Time Synchronization in Acoustic Localization for Mobile Open-Source Network Deployment

Scott Almquist

Worcester Polytechnic Institute

Major Qualifying Project

18 December 2009

Embedded Digital Systems – Group 102

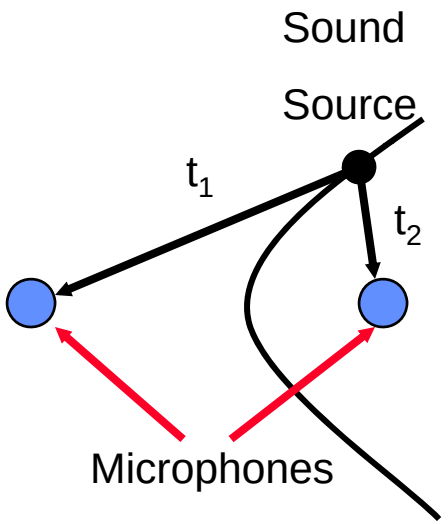
Approved for public release - distribution is unlimited

MIT Lincoln Laboratory

This work is sponsored by the Department of the Air Force under Air Force contract FA8721-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the author and are not necessarily endorsed by the United States Government.

Acoustic Localization

- **Determining the location of a sound source by using an array of sensors**
- **The difference in arrival times is used to calculate an approximate position of the sound source**



Time Differences of Two Microphones Result in a Hyperbola

Setup

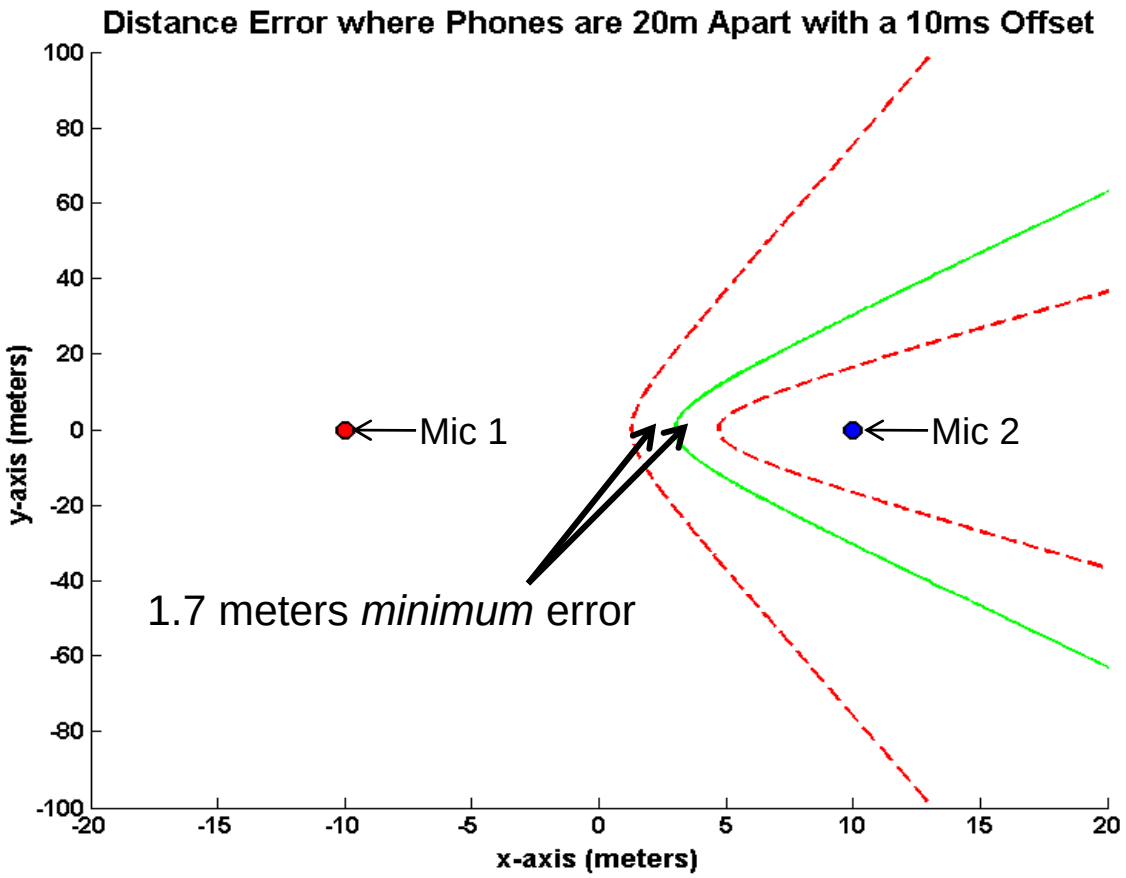
FWG114pv2 image. Retrieved October 13, 2009, from http://kbserver.netgear.com/images/fwg114pv2.gif.

Openmoko freerunner image. (2008). Retrieved October 13, 2009, from http://www.gadgetarena.com/gadget-content/uploads/2008/07/openmoko-neo-freerunner.jpg.

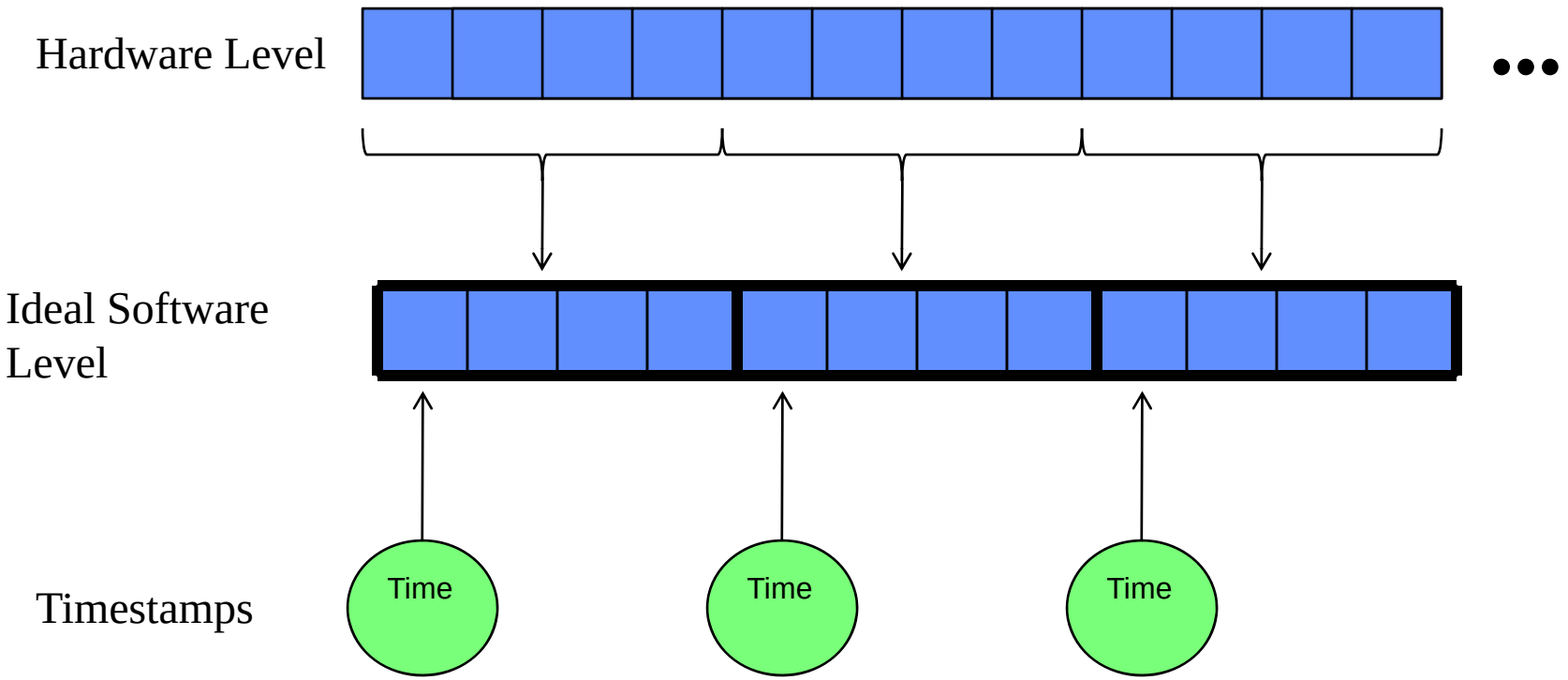
Dell inspiron image. Retrieved October 13, 2009, from http://images.bilsimser.com/dell_inspiron.jpg.

Time Synchronization

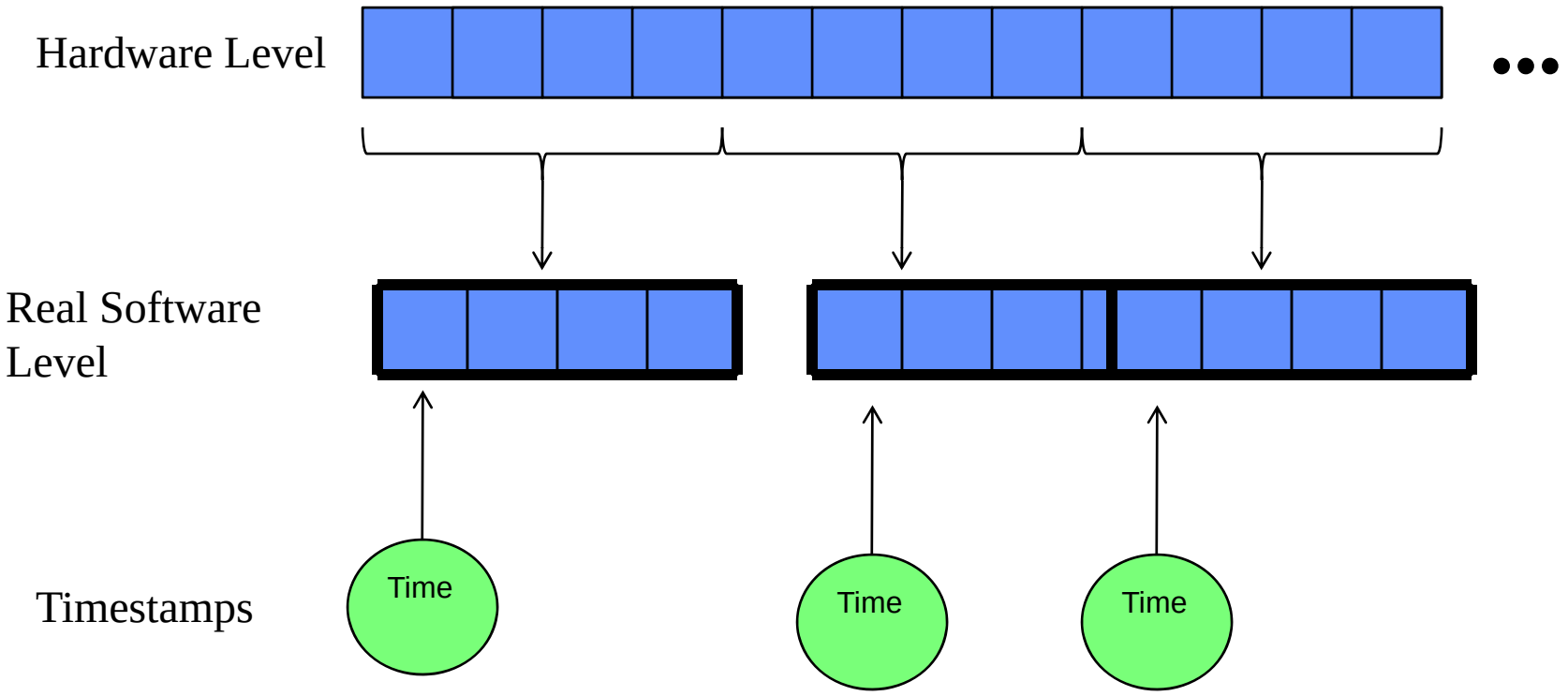
Time synchronization errors are the greatest source of positional error



Audio
Data and Timestamps

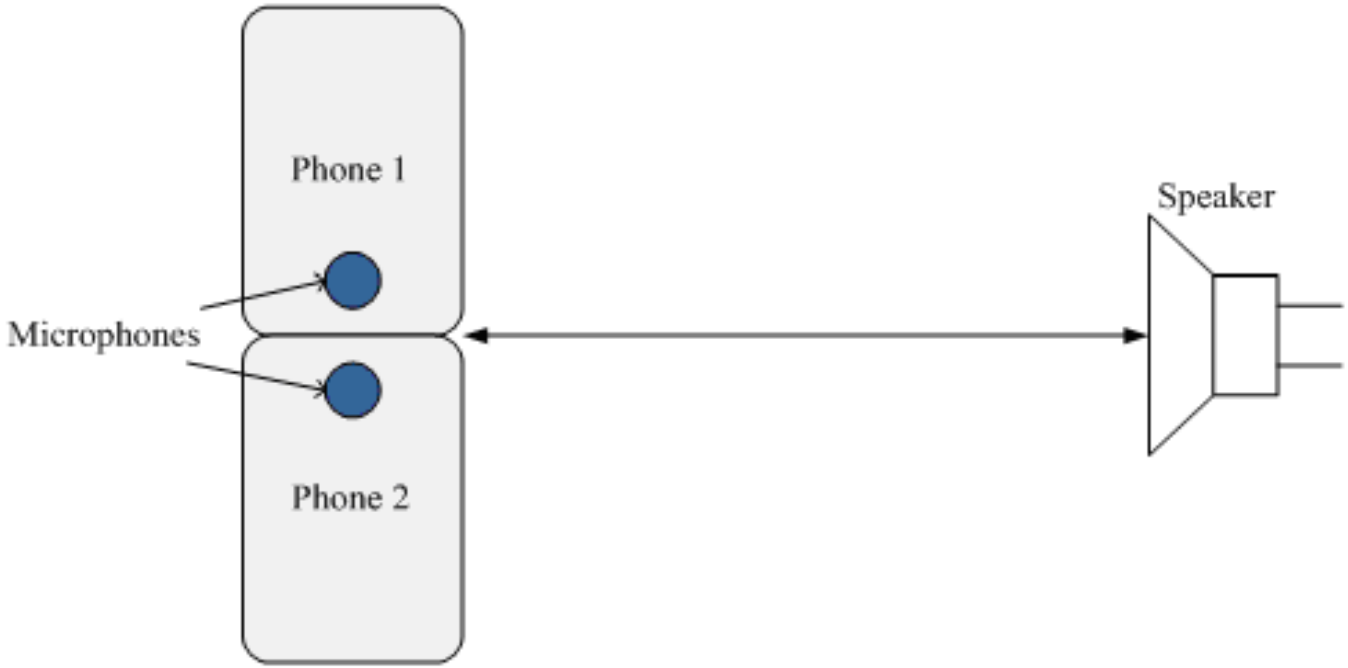


Audio
Data and Timestamps

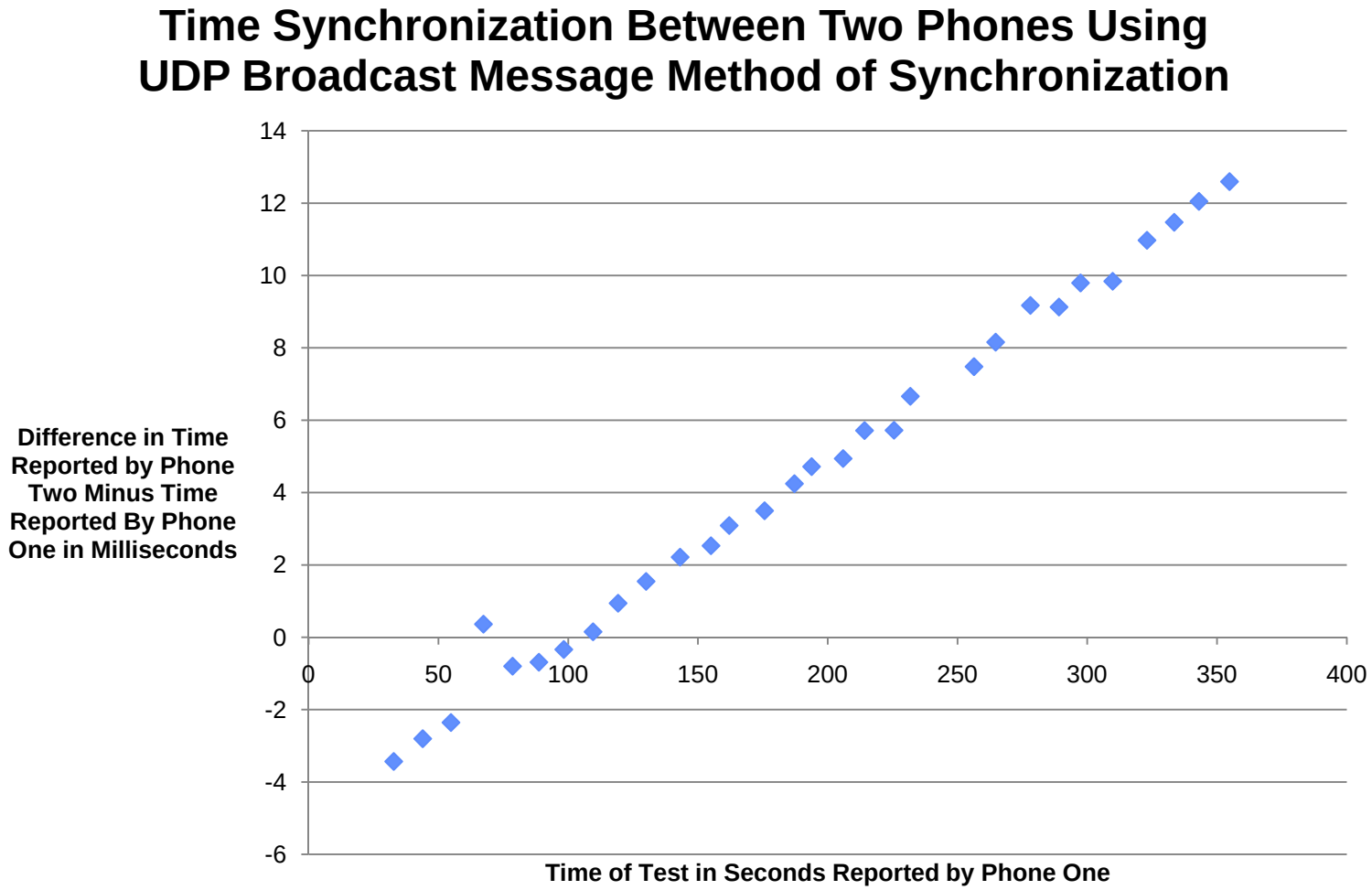


Time Synchronization Test Setup

If the phones are equidistant from the sound source, the time difference of arrival should be zero



Using a UDP Broadcast Message to Synchronize



Outline

- Goal
- Recording Audio Data
- Precision Timestamps
- Filtering Timestamps
- Synchronize the Phones
- Conclusions and Future Work

Outline

- **Goal**
- Recording Audio Data
- Precision Timestamps
- Filtering Timestamps
- Synchronize the Phones
- Conclusions and Future Work

Goal

Improve the Time Synchronization Between Two Phones

Can be sub-divided into two categories:

Precision in retrieving an individual phone's clock
Synchronization between two phones

Outline

- Goal
- **Recording Audio Data**
- Precision Timestamps
- Filtering Timestamps
- Synchronize the Phones
- Conclusions and Future Work

Recording All Audio Data

- In order to improve the results further, we need to know that all audio data is being recorded

- Initiation tests indicate that this is not an unreasonable assumption
  - Over a 549.04 seconds we would expect 24,212,664 samples.
    - Phone One sent 24,215,552 samples (1.1% more than expected)
    - Phone Two sent 24,217,600 samples (2.0% more than expected)
  - Playing a sine wave and viewing graphs of the recording results in no obvious discontinuities or phase shifts.

It is assumed that audio data being recorded is consistent and timestamps are imprecise

Outline

- Goal
- Recording Audio Data
- **Precision Timestamps**
- Filtering Timestamps
- Synchronize the Phones
- Conclusions and Future Work

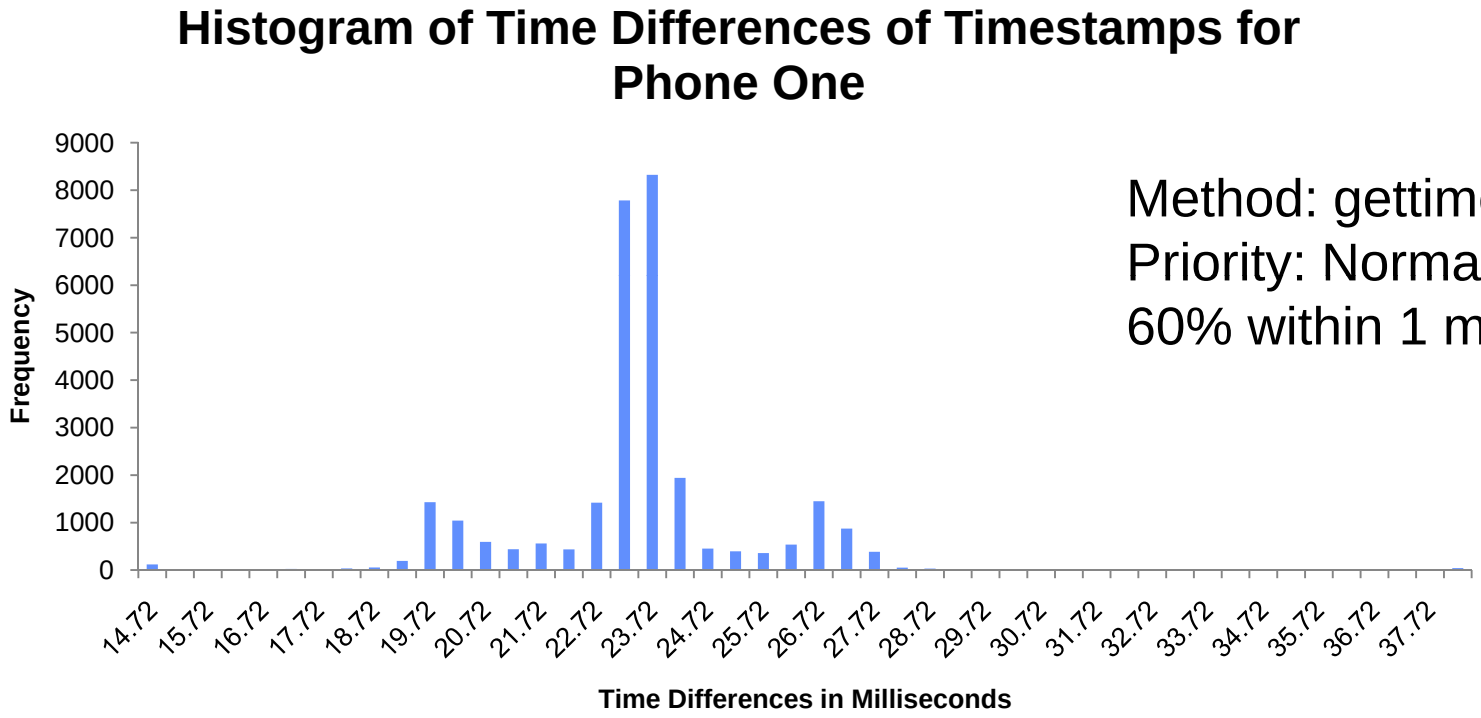
Precision in Retrieving Timestamps

- **Audio is processed in buffers**
- **Buffers are a consistent size (1024 Samples)**
- **Sampling frequency is a constant (44,100 hertz)**

$$\frac{1024\ \text{Samples}}{44{,}100\ \frac{\text{Samples}}{\text{Second}}} = 23.22\ \text{Milliseconds}$$

One measure of the precision of an individual phone's timestamps is a consistent 23.22 millisecond difference between successive timestamps

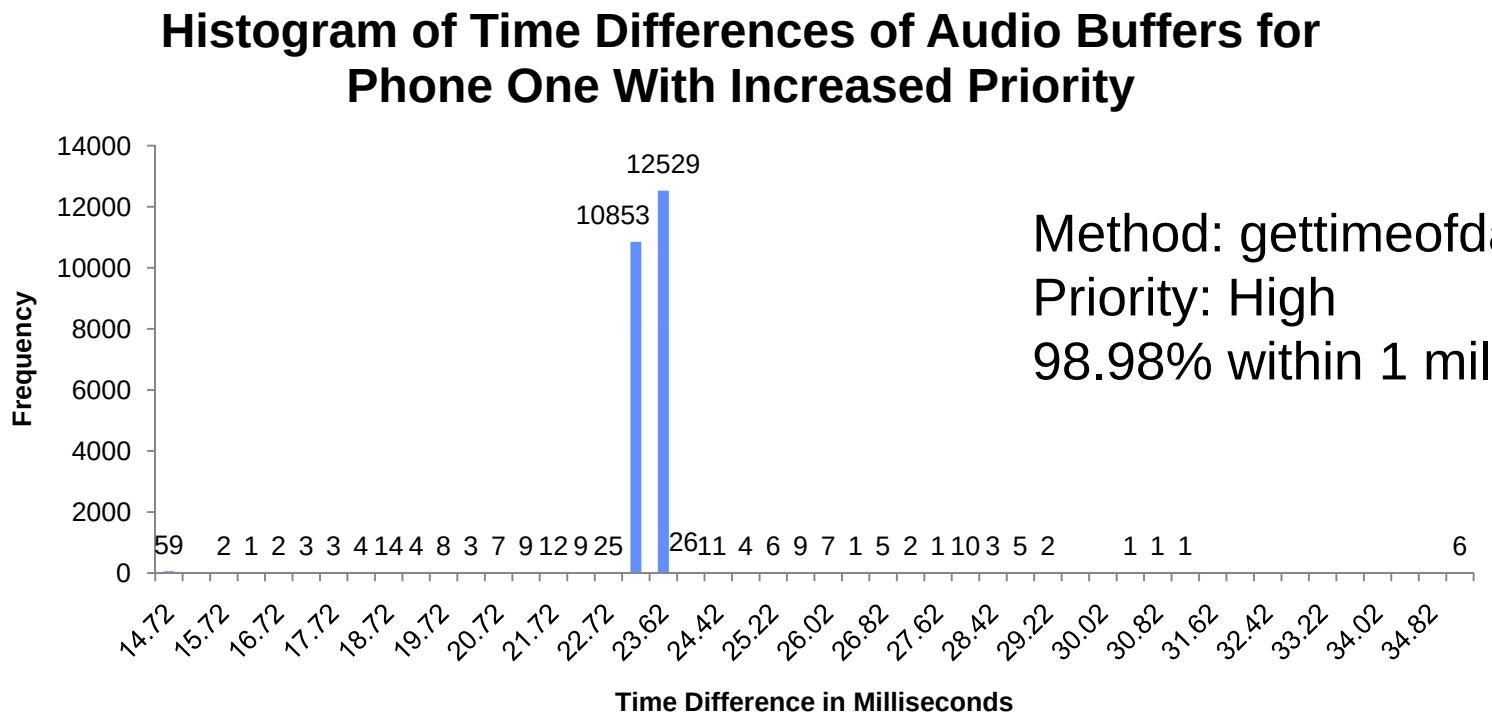
Consistency of Timestamps



Method: gettimeofday
Priority: Normal
60% within 1 millisecond

Consistency in the ALMOND project was poor

Consistency of Timestamps



Method: gettimeofday
Priority: High
98.98% within 1 millisecond

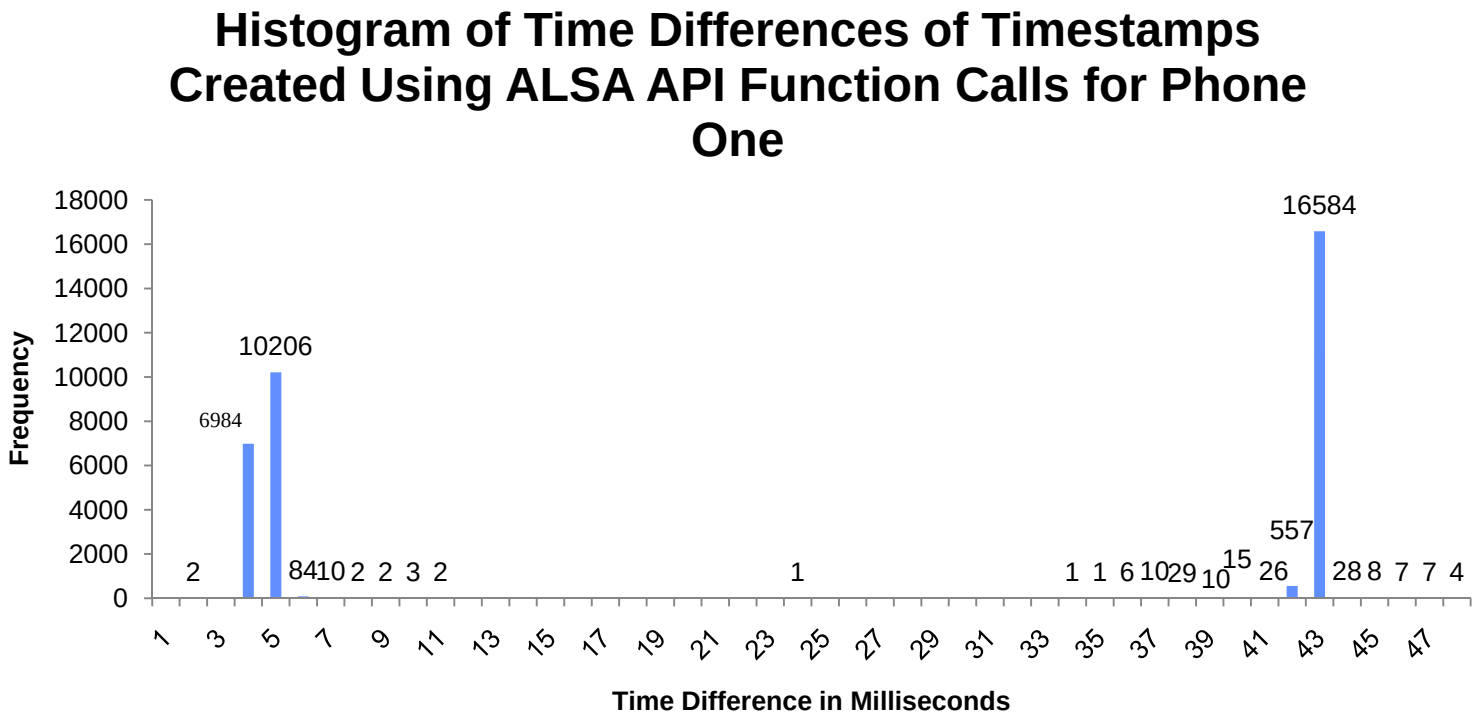
Increased priority was required for good results

Advanced Linux Sound Architecture

- Audio library and kernel level API
- Default audio library for Openmoko
- Used in ALMOND project to record audio

ALSA provides function calls to retrieve timestamps

Consistency of Timestamps

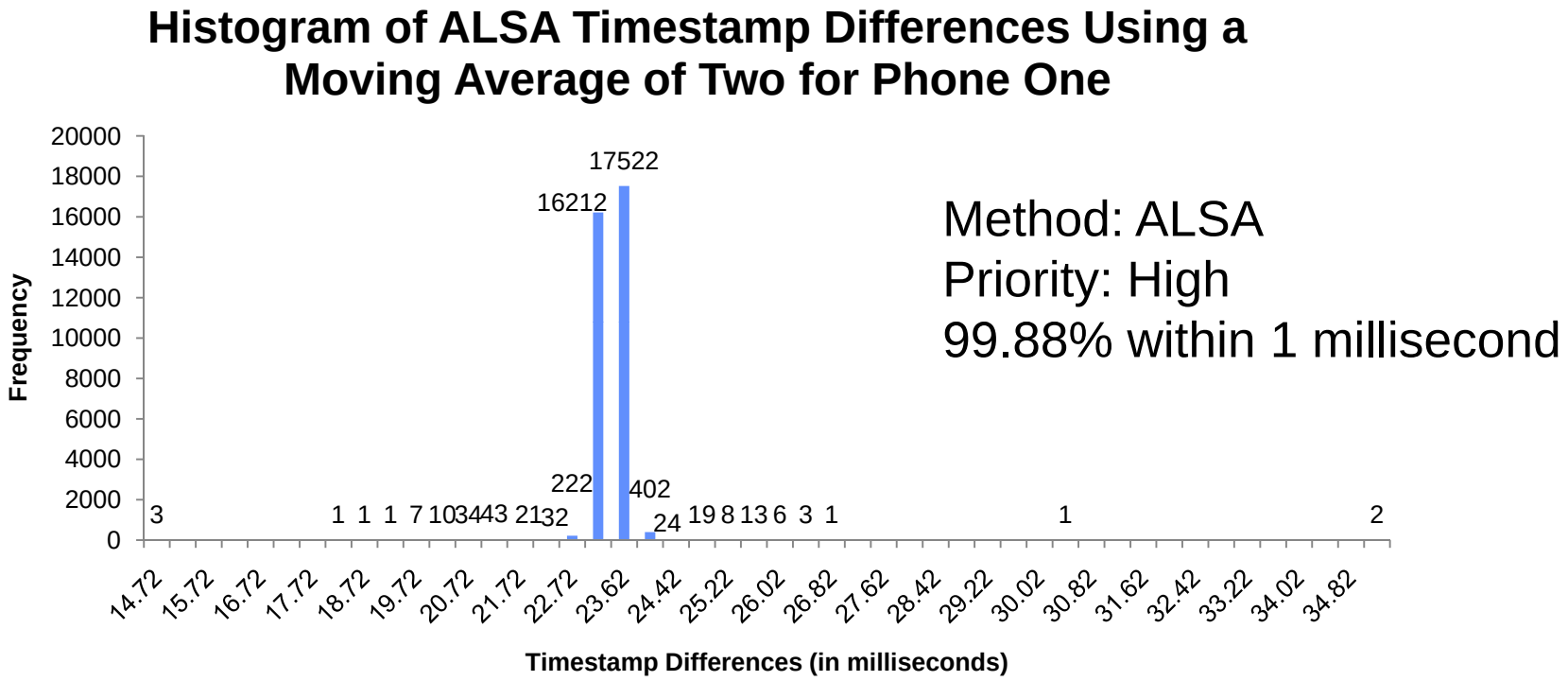


Results are bimodal, but it looks like they average to the correct value

Outline

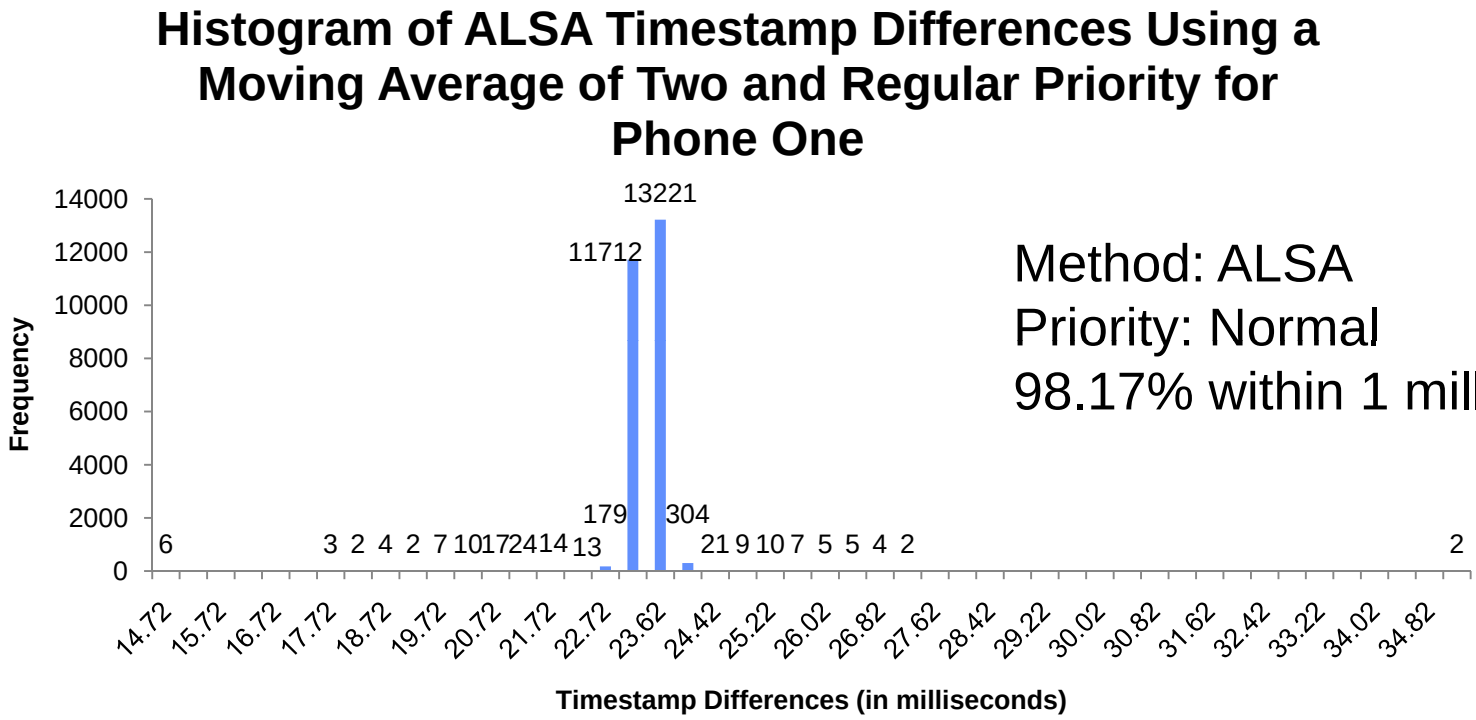
- Goal
- Recording Audio Data
- Precision Timestamps
- **Filtering Timestamps**
- Synchronize the Phones
- Conclusions and Future Work

Using a Moving Average



Using a moving average results in expected time differences

Can We Lower the Priority?

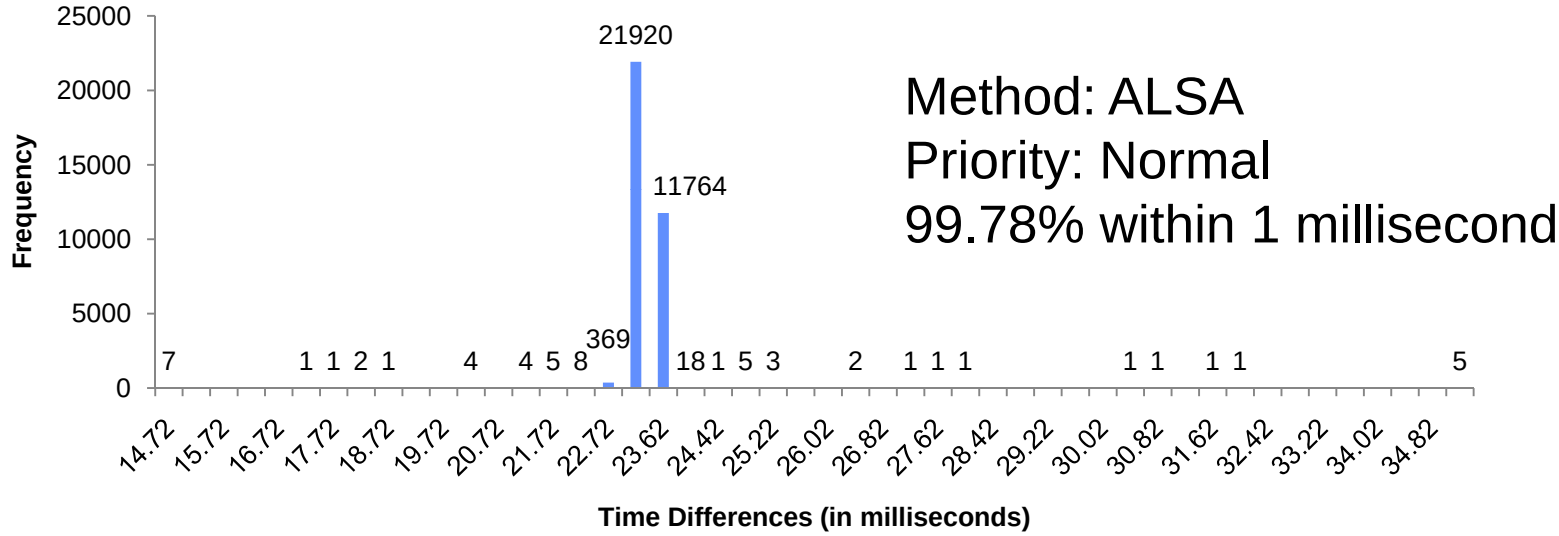


Method: ALSA
Priority: Normal
98.17% within 1 millisecond

ALSA timestamps allow us to lower the priority and achieve similar results

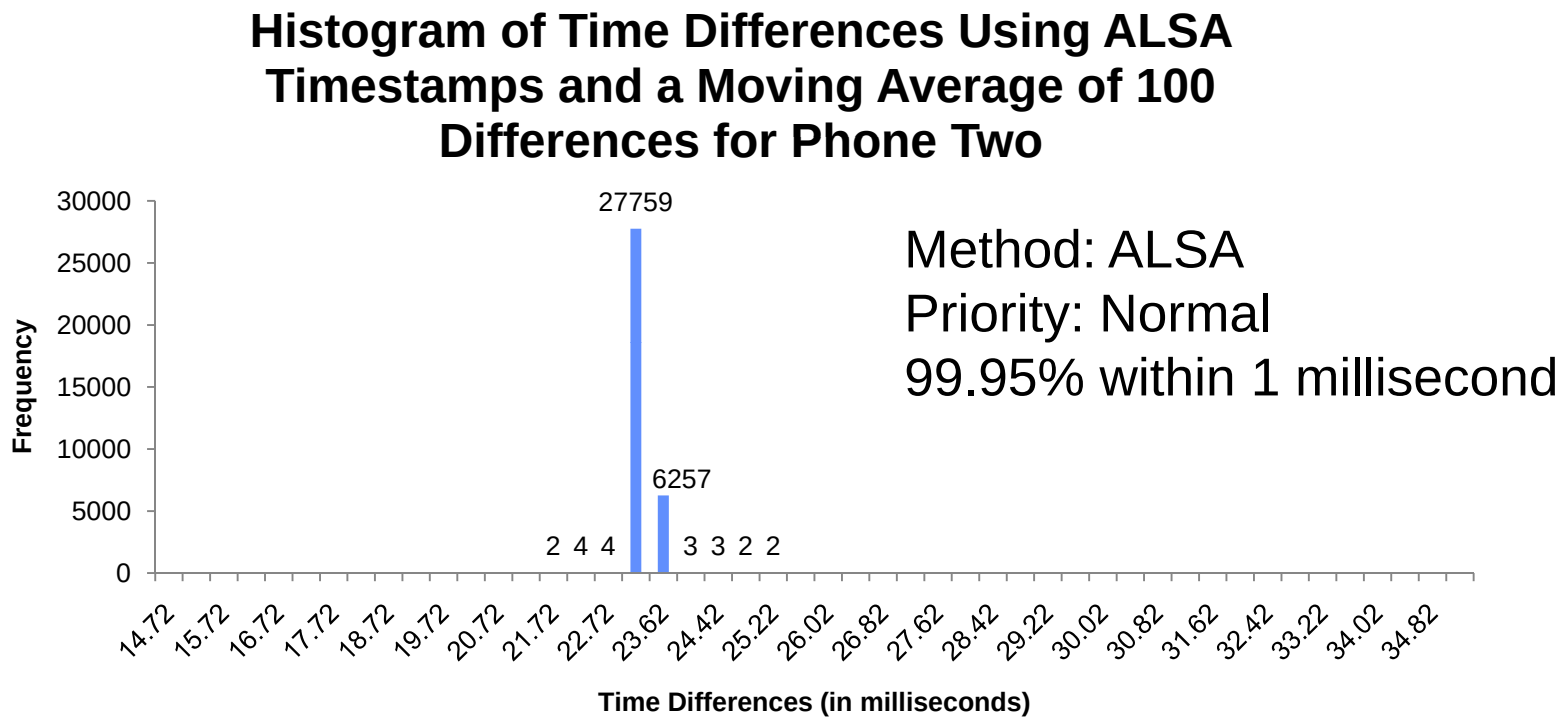
Moving Averages

Histogram of Time Differences Using ALSA Timestamps and a Moving Average of 10 Differences for Phone Two



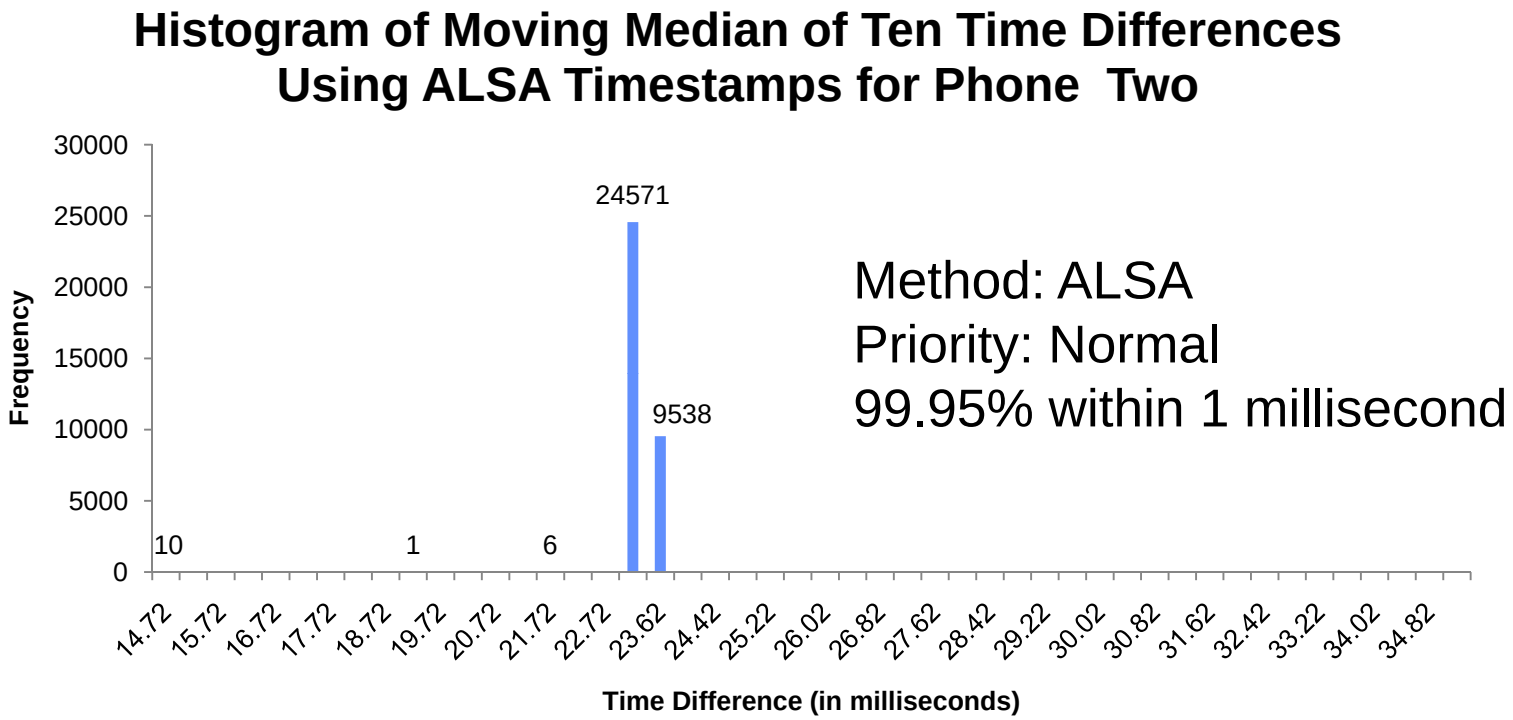
Moving averages improve the results, but outliers are still present

Moving Averages



Moving averages can be extended for better results

Moving Medians

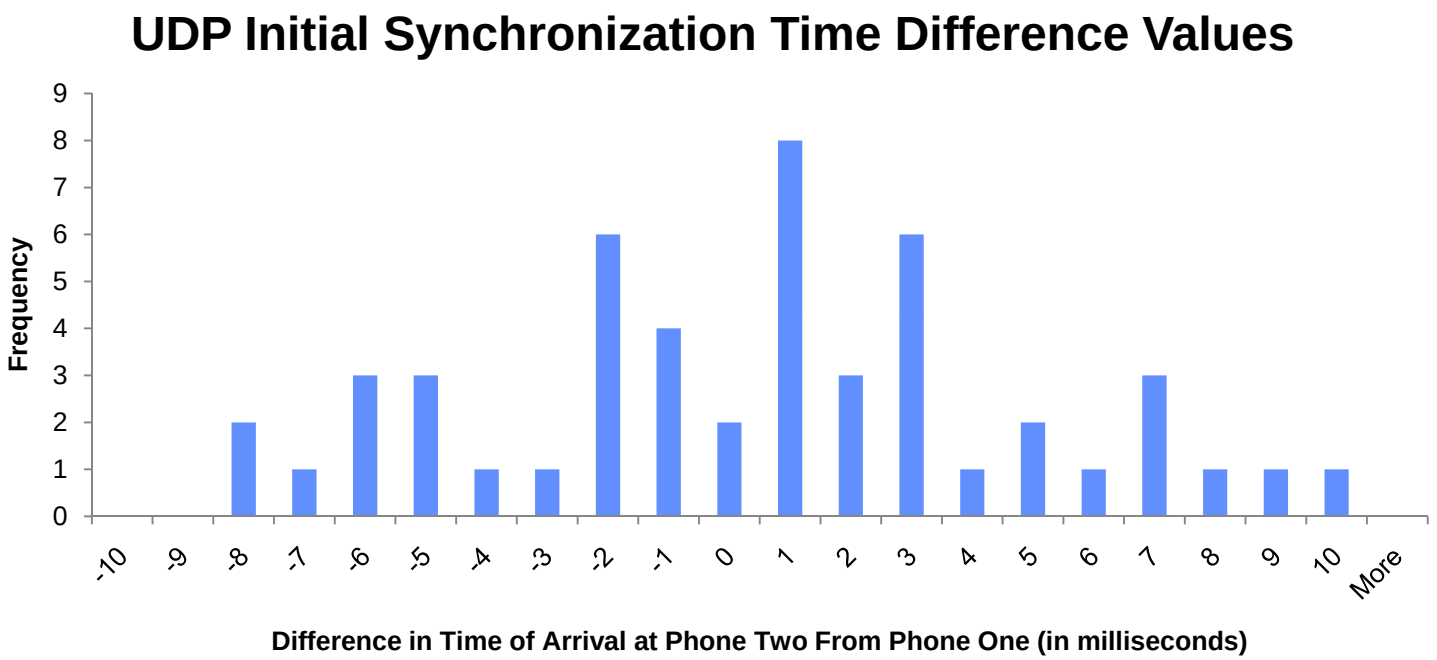


Moving medians have good results with fewer points

Outline

- Goal
- Recording Audio Data
- Precision Timestamps
- Filtering Timestamps
- **Synchronize the Phones**
- Conclusions and Future Work

UDP Initial Synchronization



Average UDP Initial Synchronization is -0.0733 Milliseconds

Outline

- Goal
- Recording Audio Data
- Precision Timestamps
- Filtering Timestamps
- Synchronize the Phones
- **Conclusions and Future Work**

Conclusions

Consistency of Retrieving Time on an Individual Phone

Method of Retrieving Timestamp	Priority	Moving Average Moving Median	Percent of Time Differences within 1 Millisecond of Expected Value
gettimeofday	Normal	None	60
gettimeofday	High	None	98.98
ALSA	High/Normal	None	<1%
ALSA	High	Moving Average of 2	99.78
ALSA	Normal	Moving Average of 2	98.17
ALSA	Normal	Moving Average of 10	99.78
ALSA	Normal	Moving Average of 100	99.95
ALSA	Normal	Moving Average of 2 Moving Median of 10	99.95

Conclusions

UDP Synchronization can achieve an average synchronization of 0.017 milliseconds between the phones.

Future options for synchronization between phones include:

- **UDP Synchronization**
- **GPS (Not with the Openmokos)**
- **NTP**

Future Work

- Prove that all audio data is being retrieved
- Synchronize time between phones
- Are timestamps even necessary?

Acknowledgements

- **ALMOND Partners:**
  - **Daniel Skehan**
  - **Muhammad Saleem**

- **WPI Advisors:**
  - **Professor Clancy**
  - **Professor Heineman**

- **Lincoln Laboratory Advisors**
  - **Albert Reuther**
  - **Glenn Schrader**