NCS Security Experimentation using DETER

Alefiya Hussain
USC/Information Sciences Institute
hussain@isi.edu

Saurabh Amin
MIT-CEE
amins@mit.edu

ABSTRACT

Numerous efforts are underway to develop testing and experimentation tools to evaluate the performance of networked control systems (NCS) and supervisory control and data acquisition (SCADA) systems. These tools offer varying levels of fidelity and scale. Yet, researchers lack an experimentation framework for systematic testing and evaluation of NCS reliability and security under a wide range of failure scenarios. In this paper, we propose a modular experimentation framework that integrates the NCS semantics with the DETERLab cyber security experimentation facilities. We develop several attack scenarios with realistic network topology and network traffic configurations to evaluate the impact of denial of service (DoS) attacks on scalar linear systems. We characterize the impact of the attack dynamics on six plants located at various levels in a hierarchical topology. Our results suggest that emulation-based evaluations can provide novel insights about the network-induced security and reliability failures in large scale NCS.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: Reliability,availability, and serviceability

General Terms

Experimentation, Reliability, Security

Keywords

Robust Control, Network Security, Experimentation

1. INTRODUCTION

Networked control systems (NCS), which are increasingly being used for operational management of large-scale physical infrastructures, inherit the vulnerabilities of commercial IT solutions. In recent years, numerous studies have focused on the interconnected physical and cyber-based processes of NCS and next-generation supervisory control and data acquisition (SCADA) systems. The interdependence between random failures caused due to sensor-actuator faults and adversarial failures caused due to malicious software are especially important [20]. Several theoretical analyses build on well known classes of attacker-defender models, and apply tools from robust control and game theory, to derive safety and performance bounds for a wide range of NCS models [16]. 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 techniques for security evaluation by combining real and simulated components. Such experimental research is necessary to complement theoretical performance bounds. It 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 and SCADA systems. There is a diverse body of literature which studies the co-simulation of NCS and SCADA processes using Matlab, Modelica, Ptolemy, and other hybrid system simulation tools with simulated network models using ns2, OMNet++, SSFnet [1, 4, 5, 10, 12, 17, 13, 18]. These approaches are sufficient to study NCS performance under unreliable communication networks with delay, jitter, and packet loss. However, for the purpose of cyber security testing and evaluation for NCS and SCADA systems, emulation-based experiments offer a richer class of scenarios. At present, multiple government-industry initiatives are exploring testbed research and development for NCS and SCADA system applications. The DHS Control Systems Security Program (CSSP) and the DOE-OE Control System Security National SCADA Testbed (NSTB) [17] have offered the red-team and blue-team training exercises for asset owners and vendors across different utility sectors over the past three years. The current and past red and blue configurations involved prototyped tools on a limited basis to see if the 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 available at the DETERLab facilities to study network attacks for NCS systems. The integration of the NCS models into DETERLab has a potential to offer a unique opportunity to use the large scale network testing capabilities to generate realistic network attacks and

HiCoNS'12, April 17–18, 2012, Beijing, China.
Copyright 2012 ACM 978-1-4503-1263-9/12/04 ...$10.00.

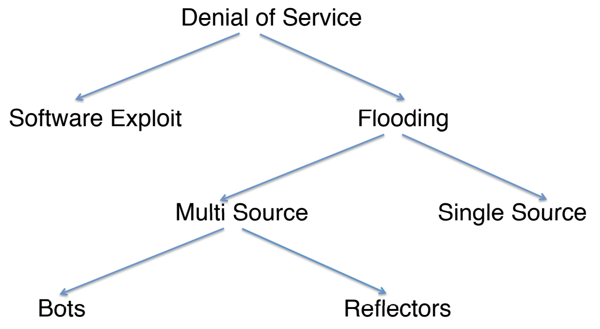

Figure 1: A taxonomy of DoS attacks based on the volume of attack packets and the number of attackers. [14].

validate resilient control algorithms for maintaining the safety and security of NCS. In particular, we integrate the dynamics of multiple NCS with a hierarchically structured network topology, where the individual NCS face different levels of flood-based denial of service (DoS) attacks. We represent the 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 the evolution and closed-loop stability under a range of attack scenarios. In our experiments, the forward network path from the plant to the controller, 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. In this paper we are primarily focused on flood-based DoS attacks which impact the timely delivery of the plant and controller feedback. There are other classes of attacks, such as *deception* attacks, where the integrity of the control data is compromised. They are not discussed in this paper.

In Section 2, we briefly discusses the taxonomy of DoS attacks in networked systems and summarizes the existing capabilities of DETERLab. In Section 3, we introduce an experimentation model of NCS and discuss our approach for testbed-based emulation. We are specifically focused on using real–world attack tools and mechanisms along with representative models of topology and cross traffic to systematically evaluate the security of linear dynamical systems. In Section 4, we evaluate the impact of the attack on the security and stability of a plant. Specifically, how the attack characteristics, such as, attacker–plant location, start time, and packet size, impact the closed feedback loop between the plant and the controller. Our results suggest that emulation-based evaluations provide novel insights into network-induced security and reliability failures in large scale NCS.

2. DOS ATTACKS

In this section we summarize the taxonomy of DoS attacks on the Internet and discuss how the taxonomy is applicable to NCS and SCADA systems that directly or indirectly use network connectivity for their operation. This taxonomy does not include DoS attacks to NCS communicating over the wireless networks. We then discuss how the DETERLab facilities and tools can be used for evaluating NCS safety and security against DoS attacks.

2.1 Taxonomy

In a DoS attack on the Internet, a malicious user exploits the network connectivity to cripple the services offered by a victim server, often by simply exhausting the resources at the victim. Typically, these resources include network bandwidth, computational power, or 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. A multi-source attack is also 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.

To launch a DDoS attack, a malicious user compromises Internet hosts and installs attack tools on the host also known as a zombies or bots. The bots is now available to attack any victim on command. With full control of the bots, 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 attacks target specific software bugs in the 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 force a programmable logic controller (PLC) device or Modbus TCP servers to repeatedly power cycle by sending a TCP request containing the 08 Diagnostics function code with sub function 01 [19]. Additionally, East et. al [8], 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 the master that the remote 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) bot directed floods, and (b) reflector attacks.

In *bot directed flooding attacks*, a malicious user installs attack tools on the host machine to generate a flood of illegal packets. Examples include, attacks that send a flood of TCP requests to a sensor node resulting in power exhaustion at the node [10] and attacks that create a flood of DNP3 messages between the master and the remote devices [8]. Several canned attack tools are available on the Internet, such as Trinoo, Tribal Flood Network, and SCADA server/client attack tools, that can generate flooding attacks using a combination of protocols.

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 network request. For example, a web servers or ftp servers that respond to TCP SYN requests with a TCP SYN-ACK packets, and a host 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 that represent improperly secured hosts, reflectors are often hosts intentionally providing Internet services, and hence reflector attacks may be more difficult to block.

2.2 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 [3, 6]. This facility has been operational since 2003 and is operated by the USC Information Sciences In-

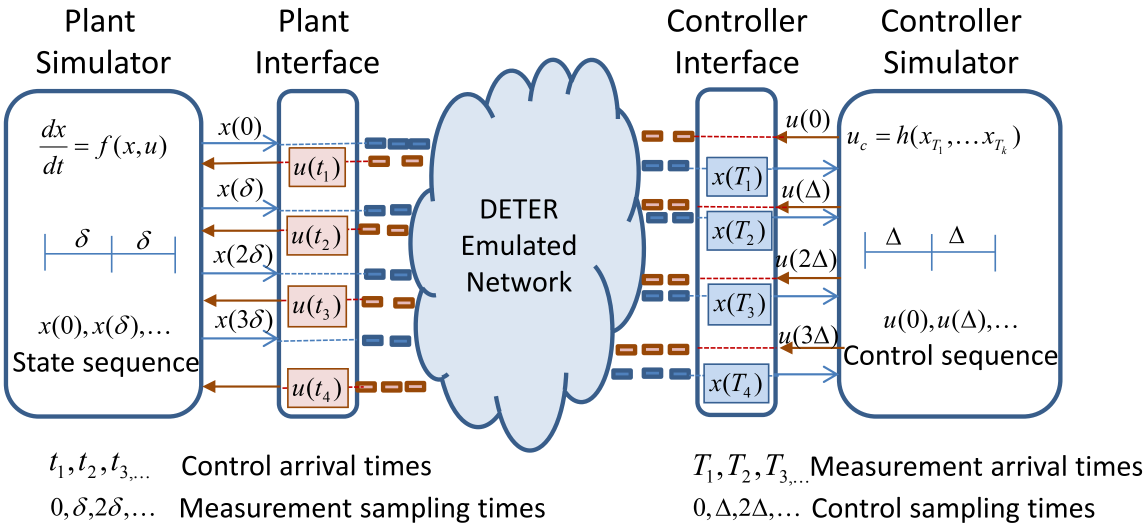

Figure 2: NCS emulation on the DETER Testbed

stitute, UC Berkeley, and Sparta Inc. As of December 2011, the DETER testbed has supported more than 2000 experimenters and students experimenting with a diverse set of cyber security technologies. The main thrusts of research on the testbed include DoS attacks, worm propagation and analysis, botnets, and anomaly detection in networks.

Using DETERLab for evaluation of NCS systems allows the experimenter to replicate the interactions between the NCS components, and the attackers with high-fidelity and accuracy. The NCS components, such as the plants and the controllers, can be implemented as simulation, emulation, or real components with the interface discussed in Section 3. The attack traffic can be generated using either real–world attack tools mentioned in the previous section or 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 suggested operating systems and end host configuration. The DETERLab facilities 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. These tools together provide a unique balance between experiment control and realism.

We note that there are several experimentation environments available or currently under development for NCS and SCADA systems. Each offers a different levels of fidelity and scale [4, 17, 5]. We believe that the DETERLab tools and facilities complement these efforts. In particular, it allows the experimenter to closely replicate the real–world end host and cyber attack models. This enables systematic and consistent evaluation of physical control systems in such environments.

3. EXPERIMENTATION FRAMEWORK

In this section, we discuss the framework for integrating NCS semantics with the DETER testbed to systematically explore the impact of network attacks on evolution and stability of such systems. Our approach is to combine NCS system tools and simulation with DETERLab tools and methodologies for networking and cyber security testing and evaluation. The experimentation framework is shown in Figure 2 and has three main components: the physical system dynamics, the physical-to-cyber network interface, and the cyber network dynamics.

In our framework, the NCS plant is remotely connected to a controller over a shared network. The plant–controller communication is hence subjected to network and shared medium effects, such as, delays, jitter, and packet losses. In addition, in presence of cyber security incidents, the communication can also be subjected to malicious losses or corruption.

The physical system dynamics of a NCS can be represented as a simulation, an emulation, or by real control system such as the interconnected four-tank water system. In this paper we implement a simulation based scalar linear system. The physical-to-cyber network interface translates between the physical and the event-based network dynamics. The cyber network dynamics can be represented as a simulation, an emulation, or an operational network such as the Internet. In this paper, we emulate the network and cyber security dynamics on the DETERLab facilities. In this paper we specifically focus on the study of DoS attacks on control systems. We now discuss each of these components in more detail.

Physical System Dynamics: 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. We build on NCS co-simulation technique developed by Branicky et. al [1]. 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 $R(t)$ the reference trajectory signal. A represents the system matrix and K represents the controller gain matrix, In this paper, we restrict our attention to scalar dynamics.

Network Interface: The network interfaces are located both at the plant and the controller. This enables the integration of NCS dynamics with the event-based communication semantics of the DE-

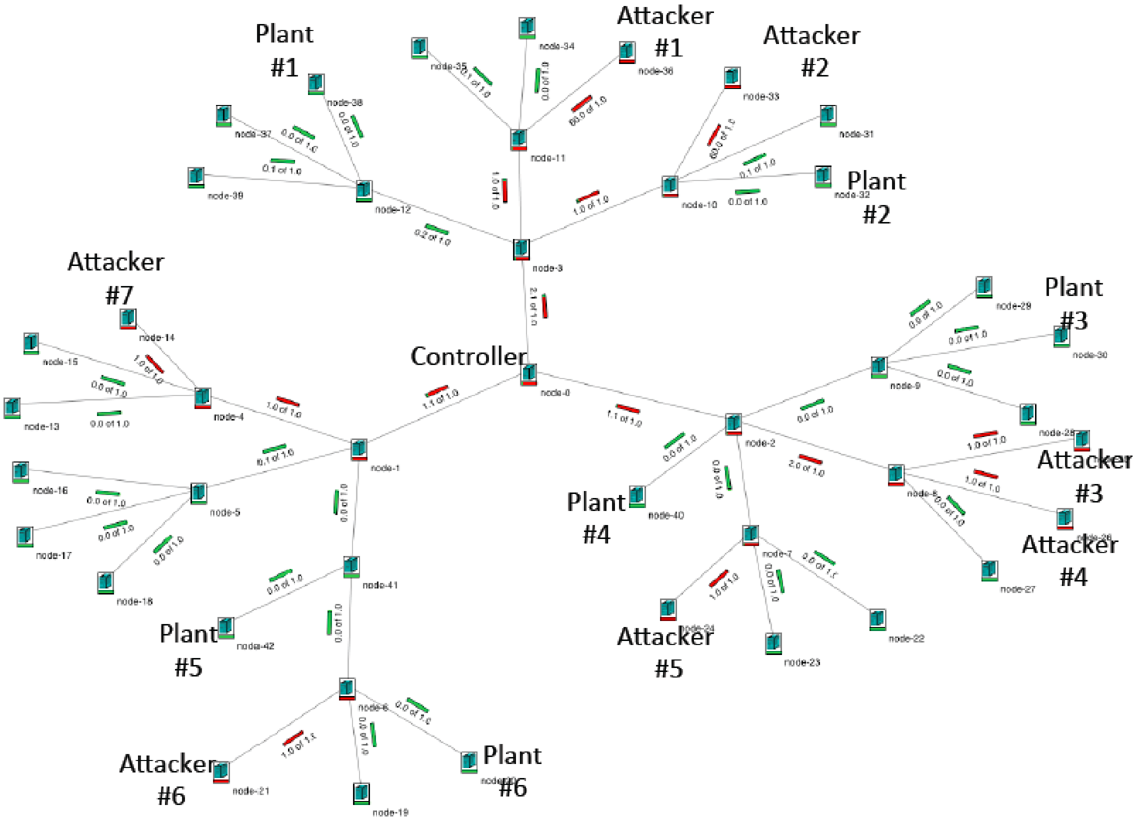

Figure 3: A hierarchically structured network topology

TER 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. This is presented as a 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 δ.

Upon receipt of the plant system state, the controller interface passes the data to the controller system. The controller calculates a control input and sends it back to the controller interface at periodic intervals Δ. The controller interface then forwards the control input on to the network. Upon receipt of the control input, the plant interface immediately forwards the control input to the plant system. The plant incorporates the control input and updates the current plant state. The plant then computes the next projected plant state.

In order to account for asynchronous and out-of-order system state arrivals, at both the plant and controller interfaces, the system has the capability to roll-back and roll-forward their respective updates.

Cyber Network Dynamics The middle section, shown as a cloud, represents the DETER testbed experimentation network that emulates topology and traffic dynamics between the plant and the controller. Such a communication network is typically modeled with two primary layers; (a) the physical topology and routing structure of the network components, and (b) the network traffic layer between the network components [15].

Selecting representative topologies for the communication network has been a subject of significant research over the last several years. It is challenging since the Internet structure constantly evolves and deployed NCS or SCADA systems rarely make their underlying network topologies publicly available due to security reasons [9, 4]. Further, the network routing structure is also impacted by the link-level communication technologies, such as wireless, satellite, or wired networks. For example, wireless mobile networks have a dynamic topological and routing structure that evolves with the movement of the nodes while wired networks have a relatively 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 [7].

The second layer, the traffic in the experimentation framework is determined by the various servers, clients, and attackers in the network. To accurately model the wide-area networks and the Internet, cyber security experiments typically model three different types of network traffic; (i) *background traffic*, for example, web server and web client traffic which is congestion reactive, the (ii) *foreground traffic* that is under study, for example, control traffic in a NCS system, and (iii) *malicious or selfish traffic* such as, attack traffic in a DoS attack, or constant-rate traffic generated by selfish nodes. This type of traffic is not congestion reactive. The three types of traffic interleave to create a complex set of dynamics that can be captured with high fidelity in an emulation-based environment such as a testbed. They are discussed in detail in Section 4. DETER provides a diverse set of traffic generators, including Harpoon,TCP replay, Apache wget clients for background and foreground traffic, and real and emulated DoS attack traffic and worm traffic generators [7].

Modularizing the experimentation framework enables us to rapidly evolve the models to systematically explore the structural and functional aspects of the system. This allows addressing existing security threats, identifying new threats, and meeting the challenges of fidelity, scale, and complexity. 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 system in an emulated testbed environment.

4. EVALUATION

Using the experimentation framework presented in the previous section, we now systematically evaluate the impact of a DoS 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.

4.1 Emulation Parameters

Our goal is to employ the experimentation framework to create a rich and complex evaluation scenario that will allow the assessment of a multi-source flooding DoS attack on the networked control system. We model complexity in both topology and traffic. We employ network topologies appropriate for the analyses of DoS attacks. We interleave background web traffic, foreground control traffic, and malicious DoS attack traffic, as expected on a real network. We use real–world attack tools to capture the complexity of the attack dynamics.

While there are several ways to model the underlying topology, as discussed in Section 3, 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 [1], 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, the first subnet has both plants located at the leaf nodes, the 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.

We deploy attackers at seven leaf nodes in the network as shown in the Figure 3. Each attacker generates a DoS attack, with a real–world tool called punk [14], that sends a maximum rate stream of TCP segments, 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. In addition, all the leaf nodes also generate webtraffic that traverses the network [7].

Consequently, plants are located at various levels in the topology. It allows systematic study of the impact of the attack on the control signals at different levels of aggregation of the attack and control traffic.

4.2 Effect of Location

We first study the impact of the DoS attacks on plants that are located at different levels in the topology. In the Figure 4, the y-axis plots the output state from the plants #3, #4, and #5, at any point in time. Each plant is located at a different tree-depth level, specifically; (a) Plant #4 located one tree-depth level from the controller, (b) Plant #5 located two tree-depth levels from the controller, (c) Plant #3 located at a leaf node.

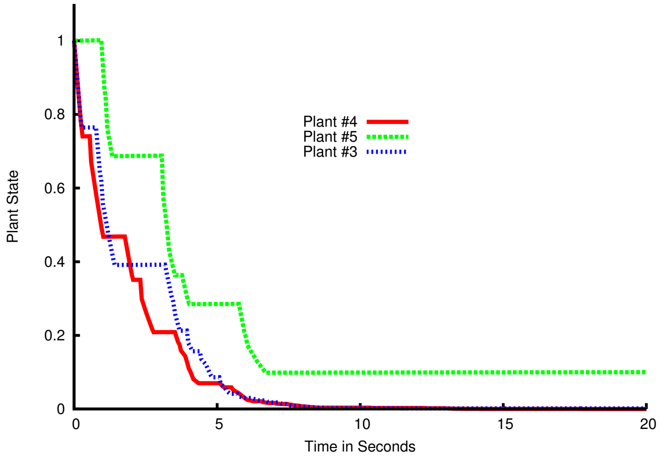

Figure 4: Impact of attack at various locations in the topology

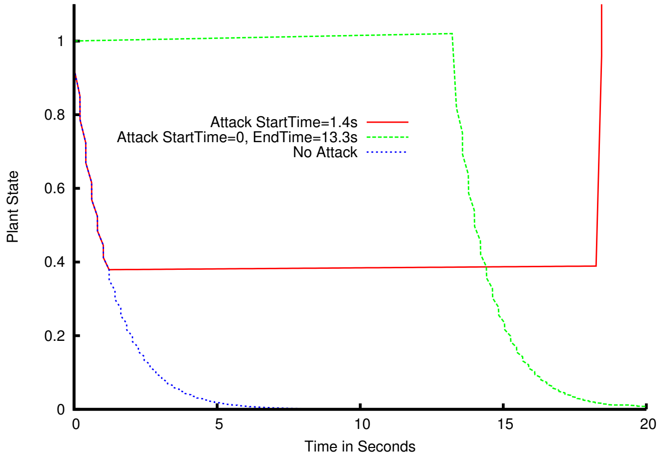

Figure 5: Effect of Start time of the attack on the plant

The attackers are configured to generate 40 byte attack packets using real–world attack tools. The aggregation of the attack traffic in the topology impacts the convergence of the plant state. As observed from the graph, each plant is subjected to a different plant convergence dynamic during the attack. When we varied the size of the attack packet, from the minimum size of 40 bytes to a large packet of 1040 bytes, we observed plant #5 and plant #3 converged very slowly and sometimes failed to converge. We observed plant #4, that is located two hops from the controller, was significantly more stable than plants located at deeper levels in the topology.

These results indicate that parsimonious analytical models may not capture the complex interaction between the various topology and traffic components.

4.3 Effect of Start time

We study the impact of the attack start time on the plant stability. In Figure 5, y-axis plots the output state from plant #3 at any point in time. Under the no attack condition, all the plants stabilize and converge. Since NCS rely on real-time feedback, both packet delays and packet losses affect the NCS stability. The route between plant #3 and the controller, is impacted by three attackers send-

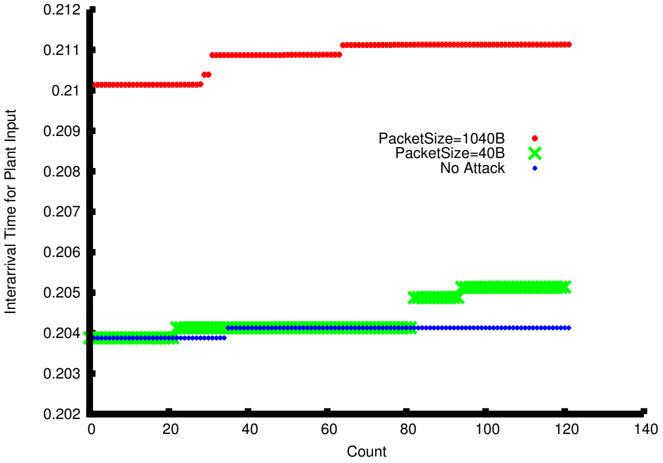

Figure 6: Effect of an attack on the Inter-arrival time of the plant state

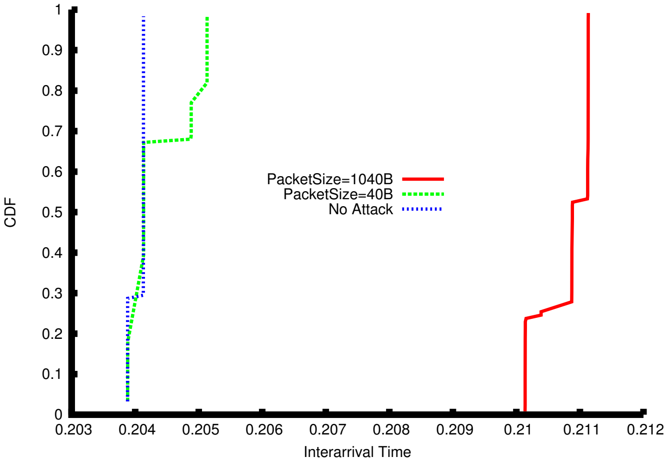

Figure 7: CDF of the Inter-arrival time

ing packets at the maximum rate with a packet size of 40 bytes. In this section we discuss the stability of plant #3 as it is farthest away from the controller and is exposed to the maximum number of attackers.

We investigate two scenarios; (a) when the attack starts during the plant operation and does not stop for a long period of time, and (b) when the attack starts before the plant operation and stops after a small period of time. In the first scenario, the plant starts to converge before the attack starts, but once the attack starts the plant state becomes highly unstable since several feedback messages are lost. This scenario is shown in the Figure 5 with the attack starting at 1.4 seconds. In the second scenario, 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. This scenario is shown in the Figure 5 with the attack ending at 13.3 seconds.

These results indicate that it is important to account for attacker start and stop dynamics when modeling cyber security scenarios.

4.4 Interarrival time

We study the impact of the attack packets on the interarrival time of the plant and controller output. The size of the attack packet can have a significant impact on time-sensitive NCS. The packet size determines the transmission time, that is, large attack packets cause longer delays as compared to small attack packets. In the presence of an attack, when there are several such packets flooding the network and interleaving with the control packets, they can cause instability in the convergence of the NCS.

For the analysis in this section, we measured network packets at the plant and controller using tcpdump [21], and then calculated the time difference between consecutive plant state output packets. In Figure 6, the y-axis plots the interarrival time of plant #4 against with the packet counts. Figure 7 shows the interarrival time as a cumulative distribution function.

We first discuss the no attack condition. Plant #4 is two hops away from the controller and the transmission time of the plant state packet of size 52 bytes, from the plant to the controller is 0.200832s. Additionally, there is a 3ms processing delay at the plant in the current implementation. The interarrival time is bimodal with a difference of $300\mu s$ between the two modes which we believe is due to the interleaving of cross traffic.

Next we investigate two attack scenarios; (a) when an attacker, generating small packets of 40 byte,s starts before the plant operation, (b) when an attacker, generating large packets of 1040 bytes, starts before the plant operation. Both scenarios have a multi-modal interarrival time distribution due to the interleaving of attack and control packets.

The results in this section indicate that depending on how the attack packets, plant state packets, and background traffic interleave, there is a significantly impact on the interarrival times between the plant output. .

5. CONCLUSION

In this paper discussed an experimentation framework for the evaluation of NCS on the DETERLab facilities. Our experimentation framework has three main components: physical dynamics, a physical-to-cyber interface, and a cyber network model. The physical system dynamics are implemented in simulation and are modeled as a scalar linear system. The physical-to-cyber interface is designed for sending and receiving data across the emulated network and allows the implementation of event-based semantics. The cyber network is modeled on the DETERLab experimentation facilities with realistic topology and traffic parameters. This modular approach provides an environment in which experiments can be rapidly configured, and can evolve 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 cyber security experimentation tools and methodologies to evaluate multiple scalar linear systems under a distributed denial of service attack. While the plant dynamics are simple, the key contributions are in the simultaneous experimentation of multiple plants under a wide range of network conditions. Our results indicate that experimentation evaluation provides unique insights into the plant dynamics that are not apparent in analytical or simulation studies. We plan to expand our analyses and compare these experimental observations with theoretical performance bounds, for example, block attacks and strategic jamming attacks [2, 11] as future work.

Acknowledgements

The authors are grateful to Anthony Joseph (Berkeley), Gabor Karsai (Vanderbilt), Blaine Nelson (Berkeley), Suzanna Schmeelk (Rutgers), Galina Schwartz (Berkeley), Terry Benzel (USC/ISI), Bob

Braden (USC/ISI), 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. This material is based on work partially supported by the United States Department of Homeland Security and Space and Naval Warfare Systems Center under the contract number N66001-10-C-2018 and the MIT faculty start-up grant. All findings and conclusions expressed in this material are those of the authors and do not reflect the views on the funding agencies.

6. REFERENCES

[1] 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, pages 16–29, Berlin, Heidelberg, 2008. Springer-Verlag.

[2] S. Amin, A. A. Cárdenas, and S. Sastry. Safe and secure networked control systems under denial-of-service attacks. In R. Majumdar and P. Tabuada, editors, *HSCC*, volume 5469 of *Lecture Notes in Computer Science*, pages 31–45. Springer, 2009.

[3] T. Benzel. The science of cyber security experimentation: The DETER project. *Annual Computer Security Applications Conference*, December 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] A. Davis. Developing SCADA simulations with c2windtunnel. Master's thesis, Vanderbilt University, Nashville, Tennessee, May 2011.

[6] The DETERLab Facilities. http://www.deter-project.org.

[7] DETER Resources. https://trac.deterlab.net/wiki/DeterResources.

[8] S. East, J. Butts, M. Papa, and S. Shenoi. A taxonomy of attacks on the DNP3 protocol. *IFIP International Federation for Information Processing*, pages 67–81, 2009.

[9] S. Floyd and E. Kohler. Internet research needs better models. *SIGCOMM Comput. Commun. Rev.*, 33:29–34, January 2003.

[10] A. Giani, G. Karsai, T. Roosta, A. Shah, B. Sinopoli, and J. Wiley. A testbed for secure and robust scada systems. *SIGBED Rev.*, 5:4:1–4:4, July 2008.

[11] A. Gupta, C. Langbort, and T. Basar. Optimal control in the presence of an intelligent jammer with limited actions. In *CDC*, pages 1096–1101. IEEE, 2010.

[12] 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, pages 21:1–21:4, New York, NY, USA, 2010. ACM.

[13] 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*, page 16, 03 2011.

[14] 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*, page 99, 2003.

[15] A. Hussain, S. Schwab, R. Thomas, S. Fahmy, and J. Mirkovic. DDOS experiment methodology. *DETER Workshop Proceedings*, 2006.

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

[17] Idaho national lab SCADA testbed program. http://www.inl.gov/scada.

[18] 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*, 82:43–59, January 2006.

[19] Xforce Attack Repository. SCADA modbus restart denial of service. http://xforce.iss.net/xforce/xfdb/20739.

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

[21] Wireshark Website. http://www.wireshark.org/.