U.S. Air Force Report [Triangulation solution update]

Date: 4/29

Location: University of Southern California

Team Lead: GUK IL KIM

Team: Sparsha Srinath, Harshul Ravindran, Yash Kulkarni

Summary of the Challenge and Purpose: The 96th Cyberspace Test Group faces a challenge with spurious signals from unauthorized devices, like remote controls, cell phones, GPS, radios, and drones, disrupting their testing processes. These unwanted signals interfere with the sensitivity of test environments, particularly in open field settings. The group is seeking to adopt cost-effective software-defined receivers that can be expanded and operated remotely, providing a strategic solution to precisely detect and manage these signals. The goal is to improve the precision in identifying disruptors, thereby enhancing the test environment's integrity. The focus on low-cost solutions reflects a commitment to managing budget efficiently while addressing the necessity for precise signal control and minimizing external interferences, ensuring reliable and accurate test outcomes.

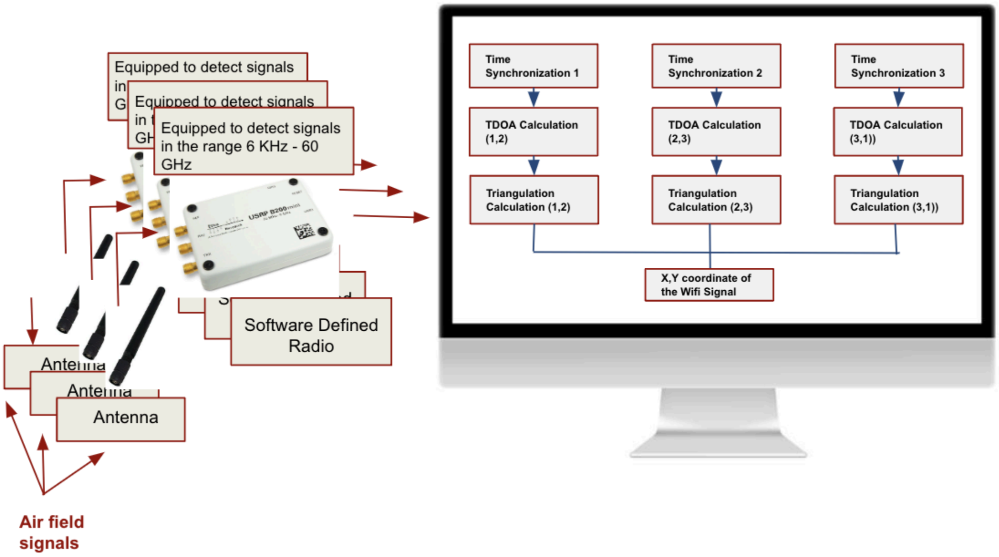
Market product:

Name / Co.	pros	cons	Price
HackRF One	- wide frequency range from 1MHZ to 6GHZ - Half-duplex transceiver; can both receive and transmit. - Open-source hardware and software, enabling extensive customization and flexibility. - Compatible with a variety of SDR software and platforms.	- Limited by a 20 MHz bandwidth, which may not be sufficient for all applications. - Half-duplex rather than full-duplex limits simultaneous send/receive operations. - Relatively high noise floor compared to more expensive devices.	$300 USD
RTL-SDR Blog V3	- Extremely cost-effective and widely available. - Receives frequencies from 500 kHz to 1.75 GHz. - Can be used in direct sampling mode to listen to HF frequencies. - Compact and easy to set up with widespread software support.	- Receive-only; lacks transmission capability. - Limited dynamic range and selectivity compared to higher-end models. - Not inherently designed for professional triangulation without additional equipment or modifications.	$30 USD

LimeSDR Mini	- Frequencies range from 10 MHz to 3.5 GHz. - Full-duplex capability, allowing simultaneous transmission and reception. - 12-bit sample depth and up to 30.72 MHz of bandwidth, offering better performance. - Open-source and compatible with many SDR applications.	- More complex to set up and use compared to simpler, cheaper alternatives. - Requires good understanding of SDR concepts to fully exploit its capabilities. - Power consumption and heat generation can be issues during prolonged use.	$160 USD
USRP B200 Mini	- Covers frequencies from 70 MHz to 6 GHz. - Full-duplex operation with a 56 MHz bandwidth for higher performance applications. - Supported by the comprehensive UHD (USRP Hardware Driver) and GNU Radio. - High-quality build and reliability suitable for academic and professional use.	- Significantly more expensive than entry-level SDRs. - Requires a more complex setup and deeper understanding of SDR for effective use. - Not as portable as some smaller, cheaper SDR models.	$1,100 USD
PlutoSDR (ADALM-PLUTO)	- Frequencies range from 325 MHz to 3.8 GHz. - Open-source software and hardware for versatility. - Compact and relatively easy to program and modify for specific needs. - Offers full-duplex capabilities with up to 20 MHz of bandwidth.	- The default frequency range requires manual unlocking to reach its full potential. - Some limitations in real-world applications due to design and hardware constraints. - Less mature ecosystem compared to other devices like the HackRF or USRP series.	$150 USD
Signal Hound BB60C	- Wide frequency range from 9 kHz to 6 GHz, allowing for comprehensive spectrum monitoring. - Real-time bandwidth up to 27 MHz and sweep speeds of 24 GHz/sec, ideal for fast, accurate spectrum analysis.	- Higher price point compared to basic SDRs. - Requires a PC for operation as it does not function standalone. - Primarily focused on spectrum analysis, so may require additional software or hardware for complex	$2,879 USD

	- Portable and USB-powered, which makes it versatile for field use. - High dynamic range and sensitivity, suitable for weak signal detection and detailed analysis. - Includes Spike software which is robust for spectrum analysis, recording, and playback.	triangulation setups.	
Keysight N9340B Handheld Spectrum Analyzer	- Frequencies range from 100 kHz to 3 GHz, extendable to 6 GHz, which covers most RF analysis needs. - Handheld, rugged design making it suitable for field operations in harsh environments. - Features a pre-amplifier and a sophisticated tracking generator for detailed analysis and testing. - High-resolution, color LCD screen and intuitive user interface for easy operation. - Offers measurements like channel power, adjacent channel power, and carrier-to-noise ratio which are crucial for professional RF analysis.	- Very expensive compared to typical SDRs, making it more suited for enterprise or professional applications. - Limited to frequency analysis; additional software or external devices might be required for complex triangulation tasks. - Heavier and bulkier than USB-powered SDRs, thus less convenient for lightweight mobile setups.	$10,000 USD

Solution:



Flow of our solution:

1. **Hardware Setup:**
   Place the three USRPs in different known locations within the area you want to monitor for WiFi signals.
   Ensure that each USRP is synchronized to a common time reference to enable accurate signal correlation

2. **Signal Acquisition:**
   Configure each USRP to scan for WiFi signals within its frequency range.
   Capture raw signal data from each USRP, including signal strength, frequency, and timing information.

3. **Signal Processing:**
   Pre-process the raw signal data to extract relevant information such as SSID, signal strength, and signal characteristics.
   Apply signal processing techniques such as filtering, synchronization, and noise reduction to improve the quality of the received signals.

4. **Time Synchronization:**
   Use the synchronized time reference to align the captured signal data from all three USRPs.
   Ensure that the timestamps associated with each captured signal are synchronized across all USRPs to facilitate accurate triangulation.

5. **Triangulation Algorithm**:
   Implement a triangulation algorithm to estimate the location of the WiFi signal emitter based on the signal strength and time of arrival information from the three USRPs.
   Common triangulation algorithms include:
   1. **Multilateration:** Using the known locations of the USRPs and the measured time differences of arrival (TDOA) of the WiFi signal at each USRP to calculate the emitter's location.
   2. **Trilateration:** Similar to multilateration but using signal strength (RSSI) instead of TDOA to estimate the emitter's location.

6. **Location Estimation:**
   Calculate the latitude and longitude of the WiFi signal emitter based on the triangulation results.
   Convert the estimated coordinates to geographic coordinates (latitude and longitude) using techniques such as geodetic calculations or mapping APIs.

7. **Output Formatting:**
   Organize the triangulation results into a list of tuples, where each tuple contains the SSID, latitude, and longitude of the detected WiFi signal emitter.
   Ensure that the output format meets your requirements for further analysis or visualization.

Code Snippet:

```
function localize_device(tdoa_measurements, anchor_locations):
    estimated_coordinates = []

    for each measurement in tdoa_measurements:
        ssid = measurement.ssid
        angles = []
        distances = []

        for each anchor_pair in measurement.anchors:
            anchor1 = anchor_pair.anchor1
            anchor2 = anchor_pair.anchor2

            angle = calculate_angle(anchor1, anchor2)
            angles.append(angle)

            tdoa = anchor_pair.tdoa
            distances.append(tdoa * speed_of_sound)

        if enough_measurements(distances):
            target_coordinates = triangulation(angles, distances)
            estimated_coordinates.append((ssid, target_coordinates))
        else:
            return "Insufficient measurements error"

    return estimated_coordinates
```

This code snippet defines a function localize_device that estimates the coordinates of a device emitting a WiFi signal based on Time Difference of Arrival (TDOA) measurements and the locations of anchor points (e.g., access points) in the environment. Here's a summary of what the function does:

1. **Input Parameters:**
   tdoa_measurements: A list of measurements containing TDOA values between the device and pairs of anchor points. Each measurement also includes the SSID (Service Set Identifier) of the WiFi signal.
   anchor_locations: A dictionary or data structure containing the locations of anchor points identified by their IDs or names.

2. **Output:**
   estimated_coordinates: A list of tuples, where each tuple contains the SSID of the WiFi signal and the estimated (latitude and longitude) of the device emitting the signal.

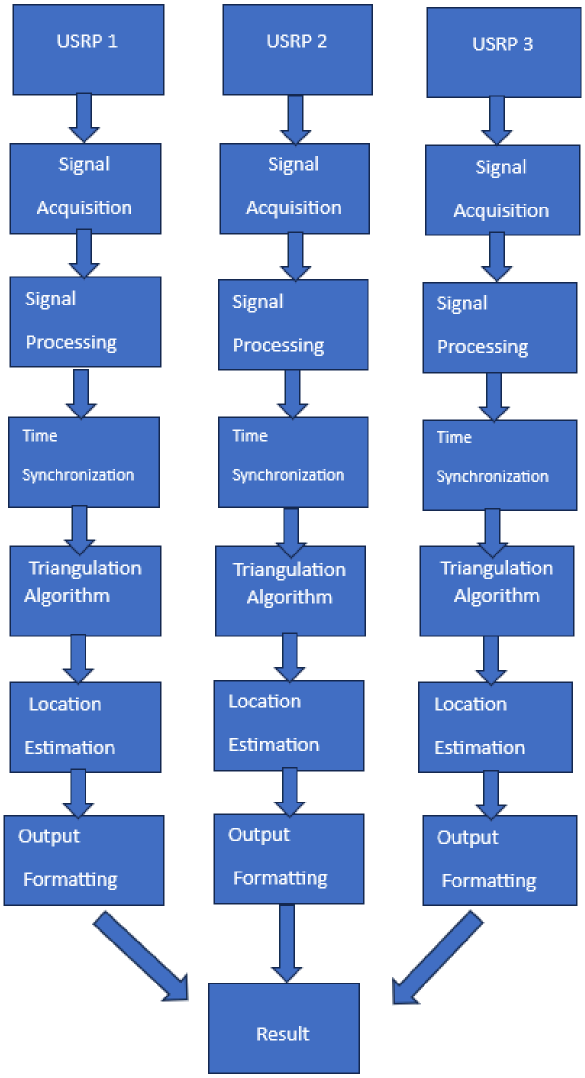
3. **Function Steps:**
   i) Initialization: Initialize an empty list estimated_coordinates to store the estimated coordinates of the device.

   ii) Iterate over Measurements: For each measurement in the input list tdoa_measurements, do the following: Extract the SSID of the WiFi signal and initialize empty lists angles and distances to store angle and distance information.

   iii) Iterate over pairs of anchor points in the measurement:
   Calculate the angle between the anchor points using the calculate_angle function.
   Calculate the distance between the anchor points based on the TDOA value and the speed of sound.
   Append the angle and distance to their respective lists.
   Check if there are enough measurements to perform triangulation using the enough_measurements function. If not, return an error message indicating insufficient measurements.
   If there are enough measurements, perform triangulation using the triangulation function to estimate the coordinates of the device based on the angles and distances.
   Append the SSID and estimated coordinates to the estimated_coordinates list.

   iv)Return Output: Return the list of estimated coordinates.

4. **Error Handling:**
   If there are insufficient measurements for triangulation, the function returns an error message indicating the issue.
   This function essentially implements a triangulation-based localization method using TDOA measurements and anchor point locations to estimate the coordinates of a device emitting a WiFi signal



DATA FLOW

1. Signal Acquisition: Three USRP devices are positioned in different locations to monitor WiFi signals. Each USRP scans for WiFi signals, capturing raw data including signal strength, frequency, and timing information.

2.Signal Processing: The raw signal data is pre-processed to extract relevant information such as SSID and signal characteristics. Signal processing techniques like filtering and noise reduction are applied to enhance the quality of received signals.

3. Time Synchronization: The captured signal data from all USRPs is synchronized using a common time reference. This ensures that the timestamps associated with each captured signal are aligned across all USRPs for accurate correlation.

4.Triangulation Algorithm: A triangulation algorithm,TDOA, is implemented to estimate the location of the WiFi signal emitter. This algorithm utilizes the synchronized signal data from multiple USRPs to calculate the emitter's location based on signal strength and time of arrival information.

5. **Location Estimation**: The estimated coordinates (latitude and longitude) of the WiFi signal emitter are calculated based on the triangulation results. Geodetic calculations or mapping APIs may be used to convert the estimated coordinates into geographic coordinates.

6. Output Formatting: The triangulation results are organized into a suitable format, typically a list of tuples containing the SSID, latitude, and longitude of the detected WiFi signal emitter.

7.Result: The final output contains the location estimates of the WiFi signal emitters, which can be used for further analysis or visualization.

Experts contact list & what was discussed:

Title / name	Email	Details
Ashton, 16th Electronic Warfare Engineer	ashtonr13@gmail.com	Identified signal ranges (dense)
Professor Keith, Professor of Electrical and Computer Engineering	chugg@usc.edu	WiFi ISM Band, USRP
Tamoghna Sarkar, PhD candidate for electrical engineering	tsarkar@usc.edu	Advertisement, TDOA
Master Sgt Thomson	norman.thomson@us.af.mil	Budget, Features pricing
Professor Andreas Molisch, Professor of Electrical and Computer Engineering	molisch@usc.edu	Arrival time accuracy
Rahul Modi	modirahul08@gmail.com	TDOA and FDOA
Andy, Element Tech Lead, 46 Test Squadron	andreas.keipert@us.af.mil	Success rate for testing
Nicholas L. Carballo, Technical Research Engineer at 96th Cyberspace Test group	nicholas.carballo.1@us.af.mil	Optimal Solution Identification