

Audio-WiFi

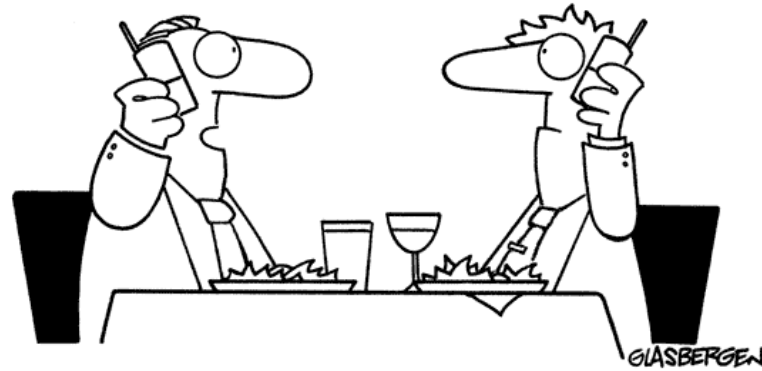
Audio-WiFi: Power Saving mechanism for 802.11
By Mostafa Uddin

Audio-WiFi: Motivation

- WiFi is a common communication interface for smart devices.
- WiFi still has some perturbation such as,
 - Poor utilization of wireless channel.
 - Energy consumption during idle state.
 - Unfairness issue due to capture effect.
- Additional channel can be utilize to enhance the performance of Wi-Fi network.
- Smart device can have following interfaces as additional channel:
 - WiFi
 - Bluetooth
 - Zigbee
 - Light/Camera
 - Audio[Can we utilize this interface?]

Audio-WiFi: Main Idea

*Utilize audio channel as an augmented channel to enhance
WiFi performance*



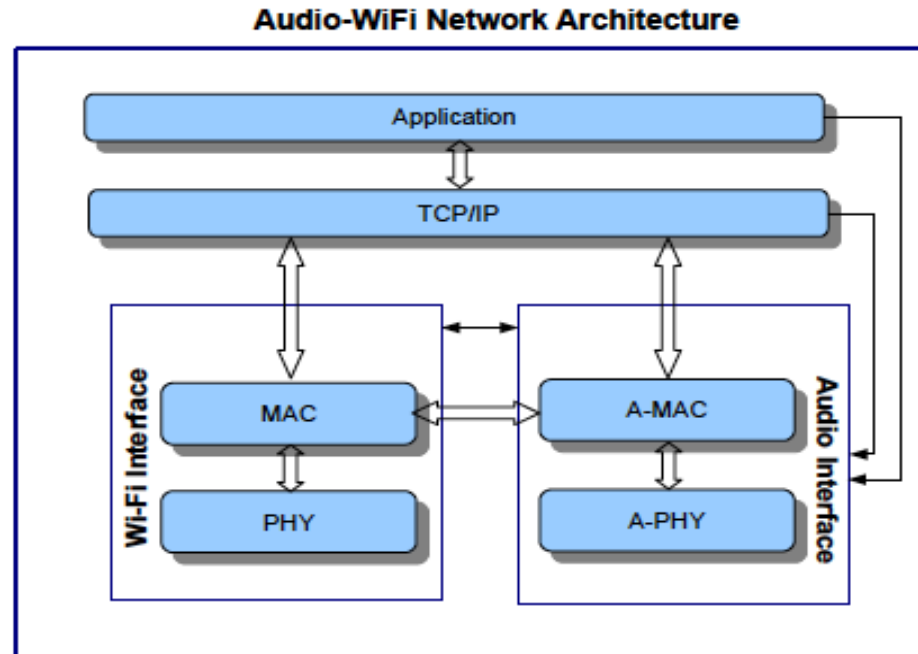
We like to exploit audio frequency beyond human ear perception as a parallel communication channel with WiFi.

Why Audio communication?

- Non-interferential with radio network.
- No additional bandwidth required from WiFi.
- Speaker/Microphone are very common hardware component in smart device.
- Smart devices are capable of generating and discerning audio frequency beyond human ear

Audio-WiFi: Network Architecture

Preliminary Architecture of proposed Audio-WiFi network



A-PHY: responsible for signal processing and sending/receiving signal using ,mic/speaker.

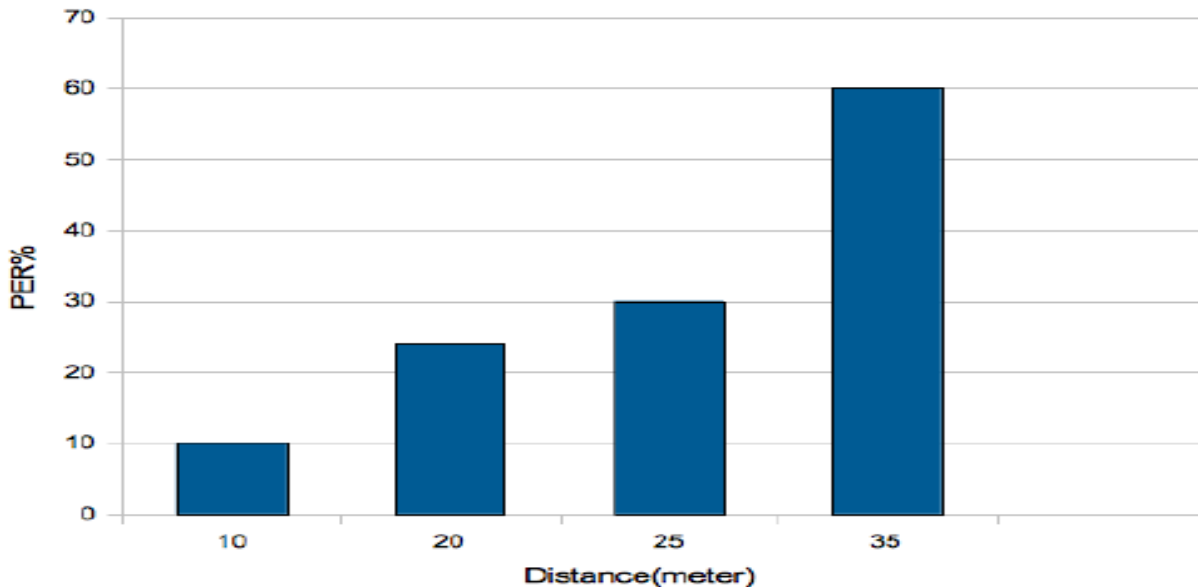
A-MAC: Can be utilized by MAC and TCP/IP layer to send small size data packet over audio channel.

TCP/IP and MAC has control path with A-MAC.

Audio-WiFi: Feasibility

Sending/Receiving data frames over audio channel

- M-array FSK modulation/demodulation.
- We use 16 frequencies for our modulation/demodulation.
- Each frequency represents a symbol of 4 bit.
- Frequency range from 18000-21200Hz.
- Equal frequency spacing.
- 30bps as data transmission rate.
- Frame size is 25byte.



Packet Error Rate(PER) over different distance

Audio-WiFi: Challenges

Challenge1: Audio channels suffer from low data rate.

Possible Solution:

- *Use audio channel to transmit only small control frames.*
- *Use different audio tones instead of actual bits for control frames.*

Challenge2: Frame-level synchronization between WiFi and audio.

Possible Solution:

- *Use single audio frame for aggregated WiFi frames.*

Audio-WiFi: Ongoing Work

- *Utilizing audio channel to enhance the performance of Power Save(PS) mechanism for 802.11.*
- *Using audio channel as an control channel for sending ACK frames while WiFi is sending data frames.*
- *Utilizing audio channel for coordinating between node to reduce the collision.*

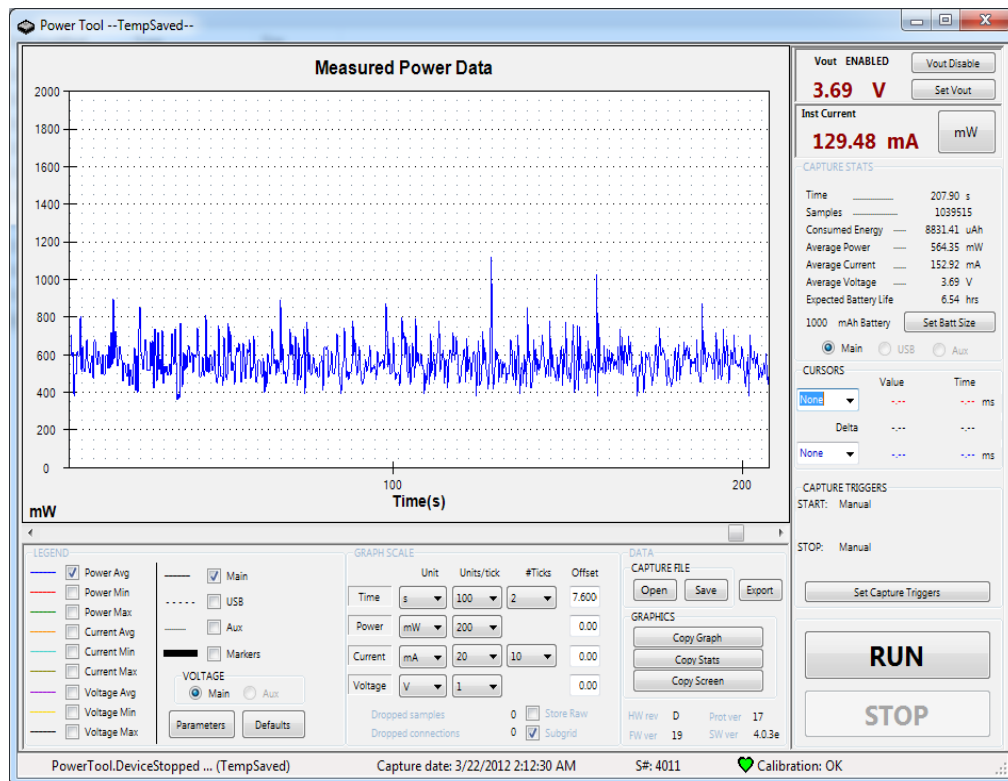
MonSoon Hardware



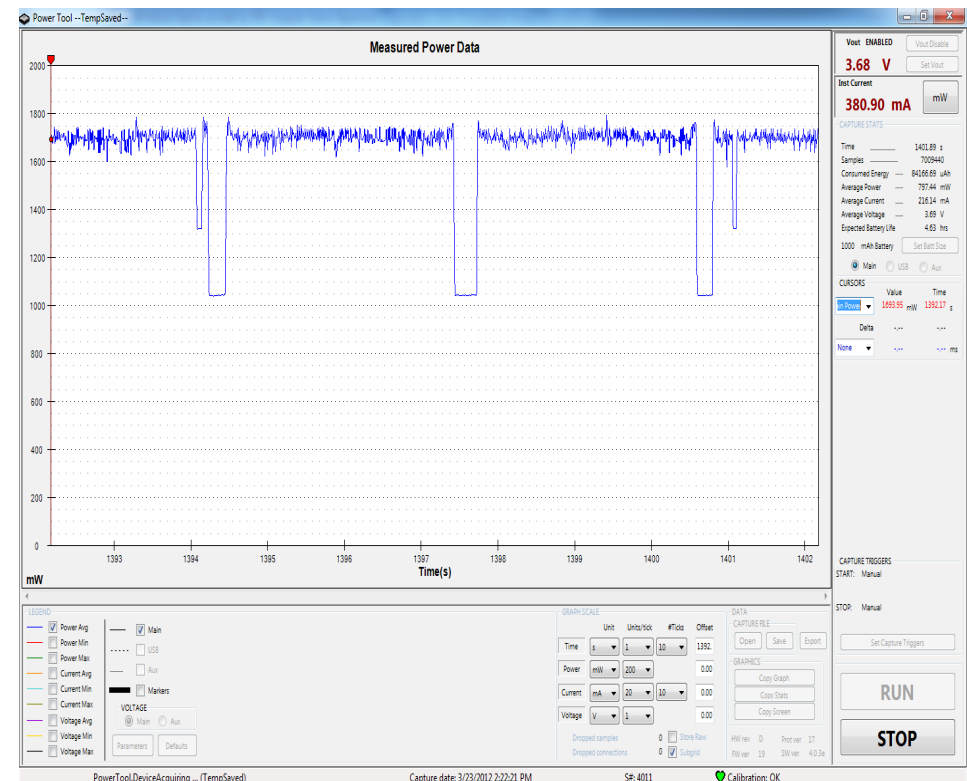
Energy Measurement

We monitor the power consumption of Audio Interface and WiFi Interface under the following configuration in Nexus S phone:

1. Phone is in Airplane mode.
2. Either audio or wifi interface in active (receiving data)
3. Screen of the phone is turned off.



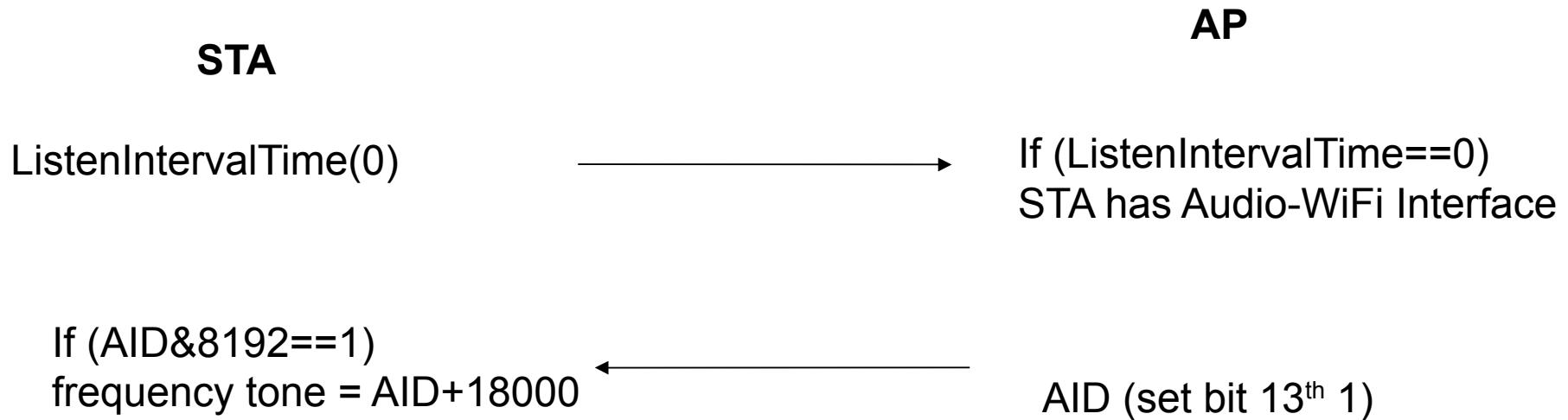
Audio Interface is receiving data
(Energy consumption~600mW)



WiFi Interface is receiving data
(Energy consumption ~1700mW)

Audio-WiFi PS Mode

Association mechanism



Power Management

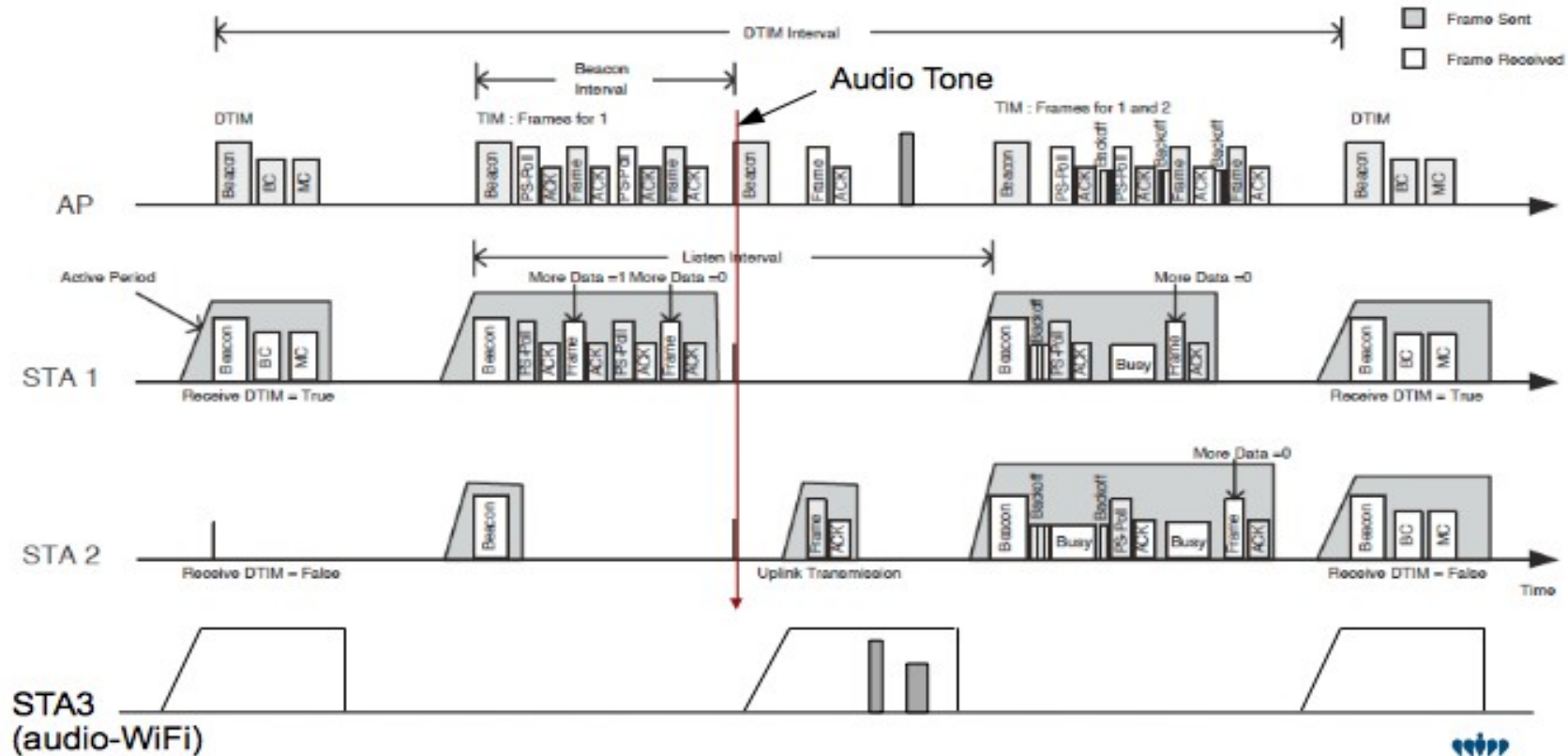
- STA turn on the audio interface instead of the wifi interface in each beacon time when it moves to Audio-WiFi PS mode.
- AP will send an audio frequency tone while it has packet for Audio-WiFi STA.

Challenge: Need to determine and minimize the length of the frequency tone.

- STA turn on the wifi interface when it receives certain frequency tone from the AP.

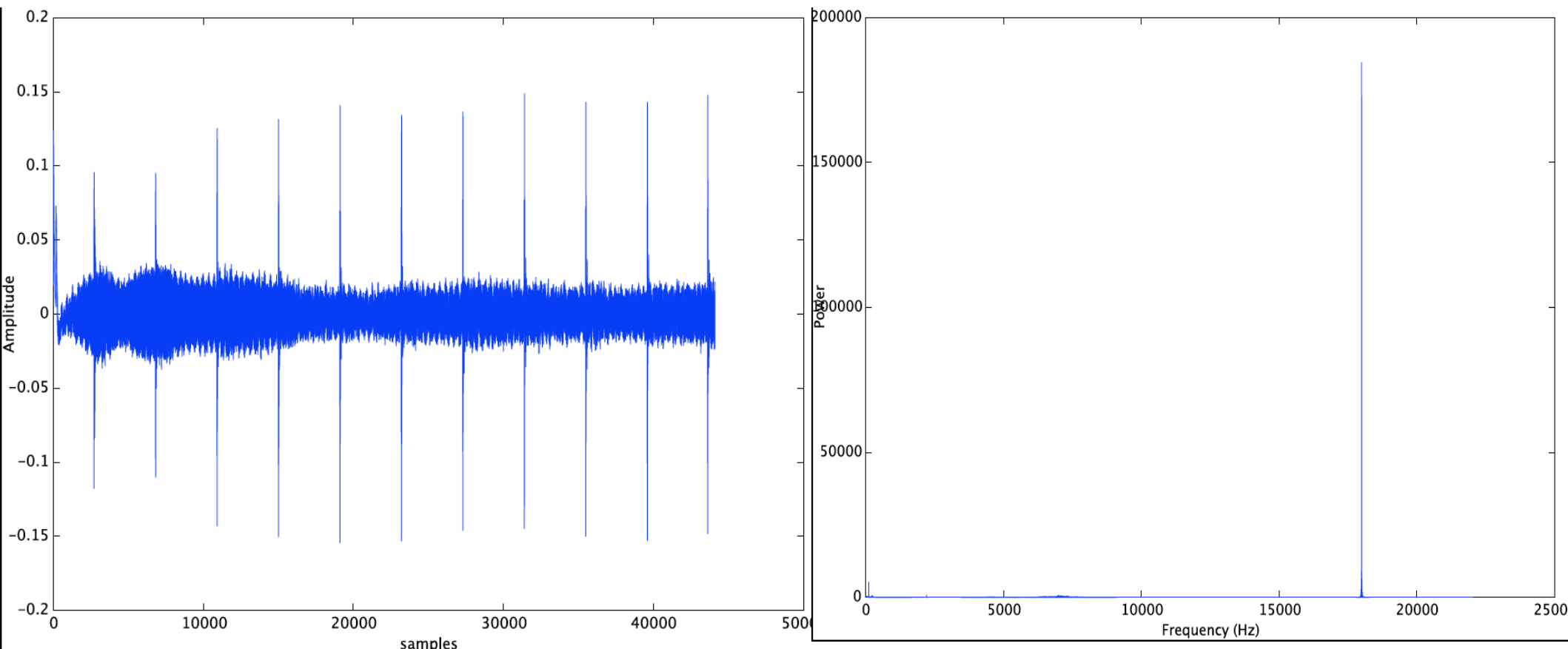
Audio-WiFi PS Mode

PS-Poll mechanism



Beacon Sound

- Usually beacon rate is 10/per second
- Generating beacon sound at that rate and detect that.



Driver replacement

- Using Two Android Phone (one as AccessPoint)
- Replace the existing Broadcom driver(bmc4329) with Texas Instrument Driver(wl1271). [Currently Working]
- If it does not work we will use N900 which already has (wl1271 driver)
- Incorporate the audio tone generator with wl1271 driver. (calling **tone generator module** from wifi driver module)

Driver Implementation challenges

- In kernel level we don't have API to generate sinusoidal values.
- Sound Driver does not write sound data to hardware unless minimum amount of data has been buffered.
- Generating sound beacon within 50 ms of beacon starting period time.

Future Work

- Driver Implementation (wl1271 + tone generator)
- Energy Measurement for new Audio-WiFi PS mode.
- Comparing energy measurement with the 802.11 PS mode

Thank you

- Questions?