This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE Globecom 2011 proceedings.

On Network Performance Evaluation toward the Smart Grid: A Case Study of DNP3 over TCP/IP

Xiang Lu*† Zhuo Lu* Wenye Wang* Jianfeng Ma†
*Department of Electrical and Computer Engineering, NC State University, Raleigh NC 27606, US.
Emails: {xlu6, zlu3, wwang}@ncsu.edu
†Department of Computer Science, Xidian University, Xi'an 710071, China.
Emails: jfma@mail.xidian.edu.cn

***Abstract*—The smart grid is the next-generation power system that incorporates power infrastructures with information technologies. In the smart grid, power devices are interconnected to support a variety of intelligent mechanisms, such as relay protection and demand response. To enable such mechanisms, messages must be delivered in a timely manner via network protocols. A cost-efficient and backward-compatible way for smart grid protocol design is to migrate current protocols in supervisory control and data acquisition (SCADA) systems to the smart grid. However, an open question is whether the performance of SCADA protocols can meet the timing requirements of smart grid applications. To address this issue, we establish a micro smart grid, *Green Hub*, to measure the delay performance of a predominant SCADA protocol, distributed network protocol 3.0 (DNP3) over TCP/IP. Our results show that although DNP3 over TCP/IP is widely considered as a smart grid communication protocol, it *cannot* be used in applications with delay constraints smaller than 16ms in Green Hub, such as relay protection. In addition, since DNP3 provides reliability mechanisms similar to TCP, we identify that such an overlapped design induces 50%-80% of the processing delay in embedded power devices. Our results indicate that DNP3 over TCP/IP can be further optimized in terms of delay efficiency, and a lightweight communication protocol is essential for time-critical smart grid applications.**

I. Introduction

The smart grid is an emerging cyber-physical system that incorporates power infrastructures with information technologies [1]. One of the most important features in the smart grid is that power equipments are interconnected via a communication infrastructure to support a variety of intelligent mechanisms, such as real-time monitoring, relay protection and demand response. To enable such mechanisms, communication messages must be delivered on time; that is, the delay performance of message delivery must satisfy its timing requirements, such as 3 ms for relay protection and 16ms for data monitoring in substations [2], [3]. Therefore, *the communication performance is of critical importance in the smart grid.*

In current power systems, supervisory control and data acquisition (SCADA) protocols, such as distributed network protocol 3.0 (DNP3) [4] and Modbus [5], are extensively used for data exchange and management. These SCADA protocols were originally designed over serial links. Toward the smart grid, migrating current SCADA protocols to smart grid applications is widely considered as a cost-efficient and backward-compatible solution [4], [6]. However, an open question is *whether the communication performance of current SCADA protocols can support time-critical smart grid applications, such as relay protection and real-time monitoring* [2]. The answer to this question not only helps to optimize the performance of SCADA protocols, but also provides guidelines for communication protocol design in the smart grid.

To address this question, we establish a micro smart grid, *Green Hub* [7], in the Future Renewable Electric Energy Delivery and Management (FREEDM) systems center [8]. Our objective is to evaluate the delay performance of a currently predominant SCADA protocol, DNP3 over TCP/IP in the Green Hub system. The DNP3 over TCP/IP architecture has been already proposed [6], [9], [10] as a communication protocol for the smart grid. However, existing work focused on security analysis of DNP3 over TCP/IP [5], [9], [11], but overlooked its communication performance in the smart grid. To the best of our knowledge, we are the first to comprehensively measure the delay performance of DNP3 over TCP/IP in an experimental smart grid system. Our findings can be summarized as three-fold.

First, we find that although Green Hub is supported by 100Mbps Ethernet, DNP3 over TCP/IP in fact leads to an average end-to-end delay ranging from 8ms to 20ms in different embedded devices in Green Hub. This shows that DNP3 over TCP/IP can not be reliably adapted to relay protection applications with delay requirements of 3ms–16ms [2], [3]. However, it can still be used to deliver messages in most real-time monitoring and low-speed applications with delay requirements larger than 16ms.

Second, since DNP3 was originally designed over serial links, it has mechanisms for transport reliability, such as retransmission and error detection [4], which provide functions similar to TCP. We identify that such an overlapped design in DNP3 over TCP/IP induces 50%-80% of the processing delay in power devices. Therefore, DNP3 over TCP/IP can be further optimized when adapted to the smart grid.

Third, our experimental results show that, due to limited computational abilities of embedded power devices, the processing delay at network protocol stacks is non-negligible in delay performance evaluation. Hence, a lightweight com-

The work was supported by ERC Program of the National Science Foundation under Award Number EEC-0812121. The first author's research was partially supported by the China Scholarship Council.

978-1-4244-9268-8/11/$26.00 ©2011 IEEE

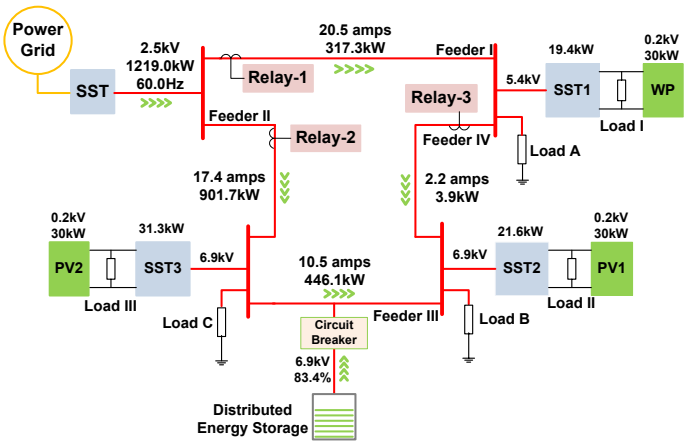

Fig. 1. The physical architecture of Green Hub.

munication protocol is essential for time-critical smart grid applications, especially for relay protection applications.

The remainder of this paper is organized as follows. In Section II, we introduce the background of the Green Hub system, the DNP3 over TCP/IP protocol, and timing requirements of Green Hub applications. In Section III, we present the experiment setups. In Section IV, we illustrate our experimental results and discuss our findings. Finally, we conclude in Section V.

II. Green Hub - A Micro Smart Grid

In this section, we first introduce the physical and communication architectures of Green Hub in the FREEDM systems center. Then, we describe the timing requirements for monitoring and control applications in Green Hub.

A. Background of Green Hub

The one megawatt Green Hub system [7] is a power electronics based electricity generation and distribution system in the FREEDM systems center [8]. It integrates a variety of distributed renewable energy generators, such as solar panels and wind farms, to demonstrate the generation, distribution, storage and management of the green energy toward the smart grid vision.

As shown in Fig. 1, there is a turbine-based wind power (WP) subsystem, and two solar-array based photovoltaic (PV) subsystems PV1 and PV2 in Green Hub. WP, PV1 and PV2 can generate energy supplies for Loads I, II and III, respectively. Energy from each renewable energy subsystem can be also delivered via solid-state transformers (SSTs) to Loads A, B and C to alleviate the burden of the conventional power grid. In addition, extra energy can be stored in the distributed energy storage (DES) system for the future use. A circuit breaker (CB) is installed in the DES system to control its connectivity to Green Hub.

To ensure the reliability of Green Hub with bi-directional power flows, protective relays (Relays 1, 2, and 3 in Fig. 1), which are electrically operated switches, are deployed to handle potential faults or anomalies on power feeders. For

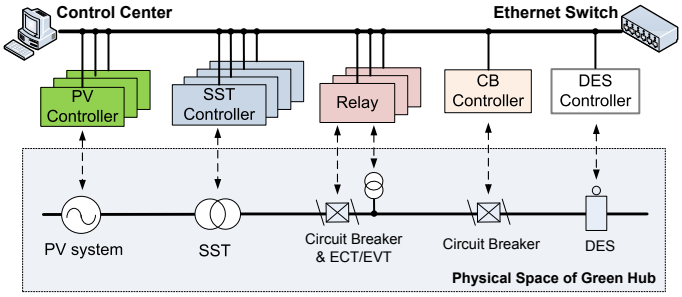

Fig. 2. The communication infrastructure in Green Hub.

example, when a short circuit happens, relays can effectively cut off power feeders to avoid possible damages of critical equipments.

B. Communication Architecture and Protocol in Green Hub

As we can see, Green Hub is a small-scale distributed energy generation and delivery system. Yet, it consists of critical smart grid components, including WP systems, PV systems, SSTs, the DES system and relays. To facilitate efficient interactions between these components, a 100Mbps Ethernet based communication infrastructure is deployed in Green Hub. As shown in Fig. 2, each power equipment is connected via bus communication to an intelligent electronic device (IED), which is a microprocessor-based controller serving as the interface between a physical power device and the communication network. For example, PV systems and SSTs are connected to PV and SST controllers, respectively; while an electronic current/voltage transducer (ECT/EVT) and a circuit breaker are connected to a relay. All IEDs are interconnected via an Ethernet switch to form a single-hop communication network. In addition, the Green Hub control center is connected in the network to support centralized management.

In Green Hub, IEDs serve as communication and data interfaces for power control equipments. Thus, the design of a full-fledged protocol stack from the application layer to the network layer is vital to support Ethernet-based communication between IEDs. Since the DNP3 over TCP/IP architecture has been widely adopted as a communication protocol for the smart grid to achieve good trade-offs between compatibility, efficiency, and simplicity [9], [12], [6], we implement the DNP3 over TCP/IP protocol stack at all IEDs for message delivery in Green Hub. Our objective is to evaluate the delay performance of DNP3 over TCP/IP in Green Hub.

C. Timing Requirements in Green Hub

As aforementioned, message exchange in the smart grid (or in Green Hub) must meet timing requirements to support real-time monitoring and control. Otherwise, out-of-date messages could result in potential system failures. For example, when a short circuit happens on a feeder, a "trip" message should be delivered immediately to a corresponding relay such that the feeder with the fault can be cut off for system protection. Therefore, the message delivery delay is an important metric

TABLE I
REQUIREMENTS OF MESSAGE DELIVERY DELAY IN GREEN HUB.

Type	Delivery delay	Applications
Protection	3 ~ 16 ms	Trip, Closing, Reclosing
Real-time monitoring	16 ~ 100 ms	State reporting
Low-speed	≥ 100 ms	file transferring

TABLE II
LIST OF DEVICE HARDWARE AND SOFTWARE IN EXPERIMENTS.

Device	CPU	Memory	Kernel Version
CB Controller	ARM 200MHz	64MB	ARM Linux 2.4.26
Relay	ARM 500MHz	128MB	ARM Linux 2.6.21
Control Center	P4 1.66GHz	1GB	Linux 2.6.32

for performance evaluation, which is formally defined as follows.

Definition 1: The message delivery delay is the elapsing period from the time instant that a message is generated at the application layer of a power device to the time instant that the message is delivered to the application layer of its destination.

Based on Definition 1, we categorize messages in Green Hub into three types, as shown in Table I.

1) *Protection* messages are used specifically for fault management, such as "tripping" a relay and "closing" a circuit breaker. The acceptable delay of such protection applications ranges from 3ms to 16ms in power systems [2], [13].
2) *Real-time monitoring* messages are designed for information collection, such as the running states of power equipments, whose delay usually ranges from 16ms to 100ms [2].
3) *Low-speed* messages are used for non-emergent events, such as file transferring, which can tolerate message delivery delay larger than 100ms [2].

In the following, we set up our testbed to evaluate whether the performance of DNP3 over TCP/IP can meet the timing requirements of these messages in Green Hub.

III. EXPERIMENTAL SETUP

In this section, we describe the communication scenario used in our experiments as well as our testbed setups.

A. Communication Scenario for Performance Measurement

Since our objective is to investigate whether DNP3 over TCP/IP can be used in time-critical applications including *protection* and *real-time monitoring* in Table I, we consider a relay protection scenario, in which both *protection* and *real-time monitoring* messages are involved.

Such a scenario consists of the following procedures [8], [14]: (i) When a fault (e.g., a three-phase short circuit) occurs on a power feeder, say Feeder I in Fig. 1, all three relays detect the fault and are "tripped" to cut off power feeders to isolate the fault. Accordingly, Green Hub is partitioned into two isolated "islands": Loads I and A are supplied by WP; other loads are supplied by PV1 and PV2. At the same time, all relays send "closing" commands to the CB controller to connect the DES system into Green Hub as additional power supply for Loads B and C to prevent them from potential blackout. (ii) Then, the relays send their reports of the fault and their status changes to the control center. (iii) When receiving the (first) "closing" command, the CB controller closes the circuit breaker to connect the DES system to Green Hub, and then also reports such a "closing" status to the control center.

As a result, the relay protection requires a series of message exchanges.

1) The "closing" commands (*protection* messages) from relays to the CB controller.
2) the reports (*real-time monitoring* messages) from relays to the control center.
3) the report (*real-time monitoring* message) from the CB controller to the control center.

B. Testbed Setups

We set up our testbed to evaluate the communication performance in such a relay protection scenario that involves the relays, the DES CB controller, and the control center. The relays and CB controller are equipped with ARM-based embedded computers, and the control center is a laptop. All parameters of IEDs are listed in Table II.

In addition, as the communication scenario for relay protection is triggered by a fault on Feeder I in Fig. 1, we use FREEDM's Real Time Digital Simulator (RTDS) system [7] to emulate a short circuit on Feeder I and notify relays of the fault to initiate subsequent message exchanges.

IV. PERFORMANCE RESULTS

In this section, we present our experimental results in the communication scenario described in Section III-A. First, to acquire the baseline performance, we measure the DNP3 message delivery delay in a simple network scenario that only consists of three communication devices: a relay, the CB controller and the control center. Then, we measure the practical delay performance in the Green Hub system where all equipments are connected as shown in Fig. 2.

A. Case Study I: Baseline Delay Performance

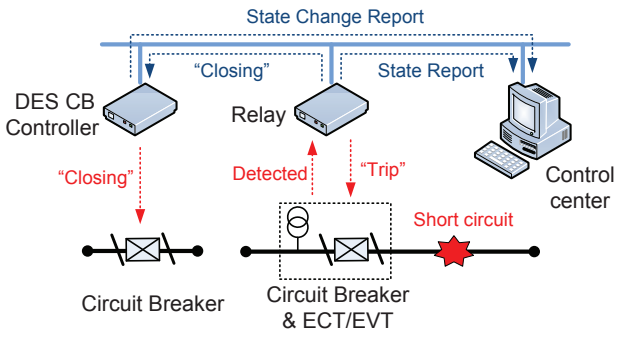

Fig. 3. Relay protection in a simple network setup.

In this case, we use one relay, the CB controller, and the control center to form an Ethernet network to measure the baseline delay performance, as shown in Fig. 3. There is no other device connected to the network.

To ensure the "closing" command and state change reports to be delivered as fast as possible, all the three devices are configured to operate at the DNP3 event-driven mode (also referred to as unsolicited response mode) [4], in which a device can initiate a message transmission immediately when a data change or event occurs. In other words, if the relay reads the fault signal from the RTDS system, the "closing" command and state change report can be instantaneously sent out to the corresponding destinations.

Note that, in a practical system deployment of equipment monitoring, each equipment pair, such as relay/CB pair and CB/control center pair, maintains an active TCP connection that is established earlier before transmissions of emergent messages, such that the control center and the CB controller can monitor corresponding devices through periodically issued data polling [4]. Thus, when the "closing" command and state change reports are transmitted, the time-consuming 3-way handshake can be exempted for TCP connection setup.

1) Experimental Results: Fig. 4 shows the delivery delay performance (with mean, maximum, and minimum) of the three messages: (i) "closing" command from the relay to the CB controller, (ii) state change report from the relay to the control center, and (iii) state change report from the CB controller to control center. We can see that the average delivery delay varies significantly, even though all devices are in the same network. The best performance is achieved by the state report from the relay to the control center whose average delivery delay is lower than 10ms. In contrast, the average delay between the CB controller and the center is 16.2ms with the maximum delay at nearly 29ms. The worst performance appears on the path between the relay and the CB controller, where the average value is over 20ms.

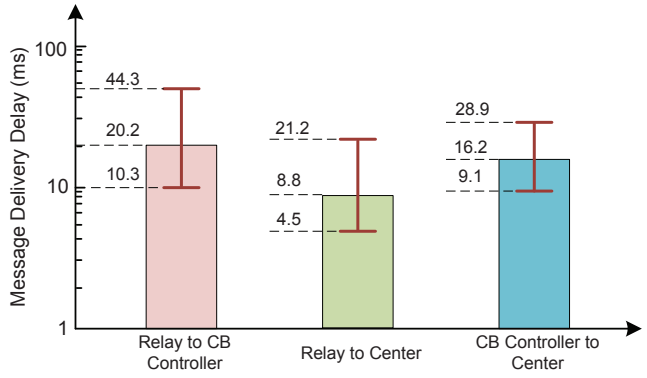

Fig. 4. The message delivery delay performance for different messages in baseline performance evaluation.

2) Results Analysis: In this experiment, we measure the delay performance of three messages delivered in a sequential manner in the same 100Mbps Ethernet-based network, but obtain evidently different performance results. This indicates that the processing time at protocol stacks in embedded computers plays an important role in the delay performance, since the relay and CB controller are both equipped with embedded computers.

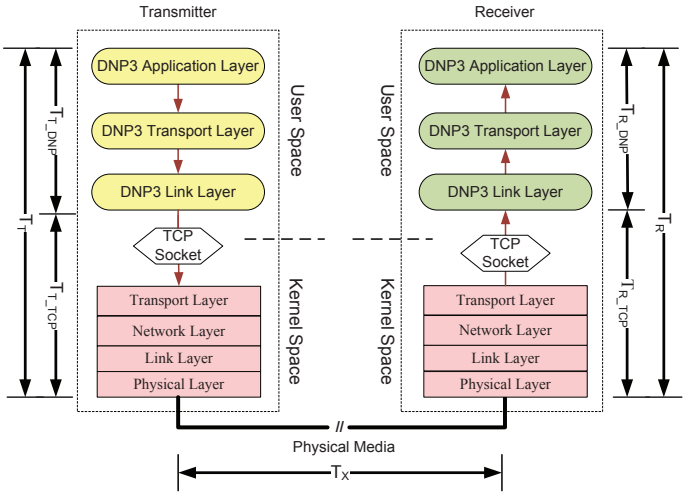

Fig. 5. Delay components in message delivery.

To further explore the effect of processing time, we take a close look at the entire delivery process of a DNP3 unsolicited response message. Fig. 5 illustrates all delay components in an unsolicited message delivery, including the transmitter's processing delay T_T, the network transmission delay T_X, and the receiver's processing delay T_R. T_X is a constant since all messages have the same length during the experiments. Yet, the processing delays, T_T and T_R, may vary significantly with different devices. Thus, we divided T_T or T_R further into two parts: the DNP3 processing delay and the TCP/IP processing delay. Namely, $T_T = T_{T\_DNP} + T_{T\_TCP}$ and $T_R = T_{R\_DNP} + T_{R\_TCP}$. Table. III shows mean values of all the delay components in different devices.

It is evident that DNP3 over TCP/IP can lead to distinct processing delay performance with different computational capabilities. According to Tables II and III, the control center performs best due to its high-speed CPU. Whereas, the two embedded computers, the CB controller and the relay, both suffer from worse processing delay performance because of limited CPU speeds. Since a large amount of smart grid devices are only equipped with embedded computers, we conclude that the processing delay is a non-negligible factor in the delay performance in the smart grid.

Note from Table III that DNP3 results in more processing delay than TCP, which implies that the protocol efficiency is another factor to influence the overall delay performance. To investigate the impact of protocol efficiency, we further break down the message delivery delay into four components: DNP3 application-layer delay, DNP3 transport/link-layer delay, TCP-layer delay, and lower-layer delay. We show the ratio of each delay component to the overall message delivery delay in Fig. 6.

From Fig. 6, we can observe that the delay induced

TABLE III
TRANSMITTING AND RECEIVING PROCESSING DELAY IN DIFFERENT DEVICES.

Device	Transmission (ms)		Receiving (ms)	
	T_{T_DNP}	T_{T_TCP}	T_{R_DNP}	T_{R_TCP}
Relay	11.856	3.087	10.828	1.871
CB Controller	6.088	1.395	5.829	0.847
Center	0.501	0.357	0.489	0.271

by the mechanisms for reliable transport, including DNP3 transport/link layers and TCP, is the most dominant delay component in the overall delay performance. However, reliability mechanisms in DNP3 and TCP are in fact similar to each other. As originally designed over serial links that provide little reliability, DNP3 has its own transport and link layers to achieve reliability mechanisms, such as connection confirmation, cyclic redundancy check, and retransmission mechanism [4]. Similarly, TCP also provides reliability for message delivery. In other words, our results show that such an overlapped design in DNP3 over TCP/IP in fact induces 50%-80% of the overall processing delay in embedded computer based power devices, as shown in Fig. 6.

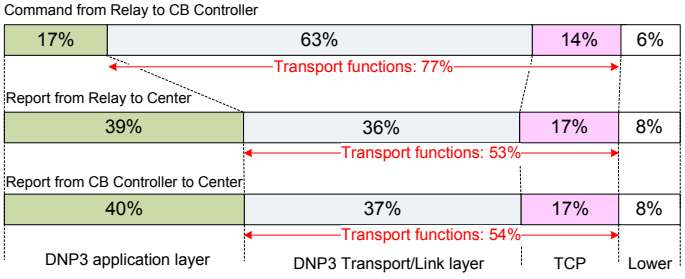

Fig. 6. Ratios of all delay components in the message delivery delay.

3) Lesson Learned: According to Fig. 4, the average delivery delay of the "closing" command is 20.2ms from the relay to the CB controller. However, the "closing" command belongs to *protection* messages whose timing requirement must be at least smaller than 16ms as shown in Table I. Therefore, we conclude that, although DNP3 over TCP/IP is widely considered as a simple and compatible solution to adopt a conventional SCADA protocol in the smart grid [6], it can not be reliably used to deliver *protection* messages in Green Hub. On the other hand, we can see that DNP3 over TCP/IP is still suitable for *real-time monitoring* and *low-speed* applications, since the average delay for status reporting ranges from 8ms to 16ms, as shown in Fig. 4.

Our experimental results indicate that there exist two major solutions to improve the delay performance of DNP3 over TCP/IP in the smart grid. The first is to upgrade the hardware of embedded power control devices with more computational capabilities, which can obviously improve the delay performance by reducing the processing delay. However, this inevitably increases the cost of smart grid devices. The second is to optimize the protocol efficiency, which is promising since we have already identified that the overlapped reliability

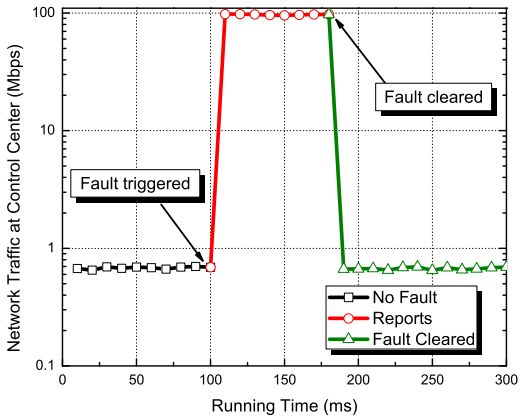

Fig. 7. Network traffic load at the control center.

design in DNP3 and TCP causes redundant processing delay as shown in Fig. 6. In addition, TCP is a throughput-oriented design that induces around 15% delay in Fig. 6. Optimizing TCP for delay-oriented transmission, or even designing a lightweight network protocol, can further reduce the processing delay for time-critical message delivery in the smart grid.

In terms of the lightweight protocol to optimize delay performances of DNP3 transmissions, there exist two intuitive solutions by modifying the overlapped design for reliable transmissions. Firstly, as a more lightweight transmission protocol, UDP can be employed to replace TCP for a DNP3 over UDP/IP architecture. In such an architecture, transmission reliability is provided only by DNP3 link layer. Based on Fig. 6, DNP3 over UDP/IP can achieve at most 17% delay reduction by removing TCP layer. However, such an architecture may result in more retransmissions on the DNP3 layer due to the underlying connectionless UDP links. The second solution is to merge DNP3 Transport/Link layer into the TCP layer for a customized transport layer for DNP3 over TCP/IP. Since DNP3 Transport/Link layer occupies more delay shares than TCP layer, as shown in Fig. 6, the combined transport layer can help reduce DNP3 transmission delay up to 63%.

B. Case Study II: Delay Performance in Green Hub

In the previous experiment, we have measured the baseline delay performance for message exchanges between the relay, the CB controller, and the control center. In this experiment, we consider the same communication scenario, but measure the delay performance in Green Hub, where all power devices are connected to the Ethernet network, as shown in Fig. 2.

1) Traffic Load: Before presenting our experimental results, we first briefly describe the traffic load in Green Hub. In the normal state, all IEDs are set to periodically transmit their running states to the control center with an aggregated traffic rate at 600kbps. In the case of fault management, IEDs that detect the fault are set to report to the control center for centralized management. To ensure a global view of the fault at the control center, all IEDs transmit data to the control

center at the rate of 4800 messages per second [2], [15]. Each message includes an instant sample of power signals, including voltage and current readings. After the fault management, the control center will send commands to devices to resume the normal state.

2) Experimental Result: We use the RTDS system to trigger a fault on Feeder I (as shown in Fig. 1) to initiate the same relay protection procedure used in baseline performance measurement. Fig. 7 illustrates the traffic dynamics at the control center: the traffic is 600Kbps initially, and dramatically increases to nearly 100Mbps when the fault is triggered. This is because, after the fault happens, all IEDs including the relays, SST controllers and PV controllers, will simultaneously detect the anomalies due to the physical correlation between power equipments in Green Hub. All the devices attempt to send abnormal data to the control center, thereby leading to a saturated network traffic load at the control center. With

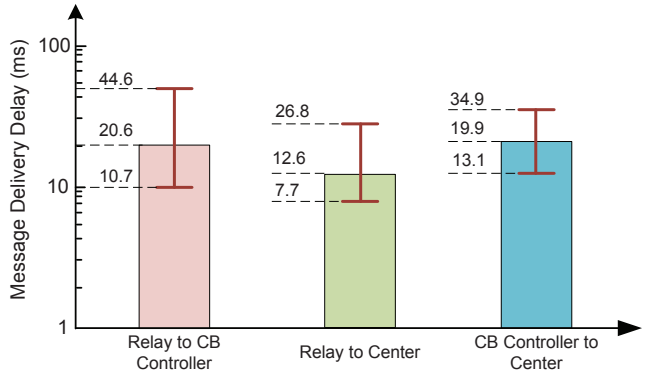

Fig. 8. The message delivery delay performance for different messages in Green Hub.

such traffic dynamics, we measure the delay performance of message exchanges in the same relay protection scenario used in Case Study I, as illustrated in Fig. 8. Comparing Fig. 4 with Fig. 8, we find that the "closing" command from relays to the CB controller remains approximately the same; however, the delay performances of the other two messages are slightly degraded due to the saturated traffic load at the control center. For example, the mean delay from the relay to the control center increases from 8.8ms to 12.6ms, and the mean delay from the CB controller to the control center goes from 16.2ms to 19.9ms. We can conclude that, in the case of fault management, the delay performance of *real-time monitoring* messages delivered to the control center can be degraded due to a traffic flooding effect. However, the experimental results show that DNP3 over TCP/IP is still suitable for *real-time monitoring* in Green Hub.

3) Lesson Learned: Our experimental results indicate that the network traffic load and performance are indeed coupled with physical architectures in the smart grid. For example, when a short circuit happens on Feeder I in Case Study II, all relays can detect the fault by observing a current increase due to the correlation between physical feeders in Green Hub. Accordingly, one event can trigger message delivery between multiple power devices, and further lead to a traffic flooding phenomenon at the control center as shown in Fig. 7.

In addition, our experimental results indicate that DNP3 over TCP/IP can be still used for *real-time monitoring* in Green Hub. On the other hand, we observe that smart grid fault management requires a large amount of information exchanges, which can in turn degrade the delay performance of message delivery, especially in large-scale smart grid networks. In this regard, DNP3 over TCP/IP for the smart grid can be further designed to assign priorities to different messages with distinct delay requirements. For example, *protection* messages must have the highest priority; and *real-time* messages should have a higher priority than *low-speed* messages.

V. Conclusion

In this paper, we established a micro smart grid system, Green Hub, to provide an experimental study on the delay performance of DNP3 over TCP/IP, which is widely considered as a cost-efficient and backward-compatible communication protocol for the smart grid. Our experimental results revealed that DNP3 over TCP/IP can not be directly adopted in time-critical protection scenarios in Green Hub. We identified that the overlapped design for transport reliability in DNP3 and TCP induces 50%-80% of the processing delay in embedded power devices. Therefore, DNP3 over TCP/IP can be further optimized in terms of delay efficiency. Our future work includes the design of a fine-grained and lightweight network protocol for the smart grid.

References

[1] Office of the National Coordinator for Smart Grid Interoperability, "NIST framework and roadmap for smart grid interoperability standards, release 1.0," *NIST Special Publication 1108*, pp. 1–145, Jan. 2010.
[2] IEC Standard, "IEC 61850: Communication networks and systems in substations," 2003.
[3] IEEE, "IEEE standard communication delivery time performance requirements for electric power substation automation," *IEEE Std 1646-2004*, 2005.
[4] K. Curtis, "DNP3 protocol primer," in *DNP User Group*, 2005.
[5] A. West, "Securing DNP3 and Modbus with AGA12-2J," in *2008 IEEE Power and Energy Society General Meeting (PES '08)*, 2008.
[6] S. Mohagheghi, J. Stoupis, and Z. Wang, "Communication protocols and networks for power systems - current status and future trends," in *Proc. of Power Systems Conference and Exposition (PSCE'09)*, Mar. 2009.
[7] M. Baran, Z. Shen, and Z. Liu, "Power management strategies for the green hub," in *FREEDM Systems Center Annual Conference*, 2010.
[8] A. Huang, "FREEDM system - a vision for the future grid," in *2010 IEEE Power and Energy Society General Meeting (PES'10)*, 2010.
[9] T. Mander, F. Nabhani, L. Wang, and R. Cheung, "Data object based security for DNP3 over TCP/IP for increased utility commercial aspects security," in *2007 IEEE Power Engineering Society General Meeting (PES'07)*, 2007, pp. 1–8.
[10] IEEE, "IEEE standard for electric power systems communications - distributed network protocol (DNP3)," 2010.
[11] S. C. Patel, G. D. Bhatt, and J. H. Graham, "Improving the cyber security of SCADA communication networks," *Communications of the ACM*, 2009.
[12] ABB, "630 series DNP3 communicaton protocol manual," 2009.
[13] DOE, "DOE specification - uninterruptible power supply (UPS) systems," 1997.
[14] Y. Jiang, P. A. Tatcho, and H. Li, "The performance analysis of FREEDM system fault with SST," in *FREEDM Systems Center Annual Conference*, 2010.
[15] A. Apostolov, "Testing of complex IEC61850 based substation automation systems," in *International Journal of Reliability and Safety*, 2008.