Deterministic Downlink Transmission in WirelessHART Networks enabling Wireless Control Applications

Johan Åkerberg [#1], Mikael Gidlund [#1], Jonas Neander [#1], Tomas Lennvall [#1], and Mats Björkman [*2]

[#] *ABB Corporate Research, Sweden*
[1] firstname.lastname@se.abb.com

[*] *Mälardalens University, School of Innovation, Design, and Technology, Sweden*
[2] mats.bjorkman@mdh.se

***Abstract*—Wireless sensor and actuator networks bring many benefits to industrial automation systems. However, unreliable wireless and multihop communications among sensors and actuators cause challenges in designing such systems. *Wireless* HART is the first standard for wireless real-time industrial applications. However, current *Wireless* HART standard does not provide services for efficient usage of actuators, which are an essential part of automation. In this paper we focus on *Wireless* HART and propose a periodic and deterministic downlink transmission functionality which enables efficient usage of actuators and control applications. Furthermore, we define new HART commands extending the interface, without affecting available services, to support the integration of actuators. This can be achieved with minor changes in the current standard.**

I. Introduction

Wireless sensor networks (WSNs) are foreseen to become a wireless technology application of major importance in the future. WSNs differ from traditional wireless networks in that they typically are self-organizing (i.e., ad hoc network structure), with a potentially huge number of often randomly deployed, battery-driven small nodes. Adopting WSNs in industrial environments is particular attractive as it allows, in principle at least, the avoidance of cabling, which in many applications turns out to be cumbersome and/or expensive. The main concerns with deploying industrial wireless sensor networks (IWSNs) are about reliability, security, integration, and lack of device interoperability, and these issues have hampered the deployment rate. *Wireless* HART [1] is the first complete and interoperable WSN standard developed for real-world industrial applications. ISA 100a [2] is becoming a standard for process automation and factory automation. ZigBee [3] has been shown to be unsuitable for several process applications since it is not really designed for reliable real-time cyclic communication [4].

Recently industrial control systems integrated with Wireless sensor and actuator networks (WSANs) have received a lot attention due to the significant advantages, e.g., in ease of sensors and actuators deployment, wide coverage, network self-organization, cost, and flexible infrastructure [5]. Still there are many technical challenges to resolve since in an industrial environment there are stringent requirements and WSANs suffer from unpredictable delay, packet loss, energy constraints and interference from other wireless technologies in the same frequency band. Delay and packet loss have been studied widely in the common networked control systems [6]–[8] but they still remain an open issue in WSANs.

Some preliminary results exist on wireless control for *Wireless* HART. In [9], [10], Nixon *et al.* presented an approach to meet the control performance requirements using a wireless mesh network (e.g., *Wireless* HART). Their main conclusion was that device and network operation must be synchronized. In addition, results on integration [11], scheduling [12], clock drift [13], and packet losses [14] have been studied with respect to control and *Wireless* HART. Nevertheless, the work in [12]–[14] assume that actuators are a supported and integrated part of the standard, which is not the case. One of the major challenges for utilizing *Wireless* HART for wireless control purposes is that the current standard lacks proper interfaces to initiate schedules for deterministic and periodic downlink transmission to field devices. Therefore, this paper addresses the use of deterministic downlink communication for enabling the use of actuators.

The main contributions can be summarized as follows:

- We propose a new service called *periodic downlink transmission* for *Wireless* HART, that enables periodic and deterministic transmissions from gateway to *Wireless* HART actuators.
- We define a new set of HART commands extending the interface, without affecting available services.
- We propose a mechanism to utilize the deterministic properties of the downlink transmission to discover errors in a control loop enabling actuators to transit into a failsafe mode.
- We show that our proposed deterministic downlink transmission scheme integrates perfectly into PROFINET IO.

The reminder of this paper is structured as follows. In Section II we describe the layered architecture of *Wireless* HART and PROFINET IO in detail. In Section III we give a brief overview of the periodic uplink transmission in *Wireless* HART. In Section IV we present our proposed periodic and

978-1-4244-5226-2/10/$26.00 ©2010 IEEE

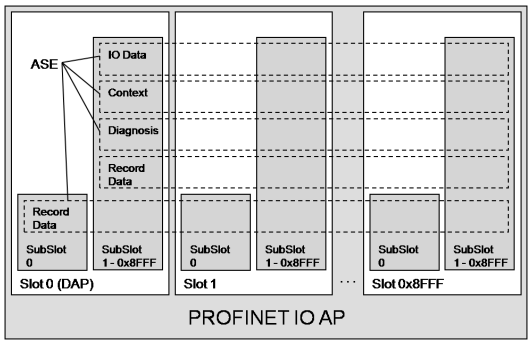

Fig. 1. PROFINET IO device model

deterministic downlink transmission scheme and in Section V we discuss what kind of wireless control applications are enabled by using the proposed downlink scheme. Section VI concludes the paper.

II. Preliminaries

In this section we introduce the basic functionality of PROFINET IO and *Wireless* HART.

A. PROFINET IO

PROFINET IO is one of the Ethernet-based fieldbus protocols from the IEC 61784 standard and is the successor of PROFIBUS. PROFINET IO uses switched $100Mbit/s$ networks to transmit both real-time and non real-time data. For non real-time communication, Distributed Computing Environment - Remote Procedure Call (DCE-RPC) [15] is used on top of UDP/IP. For real-time data, a dedicated layer is defined on top of Ethernet. The application layer can either communicate via Remote Procedure Calls (RPCs) or directly on the real-time channel [16], [17]. PROFINET IO also specifies an Isochronous Real-Time (IRT) channel mainly to be used for motion control, where cyclic messages can be sent periodically in less than $1ms$ and with jitter less than $1\mu s$.

The PROFINET IO device model assumes that there are one or several Application Processes (AP) within the device. Figure 1 shows the internal structure of an AP for a modular field device. The AP is subdivided into as many slots and subslots as needed to represent the physical I/Os of the device. The structure of an IO-Device is described in a General Station Description (GSD) file [18]. By importing the GSD file into the control system, knowledge is gained about the device, for example regarding modules, submodules, parameters, and data types. With this information the engineering tools of the control system can generate the configuration necessary for communication with the device.

It is always necessary to establish an Application Relationship (AR), and within this AR, Communication Relationships (CR) for the data objects are exchanged between the nodes (IO-Device, IO-Controller) via Application Service Element (ASE), see Fig. 2. Within the AR, both IO Data and Record Data are possible, where the former is used to transport process values from the device, and the latter to transport device configuration data.

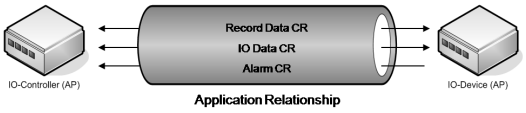

Fig. 2. Application and communication relationships

The PROFINET IO Payload Data Unit can carry at most 1412 bytes I/O data including IO Producer Status (IOPS) and IO Consumer Status (IOCS) [17]. The upper restriction in I/O length is due to the requirement that a PROFINET IO real-time frame must fit into one Ethernet frame to avoid fragmentation of messages.

B. Wireless *HART*

Wireless HART [1] is a reliable and secure mesh networking technology designed for process measurement, control, and asset management applications. *Wireless* HART is an extension of the HART protocol and is designed to be backwards compatible such that wireless segments can be deployed in combination with wired segments. It operates in the 2.4 GHz ISM band, utilizing IEEE 802.15.4 compatible Direct Sequence Spread Spectrum (DSSS) radios, channel hopping, and Time Division Multiple Access (TDMA). All devices are time synchronized and communicate in pre-scheduled fixed length time-slots. Time slots are grouped together into superframes which are repeated according to a specified rate.

Wireless HART is a robust network technology which provides 99.9% end-to-end reliability in industrial process environments [1]. This is achieved through the use of channel hopping and self-healing capabilities of the mesh network. When paths deteriorate or become obstructed the self-healing property of the network ensures it will repair itself and find alternate paths around obstructions.

The security measures provided by *Wireless* HART form a multi-layered always-on solution which is transparent to the application. End-to-end data delivery is secured using 128-bit AES-encryption and the integrity of the data is also ensured. All devices are authenticated before being allowed to join and participate in the network.

Every *Wireless* HART network could consist of five different types of devices (see Fig. 3):

1) One *Network Manager*: This is responsible for managing of the wireless network, such as scheduling, routing, and session management.
2) One *Security Manager*: This manages and distributes security encryption keys, and also holds the list of devices authorized to join the network.
3) One *Gateway*: The gateway connects the control system to the wireless network.
4) One or several *Access Points*: The access point is usually part of the Gateway and acts as the radio interface, multiple AP's are possible making it possible to communicate on different channels in parallel.

5) *Field Devices*: These are devices directly connected to the process (measurement and control), or equipment (asset monitoring) or adapters which connect wired HART devices to the wireless network (retrofit).

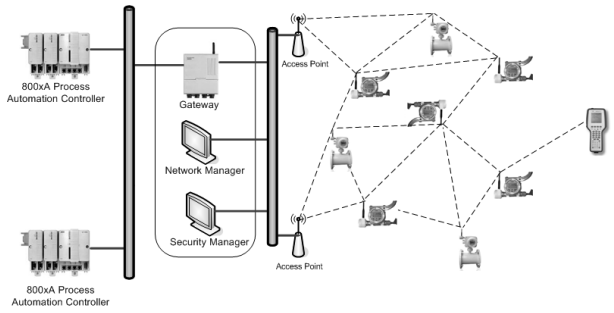

Fig. 3. An example of a *Wireless* HART network

III. Industrial Process Control

Traditionally, Programmable Logic Controllers (PLC) periodically acquire data from sensors (defined as $S_1, S_2, \ldots, S_n$), execute a control application, and finally set the output values for the actuators (defined as $A_1, A_2, \ldots, A_n$), see Fig. 4. Proper error management at the system level is just as important as control algorithms and timing constraints. Therefore, for the use case scenario illustrated in Fig.4, we start with identification of possible error cases, and show how to mitigate them.

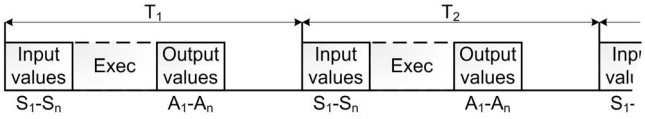

Fig. 4. An example of a PLC period: acquire data from sensors, execute the control application, and finally setting the output values to the actuators

Fig. 5. This picture illustrates a typical scenario of closed loop control, where the sensor provides feedback to the system. Correct functions of all parts (sensor, PLC, actuator, and communication) are required for proper control performance.

For simplicity we only consider two communication failures which are shown in Fig. 6 (PLC unable to acquire sensor readings) and Fig. 7 (PLC unable deliver set-points to the actuator). However, in practice, measures have to be deployed that deal with all kinds of errors. Our approach to mitigate the consequences in case of errors is to use predefined values to be executed by the system, i.e., failsafe mode, which is normally the same state as the actuator takes when it is de-energized. It is common practice to design electromechanical systems such that they take a safe mode if they fail or are de-energized (safety by design).

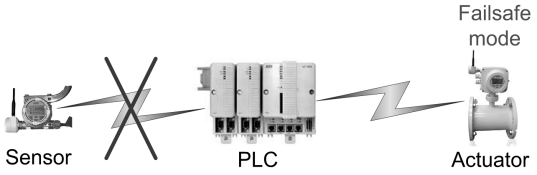

Fig. 6. In case there are problems on the wireless link providing sensor values to the control application, the actuator has to take a safe position (normally de-energize) to avoid that the process is controlled with invalid feedback.

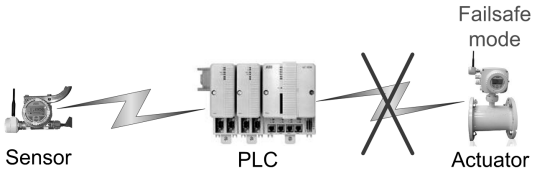

Fig. 7. In case the communication to the actuator fails, the actuator has to take a safe position, to avoid that the actuator freezes in the last position and cause a great deal of problems in the process. The PLC has to detect this communication problem as well, such that it can take parts of the process that depend of the correct function of the actuator in a safe state.

IV. Periodic Uplink Transmissions in *Wireless* HART

In the *Wireless* HART standard there are HART commands defined to enable efficient wireless communication from field devices to the gateway, called *burst mode* in *Wireless* HART, see Table I. The burst service is beneficial to use when data is transmitted periodically, or aperiodically, i.e., when next sensor reading passes a predefined threshold. Aperiodic transmissions are suitable to use when energy should be preserved, i.e., in battery operation. To enable burst mode, a *Wireless* HART device sends a burst request to the Network Manager that tries to change the actual TDMA schedule such that it fits both previous requests and the new request. A burst request is rejected if the Network Manager cannot find a suitable schedule for the *Wireless* HART network. When a burst request is granted, the *Wireless* HART device has slots available according to the requested period time. Table I presents the most important HART commands for burst mode control. Thus, *Wireless* HART provide sophisticated mechanisms and services to transmit sensor readings to the *Wireless* HART gateway, for further processing in the control systems. In process automation, processes are automated by controlling the process via actuators based on carefully selected sensor readings, thus actuators are equally important as sensors. However, the *Wireless* HART standard does not provide such services for actuators as it does for sensors. In the next section we propose extensions to *Wireless* HART that enable efficient data transmissions from the gateway to the devices.

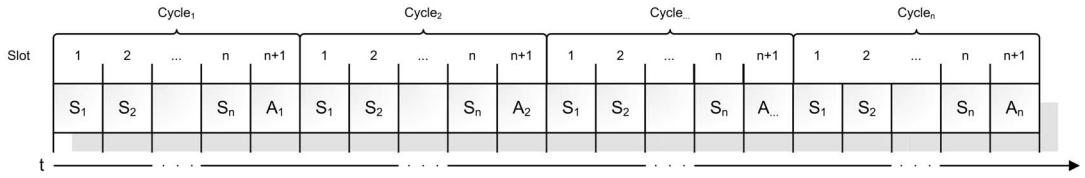

Fig. 8. An example of *Wireless* HART cycles

TABLE I
HART COMMANDS AVAILABLE FOR BURST MODE CONTROL

Cmd	Name	Description
103	Write burst period	This command selects the minimum and maximum update period of a burst message
104	Write burst trigger	This command configures the trigger that forces publishing of the burst message. Four trigger modes are supported: continuous, windowed, rising, and falling.
105	Read burst mode configuration	This command allows the burst mode configuration to be read. The field device responds with whether the device is in burst mode along with current settings.
108	Write burst mode command number	This command selects the response message that the device transmits while in burst mode.
109	Burst mode control	This command is used to enter and exit the burst mode on the device.

V. PERIODIC DOWNLINK TRANSMISSIONS IN *Wireless* HART

In this section we evaluate and extend the *Wireless* HART standard to support periodic transmissions from the control system to *Wireless* HART devices. The *Wireless* HART standard targets industrial control system applications, thus we need to include actuators as a part of *Wireless* HART, to enable it to be used in representative industrial applications. Typically actuators require deterministic communication, thus best-effort communication is not sufficient in most cases.

A. PLC Period and Wireless *HART*

Typical period times for PLC's in process automation range from $250ms$ to $1s$; however both faster ($< 250ms$) and slower ($> 1s$) period times exist. As mentioned earlier in section IV, the *Wireless* HART standard defines a method to set up efficient data transfer from a sensor to the gateway called burst mode. However, there is no definition for how to initiate efficient and periodic data transfer in the opposite direction (gateway to actuator), i.e., the standard lacks HART commands to initiate periodic data transfer to actuators. *Wireless* HART allows the use of proprietary methods to add functionality and therefore it is possible to provide efficient data transfer from the gateway to actuator. Unfortunately, current gateway/Network Manager vendors have focused on efficient data transfer from sensors to the gateway and therefore there is no support for the needed data transfer solution in the opposite direction. Fig. 8 shows an simplified example[1] of a superframe which is scheduled with links (time slots), $S_1, S_2, \ldots, S_n$, for acquiring data from the sensors to the control application, and links, $A_1, A_2, \ldots, A_n$ are empty slots available for sending data from the control application. As can be seen in the figure, all sensor data can be acquired within one superframe cycle, but it takes n superframe cycles to send data to all the actuators using the slots $A_1, A_2, \ldots, A_n$. In the schedule, we can see that the actuators are forced to share the same outgoing link. Furthermore, measurements and experiments show that the time for the actuator to receive the data from the gateway triples when the actuator is one-hop away from the gateway. Our conclusion is that the Network Manager schedules far too few slots per cycle for outgoing traffic, so-called best-effort communication.

Using best-effort communication for distributing set-points for actuators in industrial control systems is not a feasible solution. To achieve good results from a control perspective, jitter and delays should be reduced as far as possible. All the set-points for the actuators need to be distributed back to the devices within the same cycle.

B. Proposed Downlink Transmission

We propose a novel solution for which the *Wireless* HART Network Manager can schedule several outgoing slots (downlink transmission) from the gateway to the devices within the same cycle. The proposed downlink transmission allows actuators to be integrated, requiring deterministic and periodic transmissions of set-points.

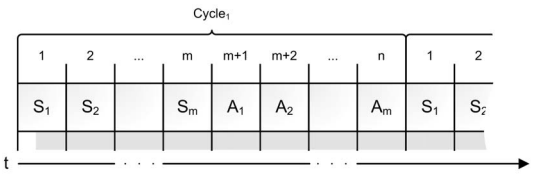

Fig. 9. An example of a desired cycle when using dedicated slots for outgoing packets (actuators).

The proposed solution includes new *Wireless* HART commands that the system can use to request downlink burst mode to the actuators (outgoing slots), see Fig. 9. New *Wireless* HART commands are necessary, as existing commands to initiate periodic transmissions assume that the gateway is the data sink. Since the typical period of an PLC is $250ms$-$1s$ and a *Wireless* HART slot is $10ms$ one can easily deduce that the maximum number of successive slots for an access point ranges from 25 to 100 slots. If a PLC serves as many sensors

[1]In this paper we have left out the details of scheduling as they do not affect the general solution.

TABLE II
NEW HART COMMANDS FOR DOWNLINK BURST MODE CONTROL

Cmd	Name	Description
k	Write burst period	This command selects the minimum and maximum update period of a burst message
l	Write burst trigger	This command configures the trigger that forces publishing of the burst message. Four trigger modes are supported: continuous, windowed, rising, and falling.
m	Read burst mode configuration	This command allows the burst mode configuration to be read. The field device responds with whether the device is in burst more along with current settings.
n	Write burst mode command number	This command selects the response message that the device transmits while in burst mode.
o	Burst mode control	This command is used to enter and exit the burst mode on the device.
p	Write failsafe value and timeout	This command is used to set the output value (i.e. hold last value or move to a failsafe value) of actuators in case of a failsafe timeout.
q	Write actuator mode	This command is used to set the mode of an actuator (in-service, or out-of-service) Iff the actuator is in-service mode and failsafe timer expires, the actuator moves to its failsafe value.

as actuators, there will only be room for 12 incoming slots for sensor data, and 12 slots for actuator data, given a period time of $250ms$. This is because we only have 25 slots available in $250ms$. The *Wireless* HART standard specifies 15 channels for communication, hence, theoretically, adding 14 access points could increase the number of available slots 15 times, if the access points are scheduled to communicate simultaneously on different channels. Adding several access points makes it possible to scale up the number of available slots from 375 to 1500, depending on the assumed PLC period of $250ms$ to $1s$.

A packet in the *Wireless* HART standard can only travel one hop per slot. This introduces delays for devices which are several hops away from the gateway. Another issue is that a device usually either only can listen for a packet, or send a packet simultaneously. Relaying other devices' data will decrease the number of available slots, and could even increase the minimum allowed PLC period. Clustering the network is a solution which could reduce delays by creating simple one-hop clusters around several Gateways, or Access Points if several are used. However, in order to create good network clustering proper planning of the architecture is important.

C. Extending Wireless HART for Periodic Downlink Transmissions

It is extremely important that the system is consistent and reflects the actual state of the process. As already mentioned, we also need to take the actuator to a pre-defined and safe state in case of communication failure etc.

To handle scenarios as illustrated in Fig. 6, we need to monitor the time between two successive sensor update transmissions to detect a communication failure/timeout and indicate it to the control application in the PLC. Then the PLC can set the actuator in a safe state if necessary. However, in the scenario illustrated in Fig 7, the communication protocol must allow both the actuator to monitor the time between two successive actuator set point transmissions, and also that the PLC detects that the communication with the actuator is down. In the same scenario, the actuator needs to take a fail-safe state without assistance of the PLC. The PLC has to detect this as well, to be able to fire alarms to get the proper attention of the operators, or to take a larger part of the process in a safe state automatically.

To address the scenario in Fig. 6, we propose a solution of transactions between the PROFINET IO and *Wireless* HART network as shown in Fig. 10. In the initialization phase, the sensor parameters are transmitted from the PLC to the *Wireless* HART gateway using native PROFINET IO services. The sensor configuration data is forwarded to the *Wireless* HART device using, *Wireless* HART command 103, 104, and 108 and the burst mode starts after the device has received the *Wireless* HART command 109. Later in the data exchange phase, the PLC and the *Wireless* HART gateway periodically exchange cached sensor readings from the *Wireless* HART network using PROFINET IO. The sensor data is received periodically by the gateway according to the burst period and the PLC retrieves cached sensor data according to its PROFINET IO cycle. The *Wireless* HART gateway acts as a proxy between the networks.

Our solution also address the scenario of Fig. 7, where actuators are added as shown in Fig. 11. The initialization phase is similar as in the case of a *Wireless* HART sensor, as shown in Table II, but with an addition of a transaction of the failsafe mode to the actuator, i.e., what should the actuator do in case of an error. In the data exchange phase the *Wireless* HART gateway forwards the actuator set-point periodically from the PLC to the actuator. In addition to this, the actuator transmits keep-alive telegrams periodically to the *Wireless* HART gateway. This enables the PLC to be able to detect a communication failure with the actuator. In case of a communication error, both the PLC and the actuator will detect the error, and take individual actions to avoid that the process is left in an uncontrolled state.

Our solution enables advanced wireless control applications which are not possible today due to the best-effort downlink communication from the gateway to the field devices. Furthermore, our solution also allows actuators to be part of the network and with the ability to enter failsafe mode if necessary.

VI. CONCLUSIONS

In the industrial automation domain, wireless control is an emerging and important application. Nevertheless there are some major challenges to overcome before wireless control can be successfully deployed in real industrial settings. *Wireless* HART is the standard for use in process automation but the major challenge is that the current standard lacks proper interfaces to initiate schedules for deterministic and periodic downlink transmission to field devices.

In this paper, we propose a new service called periodic downlink transmission for *Wireless* HART enabling the use of actuators. *Wireless* HART provide sophisticated mechanisms

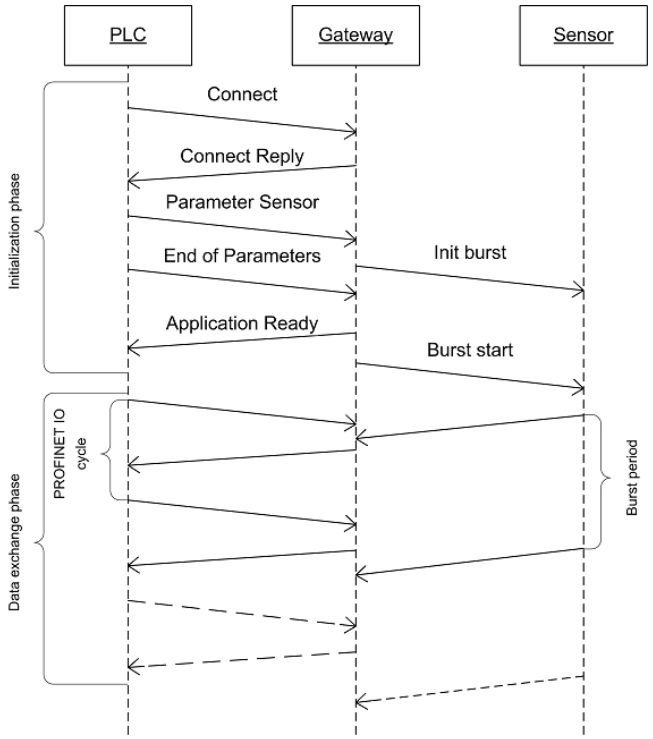

Fig. 10. Illustrates the sequence transaction between the PLC and the sensor through the gateway

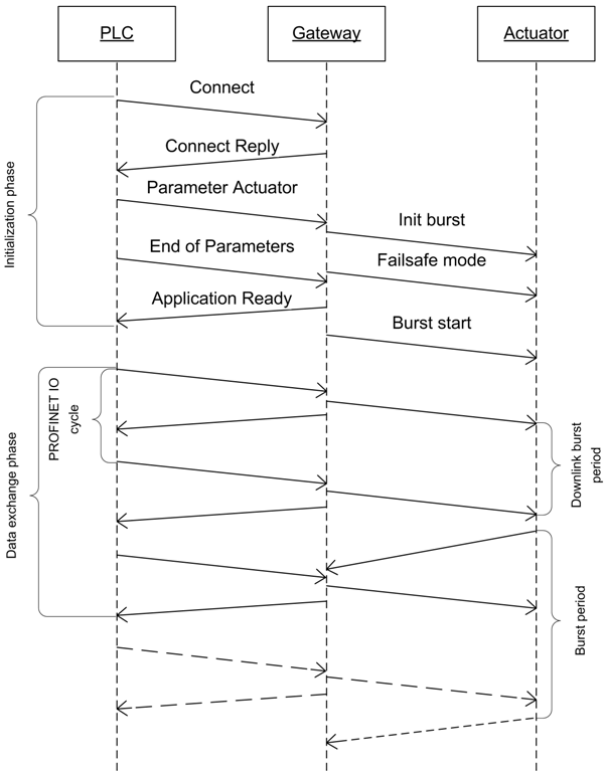

Fig. 11. Illustrates the sequence of transactions between the PLC and the gateway and between the gateway and actuator

and services to transmit sensor readings to the *Wireless* HART gateway, for further processing in the control systems. We extend these mechanisms by introducing deterministic and periodic downlink transmission to actuators. This enables new wireless control applications not possible until now. Furthermore, our holistic approach enables *Wireless* HART to be integrated into existing automation system and protocols such as PROFINET IO. In addition, our solution is in line with the concept of safety-by-design by utilizing failsafe modes to avoid processes running out of control.

REFERENCES

[1] (2010) Hart 7 specification. [Online]. Available: http://www.hartcomm.org/

[2] (2010) Isa 100, wireless systems for automation. [Online]. Available: http://www.isa.org/isa100

[3] (2010) Zigbee alliance. [Online]. Available: http://www.zigbee.org

[4] T. Lennvall, S. Svensson, and F. Hekland, "A comparison of wirelesshart and zigbee for industrial applications," in *IEEE International Workshop on Factory Communication Systems*, May 2008, pp. 85–88.

[5] V. Gungor and G. Hancke, "Industrial wireless sensor networks: Challenges, design principles, and technical approaches," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 10, pp. 4258 –4265, Oct. 2009.

[6] G.-P. Liu, Y. Xia, J. Chen, D. Rees, and W. Hu, "Networked predictive control of systems with random network delays in both forward and feedback channels," *IEEE Transactions on Industrial Electronics*, vol. 54, no. 3, pp. 1282 –1297, June 2007.

[7] T. Li and Y. Fujimoto, "Control system with high-speed and real-time communication links," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 4, pp. 1548 –1557, April 2008.

[8] F. Gil-Castineira, F. Gonzalez-Castano, and L. Franck, "Extending vehicular can fieldbuses with delay-tolerant networks," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 9, pp. 3307 –3314, Sept. 2008.

[9] M. Nixon, D. Chen, T. Blevins, and A. Mok, "Meeting control performance over a wireless mesh network," in *IEEE International Conference on Automation Science and Engineering (CASE)*, Aug. 2008, pp. 540 –547.

[10] J. Song, S. Han, A. Mok, D. Chen, M. Lucas, and M. Nixon, "Wirelesshart: Applying wireless technology in real-time industrial process control," in *IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, April 2008, pp. 377 –386.

[11] J. Åkerberg, M. Gidlund, T. Lennvall, J. Neander, and M. Björkman, "Integration of wirelesshart networks in distributed control systems using profinet io," in *8th IEEE International Conference on Industrial Informatics*, July 2010, pp. 154 –159.

[12] G. Fiore, V. Ercoli, A. Isaksson, K. Landernas, and M. Di Benedetto, "Multihop multi-channel scheduling for wireless control in wirelesshart networks," in *IEEE Conference on Emerging Technologies Factory Automation(ETFA)*, Sept. 2009, pp. 1 –8.

[13] M. De Biasi, C. Snickars, K. Landernas, and A. Isaksson, "Simulation of process control with wirelesshart networks subject to clock drift," in *32nd Annual IEEE International Computer Software and Applications (COMPSAC)*, Aug. 2008, pp. 1355 –1360.

[14] ——, "Simulation of process control with wirelesshart networks subject to packet losses," in *IEEE International Conference on Automation Science and Engineering (CASE)*, Aug. 2008, pp. 548 –553.

[15] *DCE 1.1: Remote Procedure Call.* The Open Group, 1997.

[16] *IEC 61158-5-10. Industrial communication networks - Fieldbus specifications - Part 5-10: Application layer service definition - Type 10 elements.* International Electrotechnical Commission, 2007.

[17] *IEC 61158-6-10. Industrial communication networks - Fieldbus specifications - Part 6-10: Application layer protocol specification - Type 10 elements.* International Electrotechnical Commission, 2007.

[18] *GSDML Specification for PROFINET IO. Version 2.20.* PROFIBUS Neutzerorganisation e.V., 2008.