Peng Liu

Developing a Solution for Multimedia Home Networking

School of Electrical Engineering

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 25.5.2014

Thesis supervisor:

Prof. Raimo A. Kantola

Thesis advisor:

D.Sc. (Tech.) Mikko Valimaki

Aalto University
School of Electrical Engineering

AALTO UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING

ABSTRACT OF THE
MASTER'S THESIS

Author: Peng Liu

Title: Developing a Solution for Multimedia Home Networking

Date: 25.5.2014 Language: English Number of pages: 5+45

Department of Communications and Networking

Professorship: Networking Technology Code: S-38

Supervisor: Prof. Raimo A. Kantola

Advisor: D.Sc. (Tech.) Mikko Valimaki

This thesis analyzes the solutions to home multimedia networking, compares nowadays popular home networking solutions, especially for the Digital Living Network Alliance standard. By comparing different features and implementations, our team developed a suitable mobile solution for multimedia home networking which takes advantage of both AirPlay, DIAL and DLNA protocols on Android platform.
This study provides an overview of popular streaming technologies, AirPlay, DLNA, Miracast and Chromecast. By analyzing the features and capabilities of those streaming technologies, a universal solution is proposed for multimedia home networking by supporting multiple protocols and building bridge to different platforms.
I tested different multimedia solutions and we implemented a mobile Application for home networking on Android. The architecture, features and analyze methodology are discussed in the paper. We also investigate how to bridge home networking and Internet resources, an online channel proxy is made in our app "Streambels" to stream online channels like YouTube. Finally we got a published Android application for Tuxera Inc. The application is already published in Google Play Store for 6 months, which generate a statistic report of how users use our app, so there is a short description of the user behavior, and how we could improve user experience by analyzing these data.
At the end of the paper there is a discussion on the possible further development and the future of Home network.

Keywords: Home network, Multimedia, HTTP Streaming,
UPnP, DLNA, Miracast, AirPlay

Preface

This document is my master's thesis of *Communications and Engineering Networking* at Aalto University. All research and development of this thesis was conducted at Tuxera Inc. in Helsinki from January 2013 to June 2014. Tuxera is a high-tech startup that develop kernel-level file system and multimedia solutions for leading software, hardware and electronics companies. Duration this project I worked together with my colleagues at Tuxera, I started to work on DLNA project for the first few months during which period I learned DLNA architecture and made a research about Digital Media Server solutions. After that I worked on an Android project to develop a universal solution for multimedia home networking.

Acknowledgements

First of all, I would like to thank the Streambels team at Tuxera, whom I worked together throughout the project. I would like to thank Karthik Ramakrishna, our lead developer. Every week he helped solving out problems with the project, no matter the question is theoretical or technical, he always answered my questions. As our project manager, Oscar Santolalla helped us with organisational problems we encountered and taught us to look at things from a end user perspective as well. Sakari Tanskanen, our mobile developer helped us by integrating Chromecast and FireTV support to Stremabels. Nadir Javed, our quality assurance engineer helped us with the quality management and testing of potential bugs before releasing the product to end users. Karolina Mosiadz, our PR helped to listen to user's ton's of feedback everyday and provide unique insights in improving Streambels. Hien Le, our UX designer helped us to develop a very handy user interface. And special thanks to Mikko Valimaki and Szabolcs Szakacsits who lead the company and gave me the opportunity to participate in this great project. Without them, I would not have been able to finish this report.

I thank my university supervisor Raimo Kantola, who helped me to develop a good thesis topic based on my project and helped me with initial problem description. I got great support from him with his critique and useful advice, especially during the middle and final peroid, when I wrote the report.

Finally, I thank everybody who supported me during my graduation work, especially my family, friends and house-mates.

Otaniemi, 25.5.2014

Peng Liu

Contents

Symbols and abbreviations

Symbols

Opetators

Abbreviations

DLNA	Digital Living Network Alliance
DMC	Digital Media Controller
DRM	Digital Rights Management
DMS	Digital Media Server
DMR	Digital Media Renderer
HTTP	Hyper Text Transfer Protocol
RTSP	Real Time Streaming Protocol
SDP	Session Description Protocol
UPnP	Universal Plug and Play
DIAL	DIscovery And Launch
AP	Access Point
RAOP	Remote Audio Output Protocol
GENA	General Event Notification Architecture
SOAP	Simple Object Access protocol
DRM	Digital Rights Management
DIS	DRM Interoperability Solutions
REST	Representational State Transfer

1 Introduction

1.1 Home networking

People's lives are digitalized, home multimedia devices such as digital TV, smart phone, digital camera, tablet, PC, laptop and NAS (Network Attached Storage) are all equipped with great processing power and mass storage to handle multimedia information which records our daily life. On the other hand, the deployment of networking is growing fast. According to a research [1] done in 2011, in the Western industrialized world, networking is rapidly being adopted in the home; for example, in the U.S. in 2009, approximately 63% of homes had a broadband connection, and over 50% had a "home network", which is defined as multiple computers sharing a broadband connection via either a wired or wireless network within the home. In a typical home scenario, most of these devices are connected to a local network such as a Wi-Fi hot spot, in order to allow music, pictures, videos and other content to be ported across different devices.

1.2 Problem description

While the adoption of home networks has steadily increased since the late 1990s and early 2000s, this growth also reflects deep problems and limitations [1]. For example, the usability of home-networking technologies is a key impediment to the adoption of new applications in the home. The network usability problems run deep because the technology was originally developed for research labs and enterprise networks and does not account for the unique characteristics of the home: lack of professional administrators, deep heterogeneity, and expectations of privacy.

Among all the challenges, the problem of connecting all media devices in home networking and make them work together is getting increasingly interesting because of the rapid growth of consuming electronic market. Although there are several widely used multimedia-streaming solutions in the market, these standards are not compatible with each other. Furthermore, even devices using same standard are not always compatible with each other due to different implementation approach. This has caused great inconvenience to end users.

DLNA, AirPlay, Chromecast and Miracast are four major multimedia home network digital living solutions. AirPlay is only used between Apple products; it provides various features, including tunes play for music, AirPlay for video and photos and screen mirroring. Miracast (previously called Wi-Fi display) is proposed by Wi-Fi Alliance, and in recent years it gets more and more popular, from version 4.2.2, Android has officially added support of Miracast. On the other hand, DLNA is most widely deployed solution so far, with 2.2 billion installations. It was proposed by several industrial leading electronic manufacturers and network operators like AT&T, Broadcom, Cisco, Google, Huawei, Intel, LG Electronics, Microsoft, Nokia, Panasonic, Samsung, Sony and Verizon.

As a result of the collision between technical design choices and fundamental aspects of the human condition, these standards which are proposed by different device manufactures are not compatible with each other. It usually happens that end users have several multimedia-devices that are using different protocols; sharing media between those devices is challenging and occasionally impossible.

Therefore, there is a necessity to study and compare those multimedia-streaming technologies and develop a more easy-to-use multimedia home networking solution for modern advanced home networking.

1.3 Document overview

The overall popular home networking standards and solutions will be described in chapter 2. After a short comparison of these solutions, in chapter 3, we describe a more universal solution for multimedia home networking and its implementation. Chapter 4 give some statistics from the Google Store during the past few months, and we have a user study according to the feedback we got. In Chapter 5, a discussion is given to the further development and the future of home networking.

2 Background

Home networking has been a hot topic for quite a few years. Thanks to the rapid development of electronics and computer science, home networking devices are getting more affordable and more powerful that people would have at least several multimedia devices that can be connected to network.

In this chapter, we give a overview of current solutions to home networking, compare each solution and describe the problem to solve.

2.1 Overview

Early research [2] [3] [4] in home networking area has been made mainly on how to implement home networking infrastructures, like the cable connection, wireless connection, optical connection etc. So far, it turns out that IEEE 802.11 protocol stack is most successful and most widely deployed home networking infrastructure.

Nowadays, a typical scenario is that a wireless router connected to Ethernet cable or optical cable from network operator, while other devices connect to the same local network created by the wireless router. A wireless Access Point(AP) is using 802.11 b/g/n/ac protocol, utilize 2.4 GHz or 5 GHz frequency channels and provide a 100+ Mbps network connection. This bandwidth is sufficient enough for transmitting High Definition (1080p) videos.

In terms of network and application layer technologies, different device manufactures choose their preferred multimedia-sharing protocols. The development of multimedia-sharing protocols has also experienced a long research period.

Since late 1990s, UPnP protocol has been developed for home networking usage. At that time, XML is popular and widely used by different network applications, in such a background, UPnP is also designed to make full use of XML. UPnP is independent of media and device, and it is running on TCP/IP stack, so it can be easily applied to modern network infrastructures.

In June 2003, Sony and several leading consumer electronic manufacturer companies established the Digital Living Network Alliance (DLNA), a nonprofit collaborative trade association. The DLNA standard is based on widely used UPnP protocol but it added some restrictions on media formats and some compatibility requirements. Device hardware and software can be certified by DLNA organization to prove that it should work with other devices that also passed this certification.

In 2010, Apple quitted DLNA and developed its own multimedia home networking solution, called AirPlay. By adding screen mirroring, authentication and Remote Audio Output Protocol(RAOP) music streaming, Apple tried to make a more advanced home network sharing system and create a unique user experience between Apple products. It indeed attracts people's interest, and the user experience of Apple's product is much better than other similar products in the market. And it has been developed over time to become one of the most popular streaming solution accepted by people today.

Two years later, Wi-Fi alliance released its Miracast technology, and participated in pushing new standard in wireless home networking. The Miracast uses

Wi-Fi direct technology, so that it doesn't require a wireless local network, instead, a peer-to-peer connection is created between the sharing and receiving devices. After its releasing, some major software and hardware companies soon accept the new standard. Google, for example integrated Miracast support into its Android operating system, and provide screen-mirroring feature to other Miracast receivers.

The competition is far from the end, in 2013, Google released a 35-dollar dongle, using its Chromecast protocol, Chromecast makes it possible to watch YouTube and Netflix video directly on TV with such a device. Laptop and mobile devices with official YouTube App or Chrome browser can control the dongle through the local network in home. The home networking has been pushed to cloud, since YouTube and Netflix content are directly downloaded from Internet, mobile devices just act as a controller of picking interested contents.

Almost at the same time, September 2013, Spotify, a startup music service company has also taken part in making its own home networking solution, called Spotify Connect, it provides an interface for speakers in home access its huge music database, and directly browse and stream using mobile application. Home networking has again been pushed towards cloud and Internet services.

Since so many companies would like to develop their own devices and even their own protocols. The market is a bit mess, devices are not compatible with other devices, users have to buy new device to have services from those companies like Netflix and Spotify. There is great need of creating some solution that can connect those devices in home and make them work together friendly.

Streambels project is just created for this need, we try to fill the gap between different protocols and connect devices in home networking environment.

2.2 Available protocols

2.2.1 UPnP [1]

Universal Plug and Play (UPnP) is a series of networking protocols defined to work together to seamlessly discover the presence of all devices in the network and establish functional network services for data sharing, communications, and entertainment between these discovered devices.

In most UPnP scenarios, a control point controls the operation of one or more UPnP devices. The interaction usually occurs in isolation between control point and each device. It is control point's responsibility to coordinate the operation of each devices, while the individual devices do not interact directly with each other.

The UPnP device architecture [5] includes seven parts:

1. Addressing
   UPnP devices have a DHCP client and searches for a DHCP server when connected to the network. An UPnP device firstly scans for the DHCP server and then requests an IP address when the DHCP server is found. Otherwise, if there is no response from DHCP server, the device uses automatic IP address.

[1] Universal Plug and Play

It randomly chooses an address in the 169.254/16 random and tests it using ARP probe to determine if it is already used. The same procedure repeats until an unused address is found. After first IP address is set, it then will periodically check for DHCP server, when a server responds, it will use the assigned addresses and stop using the address generated by Auto-IP after a period of parallel use to complete interactions in progress. If there is DNS server in the network, it can also use domain names instead of the numerical IP address.

2. Discovery
   UPnP devices advertise their services to network using UPnP discover protocol, which is based on Simple Service Discovery Protocol (SSDP). An UPnP control point searches the existence of UPnP devices in the network using SSDP. The discover message contains a few specific attributes of a device and its services, these attributes include device type, unique identifier and a pointer to more detailed information. The device send multicast several NOTIFY message to a pre defined address and port to advertise its availability. A control point will listen to this standard multicast address and get notifications when new devices are available in the network. An advertisement message has its lifetime, so devices in the network will periodically send NOTIFY message before the previous message expires. When the device or servers becomes unavailable or shut down intentionally, previous advertisements are canceled by sending cancellation messages, otherwise, the advertisements will eventually expire on its own. Control point can search for devices actively by multicasting an SSDP Search message. Other devices in the network will respond to the search message by unicasting directly to the requesting control point.

3. Description
   The discovery message contains URL to the description information. A control point can send HTTP GET request to the URL to get detailed UPnP description of the device. The description includes a device description and several service descriptions.

   A device description includes vender related information such as model name, serial number and manufacture name. A device may have many services, for each service, the device description lists the service type, name and URL to the detailed service description, control and eventing. A device description may also include embedded devices and a URL to a presentation page.

   A service description includes a list of actions that servers can accept, arguments of each action, and a list of state variables. The state variables reflect the device's status during runtime.

   The description is in XML syntax and is based on standard UPnP device template or service template, which is defined by UPnP forum. The template language is written in XML syntax and is derived from an XML schema language, so it is machine-readable and automated tools can parse check easily.

By using description, vender has the flexibility to extend services, embed other devices and include additional UPnP services, actions or state variables. The control point can be aware of these added features by retrieving device's description.

4. Control
   A control point can ask services in a device to invoke actions by sending control messages. The control process is a form of remote procedure call: a control point sends the action to device's service, and when the action has completed on the remote device, the service returns the results or errors of the action.

   The control messages are expressed in XML using the Simple Object Access Protocol (SOAP) and sent by HTTP request. The action results are also received by HTTP responses. The action may cause state variable change and those changes will be reflected in the eventing messages.

5. Eventing
   UPnP service description defines a list of state variables, which updates at runtime. The service publishes those changed state variables in the form of event messages, and a control point can subscribe to this information.

   A control point can subscribe the event notification by sending a subscription message to the subscription URL specified in the device description and provides a URL to receive the event messages.

   Since there is no mechanism to subscribe to a subset of evented state variables, all subscribed control points will receive all event messages regardless of why the state variable changed.

   When the subscription is accepted, the device gives a unique identifier for the subscription and the duration of the subscription. The device will also send an initialize event message, which includes the names and current values for all evented variables.

   The event messages are General Event Notification Architecture(GENA) NOTIFY messages, sent using HTTP with a XML body, which specifies the names of one or more state variables and new values of those variables. Once the state variable changed, the event message is sent immediately sent to control point, thus the control point can get timely information and could display a responsive user interface. The control point then send HTTP OK message to acknowledge device that the event message is received. The event message also contains a sequence number that allows the detection of possible lost or disordered messages.

   The subscription must be renewed periodically to extend lifetime and keep it active. The renew message which contains the subscription identifier is sent to the same URL in the subscription message. When the subscription expires, the device will stop sending eventing message to the control point, and any attempt to renew the expired subscription is rejected.

A subscription can be canceled by sending an appropriate message to the subscription URL.

6. Presentation
   Many UPnP devices provide a presentation URL to "web" interface for users. User can access the presentation URL by a standard web browser. Control point send an HTTP GET request to the presentation URL to get a HTML page from the device, and display the page in a web browser, thus provide a more user-friendly interface for control and view the status of device.

   The presentation page is totally specified by the device vender, but it must be an HTML page. UPnP architecture does not define the details of presentation page, but it should be user friendly and have some basic functionality.

So far I described the general device architecture of UPnP devices, to emphasis the topic of this paper, we focus on the multimedia device types in home networking. In particular audio and video use cases. The AV control point interacts with two or more UPnP devices acting as source and sink, respectively. While coordinated by the AV control point, the devices themselves interact with each other using a non-UPnP communication protocol. The control point configures the devices as needed, triggers the flow of content, then gets out of the way.

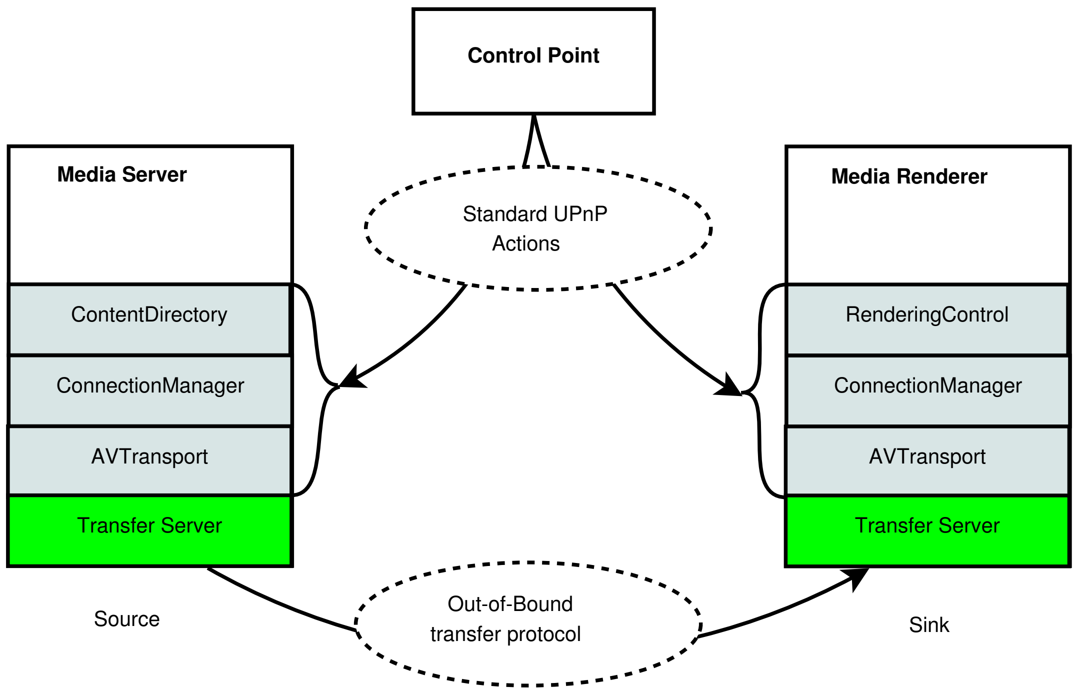

Figure 1: UPnP A/V playback architecture

1. Media Server
   The media server is used to locate content available in the home network. It's primary purpose is to allow control points to enumerate (browse or search)

content items that are available for the user to render. The media server contains a ContentDirectory Service(CDS), a ConnectionManager Service(CM) , and an optional AVTransport Service(AVT) which depends on the supported transfer protocols. Some media servers are capable of transferring multiple content items at the same time.

The ContentDirectory service is used by control point to enumerate the content on the server. The primary action is ContentDirectroy::Browse(). After invoking this action, the control point can obtain detailed information of each item that the server can provide. This detailed information includes name, artist, date created, size and also the transfer protocols and data formats that supported by the for particular item. By parsing this detailed information, control point is able to distinguish whetehr the item can be rendered by the given media renderer.

The ConnectionManager service is used to manage the connections between a control point and device. The primary action is ConnectionManager:: PrepareForConnection(), which is invoked by the control point to the server and make it prepare itself for an upcoming transfer. This action will return the instanceID of an AVTransport service that will be used later to control the flow of content, for example, to stop, pause, seek the stream etc. The instanceID is used to distinguish multiple instances of the AVTransport service, each is associated with a particular connection to renderer, thus it enables multiple renderer support at the same time. When the control point need to disconnect the connection, it will invoke the media server's ConnectionManager::ConnectionComplete() action to release the connection. When the ConnectionManager::PrepareForConnection() action is not implemented, the control point is only able to support single renderer at a time. In this case 0 will be used as InstanceID.

The AVTransport service is used by control point to control the playback of the content. Operations like Stop, Pause, Seek are supported by this service. This service is not mandatory, media server can choose to implement this feature depending on the supported transfer protocols and data formats. If this service is supported, the InstanceID included in each AVTransport action is used to distinguish multiple instances of the service. New instances of the AVTransport service can be created by ConnectionManager's ConnectionManager::PrepareForConnection() action, and new InstanceID is allocated to each new service instance.

2. Media Renderer
The media renderer is used to render the content obtained from home networking. Its main feature is that it can be discovered by control point and allow it to control how the content is rendered, for example, brightness, contrast, volume, mute, etc. The control of flow of the content like stop, pause, seek can also be supported depending on the transfer protocol used. Media renderer includes three services: RenderingControl service, ConnectionManager service

and an optional AVTransport service. Sometimes the rendering control and AVTransport services contain multiple independent instances so that the devices are capable of handling multiple content items at the same time. Those multiple instances are distinguished by a unique InstanceID.

RenderingControl service is used by control point to control how the renderer renders the incoming content. Characteristics like brightness, contrast, volume, mute etc, can be controlled by this service. RenderingControl service supports multiple, dynamic instances, which allows a renderer to mix one or more items together, such as Picture-in-Picture window on a TV or audio mixer device. Multiple connections can be distinguished by unique InstanceID.

ConnectionManager service is used to manage connections associated with a device, the primary action is ConnectionManager::GetProtocolInfo() action. Control point can invoke this action to enumerate the transfer protocols and data formats supported by media renderer. By comparing this information with protocol information got from media server, the control point is able to predetermine if a media server is capable of rendering a specific item from media server. Optionally, media renderer may also implement ConnectionManager::PrepareForConnection() action to prepare itself for an upcoming transfer, it can also assign a unique ConnectionID that can be used by 3rd party control point to obtain information about the connections that media renderer is using. In addition, depending on the transfer protocol and data format used, this action may also return a unique AVTransport InstanceID that control point can use to control the flow of content, such as stop, pause, seek, etc.

AVTransport service is used to control the flow of streamed content. Actions like play, stop, pause and seek can be controlled depending on the transfer protocol and supported data formats. AVTransport service can also support multiple logical instances to handle multiple simultaneous content items. The AVTransport InstanceID which is used to distinguish service instances can be allocated by ConnectionManager::PrepareForConnection().

3. Control Point
   Control Point is used to communicate between media server and media renderer. It also provide user interface to users. A control point does not implement UPnP services so it is not visible as a device on the network. Usually control point invokes media server or media renderer's services in order to complete the desired behavior coordinately.

   The user control point can be used in different scenarios, usually it can perform the following functions:

   - Discover A/V devices
   - Locate desired media content
   - Get renderer's supported protocol and formats

- Compare and match protocols and formats between media server and media renderer
- Configure media server and media renderer
- Select desired content
- Start content transfer between media server and media renderer
- Adjust rendering characteristics
- Select next content in the list
- Cleanup media server and media renderer

As described above, three basic functional entities are defined in the UPnP AV architecture[6]: Media Server, Media Renderer and Control Point. A physical device can consist of a combination of any of these functional entities. A typical example is DLNA Media player is a combination of a Control Point and a Media renderer.

A simplified UPnP Audio Video 3-box model [7] can be seen as below:

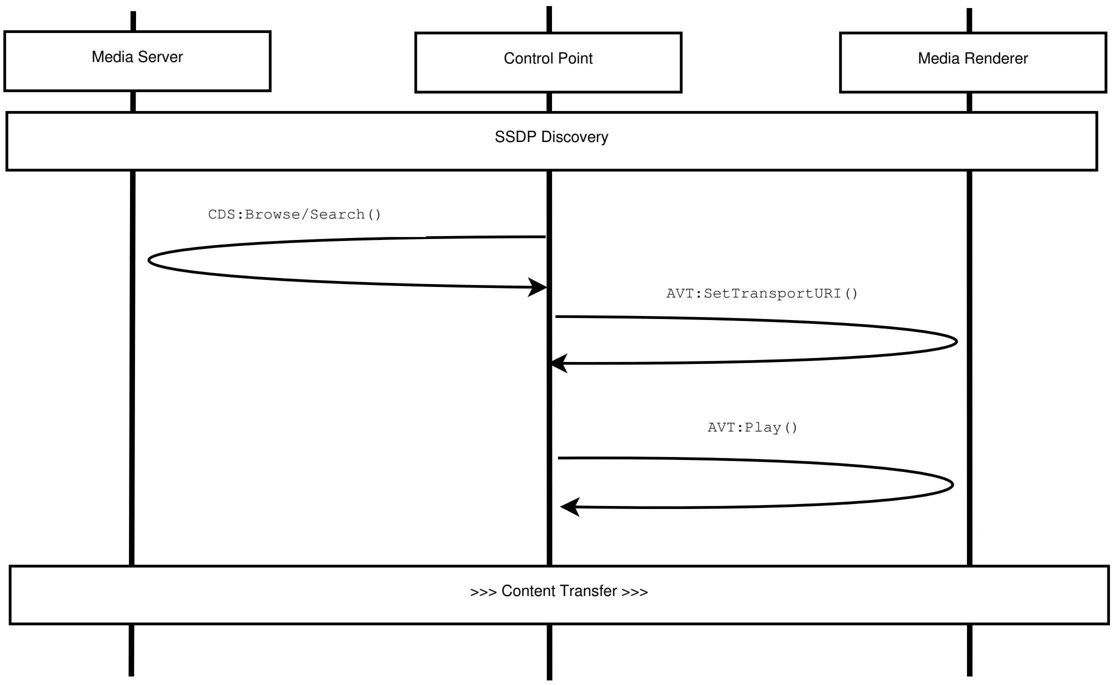

Figure 2: Typical UPnP AV use scenario

The first thing in UPnP network communication is Simple Service Discovery Protocol(SSDP)-based device discovery. A SSDP multicast message is send when a new device is added in the network. A control point will listen to these multicast messages. When the control point receives the SSDP message, the control point will send request for the device's description and services using the location provided in the SSDP discovery message. Later the control point can invoke the services action command using Simple Object Access protocol (SOAP).

In media sharing scenario, the control point will browse the information about the Content Directory Service (CDS) provided by the Media Server. A browse/Search action can be invoked to navigate through the content stored in the Media Server device. After control point selected the media content from Media Server, a Media Renderer AVTransport::SetAVTransportURI will be sent by control point to Media Renderer. Finally, the Play command is invoked by control point to Media Renderer, and the transfer begin afterwards. The media streaming happens directly between Media Server and Media Renderer using HTTP, RTP or other streaming protocols.

The media playback control action can also be invoked by control point. Methods supported include volume control, seek, pause etc.

2.2.2 DLNA[2]

DLNA is relatively a old standard in industry compared to other home networking solutions, it is mainly based on UPnP Audio/Video architecture, which is discussed in 2.2.1, so it is widely used by many manufactures, the newer home networking solution is also influenced by DLNA and followed similar technologies. In this paper we focus on the DLNA and UPnP standard architecture so we can have an overview of how a home networking solution looks like.

An overall DLNA architecture [8] can be described below:

1. Architectures and Protocols

Table 1: Key Technology Ingredients

Functional Components	Technology Ingredients
Connectivity	Ethernet, 802.11 (including Wi-Fi Direct) MoCA, HPNA and Bluetooth
Networking	IPv4 Suite
Device Discovery and Control	UPnP* Device Architecture v1.0 2.2.1
Media Management and Control	UPnP AV and UPnP Printer:1 2.2.1
Media Formats	Required and Optional Format Profiles
Media Transport	Media Transport
Remote User Interfaces	CEA-2014-A

The DLNA architecture is built upon UPnP protocol, which is discussed in in 2.2.1. In networking layer it uses IPv4 suite, on top on that, UPnP device architecture and UPnP AV architecture is used to control and manage media devices. DLNA guideline also addresses the media format compatibility and media transport interoperability issues to make devices interoperable with each other.

[2]Digital Living Network Alliance

2. Media Format Profiles
   DLNA defines the media formats used by DLNA home networking standard. There are three types of media in DLNA: music, video and photo.

   (a) Music
   Minimal requirement is LPCM format, which is PCM raw data, this format is not compressed so it does not require heavy CPU usage, but on the other hand, the bandwidth consumption is considerably bigger than other formats.

   MP3 is most popular music format in music category, it is compressed format, so it will require some CPU power to encoding or decoding, but on the other hand, the bandwidth consumption is less and suitable for low bandwidth networking.

   AAC is kind another kind of compressed audio format and it becomes popular since it is Apple's iTunes's default media format. It has similar characteristics to MP3.

   (b) Photo
   The minimal requirement in DLNA guideline is JPEG format, and sometimes the only suggested format due to its proven quality and compress ratio.

   (c) Video
   The minimal requirement in DLNA guideline is MP4 format, but the detailed audio and video codecs are also specified in DLNA media format guidelines.

   In a device-to-device use scenario, the media server may store tons of different formatted media. The communication between two devices should follow the same encoding mechanism. Normally the media server takes the responsibility to transcode the media to certain format defined by DLNA media format profile guideline.

3. Link Protection
   DLNA Link Protection is defined as the protection of a content stream between two devices on a DLNA network from illegitimate observation or interception.

   Content protection is an important mechanism for ensuring that commercial content is protected from piracy and illegitimate redistribution. Link Protection is a technique that enables distribution of protected commercial content on a home network, thus resulting in greater consumer flexibility while still preserving the rights of copyright holders and content providers.

4. Digital rights management (DRM) Interoperability Solutions (DIS)
   DIS is intended to be used to enable the secure transfer and use of protected commercial content among different implementations on network media devices. The content could be protected by different content protection technologies, which are described as DRMs in short.

5. Device Profiles
   A Device Profile is a collection of DLNA capabilities and features within a DLNA device. For a device to be compliant with a Device Profile, it has to conform to all of the guidelines listed for that Device Profile.

   In practice, Device Profiles reference existing optional or recommended DLNA guidelines, that enable certain features, and makes those DLNA guidelines mandatory within the context of a Device Profile. A Device Profile can also provide some additional guidelines that complement or modify existing DLNA guidelines for a feature.

   A particular type of DLNA Device Profile is the Commercial Video Profile (CVP). A CVP Device Profile is an extension of the DLNA guidelines that will allow content from service providers and multichannel video programming distributers to be distributed on the DLNA network. DLNA Commercial Video Profiles (CVPs) are defined as Device Profiles that consistently enable commercial content that enters the home network through a gateway device via an interface to a commercial content service provider. Since different regions of the world have different requirements for commercial content, there are multiple CVPs defined.

2.2.3 AirPlay

AirPlay is Apple Inc's home networking solution. It is a family of protocols to display different types of media content on Apple TV from other iOS devices. The AirPlay support displaying photos and slideshows from iOS device, streaming audio from iOS device or iTunes, displaying videos from iOS device and show the whole screen on Apple TV, called AirPlay Mirroring.

AirPlay's specification is not open to public, but on Internet there are hackers reverse engineered the protocol stack and made unofficial specification [9]. The specification include 6 parts, which are described below:

1. Service Discovery
   The service discovery of AirPlay roots from the IETF Zeroconf Working Group, which aims at improving ease-of-use (Zero Configuration) of network. The Zeroconf working group tries to make it possible to take two devices in the network to communicate usefully using IP, without needing specialist to manually configure the network.

   AirPlay's service discovery is based on Multicast DNS [10], which fulfills Zeroconf requirement. Multicast DNS is a way of using familiar DNS programming interfaces, packet formats and operating semantics, in a small network where no conventional DNS server has been installed. The requirements for Zeroconf name resolution could be met by designing an entirely new protocol, it is better to provide this functionality by making minimal changes to the current standard DNS protocol. By using multicast DNS, most current applications need no changes at all to work correctly using mDNS in a Zeroconf network, engineers do not have to learn an entirely new protocol, and current network packet capture tools can already decode and display DNS packets, so they do not have to be updated to understand new packet formats.

An AirPlay device such as the Apple TV publishes two services. The first one is RAOP (Remote Audio Output Protocol), used for audio streaming, and the other one is the AirPlay service, for photo and video content.

The AirPlay server is a HTTP server (RFC 2616). Two connections are made to this server, the second one being used as a reverse HTTP connection. This allows a client to receive asynchronous events, such as playback status changes, from a server.

2. Video streaming
   The video streaming uses typical HTTP streaming technology, the controller set the streaming URL to Apple TV or other AirPlay receiver. While the URL is set, Apple TV start to download video from the server using the URL and starts playing while buffered enough data. The control messages can be seen in table 2.

   Note that the Apple TV does not support Video Volume Control.

Table 2: AirPlay Video Control HTTP requests

Method	Request	Description
GET	/server-info	Fetch general informations about the AirPlay server
POST	/play	Start video playback
POST	/scrub	Seek at an arbitrary location in the video
POST	/rate	Change the playback rate,0 is paused, 1 is normal
POST	/stop	Stop playback
GET	/scrub	Retrieve the current playback position
GET	/playback-info	Retrieve playback informations like position, duration...
PUT	/setProperty	Set playback property
GET	/getProperty	Get playback property

3. Photo streaming
   The image streaming uses HTTP put message to send image raw data to Apple TV or other devices, when the whole image is received, the image is then rendered on screen. The AirPlay also supports slide show, the control message can be seen in table 3.

4. Music streaming
   AirPlay music streaming is a bit different from video and image streaming, the technology used is RTSP streaming, it is more "push like" protocol, the RTSP streaming server push UDP packets to receiver. However the RTSP

Table 3: AirPlay Photo Control HTTP requests

Method	Request	Description
GET	/slideshow-features	Fetch the list of available transitions for slideshows
PUT	/photo	Send a JPEG picture to the server
PUT	/slideshows/1	Start or stop a slideshow session
POST	/stop	Stop a photo or slideshow session

protocol Apple used is not standard RTSP, it uses its own implementation, called RAOP (Remote Audio Output Protocol). The control message can be seen in table 4.

Table 4: AirPlay Audio Control RTSP requests

RTSP request	Description
OPTIONS	Ask the RTSP server for its supported methods
ANNOUNCE	Tell the RTSP server about stream properties using SDP
SETUP	Initialize a record session
RECORD	Start the audio streaming
FLUSH	Stop the streaming
TEARDOWN	End the RTSP session

5. Screen mirroring
   AirPlay screen mirroring is achieved by transmitting H.264 encoded video stream over a TCP connection. The stream is packetized with a 128-byte header. The audio is sent using AirTunes protocol and AAC-ELD formatted. NTP(Network time protocol) is used for synchronization. The synchronization takes place on UDP port 7010(client) and 7011(server), using NTP protocol. The AirPlay server runs a NTP client. Request are sent to AirPlay client at 3 second intervals. The reference date for time stamp is the beginning of the mirroring session. The control message can be seen in table 5.

Table 5: AirPlay Mirroring Control HTTP requests

Method	Request	Description
GET	/stream.xml	Retrieve information about the server capabilities
POST	/stream	Start the live video transmission

6. Authentication
   An AirPlay server can require a password for displaying any content from the network. It is implemented using standard HTTP Digest Authentication (RFC 2617), over RTSP for AirTunes, and HTTP for everything else. Te digest realms and usernames accepted by Apple TV are described in table 6.

Table 6: AirPlay HTTP Digest Authentication

Service	Realm	Username
AirTunes	roap	iTunes
AirPlay	AirPlay	AirPlay

2.2.4 DIAL

Chromecast and FireTV use the DIAL [11] (DIscovery And Launch) protocol, co-developed by Netflix and YouTube, to search for available devices on a Wi-Fi network. Once a device is discovered, the protocol synchronizes information on how to connect to the device. The protocol is proposed by Google and Netflix so the YouTube and Netflix application has already build in with DIAL device discovery. The streaming part uses HTTP streaming, which means controller can directly set the streaming URL and the receiver will start downloading automatically.

The DIAL protocol has two components: DIAL Service Discovery and the DIAL Representational State Transfer (REST) Service. DIAL Service Discovery enables a DIAL client device to discover DIAL servers on its local network segment and obtain access to the DIAL REST Service on those devices. The DIAL REST Service enables a DIAL client to query, launch, and optionally stop applications on a DIAL server device.

1. DIAL Service Discovery
   The DIAL Service Discovery protocol is based on Simple Service Discovery Protocol (SSDP), which is defined as part of UPnP device architecture discussed in 2.2.1.

   A DIAL client will firstly send an M-Search request over UDP to the IPv4 multicast address 239.255.255.250 and UDP port 1900 including the Search Target header. After that the SSDP/UPnP server responds including a LOCATION header containing an absolute HTTP URL for the UPnP description of the root device. When receives the M-SEARCH response, the DIAL client will send an HTTP GET request to the URL received in the LOCATION header of the M-SEARCH response to get HTTP response containing the UPnP device description as a XML content. The overall flow of DIAL discovery is shown in figure 3.

2. DIAL REST Service
   The DIAL REST service allocates URLs for different resource applications

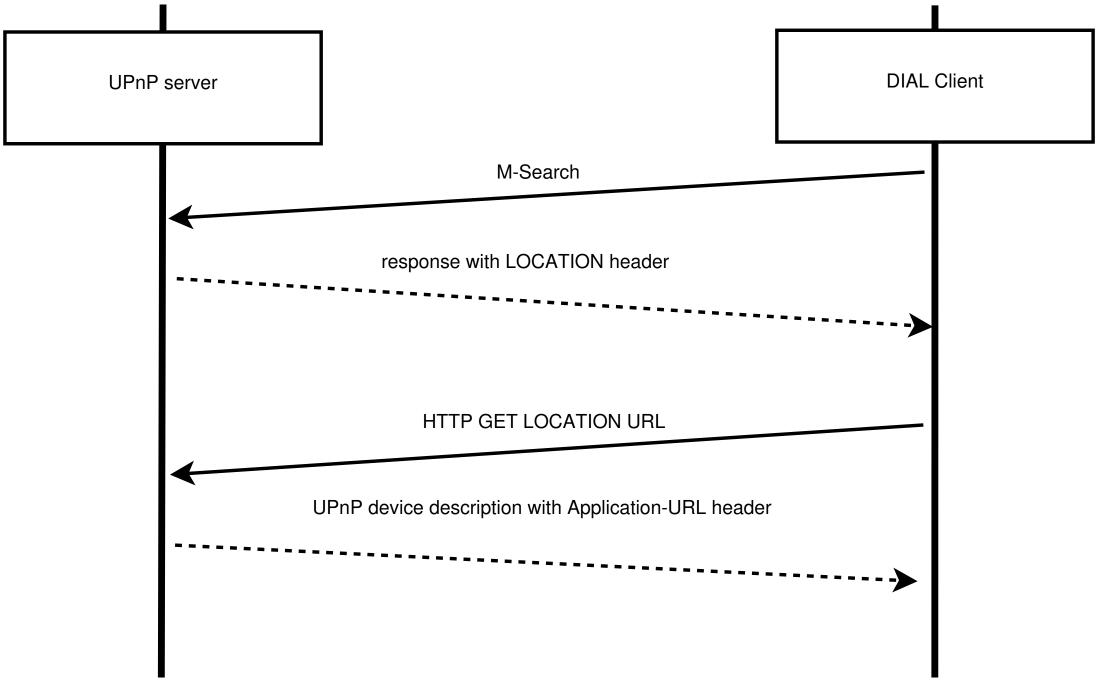

Figure 3: DIAL Discovery

such as YouTube and Netflix. Then the application can be controlled by issuing HTTP requests against the URL for that application. The Application resource URL is constructed by concatenating the DIAL REST service URL, a single slash charater ('/') and the application name. The application name must be registered in DIAL Registry to be used.

A DIAL client firstly send a HTTP GET request to the application resource URL, then the server will extract the application name and search for the application is installed or not, if the application is not recognized, the server will return 404 Not Found or trigger the installation of the specific application that is not currently installed, otherwise, the DIAL server return an HTTP response with 200 OK and a body contains MIME type in XML.

The client then sends a HTTP POST request to the application resource URL to launch the desired application. On receipt of a valid POST request, the DIAL server will extract the application name and run the application, and then send a HTTP response with LOCATION header to inform the absolute HTTP URL which identifies the running instance of the application.

The first-screen application can also send small amount of data to the DIAL server, and then DIAL server can communicate the information to DIAL clients. After the application is launched and communication is established, the DIAL client can communicate directly with the application. DIAL REST service overall flow chart can be shown as figure 4.

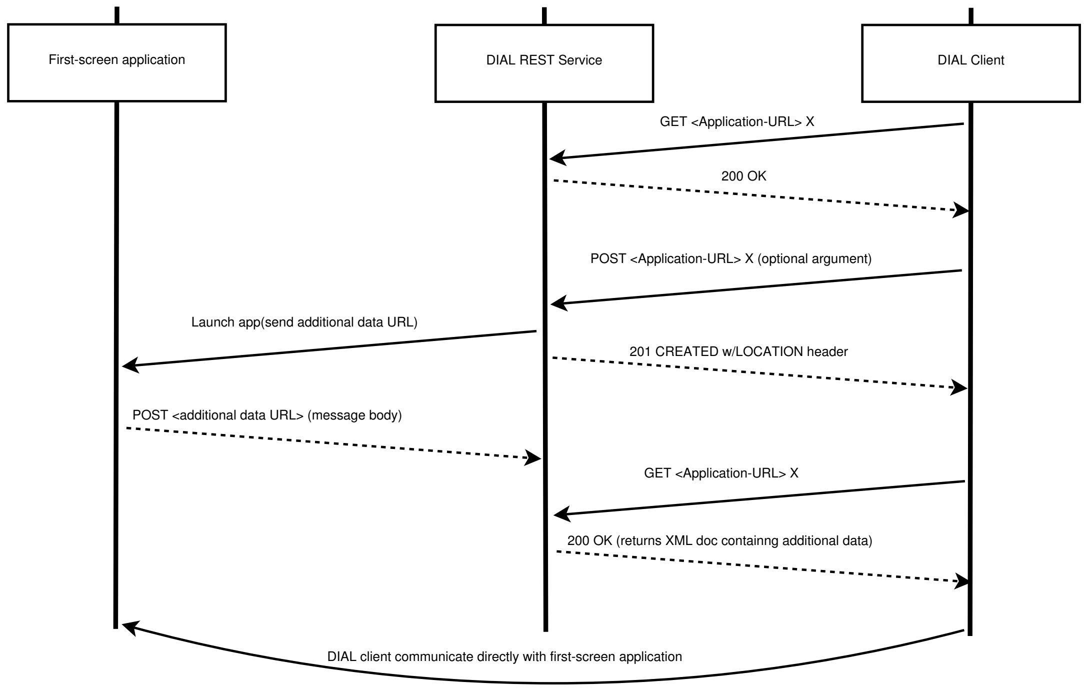

Figure 4: DIAL REST service: application launch

2.2.5 Miracast

Miracast [12] is quit different on technology perspective, instead of connecting on the same local network segment, a Wi-Fi peer to peer connection is created, so Miracast is not limited to pre-configured network infrastructure.

Miracast allows users to establish a direct Wi-Fi connection between two devices, eliminating the need for an existing network.

Miracast uses many of the building blocks that, over the years, have enriched the user experience and increased their trust in Wi-Fi, including Wi-Fi CERTIFIED n (improved throughput and coverage), Wi-Fi Direct (device-to- device connectivity), Wi-Fi Protected Access 2 (WPA2) (security), Wi-Fi Multimedia (WMM) (traffic management) and Wi-Fi Protected Setup. Some Miracast devices will also support Tunneled Direct Link Setup (TDLS), which allows them to connect via an infrastructure network. TDLS enables more efficient data transfer and the use of more advanced Wi-Fi capabilities than those supported by the legacy infrastructure network through which the devices are connected.

Miracast does not require a typical Wi-Fi infrastructure network, though many devices will take advantage of network connectivity to access content. Miracast connections are expected to be predominantly established between Wi-Fi devices connected with each other directly, without an AP acting as an intermediary. The direct link between devices is established either through Wi-Fi direct, a feature that all Miracast devices are required to support, or through TDLS, an option feature. When two devices connect with each other directly, one fulfills its role as the source(the transmitting device) and the other functions as a display(the device receiving and rendering the content to the user). Topologies supported by Miracast are shown in Figure 5.

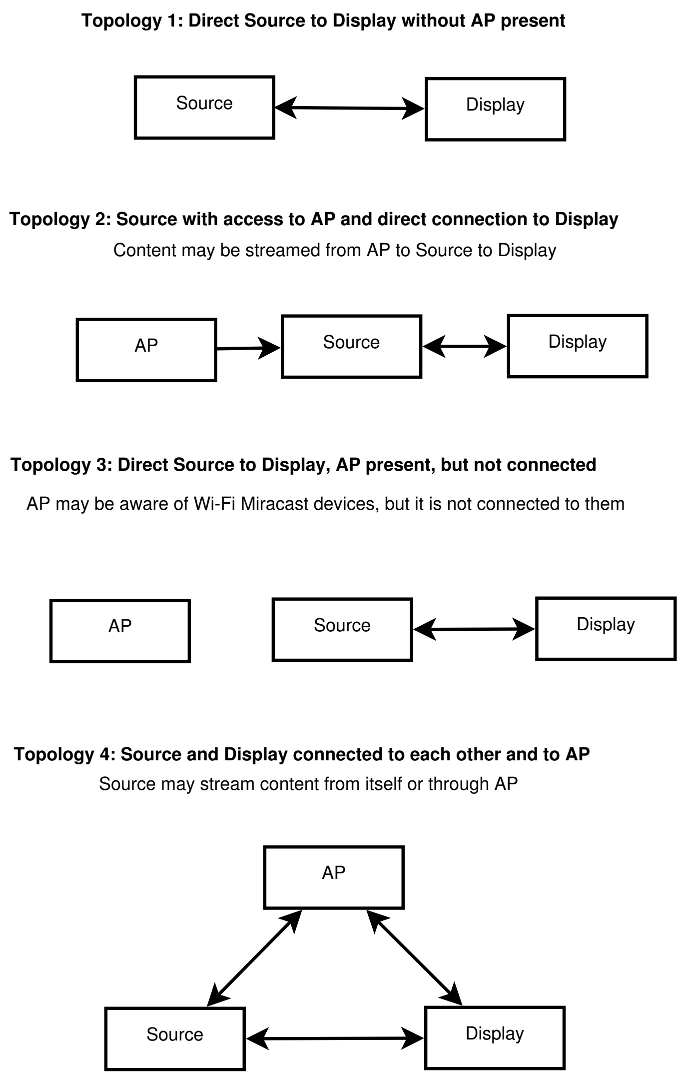

Figure 5: Miracast topologies

On technology perspective, Miracast is built upon many diffenet Wi-Fi technologies, which is shown in figure 6.

The technology component includes the following:

1. Connectivity
   Wi-Fi CERTIFIED n provides a transmission channel designed to support multimedia content.

2. Device-to-device connectivity
   Wi-Fi Direct allows devices to connect directly to each other, without the need for a Wi-Fi AP, and often requiring just the push of a button. TDLS allows devices that are associated to the same Wi-Fi network to establish a direct link with each other.

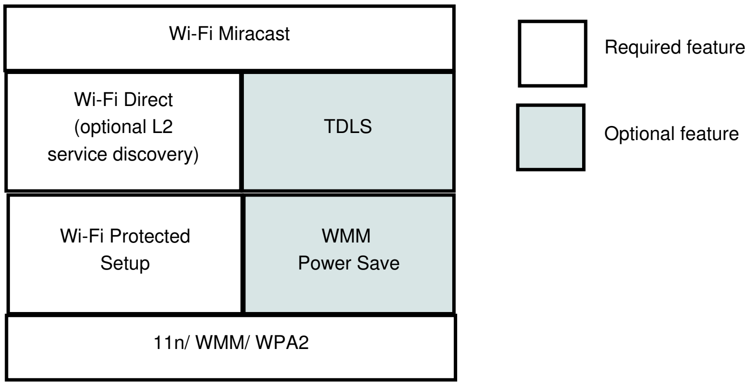

Figure 6: Miracast technology architecture

3. Security
   WPA2 makes the transport of multimedia content safe, protecting both the source and the display.

4. Quality of service (QoS)
   Wi-Fi Multimedia (WMM) gives real-time content, such as voice and audio, priority where appropriate over best-effort traffic, to support a good user experience.

5. Battery life
   WMM Power Save extends the battery life of mobile devices like smartphones or tablets by minimizing the time the device is actively connected to the AP during idle time. Power save mechanisms in Wi-Fi Direct provide similar benefits when connecting devices without an AP.

6. Ease of installation
   Wi-Fi Protected Setup helps users to automatically configure Wi-Fi networks, enable WPA2 security, and add new devices.

Miracast Session management

Miracast Session Management	
Device Discovery	Source and display devices discover each other prior to connection setup. The Device discovery mechanism is defined in the Wi-Fi Peer-to-Peer (P2P) Specification.
Service Discovery	Source and display devices discover each other's Miracast capabilities prior to connection setup. The Service discovery mechanism is defined in the Wi-Fi P2P specification.
Device selection	A remote device is selected for connection setup. User input and local policies may be used to decide which device is a display and which is a source.
Connection setup	Connection setup selects a method (Wi-Fi Direct or TDLS) to manage the connection. Wi-Fi Direct sets up a group owner and client to initiate a device-to-device link. A WPA2 single-hop link with selected devices is established. Upon the establishment of connectivity between the source and display devices, the display initiates a Transmission Control Protocol (TCP) connection, with a control port using Real-Time Streaming Protocol (RTSP) to create and manage the sessions between source and display devices.
Capability negotiation	Source and display devices determine the parameters for the Miracast session.
Content protection setup (optional)	If the devices support content protection and are streaming content requiring protection, session keys for link content protection are derived using High-bandwidth Digital Content Protection (HDCP) 2.0/2.1. HDCP session keys are established before the RTP session is initiated. This feature is designed to protect the digital rights of content owners and to encourage their efforts to make their content available.
Session establishment and streaming	Upon completion of capability negotiation, the source and display devices setup the Miracast session prior to streaming content. The audio and video content available on the source device is packetized using Moving Picture Experts Group 2 Transport Stream (MPEG2-TS) coding and encapsulated by Real-Time Protocol (RTP) User Datagram Protocol (UDP) and Internet Protocol (IP). Finally, IEEE 802.11 packetization enables the source device to send content to the display device.
User input back channel setup (optional)	A User Interface Back Channel (UIBC) for transmitting control and data information related to user interaction with the user interface is set up. User inputs at a display are packetized using a UIBC packet header and transported using Transmission Control Protocol/Internet Protocol (TCP/IP).

Miracast Session Management	
Payload control	When the payload transfer starts, devices may adapt transmission parameters on the basis of channel conditions and power consumption. Adaptation can be achieved by: Compression ratio change and macroblock skipping (using the H.264 standard); Frame skipping (if the display device supports this functionality, the source device may skip some of the frames to be transmitted according to the current resolution); Format change.
Display session teardown	Either the source or the display terminates the Miracast session.

2.2.6 Other protocols

Apart from all mentioned standards above, many other companies or associations also proposed their own protocols, for example SonosNet [13] which is based on peer to peer network, and Spotify Connect [14]. The standard war on home networking has never end.

2.3 Comparison of existing solutions

2.3.1 History

- DLNA is proposed by several leading consumer electronic manufactures based on UPnP technology, from early 2000s on, over 2.2 billion devices has shipped with DLNA solutions, making it possible to sharing audio and video seamlessly different smart devices. DLNA alliance had two annually meetings a year to discuss the marketing and developing related issues, making it a more and more accomplished standard.

- AirPlay, on the other hand, is proposed by Apple Inc. in 2010, before that actually Apple is a part of DLNA alliance, by proposing AirPlay, it enables more advanced features than DLNA, such as whole screen mirroring, RTP audio streaming, authentication etc.

- Miracast is a quite new technology, it is formerly known as Wi-Fi Display, and proposed in 2012 by Wi-Fi alliance. Different from AirPlay and DLNA, it is not based on home AP, but Wi-Fi direct instead. It provides a screen-mirroring feature just like Apple AirPlay Mirroring, and now it becomes quite popular among manufactures and software ventures. Google has launched its Android 4.2 with native support of Miracast, the latest Kitkat Android 4.4 has been certified to the Wi-Fi Alliance Wi-Fi Display Specification as Miracast compatible. There is a strong trend that this standard will soon be very popular in multi-screen sharing market.

- Chromecast or Google cast is another new trend in market, Released in 2013, a piece of 2.83-inch (72 mm) dongle hardware is becoming a hot topic recently, with 35$ price, it has been ranked as the most popular device in its category. The standard is proposed by joint effort form Google and Netflix, and as they are Internet companies, the standard is actually based on Cloud, the content is directly streamed from YouTube and Netflix to the Chromecast dongle. And applications running on mobile platforms are just acting as a control point. It also provides features like browser mirroring, with a Chromecast plugin, a Chrome browser can stream its tab to the big screen TV. In a foreseeable future, the standard could become more and more popular.

2.3.2 Market

- DLNA is one of the first proposed solutions for multimedia home networking, thus it is so far the most accepted solution, figure 7 shows the growth of DLNA-certified device sales. In 2018, the sales will reach 7.32 billion, nearly the totally population of earth.

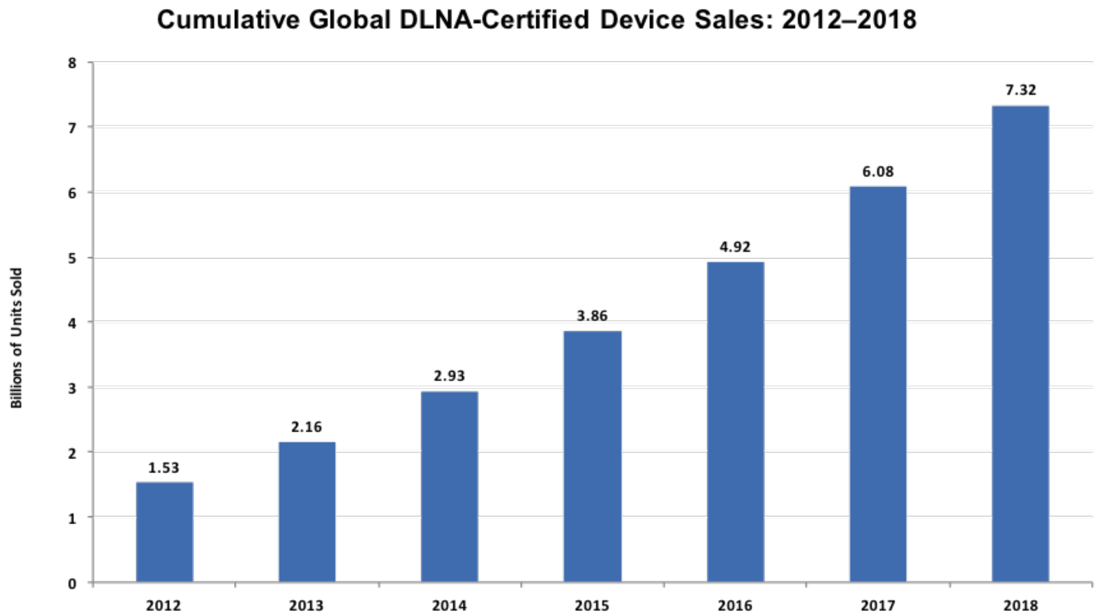

Figure 7: Cumulative Global DLNA-Certified Device sales

- AirPlay is bundled with Apple products, with great sales of Apple TV, Airport Express, Mac, iPhone, iPad, iPod, many family has just use Apple's product for everything. Thus AirPlay becomes the easiest way to build home networking solution, and maybe the only solution for those Apple users, since a lot of speaker manufactures implement their own AirPlay receiver on their AirPlay compatible speakers. And indeed it provides enough easy to use features to fulfill daily usage.
- Since bundled with Android operating system, Miracast has experienced a

fast growth in the past two years, many TVs built in the Miracast support to accept peer-to-peer Wi-Fi direct connection.

– Chromecast dongle is really cheap device that everyone wants to try, it can easily upgrade old TV to "Smart TV", and since Google has provided good content support for it, it is soon accepted by huge amount of users. Except for Google play online sales, it also ranked top 3 best selling devices recently.

2.3.3 Technology feature

1. Meida format support
   AirPlay and Chromecast has very limited media format support since there are only limited device types, so far Chromecast has only 2 device released. Apple TV now has 3rd generation box, but the media format support has not changed so much, the main improvement is the 1080p high resolution video support. On the other hand, DLNA has specified mandatory media formats in its specification, the "must have" formats only include MP3 and LPCM for music, JPEG for images and MP4 for video. Since Miracast is a screen mirroring technology, all formats can be played on device can be streamed.

2. Networking technologies used

   A short technology specification comparison is made to help better understanding the existing solutions, table 7 below shows the main technology used for different popular solutions.

Table 7: Technology used comparison

	Device discovery	Control Protocol	Streaming protocol
DLNA	SSDP	UPnP	HTTP
AirPlay	Multicast DNS	HTTP	HTTP/RTSP
DIAL	SSDP	Chromecast	HTTP
Miracast	Wi-Fi direct		Wi-Fi direct

   Apart from those basic technological details, some standards also offer advanced features compared to other solutions. For example, screen mirroring is an interesting feature that many standards offers. Table 8 below shows what advanced features each standard provides.

According to the comparison, each standard have its own features and uses different protocols to communicate. But there are common features supported by most standards, such as HTTP media server is used quite much to handle video and photo streaming. UPnP device discovery protocol SSDP is used quite a lot for device discovery.

Table 8: Advanced feature comparison

	DLNA	AirPlay	Chromecast	Miracast
Screen mirroring	No	Yes	No	Yes
Multiple connection	Yes	No	No	No
Authentication	No	Yes	No	Yes

Since there is some similarity that is used by most standards, there is possibility to make an application that include all these standard architecture and work with those protocols. Making a mobile application to connect those devices becomes a doable work.

3 Developing a solution for multimedia home networking

To fulfill the need of interoperability between devices in home networking, Tuxera Inc. started a project called Streambels(now called AllConnect). The project aims to solve the interoperability issue in multimedia home networking by making a universal solution that can connect all available devices at home and make them work together regardless of what protocol they use.

Most devices at home are embedded solutions and have their own firmwares, it is hard to update or even impossible to upgrade the software running on these devices. On the other hand, home network settings are usually simple and users don't have enough knowledge or interest to set up manually to obtain advanced network features. Most of these network infrastructures are not designed to be easily changed. The most flexible and programmable portal for home networking is mobile device.

Nowadays smart phone has great CPU power, networking capability and they are well adopted in people's everyday life. It is obvious that mobile device could be a good entry point of home networking. A mobile application that can be developed to control all the multimedia data flow and act as a personal access portal of home networking.

Through one year's development, our team developed an Android application that can be used to control and connect every multimedia device in the home. So far we have accumulated nearly 1 million users which gave us proven evidence of easy to use.

3.1 Architecture overview

The system architecture is consist of three major parts: discovery, content management and streaming.

Discovery component is responsible for device discovery which takes advantage of both Multicast DNS and Simple Service Discovery Protocol. As discussed in 2.2.1, the UPnP service uses Simple Service Discovery Protocol for device discovery, the application firstly send a M-Search request over UDP to the IPv4 multicast address 239.255.255.250 and UDP port 1900. Then the application listen to other devices' response. DIAL devices will return a response with Application URL header, while the UPnP/DLNA devices will return a message with a XML body, which gives detailed service URL and description URL. SSDP is used for both DLNA discovery and DIAL discovery. Multicast DNS is developed and implemented by Apple Inc., and there is an open source implementation on its website.

Content management component is responsible for organizing and navigating multimedia contents which can be found in the home network. In our solution, this includes both phone's local storage and DLNA digital media servers that connected to home network. The sources can be used in all of the three protocols that we support.

Streaming component is responsible for streaming multimedia content to selected multimedia receivers, such as TVs, wireless speakers, set top boxes etc. In our solution, since DLNA, AirPlay video/ photo and Chromecast all use HTTP streaming while AirPlay music uses RTSP. Two types of media servers are built inside our application. A RAOP server handles the AirPlay music streaming, a HTTP server handles all the streaming else.

In order to serve media for all the receivers, two types of serve method need to be implemented in a HTTP server. Since the DLNA guideline gave detailed specification how the media file should be served on a HTTP server, the HTTP server should be flexible to add DLNA specific headers, and should support byte based seeking operation to enable "seek" action on the receiver side. For receivers other than DLNA, a basic file server with byte range support will just do the work. After comparing multiple server implementations on Android, NanoHTTPd is the perfect library for our need, it is easy to use, Apache licensed, very tiny and efficient implementation, since it is minimal implemented, so it is also easy to be modified and add additional headers to be compatible with DLNA receivers.

A simplified version of our implementation is shown in the figure 8 below:

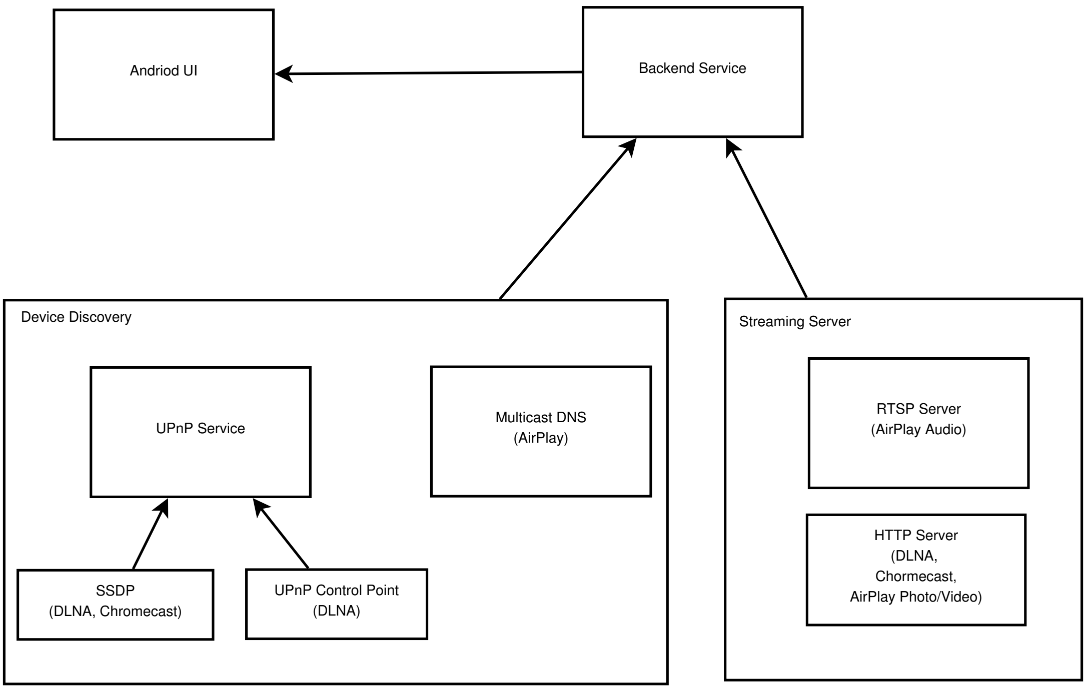

Figure 8: Simplified flow chart

In terms of data flow, as described in figure 9, there are three different types of data flow models:

If the streamed content is stored in mobile phone, a streaming server in the application is used to stream the content from phone to selected receiver.

Otherwise, if the content locates on the Internet, and the receiver is a DLNA Media Renderer, a proxy is needed to firstly download the resource stream, add the required headers by DLNA protocols and then stream the content to selected DLNA Media Renderer.

Finally, if the streamed content locates in a DLNA Digital Media Server, the source can be directly used by all receivers, in this case, the content streaming happened directly from media server to receivers, application is only used as a control point and do not participate in the media transmission.

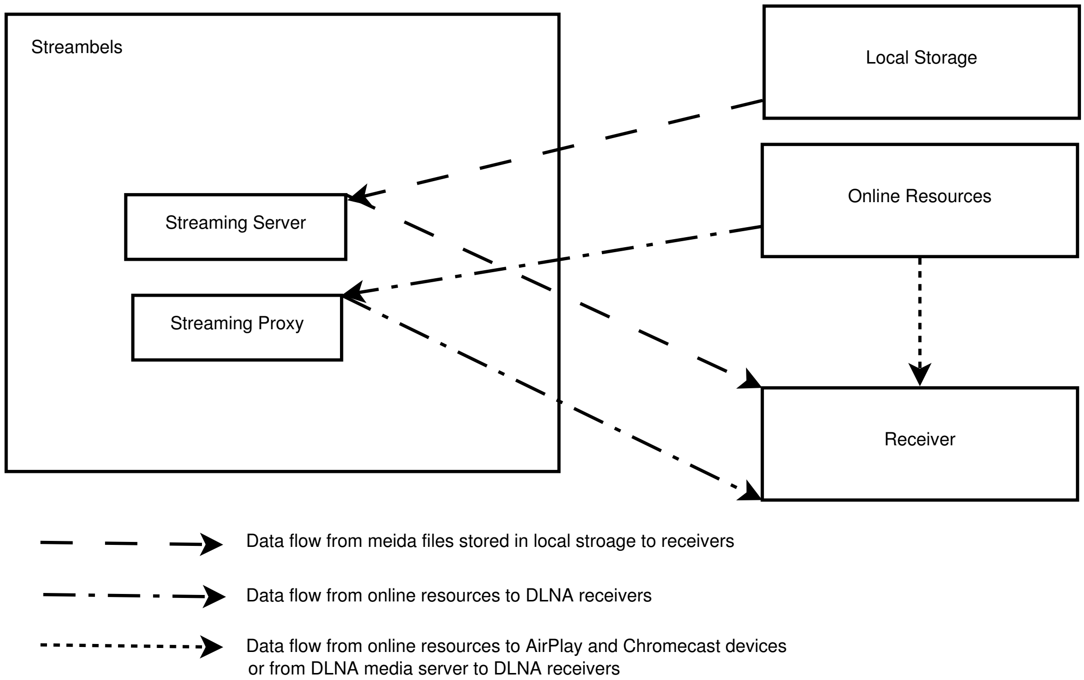

Figure 9: Simplified data flow

3.2 UX design

In terms of UI and UX, our designer Hien did a great job of defining a simple and intuitive user interface 11, which benefits a lot for the user experience. In every place in the application, selected receiver is visible to user, it is also visible as a button for switching between different receivers. 9 themes are also available for users to select, which makes the application more fun to user.

3.3 Features

As described previously that AirPlay, DLNA and other standards work differently and have different features, but according to the previous study, we could combine some common use cases of both protocols.

The Android application we developed can handle most multimedia devices in a typical home networking. It has various features that make it a useful and universal solution for multimedia home networking.

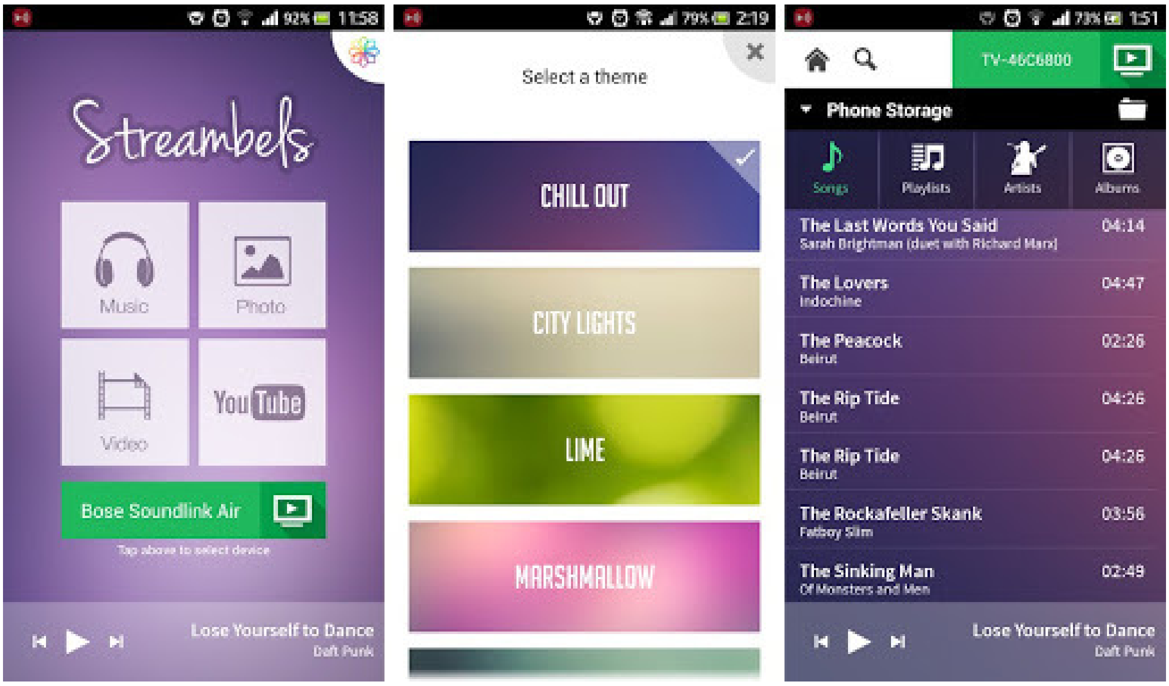

Figure 10: Application UX design

Firstly, the app itself is a multimedia playing, all media stored locally on the phone storage and all media located in the home DLNA media servers can be browsed and played on the phone.

Secondly, the app is fully compatible with AirPlay, DLNA, Chromecast and FireTV receiver devices.

Thirdly and most importantly, the application make the DLNA media server works together with all kind of receivers regardless of protocol used. The app served as a bridge of different multimedia receivers and media sources.

Last, YouTube and other on-line channels like Vimeo and Facebook are supported as media source, and can be streamed regardless of protocol to all supported receivers that are connected to home network.

- Firstly the app is a multimedia player, it can play music, photos and videos on SD card locally on Android phone
- It can stream local content to Apple TV, Airport express and AirPlay-enabled speakers.
- It can stream local content to DLNA media renderers, which has a huge device base.
- It can stream local content to Chromecast devices.
- It can browse content from the DLNA media servers, a typical source is a Network Attached Storage (NAS). And play the media locally on the Android device.

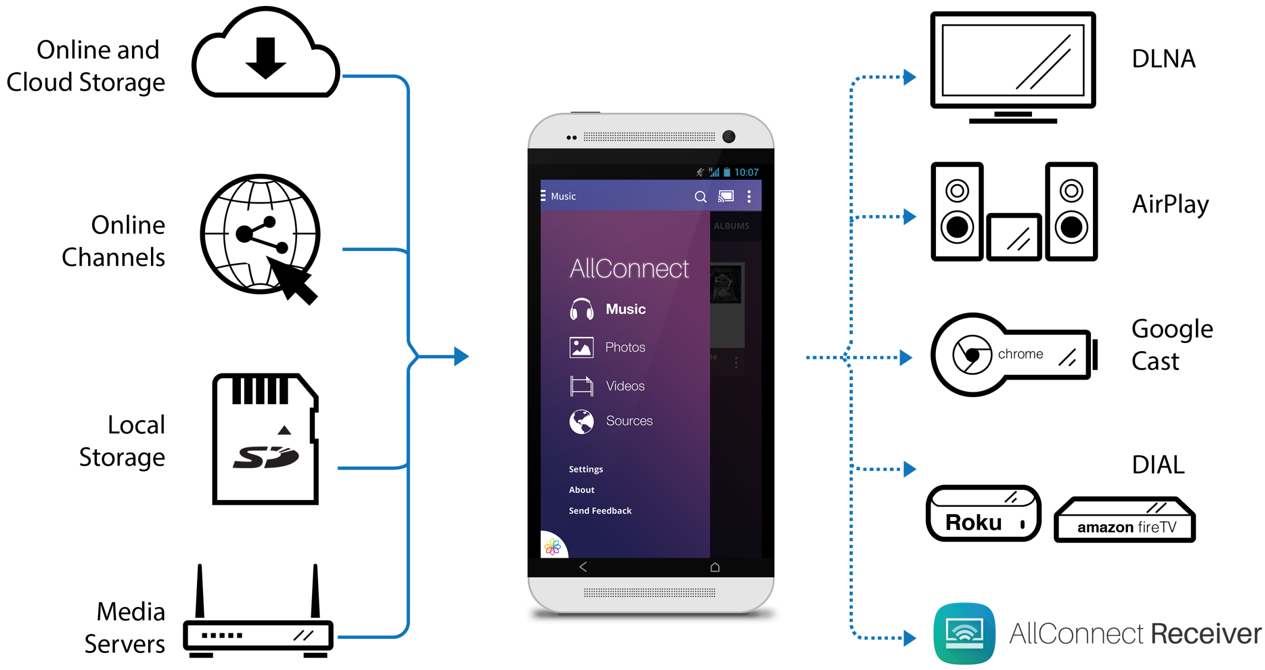

Figure 11: Application UX design

- It can browse content from the DLNA media servers and stream it to DLNA media renderers.
- It can browse content from DLNA media servers and stream it to AirPlay enabled devices using a different protocol.
- It can proxy online channels' content to DLNA and AirPlay enabled devices. (Currently YouTube and Facebook videos are supported, but integration to Spotify is still in progress).

3.4 Extensibility

In normal use cases, the data flow can be shown as figure 9, Streambels has embedded a media streaming server for local files and streaming proxy for bridging the gap between online resources and home networking. By using built-in proxy, Streambels is able to share on-line resources from Internet to devices in home networking environment. New services and new content providers can be easily added by just call the proxy interface.

The proxy system enables a huge extensibility possibility, which connects the home networking and Internet or Cloud Services.

Further development abstracted all the protocols and developed an API for all content holders or cloud services to integrate their service.

3.5 Test methodology

Software testing is quite important for a modern IT project. Through testing we can assure performance and stability, before releasing the app to App store, we conduct a series of tests.

These tests include unit test, integration test and functional test. Unit tests is written while coding, when each class is finished, unit tests will be written for each method, we set up an continuous integration server so that each time when we commit something to the git repository, a full unit tests will be run and if there were some failure in the unit tests, developer will be informed. Integration is done in a way that we ensure each function module should work together with other modules in the system. Last, we listed all the possible use cases, prepared a huge media base which contains all kind of media files, we do manual test before the app is finally released in the market.

3.6 Evaluation methodology

Since the project is targeted to Android market, and is directly used by end users, feedback is really important to us for the continuous development. We used Email for normal communication, user can edit feedback content directly inside the application, and later content is sent to us by email.

There is no perfect application, so crash sometimes happen, thanks to Google, the Google will help to collect the crash reports and show it inside developer console.

Inside Streambels, we also used Google's Analytics API, which give us great convenience to collect number of users and sessions every day. Other information like operation system version, application version, active users helped us to have insights into who are our users and how can we market for more people in the world.

It is also interesting to see what kind of technologies are most used in their daily life, thanks to Google's analytics SDK, we can trigger events when user selected their receivers, so after months of statistics, we can figure out the most popular standards and most popular online channels that user uses, and in turn make better optimizations for users need.

4 Results

After designing, implementing and testing the software. We started several evaluation tests on the streaming performance, and we have released the application to public in Nov 2013. So far, we have improved the application a lot and also brought many new features to it. As discussed in previous chapter, we used Google Analytics SDK to help us get insights to our application and our users. In this chapter we will present and analyze those results.

4.1 Performance

In terms of streaming, our solution include two major streaming components. It would be helpful to study how the media streaming protocol have the better performance while streaming multimedia contents. And also, by comparing different streaming types of media, we could investigate which protocol is best suitable for certain type of media. Two major streaming technology we used in our solution are HTTP streaming and RTSP streaming.

Hypertext Transfer Protocol (HTTP) refers to the protocol used to deliver web pages and images across the Internet worldwide. HTTP is an adopted, open standard- the most ubiquitous mode of delivery on-line. HTTP is a "stateless" protocol; think of it as an airline ticket to anywhere. HTTP can be delivered by a variety of web servers, both commercial and open source.

Real Time Streaming Protocol (RTSP) is a network control protocol used in entertainment and communications systems to control streaming media servers. RTSP is used to establish and control media sessions between two points, usually server and player client. Clients of media servers issue VCR-like commands, such as PLAY and PAUSE, to facilitate real-time control of playback of media files from the server. Microsoft's Smooth Streaming is a hybrid delivery method that acts like traditional RTSP streaming but is based on HTTP PDL. Apple's HTTP Live Streaming (HLS) is another example of using HTTP to deliver video with new techniques.

AirPlay uses RTSP streaming for the iTunes music. Thus we chose a mp3 music and try to stream it to an AirPlay Speaker and an DLNA Speaker, and we used Wireshark running on rooted Android phone to capture the packets in the network. The result is presented below, the initial packet count is relatively high, this is because there is a lot multicast messages in the network for device discovery, after that, streaming graph shows that after an initial increase in the traffic, network traffic is relatively stable, this is because the TCP protocol reaches the best optimized transmission speed.

Next step, we added packet loss in the network, and see the influence to the two streaming process, result is shown below:

4.2 Statistics

Through one year's on-line marketing, we reached 637000 downloads from 223 countries all around the world, with around 8000 daily active users. So far, the average rating of our app is 3.9 out of 7527 ratings. The application turns out to be popular in countries like France, United States, Germany, United Kingdom and Brazil.

– Totally more than 290000 downloads

– Used by people from 201 countries

– 8000+ daily active users

– 1,241,074 visits to home page

– Average rating 3.9, 3521 user gave rates

– On-line for 3 months

Table 9: Receiver type statistic

Device type	Total selection	Percentage
AirPlay device	637446	39.72
DLNA device	460139	28.67
Phone speaker	353474	22.03
Chromecast device	147333	9.18
FireTV device	6368	0.40

4.3 User study

– What information we can get back from users

– User behavior/ statistics

– Improve the application accordingly

– Strategies for decision making

Write result here.

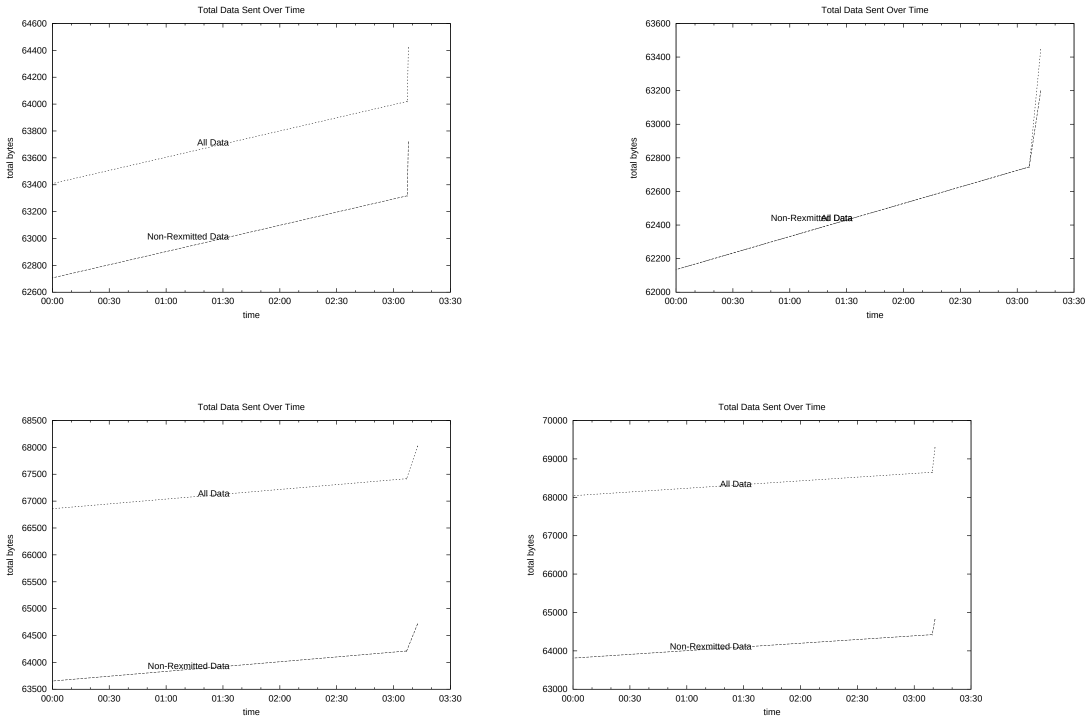

Figure 12: AirPlay streaming performance in terms of packet loss

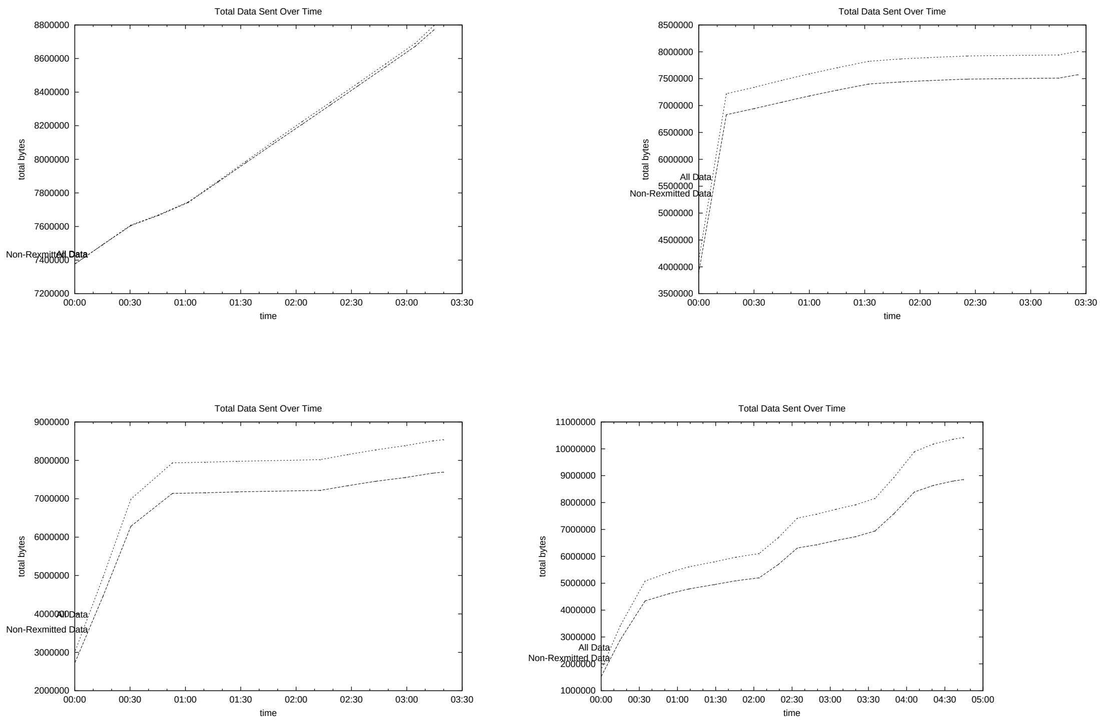

Figure 13: DLNA streaming performance in terms of packet loss

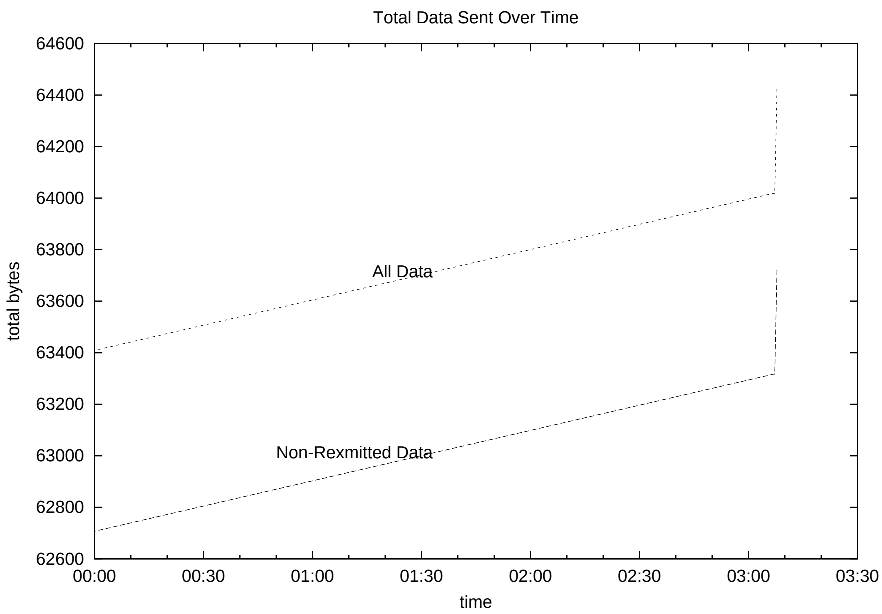

Figure 14: AirPlay traffic data chart

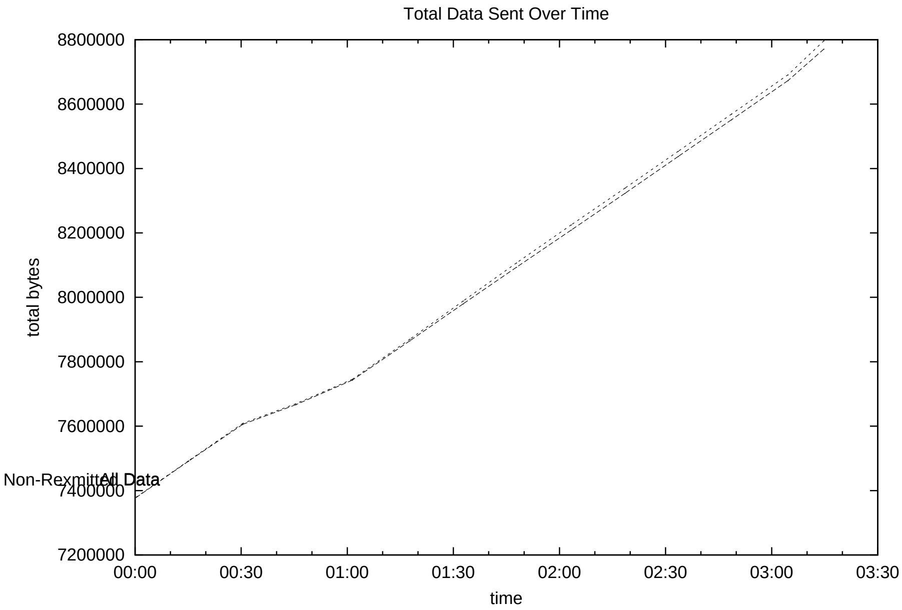

Figure 15: DLNA traffic data chart

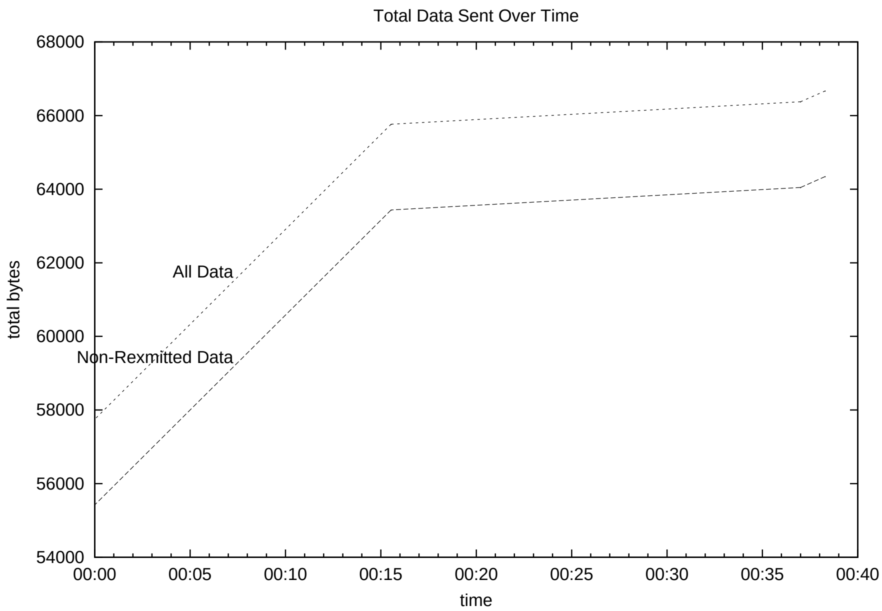

Figure 16: AirPlay bad network traffic data chart

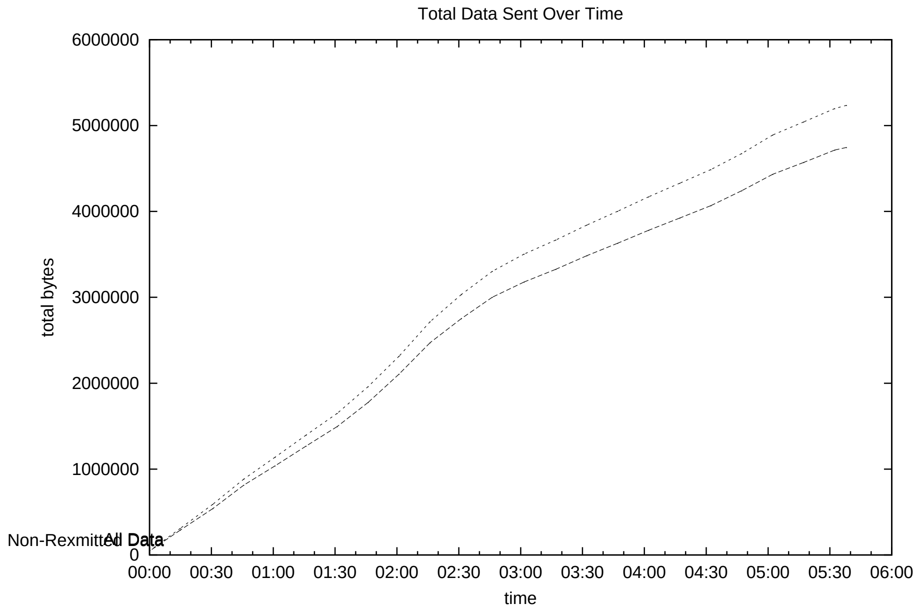

Figure 17: DLNA bad network traffic data chart

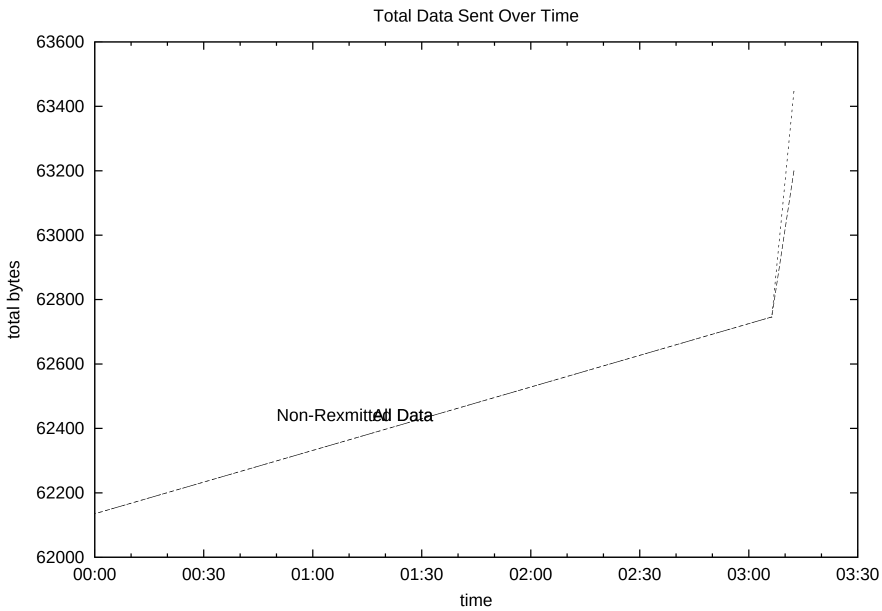

Figure 18: AirPlay bad network traffic data chart

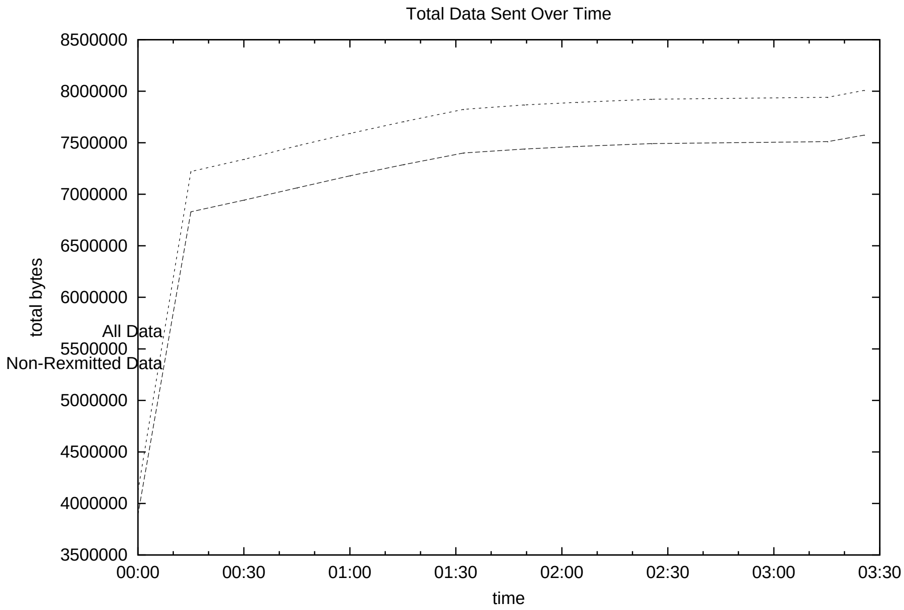

Figure 19: DLNA bad network traffic data chart

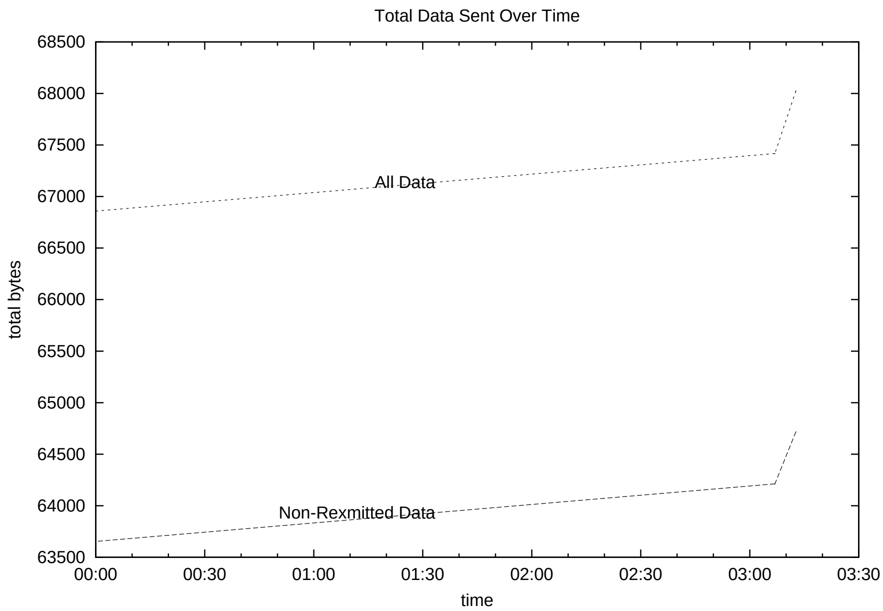

Figure 20: AirPlay bad network traffic data chart

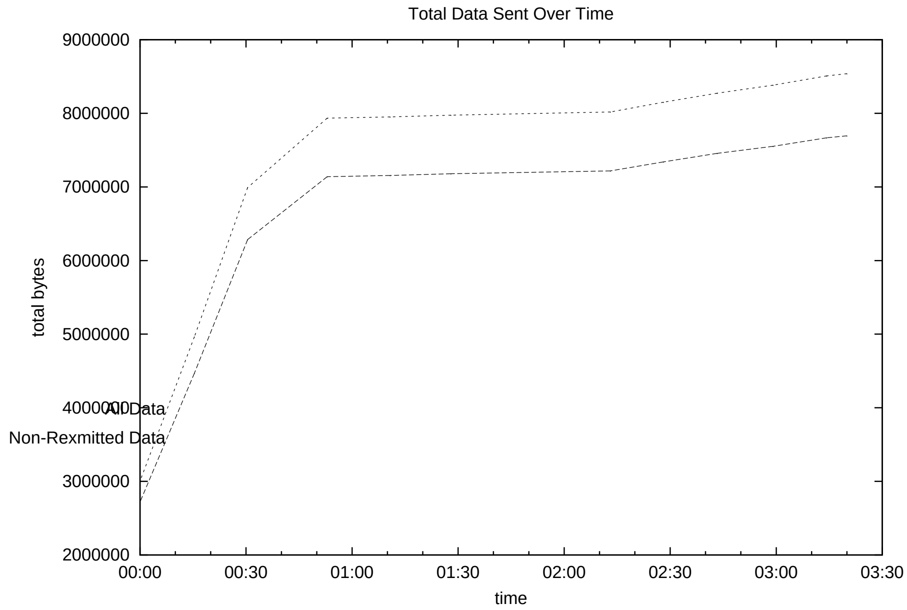

Figure 21: DLNA bad network traffic data chart

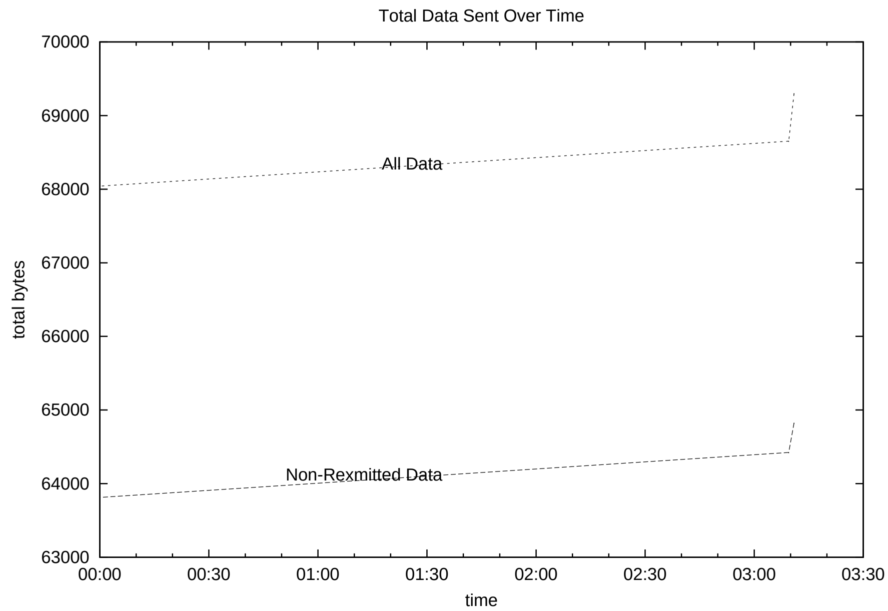

Figure 22: AirPlay bad network traffic data chart

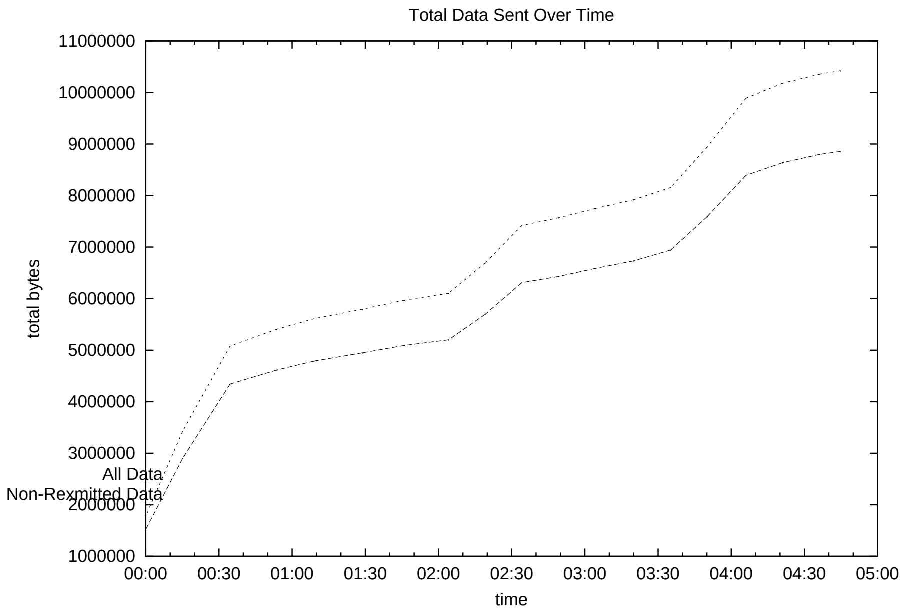

Figure 23: DLNA bad network traffic data chart

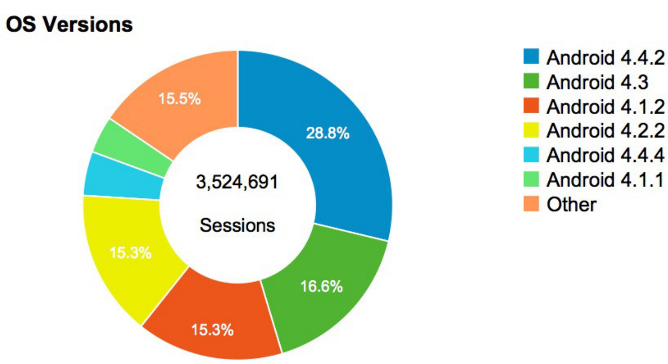

Figure 24: Android versions

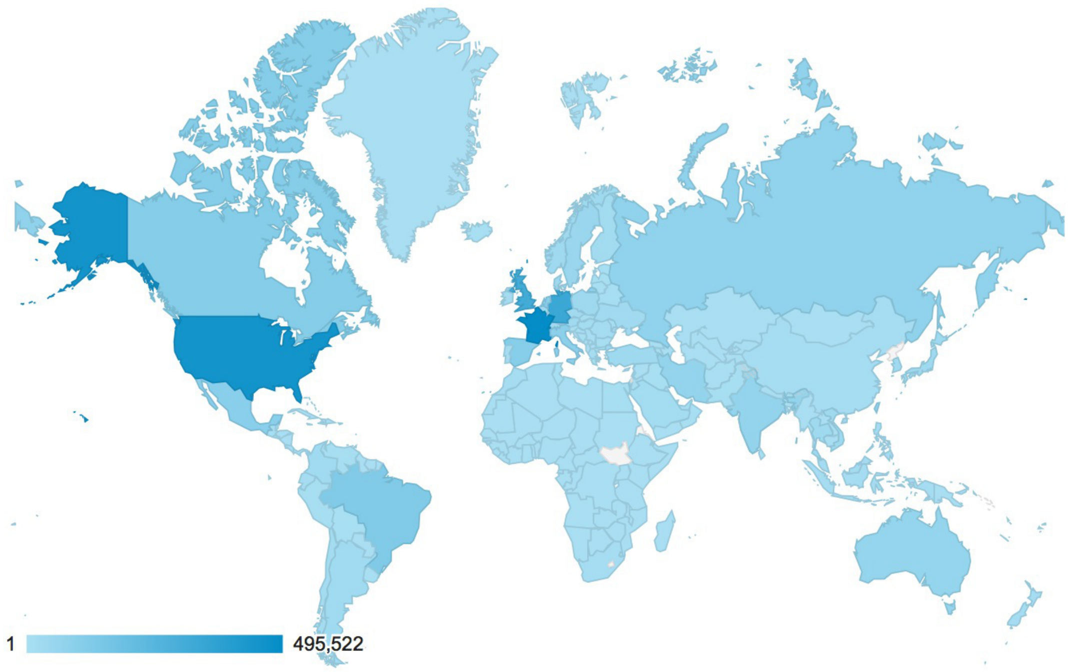

Figure 25: User world map

5 Discussion

Write discussion here.

5.1 Further development

Development of a universal streaming SDK is essential for different manufacturers and on-line content providers. (Connect SDK by LG)

Kernel level optimization, multimedia transcoding, file system and network layer optimization.

More platforms.

5.2 Future of multimedia home networking

- Multiple standards may still exist for quite a long time
- More devices will support streaming technologies, there will be more interactions between devices like mobile phone and TVs
- Cloud and local storage will both work and make it easy for users to access their multimedia content
- Internet of things, smart home

[1] [9] [15] [16] [5] [17] [7] [3] [18]

References

[1] Ratul Mahajan David Wetherall W. Keith Edwards, Rebecca E. Grinter. Advancing the state of home networking. *Communications of the ACM*, 54(6):63–71, June 2011.

[2] Xinhua Feng. Home networking. Technical report, Ohio State University, 2000.

[3] Sandy Teger and System Dynamics Inc. David J. Wakes. End-user perspectives on home networking. *IEEE Communications Magzine*, pages 114–119, April 2002.

[4] Bruce Nordman. Summary of analysis of communication link technologies. Technical report, National Lab Buildings Energy Efficiency Research Projects, September 2012.

[5] Charlie Tai Mark D. Wood Brent A. Miler, Toby Nixon. Home networking with universal plug and play. *IEEE Communications Magzine*, pages 104–109, December 2001.

[6] Wouter van der Beek Jeffrey Kang John Ritchie, Thomas Kuehnel. Upnp av architecture. Technical report, UPnP Forum, December 2010.

[7] Eui-Hyun Paik Kwang-Rog Park Yeon-Joo Oh, Jung-Tae Kim. The dlna proxy system architecture for sharing in-home media contents via internet. *ICACT2006*, pages 1855–1858, 2006.

[8] Digital Living Network Alliance. Dlna guidelines. Technical report, Digital Living Network Alliance., March 2014.

[9] Airplay specification.

[10] Marc Krochmal Stuart Cheshire. Multicast dns, February 2013.

[11] Inc. Netflix. Discovery and launch protocol specification. Technical report, Netflix, Inc., 2014.

[12] WiFi Alliance. Wi-fi certified miracast: Extending the wi-fi® experience to seamless video display. Technical report, WiFi Alliance, September 2012.

[13] 11 2013.

[14] 1 2014.

[15] Takayoshi Okodo Yohei Tsuchida Tomoyuki Yamaguchi-Takahiro Murooka Kimio Oguchi, Kunio Tojo. Next generation home networking and relevant technologies. *Proc. of SPIE*, 5626, 2005.

[16] Hoon-Ki Lee Eui-Hyun Paik Kwang-Roh Park Jung-Tae Kim, Yeon-Joo Oh. Implementation of the dlna proxy system for sharing home media contents. *IEEE Transactions on Consumer Electronics*, 53(1):139–144, February 2007.

[17] Bill Rose. Home networks: A standards perspective. *IEEE Communications Magzine*, pages 78–85, December 2001.

[18] Yueh-Min Huang Chung-Sheng Li. Upnp ipv4/ipv6 bridge for home networking environment. *IEEE Contributed Paper*, August 2008.