NCS Security Experimentation using DETER

Alefiya Hussain
USC/Information Sciences Institute
Marina del Rey, CA
Email: hussain@isi.edu

Saurabh Amin
MIT Department of Civil & Environmental Engineering
Cambridge, MA
Email: amins@mit.edu

I. Introduction

Networked control systems (NCS) are increasingly being used for operational management of large-scale physical infrastructures, and inherit the vulnerabilities of commercial IT solutions. In recent years, numerous studies have focused on the study of interconnected physical and cyber-based processes of NCS and next-generation supervisory control and data acquisition (SCADA) systems. Especially important is the interdependence between random failures (e.g., due to sensor-actuator faults) and adversarial failures (e.g., due to malicious software) [1]. The existing theoretical analyses typically assume class of attacker-defender models, and rigorously use tools from robust control and game theory to derive safety and performance bounds for NCS models [2]. However, in order to develop practically implementable diagnostic tools and real-time response mechanisms, these attacker-defender models should be benchmarked and evaluated against real-world threat scenarios. Indeed, experimental research in network security highlights the accuracy and level of modeling detail, and focuses on developing techniques for security evaluation by combining real and simulated components. Such experimental research is necessary to complement theoretical performance bounds, and will enable researchers to address new developments in smart infrastructures that face emerging threats, and yet account for the challenges of realism, fidelity, and scale as these networked systems expand in size and functionality.

Several efforts are currently underway for testing and evaluation of new IT security solutions and secure control algorithms for NCS/SCADA systems. There is a body of literature which studies the co-simulation of NCS/SCADA processes (using Matlab, Modelica, Ptolemy, and other hybrid system simulation tools) with simulated network models (using NS2, OMNet++, etc.) [3], [4], [5], [6], [7], [8], [9], [10]. These approaches are sufficient to study NCS performance under unreliable communication networks (with delay, jitter, and packet loss). However, for the purpose of cybersecurity testing and evaluation for NCS/SCADA, emulation based experiments offer a richer class of scenarios. At present, multiple government-industry initiatives are underway for exploring testbed research and development for NCS/SCADA system applications. The DHS Control Systems Security Program (CSSP) and the DOE-OE Control System Security National SCADA Test Bed (NSTB) [5] have offered the red-team blue-team training exercises for asset owners and vendors across different utility sectors over the past three years. The current and past red/blue configurations have been used to install prototyped tools on a limited basis to see if applications would install and operate on real control systems. Testing done at the DoE National Laboratories has also generated considerable interest in extending the existing SCADA training architectures to include testing of a wider range of security scenarios, and making this extension accessible as an academic research testbed.

Our goal is to use the network security testing tools developed by USC-ISI researchers to study network attacks for NCS/SCADA systems. The integration of NCS semantics into DETER has a potential to offer a unique opportunity to use DETER's large-scale network testing capabilities to generate realistic network attacks, and validate resilient control algorithms for maintaining safety and security of NCS. In this paper, we make an attempt to leverage testing capabilities of DETER by incorporating high-fidelity overlays to simulate NCS dynamics, and experiment with a range of cyber-attacks scenarios; in particular, denial-of-service attacks. In particular, we integrate dynamics of multiple NCS with a hierarchically structured network topology, where the individual NCS face different levels of flood-based denial of service attacks. We represent that communication network with different models of background traffic and network topology. The system dynamics is mathematically represented by scalar linear system, and our goal is to study evolution and closed-loop stability under a range of attack scenarios. In our experiments, the forward network path from the plant to the controller, or the backward network path from the controller to the plant, or both are flooded with a large volume of attack packets that compete for bandwidth and storage (queuing) resources at the routers[1] .

In Section II, we briefly discuss the taxonomy of DoS attacks in networked systems and summarize the existing capabilities of the DETER tested. In Section III, we first introduce an experimentation model of the NCS and the discuss the approach for testbed-based emulation. We are specifically focused on using real–world attack tools and mechanisms along with representative models of topology and

[1]While there are several other types of *deception* attacks, where the integrity of the output data is compromised such that the plant and controller can receive incorrect data, in this paper we are primarily focused on flood-based denial of service attacks which impact the timely delivery of the plant and controller feedback.

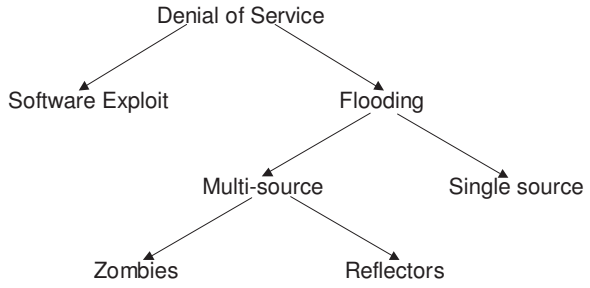

Fig. 1. A Taxonomy of DoS attacks based on the volume of attack packets and the number of attackers. [11].

cross traffic to systematically evaluate the security of linear dynamical systems. In Section IV, we evaluate the impact of the attack on the security and stability of plant, specifically how the attack characteristics, such as, attacker location, start time, and packet size impact the closed-feedback loop between the plant and the controller.

II. DETER CAPABILITIES FOR DOS ATTACKS

In this section we first summarize the taxonomy of real-world DoS attacks to Internet and argue that the same taxonomy is applicable to NCS/SCADA systems that directly or indirectly use Internet connectivity for their operation [2]. Second, we discuss how the DETERLab facilities can be used for evaluating NCS safety and security against DoS attacks.

A. Taxonomy

In a denial of service attack to Internet, a malicious user exploits the network connectivity to cripple the services offered by a victim server, often simply by exhausting the resources at the victim. Typically, these resources include network bandwidth, computational power, or the operating system data structures. A DoS attack can be either a single-source attack (originating at a single host), or a multi-source attack (where multiple hosts coordinate to flood the victim with a large volume of attack packets). The latter is called a distributed denial of service (DDoS) attack. Sophisticated attack tools that automate the procedure of compromising hosts and launching such attacks are readily available on the Internet, and detailed instructions may allow even an amateur to use them effectively.

To launch a DDoS attack, a malicious user first compromises Internet hosts by exploiting security flaws, many of which are openly disclosed by the software vendors themselves. The malicious user then installs attack tools on the compromised host (also known as a zombie), that now becomes available to attack any victim on command. With full control of the zombie, the attacker can construct any packet including illegal packets, such as packets with incorrect header field values, or an invalid combination of flags. Figure 1 presents a broad classification of DoS attacks, namely, software exploits and flooding attacks.

Software exploit-based attacks target specific software bugs in the operating system or an application, and can potentially disable the victim machine with a single or a few packets. A well known example is the SCADA Modbus attack, where a remote attacker can repeatedly force a programmable logic controller (PLC) device or Modbus TCP servers to power cycle by sending a TCP request containing the 08 Diagnostics function code with sub function 01 [12]. Additionally, East et. al [13], have documented several software exploits on the DNP3 protocol for SCADA system. For example, the DFC Flag attack demonstrates that an attacker can generate spoofed, illegal packets with the flag set to incorrectly signal to the master that the outstation device is busy.

Flooding attacks result from one or more attackers sending incessant streams of packets aimed at overwhelming link bandwidth or computing resources at the victim. These attacks can be further classified into (a) zombie directed floods, and (b) reflector attacks.

In *zombie directed flooding attacks*, a malicious user installs attack tools on the host machine that can generate illegal packets. For example, attacks that send a flood of TCP requests to a sensor node that results in power exhaustion at the node [7], or attacks that create a flood of DNP3 messages between the master and the outstation devices [13]. Several canned attack tools are available on the Internet, such as Trinoo, Tribal Flood Network, and SCADA server/client attack tools, that generate flooding attacks using a combination of TCP, UDP, and ICMP packets.

Reflector attacks are used to hide the identity of the attacker and/or to amplify an attack. A reflector is any host that responds to requests. For example web servers or ftp servers that respond to TCP SYN requests with a TCP SYN-ACK packets, or hosts that respond to echo requests with echo replies. Servers may be used as reflectors by spoofing the victim's IP address in the source field of the request, tricking the reflector into directing its response to the victim. Unlike directed attacks, reflector attacks require well-formed packets to solicit a reply. If many reflector machines are employed, such an attack can easily overwhelm the victim without adversely affecting the reflectors or triggering the local IDS. Reflectors can also be used as amplifiers by sending packets to the broadcast address on the reflector network, soliciting a response from every host on the LAN. Unlike directed floods which represent improperly secured hosts, reflectors are often hosts intentionally providing Internet services, and hence reflector attacks may be more difficult to prevent.

B. Attack Generation

The DETERLab facility provides a rich set of resources, tools, and methodologies to conduct high-fidelity, large scale network and cyber security experiments [14], [15]. This facility has been operational since 2003 and is operated by the USC Information Sciences Institute, UC Berkeley, and Sparta Inc. As of December 2010, the DETER testbed has supported 2000

[2]Notice that this taxonomy does not include DoS attacks to NCS communicating over wireless networks, although these are have many similarity with attacks to wirelines networks.

experimenters testing 74 different cyber security technologies on the testbed facility the spans across the USC Information Sciences Institute and UC Berkeley campuses. The main thrusts of research on the testbed include DoS attacks, worm propagation and analysis, botnets, data mining and anomaly detection in networks.

The advantage of using DETERLab for evaluation of NCS/SCADA systems, is that it allows the experimenter to replicate the interactions between NCS components (plants, controllers, etc.) and the attackers with high-fidelity and accuracy, thus providing a unique balance between experiment control and realism. The attack traffic can be generated using either with real–world attack tools mentioned in the previous section or using modeled attack tools provided by the DETERLab facility. Several real–world attack tools are available in binary or executable format and can be activated on the required operating systems and end host configuration. DETERLab also provides a range of DoS attack tools that model the various attack methodologies, command and control structures, attack volumes and attack types with easy to use graphical user interfaces.

We note that there are several experimentation environments available (or currently under development) for NCS/SCADA systems each offering different levels of fidelity and scale [4], [5], [6]. Although these environments have their own benefits, we believe that the DETERLab tools and facilities compliment these efforts. In particular, it allows the experimenter to closely replicate the real–world end host and cyber attack models, and this enables systematic and consistent evaluation of physical control systems in such environments.

III. Experimentation Framework

In this section, we discuss the framework for integrating NCS semantics with the DETER testbed to systematically explore the impact of DoS attacks on evolution and stability of such systems. Our approach is to combine NCS/SCADA system tools and simulation with DETERLab tools and methodologies for networking and cyber security testing. We build on NCS co-simulation technique developed by Branicky and coworkers [10], and develop additional functionalities needed for NCS security experimentation and testing using the DETER testbed. The DETER-based testbed emulation architecture is shown in Figure 2 and has three main components: the system Dynamics, the network interface, and the network dynamics. We now briefly describe each of these components.

System Dynamics: The plant is remotely connected to a controller and the planet-controller communication is subject to network effects, e.g., delays, jitter and packet losses. The denial-of-service can be induced due to random losses or malicious attacks. In our framework, the plant is represented by an ordinary differential equation (ODE) simulation engine and the controller by a simple output feedback policy that has been designed for target-tracking in the absence of the attacks. In our experiments, the NCS dynamics are defined as follows:

$$\dot{x}(t) = Ax(t) + u(t)$$
$$y(t) = x(t)$$
$$u = K[R(t) - y(t)]$$

where $x(t)$ denotes the plant state, $y(t)$ the output, $u(t)$ the control input, and A (resp. K) the system matrix (resp. controller gain matrix), and $R(t)$ the reference trajectory signal. In this abstract, we restrict our attention to scalar dynamics.

Network Interface: The network interface is located at the plant and controller, and enables the integration of NCS dynamics with event-based communication semantics of the DETER testbed. The interface is responsible for sending and receiving data signals across the emulated testbed network. At pre-specified times, represented as δ, the plant simulator provides samples of the system state, and generates the time stamped output signal to the interface. The plant interface sends the data over the emulated network to the controller, but retains it until the end of the time-interval. Upon receipt, the controller interface passes the data to the controller system which then calculates a control input and sends it back to the controller interface at periodic intervals Δ. The controller interface forwards the control input on to network. Upon receipt of the control input, the plant interface immediately forwards the control input to the plant system, which updates the state, and computes the next projected plant state. In order to account for asynchronous and out-of-order packet arrivals, both the plant and controller interfaces, and the component plant and control systems have the capability to roll-back and roll-forward their respective updates.

Network Dynamics The middle section, shown as a cloud, represents the DETER testbed experimentation network and it emulates network and communication dynamics between the plant and the controller. Such a communication network is typically modeled with two primary layers; (a) the physical topological structure of interconnections between the network components and (b) the network traffic layer between the network components [16]. Selecting representative topologies for the communication network has been a subject of significant research over the last several years, while the Internet structure constantly evolves, deployed NCS/SCADA systems also rarely make their underlying network topologies publicaly available due to security reasons [17], [4]. Additionally, the network topological structure is also impacted by the link-level communication technologies, such as wireless, satellite, or wired networks. For example, wireless mobile networks will have a dynamic topological structure that evolves with the movement of the nodes where as wired networks have a static topological structure that does not change frequently. The DETER testbed is primarily a wired testbed and offers several topology generation tools and sample topology catalogs for experimentation [18]. The second layer, the network traffic in the experimentation network determined by the various servers, clients, and peers in the network. To accurately

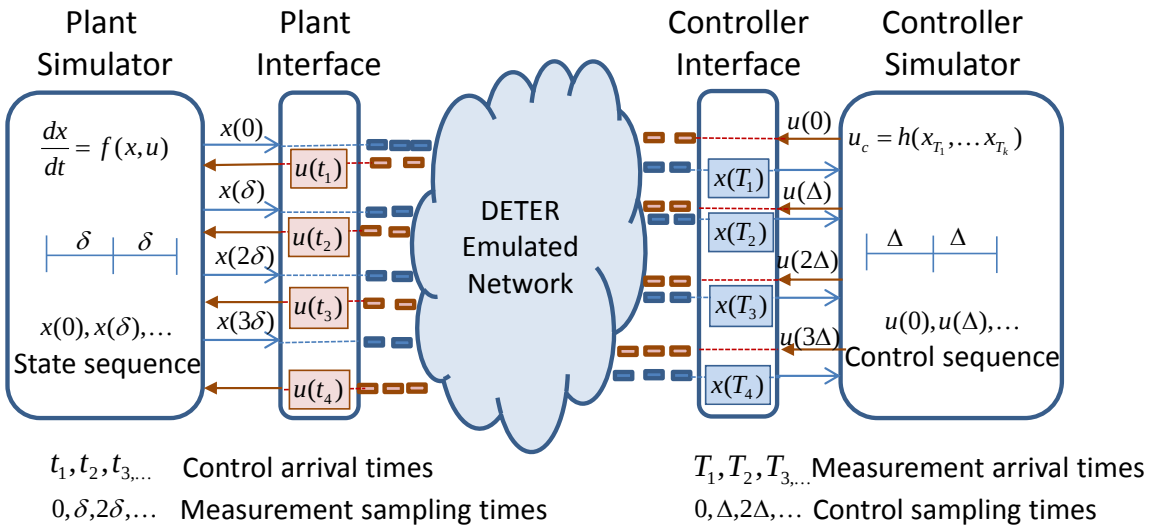

Fig. 2. NCS emulation on the DETER Testbed

model the wide-area networks and the Internet, cyber security experiments typically model three different types of network traffic; (i) background traffic (for example, webserver and client traffic), (ii) foreground traffic that is subjected to attack (for example, control traffic in a NCS/SCADA system), and (iii) attack traffic (for example, DoS traffic between attackers and the victim). DETER provides a rich set of traffic generators, including Harpoon,TCP replay, Apache wget clients for background and foreground traffic, and real and emulated denial of service attack traffic and worm traffic generators [18].

Modularizing our framework as discussed above, enables us to rapidly evolve the cyber network and attack models and the physical NCS/SCADA system models to accurately account for the structural and functional improvements to the Internet and address existing security threats, explore new threats, and meet the challenges of scale and functionality in a timely manner. In the next section, we discuss the specific experimentation scenarios along with metrics and measurements for exploring the impact of DoS attacks on a NCS/SCADA system in an emulated testbed environment.

IV. Evaluation

Using the experimentation framework presented in the previous section, we now systematically evaluate the impact of a denial of service attack on the networked control system. We first discuss how we parametrize our experimentation framework, specifically, the network topology and the network traffic, and then present our results.

A. Emulation Parameters

Our goal is to employ the experimentation framework for understanding the impact of a multi-source flooding denial of service attack on the networked control system. While there are several ways to model the underlying topology as discussed in Section III, we employ a hierarchically structured network topology, with the controller at the root of the hierarchy, and a homogeneous ensemble of six plants located at various levels in the hierarchical tree network. The parameters A and K are chosen from [10], and are identical for all plants. The bandwidth at each link is configured at 1Mbps.

The location of various plants is depicted in Figure 3. The topology has three subnets, and each subnet has two plants. Starting from the top, and going clockwise, first subnet has both plants located at the leaf nodes, second subnet has a plant at a leaf node and a plant located at tree–depth level one from the controller, and the third subnet has a plant at a leaf node and a plant located at tree–depth level two from the controller. Since the plants are located at various levels in the topology, it allow the systematic study of the impact of the attack on the control signals are different levels of aggregation of the attack and network traffic.

Next we deploy attackers at seven leaf nodes in the network as shown in the Figure 3. Each attacker generates a denial of service attack, with a real–world tool called punk [11], that sends a maximum rate stream of TCP packets, where the source address and the source and destination ports are randomized. The size of the attack packet can be configured when the attack is launched and then the attacker generates attack packets at the maximum rate of the network interface. The attack victim is the controller located at the root of the tree. Additionally, all the leaf nodes also generate webtraffic that traverses across the network using DETERLab tools [18]. The networked control system is modeled a linear time–invariant system as discussed in Section III.

B. Effect of Start time

We first study the impact of the attack start time on the plant stability. In Figure 4, y-axis plots the output state from plant #3 at any point in time. The Under no attack conditions, all the plants stabilize and converge. Since NCS rely on real-time

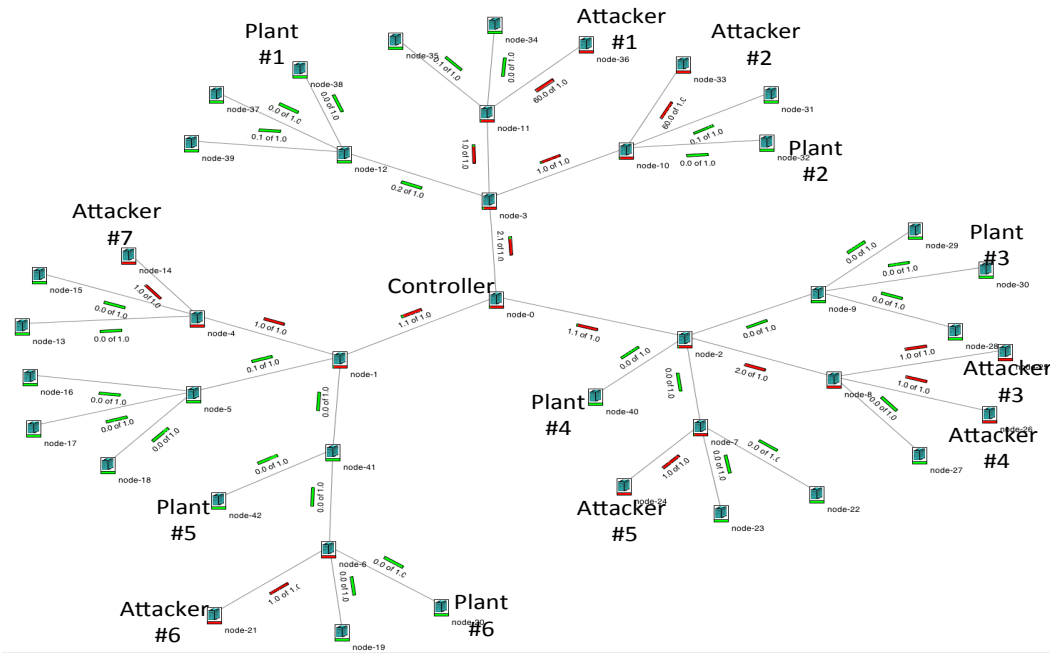

Fig. 3. A hierarchically structured network topology

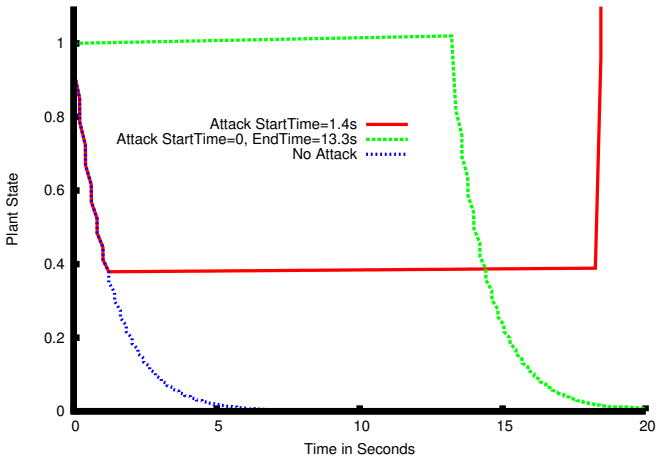

Fig. 4. Effect of Start time of the attack on the plant

feedback, both packet delays and packet losses affect the NCS stability. The route between the plant $\#3$ and the controller, is impacted by three attackers sending packets at the maximum rate of $20Kpps$ with a packet size of 40 bytes. We discuss the stability of plant $\#3$ in the current paper as it is farthest away from the controller and is exposed to the maximum number of attackers. In the final version, we will include a comparative study of the stability of all plants in the network.

We investigate two scenarios; (a) when the attack starts during the plant operation and does not stop for a long period of time (b) when the attack starts before the plant operation and stops after a small period of time. In the first scenario, before the attack starts, the plant starts to converge, but once the attack starts the plant state becomes highly unstable since several feedback messages are lost. In the second scenario, where the attack starts before the plant operation, we observe similar performance, where the plant does not stabilize, but as soon as the attack stops, the plant rapidly converges to a stable condition.

C. Interarrival time

We study the impact of the attack packets on the inter-arrival time of the plant and controller output. Intuitively, since NCS/SCADA systems rely on the timely delivery of the control packets, large attack packets cause longer delays as compared to small attack packets. For the analysis in this section, we measured network packets at the plant and controller using tcpdump [19], and then calculated the time difference between a consecutive plant state output packets.

Figure 5, the y-axis plots the interarrival time of plant $\#4$ against with the packet counts. We discuss plant $\#4$ in the current paper as it is closet to the controller (there are two hops between the controller and the plant) and is exposed to the maximum number of attackers. In the final version, we will include a comparative study of the interarrival time of all plants in the network.

Figure 6 shows the interarrival time as a cumulative distribution function. Plant $\#4$ is two hops away from the

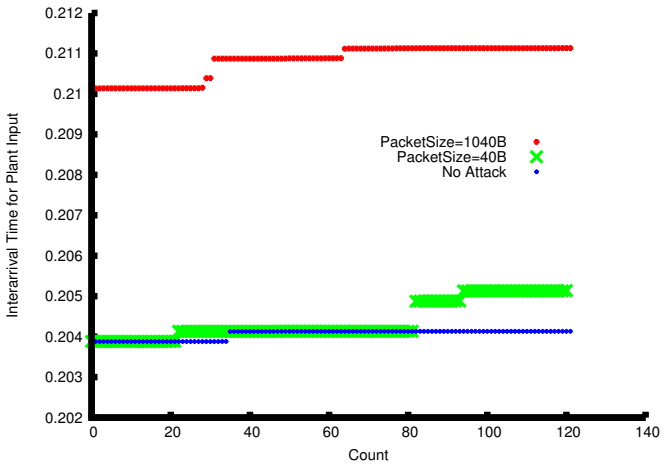

Fig. 5. Effect of an attack on the Inter-arrival time of the plant state

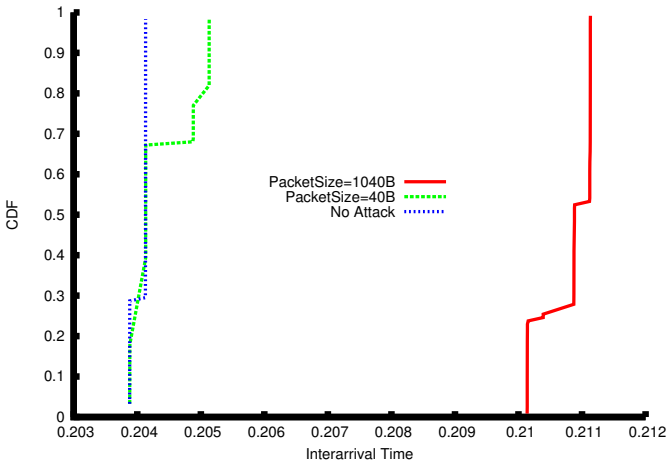

Fig. 6. CDF of the Inter-arrival time

controller and hence the transmission time of the plant state packet, size 52B, from the plant to the controller is 0.200832s. Additionally, there is a 3ms processing delay at the plant in the current implementation. Under no attack conditions, the interarrival time is is bimodal with a difference of 300μs between the two modes which we believe is due to the interleaving for cross traffic.

We specifically investigate two scenarios; (a) when an attacker, generating small packets of 40B starts before the plant operation, (b) when an attacker, generating large packets of 1040B starts before the plant operation. In the first scenario, in addition to the distribution of interarrival time that is similar to the no attack case, we observe approximately a 700μs difference time between the plant state output. The transmission time for a 40B attack packet is 320μs on a 1Mbps link and hence we believe this difference is due to the attack packet traversing from two hops from the attacker to the plant. Similarly, in the second scenario, the transmission time for a 1040B packet is 8.32ms on a 1Mbps link and hence we observe significantly large interarrival times between plant state output depending on how the attack packets, plant state packets, and background traffic interleave.

V. Conclusion

In the final version of this paper, we will further elaborate on our experimentation framework for NCS security and systematically evaluate the safety (and closed-loop stability) properties for benchmark NCS systems using the DETER testbed. We will also discuss how to configure realistic experiment scenarios. Our experimentation framework has three main components: physical dynamics, physical-to-cyber interface, and cyber network model. The physical dynamics (resp. cyber network) are implemented in simulation (resp. emulation). The interface is designed for sending/receiving data across the emulated network, and allows the implementation of event-based semantics. This modular approach provides an environment in which experiments can be rapidly configured, and hopefully evolves to keep pace with the cyber-physical security challenges in the emerging smart infrastructures. Our contribution complements other ongoing projects on NCS experimentation; in particular, we leverage the rich set of tools and methodologies developed for the DETERLab facility. In this extended abstract, we have presented results from our preliminary evaluation of multiple scalar linear systems under a distributed denial of service attack. While the plant dynamics are simple, the novelty here is the simultaneous experimentation of multiple systems under a range of background and attack traffic conditions. Indeed, as indicated by our results, large volume and large attack packets will have a significant impact on NCS stability. Finally, we plan to compare these experimental observations with theoretical performance bounds that have been reported in the literature [20],[21], for e.g., block attacks and strategic jamming attacks.

VI. Acknowledgements

The authors are grateful to Anthony Joseph (Berkeley), Gabor Karsai (Vanderbilt), Blaine Nelson (Berkeley), Suzanna Schmeelk (Rutgers), Galina Schwartz (Berkeley), and John Wroclawski (USC/ISI) for numerous discussions on NCS security. We are especially grateful to Darrel Brower for closely working with us on this project.

References

[1] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. S. Sastry, "Foundations of control and estimation over lossy networks," *Proceedings of the IEEE*, vol. 95, pp. 163–187, 2007.

[2] T. S. Khirwadkar, "Defense against network attacks using game theory," Master's thesis, University of Illinois at Urbana-Champaign, Urbana, Illinois, May 2011.

[3] G. Hemingway, H. Neema, H. Nine, J. Sztipanovits, and G. Karsai, "Rapid synthesis of high-level architecture-based heterogeneous simulation: A model-based integration approach," *SIMULATION*, p. 16, 03 2011.

[4] D. C. Bergman, "Power grid simulation, evaluation, and test framework," Master's thesis, University of Illinois at Urbana-Champaign, Urbana, Illinois, May 2010.

[5] I. N. L. (INL)., "National scada test bed program."

[6] A. Davis, "Developing SCADA simulations with c2windtunnel," Master's thesis, Vanderbilt University, Nashville, Tennessee, May 2011.

[7] A. Giani, G. Karsai, T. Roosta, A. Shah, B. Sinopoli, and J. Wiley, "A testbed for secure and robust scada systems," *SIGBED Rev.*, vol. 5, pp. 4:1–4:4, July 2008.
[8] M. Liljenstam, J. Liu, D. M. Nicol, Y. Yuan, G. Yan, and C. Grier, "Rinse: The real-time immersive network simulation environment for network security exercises (extended version)," *Simulation*, vol. 82, pp. 43–59, January 2006.
[9] A. Hahn, B. Kregel, M. Govindarasu, J. Fitzpatrick, R. Adnan, S. Sridhar, and M. Higdon, "Development of the powercyber scada security testbed," in *Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research*, CSIIRW '10, (New York, NY, USA), pp. 21:1–21:4, ACM, 2010.
[10] A. T. Al-Hammouri, M. S. Branicky, and V. Liberatore, "Co-simulation tools for networked control systems," in *Proceedings of the 11th international workshop on Hybrid Systems: Computation and Control*, HSCC '08, (Berlin, Heidelberg), pp. 16–29, Springer-Verlag, 2008.
[11] A. Hussain, J. Heidemann, and C. Papadopoulos, "A Framework For Classifying Denial of Service Attacks," *Proc. of the Conf. on Applications, Technologies, Architectures, and Protocols for Comp. Comm. - SIGCOMM*, p. 99, 2003.
[12] "SCADA modbus restart denial of service."
[13] S. East, J. Butts, M. Papa, and S. Shenoi, "A taxonomy of attacks on the DNP3 protocol," *IFIP International Federation for Information Processing*, pp. 67–81, 2009.
[14] "The DETERLab Facilities." http://www.deter-project.org.
[15] T. Benzel, "The science of cyber security experimentation: The DETER project," *Annual Computer Security Applications Conference*, December 2011.
[16] A. Hussain, S. Schwab, R. Thomas, S. Fahmy, and J. Mirkovic, "DDOS experiment methodology," *DETER Workshop Proceedings*, 2006.
[17] S. Floyd and E. Kohler, "Internet research needs better models," *SIGCOMM Comput. Commun. Rev.*, vol. 33, pp. 29–34, January 2003.
[18] "DETER Resources." https://trac.deterlab.net/wiki/DeterResources.
[19] "Wireshark Website." http://www.wireshark.org/.
[20] S. Amin, A. A. Cárdenas, and S. Sastry, "Safe and secure networked control systems under denial-of-service attacks," in *HSCC* (R. Majumdar and P. Tabuada, eds.), vol. 5469 of *Lecture Notes in Computer Science*, pp. 31–45, Springer, 2009.
[21] A. Gupta, C. Langbort, and T. Basar, "Optimal control in the presence of an intelligent jammer with limited actions," in *CDC*, pp. 1096–1101, IEEE, 2010.