Journal of Mechanical Engineering Research and Developments
ISSN: 1024-1752
Website: http://www.jmerd.org
Vol. 39, No. 1, 2016, pp. 75-82

The Development of Cyber Security Testbed for Networked Mechatronics Monitoring and Control System Based on DETER

H. P. Huang*, S. D. Xiao, & J. Duan

Electromechanical and Control Department, Mechanical Engineering College
Southwest Jiaotong University
Chengdu, Sichuan, 610031, P.R.China, *Email: hhping2@163.com

ABSTRACT: A kind of cyber security testbed for networked mechatronics monitoring and control system based on DETERlab infrastructure is designed and implemented in this paper. It is a novel platform on which researchers can do the system cyber security evaluation and countermeasures investigation. A typical networked mechatronics system architecture is emulated according to the submitted ns script file. DETERlab servers use the hardware resources to setup the test nodes such as PLC, Master IPC and etc. A kind of software tool is developed to make the test nodes can communicate with each other according to the Modbus TCP protocol. Some other software are also applied in this testbed for generating attack and monitoring the network communication status. This testbed is applied to do Dos attack test successfully. The results show that the testbed proposed in this paper has high practicability, reality and scalability. It can do great help for the cybersecurity research work about the networked mechatronics system.

KEY WORDS: Networked Mechatronics Monitoring and Control System; Cyber Security; Testbed.

INTRODUCTION

In general, the industrial control system (ICS) consists of several types of systems, such as supervisory control and data acquisition (SCADA) systems, distributed control systems (DCS), and Programmable Logic Controllers (PLC). ICS, especially SCADA system, has been playing more and more important roles in critical infrastructure, including electrical power grids, railway transportation, water treatment, chemical processing, and petroleum and natural gas [1, 2]. Consequently, there is no doubt that the existence of vulnerability of these systems will do vital harm to the cyber security of the nation.

Mechatronics monitoring and control system has been widely used in modern manufacturing industry, and has played a key role in the field of automated manufacturing. With the rapid development of network, communication and computer technology, the networked mechatronics monitoring and control system becomes to adopt different kinds of communication ways, such as Ethernet, WAN (wireless area network), and Internet, these communication channels are no longer isolated and are becoming more and more interactive with other information systems. Therefore, the control commands and state data in a mechatronics monitoring and control system are facing great threats to be eavesdropped or modified. This may result in a series of quite serious attack on the mechatronics systems, together with significant financial and property losses [3-5]. As a result, how to ensure the cyber security of the networked mechatronics system has been a critical research topic.

Nowadays, many researchers are devoted to the research on cyber security of industrial control SCADA system. In order to better understand how to protect the cyber security of the networked mechatronics system, we must take an assessment on vulnerability of the system and then find a proper security defense mechanism to protect our system from being attacked. At the same time, in order to undertake these tasks, building a cyber security testbed for the networked mechatronics system is the primary precondition. The testbed has many benefits. For example, we can take on some attack experiments the on the platform, and analyze the impact on the system facing different attacks, followed by trying different types of defense solutions. Hence, it is the fundamental work for the research of cyber security of networked mechatronics monitoring and control system.

In general, the testbed can be divided into some different types:

- A testbed with all real equipment;
- A testbed realized by software simulation;

DOI: 10.7508/jmerd.2016.01.011

- A testbed combined with real equipment and software simulation.

A testbed with all real equipment is able to provide a completely realistic situations when facing an attack. The National SCADA Testbed (NSTB), established by Idaho National Laboratory, can provide a completely realistic test environment for evaluating the vulnerability of industrial control system, together with its software and hardware [6]. Mississippi State University's laboratory-scale process control system Cybersecurity testbed uses commercial software and hardware devices, which is only capable of teaching in universities or institutions [7]. In spite of their advantages, there still exist some disadvantages. Among these disadvantages, the biggest one is that it will require a large budget, and will be difficult to maintain.

A testbed realized by software simulation is simple to establish, and nearly costless to maintain. Chabukswar *et al* developed a reliable SCADA experiment platform[8]. In the platform, they have realized the simulation of communication network by MONeT++ and established a simulation model of controller and controlled physical devices by Matlab Simulink. David Bergma developed a simulated testbed for the simulation of Electric Grid SCADA system using RINSE network simulator in his Master's thesis [9]. However, it is unable to completely present the situations faced by real system while being attacked.

In order to avoid these drawbacks and meet the needs of appropriate budget and easy maintenance, here comes another kind of testbed that is a hybrid between totally real equipment and software simulation. And the most promising kind is Emulab, especially Deterlab, a program based on Emulab [10]. The program will collect and provide different kind of hardware resource, like PLCs, master stations and exchangers. Users have access to it by submitting NS files used to define the character of the corresponding virtual systems.

This paper describes how to develop a classical networked mechatronics monitoring and control system testbed based on DETERlab. Besides, we also do some DoS attack based on the testbed to verify the reliability of it. The organization of the rest of this paper is as followed. In section II, we give an overview of the architecture of DETER. In section III, we briefly discuss the components of a typical networked mechatronics monitoring and control system and its protocols. In section IV, we present the process of designing and implementing an cyber security testbed for networked mechatronics monitoring and control system based on DETER, and the development of Modbus TCP protocol scripts and the configuration of the nodes. In section V, we introduce a series of DoS experiments on the testbed, to validate the reliability of the testbed. In section VI, we give a conclusion and our future research directions.

THE OVERVIEW OF THE ARCHITECTURE OF DETER

The DETER testbedis hosted by the University of Southern California's Information Sciences Institute and at University of California at Berkeley [11]. It is commonly used concerning the cyber security research.

As an open testbed, it provides us with plenty of useful software and reconfigurable hardware resources and a rich set of convenient experimental tools, such as SEER Workbench, Semantic Analysis Framework [12,13]. Based on these resources, tools will be deployed to create network topology easily, generate attack traffic, monitor network traffic and system state, and are able to provide data acquisition, display ,and process, which will help to go on high-fidelity, large scale network and information cyber security experiments [14, 15]. The DETER testbed can not only stimulate the intercommunication between real networked mechatronics monitoring and control system components (e.g. sensors and PLC, etc.), but also can repeat the behavior of attackers vividly and accurately. Apparently, DETER testbed is suitable to build a test platform for cyber security evaluation for networked mechatronics monitoring and control system.

When building the testbed for cyber security evaluation for networked mechatronics system, the primary step is to build a certain topology, which calls for enough hardware, such as the Industrial Personal Computer (IPC), PLC and etc. DETER platform provides powerful hardware condition, which will help users to cut their budget sufficiently. At the same time, DETER platform supports NS script files to customize uses' own system environment. According to the NS script files, DETER platform will assign necessary hardware resources from its pool of available resource to build corresponding topologies. This process can be illustrated in Fig. (1).

THE OVERVIEW OF THE ARCHITECTURE OF A TYPICAL NETWORKED MECHATRONICS MONITORING AND CONTROL SYSTEM AND ITS PROTOCOL

In general, the networked mechatronics monitoring and control systems consist of the following component:

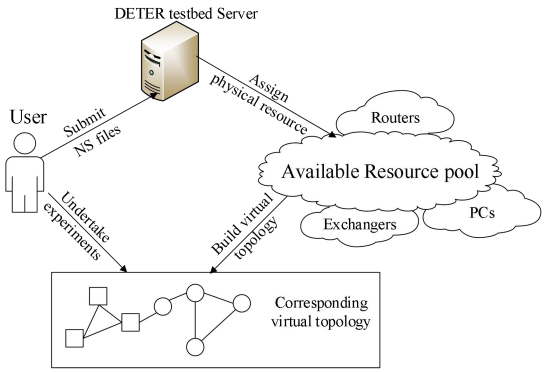

Figure 1. The process of creating a virtual topology under DETER platform.

(1) Field data interface devices, such as, Remote Terminal Unit (RTU), Process Automation Controller (PAC), Intelligent Electronic Device (IED), or Programmable Logic Controller (PLC). They are applied to communicate with sensors and controllers in the field;

(2) Central host servers, namely, Master Terminal Unit (MTU). They are applied to process data from field data interface devices;

(3) Human Machine Interface (HMI) system. It is used to provide the operators an overview of the system, and will alert the operators to the emergency situations;

(4) Communications system. It will link the data from field data interface devices and central host servers. The medium of system may be radio, GSM, GPRS, Wi-Fi, WAN, telephone, cable, microwave, satellite, etc., or any combination of these.

There exist many kinds of communication protocols in use, such as Modbus, ICCP, Distributed DNP3, and OPC [16]. Among these communication protocols, as an open-source and freely distributed protocol, Modbus protocol is widely used in the networked mechatronics system. Modbus protocol can realize efficient communication owing to its request/response methodology, and adapts function code to offers specific services. With the development of Internet, Modbus protocol has used reserved TCP port 502 and been adapted to function over TCP/IP, i.e. Modbus TCP protocol. To some extent, the emergency and prosperity of Modbus TCP protocol correspond to the prosperity of Internet. On the other hand, Modbus TCP protocol will inevitably be faced with plenty of threats inherited from Modbus protocol and TCP protocol.

THE DESIGN AND DEVELOPMENT OF CYBER SECURITY TESTBED FOR NETWORKED MECHATRONICS SYSTEM BASED ON DETER

When designing the cyber security testbed for networked mechatronics system based on DETER, we can divide our primary tasks into two parts. One is to stimulate the architecture of a networked mechatronics system built on vast hardware resources from DETER, the other is to assign relevant software resources to every node in the simulated system.

The realization of network structure of a typical networked mechatronics monitoring and control system based on DETER

In order to implement the architecture of the networked mechatronics system built on vast hardware resources provided by DETER, a topology must be constructed.

As we mentioned above, DETER will assign relevant hardware resources pool to simulate Servers, Clients, Routers, Switches, etc. according to the submitted NS files. Then a corresponding topology will be provided for following research. We needs to take the following steps when building the network architecture of a typical networked mechatronics system, which will be presented in Fig. (2)

(1) Describe the network topologies of the system in NS script grammar;

(2) Finish the NS files and submitted to DETER servers;

(3) DETER servers will allocate its resources (i.e. PLCs, MTUs, RTUs, etc.) and simulate network traffic and bandwidth according to the NS files we submitted.

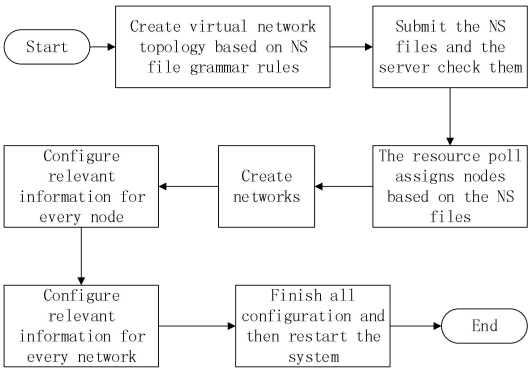

Figure 2. The process of realizing a virtual testbed with NS files based on DETER.

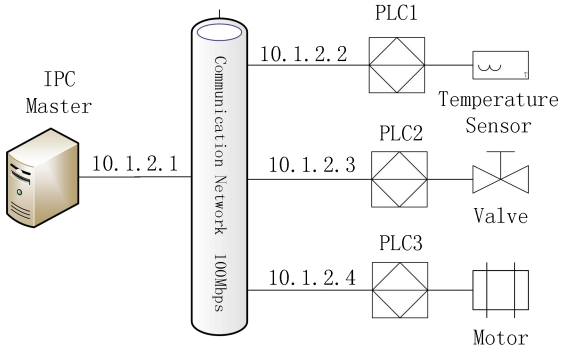

Figure 3. A typical networked mechatronics monitoring and control system.

As shown in Fig. (3), there is a typical networked mechatronics monitoring and control system, which consists of a temperature sensor, a valve, a motor, three PLCs and IPC. To create the virtual topology of this system, a piece of corresponding NS files scripts is shown as follows.

```
# This is a scada testbed ns script. Comments start with #.
set ns [new Simulator]
source tb_compat.tcl
set nodemaster [$ns node]
tb-set-node-os $nodemaster CentOS6-64-STD
set nodeplc1 [$ns node]
tb-set-node-os $nodeplc1 CentOS6-64-STD
set nodeplc2 [$ns node]
tb-set-node-os $nodeplc2 CentOS6-64-STD
set nodeplc3 [$ns node]
tb-set-node-os $nodeplc3 CentOS6-64-STD
set lan0 [$ns make-lan “$nodemaster $nodeplc1 $nodeplc2 $nodeplc3” 100Mb 0ms]
$ns rtproto Static
# Go!
$ns run
```

The development and configuration of software resource

After finishing constructing the topology, we use the following steps to realize the intercommunication between nodes, especially between sever node and client nodes. As we mentioned, these nodes will be communicated while obeying Modbus TCP protocol at the same time in the experiments.

It is a pity that DETER resources haven't support any Modbus protocols, however, there are some Python libraries for Modbus protocols. After referring to these Python libraries for Modbus, especially for Modbus TCP, the authors and our team have succeeded in developing corresponding script named VirPLC and VirIPC based on Python.

Furthermore, we already succeed in deploying these Python scripts for Modbus TCP to corresponding nodes. VirPLC and VirIPC are all based on Modbus TCP protocol scripts. Among with these scripts, class MBTCPSlaverMsgs and class MBTCPMasterMsgs are used to package and unpackage Modbus TCP message of VirPLC and VirIPC respectively. In general, the two Python scripts should implement the following common function from table I.

Table 1. Common functions of Modbus TCP protocol**.**

Function Code	Function Content
01	Read coil status
02	Read input status
03	Read holding registers
04	Read input registers
05	Force single coil
06	Preset single register
15	Force multiple coils
16	Preset multiple registers

VirPLC is used on client node. In order to implement these function, a Python library VirPLClib should be edited to provide some common function to VirPLC, such as offering help contents, comfirming command options and reporting relevant information. VirPLC script will achieve the goals of sending command to the server, receiving corresponding commands from server, decoding these commands and executing these commands.

VirIPC is used on server node. We define a class named GetOption to get commands from clients. As for listening for any new connection requests, a class named AsyncServer is defined. When a new connection has been established, a class named SocketHandler will work. After a connection has been canceled, all parameters should be initialized and a new monitor will be started again.

We configure the server node and the client0 node for example. To begin with, we should submit the Python scripts to the DETER server. Then, we should login the server node and the client0 node, and run corresponding scripts commands. To validate the configuration, we may run the commands with some parameters on the client0 node, one of the commands is as follows:

Sudo python /users/ericdj/VirPLC/VirPLC.py -h 5.6.7.8 -p 502 -f 3 -a 6789 -q 2 -t 1 -r 5 -y 60

As for the command above, -h means the server node IP address, -p means the server node's port number which default to 502 if the protocol is Modbus TCP, -f means the Modbus function number, -a 6789 means the address of Modbus server's memory is 6789, -q 2 means the quantity of addresses is 2, -t 1means the time-out of receiving messages is 1 second, -r 5 means the repeated times is 5, and –y 60 means the delay time between every two repeats is 60 milliseconds.

A DOS ATTACK EXPERIMENT ON THE TESTBED

After realizing a typical cyber security testbed for networked mechatronics system, we implement a Dos attack experiment on it to demonstrate its availability and validity. As we know, one of the widely used protocols in the networked mechatronics monitoring and control system is Modbus TCP protocol. In fact, there exist a lot of vulnerabilities in Modbus protocol that will be exploited to launch some attacks at the system [17,18]. In order to understand the effect of Dos attack applied to the networked mechatronics system, we take the system shown in figure (3) as an example to analyze the experiment results. In the experiment, a PLC is connected to a temperature sensor and the Master monitors the temperature by querying the slave .When the temperature reaches a certain level, the Master sends a command to open a safety valve and start the motor to run. In the attack scenario, the functionalities of the system will be affected. For instance, the operators would get wrong and delayed information about the temperature. Based on such wrong information the operator is likely to take wrong actions that could adversely affect the operation of the networked mechatronics monitoring and control system.

Experiment setup

By launching a DoS or DDoS (Distributed Denial of service), attackers aim at forcing the victim servers beyond their daily loads and scrambling for the other resource like band-width, which will greatly influence or even halt normal services undoubtedly. Before launching a DoS attack, attacker must acquire a knowledge of intercommunication and traffic architecture of the targeted system, finish the attack configuration, and then start the attack.

In our experiment, a SYN flooding attack has been launched on the testbed. SYN flooding will be conducted when an attacker consistently sends a large number of TCP SYN packets to a victim master server. During the TCP connections establishing, the master IPC and PLC as a client will realized 3-way handshake. Since TCP SYN packets are part of normal TCP traffic, it is impossible to tell whether the package is legal or not. Therefore, the master server should keep the huge number of the half-open state for each connection, which will exhaust the resource of the victim servers and lead to a denial of service if the attack rate is high enough. At the same time, source IP addresses can be easily faked, tracing the attacking IP is likely to be fruitless.

In order to implement the generation of DoS attack, researchers have to finish a lot of repeated but easy settings. Apparently, researchers should be trained to be sophisticated. However, not all users have mastered sufficient system knowledge to know how to most sufficient use the testbed for their research.

Given these situations, an integrated experiment management and control environment named SEER (The Security Experiment Environment) has been developed. SEER contains a java-developed terminal GUI and a rich set of versatile but easy-to-use integrated tools and agents for helping an experimenter setup, script and perform experiments in the DETER environment. It will greatly foster the process of our research, and enhance our efficiency in our research. SEER have integrated a wide range of existing tools, involving traffic generation modules (like Replay, Harpoon, web, FTP, SSH, IRC, VOIP, DNS and ping functions), attack modules (including flooder and Botnet functions), network configuration modules and data collection and presentation modules (mainly referring to TCPdump).

The analysis and discussion on the results

In this paper, a series of SYN flooding attack experiments has been performed. It will present some experiment results to demonstrate the availability, validity the reliability of the testbed. In the experiments, we simulate a situation that attack traffic suddenly rushes into the victim master IPC during the communication between the master and the PLC/RTU.

As the attack rate raises, it will be harder for PLC/RTU to maintain a connection with victim master IPC, and the interval between every connection will be longer. There is no doubt that the connection will be interrupted while the attack rate increases to a certain degree. Hence, the inter-arrival time between every connection can be a parameter to evaluate the connection situations between PLC/RTU and Master IPC.

In these experiments, we mainly focus on the inter-arrival time between every connection while victim Master IPC is under different attack rate. Several software tools have been applied in experiment. SEER has been used to configure the attack rates, TCP dump has been employed to record every connection and save them on the files server, and the furthermore analysis will be undertaken by Wireshark after downloading these records from the files server.

We find that the attack traffic is less stable with the increasing of attack rates under current situations. So the attack rates will be set as no attack traffic, 500 packages per second, 1000 packages per second, 5000 packages per second, and 10000 packages per second.

At the same time, the more the attack times is, the bigger the attack data file is, and the more complicated the analyses will be. Taken all into consideration, attack times will be set as 100 times. Obviously, the inter-arrival time between each connection is very short, we had better count 10 times attack interval as a unit for the sake of date accuracy. All results have been shown in Fig. (5).

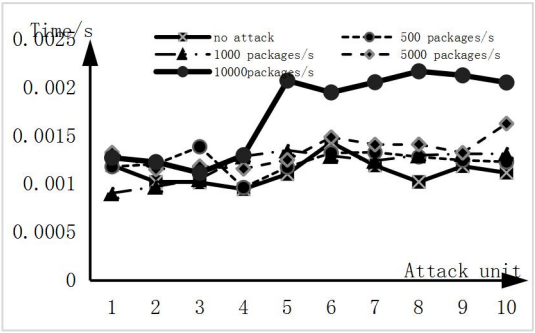

Figure 5. The effect of Dos attack on a networked mechatronics monitoring and controlling system at various attack rate.

In Fig. (5), the x-axis plots the attack unit number and the y-axis plots the inter-arrival time between each unit connections. For any polygonal line, a significant increase compared with the tendency before will stand for the arrival of massive SYN packages. The increase level reflects the impact brought by SYN flooding attack.

After observing in the graph, we can clearly draw the following conclusion:

1) To begin with, the testbed has managed to realize the connection between the master IPC and PLC/RTU;

2) For one attack process, the tendency of the polygonal line starts with a lower balance, and will meet a sharp increase while the attack appears. Finally, it will reach another higher balance;

3) If the attack rate is less than 1000 packages per second, the PLC can still communicate with Master IPC without delay, and these delays can be totally ignored;

4) If the attack rate is between 1000 packages per second and 5000 packages per second, the PLC can still communicate with Master IPC with a little delay, and these delays can be largely ignored;

5) If the attack rate is more than 10000 packages per second, PLC communicate with Master IPC with obvious delay, and these delays will cause harmful effect on the networked mechatronics system.

CONCLUSION

Mechatronics monitoring and control system has been widely used in modern manufacturing industry, and has played a key role in the field of automated manufacturing. With the rapid development of network, communication and computer technology, the networked mechatronics monitoring and control system becomes to adopt different kinds of communication ways. This may result in various kinds of cyber-attack on the mechatronics systems. As a result, we need a cyber security testbed to research the cyber-attack problems of the networked mechatronics system.

Aim to meet this research need, we develop such a testbed based on DETERLAB. Thanks to the hardware resources provided by DETERLAB servers, we setup the test nodes such as PLC, Master IPC and etc. Some software are applied in this testbed for generating attack and monitoring the network communication status. Besides, we develop a software tool to make the nodes in the testbed can communicate with each other according to the Modbus TCP protocol. This testbed has been applied to do Dos attack test successfully. The results show that the testbed proposed in this paper has high practicability, reality and scalability. It can do great help for the cybersecurity research work about the networked mechatronics system. With this testbed, we can take on some attack experiments the on it, and analyze the cyber-attack impact on the networked mechatronics monitoring and control system, and then try to find different types of defense solutions for system.

CONFLICT OF INTEREST

The author confirms that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

We acknowledge that our work is supported by the Sichuan Application Fundamental Research Funds (No. 2014JY0212).

REFERENCES

[1] Stouffer, Keith, Joe Falco, and Karen Scarfone. "Guide to industrial control systems (ICS) security." *NIST special publication* (2011): 800-82.

[2] Reaves, Bradley, and Thomas Morris. "An open virtual testbed for industrial control system security research." *International Journal of Information Security* 11.4 (2012): 215-229.

[3] Xiong Qi, Peng Yong, Dai Zhong-hua. Safety risk assessment of industrial control system. China Cyber Security, 2012,27(03):57-59. (in Chinese)

[4] Wei, Qin Zhi. "Industrial Network Control System Security and Management." *Measurement & Control Technology* 32.2(2013): 87-92.

[5] Peng, Yong, et al. "Industrial control system cybersecurity research." *Journal of Tsinghua University Science and Technology* 52.10 (2012): 1396-1408.

[6] INL's SCADA Test Bed. http://www4vip.inl.gov/research/national-supervisory-control-and-data-acquisition-test-bed/

[7] Morris, Thomas, et al. "A control system testbed to validate critical infrastructure protection concepts." *International Journal of Critical Infrastructure Protection* 4.2 (2011): 88-103.

[8] Chabukswar, Rohan, et al. "Simulation of network attacks on SCADA systems." *First Workshop on Secure Control Systems*. 2010.

[9] Bergman, David C. "Power grid simulation, evaluation, and test framework." (2010).

[10] Emulab - Network Emulation Testbed Home. http://www.emulab.net/

[11] Benzel, Terry, et al. "Experience with deter: a testbed for security research."*Testbeds and Research Infrastructures for the Development of Networks and Communities, 2006. TRIDENTCOM 2006. 2nd International Conference on*. IEEE, 2006.

[12] SEER Experimental Workbench, https://seer.isi.deterlab.net

[13] Schwab, Stephen, et al. "Seer: A security experimentation environment for deter." *Proceedings of the DETER Community Workshop on Cyber Security Experimentation and Test on DETER Community Workshop on Cyber Security Experimentation and Test 2007*. USENIX Association, 2007.

[14] Viswanathan, Arun, et al. "A Semantic Framework for Data Analysis in Networked Systems." *NSDI*. 2011.

[15] Benzel, Terry. "The science of cyber security experimentation: the DETER project." *Proceedings of the 27th Annual Computer Security Applications Conference*. ACM, 2011.

[16] Knapp, Eric D., and Joel Thomas Langill. *Industrial Network Security: Securing critical infrastructure networks for smart grid, SCADA, and other Industrial Control Systems*. Syngress, 2014.

[17] Hussain, Alefiya, John Heidemann, and Christos Papadopoulos. "A framework for classifying denial of service attacks." *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM, 2003.

[18] Hussain, Alefiya, and Saurabh Amin. "NCS security experimentation using DETER." Proceedings of the 1st international conference on High Confidence Networked Systems. ACM, 2012.