EAI Endorsed Transactions
on Scalable Information Systems
Research Article ICST.ORG

Single and Multiple UAV Cyber-Attack Simulation and Performance Evaluation

Ahmad Y. Javaid[1,*], Weiqing Sun[2], Mansoor Alam[1]

[1]2801 W. Bancroft St., EECS Department, College of Engineering, The University of Toledo, Toledo, Ohio, USA
[2]2801 W. Bancroft St., ET Department, College of Engineering, The University of Toledo, Toledo, Ohio, USA

Abstract

Usage of ground, air and underwater unmanned vehicles (UGV, UAV and UUV) has increased exponentially in the recent past with industries producing thousands of these unmanned vehicles every year. With the ongoing discussion of integration of UAVs in the US National Airspace, the need of a cost-effective way to verify the security and resilience of a group of communicating UAVs under attack has become very important. The answer to this need is a simulation testbed which can be used to simulate the UAV Network (UAVNet). One of these attempts is - UAVSim (Unmanned Aerial Vehicle Simulation testbed) developed at the University of Toledo. It has the capability of simulating large UAV networks as well as small UAV networks with large number of attack nodes. In this paper, we analyse the performance of the simulation testbed for two attacks, targeting single and multiple UAVs. Traditional and generic computing resource available in a regular computer laboratory was used. Various evaluation results have been presented and analysed which suggest the suitability of UAVSim for UAVNet attack and swarm simulation applications.

Keywords: UAV Cyber-security, performance evaluation, simulation, testbed

Received on 28 August 2013, accepted 25 September 2014, published on 17 February 2015

doi: 10.4108/sis.2.4.e4

1. Introduction

With applications in almost every field, UAVs have become really popular for applications which were limited by human element. Until a few years ago, primary focus of development was military in nature but their use in other real world civil applications are on a rapid increase. With applications like pizza delivery (Pizza Hut), local package delivery (Amazon), agricultural chemical deployment, ecological surveys [1–4], industries and academia are using UAVs for their research, businesses, etc, and there are much more applications to be thought of. Without doubt, their importance in the military domain has increased several folds in the recent past due to their impact on human effectiveness and safety. Another important point to be noted is the delay in inclusion of civil and other kinds of UAVs in the National Airspace System (NAS) due to several issues including communication security [5].

Increased attack attempts in recent past on such mobile cyber-physical systems (CPS) are alarming and have raised concerns over their use, especially with increasing autonomy level [6, 7]. Keeping this in the mind, the authors noticed the need of cost-effective and safe virtual simulation testbed environment for testing the accurate implementation of various security related technologies in an Unmanned Aerial System (UAS). Addressing various environment variations, such as weather, loss of connectivity and contested communication are some of the most important aspects of such a simulation testbed due to dependency of UAV control on communication and its security. Therefore, we focus on two basic types of attack - one targeting a single UAV and second, targeting multiple UAVs in the mission area.

The rest of the paper is organized to provide background on related and our previous works in sections 2 and 3 respectively. Section 4 provides more details about UAVSim covering its design and various features. Section 5 describes all the performance analysis done and related results and inferences. Section 6 concludes the paper and discusses possible future enhancements to the work.

2. Related Work

In this section, we discuss some of the recent advances and works related to simulation or actual hardware based evaluations. As these UAV related issues are addressed by policy makers and bureaucrats, the need of a secure and safe UAV system stays unquestionable for military as well as civil applications due to safety and privacy threats imposed by their compromise.

*Corresponding author. yazdan@ieee.org

Therefore, several researchers have been working on development of different kinds of simulation testbeds in order to validate safe states of these systems and check possibility of moving into an unsafe state. These simulation testbeds can be classified in four major categories based on the resources they employ.

2.1. Software based Single UAV Simulation

Software simulation testbeds are purely based on well-known software platforms and do not employ any kind of hardware. Testbeds developed using Matlab/Simulink [8], FlightGear [9], JSBSim/FlightGear [7, 10] and Matlab/FlightGear [11] are some of the recent outcomes of research in this area. All these simulation testbeds have focused on testing a single-UAV model instead of modeling its behavior in presence of other UAVs in the real world.

2.2. Software-Hardware based Single UAV Simulation

Some other simulation testbeds using hardware along with software, have also been developed where the hardware might be actual UAVs [12], robots [13, 14], or just laptops [15, 16]. A very recent work of this type [10] focuses on analytical and component based simulation and analysis. In this work, the area of focus for cyber attacks is sensor compromise of various degrees.

2.3. Software based Multiple UAV Simulation

This class of simulation testbeds are also solely based on software platforms and these are developed in-house as well. One of the most important works in this class, SPEEDES (Synchronous Parallel Environment for Emulation and Discrete Event Simulation) [17], simulates a swarm of UAVs on a high performance parallel computer so that it can match the speed and communication rate of a real UAVNet. Another recent work of this class, DCAS (distributed cyber attack simulator) [18], presents a distributed simulation framework for modeling cyber attacks and the evaluation of security measures. DCAS is based on Portico, an open source HLA (high-level architecture) simulation engine. Limitation of this work is that it is for a generic wired or wireless network and does not include mobile components. On the other hand, UAVSim addresses these limitations and incorporates various mobility models, mobile radio propagation models, mobile ad-hoc routing protocols, etc.

2.4. Software-Hardware based Multiple UAV Simulation

This class of simulation testbeds are primarily based on software platforms but real or emulated UAVs can also be used within them. Rather than being commercial, these are mostly developed in-house for research purposes, specifically for UAVs. The only testbed that could be found in this category, *C3UV* [19], Center for Collaborative Control of Unmanned Vehicles at UC Berkeley, has been constantly updated by their researchers since 2004. Over the years, the *C3UV* team has incorporated multiple-UAV simulation on parallel computing environment along with the capability of using real UAVs. This kind of testbed, despite all its achievements, involves huge expenses in terms of high performance parallel computing hardware and optional use of real UAVs. While in UAVSim, cost involved is quite less and once positive results are achieved, the tested mechanism can be directly implemented in real UAVs only if required.

3. Our Previous Work

After studying all the important works done until now and their limitations, UAVSim was designed and developed keeping in mind the primary objective of UAVNet security simulation. Initially, UAV system model was defined to represent the system approximately so that a software model could be created. An analytical threat and vulnerability analysis was also performed and attack impacts were demonstrated using FlightGear simulation software [6]. Further, an independent simulation module (called UAVSim) was developed and a few cyber-attacks, such as, Jamming and DDoS (Distributed Denial of Service) were implemented using the base simulation engine of OMNeT++. One of the major features developed in this phase was an interactive GUI for beginner level users. Various simulation results and related insights were presented and the accuracy of UAVSim was demonstrated [20]. This work also describes the technical details of the developed software simulation testbed. In continuation, advanced features like multi-user support, server based centralized simulation, etc., using ubiquitous computing infrastructure, were added to UAVSim and the testbed performance was analysed in different scenarios for different modes of operations for DDoS attack [21].

In this paper, we extend the analysis for DDoS attack with increased number of concurrent users and present detailed analysis for Jamming attack as well. Primary reason behind selection of these two attacks for our performance analysis is the huge computational resource requirement for simulation of both of these attacks. Most cyber attacks which aim to take control over the subject, do not involve large amount of data transmission, instead, these attacks only require minimum data transmission in terms of some unique command and control messages. Therefore, if high computing resource consumption attacks (from the testbed perspective) can be simulated, it would prove

the testbed's capability to simulate all other attacks which will consume less resources on the underlying computing infrastructure.

4. UAVSim: Design and Features

As discussed in Section 2, the primary focus of developing a simulation testbed has been simulating the behavior of a single UAV to check its proper functioning. Nowadays, use of large number of UAVs in various applications demands for their performance test in an existing swarm of aircrafts, especially when the US Government is working on integrating UAVs in the US National Airspace.

4.1. Testbed requirements

There are other important requirements to be met to make such a simulation testbed more useful. Testing of security measures in terms of hardware as well as software should be supported. Impact evaluation, on system components and overall performance must be supported as well. The testbed should allow use of various UAV models, developed in UAVSim as well as other popular software. In order to make it available to UAV-experts, who are not technically sound, the testbed should have an interactive and easy to use GUI. For advanced users, an advanced GUI can also be an option. The environment designed should be also verified and validated in order to correctly simulate the UAV model. One of the most important aspects of communication should be addressed and the UAV should be treated as a network of components which replicates the component communication behavior. Other environment variations, such as contested communication, collaborative control, mobility models and mission paths should be modeled and addressed.

4.2. Design

Keeping the above requirements in mind, UAV component level modeling, individual simulation, attack classification and attack modeling were performed [6] and later a software simulation testbed, UAVSim, for simulations of all sizes of UAV networks was developed and in-depth design was presented in [20]. Preliminary performance evaluation was done in [21] which covered simulations for the DDoS attack with maximum number of users limited to 6. UAVSim is developed using the open source network simulator OMNeT++ and one of its independently developed open source modules called INET for mobility and related protocols. For satellite communication, another open source component called OS3 (Open source satellite simulator) has been used. Network design and higher level code is coded in NED, a language specifically designed for OMNeT++ while the lower level functionality is coded in C++ [22]. Figure 1 shows the design of UAVSim in the right half. The left half shows various modes of simulations.

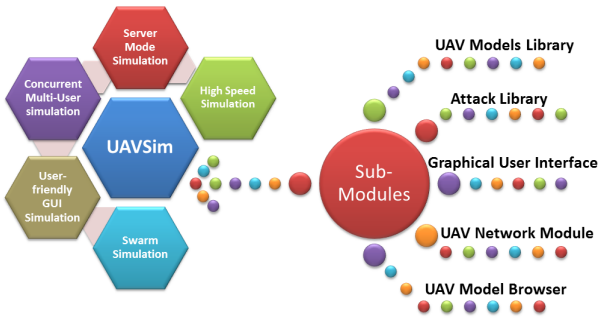

Figure 1. UAVSim: Modes of operations and various components

4.3. Features

Security simulation being the primary focus and one of the most important features of UAVSim, several attacks have been implemented in the attack library of the testbed. Further, basic and advanced models of popular UAVs have been designed. The interactive GUI allows beginner and intermediate users to vary various parameters while advanced users can directly make changes to the simulation configuration files. Apart from supporting mobile wireless communication and UAV component level modeling capability, UAVSim also supports detailed network analysis at lower levels of the protocol stack. Further, attacks targeting different layers can also be designed, launched and tested in UAVSim. One of the most important features of UAVSim from user perspective is its user-friendly design and its ability to work on generic computing environment. Figure 1 summarizes the important features and modules of UAVSim.

User-friendly GUI Simulation. The simulation testbed supports both command line and graphical user interface. We have developed a custom GUI for UAVSim which lets basic users select possible options for some parameters. Users do not get a lot of independence in the basic GUI. While, the advanced users can edit all other parameters as well using the configuration file in the simulation project. Although the GUI might cost some resource, it definitely can be counted as one of the performance parameters as the testbed has been designed to be used for all levels of users, basic, intermediate or advanced.

Server Mode Simulation. In order to enhance performance, a high performance computer can also be utilized in our simulation testbed. The connection details to a server or high performance computer can be set using the GUI by the administrator or the person setting

up the testbed for initial use. It should be noted that the core testbed simulation files should be installed on the server prior to this setup and ssh should be enabled on the high performance computer to enable seamless communication and execution.

High Speed (No-GUI) Simulation. While the testbed has a well-designed GUI, the aim of providing a non-GUI option was to enhance the performance. There is an option of express command mode execution as well, which prints the minimum required simulation statistics in order to let the user know that the simulation is running and the computer is not frozen. Using this option, the simulation can be run at the maximum speed and thus gives the best performance. This mode was primarily designed for Server mode simulation because the communication with the server might slow down execution. Nevertheless, this mode can be used on the desktop mode as well as server mode.

Concurrent Multi-User Simulation. The testbed also provides a multi-user option which allows multiple users to concurrently run their simulations through their individual machines. This option utilizes the Server Mode of the testbed. As mentioned before, if the testbed needs to be used for high speed simulation or, by several users at the same time, a non-GUI server option is available. One of the most important prerequisites to use this option is the connection oriented access availability on the server to all the user accounts. This is necessary in order to enable independent simulation for each user. The core simulation modules need to be installed on the server while users remotely connect to the server using UAVSim. The UAVSim, once configured with the server and connection details, automatically connects to the server and displays results in a console window. It should be noted that the multi-user simulation is only available in non-GUI option.

Swarm Simulation. Although the simulation testbed was primarily developed for UAVNet security simulation, it also supports UAV swarm simulation. This feature enables users to test the network behavior when large numbers of UAVs are used for any specific application. The use can be commercial, civil or military in nature but in case of swarms, usually it should be a sensor based application with a large number of sensors. The performance for swarm simulation using a large number of nodes has also been evaluated.

4.4. Attack Anatomy

Here we have described the design and implementation of the two selected attacks in brief. A detailed explanation is not really necessary because of their well known anatomy. As mentioned earlier, the focus was to select two resource intensive attacks - one which attacks a single UAV and second, which attacks multiple UAVs - in order to measure the performance accurately. DDoS attack was chosen as the attack which will target a single UAV while a Jamming attack will target multiple UAVs in the mission area. Both of these attacks are discussed below.

DDoS Attack - Single Target. The DDoS (Distributed Denial of Service) attack aims at loss of communication through network congestion. This is achieved by making the host appear unavailable to other hosts in the network, mostly, due to the increase in response time and almost 100% packet drop. The reason behind huge packet drop and response time is large number of adversary hosts sending frequent requests to the host being attacked, the requests might be PING, SYN or any other kind of packet demanding an acknowledgment.

This attack has been implemented in UAVSim using a number of attack nodes, which can be defined by the user based on the total number of UAVs in the network. Although it has been proved experimentally that even a single host is capable of launching a successful DDoS attack using a PING packet because of its small payload size [23]. In order to implement the DDoS attack, we have used the traditional way of transmitting spoofed packets to a single host from several attack nodes. All attack nodes behave similar to regular UAV hosts and are assigned the IP addresses of the same range in order to make them indistinguishable from other trustworthy UAVs. During the simulation analysis, we have varied this number to check the success rate of attack in different scenarios. All attack nodes transmit packets to a single UAV host in order to make it unreachable and thus, launch a successful attack on a single target. Approximate time taken to successfully launch this attack is only few seconds for all simulations and packet loss for the attacked node reaches 99.9% in less than 2 seconds.

Jamming Attack - Multiple Target. Any kind of radio signal based communication can be interrupted using Jamming, which involves transmission of noise in the mission area. This attack results in loss or corruption of packets. The noise usually spans over all the frequencies and prevents communication at any frequency. If the attack node has a powerful transmitter, a signal can be generated that will be strong enough to overwhelm the targeted signals and disrupt communications. The most common types of signal used in an jamming attack are random noise and pulse [24]. Jamming equipment is readily available in the market as well as on online shopping websites like *amazon* and *eBay*. In addition, jamming equipment can be mounted from a location remote to the target networks. This attack can not be handled by most of modern wireless devices and is relatively easier to launch.

This attack has been implemented in UAVSim by creating several attack nodes which send noise signals to all the hosts in a round robin fashion over different frequencies. The number of these attack nodes can be varied in the simulation. Transmitting random signals to all UAV hosts will launch a multiple target attack as aimed. Various techniques have been developed in the recent past to take care of jamming attacks, most of them expect the attack to be in a particular signal frequency and therefore, most of these methods use frequency hopping and spread spectrum communication to counter Jamming attack [25]. That is why we have implemented total-frequency band jamming attack. Although post-2010, several researchers have proposed various anti-jamming encoding, encryption, etc., we have not addressed those techniques in our jamming attack implementation. Total time for successful completion of this attack takes a little more time than DDoS attack and is about 5 seconds for most simulations while packet loss for all hosts reach above 90% for all nodes.

5. Performance Analysis and Results

The primary focus of this paper is to demonstrate the usefulness of the simulation testbed even with regular computing infrastructure. Usually, in an academic or research setup, where resources are constrained, purchasing expensive high end computing infrastructure might be quite difficult or even impossible. Therefore, the testbed should allow users to use it for any kind of UAV network in a cost-effective manner. As mentioned in Section 2, various works which allow simulation of UAV swarms for various purposes, use quite high-end computing facilities and are not available to the public. On the contrary, our simulation testbed is designed to work with already existing simulation engine and components which are open source and thus, free to use. At the same time, this testbed does not need expensive machines to get results. Clearly, one might have to compromise on computation time.

Another point to note is the expected increase in simulation run times for Jamming attack. DDoS works on principle of sending huge number of packets to one node causing congestion and stopping it from communicating with others. On the other hand, Jamming works by transmitting noise on all frequencies so that communication is jammed due to noise traffic on the wireless channel. Therefore, in order to implement DDoS attack, lesser number of attack nodes are required as only single node working at a frequency needs to be jammed. On the contrary, Jamming requires all the frequencies to be jammed for a successful attack.

5.1. Simulation Setup

Keeping in mind the primary goal of performance evaluation, it is necessary to have a clear understanding of what kind of Hardware or Software environment has been used in order to make sure that performance claims are accurate. Here we have described the hardware, software as well as simulation testbed setup used for our simulations.

Hardware Setup. The PC being used for simulations has a Intel® Core^{TM} i7-3770 CPU (1×3.40 GHz 4-core, L2/L3 Cache: 1 MB/8 MB) and a system memory of 8.0 GB while the server used during the server mode simulation has a Intel® Xeon® Processor E5-2630 (2×2.30 GHz 6-core, L2/L3 Cache: 1.5/15 MB) and a system memory of 64.0 GB. Apart from these, for use in multi-user concurrent simulations, a few generic laptops (more or less 2-3 years old) were used and their configurations are not listed due to negligible computation taking place in those systems and hence, no impact on overall testbed performance.

Software Setup. Both of the systems defined in previous paragraph, the PC and the Server machines, run Ubuntu version 12.04 LTS. Needless to say, the Server runs the x64 server version while the PC is running the x64 desktop version. Both has the OMNeT++ version 4.2.2 with the INET version 2.2 and CNI_OS3 version 1.0. As mentioned before, our simulation module UAVSim makes use of OMNeT++ and these two open-source plugins to accurately simulate a UAVNet.

Testbed Setup. All simulations were 300 seconds long while actual time taken to finish this simulation were observed. As established before in [26], actual time to attack a time-sensitive military system such as a missile defense system is only a couple of seconds. Even in our simulations, attacks were launched right after the simulation started and it was noted that the attacks were successful in only a couple of seconds. The most basic UAV Model has been used for our simulation as using more advanced models detailing various sub-modules would clearly increase simulation times. Advanced UAV models can be used while implementing more complex confidentiality or integrity compromising attacks. Frequency for UAV communication is fixed at 5 GHz for Single Target attack scenario while it varies between the range of 5-15 GHz for use in Multiple Target attack scenario.

Several cases were evaluated for different types of simulation. For Case I, running time and swarm behavior analysis, we have used both the number of attack nodes as well as the number of UAVs. In Case I_a, the number of UAVs was varied from 50 to 500 and in Case I_b, number of attack nodes was varied from 2 to 20. In case of Jamming attack, run time increased to days after 350 nodes, therefore, that was the limit

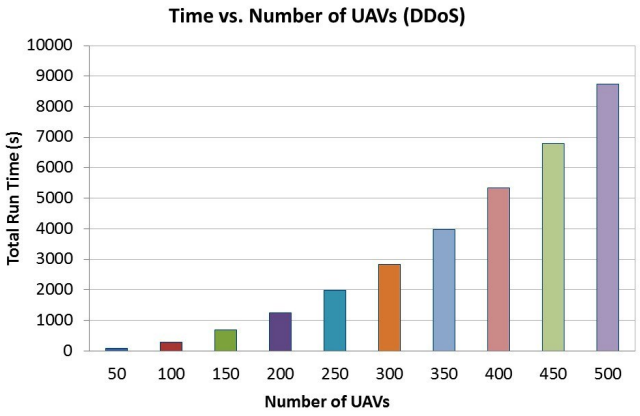

Figure 2. Run time variation with increasing number of UAVs for DDoS attack (same frequency of all hosts)

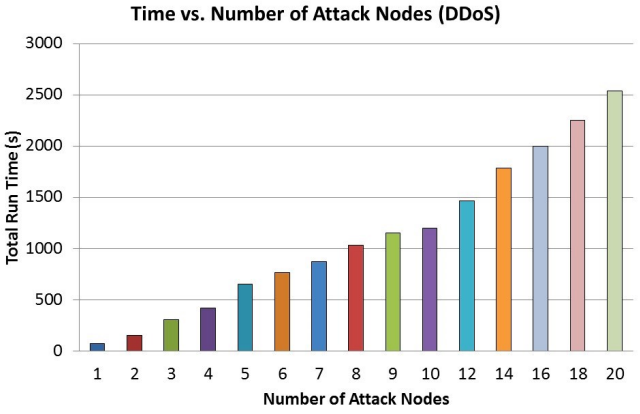

Figure 3. Run time variation with number of attack nodes for DDoS attack

for Jamming attack. Further, for checking multi-user behavior, two separate analysis were done with separate cases. Case II, where performance of swarm behavior with multiple concurrent user was evaluated. Case II_a, where 50 nodes were used for swarm behavior analysis while concurrent number of users was increased from 1 to 8. Case II_b, number of nodes was increased to 100. Case III, where performance of the testbed was evaluated for attack simulation. Case III_a being 5 attack nodes and Case III_b being 10 attack nodes, keeping the number of UAV nodes as 10 for both the scenarios, III_a and III_b. These three cases will be referred to during the discussion and analysis.

5.2. DDoS Attack - Single Target

This subsection covers the results for all the simulations for Distributed DOS attack. The various simulations were done for the 3 above mentioned cases, namely Case I, II and III. We have analysed the effect of increasing number of UAVs, attack nodes, concurrent users in Server Mode operation and use of GUI.

Number of UAVs. For this analysis, we have used Case I_a (number of UAVs varied). As mentioned earlier, Case I_a involves use of regular UAV nodes in order to ascertain the UAVSim capability for UAV swarm simulation other than the primary capability of UAVNet security simulation. It is clear from Fig. 2 that the testbed run-time varies exponentially with the increasing number of UAV nodes for this attack. Clearly, the run time is directly proportional to the powers of each 50 nodes and thus, is easily predictable for higher number of UAVs. Looking at the simulation times, it can be argued that large number of UAVs can be used for swarm simulations using single frequency scenarios.

Number of Attack Nodes. For the second analysis, we have used Case I_b (varying number of attack nodes). Fig. 3 depicts the performance for simulations with increasing number of attack nodes while the number of UAVs is kept constant as 10. Looking at the trend obtained, it is understood that the variation is exponential with respect to the number of attack nodes and instead of multiples of 50, here we have multiples of 2, therefore, large number of attack nodes may not be used for security simulations. Keeping in mind the number of attack nodes which can be simulated in reasonable time, using large value (more than 50) for this variable is neither possible nor required.

Graphical User Interface. The third performance metric was the use of GUI (graphical user interface) which displays the network animation. It is understood that having a GUI displaying the network animation and various network statistics during a CPU intensive operation might impact the performance. Therefore, we used Case I_b, where we varied the number of attack nodes and measured the speed of simulation for GUI and non-GUI options. Fig. 4 show the results obtained for GUI and non-GUI options on the server as a blue dashed line and a black dotted line. The red dashed line shows the percentage difference between the two modes with respect to the lower value (non-GUI option).

As shown in Fig. 4, the non-GUI run time follows a non-linear polynomial trend with respect to the number of attack nodes. The percentage change between GUI and non-GUI options for a DDoS attack is not more than 7% for all cases with most cases being between 2 – 5%. Therefore, it can be said that the performance is not much affected by use of GUI for this particular attack.

Number of Concurrent Users - Single Frequency Swarm Scenario. The fourth performance test was done varying number of concurrent users accessing the simulation framework in the server mode option. As mentioned earlier, the server based simulation works only in non-GUI mode to enhance execution performance and reduce the server to PC communication. In our earlier

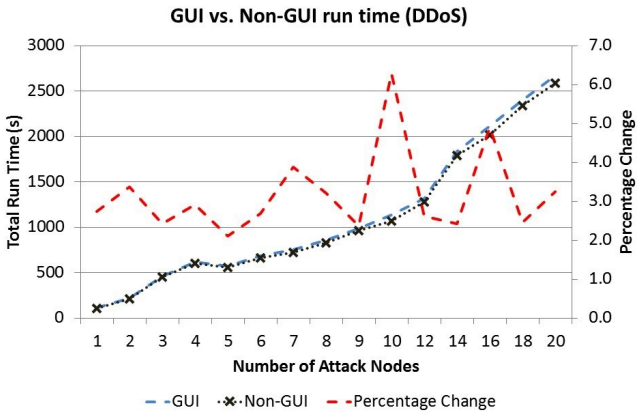

Figure 4. Run time variation with GUI and Non-GUI options, and the percentage change in two options for DDoS Attack

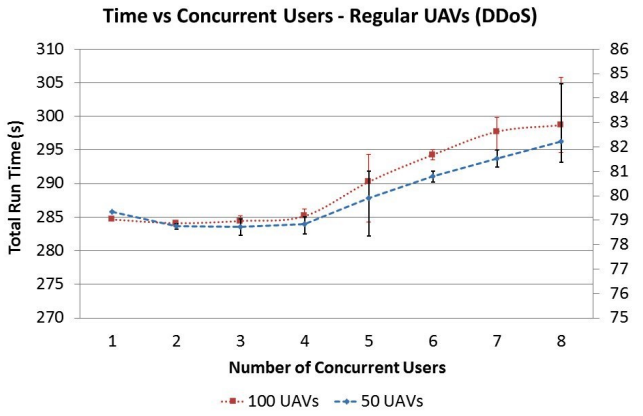

Figure 5. Run time variation of swarm simulation with increasing number of concurrent users in server mode operation for single frequency band

performance evaluation [21], the number of users was varied from 1 to 6. Here, we varied the number of concurrent users from 1 to 8 and extended the analysis to Jamming attack as well which is covered in the next subsection.

Fig. 5 show the evaluation results for Case II_a (50 UAV hosts) and II_b (100 UAV hosts) using the DDoS attack scenario. As mentioned earlier, these scenarios do not use any malicious node and all UAVs communicate at a single frequency of $5GHz$. This simulation was intended to analyze the performance variation for multiple concurrent users, simulating a swarm using single frequency in absence of an attack. Please note that the two separate vertical axes show the variation of total run time for the two Cases, II_a and II_b. The error bars show the maximum and minimum time while the points depict the average time.

Number of Concurrent Users - DDoS Security Simulation. The final analysis for DDoS attack targets the

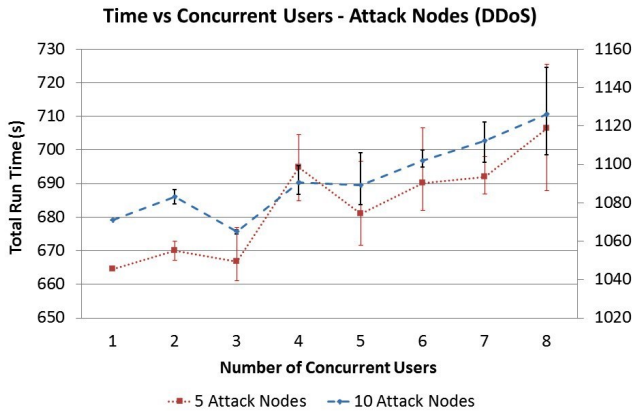

Figure 6. Run time variation of security simulation with increasing number of concurrent users in server mode operation for DDoS attack

performance evaluation of the testbed with multiple users using it concurrently in server mode operation. To this end, Cases III_a (5 attack nodes) and III_b (10 attack nodes) were used. Fig. 6 show the test results for this experiment. As mentioned earlier, the number of malicious hosts was changed for the two cases, keeping number of regular UAVs as 10. The number of malicious hosts used is much lesser than Case II because malicious nodes generate more traffic in the network and are responsible for increasing the execution time, as found in the two initial experiments discussed in this section. Just like the last analysis, the two separate vertical axes show the variation of total run time for two different numbers of attack nodes. The error bars show the maximum and minimum time while the points represent the average time.

5.3. Jamming Attack - Multiple Targets

This subsection covers the results of all simulations for Jamming attack. The various simulations were done for the 3 cases mentioned in subsection 5.1, namely Case I, II and III. We have analyzed the effect of increasing number of UAVs, attack nodes, concurrent users in Server Mode operation and use of GUI.

Number of UAVs. We have used Case I_a (number of UAVs varied) for this experiment. Please note that this simulation might seem similar to DDoS attack UAV-only simulation but it should be noted that here, multiple frequencies are being used for communication rather than single. As mentioned before, the frequency range for UAV-UAV communication lies between $5 - 15GHz$ for this case. This has been done in order to make sure that all frequencies are jammed in the attack area. It is clear from Fig. 7 that the testbed run-time varies exponentially with the increasing number of UAV nodes.

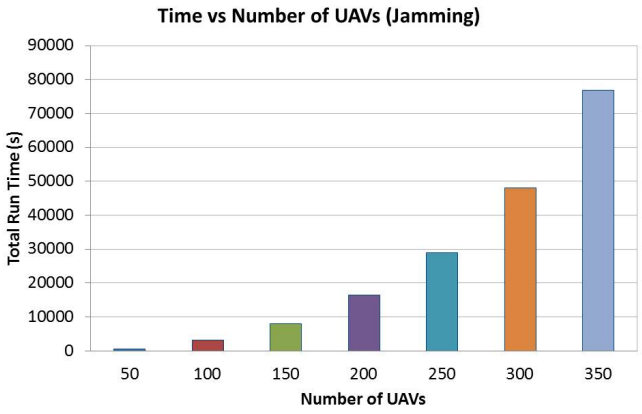

Figure 7. Run time variation with increasing number of UAVs for Jamming attack (different frequencies of hosts)

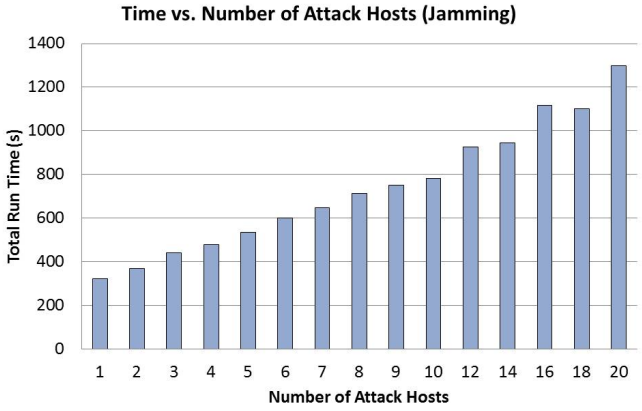

Figure 8. Run time variation with increasing number of Attack Nodes for Jamming attack (different frequencies of hosts as well as attack nodes)

Clearly, the run time is directly proportional to the power of each 50 nodes and thus, is easily predictable for higher number of UAVs and therefore, large number of UAVs can be used for swarm simulations scenarios where multiple frequency channels are used. It should also be noted that the exponent might be higher than that found in single frequency swarm simulation (DDoS UAV-only simulation) and thus, it can be seen that even for 350 nodes, the run time reaches almost 24 hours compared to one hour run time in case of single frequency communication.

Number of Attack Nodes. For the second analysis of Jamming attack, Case I_b (varying number of attack nodes) has been used. Fig. 8 shows the performance for simulations with increasing number of attack nodes while the number of UAVs is 10. It should be noted that the attack simulation trend for Jamming attack is non-linear polynomial instead of exponential as it was in case of DDoS attack. Since the trend is not

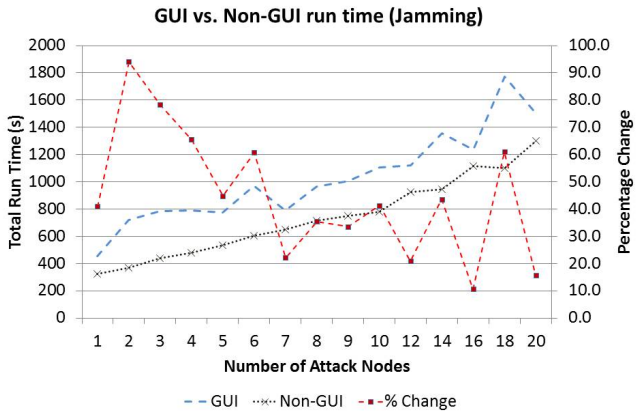

Figure 9. Run time variation with GUI and Non-GUI options, and the percentage change in two options for Jamming attack

exponential, large number of attack nodes may be used for security simulations of Jamming attack. Comparing the two attack scenarios, it can be noted that the run time for Jamming attack for Case I_b is roughly half of DDoS attack simulation run time. Once again, keeping in mind the number of attack nodes required for a successful attack, large values (more than 100) can not and need not be used.

Graphical User Interface. We used Case I_b (varying number of attack nodes) to measure the speed of simulation for GUI and non-GUI options. Fig. 9 show the results obtained for GUI and non-GUI options on the server as a blue dashed line and a black dotted line respectively. The red dashed line (showing a heartbeat trend) represents the percentage difference between the two modes with respect to the lower value (non-GUI option).

The percentage change between GUI and non-GUI options for a Jamming attack is quite random and higher for lower number of attack nodes. Mostly, it is between $10-70\%$. This trend is exactly opposite of the trend shown by DDoS attack simulation. It is quite clear that the performance is affected very badly by use of GUI for a Jamming attack. The significant changes in performance for Jamming attack in current and previous case can be attributed to its anatomy and implementation and will be discussed in subsection 5.4.

Number of Concurrent Users - Multiple frequency swarm simulation. This performance test was performed varying number of concurrent users using the simulation testbed on a single server. The number of concurrent users was varied from 1 to 8, and simulation run time for Cases II_a (50 UAV hosts) and II_b (100 UAV hosts) were evaluated. Fig. 10 show the evaluation results using the Jamming attack scenario. These scenarios do not use any malicious node and all different UAVs communicate at different frequencies in the range

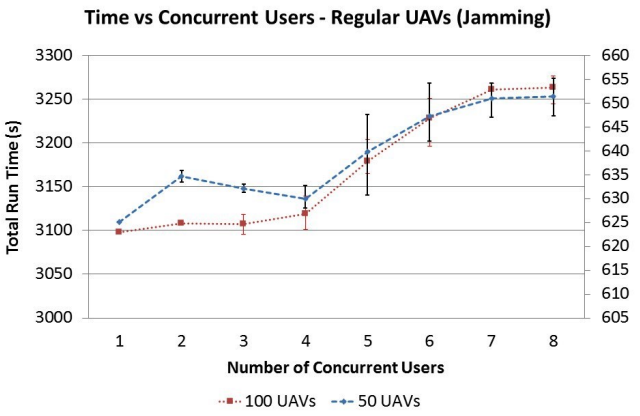

Figure 10. Run time variation of swarm simulation with increasing number of concurrent users in server mode for variable frequency bands (Jamming attack scenario)

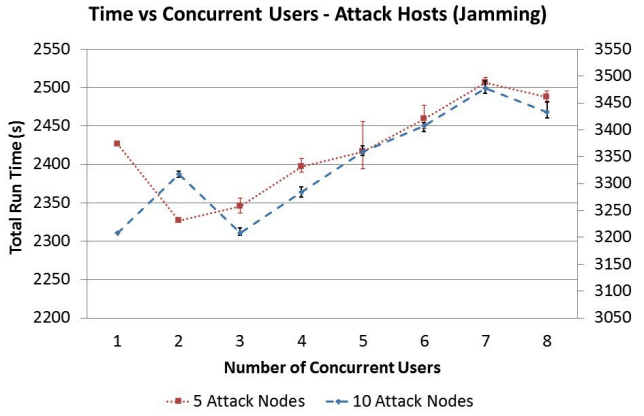

Figure 11. Run time variation of security simulation with increasing number of concurrent users in server mode operation for Jamming attack

of $5-15GHz$. This simulation was intended to analyze the performance variation for multiple concurrent users, simulating a swarm using multiple frequencies in absence of an attack. Please note that the two separate vertical axes show the variation of total run time for the two Cases, II_a and II_b, lower numbers, obviously, depicting Case II_a. The error bars show the maximum and minimum time while the points depict the average time.

It should be noted that both Cases II_a and II_b follow similar trend after certain number of users and the run time seems to be becoming invariable. Another important aspect to note is the average percentage variation in both cases is less than 5% between time taken for single and 8 concurrent users.

Number of Concurrent Users – Jamming attack security simulation. The final performance test for Jamming attack involves the performance analysis for the Jamming attack simulation with multiple users using the testbed concurrently. Run time values for Cases III_a (5 attack nodes) and III_b (10 attack nodes) were evaluated. Fig. 11 show the performance test results for these cases. The two separate vertical axes in this evaluation also show the variation of total run time for two different numbers of attack nodes. The error bars show the maximum and minimum time while the points represent the average time.

It can be again noted that the plot for both cases follow similar trend once number of concurrent users increase to 3. Similar to the previous evaluation of swarm simulation, the overall percentage variation between maximum and minimum run times for each case is less than 10%. Maximum for both cases was at 7 concurrent users while minimum for Case III_a at 2 and for Case III_b at 3 concurrent users.

5.4. Analysis

The various test categories for number of UAVs, number of attacks hosts, GUI/non-GUI option and number of concurrent users were performed, which give us valuable insights in terms of the operational capability of the testbed. Although some simulation times are quite high in case of swarm simulations of large number of nodes, the performance is reasonable for security simulations. Some important points which can be noted from the analysis are as follows:

- Total run time varies exponentially with the number of attack nodes in security simulations as well as number of UAVs in swarm simulation. Despite of the trend, it should be noted that the variation in attack nodes are only by 2 while in case of UAVs, the number is varied by 50. This gives a clear indication of how many attack nodes and UAV hosts can be deployed in simulation scenarios.
- Using the GUI for any security simulation has little impact on performance for DDoS attacks but the variations are higher for Jamming attack. The Jamming attack requires more processing in terms of creation of channels for different frequencies and transmission of packets but this can not be attributed to slower execution times due to GUI. Therefore, it can be concluded that use of network animation while using several communication channels requires more processing in the base simulation framework of OMNeT++.
- Performance analysis for multiple users using the testbed in server mode reveals that performance gets affected with increasing number of users but average variation reduces as number of concurrent users increases after 4.

- The average simulation time saturates after certain number of users and shows a trend of becoming invariable with respect to number of concurrent users. The variation in minimum and maximum shows that total system performance is not affected much.
- It can be also noted that the simulation times for concurrent user analysis follows the same trend for same cases evaluated for different number of UAVs or attack nodes after number of concurrent users increases more than 4. This implies that immaterial of the value, the trend would be similar and thus, the run time can be estimated for higher number of users using the obtained trend.
- The attack simulation run time for 20 attack nodes for both types of attacks took approximately half an hour. Practically, the number of attack host in order to launch such attacks are much less. For example, we need 4 attack nodes for a GPS spoofing attack [27] and thus, the simulation capability is quite extensive.
- Since the variation in run time for concurrent users is not exponential, the simulation testbed seems quite capable of handling more than 20 users concurrently on a regular server. The evaluation was done for up to 8 users and showed no alarming trend.
- Although most of the trends are similar for both classes of attacks, the times are much higher for Jamming than DDoS attack simulation. The primary reason behind this is the anatomy of the simulation for these two attacks. Successful execution of a Single Target DDoS attack requires one node to be stopped from communicating while in Jamming, all frequencies, and thus, all nodes are required to be stopped from communicating. The underlying simulation engine of OMNeT++ simulates single object for a single channel (single frequency) and packets are transmitted in that channel within the same object (channel). When several channels are used, each channel is a different object and an inter-process communication takes place between two object for packet transmission and thus, causes increase in total run time for simulations related to Jamming attack.

6. Conclusion

Two classes of attacks - single and multiple target - were simulated in the in-house developed simulation testbed (UAVSim) for UAVNets. Simulation run time analysis for UAVSim were presented to demonstrate its use in generic computing environment. Various simulations indicate that the performance of the testbed is reasonable and allows users to adjust various options according to the requirement. Along with attack simulation involving single and multiple targets, the testbed was proved to be capable of simulating large UAV swarm networks. Single target - single frequency network as well as multiple target - multiple frequency system (where different frequencies are used for communication between different UAVs) simulation was demonstrated. Overall use of UAVSim can also be extended in other domains of interest involving UAV networks.

Performance for server based concurrent multi-user operation was also tested for different scenarios using different number of simultaneous users were evaluated and the testbed was found to perform very well. Although a maximum of 8 concurrent users-scenario was tested, it is understood that for a reasonable amount of concurrent users, the testbed will perform without much delay in simulation. Interactive GUI, additional result analysis module, model browsing capability (from other model development software), enhanced high speed mode of operation, support of concurrent users, etc. are some of the features which makes this software simulation testbed an ideal simulation environment for UAV simulations in generic computing environment.

References

[1] IANS, "Mumbai pizza delivery drones raise security buzz." http://economictimes.indiatimes.com/industry/services/hotels-/-restaurants/mumbai-pizza-delivery-drones-raise-security-buzz/articleshow/35486155.cms, 2014. [Online; Last accessed: 30-July-2014].

[2] D. Aamoth, "Delivering Domino's Pizza by Unmanned Helicopter: What Could Possibly Go Wrong?." http://techland.time.com/2013/06/03/delivering-dominos-pizza-by-unmanned-helicopter-/-what-could-possibly-go-wrong/, 2013. [Online; Last accessed: 30-July-2014].

[3] A. Chang, "With Prime Air, Amazon plans to deliver purchases via drones." http://articles.latimes.com/2013/dec/02/business/la-fi-tn-amazon-prime-air-20131202, 2013. [Online; Last accessed: 30-July-2014].

[4] M. Bryson, A. Reid, C. Hung, F. Ramos, and S. Sukkarieh, "Cost-Effective Mapping Using Unmanned Aerial Vehicles in Ecology Monitoring Applications," in *Experimental Robotics* (O. Khatib, V. Kumar, and G. Sukhatme, eds.), vol. 79 of *Springer Tracts in Advanced Robotics*, pp. 509–523, Springer Berlin Heidelberg, 2014.

[5] FAA, "Integration of Civil Unmanned Aircraft Systems (UAS) in the National Airspace System (NAS) Roadmap, First Edition." http://www.faa.gov/about/initiatives/uas/media/uas_roadmap_2013.pdf, 2013. [Online; Published November 2013].

[6] A. Javaid, W. Sun, V. Devabhaktuni, and M. Alam, "Cyber security threat analysis and modeling of

an unmanned aerial vehicle system," in *2012 IEEE Conference on Technologies for Homeland Security (HST)*, pp. 585–590, Nov 2012.

[7] A. Kim, B. Wampler, J. Goppert, I. Hwang, and H. Aldridge, "Cyber attack vulnerabilities analysis for unmanned aerial vehicles," *The American Institute of Aeronautics and Astronautics: Reston, VA, USA*, 2012.

[8] P. Lu and Q. Geng, "Real-time simulation system for UAV based on Matlab/Simulink," in *2011 IEEE 2nd International Conference on Computing, Control and Industrial Engineering (CCIE)*, vol. 1, pp. 399–404, Aug 2011.

[9] J. Zhang, Q. Geng, and Q. Fei, "UAV flight control system modeling and simulation based on FlightGear," in *International Conference on Automatic Control and Artificial Intelligence (ACAI 2012)*, pp. 2231–2234, March 2012.

[10] J. Goppert, A. Shull, N. Sathyamoorthy, W. Liu, I. Hwang, and H. Aldridge, "Software/Hardware-in-the-Loop Analysis of Cyberattacks on Unmanned Aerial Systems," *Journal of Aerospace Information Systems*, vol. 11, no. 5, pp. 337–343, 2014.

[11] Y. Qiang, X. Bin, Z. Yao, Y. Yanping, L. Haotao, and Z. Wei, "Visual simulation system for quadrotor unmanned aerial vehicles," in *2011 30th Chinese Control Conference (CCC)*, pp. 454–459, July 2011.

[12] T. X. Brown, S. Doshi, S. Jadhav, and J. Himmelstein, "Test Bed for a Wireless Network on Small UAVs," in *In Proceedings of AIAA 3rd Unmanned Unlimited Technical Conference*, pp. 20–23, 2004.

[13] J. Wu, W. Wang, J. Zhang, and B. Wang, "Research of a kind of new UAV training simulator based on equipment simulation," in *2011 International Conference on Electronic and Mechanical Engineering and Information Technology (EMEIT)*, vol. 9, pp. 4812–4815, Aug 2011.

[14] J. Yang and H. Li, "UAV Hardware-in-loop Simulation System Based on Right-angle Robot," in *2012 4th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, vol. 1, pp. 58–61, Aug 2012.

[15] J. Corner and G. Lamont, "Parallel simulation of UAV swarm scenarios," in *Proceedings of the 2004 Winter Simulation Conference*, vol. 1, pp. –363, Dec 2004.

[16] S. Hamilton, T. Schmoyer, and J. Drew Hamilton, "Validating a network simulation testbed for army UAVs," in *2007 Winter Simulation Conference*, pp. 1300–1305, Dec 2007.

[17] S. Chaumette, R. Laplace, C. Mazel, and R. Mirault, "SCUAL, swarm of communicating uavs at LaBRI: An open UAVNet testbed," in *Wireless Personal Multimedia Communications (WPMC), 2011 14th International Symposium on*, pp. 1–5, IEEE, 2011.

[18] M. Ashtiani and M. A. Azgomi, "A distributed simulation framework for modeling cyber attacks and the evaluation of security measures," *Simulation*, p. 0037549714540221, 2014.

[19] E. Pereira, K. Hedrick, and R. Sengupta, "The C3UV Testbed for Collaborative Control and Information Acquisition Using UAVs," in *2013 American Control Conference (ACC)*, pp. 1466–1471, IEEE, 2013.

[20] A. Y. Javaid, W. Sun, and M. Alam, "UAVSim: A simulation testbed for unmanned aerial vehicle network cyber security analysis," in *2013 IEEE Globecom Workshops (GC Wkshps)*, pp. 1432–1436, Dec 2013.

[21] A. Javaid, W. Sun, and M. Alam, "UAVNet Simulation in UAVSim: A Performance Evaluation and Enhancement," in *9th International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities (TRIDENTCOM 2014)*, May 2014.

[22] A. Varga *et al.*, "The OMNeT++ discrete event simulation system," in *Proceedings of the European Simulation Multiconference (ESM-2001)*, vol. 9, p. 185, sn, 2001.

[23] K. M. Elleithy, D. Blagovic, W. Cheng, and P. Sideleau, "Denial of Service Attack Techniques: Analysis, Implementation and Comparison," vol. 3, pp. 66–71, 2006.

[24] T. Karygiannis and L. Owens, "Wireless network security," *NIST special publication*, vol. 800, p. 48, 2002.

[25] K. Pelechrinis, M. Iliofotou, and S. V. Krishnamurthy, "Denial of service attacks in wireless networks: The case of jammers," *IEEE Communications Surveys & Tutorials*, vol. 13, no. 2, pp. 245–257, 2011.

[26] P. Katopodis, G. Katsis, O. Walker, M. Tummala, and J. Michael, "A Hybrid, Large-scale Wireless Sensor Network for Missile Defense," in *IEEE International Conference on System of Systems Engineering, 2007. SoSE '07.*, pp. 1–5, April 2007.

[27] N. O. Tippenhauer, C. Pöpper, K. B. Rasmussen, and S. Capkun, "On the Requirements for Successful GPS Spoofing Attacks," in *Proceedings of the 18th ACM Conference on Computer and Communications Security*, CCS '11, (New York, NY, USA), pp. 75–86, ACM, 2011.