The GINSENG System for Wireless Monitoring and Control: Design and Deployment Experiences

TONY O'DONOVAN, University College Cork
JAMES BROWN, Lancaster University
FELIX BÜSCHING, Technische Universität Braunschweig
ALBERTO CARDOSO and JOSÉ CECÍLIO, University of Coimbra
JOSE DO Ó, Petrogal, S.A.
PEDRO FURTADO and PAULO GIL, University of Coimbra
ANJA JUGEL, SAP AG
WOLF-BASTIAN PÖTTNER, Technische Universität Braunschweig
UTZ ROEDIG, Lancaster University
JORGE SÁ SILVA and RICARDO SILVA, University of Coimbra
CORMAC J. SREENAN, University College Cork
VASOS VASSILIOU, University of Cyprus
THIEMO VOIGT, Swedish Institute of Computer Science
LARS WOLF, Technische Universität Braunschweig
ZINON ZINONOS, University of Cyprus

4

Today's industrial facilities, such as oil refineries, chemical plants, and factories, rely on wired sensor systems to monitor and control the production processes. The deployment and maintenance of such cabled systems is expensive and inflexible. It is, therefore, desirable to replace or augment these systems using wireless technology, which requires us to overcome significant technical challenges. Process automation and control applications are mission-critical and require timely and reliable data delivery, which is difficult to provide in industrial environments with harsh radio environments. In this article, we present the GINSENG system which implements performance control to allow us to use wireless sensor networks for mission-critical applications in industrial environments. GINSENG is a complete system solution that comprises on-node system software, network protocols, and back-end systems with sophisticated data processing capability. GINSENG assumes that a deployment can be carefully planned. A TDMA-based MAC protocol, tailored to the deployment environment, is employed to provide reliable and timely data delivery. Performance debugging components are used to unintrusively monitor the system performance and identify problems as they occur. The article reports on a real-world deployment of GINSENG in an especially challenging environment of an operational oil refinery in Sines, Portugal. We provide experimental results from this deployment and share the experiences gained. These results demonstate the use of GINSENG for sensing and actuation and allow an assessment of its ability to operate within the required performance bounds. We also identify shortcomings that manifested during the evaluation phase, thus giving a useful perspective on the challenges that have to be overcome in these harsh application settings.

Categories and Subject Descriptors: C.4 [**Computer Systems Organization**]: Performance of Systems—*Reliability, availability, and serviceability*; C.3 [**Computer Systems Organization**]: Special-Purpose and

---

This research received funding from the European Community Seventh Framework Programme (FP7/2007-2013) under grant agreement 224282. This work has been partially supported by FP7 CONET, the Cooperating Objects Network of Excellence.
Contact author's address: C. J. Sreenan, Department of Computer Science, University College Cork, Ireland; email: cjs@cs.ucc.ie.

DOI: http://dx.doi.org/10.1145/2529975

Application-Based Systems—*Real-time and embedded systems*; C.2.0 [**Computer-Communication Networks**]: General

General Terms: Design, Experimentation, Performance, Reliability

Additional Key Words and Phrases: Oil refinery, monitoring, performance control, wireless sensor network

ACM Reference Format:
O'Donovan, T., Brown, J., Büsching, F., Cardoso, A., Cecílio, J., Ó, J. D., Furtado, P., Gil, P., Jugel, A., Pöttner, W.-B., Roedig, U., Silva, J. S., Silva, R., Sreenan, C. J., Vassiliou, V., Voigt, T., Wolf, L., and Zinonos, Z. 2013. The GINSENG system for wireless monitoring and control: Design and deployment experiences. ACM Trans. Sensor Netw. 10, 1, Article 4 (November 2013), 40 pages.
DOI: http://dx.doi.org/10.1145/2529975

1. INTRODUCTION

Automation and control are fundamental tasks in industrial environments. Today, these are mainly based on wired analog communication, with most industries being heavily automated. Cabled sensors are used for monitoring and communication of sensor readings to a control room in which software is used for closed-loop control, to raise alarms, or for simple process observation. These operational systems are proven to be efficient and reliable and are used even in the most critical scenarios, such as nuclear power industries and oil refineries. However, they present some highly significant drawbacks regarding deployment and reconfiguration.

The deployment of an entire system is very costly, as it requires the installation of thousands of cables which must comply with industry-specific regulatory standards. Further costs accumulate as the installation needs to be protected which often requires the laying of cables underground. This severely limits installation flexibility, making it very difficult to reconfigure the system should that be necessary to accommodate new production processes.

A wireless solution seems highly desirable to overcome the outlined limitations. Wireless sensor networks (WSNs) are, in principle, the closest technology to offer such an alternative. Nodes are readily deployed and system reconfiguration can be carried out easily. The focus of WSN deployments in the research literature has been on applications which are noncritical in terms of performance, such as animal habitat or environmental monitoring. Various aspects of performance control for WSNs have been addressed, such as for real-time medium access control and routing, but there is a lack of systems that offer performance assurances with backend support and combine low energy consumption, high packet delivery rate, and closed-loop actuation. Consequently, and with some justification, industry has shown a marked reluctance to embrace WSNs for mission-critical settings.

In the recent past, several standards have become available that enable the use of wireless communication to connect industrial devices to control systems, mainly targeting remote equipment monitoring. Perhaps the most well known is WirelessHART, which was approved as a standard by the International Electrotechnical Commission (IEC) in April 2010. In Section 6, we discuss WirelessHART and related systems in detail, providing a qualitative comparison with our work. Unfortunately, there appear to be no independent peer-reviewed publications that evaluate the performance of these offerings in a real industrial deployment, and in addition, many key features that affect performance are not specified in the standards but left to individual vendors to implement. Furthermore, there are key differences in the scope of these systems, such that, while these address many key issues for wireless monitoring, our GINSENG solution stands alone as offering a complete solution that includes not just the network protocols and system software, but also encompassing performance debugging and tight integration with industry-grade middleware for complex event management.

Our solution, GINSENG, stems from a multi-partner FP7 European project led by University College Cork in which a performance-controlled WSN has been designed and evaluated. The current deployment is at the Petrogal oil refinery, located at Sines, Portugal. This is the largest refinery in the Iberian peninsula and stands among the largest in Europe. The Petrogal refinery is today completely automated but totally based on wired systems. Upgrades to the sensor/actuator system are frequently needed, but sometimes they are impossible to perform or extremely expensive. Therefore, within the scope of the GINSENG project and to directly answer the needs of Petrogal, a real WSN has been deployed in the refinery. We have studied and developed this network to assure the desired controlled performance and consequently the desired reliability. To meet these requirements, we performed several in-field evaluations. The results allowed us to assess the ability of GINSENG to operate within the required performance bounds for sensing and actuation and to identify shortcomings that manifested during the evaluation phase, thus giving a useful perspective on the challenges that have to be overcome in these harsh application settings. Furthermore, the experiments show that the GINSENG system is able to provide performance debugging information which is necessary to detect reliably problems and to provide data for system reconfiguration.

Unlike most papers that report on wireless sensor network deployments, a special aspect of our work is the opportunity to deploy in a real industrial environment and use it to conduct on-site experiments. Towards this end, we make three key contributions. The first contribution is to present the GINSENG system as an end-to-end solution for industrial monitoring and control, highlighting the novel features of the design. By end-to-end, we mean that it encompasses the backend software (middleware and applications) in addition to protocol and system software. The second contribution is to detail a real and challenging industrial application scenario and our experiences in physically deploying the GINSENG system in that context. The industrial setting we chose is deeply challenging, involving electromagnetic interference, obstacles to radio communication, explosive atmosphere, and difficulties in physical access due to site security. The third contribution is to leverage that deployment to gather valuable results from on-site experiments and analyse the system performance. Our results show that our system is highly successful in that it combines sensing and closed-loop actuation, high packet delivery rates, and low energy consumption.

The remainder of this article is organised as follows. The next section specifies the problem definition and sets the overall context by discussing the application scenarios. Section 3 presents the GINSENG system architecture and details the key elements of the system design. Section 4 describes the deployment of the complete solution at an operational oil refinery, while Section 5 presents the experimental results and analysis. Section 6 provides a summary and comparison with related work, Section 7 offers key lessons from the deployment, and Section 8 concludes.

2. APPLICATION SCENARIOS AND PROBLEM DEFINITION

The Petrogal oil refinery is a complex industrial facility that includes a wide range of processes requiring careful monitoring and control of operations. There are currently 35,000 sensors and actuators in use in the refinery to perform monitoring of industrial operations, such as leakage detection, measurement of pressure in the pipes, the temperature of burners, and fluid levels of tanks. Sensor sampling periods are generally in the order of a few seconds, with 3 s being a typical and acceptable value. The system tolerates a few missing sampling points without causing problems. Such failures are indicated, and maintenance for the sensor is scheduled. For most sensors, set-point values can be defined that, when exceeded, trigger alarms to be handled by control room operators. Some sensors are part of automated control loops where actuators react automatically to sensor input. In practice, it is usually the case that such sensors and

Fig. 1. I/O Cabinets and wiring cabinet.

actuators are located in close proximity because the points of observation and reaction are coincident.

Figure 1 illustrates the current situation at the refinery where each of the sensors is terminated in a wiring closet in a control building. In the long term, the ability to replace most of these wired links with wireless has a multitude of benefits, including cost reduction, increased flexibility in plant reconfiguration, and simplified control room infrastructure.

2.1. Monitoring and Control System

The operations in place at the refinery utilise three systems for the monitoring and control of the plant: the indicatory system, the semi-automatic control system, and the automatic control system. Although this section describes the oil refinery monitoring and control system, these three classifications of systems apply to any industrial plant. All plants have indicatory, semi-automatic control, and automatic control systems and should have similar requirements as the systems in the refinery. Therefore, it should be possible to apply the solutions found for these scenarios to the more general cases.

The indicatory system is used purely to provide the control centre with information about the status and faults of equipment and generic aspects of the environment. Within this system, information flows one way from the in-field sensors to the control centre. It is assumed that data from sensors needs to arrive to the control centre within a given time frame and with a given reliability. Reliability and delay bounds in the indicatory system are not as strict as they are in the two systems described next. Some delay between measurement and display of information in the control centre is acceptable. According to the refinery requirements, acceptable values are delays of 3 s and a transmission reliability of 99%.

The semi-automatic control system is used to control different aspects of the refinery. This system is similar to the previously described indicatory system but includes actuators as well, and information flows in both directions: from in-field sensors to the control centre, and from the control centre to actuators. Upon data arrival from sensors, an operator may decide to send commands to in-field actuators. Operators require instant feedback from sensors, as actuators are used to modify aspects of the environment. The same requirements that applied to the indicatory system exist for this type of system as well, with the exception that it applies both to upstream and downstream data.

The automatic control system is used to deploy automated control loops within the refinery. The system is similar to the previously described semi-automatic control system

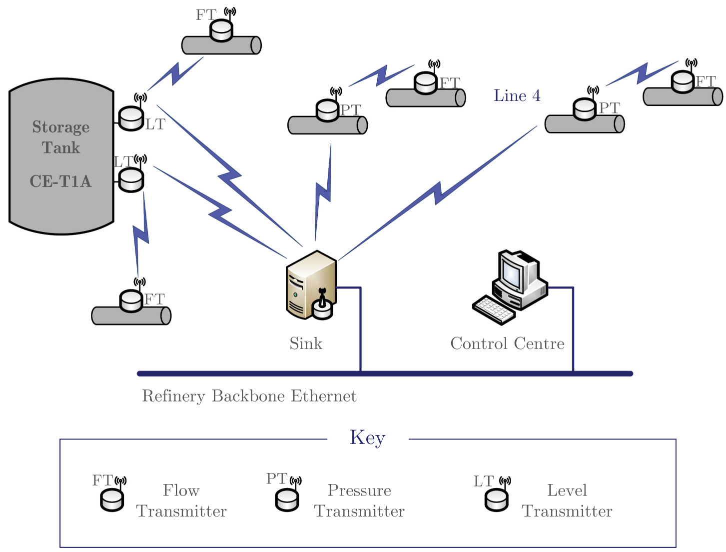


Fig. 2. Production monitoring scenario.

but commands to actuators are sent automatically (by the control system and not after operator intervention) upon receiving sensor data. Sensors and actuators in this system are part of an automated closed loop system. However, operators may be allowed to set parameters which influence the decision process. For example, an operator might configure a control loop such that a valve closes if pressure above an operator-defined threshold is measured. In this system, it is extremely important that data arrives at its intended destination in a timely and reliable manner. Therefore, the required delay is significantly lower than the indicatory and semi-automatic cases; round-trip delays of 2 s (1 s one-way delay instead of 3 s) are typical values.

In the following sections, we describe two refinery operation scenarios in more detail to better illustrate the system requirements. The first scenario describes a production monitoring scenario while the second describes a production monitoring and control scenario. The first example uses the indicatory system while the second one requires the semi-automatic control system. The second example may use elements of the automatic control system in cases where control is configured to run without human intervention.

2.2. Example 1: Production Monitoring

This application scenario is an example of an indicatory system. Figure 2 shows a small section of the plant that has a number of pipes that contain materials that are pumped into a storage tank. In this scenario, sensors of two different types are used to provide information to the control centre staff.

—*Pressure* is monitored within each pipe not only for safety reasons to keep pipe pressure within pipe tolerances, but also to detect leakage. Pressure is usually measured in Pascal (Pa). A typical pressure sensor has a P_{min} and P_{max} using 32 bits as sample size. Pressure is typically sampled every second.
—*Flow* is monitored within each pipe to determine the rate at which product is flowing. Flow is measured in m^3/h every second using 32 bits as sample size.
—*Level* is monitored within storage tanks. Levels are measured in m every second using 32 bits as sample size.

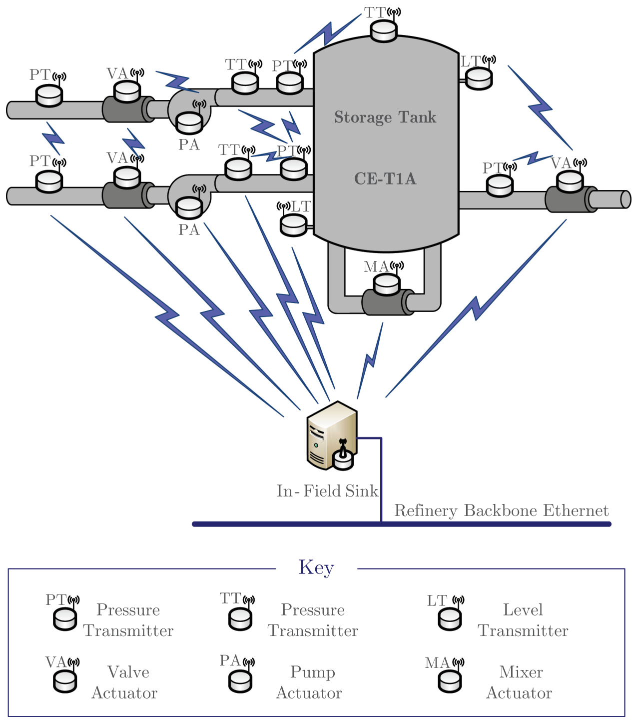


Fig. 3. Production control scenario.

In this scenario, all sensors are in close proximity (less than 50 m). However, due to environmental conditions (a built-up environment with a lot of metal pipes), it is expected that some nodes would have to fulfil two functions: sensing and data forwarding. The required network lifetime, that is, before maintenance needs to be scheduled for changing node batteries, is at least 180 days.

2.3. Example 2: Production Monitoring and Control

This scenario is an example of the semi-automatic control system. Figure 3 shows a small section of the refinery. The information is carried from sensors to the control centre to allow technicians to make production-based decisions. With the use of actuators, technicians can alter aspects of production, such as the speed of product flow, which is controlled via pumps. In addition to the sensors seen in the previous scenario, actuators are also included that can configure pumps, mixers, or open/close valves.

This scenario includes three sensor types measuring temperature, pressure, and filling levels. This information is sent to the control centre, which is monitored by technicians. In addition to the three types of network devices seen in the previous scenario, additional actuator devices are present. Technicians using this information can manage production by controlling three types of plant objects via actuators. These actuators are necessary to enable control of production. These actuators include the following.

—*Shut-off* valves are integrated into pipes and are used to interrupt product flow during day-to-day operations and in the case of emergency.

—*Pumps* can operate at different speeds to increase or decrease the pressure and thus flow of product through the piping system. A 32-bit value is sufficient to set pump speed.
—*Mixing tanks* can blend together products. When mixing is enabled, the contents of the tank are blended and forced out into the output pipe for continued processing at another location. Actuators control the speed at which the mixers operate. A 32-bit value is used to set mixing parameters.

As described at the beginning of this section, commands need to reach an actuator within 3 s and with a reliability of 99%. Again, in this scenario, all sensors are in close proximity (less than 50 m), and connection points to the refinery backbone are available. The expected network lifetime is also at least 180 days.

3. GINSENG SYSTEM ARCHITECTURE AND DESIGN

The GINSENG system assumes that sensors and actuators can be organised in relatively small wireless sensor networks which are connected to a wired backbone infrastructure. It is assumed that highly performance-critical communication (i.e., for control loops) is executed within a WSN and only communication with more relaxed performance requirements (i.e., for control room monitoring; for control loops with slow cycles) extends into the backbone infrastructure. The GINSENG system architecture follows these structural assumptions which allows us to provide a system with the objective of assuring strict performance. These assumptions generally fit production process configurations very well. Sensors and associated actuators that are part of a tightly controlled production process are often found in close proximity of each other (see Section 2). Furthermore, a dedicated wired backbone infrastructure which connects production areas and contains central elements, such as the control room, is generally present in larger facilities.

The GINSENG system uses a distributed middleware to facilitate application data exchange among nodes in different wireless clusters and backend services. Furthermore, the middleware provides a uniform Application Programming Interface (API) for:

(1) Implementing data monitoring and control loops;
(2) Attaching user interfaces (e.g., to implement control room functionality);
(3) Adding data storage (e.g., to implement data logging);
(4) Adding external systems (e.g., maintenance management and supply chain management).

GINSENG communication in the backbone network assumes a network that is able to provide deterministic performance, that is, the network is able to give bounds on data transport delay and reliability. In practice, a sufficiently provisioned best effort network fulfils this requirement for most application cases (as we will show in Section 5). GINSENG communication in the WSNs is based on a Time Division Multiple Access (TDMA)-based Medium Access Control (MAC) protocol, called GinMAC that provides strict bounds for message transport reliability and delay. A number of additional modules that must work alongside a MAC aiming at providing performance guarantees are tightly integrated with GinMAC. These include, for example, topology control, queue management, and performance debugging facilities.

GINSENG applications running on sensor/actuator nodes and middleware components involved in processing performance-critical messages are designed to provide bounds on processing times. Thus, given bounds provided by GinMAC or the backbone network (if involved in the communication path), it is possible to implement a system that provides performance assurances. The adherence to strict delay and reliability

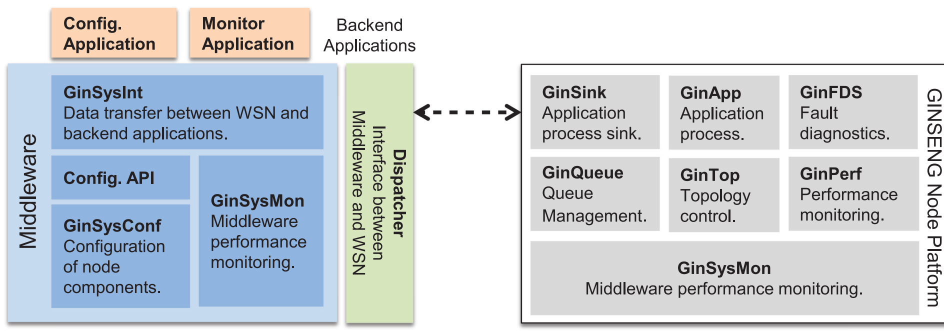


Fig. 4. GINSENG system components.

requirements is essential in the context of process automation and control as it is a requirement for critical monitoring tasks and, in particular, control loop implementation.

The GINSENG system monitors message timing in each part of the system. When a time bound is exceeded, an alarm is generated and a control engineer is informed to take action. An interface for the specification of desired latency bounds concerning processing and delivery in the WSN and backbone is provided. Latency information is collected within messages as they travel through the system. When the messages arrive at the middleware the information is evaluated against defined bounds.

3.1. GINSENG System Components

A GINSENG system employs a number of software components which are located either on nodes deployed in wireless clusters or on standard server systems within the backend infrastructure. The next paragraphs describe the main components of a GINSENG system. The system architecture is shown in Figure 4.

GINSENG Middleware. The GINSENG system uses a middleware to facilitate application data exchange among sensors and actuators in different WSNs and among backend services. To scale for large industrial scenarios, the middleware design enables the distributed deployment of the following components.

GinSysInt. The System Integration component facilitates data transfer between WSNs and backend applications. Measurement data from sensors and additional diagnosis data (such as wireless network performance monitoring data) is fed into the middleware, pre-processed, and forwarded to all subscribed consuming applications. Further, actuation commands and configuration data for sensors and actuators in WSNs are passed from the middleware to WSNs.

GinSysMon. The Performance Monitoring continuously controls the performance of the middleware during runtime. It extends and applies the data quality metrics and algebra presented in Klein and Lehner [2009b] towards the GINSENG application scenarios. Secondly, whenever it detects a quality decrease or a user triggers a quality optimisation with regard to a specific performance metric, the Performance Monitoring component starts the quality-driven optimisation of the sensor stream processing [Klein and Lehner 2009a]. Finally, the performance monitoring enables the performance-driven load shedding [Klein et al. 2009] to avoid middleware overload situations and enable reliable data processing. Essentially, this middleware element ensures that processing delays stay within required bounds.

GinSysConf. The System Configuration enables the definition of commands that configure the data gathering and actuation via GinApp and GinSink, as well as the configuration of agents in the GinFDS (all described next). Such commands can be either defined manually by users or are automatically generated by backend applications for production control. The System Configuration component provides the Config API that exposes all configuration services to external applications, for example, for user-defined commands, and interprets the user input and derives appropriate command message(s) to be sent to all relevant nodes. A key feature is the ability to select one or more data sources to be used as input to actuation decisions.

Dispatcher. Data is forwarded from a sink node to the Dispatcher software located on a host in the backbone, which then feeds data into the middleware. Likewise, data is transmitted via the Dispatcher into a WSN. The Dispatcher employs flow control mechanisms to avoid interfering with the strict TDMA schedule of the sink node. Furthermore, reliable communication is ensured by using Cycling Redundancy Checking (CRC) checksums, acknowledgements, and automatic retransmissions.

The current implementation uses a USB/serial connection between sink node and PC, whereas a GINSENG system in full production would integrate the functionality of the sink node and Dispatcher within an embedded system.

GINSENG Node Platform. Sensor and actuator nodes are implemented using the popular Contiki operating system. Each node uses a TDMA-based MAC protocol called GinMAC as the underlying communication protocol, which ensures timely and reliable data delivery. A number of additional modules required in a performance controlled network are tightly integrated with GinMAC. These components are topology control (GinTop), queue management (GinQueue), and performance debugging (GinPerf).

GinMAC. GinMAC is a single channel TDMA protocol which uses exclusive slots for data transmission. Neighbouring wireless clusters are required to use different transmission channels in order to prevent collisions. GinMAC provides redundant slots to compensate for potential packet losses. GinMAC accepts packets from the upper layers (e.g., GinApp and GinSink), which are queued and then transmitted by the radio at the appropriate time.

GinTop. A GINSENG system uses a tree topology envelope specified before network deployment for network organisation. GinTop is responsible for assigning nodes to a position in the logical network topology such that the resulting topology fits within the provisioned topology envelope.

GinQueue. GinQueue is called by GinMAC for packet scheduling decisions. Sensor data and actuator data is queued with high priority while other data, such as configuration data or performance debugging data is queued with low priority.

GinPerf. GinPerf is used to monitor the performance of the wireless networks. It is used to collect performance related information on nodes, to process this data, and to forward processing results and raw information to the middleware via the sink node, where it can be further analysed and combined with data from other WSNs.

GinApp. Each node in a WSN carries this component which is responsible for polling periodically sensor data and/or initiating actuation if required. GinApp can be configured (e.g., polling frequency, actuation parameters, etc.) using the Config API offered by the GinSysConf component. For example, sensor and actuator behaviour can be configured via a control centre using the Config API provided by the GinSysConf component.

GinFDS. The Fault Diagnosis and Supervision component is based on a multi-agent hierarchical architecture, located at node level, in order to guarantee the quality of the acquired data and fault tolerance in closed-loop control systems. Each agent is configurable and is responsible for specific tasks, such as monitoring sensors readings, outliers filtering, and handling communication faults (in the forward link) on the control system.

GinSink. The sink node in a wireless cluster carries this component which has two functions. First, GinSink is the bridge between wireless cluster and backbone infrastructure. GinSink is used to pass received messages from the wireless network to the backbone infrastructure. Second, GinSink can be used to implement control functionality. Incoming sensor messages can be processed, and messages with actuation commands can be generated as a result, which are then distributed to actuators within the wireless cluster. GinSink can be configured, for example, to set forwarding filters and control loop parameters.

3.2. The GINSENG Data Path

The GinApp process running on the sensor nodes reads and pre-processes sensor data and then decides if data must be forwarded. In addition, the GinFDS component may be used to perform outlier detection to ensure data quality (more details on outlier detection will be given in Section 5). All data is routed via the sink node in a WSN, as this is a property of the employed GinMAC protocol.

A task on the sink node then processes the data and may, as a result, generate commands to be sent to actuators within the same wireless cluster. Alternatively, the data may be forwarded by the sink node to a system in the backbone infrastructure. Data is forwarded from the sink node via USB/serial connection to the Dispatcher software which then forwards data to systems in the backbone. A combination is as well possible, whereby data is forwarded to the backbone infrastructure while command messages for actuators in the WSN are generated. In particular, the last option allows us to implement control loops with strict performance requirements, while monitoring messages travelling through the backbone infrastructure may have more relaxed performance requirements.

Middleware components in the backbone are also likely to process incoming data from sensors. In this case, input data from multiple WSNs and input from additional sources (e.g., process/risk management servers) can be used in decision making. Command messages for actuators located in WSNs that are closely linked to the middleware servers or for other secondary systems may be the result. Obviously, if control loops are constructed this way, data transport delays in the backbone infrastructure and the WSNs must be taken into account. Thus, this particular approach may not be suitable for very performance critical systems.

Sensor and actuator data have to be treated with priority. Other data, such as system maintenance messages, debugging information, and control messages, must be transported by the system without interacting negatively with application data delivery.

3.3. GINSENG Resource Provisioning

In a monitoring scenario, data has to be delivered within time T from the sensor nodes to the backend-application which presents the data. The time T can be broken down into the time T_W required for the WSN (reading sensor data, delivering the data upstream to the sink node and forwarding this data to the middleware) and time T_B required for data transport and processing in the middleware of the backend system.

The GINSENG system must be designed such that for each data source, the required deadline T can be met. Within each WSN the time T_W is determined by the TDMA schedule length F. GinMAC is designed such that it is ensured that data from all nodes can be delivered to the backend system, within this time frame. Inside the backend systems, the time T_B is determined by the software systems as well as the intermediate wired networks and the server hardware.

3.3.1. Backbone Infrastructure. As said, dimensioning of the backend system depends on a number of factors. Thus, a generic description of how to determine $T_{B_{max}}$ cannot be given, without assuming specific technologies for real-time networks in the backend and real-time operating systems on the servers.

More pragmatically, it is possible to put the backend systems through a load test to determine a worst-case delay for data transport and processing delays observed during the load test (see Section 5.2). While the numbers obtained in those tests are not strict delay bounds, we have observed that the maximum observed T_B is at least one order of magnitude lower than T_W in our system.

Therefore, we do not attempt to enforce strict real-time in the backend. Due to the significant differences in the order of the measured delay, we assume that the processing time in the backend T_B does not increase into the same order of magnitude as the time the WSN takes to transport the data into the backend system T_W for reasonable numbers of events per second. The assumption is verified by the measurements presented in Section 5.

Furthermore, if necessary, over-provisioning of both the backend network as well as the backend servers can serve to further distance the observed delays from the acceptable delays.

3.3.2. Wireless Sensor Network. A network dimensioning process is carried out before the network is deployed. The input for the dimensioning process are network and application characteristics that are known before deployment. The output of the dimensioning process is a TDMA schedule with frame length F that each node has to follow. The GinMAC protocol is detailed in Suriyachai et al. [2010]. However, to understand the evaluation presented in the next section, we give a brief summary of the protocol details here in this section.

The GinMAC TDMA frame consists of three types of slots: *basic slots*, *additional slots*, and *unused slots*. First, the frame contains a number of *basic slots* which are selected such that within frame length F, each sensor can forward one message to the sink and the sink can transmit one message to each actuator. Second, the GinMAC frame uses *additional slots* to improve transmission reliability. Finally, the frame may contain *unused slots* which are purely used to improve the duty cycle of nodes.

These types of slots within the GinMAC frame must be designed such that the delay, reliability, and energy consumption requirements are met. However, it may not always be possible to find a frame that simultaneously fulfils all three requirements. If that is the case, some dimensioning assumptions must be relaxed.

To determine the number of *basic slots* required in a GinMAC frame, a topology envelope is assumed. This topology envelope is specified as a tree rooted at the sink and described by the parameters: maximum hop distance H and fan-out degrees $O_h (0 \leq h \leq H)$ at each tree level h; we define $O_0 = 1$. The topology envelope can accommodate a maximum number of $N^{max} = \sum_{n=1}^{H} \prod_{m=1}^{n} O_m$ nodes. However, in the actual deployment a number of nodes $N \leq N^{max}$ may be used. Nodes in the later deployment can take any place in the network and even move their topological location as long as the resulting deployed topology stays within this topology envelope, of course subject to the requirement that the node's application message demands are consistent with those assumed for dimensioning purposes. The maximum number of sensor nodes N_S^{max} and actuator nodes N_A^{max} (with $N^{max} = N_S^{max} + N_A^{max}$) must also be known.

To determine the number of *additional slots* needed for reliability control, the worst-case link characteristics in the deployment area must be known. As the network is

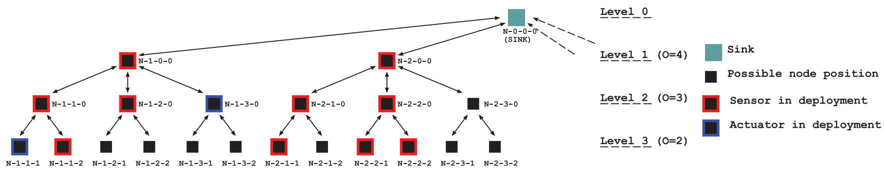


Fig. 5. Example topology with $N_A = 2$ actuators and $N_S = 10$ sensors.

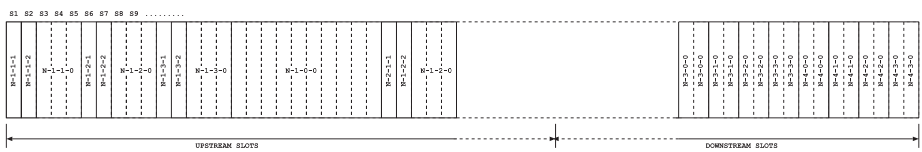


Fig. 6. Transmission slot allocation for the topology shown in Figure 5.

deployed in a known environment, it is possible to determine this value by measurement. The configuration of *basic* and *additional slots* determines an energy consumption baseline of nodes. Adding *unused slots* within the GinMAC frame can improve upon this baseline.

The allocation of basic slots for the example given in Figure 5 is depicted in Figure 6. The total number of slots in F needed to forward data to the sink in the example topology is $S_B^{up} = 100$ (see [Suriyachai et al. 2010] for generic formulae to determine required slot numbers in generic topologies). The required number of downstream slots in the topology shown in Figure 5 where there is a maximum of $N_A^{max} = 2$ actuators in the network is $S_B^{down} = 34$.

To determine the number of additional slots, we first need to choose a worst-case link reliability that GinMAC will support. The deployed system will use a topology that fits into the topology envelope and uses only links with reliability better than the selected worst-case link reliability. These links are called *good links* and are determined before deployment. A good link is defined by burst lengths. A good link must not have more than B_{max} consecutive transmission errors and must provide at least B_{min} consecutive successful transmissions between two bursts [Munir et al. 2010].

In a scenario where good links can be characterised with short B_{max} and long B_{min}, it is possible to efficiently add additional retransmission slots on the same link to deal with losses. Consider node N-1-1-0 in the example shown in Figure 5, $B_{max} = 2$ and $B_{min} = 2$. The node requires three basic slots for upstream transmissions, and in a worst case, any two of the three transmissions might be lost. However, if four additional transmission slots are allocated, all three packets are guaranteed to be delivered within the seven slots provided that the channel conforms to chosen B_{max} and B_{min}.

4. GINSENG DEPLOYMENT

Deploying a wireless sensor network in an operational oil refinery poses a number of challenges that serve to distinguish the solution from more benign contexts. In this section, we describe our deployment and explain key challenges we encountered when establishing the deployment used for scientific evaluations.

4.1. System Setup

The GINSENG deployment comprises a complete end-to-end solution for production control and automation based on wireless sensor networks. Therefore, the deployment also includes an advanced middleware backend as well as frontend application. Figure 7 depicts the components of the deployed scenario.

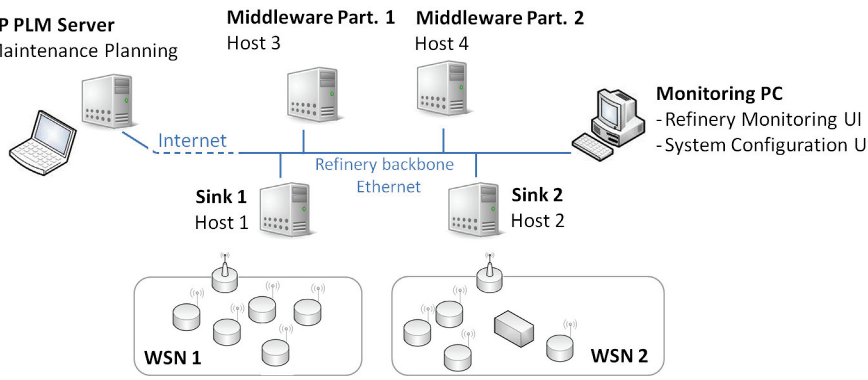


Fig. 7. The GINSENG deployment setup also includes middleware, backend, and frontend applications.

We have deployed two adjacent wireless networks, Network 1 and Network 2, in the refinery, comprised of 12 nodes each. Our sensor nodes are compatible with the well-known TelosB design. The networks differ in topology, physical layout, connected sensors, and the operational network channel. Our primary motivation for deploying two networks is twofold: first, to demonstrate and evaluate the integration and interoperability of different networks; second, to demonstrate and evaluate closed-loop control between different networks, where the sensor is in one network and the actuator is part of another network. The relatively small size of the networks is determined in consideration of the latency bounds required in the application scenarios. GINSENG sensor networks are limited in scale by virtue of having to assure message latency and thus having to bound the number transmissions on the path from each node to the sink. In the GINSENG approach, scaling to larger deployments is achieved by linking together multiple sensor networks using the GINSENG middleware.

Figure 7 depicts that we have also deployed four server hosts in the GINSENG control room in a portable office located in the heart of the water treatment zone (a restricted-access area) and a monitoring PC at the refinery control room in a building outside the water treatment zone. We have deployed two sinks, one for Network 1 (Host1) and one for Network 2 (Host2). The GINSENG middleware is distributed on two hosts (Host3 and Host4) to validate its scalability for large application scenarios. The PC at the control room runs two applications: first, the Refinery Monitoring application that compares GINSENG's performance with the performance of the existing wired analog system; second, the System Configuration front-end to send user-defined commands to the WSNs. All machines run Linux and are connected to the refinery backbone via IEEE 802.3 Ethernet Cat6e STP cables.

Finally, we deployed a application that interfaces with a SAP Product Lifecycle Management (PLM) server that enables predictive maintenance planning of WSNs and other refinery equipment. The machine hosting the PLM service host is located at the SAP Research Center in Dresden, Germany, thus demonstrating interoperability with remote and complex backend applications.

4.2. Middleware and Application Deployment

Figure 8 illustrates the deployment of the GINSENG middleware and connected applications in more detail. The Dispatcher software installed at Host1 and Host2 processes packets from the sensor networks that arrive at Sink1 and Sink2 and forwards them via a TCP/IP connection towards the GINSENG middleware. Furthermore, it transfers

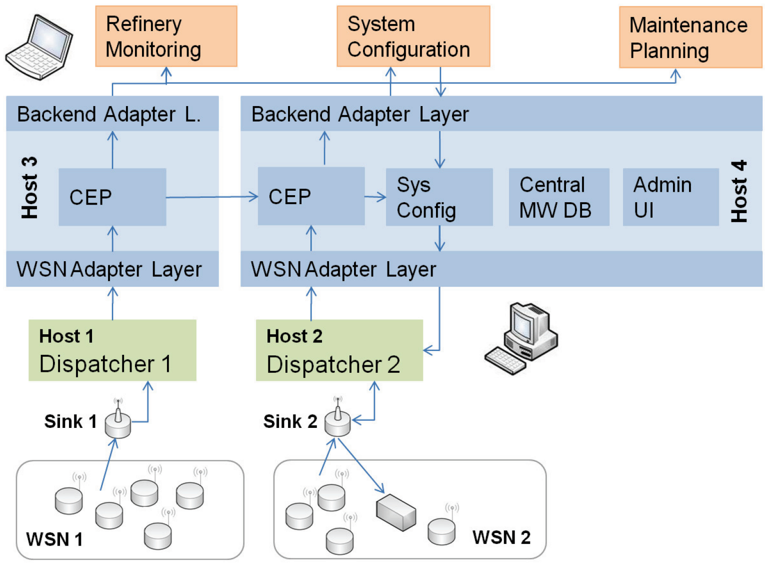


Fig. 8. Distribution of GINSENG middleware in the deployment.

system configuration commands that emanate in the middleware to individual sensor nodes.

As stated in Section 3, the middleware was designed for a distributed deployment scaling up to large numbers of connected data sources and backend applications, and hence requires high data rates at high reliability. To validate the distributed design, we deployed the middleware at two servers, Host3 and Host4. Host4 comprises the central middleware database that stores all related configuration information. This includes the number of deployed agents and connection parameters, the admin user-interface to manage these configurations, the system configuration component to derive commands, for example, for cross-WSN actuation, and the system integration component connected to Host2, (i.e., Network 2). Host3 runs the second system integration partition that connects to Network 1.

In both, system integration partitions there are key elements: (i) the WSN adapter layer for downwards connectivity, (ii) the backend adapter layer with various adapter agents to connect different back-end applications (like the Refinery Monitoring application in the refinery control room or the SAP PLM server for maintenance planning), and (iii) a complex event processing (CEP) agent to aggregate and analyse incoming sensor data.

4.3. Network Design and Topology

Following from the discussion of network dimensioning, a hierarchical topology is a natural choice for achieving scalability. The physical constraints of the test bed were the dominating factor in determining the actual topology chosen. While Network 1 follows a 3-3 hierarchical topology, the Network 2 is based on a 3-1-2 configuration. Figure 9 illustrates the physical topology of both networks relative to the GINSENG control room that is located between the two networks. Figure 10 and Figure 11 detail the logical topology of each network individually. Both networks also include actuators that control valves.

The choice of sensor nodes to be attached to transducers and to actuators are dependent on the location of the measurement points and control equipment (see Figure 9).

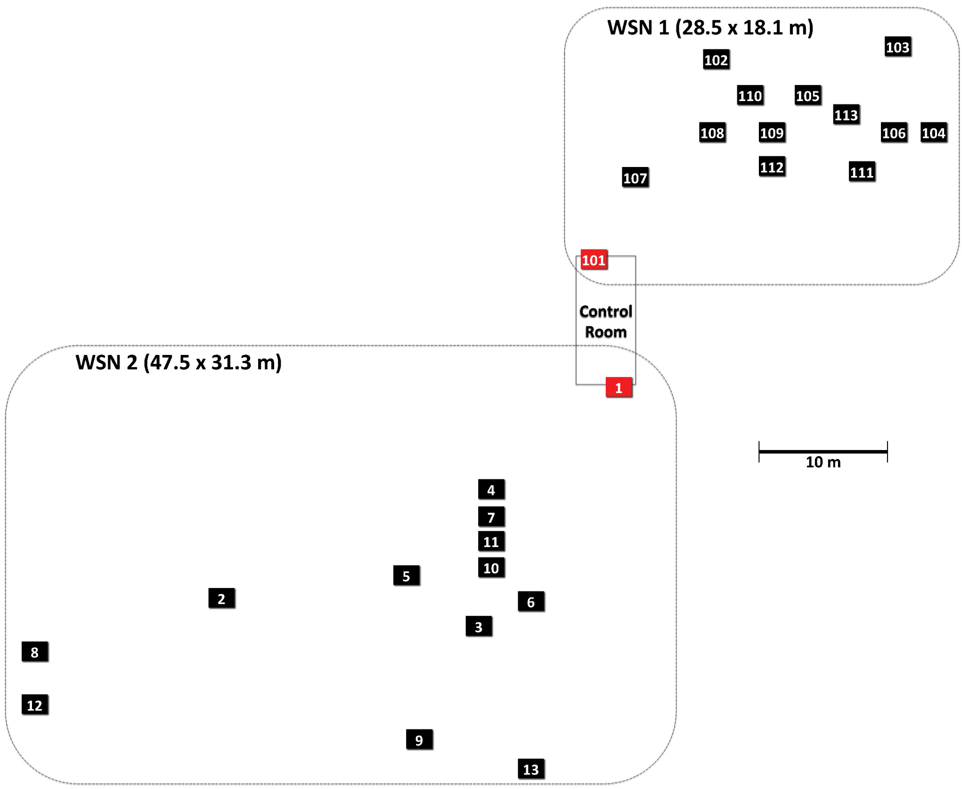


Fig. 9. Physical topology of the networks.

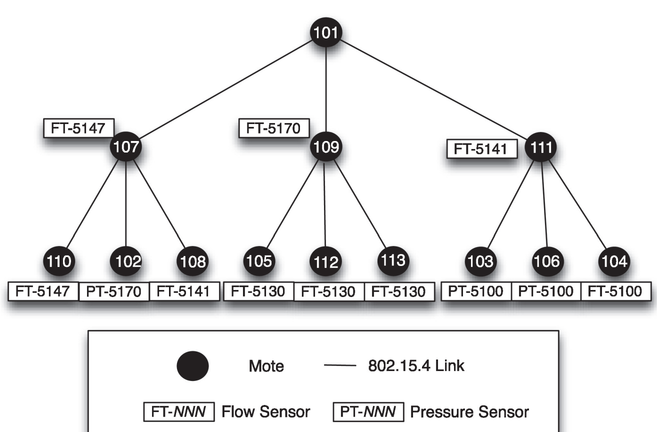


Fig. 10. Logical topology of network 1.

The hierarchical topology and the additional relaying nodes were carefully chosen to obtain good network performance. The process followed a methodical empirical approach based on visual inspection of each node's location and measurements of radio communication between nodes.

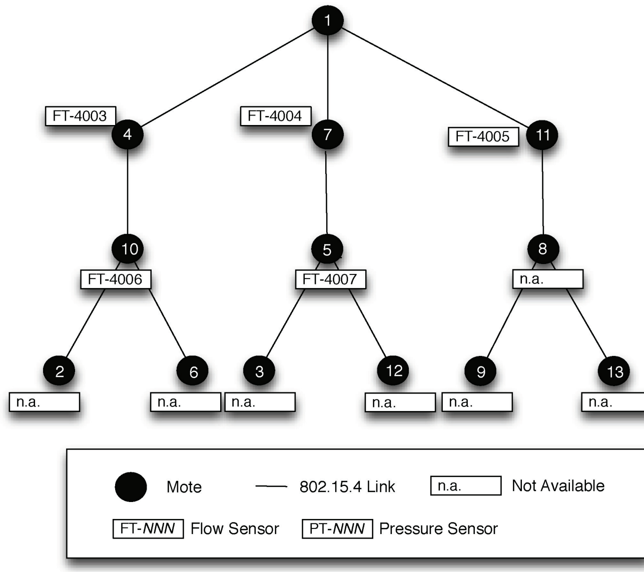


Fig. 11. Logical topology of network 2.

As a logical tree, a balanced tree was chosen in which we can use $B_{max} = B_{min} = 1$, which has one retransmission slot per message and enables a schedule that meets the desired delivery times (more details on dimensioning and schedule construction are provided in the next section).

The node identifiers in the figure (also used later in explaining the logical topology) are unique to each node.

4.4. Deployment-Specific Challenges

The refinery environment is highly challenging with respect to wireless communication. Huge metal structures and machines operating nonstop cause a high noise level, which might seriously affect the system performance. In general, however, the radio environment, while noisy, was observed to usually be fairly stable, with reliability levels on individual links lying within well-defined ranges.

The deployment of a network in a sensitive environment, such as an oil refinery, poses many new challenges. Like other heavy industrial sites, the refinery has very strict rules regarding the movement and management of personnel within certain areas. Several areas are classified as ATmosphère EXplosive (ATEX) areas, restricting access by personnel and requiring that electrical equipment, including sensor nodes, be encased within ATEX-certified enclosures.

The use of ATEX enclosures raised concerns regarding an expected impact on wireless signal attenuation. Our previous research had demonstrated the benefits of using an antenna that is located outside an ATEX enclosure [Boano et al. 2009]. In fact, this was essential because the specification of the ATEX enclosures necessary in the refinery was such that all wireless communication using the on-board (internal) antenna on the TelosB nodes was blocked. We first tried relatively low-cost external antennas with 1dB gain. These antennas enabled communication but only at very low packet delivery rates. In addition, the initial 1dB antenna, a standard WLAN external antenna, had corroded to an alarming degree within only two weeks at the refinery due to the operating environment. As a result, we studied alternative antennas, concluding that the best option would be a ceramic white 45.72cm antenna, with 9 dB of gain, capable of assuring maximum performance, quality, and durability. It was not just a question of power but also a question of resistance and durability in these corrosive environments. While operating with these antennas in the refinery, we realised that although the clamps became corroded, the ceramic antenna has never shown any problem. The 9 dB of gain were also a key point in this choice, allowing extended transmission range [Raman and Chebrolu 2008].

Naturally, these resulted in an additional per-node cost—approximately €125 for each ATEX enclosure and €220 for each antenna. In different settings, the deployment constraints will differ and the features and costs of the required enclosures and antennae may be higher or lower than these values. For example, low-end ATEX enclosures are available for under €10, and low-end external antennae for tens of Euro.

Figure 12 shows some of the sensor locations, the portable office with the GINSENG control room, as well as one of the deployed nodes within the ATEX enclosure and an attached 9dB antenna.

In the next section, we show some experimental results that demonstrate the importance of channel selection. Those experiments revealed that the selection of an appropriate IEEE 802.15.4 channel significantly increased packet delivery rates. A deeper analysis on coverage and radio interference in the refinery can be found in Tran et al. [2011].

4.5. Channel Selection

In order to achieve reliable and energy-efficient communication, we conducted some preliminary experiments to demonstrate the importance of selecting appropriate IEEE 802.15.4 channels. In the experiment, a node sends 200 packets in one hop to the sink node with an interpacket transmission time of 200 ms. The sink extracts the received signal strength indicator (Receiver Signal Strength Indicator (RSSI)) and the message ID to calculate the Packet Reception Rate (Packet Reception Rate (PRR)) measured from source to sink without retries. We performed measurements with both types of external antennae, that is, with gain of 1 dB and 9 dB, respectively. As expected, the 9dB antenna performed significantly better, so we omit the results for the 1dB antenna.

Our results are shown in Figure 13. Channels 13 to 18 exhibit good results with almost no packet losses and an RSSI between −60 and −63 dBm. Channels 25 and 26 also show a good PRR but with a lower RSSI. These results serve to emphasise the need to conduct experiments at the site to avoid choosing a bad channel.

Figure 13 shows that channel 21 would lead to significantly worse performance than the other channels and should be avoided. The reason for this behaviour is technically unknown. We can only hypothesise that some equipment at the refinery was generating interference at this particular frequency in that area of the plant. From our field experience, we have learnt that in the refinery channels, performance changes from area to area depending on the structures and operations running locally, which cause different types of noise and radio interference. Hence, for each deployment, we concluded that not only a previous spectrum analysis is needed, but also that it must be repeated periodically during the network lifetime as part of a maintenance regime in order to guarantee that the spectrum interference has not changed.

5. EVALUATION

In this section, we present and analyse results obtained from the evaluation of the GINSENG system in the Petrogal refinery at Sines. As GINSENG provides a complete solution for wireless monitoring and control, a large number of experiments were performed to capture the performance of the network and of other system components. Focused, short-duration experiments, were designed to evaluate individual components

Fig. 12. Sensor nodes deployed in the refinery.

of the system. The results were consistently positive, with targets for delay and reliability being easily achieved, and demonstrating that the network works properly and is well provisioned during this time. To capture the general characteristics of the system, a longer-term experiment was running continuously for a two week period.

Given the focus of the research, the key metrics of interest are delay and reliability of message delivery. The performance of the wireless sensor networks is investigated in detail, as it is the dominant factor in determining overall system performance. However, the performance of the backend infrastructure is analysed as well, including its scalability. System performance in the context of monitoring-only applications and also applications with automated control loops are investigated. Finally, a comparison of data obtained from the installed cabled monitoring system with data obtained through GINSENG is carried out to show that GINSENG is a valid alternative to the existing cabled system.

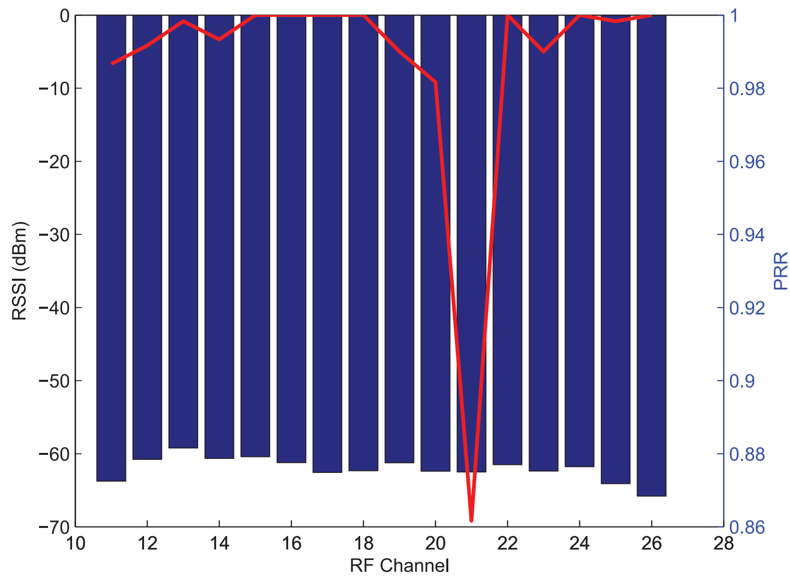


Fig. 13. Performance of the 802.15.4 channels.

5.1. System Setup for Evaluation

The network setup, as described in the previous section, was used for evaluation. For the two WSNs (Network 1 and Network 2), channels 16 and 14, respectively, were selected, as they provided a relatively clean communication environment, as explained in Section 4.5. Then, for both networks, a topology was selected that allowed us to employ a GinMAC schedule with $B_{max} = 1$ and $B_{min} = 1$ to support end-to-end reliability of 99%. These resulting logical topologies were shown in Figures 10 and 11, while the physical topologies can be seen in Figures 14 and 15. In GINSENG, it is always desirable to identify topologies that require few retransmission slots in order to minimise end-to-end delivery latencies. Other topologies are possible of course, but may require a higher number of retransmission slots within the schedule. Unlike the node identifiers used in the previous section that served simply to uniquely distinguish nodes, the node identifiers used in this section are topological, explicitly giving the location within the network tree in terms of branch and level, and thus facilitating an easier interpretation of the results.

In both networks, nodes are programmed to generate application data messages once every second. A subset of nodes in each network are configured to work as actuators as well as sensors. A minimum GinMAC schedule was devised for both networks which provides the necessary time for transmission of upstream and downstream messages and maintenance messages, such as for time synchronisation. The schedule also allows sufficient time for application processing—in these slots no communication takes place and processing cycles are ensured to be available at every node for application tasks.

Network 1 contains 12 nodes along with a sink node that is connected to a sink PC. It is a 3-3 network, as shown in Figure 10, with each child of the sink having three children of its own. There are three branches.

—*Branch* 1 consists of nodes 1-0, 1-1, 1-2, and 1-3 with node 1-0 forwarding messages from nodes 1-1, 1-2, and 1-3 to the sink, 0-0.
—*Branch* 2 consists of nodes 2-0, 2-1, 2-2, and 2-3 with node 2-0 forwarding messages from nodes 2-1, 2-2, and 2-3 to the sink, 0-0.

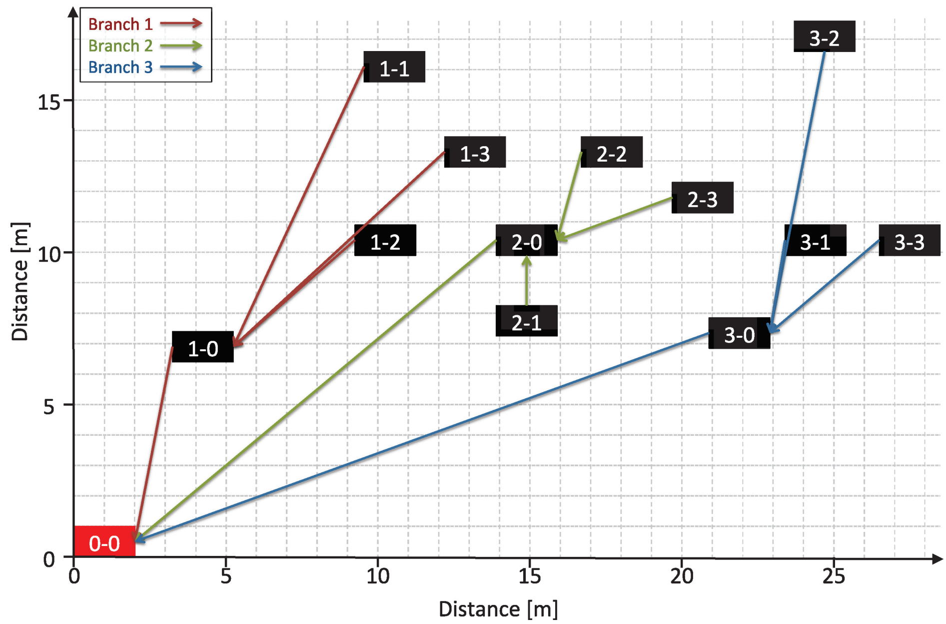


Fig. 14. Physical Topology of Network 1.

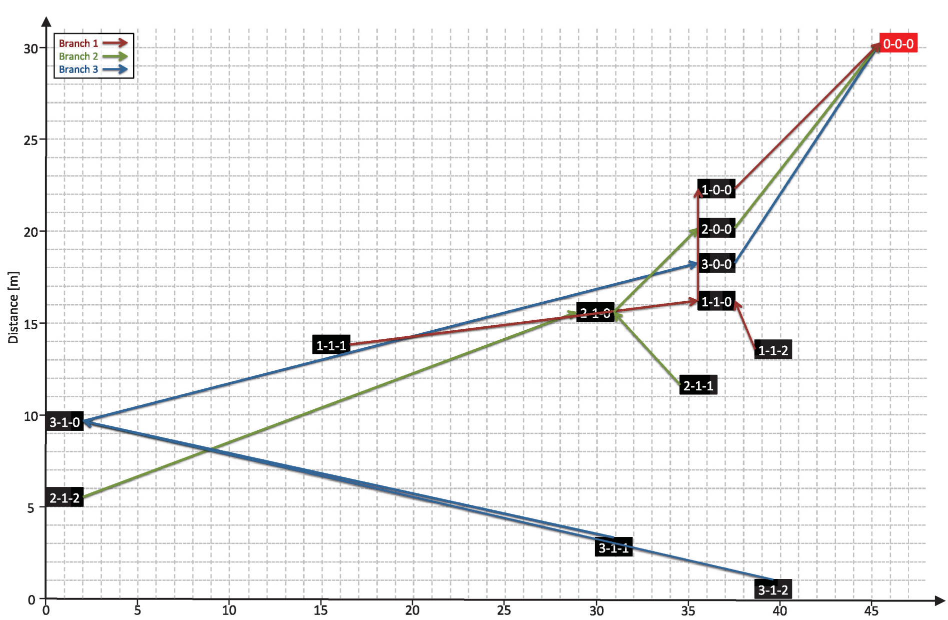


Fig. 15. Physical Topology of Network 2.

—*Branch* 3 consists of nodes 3-0, 3-1, 3-2, and 3-3 with node 3-0 forwarding messages from nodes 3-1, 3-2, and 3-3 to the sink, 0-0.

In Network 1, it was convenient to use a single node, Node 1-0, as both a sensor and an actuator. Sensor data generated by Node 1-0 is transmitted via the network and delivered as input for actuation, just as it would if the sensor and actuating functions were on different physical nodes.

To support the communication requirements, a GinMAC schedule employing 92 slots in one TDMA epoch is used in Network 1. Each slot has a duration of 10 ms leading to an epoch duration of 920 ms. The slot allocation does not follow the strict layout given in Section 3 in order to allow for optimisation for the control loop. The first two slots in the epoch are reserved for application processing. In these slots, all nodes generate sensor data that has to be transported. Next are two dedicated upstream slots for node 1-0 to transmit the sensor data necessary as input for the control loop. Thereafter, 16 slots for sensor data from nodes in Branch 1 are provisioned. Next are two unicast downstream slots for actuator messages directed to node 1-0. Thereafter, four slots for broadcast downstream traffic are provided. These slots can be used to carry sensor node configuration commands (e.g., to switch sensing on or off). These are followed by four slots used for time synchronisation of the network. Then 28 upstream slots for sensor data from Branch 2 and Branch 3 are provided. Finally, 34 processing slots are provided, which is the time in which no communication takes place but uninterrupted application processing can be carried out. This time is particularly important for the sink node, which uses this time for data forwarding over the serial port without having to also be prepared to handle newly arriving packets.

With this configuration, all sensor data generated at the start of the TDMA Epoch is guaranteed to be transported to the sink and forwarded to the sink PC within 920 ms. Even if messages must be retransmitted, this time bound will not be violated as long as link errors are not worse than the assumed characteristic of $B_{max} = 1$ and $B_{min} = 1$ as used in our testbed.

Network 2 uses a different TDMA schedule configuration. For Network 2,100 slots are provisioned which allows data transport within a delay bound of 1 s. In Network, a 3-1-2 topology with three branches are used as shown in Figure 11. In this network, node 3-1-0 is used as an actuator.

As described in Section 2, monitoring requires an overall data transport delay of 3 s, which can be easily provided with the given configuration. The wireless sensor network contributes at most 1 s and, as we show next, the contribution of the middleware is an order of magnitude below the contribution of the sensor network.

Closed-loop control can also be supported, as the transport of sensor data upstream and subsequent transport of an actuation message downstream can be completed in one TDMA epoch which is far below the 3s requirement. Even if sensor and actuator are located in different wireless sensor networks, these requirements can be met considering these network epochs.

5.2. Overall System and Middleware Performance

Our initial focus was to assess the performance of the overall system and in particular the middleware. As we can see in Figure 17(a) the delays in the backend parts, for example, in the Dispatcher, Ethernet, and Middleware components, are very small indeed, and are insignificant relative the sensor network delays. Further, the times needed within all these backend components are very small, hence, we can concentrate on the WSN part for more detailed delay evaluations in the next section.

The types of networks deployed and the applications running over them were not considered challenging enough for the middleware subsystem in terms of event generation capabilities. In other applications settings, the demand might be significantly higher, especially as one scales GINSENG by adding many more sensor networks connected via many sink nodes to one backend system. To provide the necessary stress conditions and prove the scalability of the GINSENG middleware, we artificially generated sensor event messages with a high data rate and repeated the performance

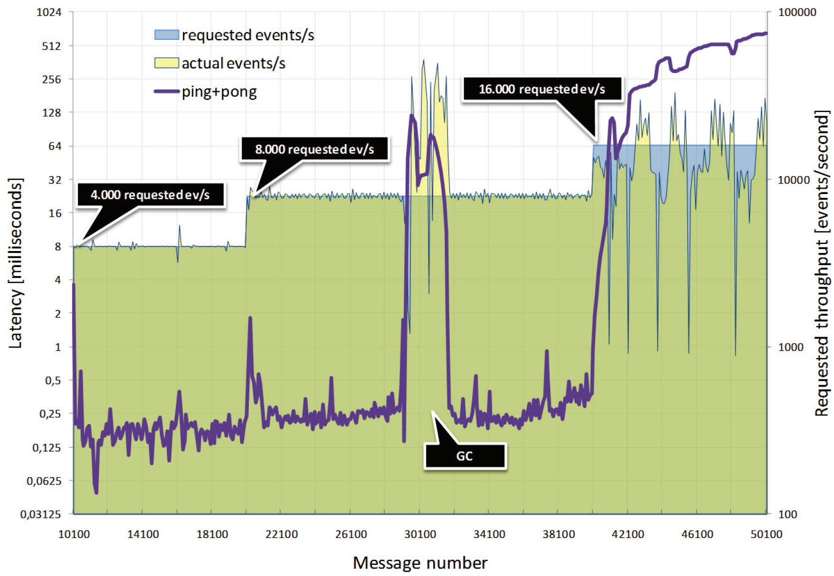


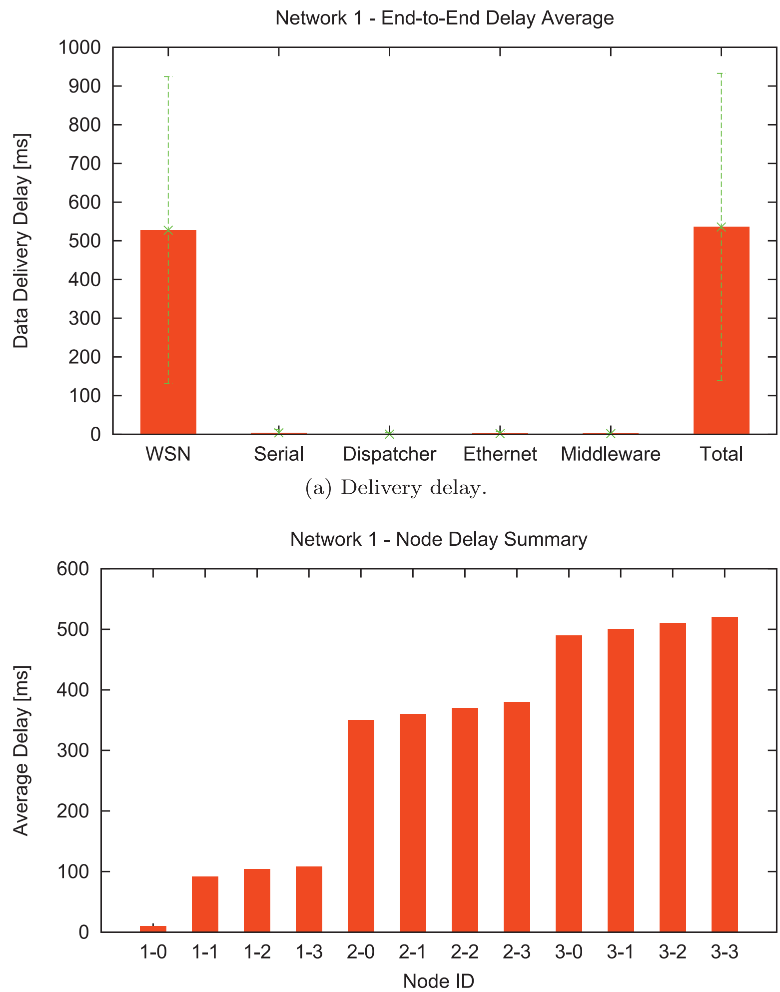
Fig. 16. Middleware scalability, up to 8,800 events/second.

evaluation with an otherwise identical experimental configuration. We then plotted the measured middleware latency versus time as we increased the sensor events in a series of steps. As shown in Figure 16, one middleware partition is able to process up to 8,800 sensor events per second without significant performance decrease. In this figure, the *ping+pong* curve shows the delay between the entry and exit agents within the middleware. The background execution of garbage collection causes a brief increase in delay, but even so, the delay total remains very low in absolute terms. Of course the upper bound on latency identified in the testbed depends on the memory capacity and CPU power of the middleware computers. For even higher scalability, the middleware design can be distributed across multiple machines, each responsible for different WSNs, and only interacting in cases where cross-WSN activity needs to be handled.

5.3. Sensor Network Performance

As shown, the end-to-end delays are dominated by the sensor network transport, and thus we now turn our attention to evaluating network performance in regard to the ability to meet delay and reliability targets. Figure 17(b) presents summary results for sensor network delays for Network 1, where it is evident that the system is operating deterministically and well within the required delay bounds. For Network 2, the results of our measurements show a similar behaviour, as shown in Figures 18(a) and 18(b), along with the per-node distribution among the system components (WSN, serial, Dispatcher, Ethernet, middleware) for these delays in Figure 18(c). We also can see here that the delays in the backend components are very small for all nodes.

Turning to reliability, Network 2 operated within the established bounds. For Network 1, our measurements show that at the beginning, the WSN is operating as designed and within performance bounds, but that after a few days, some problems were evident. The results are plotted in Figures 19, 20, and 21, presenting measurements of loss for each of the three network branches. In each case, the results are shown for per-node end-to-end losses, per-link losses, and per-node end-to-end burst losses. For Network 1, Branch 1, we see that nodes 1-0 and 1-3 experience very low losses, while nodes 1-1 and 1-2 clearly develop problems. This is also evident looking at the losses



(a) Delivery delay.

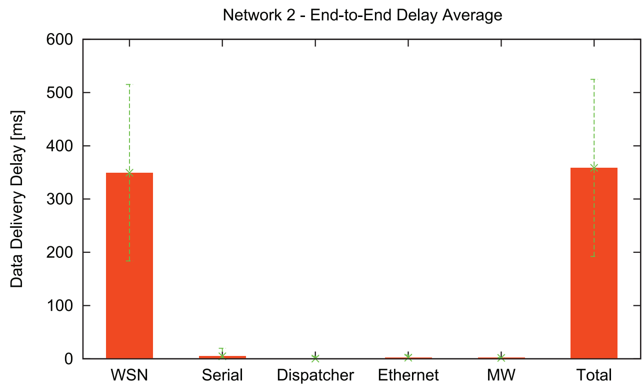
(b) Node delivery delay.

Fig. 17. Network 1 delivery delays. These are low and bounded, dominated by WSN delays.

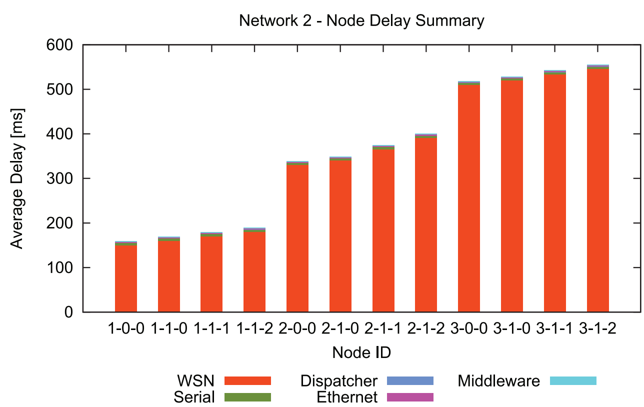
on a given link. In this case, the links from node 1-1 to node 1-0 and node 1-2 to node 1-0 are showing considerable losses. Also, the number of consecutive packets lost in each burst of losses serves to further illustrate these issues. This matter is discussed further in the context of topology adaptation.

For Network 1, Branch 2, we see that there are a small number of lost packets at the beginning (mostly well below 1%), but that losses are minimal once the network has completed its initial configuration stage and stabilises. The link loss ratio and the loss bursts show the same overall behaviour. The size of the loss bursts for Network 1, Branch 2 are very small, usually just comprising a single packet. Branch 3 in Network 1 is similar to Branch 2 in that there are just a small number of lost packets during the networks initial configuration stage and very few thereafter. End-to-end losses, per-link losses, and burst losses are all low.

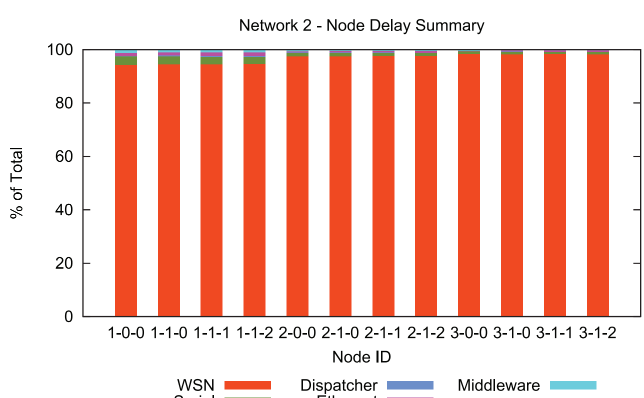
The reliability of Network 1 is summarised in Figure 22. As previously noted, nodes 1-1 and 1-2 were problematic, while the other nodes experienced losses well within operational limits.



(a) Delivery delay.



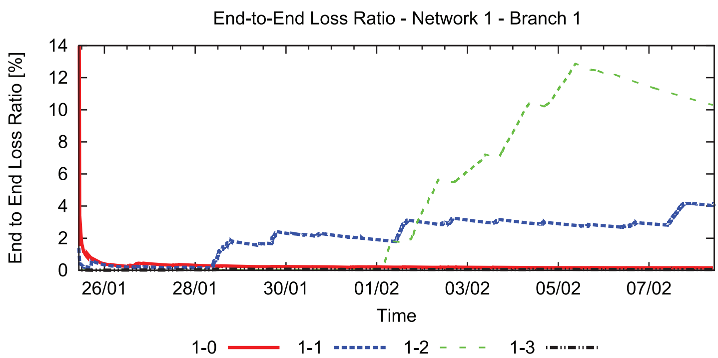
(b) Node delivery delay.



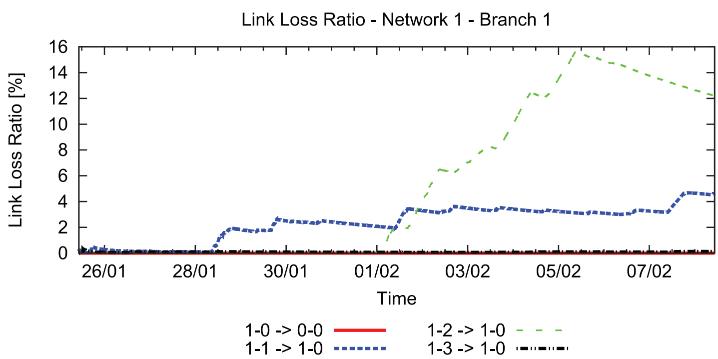
(c) Node delivery delay percentage.

Fig. 18. Network 2 delivery delays, which are low and bounded and well within scenario requirements.

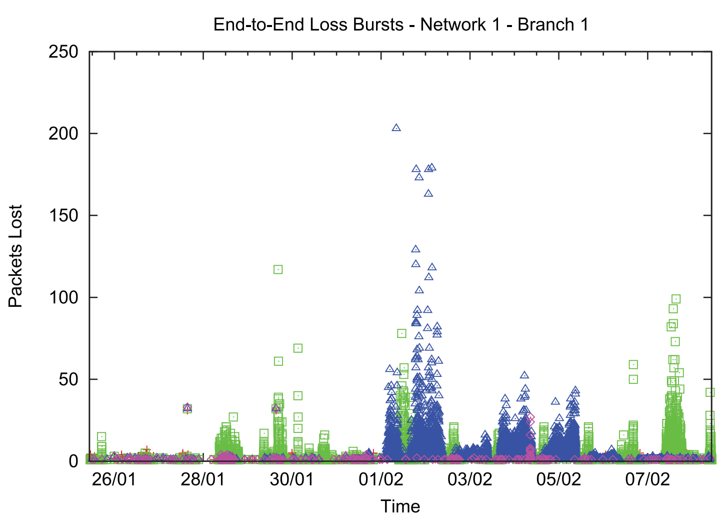
Figure 22 shows the power consumption of all nodes in Network 1. Node 0-0 (i.e., the sink) has the highest consumption, as it has to handle all network traffic, and is the only node spending any noticeable energy on processing. This is due to the fact that the sink has to forward data on the serial port to the Dispatcher. However, it is



(a) End-to-End loss ratio.

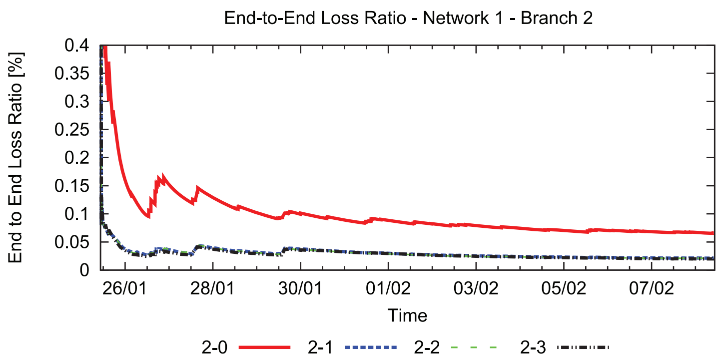


(b) Link loss ratio.

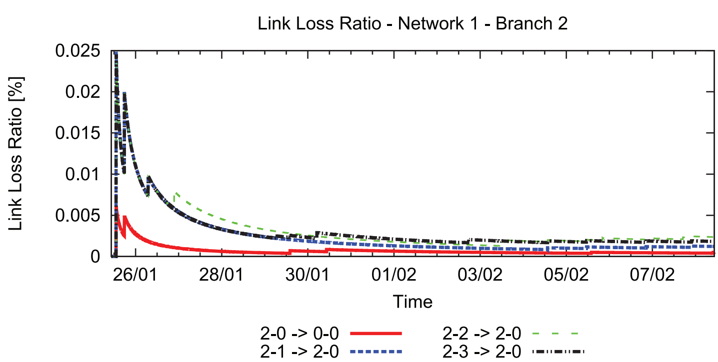


(c) Loss bursts.

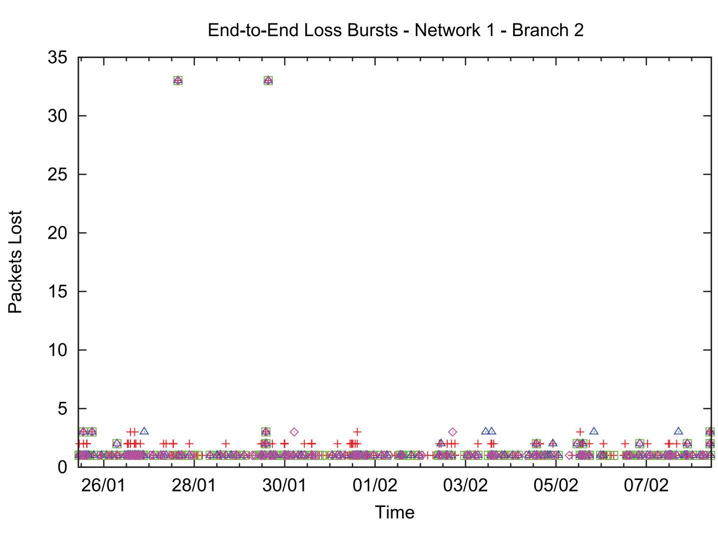
Fig. 19. Network 1, Branch 1, showing problematic links.



(a) End-to-End loss ratio.

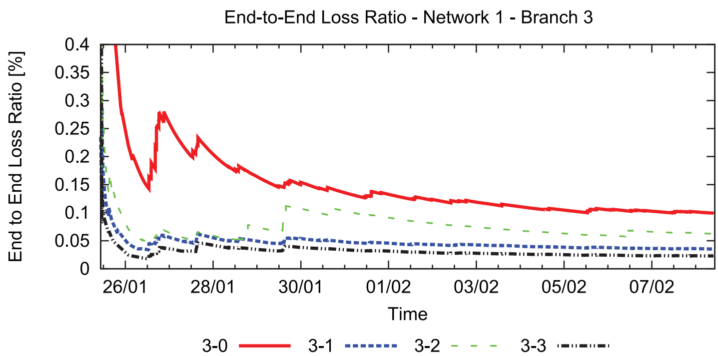


(b) Link loss ratio.

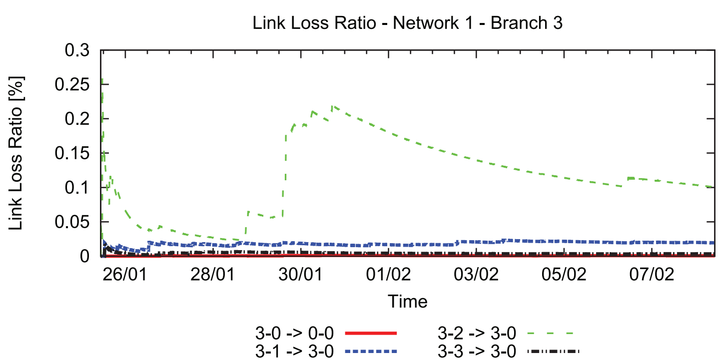


(c) Loss bursts.

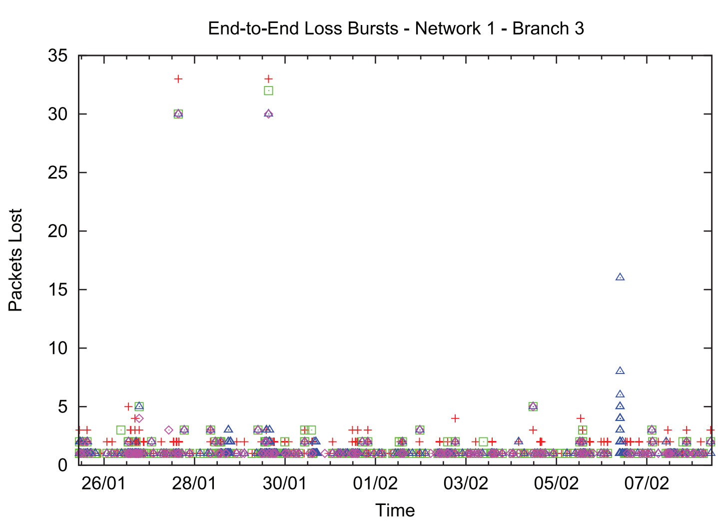
Fig. 20. Network 1, Branch 2, showing very low losses.



(a) End-to-End loss ratio.

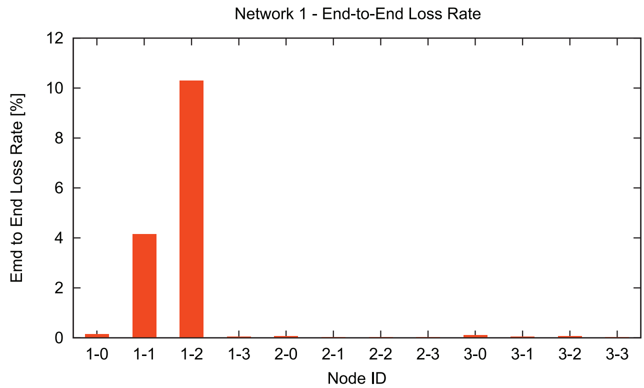


(b) Link loss ratio.

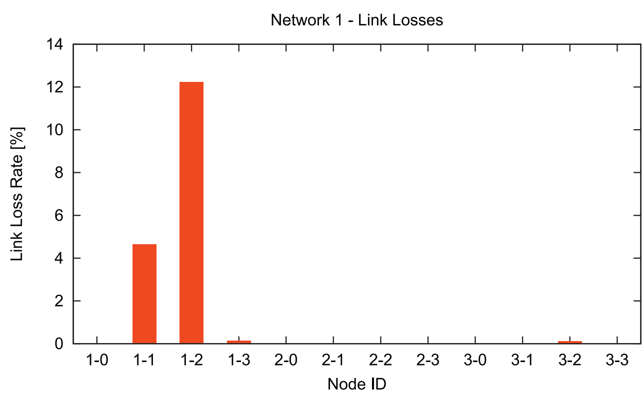


(c) Loss bursts.

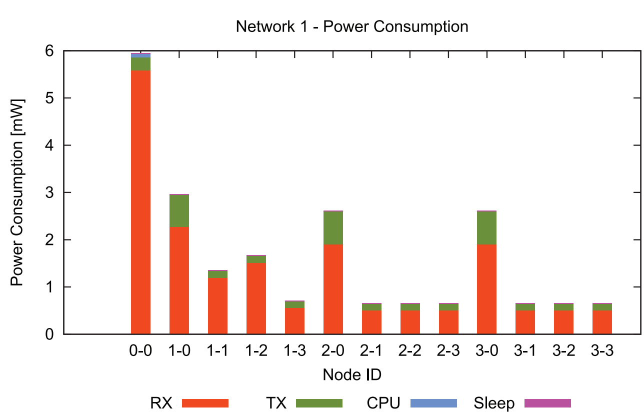
Fig. 21. Network 1, Branch 3, showing very low losses.



(a) End-to-end Loss Ratio.



(b) Link loss ratio.



(c) Power consumption.

Fig. 22. Network 1, loss summary and power consumption. Losses are generally low, except on problematic links in Branch 1. Power consumption yields node lifetimes well within requirements for the scenario.

Table I. Dynamic Tree Construction

Frequency (out of 12)	Connected nodes(out of 14)	Percentage of connected nodes	Average construction time(s)
9	14	100.00%	22.376
2	11	78.57%	16.260
1	10	71.43%	11.720

generally accepted in the literature that sink nodes would not normally rely on the same constrained power source as a regular sensor node, and thus the increased power consumption is not a significant factor. All nodes expend more energy on receiving than on transmitting. Recall that nodes listen in all potential receive slots for incoming messages. However, it is not always necessary to use retransmission slots, which explains the balance between energy consumption for transmission and reception. Nodes 1-1 and 1-2 encounter a high loss rate on the path towards node 1-0. Thus retransmission slots are often used in this situation, which explains the higher energy consumption of these nodes compared with node 1-3. In Branch 2 and Branch 3, there are lower loss levels and nodes having very similar energy consumption patterns (e.g., 2-1, 2-2, 2-3). If one excludes the sink node (node 0-0), node 1-0 has the shortest lifetime if run on batteries. To gauge the impact on lifetime, if one were to assume the use of four standard 3000mAh batteries, this node would have a life expectancy of 254 days, which is above the lifetime requirement, as outlined in Section 2.

5.4. Dynamic Topology Control

Reliability issues in both networks occurred at specific links and affected related branches. This unexpected behaviour was induced during normal operating conditions and without a specific visible or measurable related activity. The key consequence was that the basis on which the network had been provisioned was too conservative and did not anticipate this behaviour. Therefore, higher losses started to occur, which needed (i) the detection of these problems and (ii) an adaptive response. For (i), the performance debugging tools can effectively detect such problems. Regarding (ii), to deal with such changes in the environmental conditions of the system, one approach is to dynamically alter the logical network topology while still ensuring the performance bounds. The Topology Control module (GinTop) of the GINSENG architecture has many functions. The most important functions are the Dynamic Tree Construction, and the Topology Maintenance and Dynamic Control.

5.4.1. Dynamic Tree Construction. The Dynamic Tree Construction function self-organises the network, in a distributed and decentralised fashion, to create the best possible physical connectivity that conforms to the logical tree required by GinMAC. This function is one of the first to take place during the setup of the network. To evaluate the Dynamic Tree Construction functionality, a set of experiments were conducted inside the refinery area to gather information on the time taken to construct the tree topology and the number of nodes that have been attached. In this set of experiments, we were trying to allocate 14 nodes in a 3-2-1 logical tree structure (with 16 available positions). The test bed was operated for 24 hours, and during this time, we rebuilt the tree 12 times in order to obtain values from different tree-based topologies.

The results show that our solution dynamically connected all 14 nodes in 9 out of 12 tests. Based on this ratio, we verify the need of the topology maintenance and optimisation mechanism to guarantee the connection of all the nodes (see the next section). The results of the tests are summarised in Table I.

Table I also shows the average tree construction time, which is about 22.3 seconds (for a full tree case). This value corresponds to the time between the first advertisement from the sink until the joining of the last node to the tree (based on the reception of a Join Ack message). We consider that this is a satisfactory time interval for the tree self-organisation. This time is related to the epoch duration, which depends on the number of required slots, which in turn are based on the tree size. Therefore, a different tree structure will have a different construction time. The construction time can also be reduced if the number of slots per epoch is further optimised.

5.4.2. Topology Maintenance and Dynamic Control. Topology maintenance refers to the control messages and the actions taken to enable nodes in the network to recognise that they, or their parents or their children, have been disconnected from the tree and advertise or seek connections accordingly. It becomes useful during maintenance operations (e.g., battery replacement) in the sense that a node can be removed and re-introduced in the network and it can automatically find a free position for association. Dynamic Topology Control (DTC) refers to the signaling exchanged and the actions taken to enable the network to adapt to adverse conditions, such as those described in Section 5.3.

As noted, the majority of the links are stable with only few losses, while some links report an increased number of losses. An example of such link is the link from node 1-2 to 1-0, as shown in Figure 19(a). In order to address this phenomenon, we would need to dynamically reattach the lowest-level node (1-2) to a new tree position. To be able to do that, we would have to be in a network with free tree positions to move a node with problematic link quality. Sensor nodes have only local information, whereas the end-to-end loss rate is calculated at the backend system. Therefore, the trigger for searching a new attachment point must be based on metrics that are locally available. Such metrics are the link loss and the RSSI. The suitability of the RSSI was proven in Srinivasan and Levis [2006].

Due to the signal fluctuations that affect the RSSI value and the retransmission ability of the system, a combination of both metrics can give a better decision about the triggering initiation. Since the system is not linear but dynamic, we decided to use fuzzy logic for the adaptation control, as it provides the characteristic of flexibility to modify the design easily and the advantage that it can be built on top of the experience of experts and needs no training and learning procedures, like other solutions, such as neural networks. Using linguistic rules that describe the behaviour of the environment in widely differing operating conditions, the proposed fuzzy logic controller dynamically calculates the decision probability (to trigger the decision whether a sensor node has to attach to a new position or not), based on two network state inputs: the instantaneous value of the RSSI and the link loss rate.

In order to test if our solution managed to recognise bad links and to successfully reattach the problematic node to a new tree position, we run a number of short-term experiments using the refinery testbed. We have performed a set of experiments with a 3-2-1 tree (16 node positions) and 13 nodes. Figure 23 shows a representative example of the results. We observe that the link quality is sufficient to achieve 0% packet losses at the backend system until epoch 190, at which point an event caused the end-to-end loss to increase. The fuzzy controller operated and triggered the search of a new attachment point in three different instances (epochs 190, 230, and 260). Such a search may not result in a reattachment, either because no attachment point exists in the node's vicinity or because the possible new attachment points do not have performance qualities that satisfy the controller's requirements. In this particular experiment, the node managed to reattach two times during the specific experiment. The first search did not result in a reattachment due to the inability to find a better connection point. The second search resulted in a change (indicated in Figure 23 as reattach 1). After reattachment 1, we observed that the end-to-end packet loss decreased, but after a few epochs, it increased again. Then, reattachment 2 occurred, and the end-to-end packet

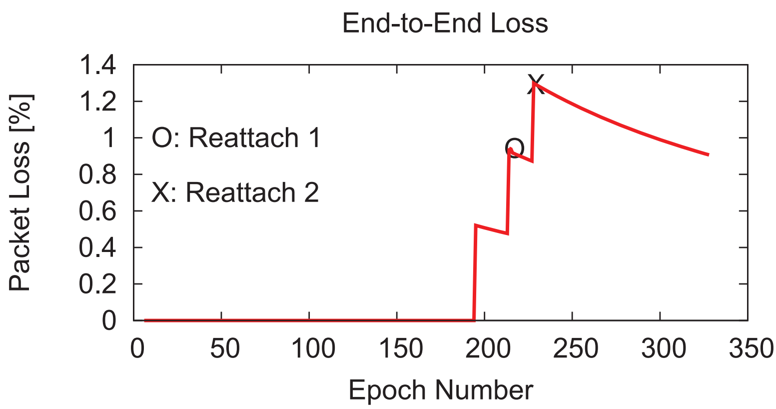


Fig. 23. DTC-triggered link change.

loss was decreasing again until the end of the experiment. Observing this behaviour, we can reach the following conclusions. First, the search for a new position is not always successful and depends on the available free positions in the logical topology. Second, the reattachment to a new position cannot guarantee that the new position can provide better performance for the rest of the connection time (as it happened with reattachment 1). Finally, the fuzzy logic controller successfully manages to control the increased end-to-end packet losses that may occur during the operation of a network.

5.5. Closed-Loop Control

For closed-loop control, several different configurations can be implemented depending on the selected location for where the control loop should be closed. When the actuator and the sensor are the same node, it is possible to implement local control. However, in most application cases, sensing and actuation would be implemented at different nodes due to constraints on the physical locations in a real deployment. For these cases GINSENG provides the option of closing the control loop at either (1) the sink node: sensor data travels to the sink node where decisions are made and actuation commands are issued; or (2) the middleware: decisions are made in the backend infrastructure, where actuation commands are issued.

Closing the control loop at the sink node enables bounded delays, as communication into the backbone infrastructure is not necessary and sensor network performance is deterministic. Closing the control loop at the middleware has the advantage of using sensors and actuators in different wireless networks but reaction times will be slower. The use of middleware also enables more sophisticated decision making for application scenarios where that may be useful.

5.5.1. Closed-Loop inside WSN through Sink Node. Figure 24 shows the latency values for the control loop in Network 1 that is closed within the sink node. In this case, the closed loop through the sink involved a sensor node sending its sensed value to the sink node in the same WSN, then the sink node evaluating a threshold and sending an actuation command to an actuator node. The actual latency is dependent on the settings of the TDMA schedule. In this case, we configured the slots so that there are 16 slots between upstream and downstream path. The resulting total latency is about 170 ms. Latency variation is due to the retransmission slots used in upstream and downstream direction. A message might have to be retransmitted in case of losses

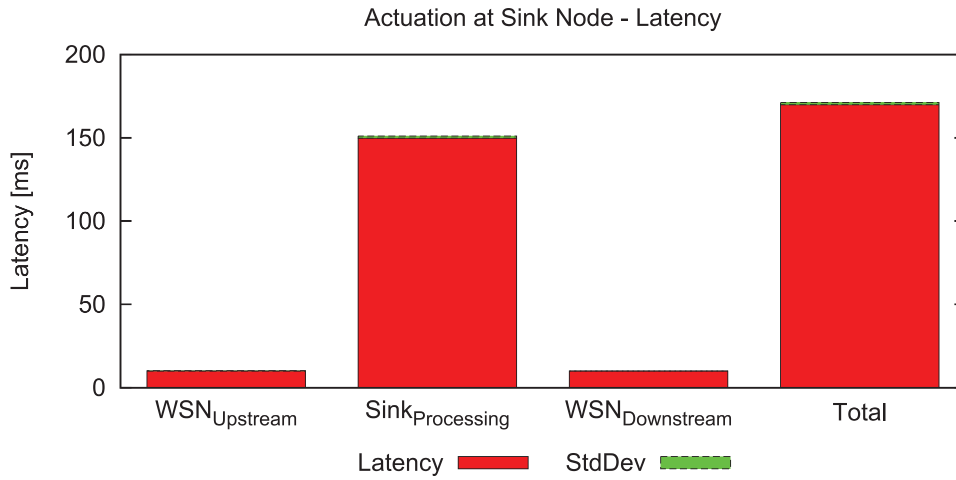


Fig. 24. Average actuation delay through sink node, which is low and bounded, lying well within scenario requirements.

which increases latency, but in any case, the delay would not exceed the end-to-end delay bound.

5.5.2. Closed-Loop through Middleware. For these experiments, both networks were used. In Network 1, node 1-0 was used as sensing node, and node 3-1-0 in Network 2 was the actuator.

When closing the control loop in the middleware, latency increases significantly by an average of 450 ms. This is due to the fact that sensing and actuation cannot be performed anymore within the same TDMA Epoch. The sink node transfers data via serial connection to a sink PC. After data is processed in the middleware, it is transferred back to the sink node, where the actuation command is queued until a slot for downstream transport is available. On average, an actuator command is queued for the duration of half a TDMA epoch before the slot is available (if only one such slot is provisioned per epoch); worst case it would wait for a full epoch length.

5.6. Data Consistency

An experiment was performed to evaluate the consistency between the sensor samples delivered by GINSENG and the values produced by the existing cabled solution. For GINSENG, sensor nodes read from an analog-to-digital converter, while in the cabled system, analog readings are recorded directly. During a one-hour experiment, we measured these two sets of readings and then compared the outputs.

The average measured relative error between the two platforms was seen to be less than 1% for all nodes. Figure 25 plots both curves (wired and wireless) for PT 5170, a Pressure Transmitter. By zooming on a small time interval (five minutes) of Figure 25, one notices that the curves do not precisely match, but that the wireless signal presents some variance, which can generally be attributed to outliers resulting from noise and errors related to analogue-to-digital conversion. In order to improve the data quality, these outliers should be detected and accommodated using domain-specific techniques. Considering the stringent resource constraints of nodes in the field, in particular computational power, a univariate statistical-based approach was implemented at the node level by means of a specific agent in the GinFDS. The algorithm relies on the assumption of quasi-stationarity of the underlying process in the neighbourhood of each sample, and using oversampling techniques.

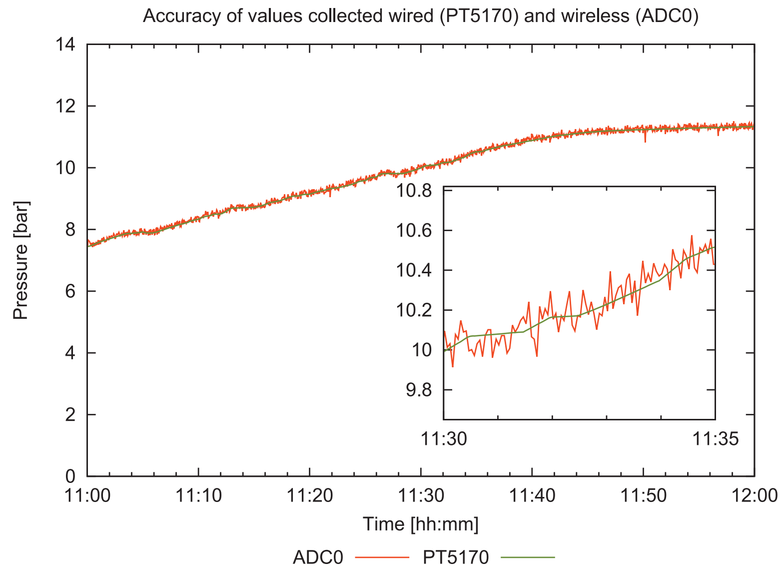


Fig. 25. Comparison of wired (PT 5170) and wireless (ADC0) values, showing strong correlation.

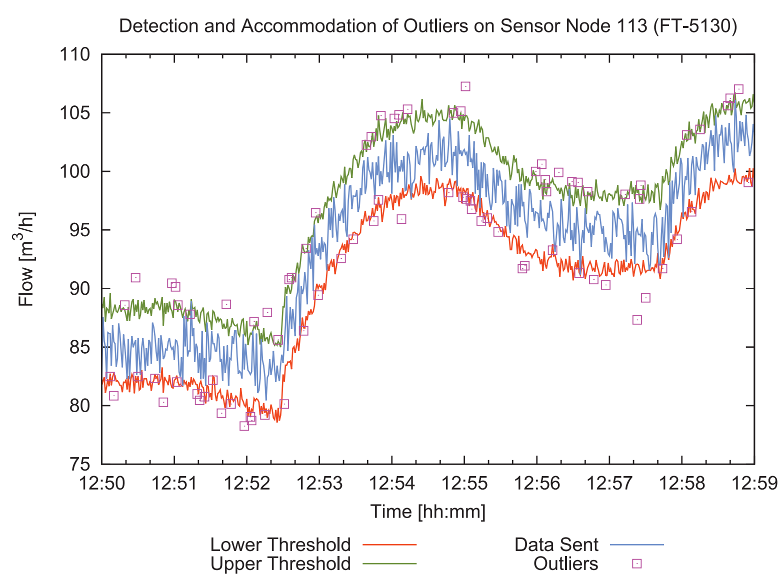


Fig. 26. Outliers detection and accommodation at node level.

Figure 26 shows results using the proposed approach for detecting and filtering of outliers, running on a particular sensor node. As can be observed from this figure, whenever a given sample falls outside the computed admissible threshold, it is assumed as an outlier and accordingly accommodated by replacing the sample with the corresponding moving average value. Overall, the root-mean-square deviation was 3.03×10^{-4} (without accommodation of outliers), respectively 2.23×10^{-4} (with accommodation of outliers).

6. RELATED WORK

In this section, we discuss related work that we divide in two parts. First we discuss some standards that are related to GINSENG, namely IEEE 802.15.4, WirelessHART, ISA100, and WIA-IP. Then we compare our GINSENG system to some other deployment efforts.

6.1. Related Standards

There are some existing and upcoming standards that are related to the GINSENG system. These include IEEE 802.15.4 [IEEE 2003], WirelessHART [Kim et al. 2008], ISA-100.11a [ISA 2009], and the Chinese standard WIA-PA [Zhong et al. 2010].

WirelessHART and the GINSENG system share many similarities at the MAC layer. Both systems are time-slotted using small time slots of 10 ms and use IEEE 802.15.4 frames. In the context of this work, we are mainly interested in the relation between the GINSENG and WirelessHART from a systems point of view. Both systems are centralised. In WirelessHART, the network manager, an entity outside the network, is responsible for computing an exact schedule that specifies when nodes (field devices in WirelessHART terminology) send and receive messages on which of the 16 channels. The network manager also determines the paths. The network manager itself is not part of the standard and the implementations of vendors are usually unknown to the network owner. In GINSENG, our approach is to precompute a schedule offline and compile it into the binary that is running on the nodes. Note, however, that the topology itself does not need to be determined before deployment, and hence the exact placement of a node within a tree may be determined first at run-time.

As Åkerberg et al. have shown, current network managers for WirelessHART (and also for ISA-100.11a) only support efficient data transfer from sensors to the gateway but not from the gateway to actuators [2011]. Hence, the provision of control loops, as we perform in GINSENG, is currently not supported in existing WirelessHART equipment that seems to focus merely on monitoring tasks.

WirelessHART networks are usually deployed as one large network where multiple channels are used to increase reliability. In GINSENG, we take a different approach. Within one network, we use only one channel which reduces complexity and simplifies debugging. Our experiments in the refinery have shown that this is sufficient to achieve very high packet delivery rates even though we do not provide for many retransmissions. Also others have concluded that the use of one channel is enough [Ortiz and Culler 2010]. In GINSENG, we construct larger networks with different subnets that use different channels.

The GINSENG system is a complete system in that we also include, for example, performance debugging, something that is not part of the WirelessHART standard. Furthermore, in this article, we also present detailed performance evaluations of the GINSENG system. We are not aware of any detailed studies regarding the performance of deployed WirelessHART systems, even though some papers have described implementations of WirelessHART [Kim et al. 2008; Song et al. 2008]. Research on WirelessHART has so far focused on scheduling [Saifullah et al. 2011, 2010, Zhang et al. 2009], energy efficiency [Khader et al. 2011], security [Raza et al. 2009, 2009b] and testing suites [Han et al. 2009].

GINSENG uses the physical layer frame format of the IEEE 802.15.4 standard [IEEE 2003] following the philosophy of WirelessHART that picks up only the physical layer of IEEE 802.15.4, which allows greater freedom in exploring design and implementation choices. At the MAC layer, IEEE 802.15.4 networks can either be nonbeacon-enabled or beacon-enabled mode [IEEE 2003]. While, for example, 6LoWPAN does not use the beacon-enabled mode, the beacon-enabled mode enables contention-free access to the wireless medium, which can be utilised to provision sensor networks that provide quality-of-service guarantees, as we do in GINSENG. For example, Tennina et al. use this mode to design the EMMON architecture that aims at large-scale sensor networks for real-time monitoring and has been demonstrated in a testbed of 300 nodes [2011]. Park et al. have designed Breath, a self-adapting protocol that aims at minimising power consumption while giving guarantees on data yield and delay [2011]. They use a MAC layer similar to the one of IEEE 802.15.4 and demonstrate improved performance compared to the standard.

ISA-100.11a is a standard that targets similar scenarios as WirelessHART but offers a "vaster coverage and broader view of process automation solution" [Wang 2011]. We are not aware of any deployment results. The same is true of WIA-PA [Zhong et al. 2010].

6.2. Related Sensor Network Projects and Deployments

While there exists a large number of deployments of sensor networks [Romer and Mattern 2004], there are only few deployments in real industrial settings. Krishnamurthy et al. deployed trial deployments of wireless sensor networks in two industrial settings, a semiconductor plant and an oil tanker [2005]. Peterson and Carlsen evaluated WirelessHART in a lab setting [2009], but as our experience has shown, this is very different from deploying a sensor network in a real industrial environment.

WINTeR is a test bed specifically targeted at radio-harsh environments that can be found in the oil and gas industry. The test bed mimics the industrial surroundings with complex multipath propagation, provides the means to generate interference, and has software that allows remote access. Unlike the GINSENG deployment, however, WINTeR is not in a real industrial plant but replicates real industrial surroundings [Slipp et al. 2008].

Similar to GINSENG, the WASP project also targets integration with the backends and existing software environments [Atallah et al. 2008]. In contrast to GINSENG, their focus is on healthcare and herd monitoring, rather than industrial monitoring and control, which has more stringent performance requirements and comprises also control.

Two deployments that have been very successful in reliable data delivery are RACNet, a large sensor network with almost 700 nodes, deployed for monitoring environmental parameters in data centres [Liang et al. 2009], and the deployment in the Torre Aquila [Ceriotti et al. 2009]. Both papers report data delivery rates above 99%. There was, however, no need for low or predictable delay, as in GINSENG. A subsequent deployment of Ceriotti et al. was the first to close the control loop, but in contrast to our deployment, the control loops were at larger time scales and the deployment was not in an industrial environment [Ceriotti et al. 2011]. The PermaSense deployment uses the Dozer protocol stack to achieve duty cycles below 1% but at much lower data rates than the GINSENG system and without closing the control loop [Beutel et al. 2009]. Lu et al. have presented RAP, a protocol that reduces the end-to-end deadline miss ratio for sensing applications but does not target control [2002].

7. DISCUSSION

In this section, we discuss the key lessons from GINSENG and summarise the feedback from the end-user company Petrogal.

7.1. Key Lessons

The result of the GINSENG project is the design, implementation, deployment and evaluation of a sensor network system for wireless monitoring and control. Here we summarise the key lessons learnt during these activities.

(1) TDMA-based medium access control proved to be an appropriate choice in allowing us to provision the sensor network for assured delivery delays.
(2) Even in environments where radio link behaviour is observed to be usually very good, it is necessary to have built-in mechanisms to adapt to link dynamics.
(3) In our experience, links were often quite stable in terms of expected operational bounds, so the additional overhead (notably in terms of energy) of mechanisms that can monitor behaviour and adapt in real time is usually not justified. Instead, techniques that adapt over longer time-scales are more likely to be more appropriate.
(4) Providing an end-to-end solution with middleware that seamlessly integrates with backend IT systems is vitally important from an end-user perspective.
(5) The ability to offer a variety of backend applications/interfaces proved decisive in end-user acceptance and evaluation.
(6) The physical environment presented several challenges that impact performance control and must be considered in deployment planning, notably restrictions on potential node locations (and hence topology), ATEX packaging and its effects on communication, and the need for corrosive-resistant antennas.
(7) The difficulty in access to the deployment site, coupled with the experimental nature of the system, meant that we relied heavily on our ability to remotely reprogram the sensor nodes.
(8) Our performance debugging tools were essential in allowing us to quickly identify problems, such as short-term or intermittent radio link failures, but in cases where the causes were due to external interference, it was usually not possible to identify the root cause due to the complexity of the physical environment.
(9) An essential requirement that must be considered up front is the need to be able to establish ground truth, including for wireless link behaviour and sensor data readings.
(10) For larger deployments of wireless monitoring and control networks, a suite of automated planning tools will be needed that take into account the key factors, including physical site limitations, topology design, and performance scheduling.

We hope these lessons will be useful to other researchers working on wireless monitoring and control and to industry practitioners involved in evolving current standards for such networks.

7.2. Industry Feedback

The role of partner Petrogal was to (i) provide domain-specific input in specifying the application scenarios, including tolerances for delay and reliability; (ii) to facilitate the test-bed deployment, including identifying the physical location, commissioning equipment, and maintaining the system; and (iii) to provide end-user analysis of the final system.

For Petrogal, a key benefit of the project has been to allow an assessment of wireless technology for monitoring and control, allowing a direct in situ contrast with their existing wired system. The provision of a backend application with a user interface that was modelled on the current system they use acted to facilitate this activity by allowing control room operators to observe GINSENG in operation alongside the system they use today. Petrogal have hosted visits by several technology companies to show them the GINSENG test bed and impress upon them their plans to embrace wireless technology in the future.

Feedback from Petrogal to the research partners highlighted several key benefits of adopting GINSENG:

(1) flexibility for post-deployment reconfiguration (in the Petrochemical industry this is frequent);
(2) lower costs for deployment, primarily by avoiding digging trenches for cables;
(3) ability of the GINSENG middleware system to integrate with the WSN and support a varied and extendable set of related backend applications;
(4) provision of a monitoring backend application that can integrate with the GINSENG middleware and represent the sensors/actuators with a graphical user interface that is similar to that currently used by the control room staff.

Petrogal also identified several open issues with the GINSENG solution.

(1) GINSENG is not a finished product, and it not an industry standard.
(2) Experimental results are convincing, but there is a need for techniques to handle poor radio links.
(3) The required effort/cost for planning a GINSENG-like network is unclear.
(4) While outside the scope of the project, the lack of security mechanisms is an obvious limitation that would inhibit commercialisation.

In addition, Petrogal believe that while GINSENG is very promising, in general, WSN hardware and software tools do not seem as mature when compared to those for wired control systems.

8. CONCLUSION

From the smallest to the biggest company, intelligent systems are crucial to assuring reliable and healthy operability. In critical scenarios, such as an oil refinery, the whole system must work 24/7, and any interruption has an impact on the economy and any accident means an extremely dangerous situation for the population and environment. Industrial process automation and control systems are used on a large scale, and we rely on their correct operation. Industry demands more flexible and cost-effective solutions which can be implemented using wireless technology. However, when employing wireless technology, a number of challenges have to be addressed in order to maintain reliability levels as present in current wired systems.

The GINSENG research project has designed and deployed a WSN-based solution that offers the benefits of WSN, such as low-cost and ease of deployment, while recognising the need for operating to required performance levels. In this article, we presented the GINSENG solution and the results of our on-site deployment and experiments. The experiments show that WSN deployment in an industrial process monitoring and control setting is extremely demanding in terms of hardware deployment, radio communication, performance assurance, and system management. The results allowed a thorough assessment of GINSENG, focusing on meeting targets for message delivery latency and reliability. It was demonstrated that GINSENG is able to integrate the required backend capabilities that are desired to support the needs of the industrial partner. Our deployment was limited in terms of the number of sensor nodes but was sufficient to evaluate the fundamental operational objectives for a GINSENG sensor network. Each individual GINSENG sensor network is limited in scale by virtue of having to assure message latency and thus having to bound the number transmissions on the path from each node to the sink. In the GINSENG approach, scaling to larger deployments is achieved by linking together multiple GINSENG sensor networks using the GINSENG middleware, which has proven scalability properties. Future work will include development of tools to automate the deployment planning process and to manage such large GINSENG deployments.

IN MEMORIAM

Dr. Tony O'Donovan, our colleague and co-author, passed away unexpectedly while we were writing this article. He is listed as the first author in recognition of his invaluable contributions to the GINSENG project and in particular to programming and experimenting with our real-world wireless sensor network. Tony's research focused on communication protocols for wireless networks and, at the time of his passing, he had published several peer-reviewed scientific papers, many of which have already been cited. On the basis of his contributions to these papers, his examiners approved that the Ph.D. degree be awarded posthumously. Tony was highly motivated, brooked no obstacles, and was a source of inspiration to others, both faculty and students alike. He will be remembered by his close colleagues as warm and gregarious, always willing to draw on his depth of knowledge to offer technical advice, and always keen to encourage us to question accepted wisdom and established beliefs.

REFERENCES

ÅKERBERG, J., GIDLUND, M., LENNVALL, T., NEANDER, J., AND BJÖRKMAN, M. 2011. Efficient integration of secure and safety critical industrial wireless sensor networks. *EURASIP. J. Wirel. Commun. Netw. 2011,* 1, 100.

ATALLAH, L., LO, B., YANG, G., AND SIEGEMUND, F. 2008. Wirelessly accessible sensor populations (WASP) for elderly care monitoring. In *Proceedings of the 2nd International Conference on Pervasive Computing Technologies for Healthcare*. IEEE, 2–7.

BEUTEL, J., GRUBER, S., HASLER, A., LIM, R., MEIER, A., PLESSL, C., TALZI, I., THIELE, L., TSCHUDIN, C., WOEHRLE, M., AND YUECEL, M. 2009. PermaDAQ: A scientific instrument for precision sensing and data recovery in environmental extremes. In *Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE Computer Society, 265–276.

BOANO, C., BROWN, J., HE, Z., ROEDIG, U., AND VOIGT, T. 2009. Low-power radio communication in industrial outdoor deployments: The impact of weather conditions and ATEX-compliance. In *Proceedings of the 1st International Conference on Sensor Networks Applications, Experimentation and Logistics*. 159–176.

CERIOTTI, M., CORRÀ, M., D'ORAZIO, L., DORIGUZZI, R., FACCHIN, D., GUNA, S., JESI, G. P., LO CIGNO, R., MOTTOLA, L., MURPHY, A., PESCALLI, M., PICCO, G. P., PREGNOLATO, D., AND TORGHELE, C. 2011. Is there light at the ends of the tunnel? Wireless sensor networks for adaptive lighting in road tunnels. In *Proceedings of the 10th International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 187–198.

CERIOTTI, M., MOTTOLA, L., PICCO, G., MURPHY, A., GUNA, S., CORRA, M., POZZI, M., ZONTA, D., AND ZANON, P. 2009. Monitoring heritage buildings with wireless sensor networks: The Torre Aquila deployment. In *Proceedings of the International Conference on Information Processing in Sensor Networks*. IEEE Computer Society, 277–288.

HAN, S., SONG, J., ZHU, X., MOK, A., CHEN, D., NIXON, M., PRATT, W., AND GONDHALEKAR, V. 2009. Wi-HTest: Compliance test suite for diagnosing devices in real-time wirelesshart network. In *Proceedings of the Real-Time and Embedded Technology and Applications Symposium (RTAS'09)*. IEEE, 327–336.

IEEE. 2003. IEEE Standard for Information Technology – Telecommunications and information exchange between systems – Local and Metropolitan area networks – Specific Requirements. 802.15.4-2003.

ISA. 2009. ISA-100.11a, wireless systems for industrial automation: Process control and related applications. ISA Standardization.

KHADER, O., WILLIG, A., AND WOLISZ, A. 2011. WirelessHART TDMA protocol performance evaluation using response surface methodology. In *Proceedings of the International Conference on Broadband and Wireless Computing, Communication and Applications*. 197–206.

KIM, A. N., HEKLAND, F., PETERSEN, S., AND DOYLE, P. 2008. When HART goes wireless: Understanding and implementing the WirelessHART standard. In *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. 899–907.

KLEIN, A., HACKENBROICH, G., AND LEHNER, W. 2009. How to screen a data stream - quality-driven load shedding in sensor data streams. In *Proceedings of the 14th International Conference on Information Quality (ICIQ)*. 76–90.

KLEIN, A. AND LEHNER, W. 2009a. How to optimize the quality of sensor data streams. In *Proceedings of the 4th International Multi-Conference on Computing in the Global Information Technology*. 13–19.

KLEIN, A. AND LEHNER, W. 2009b. Representing data quality in sensor data streaming environments. *J. Data Info. Quality 1*, 10:1–10:28.

KRISHNAMURTHY, L., ADLER, R., BUONADONNA, P., CHHABRA, J., FLANIGAN, M., KUSHALNAGAR, N., NACHMAN, L., AND YARVIS, M. 2005. Design and deployment of industrial sensor networks: Experiences from a semiconductor plant and the North Sea. In *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems (SenSys)*. ACM, 64–75.

LIANG, C., LIU, J., LUO, L., TERZIS, A., AND ZHAO, F. 2009. Racnet: A high-fidelity data center sensing network. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys)*. ACM, 15–28.

LU, C., BLUM, B., ABDELZAHER, T., STANKOVIC, J., AND HE, T. 2002. RAP: A real-time communication architecture for large-scale wireless sensor networks. In *Proceedings of the 8th IEEE Real-Time and Embedded Technology and Applications Symposium*. IEEE, 55–66.

MUNIR, S., LIN, S., HOQUE, E., NIRJON, S. M. S., STANKOVIC, J. A., AND WHITEHOUSE, K. 2010. Addressing burstiness for reliable communication and latency bound generation in wireless sensor networks. In *Proceedings of the International Conference on Information Processing in Sensor Networks*. 303–314.

ORTIZ, J. AND CULLER, D. 2010. Multichannel reliability assessment in real world WSNs. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*. 162–173.

PARK, P., FISCHIONE, C., BONIVENTO, A., JOHANSSON, K., AND SANGIOVANNI-VINCENT, A. 2011. Breath: An adaptive protocol for industrial control applications using wireless sensor networks. *IEEE Trans. Mobile Comput. 10,* 6, 821–838.

PETERSEN, S. AND CARLSEN, S. 2009. Performance evaluation of WirelessHART for factory automation. In *Proceedings of the IEEE Conference on Emerging Technologies & Factory Automation (EFTA)*. IEEE, 1–9.

RAMAN, B. AND CHEBROLU, K. 2008. Censor networks: A critique of sensor networks from a systems perspective. *ACM SIGCOMM Comput. Commun. Rev. 38,* 3, 75–78.

RAZA, S., SLABBERT, A., VOIGT, T., AND LANDERNAS, K. 2009a. Security considerations for the WirelessHART protocol. In *Proceedings of the IEEE International Conference on Emerging Technologies & Factory Automation (ETFA)*. 1–8.

RAZA, S., VOIGT, T., SLABBERT, A., AND LANDERNAS, K. 2009b. Design and implementation of a security manager for WirelessHART networks. In *Proceedings of the IEEE International Workshop on Wireless and Sensor Networks Security (WSN'09)*. 995–1004.

ROMER, K. AND MATTERN, F. 2004. The design space of wireless sensor networks. *IEEE Wirel. Commun. 11,* 6, 54–61.

SAIFULLAH, A., XU, Y., LU, C., AND CHEN, Y. 2010. Real-time scheduling for WirelessHART networks. In *Proceedings of the Real-Time Systems Symposium (RTSS)*. 150–159.

SAIFULLAH, A., XU, Y., LU, C., AND CHEN, Y. 2011. End-to-end delay analysis for fixed priority scheduling in WirelessHART networks. In *Proceedings of the Real-Time and Embedded Technology and Applications Symposium (RTAS)*. 13–22.

SLIPP, J., MA, C., POLU, N., NICHOLSON, J., MURILLO, M., AND HUSSAIN, S. 2008. WINTeR: Architecture and applications of a wireless industrial sensor network testbed for radio-harsh environments. In *Proceedings of the 6th Annual Communication Networks and Services Research Conference (CNSR)*. IEEE, 422–431.

SONG, J., HAN, S., MOK, A., CHEN, D., LUCAS, M., AND NIXON, M. 2008. Wirelesshart: Applying wireless technology in real-time industrial process control. In *Proceedings of the IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'08)*. IEEE, 377–386.

SRINIVASAN, K. AND LEVIS, P. 2006. RSSI is under appreciated. In *Proceedings of the 3rd Workshop on Embedded Networked Sensors (EmNets)*.

SURIYACHAI, P., BROWN, J., AND ROEDIG, U. 2010. Time-critical data delivery in wireless sensor networks. In *Proceedings of the 6th IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS'10)*. IEEE.

TENNINA, S., BOUROCHE, M., BRAGA, P., GOMES, R., ALVES, M., MIRZA, F., CIRIELLO, V., CARROZZA, G., OLIVEIRA, P., AND CAHILL, V. 2011. EMMON: A WSN system architecture for large scale and dense real-time embedded monitoring. In *Proceedings of the 9th IEEE/IFIP International Conference on Embedded and Ubiquitous Computing (EUC'11)*. 150–157.

TRAN, T.-D., SILVA, R., NUNES, D., AND SILVA, J. 2011. Characteristics of channels of IEEE 802.15.4 compliant sensor networks. *Wirel. Personal Commun. 67*, 3, 541–556.

WANG, G. 2011. Comparison and evaluation of industrial wireless sensor standards ISA100.11A and WirelessHART. M.Sc. Thesis EX036/2011, Chalmers University of Technology.

Zhang, H., Soldati, P., and Johansson, M. 2009. Optimal link scheduling and channel assignment for convergecast in linear wirelesshart networks. In *Proceedings of the International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOPT)*. 1–8.

Zhong, T., Mengjin, C., Peng, Z., and Hong, W. 2010. Real-time communication in WIA-PA industrial wireless networks. In *Proceedings of the 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT)*. Vol. 2. IEEE, 600–605.

Received May 2012; revised October, December 2012; accepted December 2012