

Wireless Communication in Process Control Loop:

Requirements Analysis, Industry Practices and Experimental Evaluation

Waqas Ikram¹, Niklas Jansson¹, Trygve Harvei¹, Niels Aakvaag², Ivar Halvorsen², Stig Petersen²,
Simon Carlsen³, Nina F. Thornhill⁴

¹ ABB, Ole Deviks Vei 10, Oslo, Norway, waqas.ikram@no.abb.com

² SINTEF, Strindveien 4, Trondheim, Norway, stig.petersen@sintef.no

³ Statoil, Mølnholtet 42, Harstad, Norway, scar@statoil.com

⁴ Imperial College London, Exhibition Road, London, England, n.thornhill@ic.ac.uk

Abstract—Wireless communication is already used in process automation for process monitoring. The next stage of implementation of wireless technology in industrial applications is for process control. The need for wireless networked control systems has evolved because of the necessity for extensibility, mobility, modularity, fast deployment, and reduced installation and maintenance cost. These benefits are only applicable given that the wireless network of choice can meet the strict requirements of process control applications, such as latency. In this regard, this paper is an effort towards identifying current industry practices related to implementing process control over a wireless link and evaluates the suitability of ISA100.11a network for use in process control through experiments.

Keywords — *wireless sensor networks; wireless sensor and actuator networks; ISA100.11a; PROFINET; controller; NCS*

I. INTRODUCTION

The benefits of wireless technology paved the way for its use in process monitoring applications and are now seeing industrial interest towards its use for process control applications. Traditional control systems operate at fixed rate and assume that the communication networks offer instant communication service without data-loss (1)-(3). This assumption in control implementation is valid given that the dedicated point-to-point communication links are used. However, the use of shared communication networks in control loops violates these arguments (4, 5). It is because in a shared network the data packets are subject to network induced delays, processing delays, and may even experience data-loss. These delays can degrade a system performance and can lead to instability (6,7). Therefore, they have to be carefully taken into consideration in Networked Control System (NCS) design.

In a NCS, the control loops are coordinated and closed over a shared real-time communication network such as a fieldbus. The measurement and controller outputs are transmitted in the form of digital information packets. If wireless communication is employed in a NCS, it is termed a Wireless Networked Controlled System (WNCS). A move towards WNCS also enables wireless sensor and actuator networks as opposed to just sensor networks. In process automation, in the last few

decades, the move from pneumatic control was superseded by 4-20 mA analog loops and later complemented by 4-20 mA smart networks. The drive for reducing the required cabling and minimizing maintenance cost led to the arrival of fieldbuses. All of these succeeding technologies were based on industrial drive and advancements as were observed in communication technology at the time. The current phase to this evolution is adoption of wireless technology which brings alongside unique research challenges. To move from monitoring to closed-loop control is a leap forward and imposes certain constraints on the system design which are the topics of this paper.

The academic papers which have dealt with this research issue are either mostly theoretical papers like (8-10) while others have considered moderate requirements for control application as in (11-13). An effort towards bridging this gap is a drive behind this research. The contributions of this paper are as follows:

- An overview from control system perspective regarding the handling of network and system related time delays.
- Identification of industry practices and constraints relevant to the use of wireless communication in control.
- The ISA100.11a protocol and its relevant features for process control.
- Presentation and evaluation of experimental results obtained from operating an ISA100.11a network.

The layout of the rest of the paper is as follows. Section II brings forth the requirements for the control system to handle network dynamics. Next, section III identifies the industry practices related to the use of process controller in wireless automation. Moreover, section IV highlights the relevant features of ISA100.11a protocol which can address latency concerns. In addition, section V shows the setup used for experimental evaluation of the ISA100.11a protocol. Afterwards, section VI presents the outcome of lab experiments and finally concludes in section VII.

II. WIRELESS NETWORKED CONTROL SYSTEMS

In a classical feedback control loop, the obtainable control loop bandwidth is determined by the process dynamics as seen from the controller. It is well known that time delays in the control loop is the controllers worst enemy and constitutes a hard limit on the obtainable loop bandwidth which again affects the controllers ability to suppress disturbances and handle set point changes in a desired manner. Thus, as a rule of thumb in all control loop design, it must be ensured that the time delays are kept at a minimum. This also includes any lags and delays of the instrumentation system and the selected sampling rate in addition to the inherent process dynamics itself.

In most modern wired process control systems the sampling rates can usually be set much faster than the process dynamics unless for special cases where the processes have very fast dynamics. Thus, for practical purposes, most control loops can be designed without taking special precautions for additional communication and sampling delays.

There may be more than a single wireless channel in and out of each controller. The most common is wireless transmission of a measurement as the feedback signal. But wireless transmission of actuator signals and operator communication is also possible. In the following, focus is on wireless measurements.

There are some properties of wireless systems that bring back the need for closer consideration of the sampling rates and the impact from the properties of the communication channel when the control loops are designed. This is because the sampling rates with a wireless instrument are normally desired to be as slow as possible. This is both for the energy savings perspective, battery lifetime in the transmitter, and also to minimize the total radio traffic.

When looking into the control theory there will be a certain lower limit of the sampling rate before the loop behavior is degraded. The only way to get around this is to trade performance for stability margin and relax the controller settings in order to obtain sufficient stability margin, but at the cost of reduced speed and magnitude of the control actions and by that reduced control loop bandwidth. The need for feedback control actions is mainly determined by the magnitude and frequencies of the process disturbances. Thus, for the same loop, the need for actions will be different when there are little or slow acting disturbances compared to situations when there are large and fast acting disturbances. However, this could be exploited to relax the controller bandwidth and by that enable sampling rates being reduced in calm periods. A key issue in that context is how to detect and decide when the system should switch from low sampling rate to high sampling rate.

In addition to the selection of sampling rate, the controller must also handle packet losses and variations in latency that is expected from the wireless system, both in normal operation and in exceptional situations with increased errors and latency.

The key to a robust and safe operation is to adjust the controller parameters automatically in order to maintain maximum performance with acceptable control loop behavior and stability margins for all expected situations.

To do this properly, a key issue is to have smart signal validation procedure on the controller side that provides the

controller sufficient information to adapt to expected variations in the wireless communication channel.

A simple solution is to ensure time tagged measurements. This requires a reasonable accurate time counter in the transmitter, but the timer does not have to be synchronized with the controller since a particular instrument's clock bias can be found by monitoring a number of incoming packets. If time tagging is not available, this may also be inferred by packet sequence numbers and even signal values.

The control system should have a learning ability and be able to classify the quality of the incoming wireless data:

- *Normal quality with occasional packet losses and minor data latency.* In this case, no particular action needed in the controller. E.g. just freeze output on no update.
- *Reduced quality with some succeeding packet losses and more significant data latency.* In this case, relax the controller performance to adapt to the reduced data quality.
- *Severe reduced quality.* Relax the controller performance further. Consider reconfiguration of communication channel, redundancy etc.
- *Unacceptable loss of consecutive packets and/or data latency that makes impossible to maintain acceptable control performance.* Activate controller fail safe action.

In addition to the basic control loops, all process control systems are equipped with configurable logic. This is to handle normal sequencing like startup and shutdown and also to handle exceptions, for example to realize fail safe actions in case of an instrument error. It is important that the typical failures that may occur in a wireless instrumentation system are detected and the status is made available to the process control system logic.

III. INDUSTRY PRACTICES

In this section, a brief overview of the current practices that are employed in wired NCS is introduced. It will provide a platform to introduce the limitations which are imposed on wireless NCS, if they mimic the conventional control practices.

A. Control loop execution:

In a conventional wired communication network, the control commands are executed periodically. The sensors provide data to the controllers in a periodic manner. The controllers then compute the controller output based on the information received from the sensors. The general rule for executing control command is that it should be executed 4-10 times faster than the process response time (11).

B. Network architecture:

The network architecture is equally important in the performance of a closed-loop system and its integration with the legacy systems. In a conventional NCS, the control is often implemented in a dedicated controller. However, in the case of WNCS it may not be the only option. Based on control systems evolution the control algorithm can be implemented in the following manners (14,15):

- Control algorithm implemented in a controller residing in a plant automation network with cache updated in the gateway. This is the default option.
- Control algorithm can reside inside the wireless gateway.
- Control is implemented in the field. In this case, two variants exist. It can involve running the control algorithm in the field device like in an actuator or at the sensor node. This option is also available in some wired based networks, such as Foundation Fieldbus.

C. Periodic vs. event triggered control:

The controller output works on the process variable value, the values should also be updated often to meet the application demand. This model of periodic execution can be adapted in an alternative wireless communication network, but may not be a feasible option because of limited bandwidth and onboard power. An alternative solution to periodic sampling and control is event-triggered strategy. The use of event-triggered control is not new in some industry sectors, for instance it is used in the power industry. However, it has not been widely used in the process industries. The event-triggered control if used for wireless control can help avoid the unnecessary periodic communication. It will result in reduction in power consumption and network traffic. The work from the researchers in this arena is focusing on achieving similar levels of performance as in the case of periodic control (15-16).

D. Medium Access Control (MAC) methodology:

So far, to ensure determinism, Time Division Multiple Access (TDMA) based MAC protocols have been favored by research community for process automation. The standardised industrial wireless networks are based on this MAC, such as, WirelessHART and ISA100.11a. However, some researchers such as (17), have presented a protocol referred to as BREATH for real-time systems and is based on Carrier Sense Multiple Access (CSMA).

E. Others:

Sensor sampling, controller execution and actuation are often not synchronized. Therefore, over sampling is often used. Frequent periodic sampling and communication of sensor readings can deplete the battery quickly, and is hence a concern for maintenance. Sampling rate adaption is currently ignored.

IV. ISA100.11A PROTOCOL

The ISA100.11a specification (18) was designed to be a comprehensive standard capable of addressing the stringent requirements of monitoring, control and safety applications within the process industries. This section will discuss the specific functionality and capabilities of ISA100.11a related to the time-critical behavior needed for closed-loop control.

A. Data exchange:

ISA100.11a supports two different methods for data exchange: *client/server* and *publishing*. In the former there is an end-to-end acknowledge of the data packet. By contrast, publish mode supports data link layer acknowledge, but not end-to-end. One

fundamental difference between the two modes is how they are handled internally in the transmit buffers. If a new version of the publish variable is sent to the buffer before the last one is transmitted, the last will be overwritten. With client/server, all packets are considered unique and no data will be deliberately overwritten. This difference in behavior makes the two modes suitable for different use. Client/server message exchange will typically be used for data transfer where all packets are required. Typical examples include parameter setting, software download, and safety. Publish will typically be used for control and monitoring application, where old data is without value if new data is available.

B. Packet priority:

In a system with many types of packets, relative priority in the network may be achieved by assigning both a message- and contract priority. This should give the contract requester the possibility to balance the use of resources to prioritize the important packets based on the application they are serving. However, the specification is unclear on the use of the different priority levels. The use of packet priority appears to be at the discretion of the stack implementer.

C. Timestamping:

Published values are in ISA100.11a identified with a single octet monotonically increasing counter. No time value is attached to the message. The counter allows for out-of-sequence detection and loss estimation. If an association to real time is required, this has to be handled by the application.

D. Latency:

The performance of a given wireless transmission path (end-to-end) in ISA100.11a is governed by *contracts*. When a wireless node requires a certain amount of resources, in either publish or client/server mode, it will request a contract from the network manager. For publish data, the request will contain information on the periodicity of the data, the bandwidth required, the ideal data delivery time, and the maximum tolerated latency. The network manager will grant the contract if it has sufficient network resources available. When a contract is granted, it will be guaranteed for complete end-to-end data delivery, possible over multiple hops. However, the contract is only guaranteed for the primary path, so if a node switches to its secondary path due to poor conditions on the primary link, there is no longer any guarantees for the performance related to the given contract.

E. Packet loss:

Data packets occasionally get lost in wireless networks due to a multitude of reasons; noise, interference, fading, shadowing, obstructions, and others. The quality of a communication link can therefore be quantified by its Packet Error Rate (PER), which is the number of packets which do not arrive at their destination. In wireless multi-hop networks, the definition of PER is slightly more complicated. Data packets may get temporarily detained at certain intermediary nodes, and finally make it to the gateway after a long delay. The packet has

arrived, but the data may be too old to be of any value. In ISA100.11a, packets are removed from the network when they are 60 seconds old. For many control applications, the sample rate is much higher than 60 seconds, and data will be obsolete long before they reach the ISA100.11a timeout.

F. Availability:

The availability of a wireless system depends on a number of factors. Centrally controlled TDMA-based systems like ISA100.11a will never suffer from self-interference, i.e. interference from other nodes in the same network. The main reasons for loss of availability are static fading and external noise and interference.

G. Path redundancy and duo cast:

ISA100.11a combats fading problems by using path redundancy. Several parent nodes are available to receive data from a source, hence increasing the probability of successful packet delivery. ISA100.11a furthermore allows for a mechanism known as duo cast. On the final hop to the gateway, the standard allows for redundant baseband routers. Both routers can be set up as recipients of the final hop packets. This redundancy on the final hop is expected to improve the network performance.

V. EXPERIMENTAL SETUP

The experimental setup comprises of the following:

- 8 ISA100.11a sensor nodes as part of the network under investigation. One ISA100.11a gateway with built-in access point, security manager and network manager.
- 1 interfering ISA100.11a network with 3 nodes.
- 35 interfering WirelessHART nodes utilizing all available 15 channels for frequency hopping.
- 6 interfering Wi-Fi networks occupying channels 1, 6, 9 and 13.

The 2-dimensional layout of the floor arrangement where the experiments are conducted is shown in Figure1. It can be seen from the floor map that the maximum distance of communication for a hop is approximately 10m. The

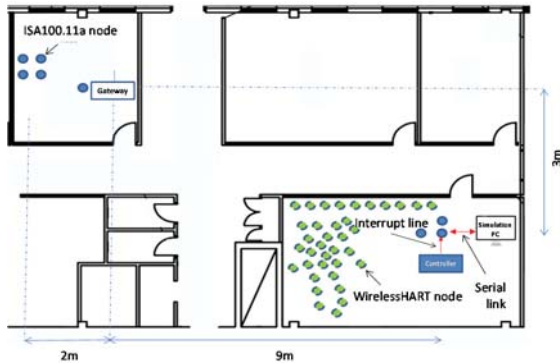


Figure1. Floor plan with the positioning of the nodes.

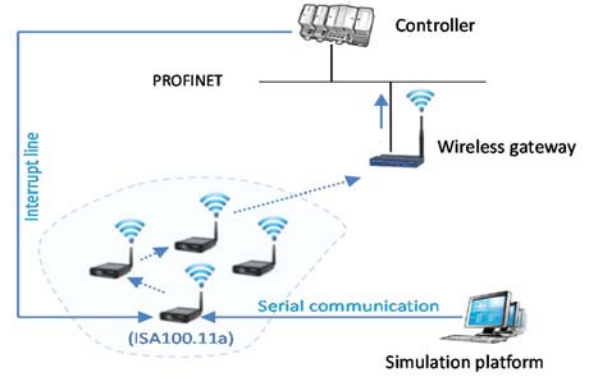


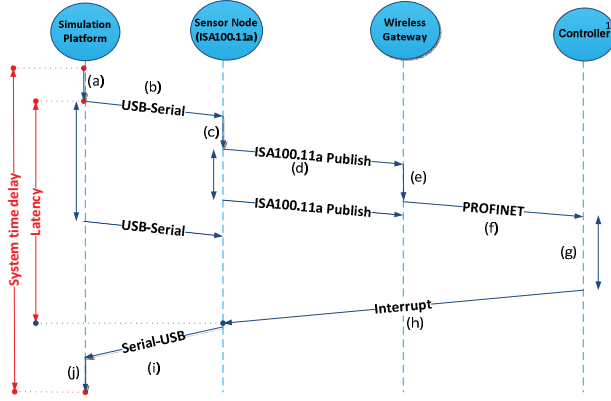
Figure2. Hardware-in-Loop system architecture.

WirelessHART and Wi-Fi networks were already established and were operational during the duration of the experiments. The data traffic from these interfering nodes and clients were independent of the formation of the ISA100.11a network.

Figure2 shows the system architecture used in the experiments. It highlights the components involved, the data flow and the protocols used. The simulation platform is a PC running MATLAB which is used to generate and log the simulated sensor readings and the network Key Performance Indicators (KPIs). Only uplink wireless communication is highlighted with dashed arrows. The data from the simulation platform is accessed by the sensor node (i.e., ISA100.11a node) via serial link. The data then flows through the wireless mesh network towards the wireless gateway. At the gateway, the wireless packets are received and the gateway cache is updated. The controller then polls the data from the gateway by using PROFINET as the protocol. On the reception of a packet delivery the controller generates an interrupt signal back to the sensor node which is used to calculate latency.

Figure3 depicts the detailed view of the timing involved in the process data flow, from value being generated at the simulation PC to being received, acknowledged and recorded. Amongst others, *latency* and *system time delay* (i.e. *delay* for short) are two key values being logged. What they entail is shown in Figure3. In short, latency includes all the delays incurred from sensor data generation to its reception and acknowledgment by the controller via the interrupt line back to the sensor node. The latency value is calculated in the sensor node using the local timer.

The detailed timeline information, shown in Figure3, is provided in Table1. The type column shows the information about the parameter used. Therefore, it can be seen that latency primarily constitutes of wireless sensor network delay; whereas, system time delay also incorporates the time delay incurred within the simulation platform. Latency can be represented using Eq1. The subscripts used are the same sequence numbers shown in Table1 and refer to the time delay in each phase. This simplification assumes no packet loss and that all packets are transmitted immediately on the next available time slot.



¹ CEX bus communication delay is currently omitted

Figure3. Data flow and the associated timeline for the overall system architecture shown in Figure2.

$$Latency = t_c + t_d + t_e + t_f + t_g + t_h \quad (1)$$

Therefore, Eq1 can be simplified to the following equation assuming t_c wait time to be equal to uplink time:

$$Latency(ms) = 148 + 2 * uplink contract$$

Similarly, system time delay can be represented using Eq2.

$$Delay = t_a + t_b + Latency + t_i + t_j \quad (2)$$

In the results shown in the next section, experiments are conducted by varying uplink contracts, numbers of hops and data polling duration (i.e. t_a). Subsequently, their impact on latency and delay are observed and analyzed.

VI. EXPERIMENTAL RESULTS

Deterministic latency is critical for having a network in a process control loop. In order to study the behavior of the latency when subjected to varying hops and uplink contracts various experiments were conducted and the outcome is shown in Figure4. Figure 4 (a)-(c) represents the outcome for 1 hop experiments whereas (d)-(e) represents 2 hop experiments. The results show that there is no explicit common trend in the observations made. Summary of the data is represented in Table2 along with the 95th percentile value. The mean latency in all cases satisfies Eq1 when using maximum bound theoretical values. In addition, the 95th percentile value is also in accordance with Eq1 with the exception of 250ms uplink configurations, for both 1 and 2 hops.

Similarly, Table3 shows the summary of mean delay and the 95th percentile value. The reason for including the total system delay results is to mimic the real world behavior of instruments used in asynchronized mode. In this case the total maximum bound expected is governed by Eq2. The mean delay and the 95th percentile values are within the bounds in all scenarios with the exception of data shown in Figure 5(d). This total delay value shown in Figure 5(d) incorporates the latency shown in Figure 4(d) which itself was a violation of maximum bound of

Eq1. However, this violation can be linked to retransmission of data packets in the network to compensate packet losses. Furthermore, Figure6 shows experimental outcomes with varying data polling rate from the sensor while fixing the number of hops and uplink contract. In this case, most of the observed data was within the limit specified by Eq1 and Eq2. The one hop latency plots for 250ms and 500ms (figures 6c and 6e) are indistinguishable. That is normal as long as the uplink contract is the same, i.e. 250ms. The latency should not be affected by changes in the polling rate. However, in the case of 100ms polling time, the latency is somewhat reduced.

TABLE1. TIMELINE INFORMATION FOR DATA FLOW SHOWN IN FIGURE3.

Sequence	Type	Description
(a)	Configurable	Sensor data polling from the simulation platform to the sensor node software running on the same hardware platform.
(b)	Fixed	USB to serial communication link for connectivity.
(c)	Random	Delay incurred in the sensor node while waiting for an opportunity to transmit the data. Max of t_c is configurable.
(d)	Random	Wireless sensor network related communication delay. Published uplink contracts are used in the experiments rather than client/server mechanism.
(e)	Random	Computational delays inside the wireless gateway.
(f)	Fixed	The cycle time related to PROFINET. The gateway writes to the PROFINET buffer every 128ms.
(g)	Fixed	Controller application task time fixed at 10ms.
(h)	Fixed	Input/Output (IO) cycle time executes every 10ms.
(i)	Fixed	Serial to USB communication link for connectivity
(j)	Fixed	Actuator data polling, fixed to 500ms.

TABLE2. SUMMARY OF THE LATENCY DATA, SHOWN IN, FIGURE4.

Uplink(ms)	Mean latency(s)	Standard deviation	Percentile (95%)
1 hop			
250	0.349	0.167	0.762
500	0.554	0.192	0.768
1000	0.709	0.291	1.014
2 hops			
250	0.303	0.227	0.777
500	0.498	0.315	1.143
1000	0.663	0.417	1.573

TABLE3. SUMMARY OF THE TOTAL SYSTEM DELAY DATA, SHOWN IN, FIGURE5.

Uplink(ms)	Mean delay(s)	Standard deviation	Percentile (95%)
1 hop			
250	0.780	0.274	1.23
500	1.010	0.391	1.69
1000	1.242	0.445	1.92
2 hops			
250	0.881	0.353	1.76
500	1.077	0.478	1.96
1000	1.370	0.644	2.74

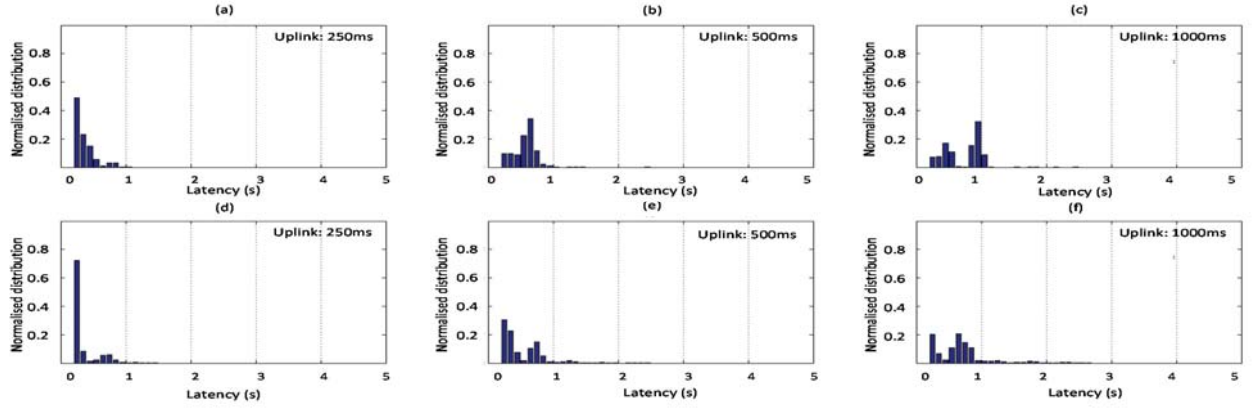


Figure4. Latency plots, (a)-(c) for 1 hop, and (d)-(f) for 2 hop experiments.

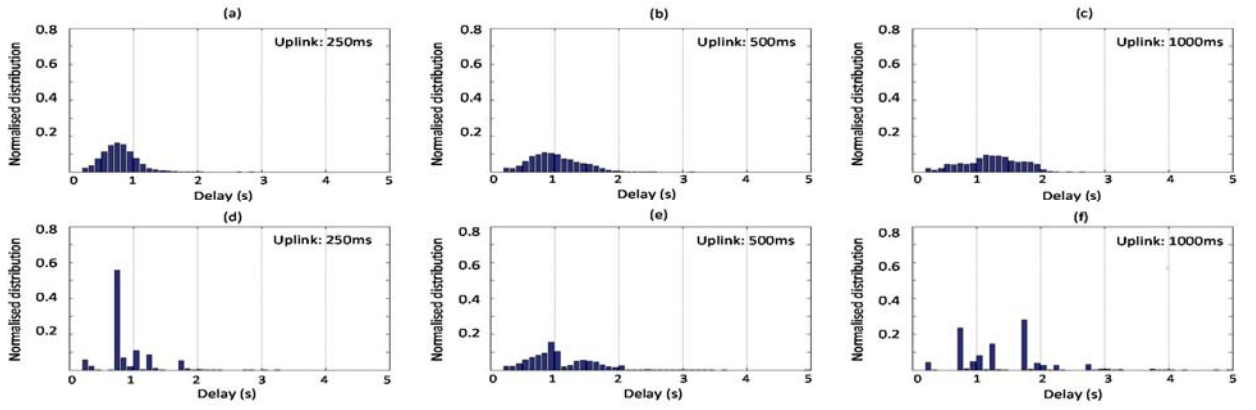


Figure5. Total system delay plots, (a)-(c) for 1 hop, and (d)-(f) for 2 hop experiments.

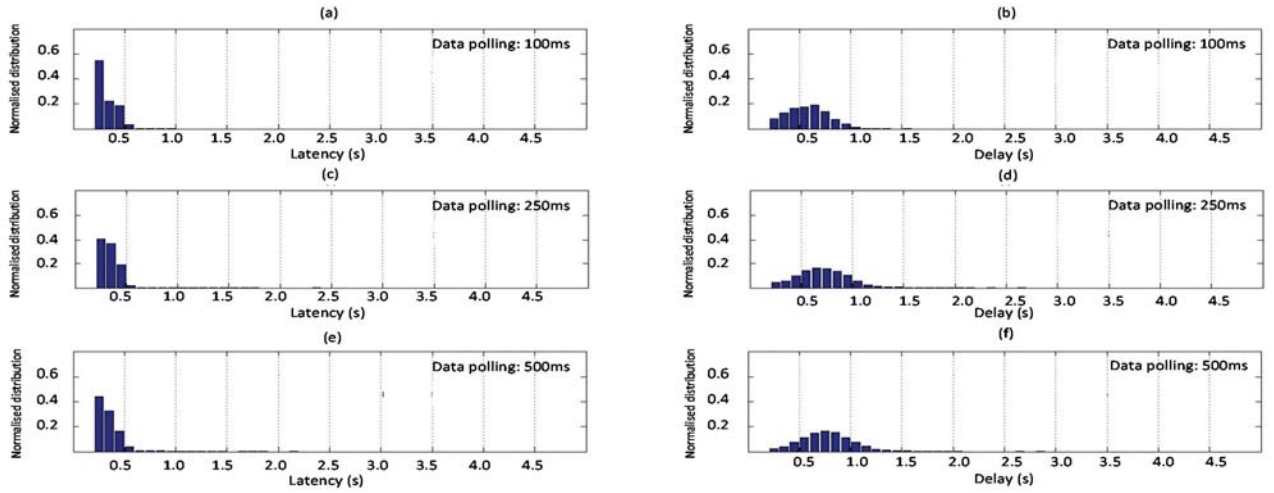


Figure6. (a), (c) and (e) subplots represent latency plots, whereas, (b), (d) and (f) represents total system delay for 1 hop experiment operating at fixed 250ms uplink contract and varying polling rates.

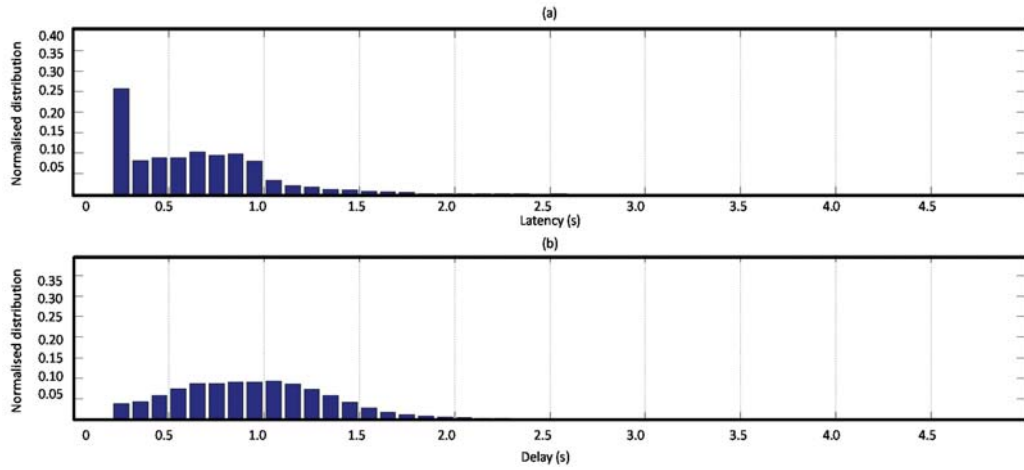


Figure7. Latency and total system delay plots for 3 hop experiment with 250ms data polling and 1s uplink.

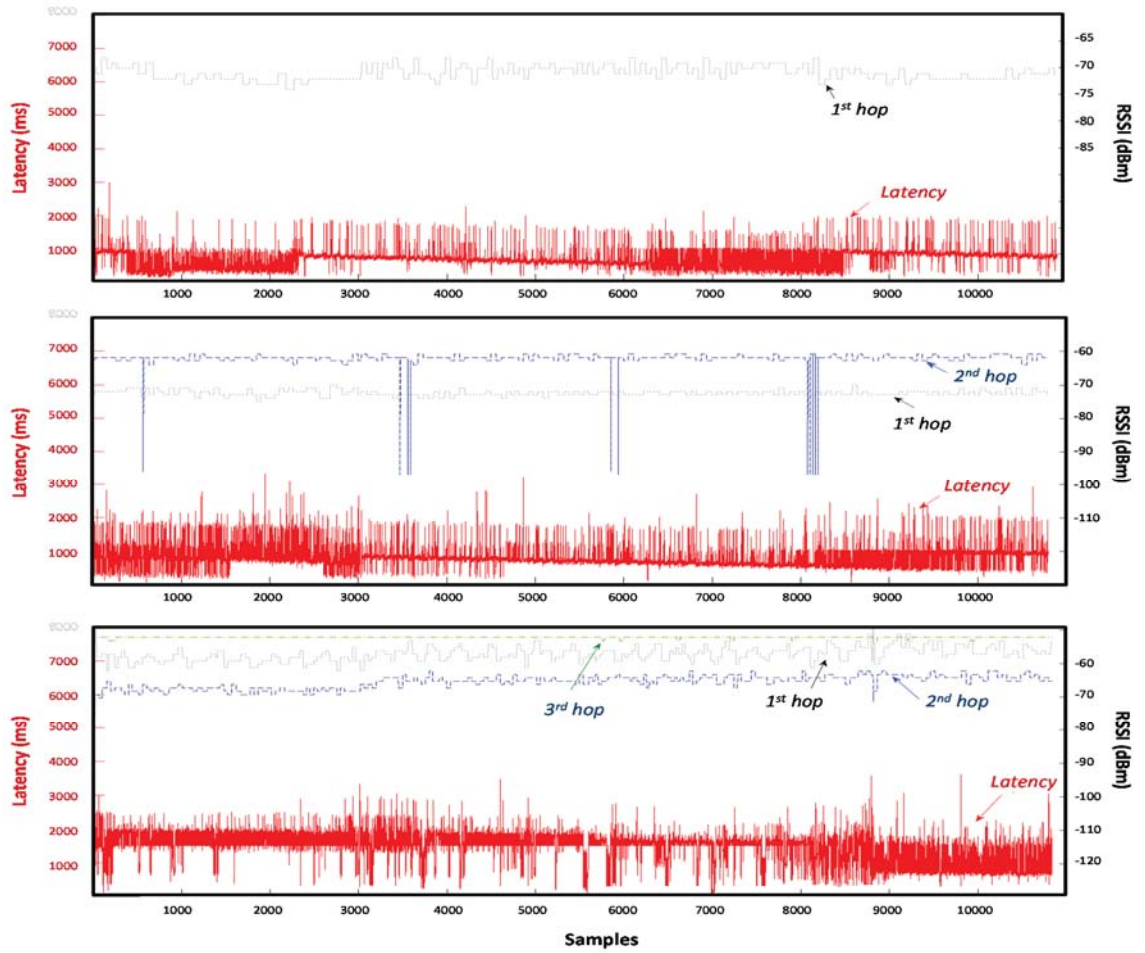


Figure8. Latency and RSSI trends over time for (a) 1 hop (b) 2 hops and (c) 3 hops experiments. Uplink and sensor data polling was constant and set to 1s and 500ms respectively.

This can be explained by the loop settling on a scheme where almost every slot uplink is used, i.e. the polling is fast enough to provide new data for the device in time for every transmit opportunity.

In order to observe the latency and total system delay behavior with a total of 8 ISA100.11a nodes with 3 hops from target sensor to gateway an experiment was conducted and the results are illustrated in Figure 7. Again in this scenario the results show that the latency and delay are within the bounds with few minor exceptions. Moreover, in order to capture the time trend of the network KPIs like RSSI per hop with varying number of hops, and constant sensor polling duration and uplink contract experiment was conducted and the results are shown in Figure 8 along with the latency plot. In this particular experiment the data path was determined and was kept constant throughout the experiment. External antennae were removed from two nodes in the wireless path to restrict the network coverage. Therefore, this experiment highlights the impact of multi-hop on network delay but does not utilize spatial diversity. In the case of Figure 8(a) the link quality was good as can be seen from the RSSI value of the hop presented on the same graph. Even then, latency value has been varying and the clock drift on the sensor node can be observed. There were no packets losses in this case. Furthermore, Figure 8(b) shows 2 hop trends with varying link quality at the 2nd hop. In this case, the total number of packets lost was 10. Similarly, clock drift can be observed here as well. In addition, in Figure 8(c) it is observed that in 3 hop experiment the latency was quite volatile and the clock drift was less visible. A total of 14 packets were lost in this experiment.

Summary of key observations:

- There is no explicit common trend in the latency plots which captures the behavior of 1 and 2 hops with varying uplink contracts.
- Asynchronized sampling and communication can further introduce delays and in the results presented above the maximum system induced delay is used for analysis. The latency will therefore be dominated by when in the cycle the data is written to the transmitting node. Therefore, synchronized sampling and communication, or oversampling can address this issue.
- Even with adequate and relatively stable RSSI values latency has been fluctuating.
- Even with adequate and relatively stable RSSI values data packets were lost.
- Clock skew and drift has been visible in the latency trend plots which is further linked to opportunity to transmit on available uplink slots. Executing the test for a very long time would make the clock drift spread the results more.
- As the number of hops is increased there is more jitter in latency. Packet retransmissions can be a cause of such a behavior.

VII. CONCLUSIONS

This paper has investigated the possibilities of implementing a wireless NCS over the ISA100.11a protocol. The real-time properties of ISA100.11a have been evaluated in an experimental setup, where the network performance has been monitored and compared for different network configurations. The experiments highlight that the observed latency, as measured from the sensor sampling instant and until data is available at the controller, is relatively high even when pushing the recommended boundaries for sensor sampling rates into the sub-second domain. Furthermore, even with relatively good link qualities and low packet loss, variable latencies are observed, which is not ideal for traditional control loop algorithms that assume deterministic data availability with low jitter.

ACKNOWLEDGMENT

The authors acknowledge the support of the WiCon project, funded by the Research Council of Norway (grant 201376/S10).

REFERENCES

- [1] Yang, T. C. (2006), "Networked control system: A Brief Survey", IEEE Proceedings: Control Theory Applications. Vol. 153 (4), pp. 403-412.
- [2] Bai et al., (2011), "Distributed sampling rate adaptation for networked control systems", The 1st International Workshop on Cyber-Physical Networking Systems, Shanghai, China, (Apr. 10-15, 2011).
- [3] Arzen et al., (2006), "On the design and control of wireless networked embedded systems", IEEE Conference on Computer Aided Control Systems Design Munich, Germany, (Oct. 4-6, 2006), pp. 440-445.
- [4] Bemporad et al. (2010), "Networked Control Systems", Springer, Berlin.
- [5] Hristu-Varsakelis, D and Levine, W (2005), "Handbook of Networked and Embedded Control Systems", Birkhauser, Boston, USA.
- [6] Kim, D., Park, S., Kamg, k., and Lee, D. (2007), "Time-triggered wireless sensor network for feedback control", Mobile Communication Division, IEICE Electronics Express, Vol. 4 (21), pp. 644-649.
- [7] Zhang et al., (2001), "Stability of networked control systems", IEEE Control Systems Magazine, Vol. 21, pp. 84-99.
- [8] Nair et al., (2007), "Feedback control under data rate constraints: An overview", Proceedings of the IEEE Vol. 95 (1), pp. 108-137.
- [9] Hespanha et al., (2007), "A survey of recent results in networked control systems", Proceedings of the IEEE Vol. 95 (1), pp. 138-162.
- [10] Schenato, L. (2007), "Foundations of control and estimation over lossy networks", Proceedings of the IEEE Vol. 95, No. 1, (Jan. 2007).
- [11] HCF (2009), "Control over wirelessHART", http://www.hartcomm.org/protocol/training/resources/wiHART_resources/Control_with_WirelessHART.pdf (Sep. 16, 2010).
- [12] Han et al., (2010), "Control over wireless HART network", 36th Annual Conference on Industrial Electronics, Glendale, USA (Nov 7-10, 2010).
- [13] Chen et al., (2010), "WirelessHART: Real-Time Mesh Network for Industrial Automation", Springer Science, New York, USA.
- [14] Biasi et al., (2008), "Simulation of process control with wireless HART networks subjected to packet losses", Conference on Automation Science and Engineering, USA, (August 23-26), pp. 548-553.
- [15] Johansson, K. H. (2009), "Wireless Control: Opportunities and challenges", [Presentation given at: ACCESS Linnaeus Centre School of Electrical Engineering, KTH, Stockholm, Sweden], (Oct 31st, 2009).
- [16] FeedNetBack (2010), "Co-design for Networked Control Systems", <http://www.feednetback.eu/> (On-line: Sep 9, 2010).
- [17] Park et al., (2011), "Wireless networked control system co-design", International Conference on Networking, Sensing and Control, pp. 486-491, (Apr 11-13, 2011).
- [18] ISA-100.11a. (2009), "Wireless systems for industrial automation: Process control and related applications", ISA-100.11a-2009 standard.