A Preliminary Study of a Wireless Process Control Network Using Emulation Testbeds

Michele Guglielmi, Igor Nai, Andres Perez-Garcia, and Christos Siaterlis

European Commission, Joint Research Centre,
Institute for the Protection and Security of the Citizen
Via E. Fermi 1, 21027 Ispra, VA, Italy
firstname.surname@jrc.ec.europa.eu

Abstract. The increasing dependence of Critical Infrastructures (CI) from Information and Communication Technologies might encompass significant risks to our society. Experimentation with CI before introducing a new technology has always been difficult mainly because the architecture complexity, the inability to conduct experiments within a mission critical environment as well as the lack of specialized tools for recreating a CI. In this paper we present the first results of a study that was conducted in a specialized environment for experimenting with CI. We propose the use of an emulation testbed (Emulab driven) along with SCADA-aware components in order to recreate a typical Process Control Network (PCN). We present here experimental results of the risks that operators might face while installing Wi-Fi access technologies within a PCN. This work is indicative of the approach, that operators could follow, to measure, understand and minimize undesirable consequences to the resilience of a CI.

Keywords: Critical Infrastructures, SCADA, emulation, resilience.

1 Introduction

IP network technologies, due to their efficiency and the potential for cost-savings, have been increasingly deployed within many different types of Critical Infrastructures (CI), e.g., in the energy and the gas and oil sectors. Recently, this increasing use and dependence of CI from the Internet and IP data networks in general, has triggered concerns about the security of our CI. These concerns are depicted in many policy initiatives under the theme of Critical Infrastructure Protection (CIP) [6],[9] and have triggered the launch of several research activities for the protection of our CI.

This trend, i.e., the proliferation of IP networks, is expected to continue in the years to come as Critical Next Generation Infrastructures will utilize and depend from -a still undefined- Future Internet. One of the main elements of the Future Internet is in general agreement the massive use of wireless technologies. Activities that have reached the news, already indicate that the use of wireless IP networks within Critical Next Generation Infrastructures, e.g., the SmartGrid,

P. Chatzimisios et al. (Eds.): MobiLight 2010, LNICST 45, pp. 268–279, 2010.

is highly probable [2]. But whenever new technologies and architectures are introduced a systematic study of security and resilience aspects of a CI is deemed mandatory. On the other hand, in contrast with other application environments, experimentation within the production network of a CI is prohibited by the high risk of disruption of mission critical systems. We approach this problem by using emulation testbeds, e.g., like Emulab [1] and DETER [3], in order to abstract CI networks and conduct security related experiments. The use of emulation testbeds as a platform to systematically study Process Control Networks (PCN) and Supervisory Control And Data Acquisition protocols (SCADA) has been mentioned in 2008 by Giani et al. [7] but this effort is still in the first stages of development.

In our paper we present a full scale implementation of an emulation testbed suitable for testing and evaluating of resilience characteristics of a CI network.The testbed is augmented by our Programmable Logic Controller (PLC) simulator, that allows the instantiation of a SCADA speaking PLC in a generic PC, and a SCADA master simulator that communicates with PLCs using the Modbus protocol. The developed components allow any researcher working in this field to easily recreate a typical PCN. Furthermore, we present the first experimental results that display the effects that the introduction of wireless networking can incur in a Modbus/SCADA controlled industrial site. Our results indicate that the use of a wireless network can introduce significant delays with undesirable consequences to SCADA controlled processes even in cases of low network utilization.

The paper is structured as follows. We begin in Section 2 with a presentation of a typical industrial CI network and the challenges that the introduction of new technologies can bring. In Section 3 we describe briefly how we can recreate a CI network architecture using an emulation testbed and we continue in Section 4 with the details of our experiments, that investigate concerns about the resilience of SCADA communications over a IEEE 802.11 Wi-Fi network. Finally in Section 5 we summarize the main conclusions of our study.

2 Industrial ICT Critical Infrastructures

In modern "Industrial Process Control Network Architectures", one can identify two different control layers: (i) the physical Layer composed of all the actuators, sensors, and generally speaking hardware devices that physically perform the actions on the system (e.g. open a valve, measure the voltage in a cable etc.); (ii)the Cyber-Layer composed of all the ICT devices and softwares which acquire the data, elaborate low level process strategies and deliver the commands to the physical layer. The cyber-layer is typically using SCADA protocols to control and manage the industrial installation. The whole architecture can be thought as a "distributed control system" spread among two networks: the *Control Network* and the *Process Network*. The process network hosts usually all the SCADA servers and HMI(Human Machine Interface). The control network hosts all the devices which, on one side control the actuators and sensors of the physical layer and on the other side provide the "control interface" to the process network.

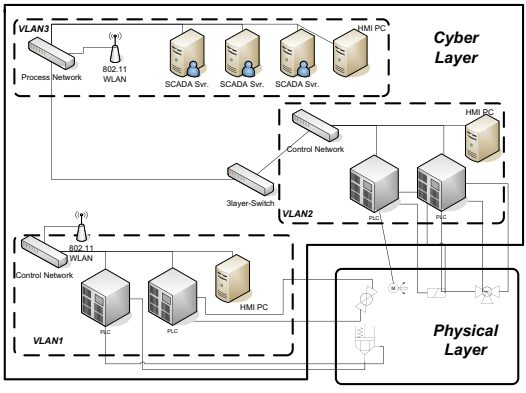


Fig. 1. Example of an Industrial Plant Network

A typical control network is composed by a mesh of *PLC*(Programmable Logic Controller) as shown in Figure 1.

The PLCs receive data from the physical layer, elaborate on the basis of that data a "local actuation strategy", and send back to the actuators commands. The same PLCs provide, when requested, the data received from the physical layer to the SCADA servers (Masters) in the process network, and eventually execute the commands received by them.

In modern SCADA architectures, the communication between Master and PLCs, is usually implemented in two different ways: (i) Through an OPC (Object Linking and Embedding (OLE) for Process Control)layer which help in mapping the PLC devices, (ii) through a direct memory mapping notation making use of the support of some well known SCADA protocol like Modbus (which we will use in this paper as reference since is one of the most used in the field of Industrial Informatics). Modbus is an application layer messaging protocol, positioned at level 7 of the OSI model providing client/server communication between devices connected on different types of buses or networks. The devices can be connected with different networks or buses: EIA-232, EIA-422, kEIA-485 or TCP/IP. A communication transaction comprises a single query and single response frame or a single broadcast frame. Important parameters in a Modbus communication are the **scan rate** with which a Master queries a set of PLCs and the **response timeout**. The exact scan rates depend on the type of installations and processes controlled. The Modbus protocol defines a simple protocol data unit (PDU) independent from the underlying communication layers. Usually, to map Modbus on a specific bus or network, a set of additional fields are added to the application data unit (ADU). The Modbus application data unit is built by the entity that initiates a Modbus transaction (Figure 2). The function code indicates to the slave what kind of action to perform. The data field contains additional information that the PLC uses to take the requested action. The data field may be empty if additional information is not needed. If no errors occur, the requested data are sent back from the PLC to the Master. If an error occurs,

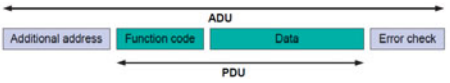


Fig. 2. Illustration of the Modbus ADU

the returning ADU contains an exception code which will be used by the Master to determine the next action to be taken. In average the size of a typical Modbus ADU is 100 Bytes.

3 Recreating a PCN Using an Emulation Testbed

The computing and networking architectures that are typically found within a CI are of significant size and complexity. A typical Process Control Network (PCN) consists of a large number of similar devices (e.g. PLCs), a few standard components (e.g. workstations for HMI) and a small number of specialized servers (SCADA servers/DCS). Using an ad-hoc testbed for experimentation is not recommended because it is very time-consuming and error-prone to setup, maintain and change. An alternative approach is the use of an emulation testbed to recreate the network architecture of a CI. Specifically, our laboratory's testbed is using the Emulab architecture and software [1] which allows us to automatically and dynamically map physical components (e.g. servers, switches) to a virtual topology. In other words the Emulab software configures the physical topology in way that it emulates the virtual topology as transparently as possible [3]. This way we gain significant advantages in terms of repeatability, scalability and high level of realism of our experiments [10].

Our emulation testbed consists mainly of two servers running the Emulab software and a pool of physical resources (e.g. generic PCs and network switches) that are free to be used as experimental nodes. The following steps (Figure 3) describe the re-creation of a PCN network architecture within our testbed:

1. First we need to create a detailed description of the PCN using an extension of the NS language [8];
2. In our description we enumerate similar components as different instances of the same component type (e.g a Virtual-PLC). This way pre-defined templates of different components can be easily reused and automatically deployed and configured.
3. Whenever we want to run an experiment we instantiate it by using the Emulab software. The Emulab server automatically reserves and allocates the physical resources that are needed from the pool of available components;
4. Furthermore the software configures network switches in order to recreate the virtual topology by connecting experimental nodes using multiple VLANs;
5. Finally before the testbed is released for experimentation the software configures packet capturing of predefined links for monitoring purposes.

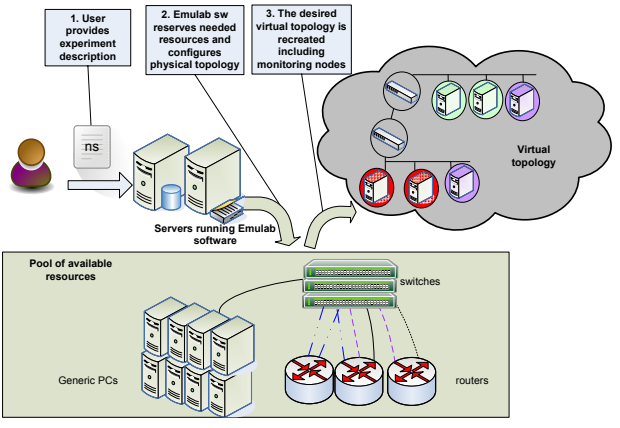


Fig. 3. Main steps for recreating a PCN network within an Emulab-based testbed

As we mentioned in step 2, we have developed templates for typical PCN components. These are practically disk images that can be transparently loaded and run on top of generic PCs. Their main elements are: an Operating System (Windows) and the simulation software that corresponds to the specific component type. In our case we have developed:

- a Virtual-PLC, that simulates a Modbus speaking PLC able to a) Read Coils b) Write Coils etc.
- a Virtual-SCADA-Master that simulates the behavior of a SCADA server that periodically collects information from PLCs using the Modbus protocol, with configurable scan rate and response timeout, and acts based on the collected information.

These components could allow any researcher working in this field to easily recreate a typical PCN and we intend to publish them on the web as soon as we port them into a free OS, e.g., Linux.

4 Introduction oOf Wi-Fi in a PCN, a Case Study

The evolution of telecommunication technologies has also changed Industrial networks. First, there was a migration from the traditional RS-485 communication channel to IP based networks. Today we see that the use of IP wireless networks for industrial systems has been proposed. Of course, classic IEEE 802.11 systems can be only used by devices having direct access to a power source, since the Wi-Fi technology is power consuming. This fact, was the starting point for the development of new standards, such as 6LoWPAN, the IETF draft standard for IPv6 over 802.15.4 that provides a wireless connection service with low power

consumption, at the price of lower transmission power. The IEEE 802.15.4 protocol [4] is already used today by ZigBee [5] and WirelessHart [11]. The advantages of the introduction of wireless technologies in critical industrial infrastructures are evident: less cables within the installation, more flexibility etc. However, the use of wireless communications and especially technologies with low-power transmission, in industrial settings poses serious questions about their reliability, robustness and resilience. One of the concerns is the effect of "interferences", because an industrial CI is a place that is exposed, per se, to electromagnetic interferences, for example generated by electro-mechanical devices like gas turbines. The need for systematic studies on the drawbacks deriving from the use of wireless communications into critical industrial infrastructures is high. However, due to the highly complex interactions between the different elements of a process system, every theoretical and "off-line" analysis, in order to be considered consistent, has to be supported by field tests. These tests, considering the criticality of the process system of a power plant, could be hardly performed safely into a real, in production, installation. An architecture such as the one presented in the previous sections could be extremely useful for process engineers, to quickly re-create industrial environments and study how a process system would react, to interferences and attacks. The approach could be used to answer questions like:

1. How could the use of Wi-Fi within a PCN affect the process reaction delay under different traffic loads ?
2. Which is the maximum bandwidth that a process network can sustain without performance degradation, given parameters like Master scan rate, PLC response timeout etc. ?
3. How many PLCs can an access point handle under different traffic loads?
4. Is there a difference in this number if the process system is in a different state?
5. Is a Wi-Fi network equivalent to the wired network, or are there constraints particular to specific process systems that might discourage the use of Wi-Fi?

To show the potential of our approach, we describe in the next section how we have tried to provide an answer to the first question for a well defined, even if simplified, process system, composed by Virtual PLCs controlling valves and temperature sensors, and a typical SCADA Master.

4.1 Experimental Setup

With the use of our Emulab testbed, we have recreated an experimental environment consisting of a SCADA Master, a wireless PLC, a wired PLC, a traffic generator and a traffic sink. The wireless network is composed of a "commercial, off-the-shelf" (COTS) access point and all components are connected to the same IP subnet (10.1.1.0/24) as shown in Figure 4 and described in the table 1 that follows. All nodes have either wireshark or tcpdump for monitoring and troubleshooting purposes. Iperf is used both in the traffic generator and the sink node to inject background traffic (bidirectional 200 byte UDP packet streams)

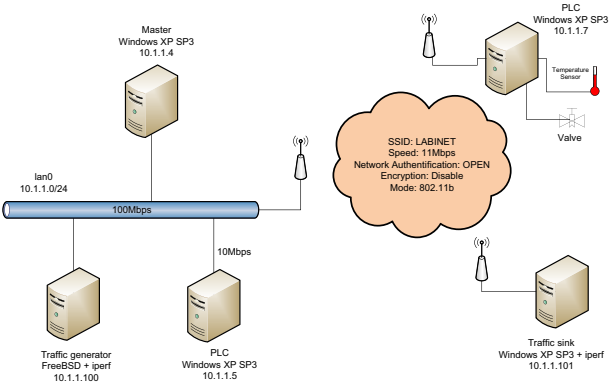


Fig. 4. Topology of the PCN that is used in our experiments

Table 1. Description of experimental nodes

	Master	Wireless PLC	Traffic generator	Traffic sink
Hardware	PC Dell	PC Dell	PC Dell	PC Dell
Interface	Intel Pro 10/100/1000	USB Wireless	Intel Pro 10/100/1000	USB Wireless
Operating System	Win. XP SP3	Win. XP SP3	FreeBSD 6.3	Win. XP SP3
Additional software	Virtual-SCADA-Master	Virtual-PLC	Iperf	Iperf

within the wireless network. This background traffic is intended to simulate traffic from other wireless PLCs associated to the same access point. Therefore, it allows us to test the system's behavior under different load conditions. Emulab inserts transparent nodes in order to model the network in terms of delay, packet loss and bandwidth. Initially, we define a LAN with neither delay nor packet loss, and a bandwidth of 100Mbps. During the experiments, we are able to change these parameters through dynamic events. The wireless network is configured with a bandwidth of 11Mbps, Open authentication and no encryption. Using this topology, we have performed two sets of experiments:

1. Application delay and packet discard rate vs. background traffic. Discarded packets are the packets that either arrive at the application level after the expiration of the response timeout (e.g., 20 ms) or are dropped in the network. In this experiment, the Master sends 1000 queries to the PLC with a scan rate of 60ms and measures the delay of the responses. This scenario intends to simulate a near to real time process, where delay in the response above 20ms is considered high. At the beginning we start without background traffic and we increase it gradually.
2. Application delay vs. packet discard rate. Here, packet discard rate is the only parameter we change through dynamic events in order to test the application delay. The PLC measures a variable that changes with time. The Master has to be able to accurately capture this variation in order to act, e.g., to open a safety valve when the temperature rises above a certain threshold.

As the discard rate increases, the Master will be aware of the variation with a potentially critical delay.

Eventually, we can correlate both experiments to infer how much traffic an access point is able to handle without provoking degradation at the application level.

4.2 Experimental Results

In the first experiment we study the influence of background traffic to the application delay and to the packet discard rate. In the case of a wired network this effect is minimal due to the high performance of the network components. In the wireless network all devices are connected to the same access point and share the collision domain, limiting the network performance. Furthermore it is worth to point out that we have installed the access point and the wireless nodes inside the lab, where we have an infrastructure consisting of switches, servers, PCs, monitors, cooling systems etc. All them are source of electromagnetic interferences which also affects the wireless network performance and stability. We expect that the interferences could be even more severe in an industrial environment such as an electricity power plant.

Figure 5(a) shows the percentage of packets that either arrive out of time at the application level (20 ms) or get dropped in the network (discarded packets), versus the background traffic for both a IEEE 802.11b PLC and a wired 10BASE-T PLC. In Figure 5(b) we show the change of the average delay at the application level. With the wired PLC the network is able to handle nearly 7Mbps of bidirectional traffic without affecting the application, whereas in the wireless experiment we reach some conclusions:

1. There is a variation in the performance of the wireless network therefor we have added a trend line to make the analysis easier.
2. The amount of background traffic the network is able to handle is limited to less than 2Mbps if we require a delay lower than 20ms and a packet discard rate below 1%.

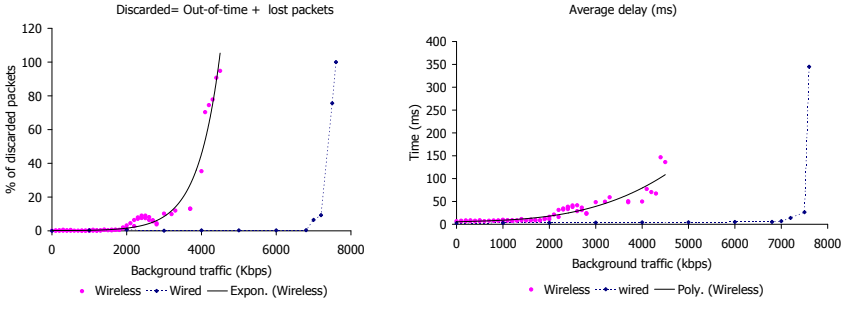


Fig. 5. (a) Experienced packet discards with varying background traffic (wireless). (b) Experienced delay with varying background traffic (wireless).

3. These results could significantly change depending on the hardware due to the use of relatively small packets as background traffic. Since we have used a COTS access point, its performance is limited in terms of packet switching.

Our experiment shows how the introduction of wireless connections in industrial systems, can cause a non negligible effect in terms of delay and packet discard rate, even under moderate traffic load. Therefore an in-depth study of the number of PLCs that will be connected to the wireless network has to be undertaken in order to guarantee the stability of the application performance. The following experiment demonstrates the impact of increasing packet discard rate on the reaction time of a process control application.

In the second experiment, a PLC is connected to a temperature sensor and the Master monitors the temperature by querying the slave every n-milliseconds, where n is the "scan rate". We use "scan rates" of 60 and 120 milliseconds. When the temperature reaches a certain level, the Master sends a command to open a safety valve. The temperature sensor in our platform is simulated by our Virtual-PLC and the temperature increases each 1 millisecond, following a quadratic curve. For presentation purposes, in the figures that follow we focus on a small portion of the whole curve. In Figure 6 the thick line represents the temperature of the sensor which changes every millisecond, the dashed line represents the temperature read by the master with a scan rate of 60 ms and the thin line the temperature read by the master with a scan rate of 120 ms. For example if we assume that the master should trigger an action when the temperature exceeds 24.1 degrees:

- The real temperature of the sensor exceeds 24.12 degrees at time 4,9113.
- The master with 60 ms scan rate notices it at time 4,9171 (58 ms later).
- The master with 120 ms scan rate notices it at time 4,9203 (90 ms later).

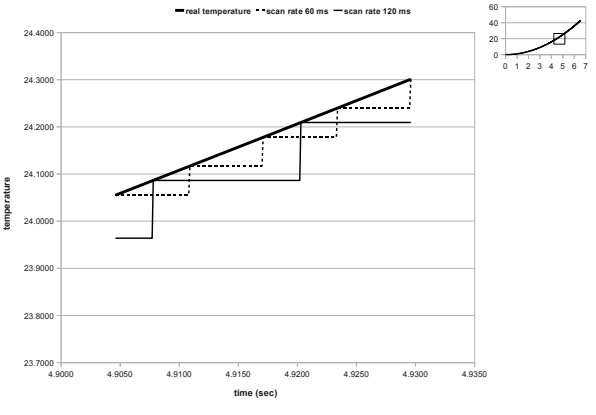
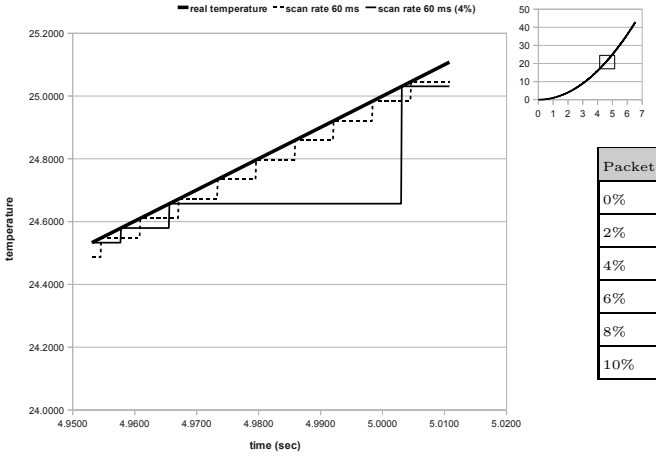


Fig. 6. The temperature perceived by the master with scan rate of 60 and 120 ms



Packet Discards (%)	Average Delay
0%	68.53
2%	107.62
4%	213.11
6%	363.38
8%	618.31
10%	983,5

Fig. 7. The temperature perceived by the master under different packet discard rates

These kind of tests are very useful for designers of CI. The choice of scan rate in the master can significantly influence the control of remote devices in real-time processes. Here, we will illustrate the magnitude of the delay that a monitoring process might experience under different packet discard rates. Using our emulation testbed, we conduct an experiment where the master reads the temperature from the PLC every 60 ms, under different packet discard rates: 0%, 2%, 4%, 6%, 8%, 10%; Figure 7 shows the average delays that the master experiences while getting the temperature from the PLC. If we consider for example the first line of the table in Figure 7 and that the temperature grows every 1 ms, then the master with a scan rate of 60 ms will lose 60 values between a request and the next one. The value in the first row is 68, because the scan rate is 60 ms, but considering the time to send the request to the slave and the time to read the value from the register, then 68 is the number of actual temperature values lost by the master. The average delay will eventually increase once the packet discard rate gets higher. In more details, the thick line represents the real temperature, the dashed line represents the temperature read by the master with a scan rate of 60 ms (0% packet loss) and the thin line the temperature read by the master with a scan rate of 60 ms and a packet discard rate of 6%. The thin line follows the same trend as the dashed line except for a big jump between time 4.9656 and time 5.0030 (374 ms). This jump represents a lost packet and has a great impact on performance, because in the case of the dashed line the master has lost 6 values more than in the case of the thin line.

Let us now discuss the potential consequences. The Master should monitor the temperature sensor, and if the temperature become higher than a certain value, it should immediately open a safety valve. In normal conditions, without packet discards, we would expect a delay in taking the "emergency action" of 68 ms. Considering instead the case of 10% packet losses, the delay would be near to 1

sec., a delay that be critical. Coming back to the results of the first experiment, it is obvious that wireless connections are more vulnerable to such events than wired connections. Moreover, it is evident how industrial protocols like Modbus, that require low time-out and fast scanning rates, might pay a high cost for been ported on top of TCP rather then on UDP. Even if some "implementations" of Modbus over UDP exist, traditionally the industrial community encourage their use only in very limited and specific cases. Unfortunately typical control processes assume near-to-real-time responses, imposing significant constraints on the performance of TCP industrial protocols. But with the use of a platform like the one we have presented, the effects of the introduction of new technologies such as a Wi-Fi network could be studied using an empirical approach.

5 Conclusions and Future Work

In this paper we have demonstrated how emulation testbeds (e.g. based on Emulab) can be used to study the resilience characteristics of Process Control Networks that lie in heart of many Critical Infrastructures. The motivation is threefold: a) experimentation on top of production infrastructures is impossible; b) it is cumbersome and inefficient to recreate CI network architectures with ad-hoc testbeds due to their scale and complexity; c) the use of advanced emulation testbeds can offer significant advantages in terms of experiment repeatability and thus reliability of the results.

Furthermore, we present a preliminary study of the effects that the introduction of wireless networking technologies (such as Wi-Fi) can have on the SCADA communications within an industrial network. Our results show that the use of a wireless network can introduce significant delays in comparison to a wired network with undesirable consequences to SCADA controlled processes even in cases of low network utilization. Our results could help process engineers make informed decisions about the use of Wi-Fi technologies to support SCADA communications.

Our work may be extended in the future towards multiple directions:

1. Assess the impact of other parameters that influence real SCADA applications (e.g., packet re-ordering);
2. Assess the impact of parameters related to the wireless technology (e.g., background noise and interferences, use of other technologies like GPRS, Tetra, WiMax, IEEE 802.15.4);
3. Assess the impact of cyber-attacks (e.g. Denial of Service attacks, Ad-hoc Malware attacks);
4. Assess the impact of lightweight encryption technologies (ID Based Signatures, Elliptic Curves etc.) on the performances of SCADA applications;
5. Connect the emulation testbed with a simulator of physical processes (e.g. generators and turbines) in order to study how events happening in cyberspace can affect physical systems.

References

1. Emulab, http://www.emulab.net/
2. Alvarion. Alvarion and National Grid conduct smart power grid proof of concept in the U.S. Press release (2009)
3. Benzel, T., Braden, R., Kim, D., Neuman, C., Joseph, A.D., Sklower, K.: Experience with DETER: A testbed for security research. In: TRIDENTCOM (2006)
4. De Nardis, L., Di Benedetto, M.-G.: Overview of the IEEE 802.15.4/4a standards for low data rate wireless personal data networks. In: 4th Workshop on Positioning, Navigation and Communication, pp. 285–289 (2007)
5. Egan, D.: The emergence of Zigbee in building automation and industrial control. Computing & Control Engineering Journal 16(2), 14–19 (2005)
6. European Commission: Communication on CIIP - "Protecting Europe from large scale cyber-attacks and disruptions: enhancing preparedness, security and resilience. COM, 149 (2009)
7. Giani, A., Karsai, G., Roosta, T., Shah, A., Sinopoli, B., Wiley, J.: A testbed for secure and robust SCADA systems. SIGBED Rev. 5(2), 1–4 (2008)
8. ISI. Network simulator NS-2, http://www.isi.edu/nsnam/ns/
9. U.S. Department of Homeland Security (DHS). Protecting infrastructure: Critical infrastructure and key resources (cikr), http://www.dhs.gov/files/programs/gc_1189168948944.shtm
10. Siaterlis, C., Masera, M.: A review of available software for the creation of testbeds for internet security research. In: 1st International Conference on Advances in System Simulation, pp. 79–87 (2009)
11. Song, J., Han, S., Mok, A.K., Chen, D., Lucas, M., Nixon, M.: Wirelesshart: Applying wireless technology in real-time industrial process control. In: IEEE Real-Time and Embedded Technology and Applications Symposium, pp. 377–386 (2008)