4 Results

After designing, implementing and testing the application. We conducted several evaluation tests on the streaming performance. We had released the first edition of the application in Google Play Store in Nov 2013. Since then, we had been updating the application in order to improve its performance and enrich its features. During the past one year, we have collected a big amount of useful data and interesting results.

This chapter presents the results of our evaluation. Section 4.1 evaluates the streaming performance of different streaming protocols. Section 4.2 presents the statistics from our users. Section 4.3 provides a user study and describes how we have improved the usability of our solution by utilizing this study.

4.1 Evaluation of streaming performance

In terms of streaming, our solution includes two major streaming components. It would be helpful to study and compare which streaming protocol has the better performance while streaming multimedia contents in different network conditions. The two major streaming technologies we used in our solution are HTTP streaming and RAOP streaming, respectively.

Hypertext Transfer Protocol (HTTP) is the protocol used to deliver web pages and images across the World Wide Web. HTTP is a most widely adopted, open standard and the most ubiquitous method of content delivery on the Internet. HTTP objects can be delivered by a variety of web servers, including commercially used servers and open source servers. Both DLNA and Chromecast use HTTP to realize their streaming functionality.

Unlike HTTP, another popular protocol, the Real Time Streaming Protocol (RTSP) is a network control protocol used in entertainment and communications systems to control media streaming servers. RTSP is used to establish and control media sessions between two points, usually the 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. The RAOP protocol, which is virtually another version of RTSP, is used by AirPlay, for the streaming of iTunes music.

This section presents the evaluation of HTTP and RTSP streaming servers and compares the streaming performance in different network conditions. Since we have both protocols implemented in our application. We could compare the performance by streaming the same content to two receivers using the two different protocols. Typically, DLNA music streaming uses HTTP protocol and AirPlay music streaming uses ROAP protocol. After the experimental setup described in Section 3.7.1, we conducted three different experiments. Section 4.1.1 compares the streaming traffic of DLNA and AirPlay standards. Section 4.1.2 compares the streaming performance of DLNA and AirPlay in limited bandwidth situation. Section 4.1.3 presents the similar comparison but under high packet loss scenarios.

4.1.1 Comparison of AirPlay and DLNA traffic

After the initial experimental setup described in Section 3.7.1, we selected a mp3 music file and try to stream it to both an AirPlay receiver and a DLNA receiver, and we used Wireshark running on the laptop to capture the packets in the network.

After the experiment environment is set up, a series of test is conducted and the result is presented in Figure 12.

According to the result, the stream traffics of AirPlay and DLNA streaming are very different. Figure 12(a) shows the scenario of AirPlay music streaming, x-axis is the duration of the stream since the beginning of the music; y-axis is the total traffic in bytes accumulated since the beginning of the stream. There are two lines which represents the total data and non-retransmitted data separately. According to this figure, the traffic growth of AirPlay streaming is nearly linear, only two slow-downs occurred at 1:30 and 2:00 due to network condition. This is because ROAP is a push like process and content can be streamed in real time.

In contrast, Figure 12(b) shows that in DLNA streaming, there are clearly two phases in the traffic graph. Before 0:20, the amount of traffic grows rapidly. After that, the traffic growth slowed down and keeps the same increase rate till the end of the stream. The reason behind it is that HTTP streaming is a pull like process, the server can actively fetch content from the media server and the content can be buffered up since the beginning of the playback, with the best effort of the network. Therefore, there is a short download period at the beginning of the DLNA streaming graph. After all the content is buffered already, the traffic growth is the result of constantly update of playback status.

This comparison describes the different characteristics of HTTP streaming and RAOP streaming. These differences could result in different performances and user experiences. In the following sections, we will identify these differences by changing the network conditions. And finally we could draw the conclusion which protocol is more suitable for multimedia home networking.

Figure 12: AirPlay vs DLNA streaming traffic comparison

4.1.2 Performance under limited bandwidth

During this experiment, we reused the same setup mentioned in Section 3.7.1, in addition, different bandwidth limitations are introduced. We evaluate the performance of the two streaming solutions under limited bandwidth. The bandwidth is limited to 500 kbps, 700 kbps and 1000 kbps respectively in three rounds of tests. During each round of the test, the same mp3 music is streamed to XBMC receiver using the DLNA standard and the AirPlay standard. Figure 13 and 14 show the result of DLNA streaming.

Figure 13(a) shows that when there is no limit in bandwidth, the graph of total types is the steepest, which means the loading speed is the fastest. Figure 13(b) shows the traffic graph when bandwidth is limited to 500kbps, Figure 14(a) shows the total traffic graph when the bandwidth is limited to 700kbps and Figure 14(b) shows the traffic graph when the bandwidth is limited to 1000kbps. According to the Figure 13(b), 14(a) and 14(b), the initial loading speed is dependent on the bandwidth of network. When the network bandwidth is limited, as the bandwidth of network increases, the initial loading speed would also increase. This proves that in DLNA streaming, most of the content is fully downloaded in the initial phase of streaming, because HTTP streaming is used in this case. The quality of steaming can be guaranteed when the initial buffering is finished. The receiver will take the buffered content and play locally.

In contrast, AirPlay music streaming is based on ROAP, which is a RTP-like proto- col. The transport layer protocol used is UDP, thus not all packets are successfully delivered to the receiver. The UDP based delivery only provides a best-effort trans- mission. The sound quality can not be guaranteed because there is no buffer or retransmission mechanism on the receiver side and the transmission is almost real- time. Given this reason, when the bandwidth is limited to 500 kbps, the AirPlay playback is heavily interrupted. All music information is lost since too many packets are lost during the transmission. When the bandwidth is increased to 700 kbps, the AirPlay playback still can not work properly. The playback is choppy and noisy. After the bandwidth is increased to 1000 kbps, the music can then work smoothly and no noticeable noise can be heard.

Another thing worth mentioning is that in the experiment, the same mp3 music is used in both tests. But obviously the playback quality of DLNA is better than the playback using AirPlay. For instance, when the bandwidth is limited to 500 kbps, AirPlay streaming is basically not usable any more, while DLNA streaming is still working properly. The reason behind this is that in the case of DLNA streaming, the original mp3 music is directly streamed to receiver, while in the case of AirPlay music streaming, the same mp3 music is firstly decoded to PCM format and then encoded to Apple Lossless format. Since mp3 is a compressed media format while Apple lossless format is an uncompressed media format, the bandwidth required by DLNA streaming is considerably smaller than AirPlay streaming.

Figure 14: DLNA streaming traffic comparison in bandwidth constrained situation

As a conclusion, in the scenario of limited bandwidth, DLNA have the advantage of sound quality compared to AirPlay standard. The buffer system, benefited from HTTP streaming mechanism, gives DLNA a more reliable data flow. While AirPlay streaming suffered from the packet loss of UDP and the sound quality is heavily affected. Generally speaking, DLNA is more tolerant than AirPlay streaming in the case of limited bandwidth.

4.1.3 Influence of packet loss

After the bandwidth test, we then simulated packet loss on the receiver side and conducted the same test for DLNA and AirPlay streaming. 5%, 10% and 15% packet loss are introduced to test both DLNA and AirPlay streaming. The result of DLNA streaming is shown in Figure 15 and 16.

According to the figure, when there is no packet loss, very little retransmission data is seen in the graph. When the packet loss ratio is increased from 5% to 10%, the portion of retransmitted data is getting larger and larger. However, there is no noticeable loss of sound quality. When the packet loss ratio is increased to 15%, significant amount of retransmission can be found in the graph, and the streaming stopped for buffering three times during one song’s playback. In the case of DLNA streaming, packet loss is a key impediment. When packet loss rate climbs to a cer- tain point, for instance 15% in our test, the streaming becomes unusable.

The same packet loss tests on AirPlay streaming was also conducted and the re- sults are shown in Figure 17 and 18. According to the figure, in the case of AirPlay streaming, since UDP is used, the ROAP server embedded in Streambels keeps send- ing data to the receiver using UDP regardless of the packet loss. On the receiver side, the player tries its best to decode the broken data. There is no mechanism for acknowledgement or retransmission. Surprisingly, the sound quality is much better compared with DLNA streaming. The reason behind is that retransmission of TCP consumes more and more bandwidth in the case of DLNA streaming, in the same situation. AirPlay streaming instead tries to deliver all contents with its best effort, without creating extra demand for retransmission.

In a nutshell, in the case of packet loss, AirPlay is more tolerant to packet loss than DLNA.

Figure 18: AirPlay streaming performance in terms of packet loss

4.2 Statistics

16 months after its release, our application has achieved 924000 downloads from 223 countries all around the world, with a daily active user number of over 15000. Our users almost cover 99% of all continent and a world map of our user distribution is shown in Figure 19. So far, we have received ratings from 10253 users and currently

Figure 19: World map of visits

the average rating is 3.9 out of 5. The distribution is shown in Figure 20. According to the rating distribution graph, most users are satisfied with our solution and give the 5 stars rating. However, the average rating is heavily influenced by the 1 star rating users. The reason for those low ratings is that the receivers some users have in home are not compatible with our application due to different reasons. It might be that some protocols,such as Roku box, are not supported yet. Network condition problem also contributes to the incompatibility issues. For example some routers have by default disabled multicast due to security reasons. Another major cause of the incompatibility problem is that even with the same protocol, such as DLNA , a minor implementation difference may cause the break of connections. Thus, in the later phase, we have made receiver specific hacks to make our application work with most DLNA receivers, regardless of which implementation they use. In terms of user distribution, in the last 16 months, our application turns out to be very popular in countries like France, United States, Germany, United Kingdom and Brazil. The user distribution is shown in Figure 21. We have also translated the description of our application to nine languages, which include English, Russian, German, Italian, Japanese, French, Chinese, Spanish and Korean. The multiple language support may also contribute to the popularity of our application in all parts of the world. The most popular operating system used for our application is Android, but interestingly there are also tens of users was using our application on BlackBerry 7, which is another mobile operating system developed by BlackBerry Inc. For all the users on Android system, the distribution is shown in Figure 22. According to the statistics, Android Kitkart is the most widely used Android version, which is also our target system version. Most users have updated to the Kitkart version 4.4. We also have a very small part of users who are using the latest Android 5.0(Lollipop). These users are more likely to experience more bugs. The reason is that the latest Android uses Android RunTime (ART) to replace Dalvik runtime[16]. The native code (C code) support is not optimized so well for the new ART architecture. How- ever the native code support shall be improved in later update of Android Lollipop operating system.

The most interesting statistic of our application is the receiver popularity, which is shown in Figure 23. Since our project aimed at developing a solution for multimedia home networking, it is especially useful to find out which protocol is the most popular one. According to the result from Google Analytics, the AirPlay and DLNA are the most popular standards, with a combined usage rate of 87% among all streaming sessions. Chromecast is the third most popular receiver while Amazon Fire TV is the least adopted receiver. The result can also suggest the future trend of multimedia home networking. Our application has gone public for 16 months and it has seen a series of small updates and one major update on last Christmas. The number of daily visits is shown in Figure 24. As shown in the figure, after the release, the number of users has seen a great increase in the first two months. After that, the number of active users has remained steady over the following six months, until we launched a major release after around a year. The new release included an updated UI and a better written streaming component. This release has brought a significant growth in users. Currently our application is hitting 15000 visits per day, proving that our solution has been successful.

4.3 User study

It is essential to listen to feedback from users and our application users can send their feedback to us in many ways, such as commenting on the Google Play Store or sending E-mail directly to us.

Having been collecting user feedbacks for over 16 months, we have made several improvements accordingly. Most of the feedbacks are about trouble shooting. For example, device discovery problem has been the most common issue for most complaining users. The reason could be that phones and other receivers are not connected to the same Wi-Fi network, or the receivers are not turned on. With more and more user feedback collected, we felt obliged to set up a website for trouble shooting. After the setup of the feedback website we kept updating and improving the trouble shooting pages. Now most complaining users could find their answer in this trouble shooting pages.