Hyperbolic Location Estimation for Mobile Rogue Attackers

April 9th, 2008
Scott Broschell, sbrosche@connect.carleton.ca

1. Introduction

1.1 Context/Background

Over the past decade wireless technologies have become more prevent in day to day society and we have come to rely upon them more. With advances in wireless data streaming for uses that vary from cell phones, internet connections, vehicular networks and many other applications wireless environments are quickly replacing traditional wired networks and being used in a multitude of new ways. With this new wide spread adaptation of wireless technologies the traditional ideas of security are very often not adequate as the wireless medium is an inherently insecure one as the endpoints are not physically connected, which leaves them open to new types of both passive and active attacks.

There is also a major concern over privacy in these new wireless environments as GPS systems in cell phone and vehicles are extremely helpful to end users but could allow for an attacker to gain both the identity and location of a transmitter. Wireless Access in Vehicular Environments (WAVE) sets forth a set of standards that have explored this potential problem [1]. There have been a few measures introduced in order to prevent this security problem, one of them being Secure Anonymous Broadcasting (SAB) [2]. In this system only the certificate authority can obtain the logical identity of the vehicle which attempts to solve the problem of using digital signatures with unencrypted messages which traditional WAVE communications have. This system while doing a good job of solving this privacy concern also leads the system being much more attractive for hijacking by a rouge insider as all of the vehicles are now anonymous to each other. This presents a problem for taking action against a rouge insider as the normal course of action would be to revoke the attacker's certificate when they had been identified inside the network, but because of the anonymous property of all vehicles inside the network it is trivial for an attacker to assume a different identity. Therefore we must look to information that the attacker cannot change and that is inherent to the system. We can look to the Received Signal Strength (RSS) at each of the receivers. The RSS can be used to in order to create minimum and maximum hyperbola pairs between each of the receiver pairs in order to get a bounded area in which a transmitter can be expected to be with a certain probability. This process is referred to as Received Signal Strength Based Location and Estimation [3] or Hyperbolic Rouge Location Estimation [4].

1.2 Definition of the Problem

The Hyperbolic Rouge Estimation system can be simulated in a static environment such as MATLAB. Such scenarios are useful as a proof of the theory and mathematical concepts behind the system, but lack the flexibility for running multiple tests as any scenarios with different variables or conditions must be run independently which is very time consuming. A dynamic simulation that incorporates all of the ideas is preferable as a multitude of scenarios can be tested without having to reset the simulation variables. Such a simulation has been created [5] and achieves much of the desired functionality. The simulation in its original form provides a basis for a useable tool but lacks many traits that would make it more useful than a static MATLAB scenario. This project intends to extend the functionality of this simulation to make it a fully functional tool to not only model the Hyperbolic Rouge Estimation system but to also allow the user enhanced control to improve the value of the simulation overall.

1.3 Summary of the Results

Using the existing Ogre 3D [6] simulation tool as a basis, it has been expanded to add an array of new features. The simulation now has a fully functional graphical user interface that was created using the Crazy Eddies GUI system (CEGUI) for Ogre [7]. The new simulation GUI can be used to modify the simulation variables, parameters and move or add road side units (RSUs) adding a degree of flexibility needed to perform multiple scenarios without restarting the simulation. The simulation has also been enhanced to collect all of the relevant metrics in a log file. Finally the simulation has been enhanced with the addition of signal shadowing to more correctly model a real environment as well as the ability to highlight the intersection of all hyperbolic areas both of which can be very helpful to end users.

1.4 Overview of the Report

The detailed background information about the simulation and added feature are in Section 2. This is followed by a review of the results in Section 3 and a conclusion is provided in Section 4.

2. Background

2.1 Original Simulation Specification

The original simulation used one of the standard Ogre templates as the basis for the application which easily creates many of the basic entities needed. The attacker is represented as a single vehicle that can be moved onscreen by user input from the keyboard. The RSUs are represented as other immovable objects in the environment. Every second each RSU creates new minimum and maximum hyperbolas which bound the location of the vehicle. This creates an area in which the transmitter can be expected to be in with a degree of probability.

2.2 Ogre and Related Technologies

Ogre is a 3D graphics rendering engine written in C++ that allows users to use its SDK as a means for easily using and abstracting the 3D rendering libraries of Open GL or Direct 3D. The current release used in the simulation is 1.4.6 "Eihort" and was released in December 2007. CEGUI is a graphical user interface library that is built in C++ with modules for Direct 3D, Open GL and Ogre 3D. The CEGUI libraries are highly configurable as they do not load textures or render input themselves but accesses this information through code written by the user giving it a huge degree of flexibility.

3. Results

Starting from the original simulation a CEGUI has been created and merged with the current simulation. The enhancements for simulating signal shadowing and displaying the common hyperbolic intersection area have also been added into the main class file. An additional module has been created and linked to that allows for the logging of simulation metrics.

The CEGUI interface was built to go on top of current simulation and is broken down into five submenus that can be displayed and hidden by buttons that are located in the top of the screen. In order to facilitate communication between the existing simulation and interface a module was created to convert the standard OSI inputs for the mouse into their CEGUI counterparts. A new listener class was created to house all of the elements for the interface as this leaves all of the original simulation listeners untouched and running at their maximum performance as the GUI listener is only invoked when interface elements are being used. Figure 1 shows all of the new menus open in the simulation.

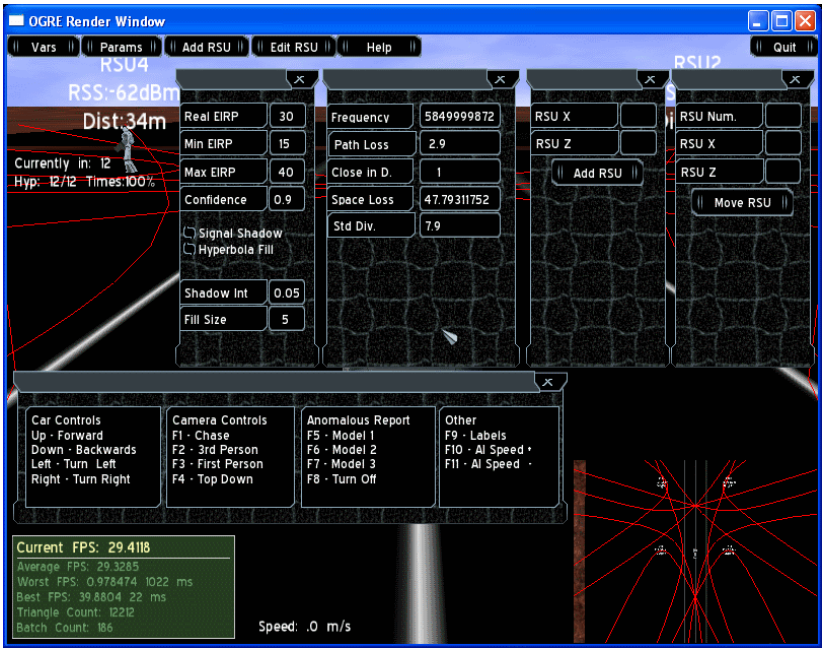

Figure 1: All of the GUI menus on top of the simulation.

The original simulation assumed that there were prefect conditions and there was no fluctuation of signal strength due to either environmental or equipment problems. To create the appearance of signal shadowing, the first step needed is to create a random number generator that creates numbers with a normal distribution as the signal shadowing variation can be modeled with a standard deviation and distribution based on the frequency and the confidence level [3]. Using a Box-Muller transformation, a uniformly distributed random number can be transformed into one that is now normally distributed for use in the signal shadowing algorithm. Figure 2 shows a sequence of outputs from the implemented algorithm along side a normal distribution curve and well not identical, the Box-Muller curve is very close to the normal distribution curve giving confidence of its accuracy. Also it can be shown that 95% of all generated numbers fall within two standard deviations of the mean as would be expected in a normal distribution. The graph uses the normal distribution from a confidence level of 0.8 and a standard deviation equal to a sigma value defined by the current signal frequency multiplied by the normal distribution for the confidence level. This normal random number generator was used as the basis to create a smooth signal shadowing variable which simulates a gradual loss or gain of signal strength. Figure 3 shows a set of these values.

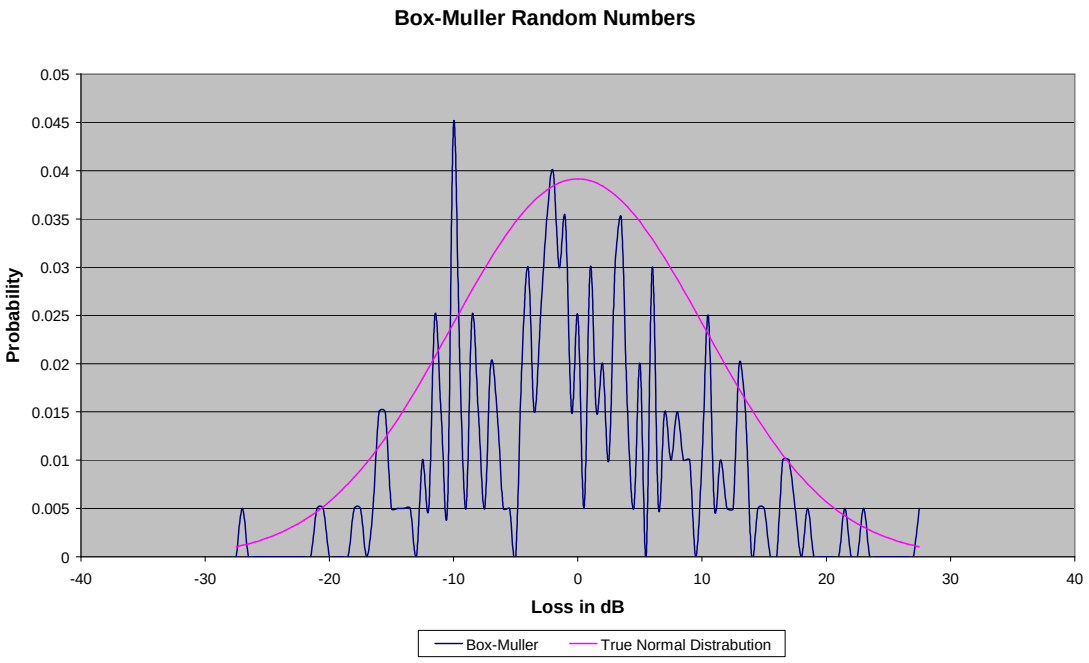

Figure 2: Analysis of the Box-Muller transformation

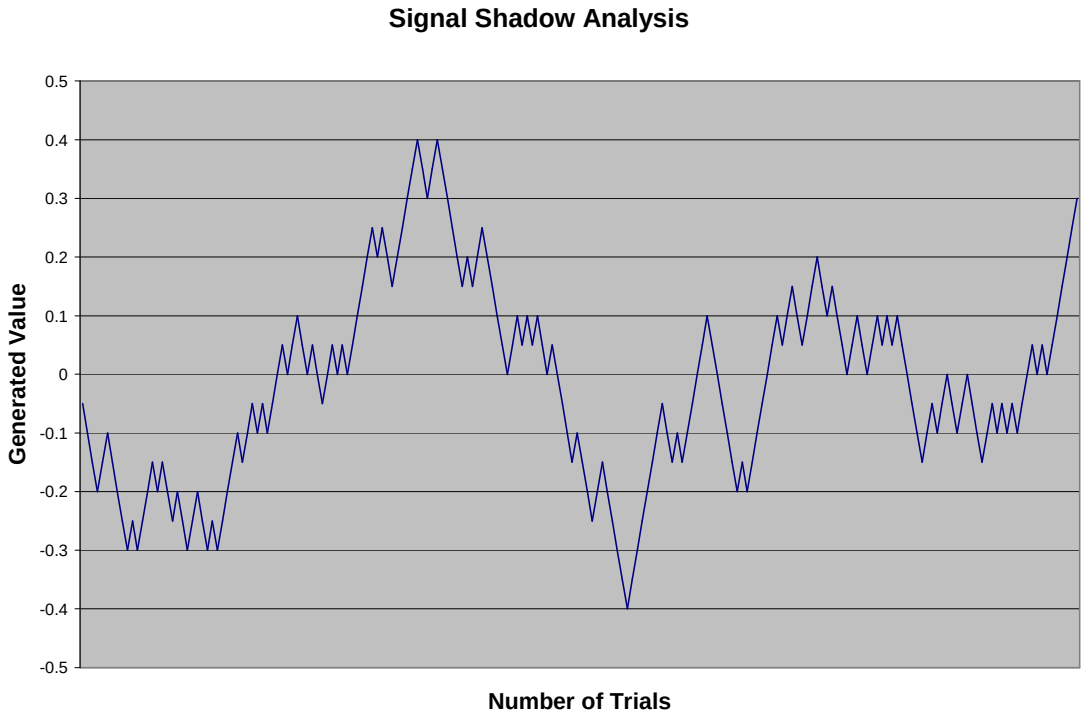

Figure 3: Analysis of the Signal Shadow output.

The last enhancement added to the simulation is the calculation and visual highlighting of the common intersection area of all of the hyperbolas. This system creates a grid of squares on top of the terrain and then uses them in order to display an approximation of the intersection area. The algorithm uses the formulas for calculating the minimum and maximum distances between receives and transmitters outlined in Lemma 3 of the paper by Laurendeau and Barbeau [3]. Using these formulas the set of midpoints of the squares is checked against each of the hyperbola minimum and maximum in order to find a bounded area which approximates the intersection of all of the hyperbolas. The size of the squares can be modified in order to gain a more accurate bounded area at a cost of extra computational time and storage.

In the simulation the set of tested squares is based on the current position of the car as to not use unnecessary system resources with calculation for squares which are not visible. In a real world setting this information would not be know. In a real world scenario this technique could be used with a set of static squares in order to quickly identify the hyperbolic intersection area which is the most probable area for the vehicle to be in. This could allow for streamlined checking of the area in which the vehicle is supposed to be at a given time or expanded in order to calculate a set of bounded probability regions where the vehicle could be.

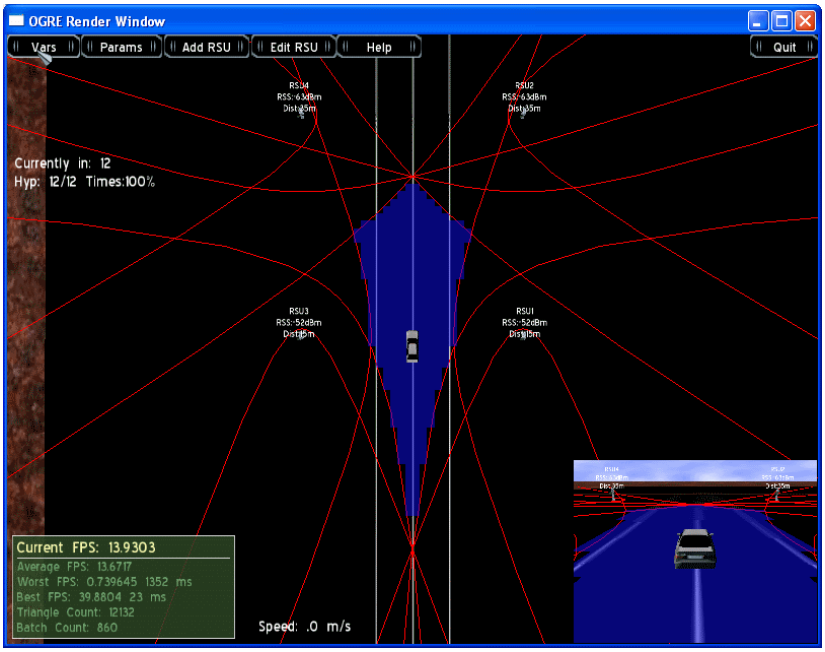

Figure 4: Simulation with hyperbola intersection area highlighted

4. Conclusion

From the original simulation, we have been able to create a program that can be used for a multitude of different situations. The strength of creating a dynamic 3D simulation is that the model can be evaluated in many different scenarios in order to understand where it works well and where it fails. The original simulation provided evidence that Hyperbolic Location is feasible and works well in vehicular environments. The simulation can more accurately portrait environmental and system changes that could very much affect the behaviour of the model. These changes can be made while the simulation is running. This means multiple situations can be tested in one run of the program. These enhancements provide stronger evidence that the Hyperbolic Location model works in situations that are closer to those in the real world and are not always ideal. This modifiable simulation provides information about the accuracy of the model with different parameters. The collection of metrics allows the visual analysis to be confirmed or possibly. Overall the enhancements add an ease of use. Extra features in the simulation create an environment conducive to testing new and unique situations that will be helpful to expand the knowledge of how Hyperbolic Location is useful in a variety of circumstances.

References

[1] C. Laurendeau and M. Barbeau. *Threats to Security in DSRC/WAVE.* Proceedings of the 5th International Conference on Ad Hoc Networks and Wireless (ADHOC-NOW). Lecture Notes in Computer Science, Springer Berlin / Heidelberg, Volume 4104, 2006.

[2] C. Laurendeau and M. Barbeau. *Secure Anonymous Broadcasting in Vehicular Networks*. First IEEE LCN Workshop on User Mobility and Vehicular Networks (ON-MOVE), Dublin, Ireland, 2007.

[3] C. Laurendeau and M. Barbeau. *Rogue Attribution Using Relative Signal Strength Based Location Estimation.* School of Computer Science Technical Report TR-08-01, January 2008

[4] C. Laurendeau and M. Barbeau. *Compounding Probabilistic Evidence for Hyperbolic Rouge Location Estimation.* School of Computer Science Technical Report TR-08-04, February 2008

[5] K. Nelson. *Simulation of Signal Strength Based Hyperbolic Localization to Achieve Rogue Attribution.*
http://www.scs.carleton.ca/~barbeau/Honours/Kevin_Nelson_b.pdf

[6] Ogre SDK and Documentation. http://www.ogre3d.org

[7] CEGUI SDK and Documentation. http://www.cegui.org.uk