Abstract

The purpose of this project is to build a household real-time synchronous wireless multimedia system(RT-SWMS) and potentially explore the limitation of such a system. The timeline for this project is roughly separated into two parts. Real-time wireless network protocol using multicast and synchronization under UDP multicasting environment. So far a multicast protocol is implemented by using an adapted version of the Scalable Reliable Multicast. Furthermore, changes should be made in order to develope a synchronous system.

Acknowledgements

Many thanks to Dr. Klaus Peter Zauner and my supervisor Denis A Nicole for their guidence in this project.

1 Introduction

The purpose of this project is to build a household real-time synchronous wireless multimedia system(RT-SWMS) and potentially explore the limitation of such a system. The system consists of many nodes in a network all connected to a home router using WiFi. Each of this node is modeled by a Raspberry Pi Zero connected to a portable speaker and listen to a multicast server in the same private network. However, unlike Peercasting [**?**], which is multicasting via peer-to-peer technology, our system uses IP Multicast in a private network and therefore will follow the classic client-server model. Each node will return an acknowledgment back to the server upon receiving each datagram, such response time will be measured and the server will attempt to synchronize all nodes base on the delay in acknowledgement.

2 Background

2.1 IP Multicast

Multicasting is a method for sending UDP datagram to a group receivers in a single transmission. Which is different to unicasting where a unique datagram is sent to each receiver. One of the common difference is that Multicasting often uses UDP protocol, whereas broadcasting usually requires TCP protocol. The different in protocol allow Multicast to reduce a large amount of redundant traffic introduced in the three-way handshake. However, the use of UDP would compromise its reliability as there is no way of knowing if the datagram has successfully arrived at the receiver.

2.1.1 Reliable Multicast

Reliability of Reliable Multicast(RM)[**?**] depends on the specific protocol, the vague definition of RM is mentioned by S. Floyd, V. Jacobson and C.-G. Liu such that *"eventual delivery of all the data to all the group members, without enforcing any particular delivery order"* [**?**]. However depending on the protocol, reliability is sometimes a trade off for efficiency or performance. Since RM is a multicast protocol, it highly relies on NAK-based protocol[**?**] to ensure it's reliability and shifts the responsibility to the receivers.

Scalable Reliable Multicast(SRM) is an NAK-based protocol based on IP

multicast. The receiver will request a retransmission by IP multicast if it detects a data lost. The server receiving a retransmission request will then repair the lost through IP multicast again. A major advantage of SRM is involved the use of a randomized mechanism to avoid a storm of retransmission. The receiver has a randomized delay before determining the data was lost and request for retransmission.

2.2 Audio Quality

The router which the project is using is running on 802.11n Wifi Protocol, the lowest maximum data rate running at 2.4 GHz is 6.5 Mbps[**?**]Wifi:1. In comparison, the audio traffic when running at the setting of 16 bit, 44.1 kHz has a bitrate of 705.6 Kbps. This means the audio traffic alone without any real-time audio compression would take up just over 10% of the data rate.

$$Bitrate = Bit\ depth \times Sample\ Rate$$

3 Implementation

Although this project is fundamentally a software base project, there are also a few hardware issues need addressing.

3.1 Hardware

Single-board Computer(SBC) is used to model each node in the multicasting group. A Raspberry Pi Zero is used due to its affordability and portability.

3.1.1 Audio

One of the problems of the full-size Raspberry-Pi is that its analog output interface (which is not available on the Raspberry Pi Zero) produces a noticeable amount of noise. Such noise is induced by the power supply and there is no simple way of removing it. One of the solutions is to utilize the high-quality HDMI output and split the video and audio signal apart. However, such solution requires an expensive HDMI video/audio splitter and would also reduce the portability of the SBC. Alternatively, audio signals could also be produced by the GPIO pins on the Raspberry-Pi. However,

such audio output was tested to be much worst than the original analog output. Therefore a USB audio interface is used to produce much cleaner audio performance without compromising the SBC's portability.

3.2 Adaptation to the SRM

As mentioned previously about SRM's random delay machanism, such delay has been randomized to between 1-5ms.

In order to increase the reliability of SRM, an Exponential Backoff(EB) algorithm[**?**] was introduced for sending NAK. This enables SRM to send out NAK repeatedly without introducing a constant volume of traffic incase the client could not reach the server for a period of time. For example if the initial delay is 5ms:

1. Expected time for datagram to arrive from the server.
2. If datagram is not received, wait between 1-5ms.
3. Send NAK
4. If datagram still not received, wait for 5ms, then 10ms, then 20ms... before sending NAK

3.3 Client Recovery

A drop of a client is always expected due to poor connections. If the Exponential Backoff(EB) algorithm exceed 30,000ms (30 seconds), the Exponential Backoff NAK will terminate to prevent further waiting time. The client software will then restart and attempt to rejoin the multicast group. The multicast group discovery is triggered with a uniform time as multicasting server could reboot and appear at any given time. Contrary to the NAK sending time, server discovery signals are sent much further apart in time than NAK. Therefore server discovery signals will not introduce a large volume of traffic even if a large number of nodes are searching for a crashed server.

4 Remaining Work

4.1 SRM delay

The SRM delay mechanism has been randomized in a hard-coded fashion. However, such randomized delay should more or less be correlated to the estimated population of the multicasting group. For example, if the group size remains small, such delay could be minimal as NAK flooding would not cause any serious traffic. However, as the population of the group increase, the upper bound of the randomized delay should also increase.

4.2 Synchronization

Synchronization mechanism remains largely untouched at the time of this report. A UDP-based protocol will be implemented without using external protocol like the classic Network Time Protocol(NTP). An early implementation is developed base on the worse performing node. An ACK is expected from each node after a period of time to estimate the worst performing node. The server will monitor the worst delay experienced by the node and enforce such delay between the time of receiving data and data output at every node. Further research will be carried out in order to find better synchronization method.

As part of evaluation, the test must consist of qualitative method of measuring the quality of synchronization. This will be part of the next milestone for the testing of synchronization of the system.

5 Conclusion

Significant progress has been made in implementing RM as well as the basics on client recovery. Further client recovery mechanism will soon be added as well as the client synchronization functionality. A rigorous testing will be put in place to ensure the system perform as expected under undesirable network conditions.

6 Timeline

Note: Completed task indicated in purple, incomplete task indicated in grey