Experimenting With a Framework for Networked Mobile Audio Arrays

Peter J. Taillon[1], Karen Collins[1], Bill Kapralos[2]

[1] *Canadian Centre of Arts and Technology, University of Waterloo, Waterloo, Canada*

[2] *Faculty of Business and Information Technology, University of Ontario Institute of Technology, Oshawa, Canada*

Correspondence should be addressed to P.J. Taillon, K. Collins (p.taillon@uwaterloo.ca, collinsk@uwaterloo.ca)

ABSTRACT

This project investigates the use of networked smartphones for distributed audio. The core application is to convert smartphones to loudspeakers, to which audio events can be transmitted over a wireless network in an array. The system has applications in business, education, gaming, and social environments. We report on initial experiments with the system, where investigation revolves around gauging the end-user experience. We also provide design suggestions for future social-audio games based on mobile technology.

1. INTRODUCTION

Smartphone sales in 2010 almost doubled those of the previous year, with Android-based units showing a startling 1,580% growth in the last quarter of 2010, as compared to the same period the previous year [1]. Moreover, smartphones have pushed the handheld gaming market into the mainstream, particularly as adults are no longer embarrassed to be carrying around a catalogue of games in the guise of a work-related device. This is evidenced from the results of a study conducted by DFC Intelligence (a US-based strategic market and research consulting firm) [2] that examined the mobile game habits of over 8,000 game players in North America and Europe, indicating 54% and 69% of respondents, in North America and Europe respectively, had played a game on their mobile phone in the past year. Furthermore, the mobile gaming market is expected to reach revenues of $11.7 billion dollars by the year 2014.

The processing power in the current smartphones rival that of traditional desktop computers allowing for 3D graphics, desktop-like Internet browsing, and HD video playback. Given a proliferation of mobile computing power in addition to the rising popularity of smartphones and gaming/multimedia-based applications, there is much need for further exploration in mobile multimedia, particularly with respect to audio, a commonly neglected area. This project explores the use of smartphone technology as computing units for ad hoc multimedia support, in professional, educational and personal entertainment context. The underlying environment assumes a close-proximity setting, typically encountered in a classroom, conference room, or social setting (e.g. a bar, café, or other area on the order of $15m^2$). The work presented here is the first of four phases encompassing our research programme. The objective of our research program is to explore the potential for networking smartphones for distributed audio-based applications:

1. using smartphones for audio support, e.g. for multimedia presentations;
2. audio playback using smartphones of synchronized tracks for social entertainment applications, providing opportunity for interactivity between participants;
3. exploration of the role smartphones can play interacting with a smart-table in gaming or education applications;
4. using smartphones to explore sound field effects created by physically moving the phones in a monitored environment.

This phase of activity details the system framework for networking phones for distributed audio applications. We present here the high-level view of the framework, describe the constituent components, and report on preliminary experimental results of our prototype used to create a simple sound array of a small number of smartphones. Experimental work revolves around gauging user experience with smartphone deployment in a confined space.

2. BACKGROUND AND PREVIOUS WORK

2.1. Wireless Networking

IEEE 802.11 defines a set of standards for implementing wireless local area network (WLAN) computer communication in the 2.4, 3.6 and 5 GHz frequency bands [3]. The typical WiFi cards (802.11/n) that ship with current computer equipment have an effective indoor range of approximately 100m and are characterized by a theoretical bitrate on the order of 100-210 Mbps [4]. Exchanging data over short distances is commonly accomplished using Bluetooth, a proprietary open wireless technology standard. The standard prescribes a wire-replacement communications protocol primarily designed for low power consumption, with a short range based on low-cost transceiver microchips. Bluetooth power classes dictate the effective range that the device can cover (Class 1: $\sim$100m; Class 2: $\sim$10m; Class 3: $\sim$1m). Data rates for Version 2.0 + EDR are on the order of 3 Mbps. Bluetooth devices can be aggregated into a small network called a *piconet*, where one node is designated the master device interconnecting with up to seven active slave devices.

2.2. Wireless Ad Hoc Networks and Localization

There has been a significant body of research related to sensor networks and generalized wireless mesh networks, consisting of simple (possibly mobile) compute nodes communicating over wireless network. The scenarios typically assume Global Positioning System (GPS) support for position identification [5], although cases where GPS is not available have been studied [6]; in the latter publication, inter-node distances are assumed to be large, on the order of 100m. In a small area, establishing a close-proximity positioning scheme presents a number of physical and technical issues, including interference from objects (not the least of which are human bodies), signal echo and reflexion off of surfaces, etc. [7, 8] Several WiFi triangulation methods have been proposed, but assume at least two hotspots are within range. Given its greater range, wireless transmission is a fairly power-intensive operation, resulting in higher power consumption [9, 10]. When nodes are within very close proximity, e.g. within a confined space, identifying nearest neighbours using standard GPS is not possible given its coarse accuracy. Several attempts to design and implement close-proximity GPS-type monitoring networks have met with mixed success, and generally rely on some form of specialized hardware support, e.g. [11]. Most approaches are based on measuring the relative signal strength between nodes to establish a distance approximation, e.g. received signal strength indicator (RSSI). Largely WiFi and Class 1 or Class 2 Bluetooth, provide too strong a signal in a confined area for such a technique to work accurately [8, 9, 10].

2.3. Audio Streaming Over Networks

A media stream can be streamed either *live* or *on-demand*. A live stream delivers the media data directly to a computer or output device for real-time playback, using a minimum amount of buffer storage: streaming audio begins playing after only a small amount of audio data is received, and the data is not stored permanently on the destination device. In contrast, on-demand streaming uses a progressive download scheme, whereby the media file is stored locally, e.g. to a target hard disk, and then played from that location. From a user perspective, play of the media stream can thus be started from the beginning, at any time.

Audio streaming and synchronization across mobile devices/audio arrays has been explored by Manvi and Venkataram [12], Tariq and Takeshita [13], and others with a variety of methods chosen. Bouillot, et al. [14] provide a useful overview of packetization and audio codecs, and compare some of the problems and provide suggested solutions for missing audio data, latency, jitter, and glitches. The limitations of loudspeakers on phones have been overcome with a variety of methods. Oh, et al. [15] describe using bass boost loudspeakers to assist the amplification of the mobile phone signal for their orchestra. Mars [16], Minnaar [17] and Oh [18] likewise explore alternate methods for boosting bass.

Audio streaming over a wireless network requires sufficient bandwidth and has time-critical considerations to avoid jitter and synchronization issues, amongst others. The Bluetooth Audio Video Working Group proposed the Advanced Audio Distribution Profile (A2DP) [19] to define the protocols and procedures for the distribution of audio content, where an audio source streams con-

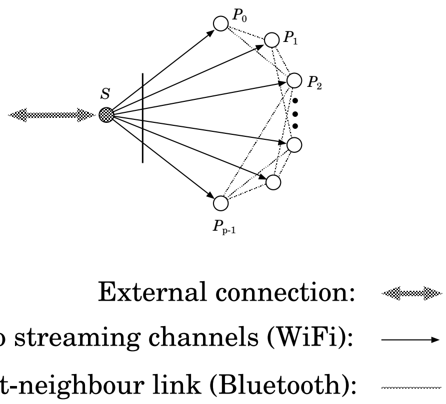

Fig. 1: Network configuration for media streaming and playback.

tent to devices such as headphones or speakers. The Audio Engineering Society generally recommend the WiFi 802.11 set of standards for meeting the performance requirements for streaming audio, while not recommending Bluetooth, given the inherent lower bandwidth capability and limited range [4]. Nonetheless, efforts have been made to mitigate the bandwidth constraints [20].

3. DESIGN CONSIDERATIONS

Our system is meant as a subordinate component, rather than a monolithic function. In this regard, we target consumer smartphones and handheld computer equipment, while adhering to the precept that the framework must not unduly consume resources of the phone. Our constraints are as follows:

- as much as possible, the framework should not rely on specialized hardware, beyond what is packaged in any off-the-shelf desktop/laptop and smartphone;
- the support software on the phone itself must present modest processing requirements, and the data storage should leave a minimum footprint;
- the system must impose low power demands on the phone battery.

4. SYSTEM ARCHITECTURE

4.1. Framework

The system framework consists of two networking subsystems:

1. media streaming from a server to the phones over WiFi;
2. a Bluetooth control network established between the phones, that enables initial position determination; information exchange; and network teardown.

A sound field is created using a cluster of smartphones. The field is generated by streaming specific audio to each phone, and requires a preliminary phase whereby the phones determine which stream they should play to produce the desired effect, based on their relative position. See Figure 1. A basic example is a two-phone cluster producing a stereo effect (i.e. one phone playing the left audio channel and the other phone playing the right channel). A laptop would use an RTSP server to stream the left channel over some port/channel, and the right channel over a separate port/channel. The system provides lightweight communication between nodes using the control network to minimize any impact on the quality of the audio playback, by dedicating WiFi resources to streaming. After the nodes determine their relative positions, minimal information needs to be exchanged. To create additional effects, the phones need only exchange short messages containing IP, URL, port, or channel information.

4.2. Smartphone Configuration

It is important to note that the smartphones and their native operating system in no way perform as a dedicated computing unit: many services execute in the background that can impact the overall performance of the phone (e.g., marketing and synchronization applications, orientation sensing, display management, amongst others). A constraint of our system is that each phone runs its RTSP client in a somewhat quiescent state, to ensure uniformity in the audio playback: this necessitates disabling as many services as possible that are not essential to basic system functionality. Note that for the tests described using our framework, these services were disabled via the user menus only, hence no special commands or privileges were required. In all other respects the system units tested (detailed below) were identical both in hardware, with the same memory capacity, processor model, etc. and software, operating system version, resident applications, etc.

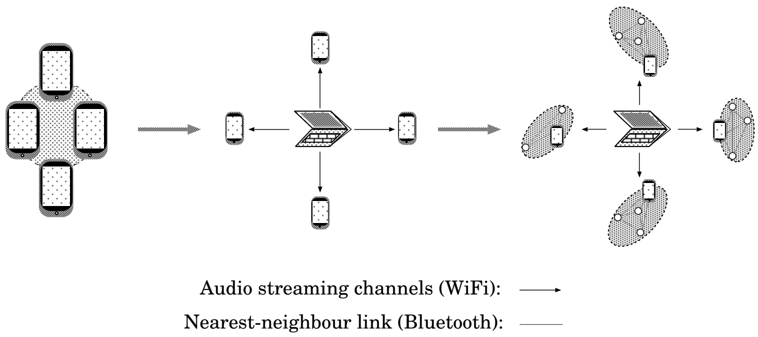

Fig. 2: Three phases of network initialization.

4.3. Server

We configured the WiFi/Bluetooth card on the server to act as a local hotspot, making it discoverable by the phones. A main consideration with audio streaming is the control and synchronization of the transmission and playback. The server on the laptop publishes one or more live audio streams that produce the sound field effect. We use the Real Time Streaming Protocol (RTSP) [21] to establish and control one or more time-synchronized streams of continuous audio media. The server operates in a manner oblivious to the number of phones participating (i.e. it simply broadcasts the streams over WiFi). When the server is initially launched, it uses a scheduling file that associates time/date information with the broadcasting of a particular stream. At the appropriate time, the server begins publishing the stream associated with an audio media file.

4.4. Phone Cluster

The algorithmic playback that produces the sound field is determined by the available-stream information in an application resident on each smartphones. Up to four of the phones are initially deployed in a close-proximity grid, whereby a form of near-field communication (e.g. native NFC or Bluetooth Class-3) can be used to determine nearest-neighbours by analyzing signal strength[1]. Once this relationship is established, the node software can determine an appropriate audio stream to receive and play, based on the intended effect (Figure 2). Using relative proximity, a phone identifies a particular port/channel associated with an audio stream published by the RTSP server; the phone invokes a RTSP client application that receives the stream and plays back over its own speaker.

[1]If necessary, these seed phones could be manually configured should the mobile operating system not support NFC.

4.5. Generalized Configuration

4.5.1. Smartphone Scaling

A more generalized solution, assuming the availability of a large number of phones, would allow the units to organize into four piconets, again, based on their proximity. The seed phones, identified as per Subsection 4.4, would be assigned the role of Master node to a specific cluster, and up to seven phones could join a particular piconet (see Figure 3). The algorithmic control of the sound field would be driven by the four Master nodes, each of which could push out different stream designations to its individual slave nodes, thus creating a heterogeneous playback environment for that subcluster. This can be accomplished by forwarding port/channel information over the control network to the slaves.

4.5.2. Server and Channel Scaling

The number of streams (potential audio channels) can be extended with additional hardware support. Assuming the availability of extra server machines, the scheduling scheme described in Subsection 4.3 can be used to publish a larger number of streams/channels, with the following two steps:

1. system clocks of the servers synchronize: this can be done, for example, by referencing a Network Time Protocol (NTP) server on one of the machines on the local network;

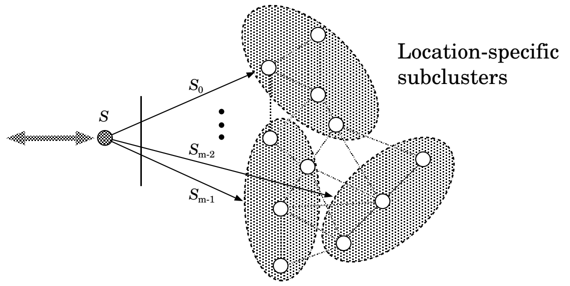

Fig. 3: Network configuration based on physical location.

2. the servers all reference a fixed, future start-time at which point publishing begins; the phone nodes need only possess a catalogue consisting of IP address, port, and channel designations for the specific streams.

5. EXPERIMENTAL RESULTS AND DISCUSSION

5.1. Setup

The server hardware consisted of a Toshiba Qosmio laptop, with Intel i7 processor, 6 GB RAM, built-in WiFi IEEE 802.11b/g/n and Bluetooth V2.1 + EDR, running the Windows 7 operating system. The cluster consisted of four Nexus One smartphones, each with a Qualcomm QSD8250 1GHz processor, 4 GB of memory, WiFi IEEE 802.11b/g and Bluetooth V2.0 + EDR, running the Android 2.2 operating system. The streamed media that constituted the sound field data was presented as audio files in WAV and MP3 format. The content of the media files were appropriately time-synchronized with respect to each other; i.e. each stream consisted of the instrumental tracks from a multi-instrumental musical piece. Playback on each phone was accomplished using the media player library native to the Android operating system. The experimental work was limited in this phase to four phones, although future iterations will extend this network. Android 2.2 does not support near-field communication. Given the limited number of phones, we manually designated one unit as a Master node, having the remaining three phones dynamically detect and connect with it, forming a small piconet (as described in Section 4.5.1).

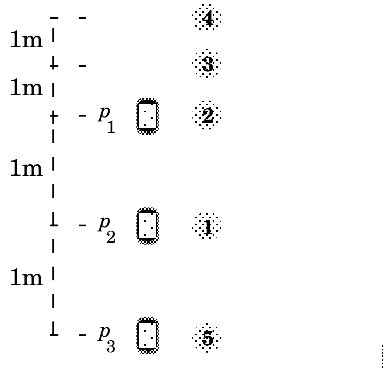

Fig. 4: Dilemma of sound level versus placement.

5.2. System Performance

Keeping in mind the factors described in Subsection 4.2, the RTSP server-client model performed well. In general the phones played in a consistently synchronous manner, occasionally exhibiting a slight flanging. Interestingly, this effect did not seem to impact our listening experience for the audio files we streamed: the content consisted of popular music (specifically samples of metal, rock and techno), for which flanging is sometimes used for aesthetic effect. In cases where an RTSP client on a phone was re-set, it was sometimes found that it resumed playback out of time with the other phones. There was an expected lag of approximately two seconds between the playback on the server and the sounds reproduced by the phone cluster. This is attributable to a combination of encoding and decoding delay; transmission overhead incurred by the phones; and a function of the slower processor speed. Playback on the server was only used occasionally to monitor the broadcasts, and so was not considered an artefact.

5.3. Audio Issues

5.3.1. Inter-phone Distances

Because a social audio environment assumes a bigger playback area than that for which the speaker was engineered, we decided to investigate the effects of phone placement in regard to maintaining sound level. For a simple experiment, we wished to determine at which distance the sound from a specific phone becomes inaudible due to intervening phone playback. We arranged three phones, p_1, p_2, p_3 in the configuration described in Figure 4, to which we streamed separate MP3 files with identical peak values. The phones were pre-set to the same sound level. We then attempted to gauge the distance at which streamed playback could no longer be meaningfully heard (more scientific measures of the amplitude will be forthcoming). During playback, we made the following observations at various locations in the arrangement:

- Position 1: the playback sound level of p_1, p_2, and p_3 are satisfactorily compatible;
- Position 2: the sound level of p_1 begins to dominate the sound produced by p_3 and to a lesser extent, p_2;
- Position 3: the sound from p_3 is almost completely masked by the sound from p_2 and p_3;
- Position 4: the sound from p_3 is inaudible;
- Position 5: we increased the volume of p_3, so that it became audible at position 4; returning to position 4, not surprisingly, we find the sound from p_2 is now almost completely masked by p_3.

We readily concede that this experiment does not represent a universal law, and is dependent on the audio stream content, from where it is broadcast in the sound field, etc. Nonetheless, it highlights the sensitive nature of generating a sound field with separate speakers, compounded by the relative positions of multiple listeners.

5.3.2. Orientation

The smartphone loudspeakers were designed to deliver sound consistent with that of telephone use (i.e. a limited frequency band). We investigated the impact of phone orientation on the sound level, with the understanding that this will change the overall sound field effect. The loudspeaker sound in our test units projected out of the back of the phone, and the microphone was located on the opposite side of the unit (i.e the front). We examined whether the sound level was significantly impacted by the speaker pointing towards or away from a listener, and found that there was no noticeable difference. On the other hand, we did notice differences in sound level if the phone was laying face-up or face-down, more dramatically if the surface was hard, e.g. a table-top, as compared to some absorbent material (e.g., cloth). In the latter case, we estimated that the sound level of a smartphone placed face up was decreased by as much as 30%. We will be testing these different scenarios with an SPL meter in the near future.

5.3.3. Sound Reproduction Quality

The sound playback on each phone is limited by the quality of the loudspeakers. In this case, the Nexus One test units produced an understandably poor listening experience. Loudspeakers on phones are generally designed to reproduce the mid-range frequencies (i.e. human voice) at the expense of having a wider frequency range. Given the size of the loudspeaker in smartphones, the reproduction of lower range (bass) frequencies is particularly problematic. Moreover, given the prevalence of smartphones as multimedia playback and gaming devices, we anticipate that future smartphones will greatly improve loudspeaker quality and aftermarket loudspeakers will likewise be able to fill a niche for audiophiles (the prevalence of add-on bass booster devices is testament to this supposition).

5.3.4. Audio Synchronization and Phasing Issues

The synchronization of audio files across an array of phones poses interesting challenges. While as mentioned, our use of the RTSP protocol has enabled the temporal synchronization of files, depending on the positioning of the smartphones, a variety of interactions between the sound playing through the phones may occur. We anticipate that users will not remain in a static location, and therefore there may be sound wave constructive or destructive interference and phasing effects. These problems have not been a consideration for the present prototype, but we conjecture further exploration with simultaneous microphone capture on the phones may help to alleviate some of the potential acoustical issues.

6. COMMENTS ON SOCIAL GAME AUDIO DESIGN USING SMARTPHONES

A smartphone playback environment is unique, unlike a fixed-position playback system such as Surround Sound 5.1, and presents significant audio engineering challenges from a end-user perspective (c.f. Subsection 5.3):

1. the loudspeaker arrangement is not necessarily known *a priori*, and phone nodes may be distributed unevenly in the auditory environment;
2. the phone loudspeakers themselves are meant to deliver sound accurately only a short distance; phone orientation can have an impact on the sound level and quality, depending on which direction the loudspeaker is pointed;

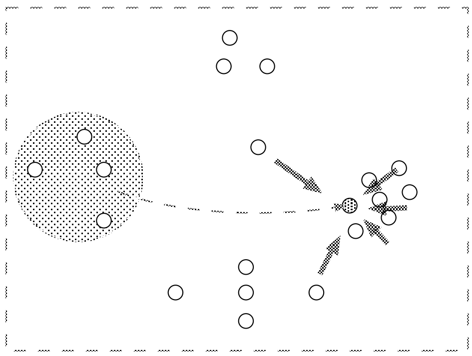

Fig. 5: Volume balancing issues: a listener can not hear a particular sound effect from a remote grouping of phones as a result of local sound interference.

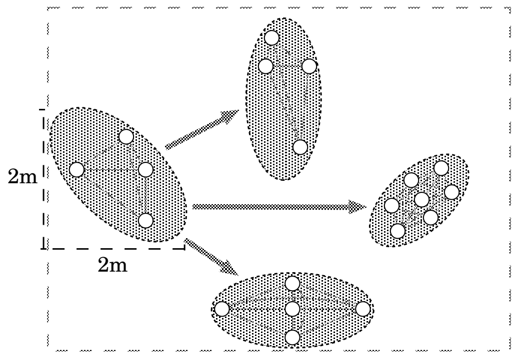

Fig. 6: Volume management through localization.

3. the social aspect of smartphone audio assumes typically more than one listener, and with respect to (1) and (2), as the distance between the phones increases, a consistent sound field effect becomes harder to maintain equally for all participants (e.g. see Figure 5); this situation is exacerbated if the phones are mobile.

A baseline observation is that smartphones cannot be used satisfactorily for coordinated playback, e.g. to produce a choral effect, given the inherent volume imbalances attributed to distance. A suggested design framework for any audio game would see localized playback within smartphones clusters of small proximity. The local playback effect, specified by servers/ports/channels, could then be communicated to other clusters of players, to share the listening experience (see Figure 6).

7. ADAPTATION TO COMPLEX SCENARIOS

7.1. System Improvements

7.1.1. Fault-tolerance and Synchronization

By its nature, live streaming synchronization problems are never a case of temporally playing too far forward in stream; rather the pathological case is lag. We observed some instances where, if an application interrupts playback for any reason, re-establishing the channel playback can result in it being slightly out of phase with respect to the other phones. To ensure greater fault-tolerance, a light-weight synchronization scheme can be implemented, whereby the phones attempt to re-calibrate their playback relative to other phones either explicitly, or at fixed time intervals. Most development media player libraries support functions that query the current play index and seek to a future point in the stream (limited of course by the amount data in the buffer). A simple algorithm would have a phone at playing time t send an index query to two randomly selected phones: it would take the average of the two returned index values, t_1, t_2, and compare the result to its own current playing position; if there is sufficient discrepancy, e.g. $(t_{avg} \leftarrow t_1 + t_2/2) - t \geq \Delta$ it would reposition it's index to the averaged value t_{avg} and continue playback. On each phone, a small system service component can query the reference phones over Bluetooth, calculate a playback index, and if necessary pass the adjusted index to the media player component which can reset its current index.

7.1.2. Sound Level Control

As discussed in Section 5, establishing a common sound level is crucial to create a uniform sound field. A valuable enhancement would be to include an initialization phase in which the phone applications negotiate a common sound level before beginning playback, using the Bluetooth network. An important caveat is that such a feature could only guarantee accuracy if the phones are identical models, given a lack of industry standard.

7.2. Adaptation of the System to Subsequent Project Phases

Interactive audio experiments can be investigated once the run-time configuration process of Section 4 has been completed (see Figure 7):

- by configuring the server accordingly, several audio streams can be made available to the phones;

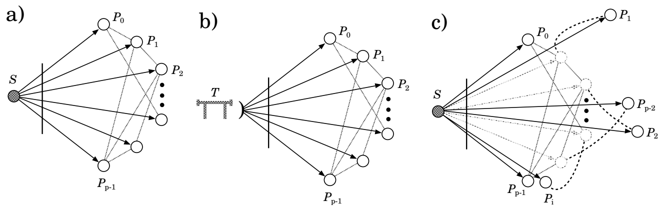

Fig. 7: Network infrastructure for project phases: a) Business/multimedia application, b) Audio-based game using smart-table, c) Social application and Educational game.

an application on each phone could present the user with a list of options corresponding to audio streams being broadcast. The participant could select the appropriate option based on their involvement with other users. For example, a social application could be developed whereby users are given or select one instrument sound in a song that has been separated into its individual channels (such as with the MXP4 interactive audio format). The users could then "mix" the song in real time by changing their physical location;

- peripheral computing devices such as smart-tables could be used to drive an educational or game application that incorporates smartphones, by providing server support for the audio component. Shirazi, et al. [22] have recently designed games that explore the combination of smartphones, smart tables and gesture recognition (tilting, throwing, and shaking the phone). One can imagine incorporating gesture-based interaction into a musical jam session whereby one person controls a type of sequencer at the tabletop and sends sounds or sequences out to individuals who then "play" those sounds/instruments;
- assuming an environment when the movement of the phone could be precisely determined in real-time, the phones could be used to play back streamed audio based on its location in a room, relative to objects, other phones, etc. This phase would require additional hardware and software, given the limitations outlined in Subsection 2.2.

8. CONCLUSION

This multi-phase research project explores the use of commodity smartphone clusters to enhance audio experience. Our prototype system framework demonstrates the feasibility and utility of the paradigm. Subsequent research phases will focus on exploring more complex audio media environments, and we hope to secure the use of a larger number of phones to fully test our piconet feature. We also anticipate investigating a simple synchronization scheme that will provide some level of fault-tolerance, in the event that one or more phones lose their audio stream and need to re-initialize their RTSP client. Results from preliminary experiments show the network and streaming strategy works well, although the sound quality from the loudspeakers has significant impact of the auditory experience. Future experiments would be better served incorporating small low-cost USB loudspeakers to augment the phones. Our next stage is to measure the amplitude differences as mentioned above, and to conduct subjective and objective user testing of the problems observed to determine their significance.

Our experiments indicate future game audio designers might consider using smartphone clusters rather than a single large bank of phones in order to avoid sound-level imbalances that occur with inter-phone distances.

ACKNOWLEDGEMENTS

The authors acknowledge the support of Google, Inc. in providing funding for this project.

9. REFERENCES

[1] Gartner Inc., "Worldwide mobile phone sales grew 17 per cent in first quarter 2010," May 2010. [Online]. Available: http://www.gartner.com/it/page.jsp?id=1372013.

[2] DFC Intelligence, "Gamers and the Usage of Mobile Phones," October 2009. [Online]. Available: http://www.dfcint.com/index.php.

[3] Standards for Information Technology, ISO/IEC 8802-11:1999, "Telecommunications and Information Exchange Between Systems–Local and Metropolitan Area Network - Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Higher-Speed Physical Layer (PHY) Extension in the 2.4GHz band," September 1999. IEEE Standards Office, NY, NY USA.

[4] AES, “Best Practices in Network Audio,” *Journal of Audio Engineering Society*, Vol.57, No.9, pp.729–741, 2009.

[5] I.F. Akyildiz, W. Xudong, “Wireless mesh networks: A survey,” in *IEEE Communications Magazine*, Vol.43, No.9, pp.S23–S30, 2005.

[6] Čapkun, et al., “GPS-free positioning in mobile ad hoc networks,” *Journal of Cluster Computing*, Vol.5, No.2, pp.157–167, 2002.

[7] A. Ault, et al., “*K*-Nearest-neighbor analysis of received signal strength distance estimation across environments,” in *Proceedings of the First Workshop on Wireless Network Measurements, WiNMee 2005*, Trentino, Italy, April 2005.

[8] X. Li, “Collaborative localization with received-signal strength in wireless sensor networks,” *IEEE Transactions on Vehicular Technology*, Vol.56, No.6, pp.3807–3817, 2007.

[9] E. Elnahrawy, et al., “The limits of localization using signal strength: A comparative study,” Santa Clara, CA, October 2004. [Online]. Available: `http://paul.rutgers.edu/~eiman/elnahrawy04limits.pdf`.

[10] L.-W. Chan, et al., “Collaborative localization: Enhancing WiFi-based position estimation with neighbourhood links in clusters,” in *Proceedings of the 4th International Conference on Pervasive Computing, Pervasive 2006*, Dublin, Ireland, pp.50–66, 2006.

[11] U. Bandara, et al., “Design and implementation of a Bluetooth signal strength based location sensing system,” in *Proceedings of the Radio and Wireless Conference, RAWCON 2004*, Atlanta, USA, pp.319–322, 2004.

[12] S.S. Manvi, P. Venkataram, “An agent based synchronization scheme for multimedia applications,” *Journal of Systems and Software*, Vol.79, No.5, pp.701–713, 2006.

[13] M.M.B. Tariq, A. Takeshita, “Management of cacheable streaming multimedia content in networks with mobile nodes,” in *Proceedings of Global Telecommunications Conference, IEEE GLOBECOM 2002*, Taipei, Taiwan, pp.2245–2249, 2002.

[14] N. Bouillot, et al., “Performance metrics for network audio systems: Methodology and comparison,” AES Convention 127, Oct. 2009. [Online]. Available: `http://www.aes.org/e-lib/browse.cfm?elib=15134`.

[15] J.J.H. Oh, et al., “Evolving the mobile phone orchestra,” in *Proceedings of the International Conference on New Interfaces for Musical Expression, NIME 2010*, Sydney, Australia, pp.82–87, 2010.

[16] P. Mars, “Pump up the Volume: Enhancing music phone audio quality and power using supercapacitors for power management,” AES Convention 123, October 2007. [Online]. Available: `http://www.aes.org/e-lib/browse.cfm?elib=14264`.

[17] P. Minnaar, “New algorithm designed to enhance low frequencies in open-fit hearing aids,” *The Hearing Journal*, Vol.63, No.3, pp.40–44, 2010.

[18] J.J.H Oh, “Full digital amplifier for mobile and handheld devices,” AES 29th International Conference, Seoul, South Korea, September 2006. [Online]. Available: `http://www.aes.org/e-lib/browse.cfm?elib=13860`.

[19] Bluetooth Audio Video Working Group, “Advanced Audio Distribution Profile Specification,” May 2003. [Online]. Available: `https://www.Bluetooth.org/Technical/Specifications/adopted.htm`.

[20] K. Selvaradjou, et al., “Optimization of Bluetooth audio stream based on the estimation of proximity,” *International Journal of Computer and Electrical Engineering*, Vol.2, No.3, pp.550–554, 2010.

[21] Internet Engineering Task Force, “Real Time Streaming Protocol (RTSP),” April 1998. [Online]. Available: `http://www.ietf.org/rfc/rfc2326.txt`.

[22] A. S Shirazi, et al., “Poker surface: Combining a multi-touch table and mobile phones in interactive card games,” in *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services, MobileHCI 2009*, Bonn, Germany, September 2009.