Invited poster: GINSENG: Performance Control in Wireless Sensor Networks

J. Brown[†], U. Roedig[†], T. O'Donovan[*], C. J. Sreenan[*], Z. He[‡], T. Voigt[‡], B. Pöttner[**], and L. Wolf[**], A. Klein[††], J. Sá Silva[‖], V. Vassiliou[§], J. do Ó[¶]

[*]University College Cork
Cork, Ireland

[†]Lancaster University
Lancaster, UK

[‡]Swedish Institute of Computer Science
Kista, Sweden

[§]University of Cyprus
Nicosia, Cyprus

[‖]University of Coimbra
Coimbra, Portugal

[¶]Petrogal
Lisbon, Portugal

[**]Technische Universität Braunschweig
Brunswick, Germany

[††]SAP AG
Walldorf, Germany

***Abstract*—The goal of the GINSENG project is a performance-controlled sensor network suitable for time-critical applications like plant automation and control. Such sensor networks must monitor performance constantly and collect data in order to detect, diagnose and rectify problems. The GINSENG system contains a number of mechanisms used to collect, transport and analyse performance critical data. Performance data processing is carried out in a manner that ensures time-critical application data processing is not affected. In this paper we present the different performance debugging mechanisms included in the GINSENG system.**

I. Introduction

Most industrial process control and automation applications can be considered time-critical. In these scenarios, control loops are mapped onto wireless sensor networks. For such control loops to function, data needs to be delivered to a sink reliably and within a given time bound [1]. Furthermore, it might be necessary to react upon the received data and take a corrective action. Such a command needs to be sent from the sink to an actuator in the sensor network reliably and within a given time bound. As WSNs are considered to be relatively unreliable, they have seen little use for such tasks so far.

The EU-funded GINSENG project [2] aims to develop a WSN system that can be used to support the aforementioned time-critical applications. Unlike applications that are based on random deployment, GINSENG assumes a carefully planned and deployed network of sensor nodes as a basis to achieve performance control. GINSENG provides novel software components (for example, Operating System and TDMA Medium Access Control) and algorithms (for example, topology control and flow control) to ensure time-critical data delivery.

Due to the inherent uncertainties of the wireless communication medium, it is possible to experience changes in the operating environment that result in undesired deterioration in network performance, motivating the need to monitor and potentially debug the performance of a deployed GINSENG system. Therefore, GINSENG provides mechanisms and tools to carry out performance debugging of deployed systems and enable reconfiguration when given performance metrics can no longer be achieved. Within this paper the GINSENG performance debugging mechanisms are detailed.

Fig. 1. The GALP oil refinery in Sines, Portugal.

II. The GALP Refinery Application Scenario

The GALP oil refinery at Sines, Portugal, is used as a test application scenario for GINSENG. The refinery is a complex industrial facility that includes a wide range of processing, and such processing needs careful monitoring and control of operations. There are currently approximately 35000 sensors (some shown in Figure 1) and actuators in use in the refinery to perform real-time monitoring of industrial operations and systems allowing detection of leaks, measurement of pressure in pipes, fluid levels in tanks and providing data on environmental conditions. The monitoring of the environment in a refinery provides essential information to ensure the good health of both the refinery and its production processes.

The refinery provides a challenging environment as machinery and metal structures create interferences and obstacles that a deployed sensor network must handle. In such a testing domain performance debugging is essential.

III. The GINSENG System

Sensor nodes in a GINSENG system are deployed in groups (cells) of a few dozen nodes. Each cell uses a TDMA MAC protocol to ensure that data is delivered to a sink in a timely and reliable manner. The sink is connected to a wired backbone infrastructure. Multiple cells operating on different transmission frequencies might be used to accommodate a large number of nodes. Before network deployment a TDMA schedule for each cell is determined. This schedule must incorporate room for application and performance related data, corrective action messages and also allowing for potential message re-transmissions. Thus, the deployed network can compensate for link quality fluctuations to some degree by using retransmissions and topology re-structuring. However, these compensation mechanisms are limited and fail if link quality fluctuations exceed bounds anticipated at deployment time. Therefore, it is necessary to monitor the deployed network and to collect data that allows us to calculate a new schedule and adjust the deployment if needed.

IV. Performance Debugging in GINSENG

Each sensor node in a GINSENG system monitors network performance parameters. The majority of collected information is stored on nodes and is only transported to the sink when performance problems occur. After requesting performance debugging information from nodes, tools can be used to debug the network and isolate performance problems. The following describes the performance debugging system.

A. Data Collection

Active and passive methods are used to collect performance debugging data. Active data collection requires nodes to send dedicated probing packets in addition to application data packets. For example, GINSENG uses a probing mechanism to periodically measure burst error patterns of all links used in the deployment [3]. Passive data collection uses existing data transmissions to gather additional performance information. For example, the received signal strength of incoming data packets. The combination of active and passive methods ensures collection of rich performance debugging data.

B. Data Storage

A large amount of operations information is kept on each node within a Management Information Base (MIB). This MIB is available to all modules of the system and can be used for diagnosis of detected performance problems.

The limited bandwidth available in WSNs means only a fraction of the information available to nodes can be transported to the sink over the air. Rather than disregarding all of this unsent data we store some of it in the nodes onboard flash. Two flash storage systems are used, the Coffee file system provided by Contiki and a ring-storage system we designed. The ring-storage system targets scenarios with mainly sequential read and write operations, such as for error logging. Storing data in the flash provides a comprehensive archive that can be queried when required.

C. Data Transport

Along with the application data and the necessary fields required for message transport we include a timestamp, hopcount and sequence number in each message. This small amount of 'piggybacked' data provides delivery delay and packet losses which are key performance metrics in any WSN.

Excessive delivery delays or packet losses indicate performance anomalies in the network. However there are many possible causes of such problems, in order to diagnose them further information may be required. We provide a system to request this data from a specified node along with all the nodes along the path between the node and the sink.

D. Data Analysis

Data analysis can be performed both in the network and at the sink. Previously we presented a storage-centric approach for analysing performance anomalies in deployed sensor networks [4]. In this approach we have leveraged the local flash storage of sensor nodes to log performance data and combined this with on-node statistical analysis. For example, we have used data at the node to verify that there is a correlation between environmental conditions and the required transmission power for successful communication. Our results have shown that it is not only feasible to store and analyze large quantities of data on the nodes, but it is also more energy efficient than sending all data to the sink for analysis.

For analysis purposes at the sink resp. in the backbone infrastructure, we introduced a dispatcher component between the WSN and the wired backbone. This component reads, parses and augments incoming raw data (e.g., with packet delay). It formats the data and makes it available to other consumer components. Consumers can be backend data processing systems as well as monitor and debugging components. The monitor application is one such consumer that provides a GUI that network operators can use to get information on the overall network and individual nodes helping with the diagnosis of performance anomalies in the WSN.

V. Conclusions

The main goal of the GINSENG project is to provide a wireless sensor networks with performance assurances for important metrics such as delay and reliability which are crucial for a broad class of emerging sensor network applications. In this paper, we have presented the GINSENG approach to the performance debugging of the wireless sensor network.

Acknowledgments

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 224282.

References

[1] F. Lian, J. Moyne, and D. Tilbury, "Network design consideration for distributed control systems," *IEEE Trans. Control Syst. Technol*, vol. 10, pp. 297–307, Mar 2002.

[2] "The GINSENG project," Web page, visited 2010-05-26. [Online]. Available: http://www.ict-ginseng.eu/

[3] J. Brown, B. McCarthy, U. Roedig, T. Voigt, and C. J. Sreenan, "Burst-probe: Debugging time-critical data delivery in wireless sensor networks," in *Proceedings of the Eighth European Conference on Wireless Sensor Networks (EWSN 2011)*, Feb 2011.

[4] T. O'Donovan, N. Tsiftes, Z. He, T. Voigt, and C. J. Sreenan, "Detailed diagnosis of performance anomalies in sensornets," in *Proceedings of the Sixth ACM Workshop on Hot Topics in Embedded Networked Sensors (HotEmNets 2010)*, Jun 2010.