connected to TV and playing media on it, user can control playback of it. If user presses pause button it will send pause command to the media player on TV and pause the video if it was already playing. If media was paused on TV and user selects play then it will resume the playback from when it paused. Also, when application is not connected to TV, user can play, pause, fast forward and rewind video or songs on phone.

5.6 Android Wear Smartphone Media Playback Control Algorithm

```
if user selects media for playing then  
    send media info and control to android wear smartwatch  
if user clicks pause and media is playing then  
    pause the media playback  
else if user clicks play and media is paused then
```

start playing the media

We have given this functionality as an additional feature, which is not crucial for media streaming but provides more convenience if the user had Android Wear watch. Here, user can control playback of media files playing either on TV or on application itself through Android wear. Android wear is connected to phone via Bluetooth technology. When user plays any media file, it will send media information like name of song or video to watch and attach controls to it. So, user will have option of either pausing or playing the media.

6. Evaluation

M2TV works for sending local media content from smartphone or laptop to TV or HD projector. The application is now available on Google Play store with the name “Vbio” for anyone to download. It has been downloaded and relished by thousands of users around the world on phones and tablets with very good reviews [24]. Figure 8 shows the application on Google Play Store.

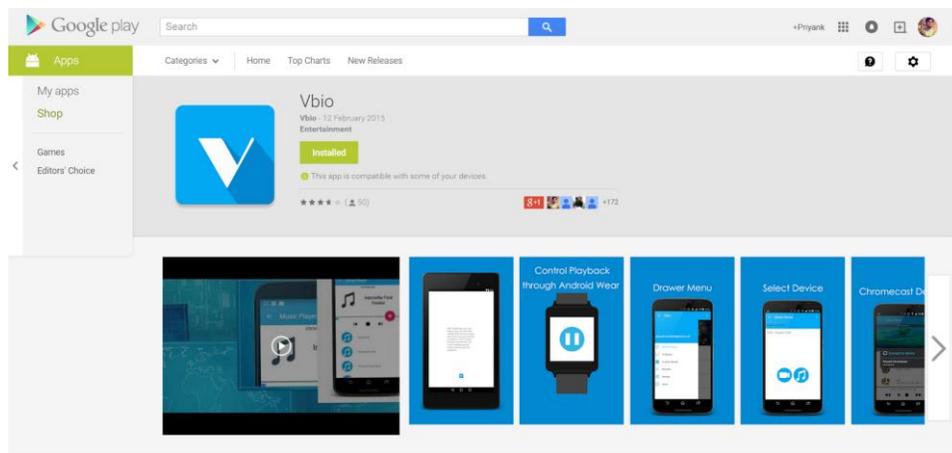


Figure 8. Vbio on Google Play Store

Figure 9 shows ratings and feedback given by the application user on Google Play store.

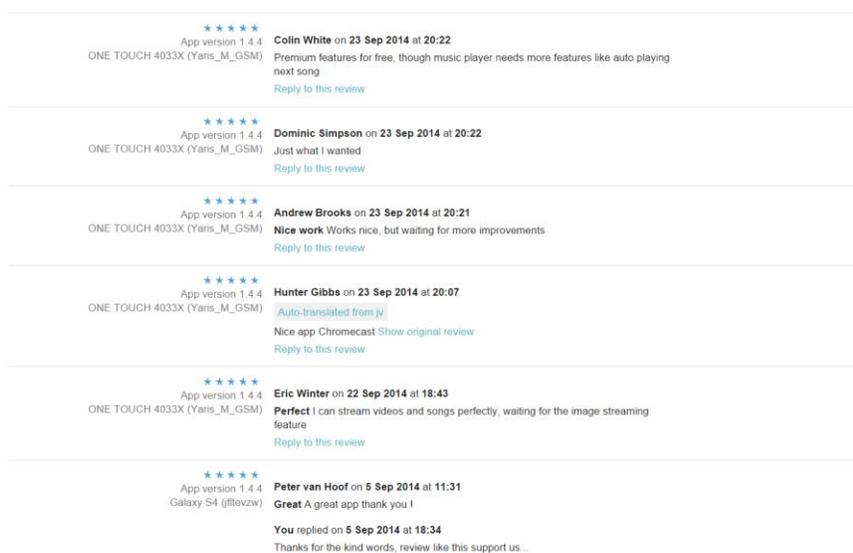


Figure 9. Ratings and feedback

Figure 8 and figure 9 are the statistics of the application at the time of writing this paper.

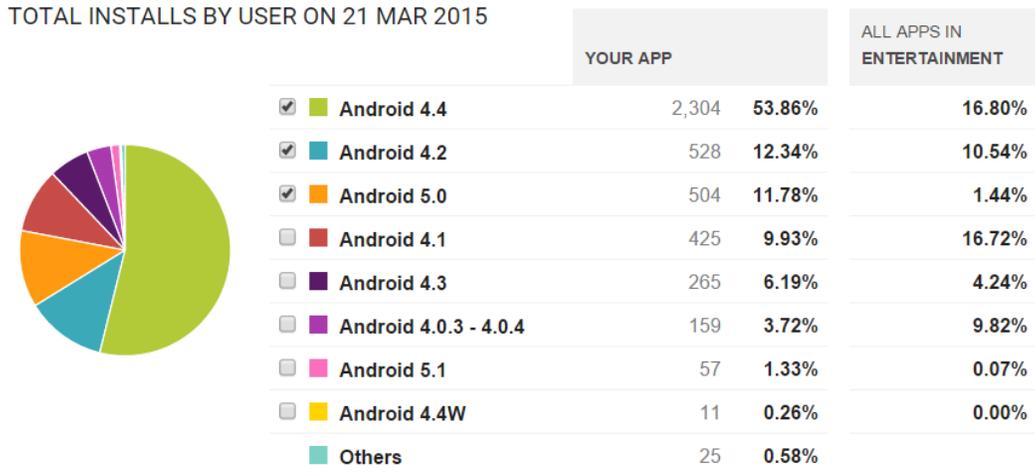


Figure 10. Download statics based on Android version

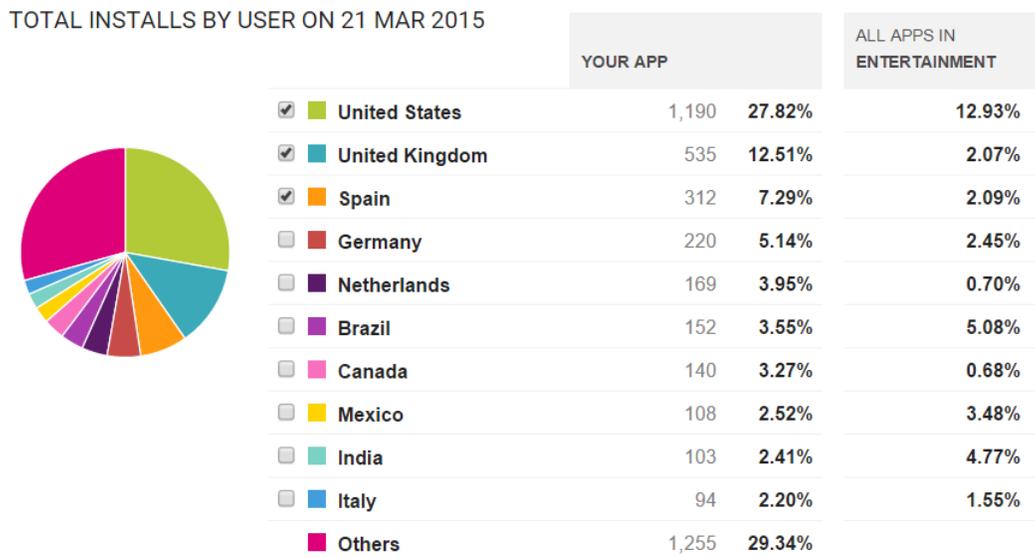


Figure 11. Download statics based on Country

7. Conclusion

Streaming media content to TV is an incipient field and is utilized by constrained number of users, due to lack of features like streaming local media files. In this paper, we designed, implemented and evaluated a novel mobile application, M2TV system that enables user to stream media files from mobile phones or laptop to TV or HD projector over Wi-Fi network. We developed algorithms to achieve it. As a supplemental feature we have elongated the remote control feature to Android Wear watch. The application is self-explanatory and facile to utilize and support wide range of Android devices from 3.7 inch smartphone to 10 inch tablet. The application had been available on Google Play store with the name "Vbio" for anyone to download. It shows from Google store that the application

has been downloaded and relished by thousands of users around the world with very good reviews. The application can be further enhanced with exhibiting all the phone's notification on the TV.

8. References

- [1] Apple Airplay <https://www.apple.com/uk/airplay/>
- [2] Android Wear <http://www.android.com/wear/>
- [3] Android UPnP library <http://4thline.org/projects/cling/>
- [4] Android TV <http://www.android.com/tv/>
- [5] A view of cloud computing. Communications of the ACM Volume 53 Issue 4, April 2010 Pages 50-58 ACM New York, NY, USA <http://dl.acm.org/citation.cfm?id=1721672>
- [6] Dutton, William H. and Blank, Grant, Next Generation Users: The Internet in Britain (October 16, 2011). Oxford Internet Survey, 2011. Available at SSRN: <http://ssrn.com/abstract=1960655>
- [7] DLNA for Industry <http://www.dlna.org/>
- [8] Fasbender, A., Gerdes, M., Matsumura, T., Haber, A and Reichert, F. Media Delivery to Remote Renderers Controlled by the Mobile Phone. Published in Consumer Communication and Networking Conference, 2009. CCNC 2009. 6th IEEE. Pages 1 - 2, 10 - 13 Jan. 2009.
- [9] Google Chromecast <http://www.google.com/chrome/devices/chromecast/>
- [10] Google <https://www.google.co.uk/about/company/>
- [11] Google Cast <https://developers.google.com/cast/>
- [12] Goo Jun. Home media center and media clients for multi-room audio and video applications. Published in Consumer Communications and Networking Conference, 2005. CCNC. 2005 Second IEEE, Pages 257 - 260, 3-6 Jan. 2005.
- [13] HDMI <http://www.hdmi.org/learningcenter/publications.aspx>
- [14] HBOGo <http://www.hbogo.com/>
- [15] Lorenzo Keller, Anh Le and Blerim Cici. MicroCast: cooperative video streaming on smartphones. Published in MobiSys '12 Proceedings of the 10th international conference on Mobile systems, applications, and services, Page: 57 - 70, 25 June 2012.
- [16] Netflix <http://www.netflix.com/>
- [17] Streaming media content to TV <http://www.informatandm.com/wp-content/uploads/2012/02/ITM-CES-Connected-TV-White-Paper.pdf>
- [18] Smartphone Users Worldwide Will Total 1.75 Billion in 2014 <http://www.emarketer.com/Article/Smartphone-Users-Worldwide-Will-Total-175-Billion-2014/1010536>
- [19] Smartphone OS Market Share, Q3 2014 <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>
- [20] Sony <http://www.sony.net/SonyInfo/>
- [21] Stauffer, M.: Connectivity solutions for Smart TVs. Published in Consumer Electronics - Berlin (ICCE-Berlin), 2012 IEEE International Conference. Pages 245 - 249, 3-5 Sept. 2012.
- [22] Ti-Hsin Yu, Shou-Chih Lo: A Remote Control and Media Sharing System Based on DLNA/UPnP Technology for Smart Home. Multimedia and Ubiquitous Engineering. Publisher Springer Netherlands. Pages pp 329-335, May 2013.
- [23] UPnP <http://www.upnp.org/>
- [24] Vbio <https://play.google.com/store/apps/details?id=com.stream.priyankmandalia.vbio>
- [25] Wei-Shun Liao, Yen-Ju Huang and Chih-Lin Hu. Mobile Media Content Sharing in UPnP-Based Home Network Environment. Published in Applications and the Internet Workshops, 2007. SAINT Workshops 2007. International Symposium, Page: 52, Jan. 2007.
- [26] Venkitaraman, N. Wide-Area Media Sharing with UPnP/DLNA. Published in Consumer Communication and Networking Conference, 2008. CCNC 2008. 5th IEEE. Pages 249 - 298, 10 - 12 Jan. 2008.
- [27] Wi-Fi Alliance. Wi-Fi. <http://www.wi-fi.org/>
- [28] YouTube <https://www.youtube.com/>