2015

Implementation of GNSS/GPS navigation and its attacks in UAVSim testbed

Farha Jahan

University of Toledo

Follow this and additional works at: http://utdr.utoledo.edu/theses-dissertations

Recommended Citation
http://utdr.utoledo.edu/theses-dissertations/1865

This Thesis is brought to you for free and open access by The University of Toledo Digital Repository. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of The University of Toledo Digital Repository. For more information, please see the repository’s About page.
A Thesis

entitled

Implementation of GNSS/GPS Navigation and its Attacks in UAVSim Testbed

by

Farha Jahan

Submitted to the Graduate Faculty as partial fulfillment of the requirements for the
Masters of Science Degree in Engineering

Dr. Weiqing Sun, Committee Chair

Dr. Mansoor Alam, Committee Co-Chair

Dr. Hong Wang, Committee Member

Dr. Patricia R. Komuniecki, Dean
College of Graduate Studies

The University of Toledo
May 2015
Unmanned systems or remotely piloted vehicles can easily accomplish tasks where human lives would be at risk. These systems are being deployed in areas which would be time-consuming, expensive and inconclusive if done by human intervention. Air, ground and underwater vehicles are three major classes of unmanned systems based on their operational environment. Clearly, in terms of causing damage, unmanned aerial vehicles (UAVs) are most efficient and have been known to change the course of several recent wars. If security of these systems is compromised, it will pose a serious threat to human lives as well as the nation. Therefore, it is important to analyze various possible attacks that can be attempted on these systems. Federal Aviation Administration (FAA) has limited the use of UAVs to 400 feet or below in the US National Airspace (NAS), primarily, due to the threat to the general population. This makes real world testing difficult in an academic setup. The best solution to this problem is to have a simulation based environment where different operational scenarios, related cyber-attacks, and their impacts on UAVs can be easily studied. Software based simulators are very economical to test different features of a UAV in terms of various defense mechanisms against cyber-attacks. In this thesis, we enhance UAVSim, a simulation test-bed for UAV Network cyber-security analysis, to include the Global Navigation Satellite System (GNSS), or more specifically, the Global Positioning System (GPS). The testbed allows users to perform security
experiments by adjusting different parameters of the satellites and UAVs. It also al-
allows implementation of different attacks in attack hosts. In addition, each UAV host
works on well-defined mobility framework, radio propagation models, etc., resembling
real-world operational scenarios.
[To my parents for their hard work and perseverance to raise me to this level and higher in all odds.]
Acknowledgments

A few words of acknowledgment is not enough to express my gratitude towards every person who has directly or indirectly influenced and motivated me towards accomplishing my Master’s degree. I would like to thank my adviser Dr. Weiqing Sun without whose leadership and guidance this accomplishment would have taken much longer. I would also like to thank the EECS Department chairman and my co-adviser, Dr. Mansoor Alam, and my thesis committee member Dr. Hong Wang, for taking time out of their busy schedule to serve on my thesis evaluation committee and give valuable comments and suggestions to improve my work.

My family has always been my pillar of support and encouragement. Undoubtedly, I am grateful to them. I would also like to thank Mr. Ahmad Yazdan Javaid. He has been my friend, philosopher and guide. My journey would be 'all work and no play', without my beloved friends Sami, Niyaz, Salma, Rubia and others. They have spiced up my life with healthy discussions, fun, trips, celebrations and compliments.

I would like to thank COGS for providing me graduate assistantship. It was a pleasure working with Dr. Patricia Komuniecki on occasions and all the COGS staff, especially my supervisor Teresa Lepiarz-Hayes.

My gratitude would be incomplete without thanking Cheryl, Eric, Christy, John and Michelle who welcomed me as their family. Last but not the least, I would like to thank my aunt Nikhat and cousins Dr. Zafar Hassan, Nusrat Hassan and Ishrat Hassan for being there for me in a land away from home.
Contents

Abstract iii

Acknowledgments vi

Contents vii

List of Tables x

List of Figures xi

List of Abbreviations xiv

1 Introduction 1
   1.1 Unmanned Aerial Vehicles ................................. 1
   1.2 Role and Importance of Navigation .......................... 2
   1.3 Recent Attacks and Failures ............................... 4
   1.4 Goals and Objectives ................................... 5
   1.5 Organization of Thesis .................................. 6

2 Related Work 8
   2.1 Literature Survey ..................................... 8
   2.2 UAV Simulators without GPS Implementation ............... 9
   2.3 UAV Simulators with GPS Implementation .................. 10
   2.4 Independent GPS Simulation ............................... 10
   2.5 GPS Security Related Works ............................... 11
5 Simulation Analysis and Results

5.1 Introduction ......................................................... 44

5.2 Results: GPS Implementation Based ............................... 46
  5.2.1 Average Localization versus Simulation Time/Number of Host 46
  5.2.2 Effect of Seed Value variation ................................. 47
  5.2.3 Average Localization versus Satellite Lock .................... 48
  5.2.4 Implementation of Circular Mobility Model ................... 51
  5.2.5 Average Localization versus Speed ........................... 52
  5.2.6 Variation in Angle of Linear Motion ........................ 53
  5.2.7 Sleep Duration versus Localization .......................... 55

5.3 Results: GPS Spoofing Attack Based ............................. 58
  5.3.1 Effect of GPS Spoofing on Linear Path ....................... 58
  5.3.2 Effects of GPS Spoofing on Circular Path .................... 61
  5.3.3 Analysis ......................................................... 66

6 Conclusion and Future Work ......................................... 68

6.1 Conclusion .......................................................... 68

6.2 Future Work ......................................................... 69
List of Tables

5.1 Default Satellite Parameters ........................................ 44
5.2 Default Host Parameters ............................................. 45
5.3 Default Attack Host Parameters ................................... 45
# List of Figures

1-1 UAV market to grow by 12% CAGR [14] ............................ 2
1-2 Role of navigation in UAV operations ............................. 3
1-3 Figure demonstrates the 47 Class A UAV crashes between 2001 and 2013 and expansion in drone operations to 110 bases [20]. .................... 5

3-1 Global Positioning System/Global Navigation Satellite System .......... 14
3-2 Trilateration .............................................................. 15
3-3 Navigation Message Content and Format Overview [58] .................... 19
3-4 Some of the applications of UAVs ................................... 20

4-1 Architectural Design of UAVSim ..................................... 31
4-2 Simulation World Map ................................................... 33
4-3 Satellite Ground Track [73] ............................................ 35
4-4 GNSS/GPS Implementation ........................................... 36
4-5 GPS implementation process flow diagram ........................... 38
4-6 Class Diagram .......................................................... 40
4-7 GPS spoofing attack implementation process flow diagram .......... 42

5-1 Variation in localization with Simulation Time ....................... 46
5-2 Variation in localization with number of host ....................... 47
5-3 UAV host on the lower left corner of the map ....................... 48
5-4 UAV host on the lower right corner of the map .................... 48
5-5 Localization curve without lock implemented on satellites during localization 49
5-6 Localization curve with lock implemented on satellites during localization
5-7 Circle mobility ......................................................... 50
5-8 Circular mobility complete trajectory .............................. 51
5-9 Distance Traveled ..................................................... 52
5-10 Speed Vs Average Error ............................................ 53
5-11 Path of Host at different Angles .................................. 54
5-12 Angle vs Average Error ............................................. 54
5-13 Localization Error versus Simulation Time ....................... 55
5-14 Sleep Duration Vs Localization Error .......................... 56
5-15 Sleep Duration Vs Average Localization Error ............... 57
5-16 Effect of discrepancy introduction in X-values of the spoofed GPS packet on a Linear path of the UAV .................. 58
5-17 Effect of discrepancy introduction in Y-values of the spoofed GPS packet on a Linear path of the UAV .................. 59
5-18 Effect of discrepancy introduction in both X and Y-values of the spoofed GPS packet on a Linear path of the UAV .......... 60
5-19 Effect of discrepancy introduction in distance as well as X and Y-values of the spoofed GPS packet on Linear path of UAV 61
5-20 Effect of low discrepancy introduction in X-values of the spoofed GPS packet on Circular Path ............................. 62
5-21 Effect of higher discrepancy introduction in X-values of the spoofed GPS packet on Circular Path ............................. 63
5-22 Effect of +ve discrepancy introduction in both X and Y-values of the spoofed GPS packet on Circular Path .................... 64
5-23 Effect of -ve discrepancy introduction in both X and Y-values of the spoofed GPS packet on Circular Path .................... 65
Effect of +ve discrepancy introduction in X-values and -ve discrepancy in Y-values of the spoofed GPS packet on Circular Path
## List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADS-B</td>
<td>Automatic Dependent Surveillance - Broadcast</td>
</tr>
<tr>
<td>C3UV</td>
<td>Center for Collaborative Control of Unmanned Vehicles</td>
</tr>
<tr>
<td>DDoS</td>
<td>Distributed Denial of Service</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>GNSS</td>
<td>Global Navigation Satellite System</td>
</tr>
<tr>
<td>GCS</td>
<td>Ground Control Station</td>
</tr>
<tr>
<td>GSCS</td>
<td>Galileo Satellite Communication Simulator</td>
</tr>
<tr>
<td>GSSF</td>
<td>Galileo System Simulation Facility</td>
</tr>
<tr>
<td>LOS</td>
<td>Line of Sight</td>
</tr>
<tr>
<td>MSSE</td>
<td>Multi-Scale Satellite Simulation environment</td>
</tr>
<tr>
<td>NORAD</td>
<td>North American Aerospace Defense Command</td>
</tr>
<tr>
<td>Open-SESSAME</td>
<td>Open-Source, Extensible Spacecraft Simulation and Modelling Environment Framework</td>
</tr>
<tr>
<td>OS3</td>
<td>Open Source Satellite Simulator</td>
</tr>
<tr>
<td>PNT</td>
<td>Position Navigation and Time</td>
</tr>
<tr>
<td>SNACS</td>
<td>Satellite Navigation Radio Channel Signal Simulator</td>
</tr>
<tr>
<td>SPEEDES</td>
<td>Synchronous Parallel Environment for Emulation and Discrete Event Simulation</td>
</tr>
<tr>
<td>TLE</td>
<td>Two-Line Element</td>
</tr>
<tr>
<td>UAV</td>
<td>Unmanned Aerial Vehicle</td>
</tr>
<tr>
<td>RAIM</td>
<td>Receiver Autonomous Integrity Monitoring</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Unmanned Aerial Vehicles

There has been a remarkable growth in the use of UAVs in various military and civilian application domains. Unmanned systems is expected to be a $89 billion dollar market over next 5 years in US military alone and its compound annual growth rate is expected to be 12% (Figure 1-1). From intelligence, surveillance, reconnaissance, etc., to areas like agricultural imaging, traffic monitoring, cartography, package delivery, etc. UAVs have played an important role where human reach is difficult or limited and/or lives are at risk. For example, these systems can be sent to distant or inaccessible planetary bodies for research (e.g., Philae aircraft landed on a comet after a 10 year journey [1]) or to detect and survey real-time catastrophes like earthquake [2] and forest fires [3]. UAVs have found their use in applications like proposed pizza delivery (Pizza Hut) [4], proposed local package delivery as well as hiring drone pilots for tests (Amazon) [5, 6], agricultural chemical deployment [7], ecological surveys [8], natural event monitoring (Hurricane Hunter [9]), disaster management [10] and humanitarian response (e.g., damage assessment, search and rescue operations, dropping relief supplies in case of emergency [11]), 3-D Mapping and photogrammetry [12], wildlife protection [13], etc. Primarily due to their positive impact on human effectiveness and
safety, UAVs have found their increasing importance in the military domain. In early 2015, commercial UAVs were authorized by FAA and more than 7500 are expected to be seen in the air by 2020 [9] compared to this number being negligible in 2014. These have also promoted research in industries and academia. FAA doesn’t require people to obtain any license for drones that are used for recreational purpose, but they do limit its usage up to a height of 500 ft and away from airport and air traffic [15]. Availability of low-cost mini-UAVs and DIY drones have promoted individual use and research as well. Constructing a drone is nowadays possible in as less as $300 by purchasing separate parts from online stores [16]. Non-requirement of license for such cases and easy construction of a drone poses a huge threat to the general population because of its possible malicious usage, especially, if their navigational equipment is not secure.

1.2 Role and Importance of Navigation

To understand the importance of navigation in any task, it is necessary to understand the role it plays. Navigation serves as our eyes and ears in any unknown terrain.
Be it an aircraft, a ship, a car or any other vehicle, knowing the task area in advance clearly gives the vehicle operator necessary directions to move around and ensures safety of all involved in the given unknown environment. In the military domain, aircraft, gunship and UAV operators already suffer with stress and performance pressure due to the responsibility of human lives. Accurate and reliable navigation data reduces mission related anxiety, impact on their mental health and optimal operational performance of the team [17].

Figure 1-2: Role of navigation in UAV operations

Figure 1-2 depicts the role of navigation in any operation carried out in an unknown terrain. Other roles navigation plays in defense operations include, but are not limited to:

- Prior terrain mapping and safety zone information
- Mission reconnaissance and surveillance
- Real-time navigation in unknown indoor/outdoor environment
• Enemy location awareness and safe path planning

• Reducing civilian casualties through prior planning

1.3 Recent Attacks and Failures

With advancement in technology, the application domains of drones are no more limited to labs and defense. They can be used by hobbyists, pranksters and trouble-makers as well. This increases threat to the general public and chances of adverse usage of an increasingly cheaper technology. After Iran’s claim of RQ-170 capture, an in-depth study of UAV vulnerabilities was made and it was understood how easily a UAV can be compromised and attacked. In 2012, North Korea launched a GPS Jamming attack on the border of South Korea, which disrupted navigation of aircraft, ships and ground vehicles [18]. Several other works discuss recent attacks on UAVs and about 49 large drone failures since 2001 [19, 20]. Figure 1-3 shows the location of these drone crash incidents of severe category (called class A) and the Pentagon’s plan to extend the operation base to 110 in 39 US states.

Another work evaluates the risk and vulnerability of UAV for cyber-attacks based on the components that makeup the UAV architecture including its environment, communication links, data storage, fault handling mechanisms etc. A report published in 2001 lists the causes of failure for UAVs as ”insufficient testing before purchase” and suggests that it is common to all failed programs [21]. A lot of money invested in these programs also went in vain. If we have a proper testbed to test these systems before flight, midair collision and ground casualty can be prevented and hence, loss of human lives and large investments in failed experiments can be minimized.
Figure 1-3: Figure demonstrates the 47 Class A UAV crashes between 2001 and 2013 and expansion in drone operations to 110 bases [20].

1.4 Goals and Objectives

Although UAV development started in early 1960s, primary objective of the research has been its mission-accomplishment capability, reliability, and efficiency in terms of time and power. Not much attention has been given to the cyber-security aspect of such systems until recently. The most important issues in this area are vulnerability, breach and threat identification; and corresponding attack prevention, mitigation and recovery. Out of all the works that have been done until now in the area of UAVs, most of them focus on the causes and methods of security breaches at the lower-level system components, surprisingly, very few works focus on the application or communication security of these unmanned systems. The need of a simulation testbed which can simulate single or multi-UAV behavior and provide a realistic response in case of an attack, served as our initial motivation. It is clear that navigation is one of the vital aspects of unmanned systems, therefore, its availability is signif-
icant. The auxiliary goal of providing the academia with a cost-effective mode of simulations, was also accomplished simultaneously.

This thesis focuses on enhancing the UAVSim simulation testbed, developed at Advanced Computing Research Laboratory (ACRL) of the University of Toledo, to include navigation module so that next level of navigation related attacks can be implemented and simulated. The main component of a navigation module is its GPS unit which helps it to know its actual position. It also helps the UAV during different preset navigation modes, such as, position hold, return-to-home, autonomous flight and collision avoidance [22]. Depending upon the area in which UAV is deployed, a GPS can be used to link data to its spatial position. This method is termed as geo-referencing [23]. It is due to their navigation capability that UAVs have found their application in various fields and operations. The GPS signals are typically very weak, less than 100W and are transmitted over a range of $20 - 25,000$ kilometers. This makes them fall below the noise floor spectrum when they reach the earth’s surface [24]. These signals are vulnerable to failure, disruption and unintentional or deliberate interference. Clearly, navigation is one of the most important modules of a remotely controlled UAV. The increased dependency of UAVs on GPS signals for localization, navigation and time-synchronization has made it a focus area for adversaries and thus led to the discovery of its vulnerabilities to attacks like Spoofing and Jamming. Simulations related to UAV operations involving GPS and related navigational aspects are quite important for correct simulations.

1.5 Organization of Thesis

Several works in the past have focused on single UAV dynamics simulation for model development and testing various enhancements. We discuss more about related works in the area of UAV testbed development and studies related to GPS
attacks on UAVs in Section 2.1 and previous work done by our ACRL team in Section 2.6. Section 3 introduces GPS and presents our implementation of GNSS/GPS and related attacks. Section 4 presents the design, architecture, operation and attacks performed on the UAVs by our enhanced testbed, UAVSim. Section 5 presents simulation analysis and results, and the thesis is finally concluded in Section 6.
Chapter 2

Related Work

2.1 Literature Survey

This section discusses the works that have been done in the area of UAV and GPS simulation. Out of these, many focus on the modeling of a single UAV in a closed lab or in an open, but controlled, environment to improve system performance, flight range and usability. These have been accomplished without necessarily implementing cyber-attacks, their impact and related risks. A lack of close to real simulation of a UAV Network (UAVNet) including UAVs, Satellites, Ground Control Stations (GCS) and adversaries, was noticed. An ideal testbed should allow inter and intra-component communication, and component level behavioral analysis. Although many of these simulations implement GPS device or a GPS software simulator to generate Position, Navigation and Time (PNT) data signals, none of them have designed a working close-to-real GPS or GNSS system. We present a classification of these existing simulators based on GPS implementation. We also discuss some UAV-independent GPS/GNSS implementations available.
2.2 UAV Simulators without GPS Implementation

The paper [25] mentions GPS spoofing, but its implementation was out of scope while [26] uses the carrier phase of GPS signals through GPS receivers to obtain altitude and positional measurement. A visual training simulator based on mission equipment is discussed in [27] that trains an operator as well as test subsystems. GUI based simulation testbed used UAVs equipped with Piccolo II auto-pilots along with other hardware to demonstrate a wide range of information-oriented applications [28]. Another testbed was developed for wireless networks on small UAVs to analyze its monitoring architecture and parameters such as delay, throughput, range, etc [29]. All of these simulators and testbeds either works in a way in which GPS is not required or employs an external GPS device or software.

Another work, focused on simulation of a swarm of UAVs, LaBRI involves deployment of actual UAVs on a field for specific applications and check their survivability [29]. Software based simulators are quite economical in testing different features of a UAV in terms of various defense mechanisms against cyber-attacks. More software based network simulation systems were developed for a swarm of UAVs [30], [31], but these involved use of laptops or other hardware as UAVs. Two more important simulation testbeds for such swarms of UAVs were also developed, SPEEDES (Synchronous Parallel Environment for Emulation and Discrete Event Simulation) [32, 33] and C3UV (Center for Collaborative Control of Unmanned Vehicles) [28]. SPEEDES simulates a swarm of UAVs on a high performance parallel computer in order to match the actual speed and communication rate of the UAV network and C3UV testbed focuses on the fact that information acquisition through collaborative sensing and control are highly coupled. Two related works in the area of network security simulation are also quite different due to the unavailability of wireless security analysis through simulation incorporating node mobility. One of them, ARENA, was pro-
posed in 2007 and includes multi-level attack simulation in the network, but does not focus on all layers as well as individual modules of vital network components [34], such as UAV in our case. The other one, Ordered Scenario based Network Security Simulator, was proposed in 2005 and has the same limitation of simulating only wired components [35].

2.3 UAV Simulators with GPS Implementation

A visual 3D flight simulation software based on Matlab and Simulink was an early attempt towards simulating UAV which used navigation module of FlightGear [36]. FlightGear provides a generic GPS support with GPS receivers yet to be implemented [37]. Another project called UAV Playground, developed in Java, used FlightGear to receive GPS data and implemented GPS tracking using Google Earth [38]. In industry, aeronautical division of IDS Corporation has implemented an unmanned aerial vehicle simulator Hero UAVSim [39] composed of ground control stations (GCS) [40], UAV Simulator and a sensor payload simulator [41]. The UAV Simulator has GPS based auto tracking capability and GPS outage mitigation measures, while GCS has an integrated GPS receiver to determine its actual position.

2.4 Independent GPS Simulation

Several works have been done in the industry to simulate GPS and GNSS systems. LabSat simulator [42] is a low cost simulator, which provides the option of selecting from different GNSS such as GPS, GLONASS, Beidou, and Galileo. It generates genuine navigation signal that can be stored, replayed and used in different applications. Spirent implements GPS and GNSS through hardware (e.g. GSS9000 multi-frequency, multi-GNSS RF constellation simulator) as well as software (e.g.
SimGEN, SimAUTO, SimINERTIAL, etc.) to simulate navigation signals for professional, controllable and repeatable testing in the lab [43]. National Instruments (NI) has a GPS Simulator, which can produce the GPS signals of up to 12 satellites C/A codes for 24 hours to test GPS receivers [44]. Many other simulators such as IFEN Inc. NavX-NCS Professional/Essential, CAST Navigation SGX GPS Satellite Simulator, AeroFlex Portable GPS/Galileo/SBAS Positional Simulator GPSG-1000, etc., also simulate GPS and GNSS, but differ with respect to the range of signals produced and constellation implemented. All these devices are quite expensive and could not be considered for a UAV simulation testbed developed in an academic setup.

2.5 GPS Security Related Works

In 2012, Todd Humphreys’ and his research team successfully demonstrated that UAVs are vulnerable to GPS Spoofing and led it to believe that it was rising up while deceiving it to fly downwards [45]. They have presented their work in [46] explaining how a civil GPS receiver can be easily spoofed. Several works have been published by the author in context of spoofing in GPS receivers such as [47], [48]. Similar works focused on assessing the effect of time accuracy and time-based spoofing to demonstrate the vulnerability of time synchronization protocols to spoofing attacks [49] while the requirements of a successful GPS spoofing attack is discussed in [50].

2.6 Previous Work

Several works have been done during the study, design and development phase of our software based simulation environment UAVSim. In the first phase of development, single UAV and overall UAVNet models were defined in order to establish a close to real representation of the system and its components. This representation, in turn,
facilitated the creation of a software model. Other contribution of this work included:
analytical threat analysis, risk analysis, and attack impact evaluation using Flight-
Gear simulation software [19]. During the second phase of development, UAVSim de-
sign was implemented using OMNET++ and INET, and few cyber-attacks (Jamming
and Distributed Denial of Service(DDoS)) were performed. One of the major contri-
bution of this phase was an interactive GUI for users [51]. Continuing the work of the
next phase of enhancement and further validation, advanced features like multi-user
support, server based centralized simulation, etc. were implemented and the testbed
performed reasonably well with limited resources in a generic computing infrastruc-
ture for DDoS attack [52]. An extended performance analysis was then carried out
using two resource intensive attacks, DDoS and Jamming, as well as UAV swarm
simulations, to showcase the capability of the testbed to simulate all other attacks
that will consume less resources on the underlying computing infrastructure [53].

The work presented in this thesis is an extension of the development phase of the
simulation testbed. The previous work [51] describes UAVSim in detail. The third
phase proved the capability of UAVSim, that it can be used for swarm simulation
along with its primary purpose of security simulations of UAVNet.
Chapter 3

GPS/SATNAV Overview

3.1 GPS Basics

A satellite navigation or SATNAV is a system of satellites that provides autonomous geo-spatial positioning with global coverage and is termed as Global Navigation Satellite System (GNSS). Satellites send radio signals along a line of sight (LOS) to electronic receiver that calculates its position. These receivers are called Global Positioning System or GPS receivers [54].

Currently, the US GPS and the Russian GLONASS are the only GNSS that supports global coverage. Other prospective GNSSs are Compass of China and Galileo of Europe. Any GNSS has three operational segments - the space segment, the ground/control segment and the user segment. The figure 3-1 shows the three segments.

The GPS space segment consists of 32 satellites as of 7 October 2014, in six orbital planes [55]. The satellites orbits at an inclination of 55 degrees to the equator and complete their revolution around the earth in 12 hours, i.e., two revolutions around the earth in one day. A GPS satellite sends two ranging codes: C/A, used for public usage, and Precision code, used by the military. It has five frequency bands $L_1 - L_5$ where $L_1$ and $L_2$ are used and $L_5$ is reserved for Safety-of-Life. The signal consists
of GPS data and time, ephemeris data and the almanac. $L_3$ band is used for nuclear weapon detonation, detection and treaty or ban enforcement. $L_4$ is not used yet.

The ground/control segment consists of ground stations that track the GPS satellites, monitor their transmissions, perform analysis, and send commands and data to the constellation [55]. The ground network consists of one master control station, one alternate master station, 12 command and control antennas, and 16 monitoring sites.

The end-user antenna and GPS receiver that make the user segment that receives GPS signals and calculates the position, velocity, and time based on the ephemeris and almanac data in these signals. The receiver has to acquire and maintain lock on four unique satellites to get its position accurate in the 3D space.
3.2 Trilateration

Trilateration is the process of determining the location or position of an object using the geometrical concepts of circles, spheres and triangles [56]. It is used in navigation, surveying and in GPS. The sensor nodes in mobile sensor networks are usually used in various applications to collect sensor data and location information for location based services. To be more clear, suppose a man M standing on the world map has lost his location information. If a friend calls him and say that he is x distance away from place A and another says that he is y distance from place B, M can make two circles with A and B as centers and x and y as the radii of the circles respectively. He can be at either of the two intersection points of the circles. If a third friend says that he is at a distance z from location C, then the intersection of the three circles gives his approximate location. This concept is called
2D Trilateration and is shown in Figure 3-2. Substituting friends with satellites and considering circles to be spheres, we can calculate our location in 3D space using what is called a 3D Trilateration. Many 2D and 3D trilateration algorithms have been proposed to determine location accurately. The mathematical expressions to calculate a position in 2D and 3D are discussed below as proposed in [57].

3.2.1 2D Trilateration

An existing localization algorithm has been described here, which uses the fundamental concept of trilateration technique in 2D. The position of an object can be determined by solving the following equations:

\[(x-x_1)^2 + (y-y_1)^2 = (d_1)^2\]
\[(x-x_2)^2 + (y-y_2)^2 = (d_2)^2\]
\[(x-x_3)^2 + (y-y_3)^2 = (d_3)^2\]

The equations are converted to linear equations by subtraction and substitution.

\[2.(x_2 - x_1).x + 2.(y_2 - y_1).y = \alpha\]
\[2.(x_3 - x_1).x + 2.(y_3 - y_1).y = \beta\]

where

\[\alpha = (d_1^2 - d_2^2) - (x_1^2 - x_2^2) - (y_1^2 - y_2^2)\]
\[\beta = (d_1^2 - d_3^2) - (x_1^2 - x_3^2) - (y_1^2 - y_3^2)\]

In 2D space, the position (x, y) is obtained by solving the following matrix oper-
3.2.2 3D Trilateration

Similar to 2D trilateration, 3D trilateration uses the following equations.

\[
x = f(d_1, d_2, d_3) = \begin{vmatrix}
\alpha & 2Y_1^2 \\
\beta & 2Y_1^3 \\
2X_1^2 & 2Y_1^2 \\
2X_1^3 & 2Y_1^3
\end{vmatrix}
\]

\[
y = g(d_1, d_2, d_3) = \begin{vmatrix}
2X_1^2 & \alpha \\
2X_1^3 & \beta \\
2X_1^2 & 2Y_1^2 \\
2X_1^3 & 2Y_1^3
\end{vmatrix}
\]

where \(X_i^j\) and \(Y_i^j\) are referred to \((x_i - x_j)\) and \((y_i - y_j)\) respectively.

By solving the following matrix based equations, we get the value of \((x, y, z)\) in 3D space.
\[ x = f(d_1, d_2, d_3, d_4) = \begin{vmatrix} \alpha & 2Y^2_i & 2Z^2_i \\ \beta & 2Y^3_i & 2Z^3_i \\ \gamma & 2Y^4_i & 2Z^4_i \end{vmatrix} \]

\[ y = g(d_1, d_2, d_3, d_4) = \begin{vmatrix} 2X^2_i & \alpha & 2Z^2_i \\ 2X^3_i & \beta & 2Z^3_i \\ 2X^4_i & \gamma & 2Z^4_i \end{vmatrix} \]

where \( X^i_j \), \( Y^i_j \) and \( Z^i_j \) are referred to \((x_i - x_j), (y_i - y_j) \) and \((z_i - z_j) \) respectively.

### 3.3 GPS Receiver

As discussed above, GPS receiver is the user segment of the GPS system. The signal sent by the satellite contains ephimeris and almanac data along with time and the satellite code.
The figure 3-3 shows the sub-frames of the message. Ephemeris contains information about the health of all GPS satellites and their exact position in the orbit. Almanac contains the coarse orbit and status information of each satellite, an ionosphere model and information to relate GPS derived time to Coordinated Universal Time (UTC). It takes around 25 frames and 12.5 minutes to transmit the complete almanac. Ephemeris data take 18s to download with a speed of 50bps. Therefore, when a GPS receiver is powered on for the first time, it downloads this data and obtains a lock on the 3(2D) or 4(3D) satellites available in its LOS. It takes around 45s to start. This is called the Cold Start. Once the lock is obtained, this data is updated every second.

The distance of a satellite from a receiver is calculated using the time taken by the satellite signal to reach the receiver and speed of light. Finally, it calculates its position using ephemeris and almanac data, and apply them to the aforementioned trilateration equations. It also takes into consideration the modeling errors, troposphere and ionosphere errors, and clock errors from the satellite.
3.4 Applications of GNSS/GPS

As stated earlier, GPS was originally invented for military usage and later was made available to the public. With the advancement in technology and devices that incorporates GPS, civilian usage has also increased nowadays. GPS devices like Garmin, and cellphones, give very accurate location and time information with their 12 parallel channel receiver. Its application is not limited to transport services or location information. Listed below are some of the areas where the GNSS/GPS signals are used:
• Aviation industry uses GNSS signals for en-route navigation, automatic dependent surveillance - broadcast (ADS-B) in case of absence of radar signals, mid-air refueling, photogrammetry, etc.

• The road transport applications include in-car navigation, autonomous car driving, route guidance, traffic and lane control, trip travel information such as construction and traffic details, alternate routes, speed limits, etc.

• GPS ambulance and other vehicles can be used for emergency relief to gauge the level of disaster and find stranded motorists and victims. It can be used in emergency calls, rescue operations, crime prevention, tracking and stolen vehicle recovery.

• Marine transport services find GPS signals useful during ocean, coastal and in-land waterway navigation, automatic docking, cargo handling, dredging, automatic collision avoidance, vessel traffic services, etc.

• The rail transport industry uses GPS for signaling and train control, high speed warning, power supply control, door control supervision, level-crossing protection, etc. It also uses GPS for management services like fleet management, cargo monitoring, etc. and as well as for passenger information system such as pre-trip and on-trip information.

• Outdoor sports like geo-caching, cycling and running employs GPS for measurement of distance, tracks and direction info.

• GPS finds its usage in scientific applications like surveying, meteorology and climate research, environmental and construction monitoring, global reference systems and geo-dynamics, etc.

• It can be used as a personnel protection and tracking aid for blind and handicapped civilians or people suffering from Alzheimer’s disease, etc.
• Agricultural farms and fisheries uses GPS for land area mapping, yield moni-
toring, planting, spraying, etc.

• Time sensitive applications like digital broadcasting, power generation and distri-
bution, frequency/time calibration services and maintenance of time stan-
dards, etc., uses time data of the GPS signals.

3.5 GPS Vulnerabilities

Since GPS is low-cost and easily available worldwide, its application has found many uses as discussed in the above section. This doesn’t not make it invulnera-
ble to attacks or failures. Rather it can more easily be subjected to intentional and unintentional threats. GNSS itself is vulnerable to natural causes or technical lim-
itations. These vulnerabilities can generate incorrect signals and misleading PNT information which could be hazardous. In this section we discuss the vulnerabilities of GNSS/GPS, factors that impact GPS vulnerabilities and consequences.

3.5.1 Vulnerabilities of GNSS/GPS

Vulnerabilities of GNSS/GPS can be classified as: unintentional interference, inten-
tentional interference and human factors.

3.5.1.1 Unintentional Interference

Radio Frequency Interference- Various radio frequency signals from undesired sources are considered as interference. These RF waves are emitted by high power transmissions, television, mobile communication, mobile satellite services, and elec-
tronic devices. Other navigational systems or sensors such as RADAR, Tactical Air Navigation (TACAN), and pseudolites can cause unintentional interference due to
higher signal strength. Distance Measuring Equipment (DME) and Automatic Dependent Surveillance (ADS) are other such systems that send navigational information and can interrupt the GPS signals.

*Intense Solar Radiations*- As the sun approaches its eleven year cycle, it emits high intensity solar flares and solar storms known as coronal mass ejections (CMEs). These flares and storms are bursts of magnetic energy equal to millions of 100-megaton hydrogen bombs exploding at the same time. The radio waves produced across the entire electromagnetic spectrum reaches the earth and causes disruptions in the various signals including the GPS and other navigational systems. These events can also cause disruptions in the satellites that can temporarily shut them down leading to widespread disruptions on the ground. One such event occurred in December 6, 2006 which affected the GPS system, causing large number of receivers to stop tracking the GPS signals [59]. Similar events occurred in 2012 [60] and 2014 [61], but not much damage to the GPS system was reported.

*Space Weather*- It can be defined as conditions on the sun and in the solar wind, magnetosphere, ionosphere and thermosphere that can influence the performance and reliability of space-borne and ground-based technological systems and can endanger human life or health [62]. These activities cause ionic disturbances. During geomagnetic storms total electron count (TEC) in the ionosphere can increase more than 100%. This changes the refraction indexes for ionosphere and troposphere causing signal information delay and hence result in position error. Ionospheric scintillation in GPS signals arises from rapid spatial and temporal variations (less than about 15 seconds) in the ionosphere. This causes loss of lock on the satellites or ‘cycle slippage’ and can even affect dual-frequency GPS receivers. Solar energetic particles (SEP)-a type of cosmic rays, X-rays, radio bursts, extreme ultraviolet (EUV) radiations can cause spacecraft damage, satellite orbit decay and disorientation, geolocation errors, space track and launch trajectory errors, etc.; affecting all the segments of a GNSS.
Nuclear explosions in the upper atmosphere can cause similar effects to solar storms, potentially affecting the operation of GNSS for weeks due to propagation anomalies [24].

**Multipath Error** - Multipath interference occurs when a GPS receiver picks up reflected GPS signals from buildings or high towers along with the normal signals from satellites to calculate its position. This can give grossly erroneous results. With advanced antenna-filtering techniques, receiver filtering and processing techniques, most new GPS receivers are very effective at mitigating multipath errors. If unchecked, multipath can still cause an error of ten to a hundred meters.

**Atomic Clock Drift** Accurate timing has become a crucial parameter for companies which uses GPS to time-stamp their business activities. Since business transactions or power utilities requires precise timing, GPS clocks need to be highly synchronized and traceable to national and international standards. On occasions, these clocks behave unpredictably which can produce a high discrepancy in time and location data way before it is detected and corrected. Clock anomalies in GPS satellite atomic clock can be caused due to natural aging or bad navigation data upload. Various studies has been done to detect such anomalies and techniques to mitigate the error.

**Bad Signal** - Unusual signal envelope transmission due to fault in signal modulation or generation causes unpredictable behavior in receivers. Most recent incident was in March 2009, when SVN-49 had multipath effect on GPS signals due to an on-board experimental L5 signal generator [37].

### 3.5.2 Deliberate Interference

**Jamming** - Jamming devices basically transmit radio frequency as noise which interfere with lawful communications such as cell phone calls, messaging, Wi-fi and GPS system and hence leads to denial of service. GPS Jamming is the act of interfering with the ability of receivers to lock onto the GPS signal, eliminating the ability of
the user to determine 3D position or calculate other information such as time, speed, bearing, track, trip distance, and distance to destination [63]. These frequencies can be filtered to some degree by adaptive antennas and noise-filtering in well-designed receivers. The higher the power of jamming signal, the more wider the circle of damage it will reach. These jammers can have continuous wave form signal or pulsed signal. Several anti-jamming techniques have been proposed to prevent narrow band interference, wide band interference, etc, but still there is no absolute solution to this problem. Jammers are illegal to market, sell or use in US soil.

*Spoofing*- Seemingly, the most easy method to mislead a GPS receiver would be to transmit false measurements leading to a wrong position, velocity, and time calculations. Spoofing involves fabricating and transmitting seemingly genuine GNSS signal. It aims at spoofing GPS signals to give a false sense of accurate physical location and results in mission path diversion. It is nowadays comparatively easier to launch such an attack due to the availability of off-the-shelf GPS signal generators. Satellite constellation preservation and signal transmission precision are of utmost importance in such an attack so that spoofing is not detected. A mechanism called Receiver Autonomous Integrity Monitoring (RAIM) has been used in the design and development of anti-spoofing methods through fault detection and exclusion implementation.

*Meaconing*- Meaconing involves rebroadcasting the received GNSS signal with some delay in order to confuse enemy navigation. Meaconing has been noticed to happen unintentionally as well due to close by low impedance GNSS antennas. These attacks can be detected by analyzing the clock bias of the GPS receiver over time [64].

### 3.5.3 Human Factors

*Receiver Bugs*— Design requirements for a GNSS receiver does not require certification testing in all sectors. Certain specific circumstances can cause firmware bugs to be unearthed, such as handling of unhealthy satellites or tracking nonstandard codes.
One such experiment was done to exploit software bugs in underlying receivers using a hardware that cost only $2,500 and can cause a wide variety of GPS devices to malfunction within a 30 mile radius. Middle-of-the-earth attack was successfully conducted on Trimble NetRS that cost $19,000. It went into an endless reboot loop that persisted even after incorrect data was no longer supplied [65].

System Upgrade Bad Navigation Data- Navigation data for the next 24 hours is uploaded to GPS satellites every 24 hours by a Master Control Station in advance. Even a chunk of bad data can cause catastrophes. Three reported incidents without much harm happened in GLONASS in June 2002, March 2000 and March 1993 [66].

Leap seconds and roll-overs- Leap seconds are not handled correctly in all GPS receivers and such a roll-over in August 1999 resulted in the malfunction of few receivers.
Chapter 4

Enhanced UAVSim Design

4.1 Introduction

As discussed in section 2.6, OMNET++ is the base simulator for our testbed. INET2.2 has been used for various mobility, radio propagation models, and for wired and wireless communication. As part of our primary goal of security simulation of UAVNet, DDoS and Jamming attacks were simulated, as mentioned in section 2.6. GPS related attacks were chosen to be implemented next. These attacks would not be possible without a GPS signal receiver module in a UAV. To implement the navigation module as an enhancement to the existing UAVSim, the first requirement was to have a stable implementation of GNSS, which would provide GPS signals for position determination. Using any of the GPS Simulators, discussed in section 2.4, was not feasible due to the cost and scale of implementation. The other requirement was to find a software simulator that would integrate easily with OMNET++. A study of different navigation simulators was done and compared to find the best solution for our research.

Galileo System Simulation Facility (GSSF) [67] mainly reproduces and analyzes the functionality and performance of the Galileo navigation system. Implemented in C++, it provides simulation for longer time periods and large geographical area
coverage. Available as a free licensed software from its website, [www.gssf.info], it also provides raw Galileo and GPS signal generation, express mode simulation and good functionality to analyze and visualize data.

The Satellite Navigation Radio Channel Signal Simulator (SNACS) [54] is a single satellite source GNSS signal generator. It is open-sourced and implemented in C++ with parallel processing. Its radio channel input and simulation results can be analyzed in MATLAB.

Open-SESSAME (Open-SOURCE, EXTENSIBLE SPACECRAFT SIMULATION AND MODELING ENVIRONMENT FRAMEWORK) [68] is another simulator that provides dynamics simulation for spacecrafts to develop hardware as well as to test flight algorithms. Based on C++, it not only provides attitude and orbit modeling, but can also be applied to orbit simulation, space environment assessment or control algorithm validation.

OS3 (Open Source Satellite Simulator) [69, 70] is an OMNET++ based simulation platform for evaluating satellite communication protocol. It was developed as a framework for simulating various kinds of satellite-based communication. OS3 provides a generic satellite constellation that seamlessly integrates real satellite tracks and weather data to simulate different conditions along with good visualization. OS3 was released under public license and is now part of the INET framework. Its code is available for modification and enhancement. Implementation of a highly accurate and stable satellite movement and modeling in OS3 provided a good base for the development of our GPS system. Being platform independent, OS3 can be employed easily on any system. We discuss more about why this particular simulation platform was chosen.
4.2 CNI_OS3

The common and major limitation of above mentioned simulation software, other than CNI_OS3, was difficulty in integration with OMNET++. CNI_OS3 implemented simple satellite mobility (such as Global Positioning System) without any satellite communication implementation and satisfied all our other requirements. The foundation of this work was Galileo Satellite Communication Simulator (GSCS) [71] (also known as Multi-scale Satellite Simulation Environment (MSSE)). Although GSCS was based on the INET framework of OMNET++ simulation engine, it was Galileo satellite navigation system specific. CNI_OS implements a generic satellite constellation. CNI_OS3 uses a TLE (Two-Line Element) format file for fetching initial positions of various satellites being used for simulation. Depending upon the Navigation System TLE file used, specific navigation system can be simulated. For example, we can use TLE file of 31 GNSS satellites to simulate GNSS while we can use a TLE file of 30 Galileo satellites to simulate Galileo Navigation System. It provides an accurate satellite movement simulation with live weather data, high resolution altitude data, different visualization options, etc. The comparison of these existing satellite frameworks has been done in [69]. Therefore, it can be concluded that other than communication, all other features were available in CNI_OS3.

4.2.1 Features

Below are some of the additional features of CNI_OS3 that supported its selection for use in UAVSim [72].

- Detailed modeling of communication aspects
- Dynamic integration of already implemented protocol stacks
- Modular architecture and easy extensibility
• Usable for any arbitrary constellation

• GUI for easy and comfortable handling

• Configuration of scenario-specific parameters

• Two different visualization methods
  – Worldview
  – Local Evaluation/Azimuth representation

### 4.2.2 Limitations

Below are the limitations of CNI_OS3, some of which were required to be addressed before moving on to the implementation:

• Satellites were not capable to send signals and establish a communication link.

• GPS or any other navigation technique was not implemented. A crude localization was available which used object oriented programming where an observer could directly access the satellite coordinates using its object and then save them without calculating its own position using those coordinates.

• Only two types of satellite mobility models were implemented.

### 4.3 UAVSim Modules

To overcome the limitations of CNI_OS3, we added new modules to our existing work UAVSim, in addition to the six core modules to the UAVSim as depicted in figure 4-1. The new modules are shown in pink in the figure and defined in brief below. More about the core modules can be found in [51].
4.3.1 Satellite Model Library

Satellite model library has the standard satellite model which inherits its basic features from the satellite model defined in CNI_OS3. The GPS functionality has been added to the satellite model through the development of a broadcast based application which sends position information to the receivers through radio signals of L1 frequency range as per the standard GPS implementation. The GPS/SATNAV implementation has been described in detail in section 3.

4.3.2 Satellite Network Module

Similar to UAV network module, this module defines network stack of the satellites. It defines the communication protocol, transmission power, access points, etc. This module extends basic node package from INET for satellite communication while satellite mobility packages are derived from CNI_OS3.
4.3.3 Navigation Module

The navigation module has receiver-end GPS application which enable UAVs to receive satellite navigation signals carrying data for calculation. The data from four or more satellites are required to calculate the position of a UAV using Multi-lateration in the 3D space. In our implementation, we have approximated the implementation to a 2D localization, therefore, a minimum of three satellites are required.

4.3.4 Attack Library

The attack library contains various attacks that were implemented in the previous work. These include single target DDoS and multiple target Jamming attacks. With the implementation of GNSS module (considering, space segment and user segment), GPS related attacks were included in this library.

4.4 Implementation

This section elaborates the implementation details of our enhanced UAVSim. The communication between UAVs and satellites or any GPS receiver is unidirectional. Satellites broadcast signals without waiting for any acknowledgment. That is why, a connectionless broadcast protocol was used to implement GPS. The packets contain satellite index, X and Y coordinates of the satellite sending the packet, in map-pixels, and distance of the satellite from each host. In its initial start, when a UAV receives packets from three different satellites, it calculates its position using the 2D trilateration equations as defined in section 3.2.1 and creates a lock on them. This is called ‘Cold Start’. It starts accepting packets from these three locked satellites for position calculation. Point to note here is that that the packets should be from each locked satellite otherwise the packets are discarded. If UAV doesn’t receive packet from any of the locked satellites for 10s, the lock on that satellite is released and a new
satellite is searched for, to lock on to. Such a situation is termed as Locate-the-sky.

The implementation of GPS is very basic and an approximation of the real GPS. We have implemented the localization in 2D, considering the position of the satellites and UAVs on the map as per map dimensions. The scaling of the earth to the map has been done as follows. Considering, the radius of the earth is 6371 km, the circumference of the earth will be

\[ 2\pi r = 2 \times 3.14 \times 6371 \]

If we imagine to open the globe vertically, the circumference will give the length of the map while the width of the map will be half the circumference. The map dimensions are defined as

\[ 1080 \times 2160 \]
The figure 4-2 shows the world map used in the simulation environment. So horizontally, each pixel on the map corresponds to

\[
\frac{2 \times 3.14 \times 6371 \times 10^3}{2160} = 18.5km
\]

or vertically,

\[
\frac{3.14 \times 6371 \times 10^3}{1080} = 18.5km \text{ on earth}
\]

Therefore,

\[
1m \text{ on earth} = \frac{1}{(18.5 \times 10^3)}m \text{ on map}
\]

Different UAV models available today have different speeds depending upon its design and usage. The Arcangel-1 has a speed of 150 km/h while airforce mission UAVs like Patroller range of UAVs can fly at a speed of 241 km/h. Taking an approximate speed of 250 km/h of a UAV or 69.5 m/s, the speed of the satellite on the simulation map will be

\[
69.5/(18.5 \times 10^3) = 0.0037m \text{ per second}
\]

The user defined parameters can be defined in INI file such as the number of satellites and UAV hosts. The simulation world map shown in figure 4-2 shows 30 satellites in orbit. The satellites revolve around the earth in an orbit which is inclined at an angle of 55 degrees. The motion of the satellites in their orbit around the planet appears as a sine wave when tracked from ground as shown in figure 4-3.

For illustration, 20 hosts are shown on the map. The number of hosts can be varied for each simulation. Along with satellites and hosts, there is a CNI_OS3, channel control and mission control centers depicted on the map. The CNI_OS3 module defines the webservice related data, weather control data and calculation module which calculates the distance, attenuation and other channel characteristics.
There are access points which act as a central transmitter and receiver of wireless radio signals. The blue circles show the range of each access point. If a host is outside the range it won’t receive any packets. Such that the host is always in the line-of-sight of one or more satellites, range of access points needs to cover the whole area of the map. A minimum of four access points is sufficient to cover the whole area with the given range.

Figure 4-4 explains the implementation of the GPS system. The space segment constitute GPS satellites while UAV hosts are the user segment as it receives GPS signal. The base satellite model with NORAD (North American Aerospace Defense Command) module, mobility model, and notification board has been defined in CNI_O3S. Standard satellite model extends base satellite while adding communication protocol stack from INET module. This protocol allows communication between network layers of the satellite so that the packets can be broadcast.

The GPS Application has been designed over a connectionless broadcast protocol provided by INET. This application is required in the standard satellite model to
create and broadcast packets. It can be enabled/disabled in .INI file for each simulation. On UAV host, this application acts as a GPS receiver in the navigation module. It receive packets from satellites and process packet data for localization. Other than enable/disable parameter, .INI file also defines local port, destination port, message length, packet send interval, mode of traffic to be sent, and the destination addresses. Mode of traffic to be sent can be set to "ONCE", "PER_BURST" or "PER_SEND". If there are more than one destination address and mode of traffic parameter-'chooseDestAddrMode' is set to ‘ONCE’, then one of the address is randomly selected for the whole simulation run. Similarly, destination address is chosen for each burst or for each packet depending on the value of the chooseDestAddrMode parameter as PER_BURST or PER_SEND respectively.

Algorithm 1 define the processes of GPS Application. After initializing global variables, it fetches destination addresses. The function ”processSend()” calls gen-
erateBurst() function which creates packets and send them to destination hosts as defined by the mode of traffic parameter ”chooseDestAddrMode”. For our simulation, chooseDestAddrMode is set to PER_BURST. processPacket() is defined for GPS receivers i.e. UAVHosts. It is non-functional for satellites and attackHosts. When GPS packets are received by UAVs, they are processed and data sent by the satellites are extracted to evaluate its position. Finally, statistics are collected on outgoing and incoming packets.

Algorithm 1: GPS App

1: **procedure** GPSApp
2: \hspace{1em} initialize()
3: \hspace{1em} processStart() \leftarrow destination addresses
4: \hspace{1em} processSend() \leftarrow generateBurst of packets
5: \hspace{1em} generateBurst() \leftarrow createPacket()
6: \hspace{1em} processPacket()
7: \hspace{1em} finish()

Algorithm 2 shows the steps and parameters required to create a packet for a GPS signal. The packet has sourceId and msgId as two parameters. msgId gets incremented by 1 for each packet sent. Each packet has satellite data, its location and distance information.

Algorithm 2: Create Packet

\begin{verbatim}
procedure createPacket
2: \hspace{1em} sourceId \leftarrow getId
3: \hspace{1em} msgId \leftarrow numSent
4: \hspace{1em} satId \leftarrow getIndex
5: \hspace{1em} Xcoord \leftarrow PositionX
6: \hspace{1em} Ycoord \leftarrow PositionY
7: \hspace{1em} setDistance \leftarrow distance of host from satellite
\end{verbatim}

Algorithm 3 shows how the packet is processed. The packet is checked for sourceId and msgId. When the simulation starts, UAV would have no lock on any satellites.
It will get position data from packets received from three unique satellites. Satellite index helps it to identify the satellites. A counter variable 'countPacket' is incremented to one after the packet is received from distinct satellite index otherwise it is discarded. Once the count of packets becomes equal to three, position of the UAV is calculated using trilateration and a lock is set on those satellites while counter is set to zero. Next time when the packets are received, algorithm checks for those satellite indexes on which lock has been established. It keeps on discarding the packets from other satellites and wait until it receive packets from the locked satellites. In the real world, such a scenario happens when there is no satellite signal and GPS device wait to receive signals. Figure 4-5 shows the process flow of the GPS implementation.

The figure 4-6 describes the classes defined in the UAVSim. INET and CNI,OS3 are predefined packages. UAVHost has been defined as previous work. GPSApp
Algorithm 3: Process Packet

1: procedure PROCESS PACKET
2: if packet has sourceId and msgId then
3:     satelliteId ← satId
4:     if (countPacket ≤ 3) then
5:         if (lock == 0) then
6:             countSat[countPacket] ← satelliteId
7:             positionData[countPacket] ← position details of three unique satellites
8:             countPacket ← countPacket + 1
9:         else
10:             positionData[countPacket] ← position details of the three locked satellites
11:             countPacket ← countPacket + 1
12:     if countPacket==3 then
13:         calculatePosition()
14:         countPacket ← 0
15:         lock ← 1
16: end if

inherits Linear/Circle mobility from INET and SATSGP4Mobility from CNI OS3 modules. Satellite Model inherits basic satellite feature from CNI OS3 and communication protocols from INET through which it can communicate with other satellites. It inherits GPS Application to act as a GPS Satellites. GPSSimulation module defines the GNSS/GPS in the NED file with UAVs, attack hosts and mission control centers. The new modules are depicted in orange. UAV model has UAVHost and UAV attack host. UAV attack host has features equal to the UAV host except that it sends counterfeit packets similar to the packets sent by GPS Satellites. Also, packet processing feature of the attack host has been disabled as they won’t be processing any packets.

4.4.1 GPS Attacks Implementation

Todd Humphreys’ study has shown that UAVs are vulnerable to GPS Spoofing attack. We have demonstrated that this attack can be simulated in our testbed. It
Figure 4-6: Class Diagram
was implemented by an attack host that had equivalent or more features as the UAV host and it could broadcast counterfeit GPS signals. The GPSAttack App shown in figure 4-6 is enabled for attack hosts to send the spoofed signals. It has been described in detail as below.

4.4.1.1 GPS Spoofing

We implemented GPS Spoofing attack using a spoofed GPS Signal Generator, which is in fact another UAV at almost double the altitude of the UAV being attacked. Due to the public nature of GPS implementation details, building such a generator would be quite easy. We assume higher height in order to mislead directional antennas installed on a UAV for GPS signal reception. Also, the attack host maintains same angle and distance with the host at all times so that the host does not detect any suspicious activity. The spoofed signal can contain discrepancy in x co-ordinates, y co-ordinates or distance as represented in the algorithm below and the process flow of the attack implementation in figure 4-7:

Algorithm 4: Create Attack Packet

procedure CREATE_PACKET
2: sourceId ← getId
3: msgId ← numSent
4: satId ← getIndex
5: Xcoord ← PositionX+discrepancy
6: Ycoord ← PositionY
7: setDistance ← distance of host from satellite

4.5 Constraints and Assumptions

Various constraints and assumptions made during the development of GNSS/GPS in UAVSim include the following:
Although UAVSim implements both the space segment and user segment, the control segment is not implemented.

Only the primary functionality of GPS, position calculation of the UAV is implemented at the receiver end.

Use of ephemeris and almanac data for calculation of the position is implemented approximately. The satellites send its position information relative to the UAV in the packet as a parameter rather than calculating the distance using speed of light and time of transmission.

The project being non-funded, available medium-end systems were used rather than high performance parallel computing system.
• Most of the system information is not available in the public domain and thus gathering various networks and communication related information posed a major challenge.

• Instead of calculating distance between the satellite and the host through speed of light and time difference in transmission and reception, this distance is being sent in the packet itself. Reason being the limitation of OMNeT++ of Tx/Rx event timing being exactly the same (correct up to a nanosecond) to make it seem real-time.

• We have approximated the implementation to a 2-D localization (trilateration) instead of 3-D (multilateration). This is due to the 2-D nature of OMNeT++ simulations.

• Capability of the attackers have been assumed to be equal or more than the UAVs in simulations. This enables impact evaluation of more powerful adversaries as well as when our own systems are compromised.

• The communication environment of the simulation testbed takes into consideration disturbances caused by random noise, upper layers of atmosphere and communication signals present in the lower layers.
Chapter 5

Simulation Analysis and Results

5.1 Introduction

In this section, we demonstrate simulation results based on implementation of the navigation module in UAV and efficiency of an attack host to successfully launch a GPS Spoofing attack. The first part of the results are analyzed to demonstrate the correctness of the GPS implementation while the later part discusses how changes in the parameters of attack host signals can alter the path of UAV. The various scenarios and variation in parameters have been analyzed with respect to the localization and mobility type. The below tables shows the default values of the satellites, host and attack host during normal simulation. The default simulation time is taken as 600s unless stated otherwise. Attack simulations have been considered with one UAV and one attack host.

Table 5.1: Default Satellite Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mobility Type</td>
<td>SatSGP4Mobility</td>
</tr>
<tr>
<td>Transmitter Power</td>
<td>500W</td>
</tr>
<tr>
<td>Packet Interval</td>
<td>0.5s</td>
</tr>
<tr>
<td>Burst Duration</td>
<td>10s</td>
</tr>
<tr>
<td>Sleep Duration</td>
<td>0s</td>
</tr>
<tr>
<td>Position Update Interval</td>
<td>1s</td>
</tr>
</tbody>
</table>
Table 5.2: Default Host Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mobility Type</td>
<td>LinearMobility</td>
</tr>
<tr>
<td>Transmitter Power</td>
<td>10W</td>
</tr>
<tr>
<td>Speed</td>
<td>0.0037s</td>
</tr>
<tr>
<td>Burst Duration</td>
<td>10s</td>
</tr>
<tr>
<td>Sleep Duration</td>
<td>0s</td>
</tr>
<tr>
<td>Position Update Interv</td>
<td>1s</td>
</tr>
</tbody>
</table>

Table 5.3: Default Attack Host Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mobility Type</td>
<td>LinearMobility</td>
</tr>
<tr>
<td>Transmitter Power</td>
<td>10W</td>
</tr>
<tr>
<td>Speed</td>
<td>0.0037s</td>
</tr>
<tr>
<td>Burst Duration</td>
<td>10s</td>
</tr>
<tr>
<td>Sleep Duration</td>
<td>0s</td>
</tr>
<tr>
<td>Position Update Interv</td>
<td>1s</td>
</tr>
</tbody>
</table>

The background image, the world map, is the default one used in CNI_OS3. Other background images can easily be used by changing it in the configuration file. Using the testbed, we analyzed how a UAV would perform with implementation of the navigation module. The below analysis shows the accuracy of UAV localization implementation with different parameters and the average error. Over each simulation run, error is calculated by averaging difference between each actual and corresponding calculated value.
5.2 Results: GPS Implementation Based

5.2.1 Average Localization versus Simulation Time/Number of Host

Although the localization count is not the same in every run, with an increase in simulation time, average localization increases as shown in figure 5-1. It should be noted that average localization decreases with increase in the number of hosts as shown clearly in figure 5-2.
Figure 5-2: Variation in localization with number of host

5.2.2 Effect of Seed Value variation

Seed value is a number used by the PRNG (pseudo random number generator) algorithm of OMNeT++. This random number is used to randomize initial values of the position. Changing the seed value changes the position of the satellite on the map as shown in figures 5-3 and 5-4. When the seed value is 2352610, the host is on the lower left corner of the map and when it is changed to 7399210, host moves to the lower right corner. It is important to evaluate the effect of seed value because the initial position impacts the localization of the host as it affects the possibility of its presence in the communication range of an access point. In order to cover the whole map for correct localization, we have deployed four access points. These access
points resemble the Differential GPS (DGPS) stations on earth, which help in easier localization and communication.

5.2.3 Average Localization versus Satellite Lock

Obtaining a lock is important to get a very accurate position. Satellites broadcast signals all the time. Also, there can be more than 3 satellites (2D implementation) in line-of-sight but at different distances with the host. The calculated distance is prone to error. If every computation of the localization uses data from different sets
Figure 5-5: Localization curve without lock implemented on satellites during localization

Figure 5-6: Localization curve with lock implemented on satellites during localization
of satellites, it will lead to large variation in error between the calculated position and the original position. Figure 5-6 shows the variation in localization when no lock on the satellites is implemented. The variation between the calculated localization curve and the actual curve was so large that they could not be plotted together.

When the lock on the satellites is implemented the variation in error reduces to a minimum of 0.001. The calculated position almost overlaps the original position. Figure 5-5 shows the localization curve to be in sync with the original path through the lock on the satellites.
5.2.4 Implementation of Circular Mobility Model

The simulation runs as expected in different mobility models. For Linear Mobility, UAV travels a linear path as shown in above figures. Most of our simulation analysis is performed with Linear Mobility. The figure 5-7 shows the circular path as traversed by the UAV when mobility model is changed to Circle Mobility. One need to define the center of the path and the radius. To get the complete circular trajectory as in figure 5-8, the simulation was run for 1000 simulation seconds. For this particular case, the center was taken as (x,y)=(561m, 432m) in map pixels and radius as 1meter which corresponds to 18.5 km in the real world. These mobility model are already defined in INET module. Other mobility models can also be selected and simulated.
5.2.5 Average Localization versus Speed

As the speed of the UAV host increases, as expected, distance covered by the UAV increases. Figure 5-9 shows the distance traveled by the UAV with different speeds. The green curve shows the distance traveled when the speed is 250 km/h and the red curve shows the distance traveled when the speed is 500 km/h keeping the simulation time as 400 seconds. With an increase in speed, average localization error increases as shown in figure 5-10. The default value of speed is 0 mps i.e., the host is stationary. The speed is defined in the configuration file as a PRNG function of normal distribution truncated to non-negative values. The standard deviation was...
set as 0.005.

5.2.6 Variation in Angle of Linear Motion

The default value of angle of Linear Motion is a random value with a uniform distribution between 0 degrees to 360 degrees. The graph 5-11 shows the calculated path of the satellite at different angles. As we can see from the graph, that the host follows the correct angle path from its calculated localization data. The graph 5-12 shows the average error at different angles. Average error increases as we move from its linear axis and is again minimum at 90 degrees. The graph 5-13 shows the variation in localization error with simulation time for three different angles. The
Figure 5-11: Path of Host at different Angles

Figure 5-12: Angle vs Average Error
localization error is approx 1% only. The graph shows that the number of times error in localization occurs increases with an increase in the angle.

5.2.7 Sleep Duration versus Localization

GPS based navigation can offer relatively consistent accuracy if sufficient GPS signals can be tracked during the entire UAV mission [74]. Due to its low power range, intentional or unintentional interference can cause UAVs to lose the signals. The GPS signal outage can cause a significant deviation in the navigation solutions. To represent such a scenario where UAVs lose the signals because of interference from television signals, mobile signals, ultra wideband communications or when traveling around high buildings and trees, our testbed has a parameter called ‘sleep duration’,
which corresponds to such a GPS signal outage. The parameter can be assigned to both the satellites and the host. In case of satellites, during the specified sleep duration, they would be inactive and would not broadcast any packets while the hosts would reject packets even if the satellites are sending them. The default sleep duration has been kept at 0s for the satellites as well as the host, assuming that there is no outage and the UAV receives signals during its entire simulation time.

Figure 5-14 shows the effect of change of sleep duration on the localization error. The simulation has been mapped from 0s - 10s of sleep duration with an interval of 2s for 200 localization values. It clearly shows that the maximum error reaches about
Figure 5-15: Sleep Duration Vs Average Localization Error

1%. Figure 5-15 shows the overall localization error with respect to sleep duration that is less than 0.1%. Average localization error is almost negligible, which shows the accurateness of the localization algorithm.
5.3 Results: GPS Spoofing Attack Based

5.3.1 Effect of GPS Spoofing on Linear Path

Case I: Discrepancy in X-direction - In this case, we vary the x-value and keep the discrepancy increasing using the expression

\[ x = x + (0.005 \times s) \]

where \( s \) is initialized as 0 and incremented by 1 in each new packet generated. Figure 5-16 shows the results of this experiment. As seen in the figure, the original
southwest direction of UAV is quite different than the spoofed direction of west. This shows an increase in calculated Y-values while a decrease in calculated X-values.

*Case II: Discrepancy in Y-direction* - In this case, we vary the y-value and keep the discrepancy increasing using the expression

\[ y = y + (0.005 \times s) \]

where \( s \) is initialized as 0 and incremented by 1 in each new packet generated. Figure 5-17 shows the result of this experiment. As seen in the figure, the (almost) west direction of UAV is the actual path while spoofed GPS makes the UAV think that
it is going in the northwest direction. This shows a huge decrease in Y-values while very minimal impact on X-values comparatively. Clearly, this angle of variation will increase if we increase the discrepancy factor of 0.005.

**Case III: Discrepancy in X and Y-direction** - In this case, we vary the y-value and keep the discrepancy increasing using the similar expressions of Case I and II. Figure 5-18 shows the result of this experiment. As seen in the figure, the UAV thinks that it is moving in almost the reverse of its actual direction. This shows that both X and Y values are now increasing very rapidly.

**Case IV: Discrepancy in X, Y-direction and Distance** - In this case, a similar expression is used to introduce a discrepancy in all the three variables of x, y and
Figure 5-19: Effect of discrepancy introduction in distance as well as X and Y-values of the spoofed GPS packet on Linear path of UAV

the distance. Figure 5-19 shows the result of this experiment. Such discrepancy introduction shows that the spoofed path is similar to the one obtained when the discrepancy was introduced only in Y-values. This indicates that discrepancy in distance values somewhat negates the effect of discrepancy in X-values.

5.3.2 Effects of GPS Spoofing on Circular Path

In a second set of experiment, the GPS spoofing attack was carried out on a host moving in a circular path. Its initial position can be anywhere on the circular path with radius of 1m and center (561m, 432m) on the map. The starting position was selected randomly in order to introduce randomness of UAV position and see if
results were location independent. The attack host also moves in a circular path with its starting position on a circular path of radius 2m and center (565m, 435m). The attacks were designed considering different data broadcast from the attack host. Five cases were analysed for this particular scenario which are different from linear path scenarios. These are discussed below:

**Case I: Discrepancy in X-direction** - In this case, a discrepancy $s$ is added to X-values with a factor of 0.005 using the expression

$$x = x + (0.005 \times s)$$
where, \( s \) is initialized as 10 and incremented by 1 as each new packet is generated. Figure 5-20 shows the results obtained in this experiment. It is clear that there was a very minor deviation of the host from its original circular path and the host traverses almost the same original path.

*Case II: Higher discrepancy in X-direction* - Since increasing the discrepancy factor little by little was not resulting in tangible changes, we increased the discrepancy factor in X-values by 3 times to 0.015 while keeping Y and distance values the same for this case. Similar to case I, \( s \) was initialized as 10 and incremented by 1 as each new packet is generated. Figure 5-21 shows the results obtained in this case. As shown, the spoofed path is quite different from the original path and becomes linear.
Figure 5-22: Effect of +ve discrepancy introduction in both X and Y-values of the spoofed GPS packet on Circular Path starting from the original starting point in the opposite direction.

**Case III: Positive discrepancy in X and Y-directions** - In this case, we introduce positive discrepancy in both X and Y-values using similar expression as Case I. Result for this case is shown in Figure 5-22 and it shows that the host is actually moving outward in a helical path with varying pitch, while believing that it is moving in a circular path. It should be noted that the variation is mostly increasing Y-values and thus results in a helical path.

**Case IV: Negative discrepancy in X and Y-directions** - This case involves negative discrepancy introduction in both X and Y-values using similar expression as Case I. Related result are shown in Figure 5-23 which clearly shows that the host followed
the original path approximately and then moved outward following a modified helical paths. It should be noted that such a discrepancy is resulting in large negative variations in both X and Y-values.

**Case V: Positive discrepancy in X-direction and Negative in Y-direction**- In this last case, discrepancy was added to X-values while subtracting from Y-values. Figure 5-24 shows the results obtained in this case. It can be seen that the host follows the inward helical path moving away from its original position. It is clear from the graph that this kind of discrepancy is resulting in lower positive variation in Y-values while higher, or almost double positive variation in X-values, and this results in such an helical path.
5.3.3 Analysis

Through all these attack implementations, various results were obtained and valuable insights were gained. Some of them are listed here:

- Regarding GPS Jamming attack, the results were quite expected and variable, quite similar to real world jamming scenarios. As number of attack hosts were increased, average GPS packet loss increased and reached up to 90% which indicates quite successful jamming.
• When the discrepancy was introduced in only X-values, it was noticed that different motion paths have different variations, which implies that the variations could not be generalized.

• Discrepancy factor variation results in variation of spoofed path as well. In case of the circular path, this increase led to a spoofed linear path compared to a spoofed circular path when the factor was lower. Thus, low discrepancy factors would be hard to detect and can make a UAV lock on it as a real satellite and then increase the factor to cause path deviation.

• In general, it was noticed that variation in Y-values are resulting in worse effects. In case of the original linear path, the resultant deviations were huge while in case of the original circular path, Y-value discrepancies caused resultant helical path, which could confuse the UAV and correction made to correct its path may lead to crash.

• In general, the spoofed paths are similar to the original paths in terms of the class of curve, i.e., spoofed paths for original linear paths were linear while for circular paths, they were curved paths. This would result in tougher detection of discrepancy or path deviation if the discrepancy factor is quite low.
Chapter 6

Conclusion and Future Work

6.1 Conclusion

This work is an extension of previous work on UAVNet in the UAVSim testbed. UAV is incomplete without a navigation module. It works on global positioning system in which UAVs receive GPS signals containing positional information broadcast by satellites. This information help UAVs know its location which can be used to plot its trajectory. To fully implement GPS, we needed GPS satellite constellation and a GPS signal receiver in UAV. This has been successfully implemented in UAVSim. Simulations where performed with different parameter values in the simulation environment to test the robustness of the testbed. Simulations were also done with different UAV mobility and results have been analyzed and discussed. To implement GPS attacks, some UAV host were modified to be an attacker which would send counterfeit signals that would give false location information to UAVs. This attack would make target UAV go astray or confuse its path. The results of such an attack was used to analyze the behavior of attacked UAV in different scenarios. Hence, this simulation testbed can be used to model real UAV and analyze all the factors and scenarios before testing it on civil grounds and avoid mishaps, major or minor. This will not only minimize financial loss, but would also avoid any causality to happen.
6.2 Future Work

The implementation of GPS in this work is in 2D. Future works will be to mitigate the limitations of UAVSim and GPS/SATNAV implementation and to simulate it in a 3D environment. The attack simulations were done and analyzed with only one UAV. The next step would be to analyze how targeted UAV would behave if it was communicating with other UAVs. GPS Jamming is another such attack that could be performed on this testbed. Also, anti-spoofing such as RAIM (Receiver Autonomous Integrity Monitoring) and anti-Jamming techniques can be simulated to study how UAVs can be shielded from these attacks. Graphical User Interface and UAV model browser would be enhanced to include GPS attacks and plot its trajectory in the browser itself.
References


