PROVIDING ALARM DELIVERY GUARANTEES IN HIGH-RATE INDUSTRIAL WIRELESS SENSOR NETWORK DEPLOYMENTS

José Cecílio, João Costa, Pedro Martins and Pedro Furtado
University of Coimbra, Coimbra, Portugal

Keywords: Wireless Sensor Networks (WSN), Data Processing, Alarm delivery guarantees.

Abstract: We are witnessing a large increase of Wireless Sensor Network (WSN) deployments, used to sense, monitor and act on the environment, because cable-free solutions are easier to deploy. However, in industrial applications, WSNs still aren't seen as a viable option because of the required fast sampling rates and the reduced time delay, particularly in a multi-hop deployment where traffic congestion may occur. Message losses are unacceptable, particularly in what concerns critical messages containing actuation instructions or other urgent data, which may have time-constraint requirements that cannot be guaranteed. Our research focuses on integrating traffic-aware data processing strategies and network traffic prioritization to overcome congestion states, in order to guarantee that urgent alarms and commands are enacted in time. Traffic is divided into urgent messages (alarms and actuation commands), that have time-delivery requirements; and the remaining periodic sensed data. In this paper we propose an integrated approach NetDP, which applies adaptable data processing strategies (DP) and traffic reduction (DP-Manager) policies to ensure that application requirements are satisfied with minimal message losses, while simultaneously guaranteeing timely delivery of alarms. We demonstrate that NETDP solution with different data processing strategies and levels of system stress can efficiently guarantee the timely delivery of alarms and actuation messages.

1 INTRODUCTION

Sensor network applications are mostly data-driven, and are deployed to monitor, understand and to act upon the physical world. The potential of Wireless Sensor Network (WSN) technology has been clearly demonstrated, and this has led to increasing research, in both academia and industry, concerning design and implementation methods to efficiently operate wireless sensor networks. A very important issue is how to offer a high degree of reliability in critical systems, since sensor networks have several limitations concerning energy, interference, signal strength, congestion and timing limitations.

Our research was done in the context of FP7 European project GINSENG (Ginseng, 2010), which aims to plan and develop a performance controlled WSN to apply in critical scenarios. The overall goal of GINSENG is to ensure that wireless sensor networks meet application-specific performance targets and that will integrate with industry resource managements systems. The project application scenarios include the Petrogal oil refinery, located at Sines, Portugal.

We have created a twin lab testbed setup to test solutions for the refinery testbed before deploying them. This is an important step, since the refinery plant zone is an ATEX security area, where only certified personnel is allowed and ATEX compliant, sealed equipment can be deployed. By mirroring the actual deployment in our lab, we are able to test alternative solutions, and to deploy only the final choices in the actual refinery locations.

In such industrial scenarios, where wireless sensor networks with tens or hundreds of nodes are programmed to sense and react within tens of milliseconds range, in conjunction with high data rates and complex multi-hop network layouts, high message loss ratios may result. If 50% to 70% of application-level messages are lost, it's almost impossible to guarantee the timely delivery of urgent messages and commands.

In this paper we propose NetDP, which is an approach designed to provide the necessary means to configure, test and handle high-rate sensor data

Cecílio J., Costa J., Martins P. and Furtado P..
PROVIDING ALARM DELIVERY GUARANTEES IN HIGH-RATE INDUSTRIAL WIRELESS SENSOR NETWORK DEPLOYMENTS.
DOI: 10.5220/0003357904270433
In *Proceedings of the 1st International Conference on Pervasive and Embedded Computing and Communication Systems* (PECCS-2011), pages 427-433
ISBN: 978-989-8425-48-5

using in-network approaches. After a user makes an initial deployment plan and programs the nodes with the NetDP software, he deploys the nodes and uses our tool to test whether the system is functioning well. If the system is not responding adequately due to excess traffic, the user makes changes to the way information is processed using a set of alternatives that are provided by NetDP.

Application-level messages can be divided into Urgent messages, which need delivery guarantees and time-delivery constraints (e.g. alarms triggered by sensors nodes and actuation commands to motes), and Periodic sensor data that should be delivered according to user specified and application requirements. Periodic data transmission rates should be set to a level that allows the timely delivery of urgent messages, while complying with the application requirements. We provide mechanisms to measure and test the network, offering users some valuable feedback to help them to adjust system configuration parameters to acceptable boundaries.

The proposed approach consists of three main modules: a Data Processing module (DP) that implements different data processing algorithms, a Network Status module (NS) which gathers and produces network status information and a DP-Manager module that allows users to adjust data processing configuration parameters to tune the overall system.

The paper is organized as follows: section 2 discusses related work; section 3 presents our integrated in-network and data processing solution, including network status measures and implemented data processing solutions; in section 4 we analyze experimental results and section 5 concludes the paper.

2 RELATED WORK

Our proposal is an integrated approach for providing both guarantees to urgent messages and reliable data collection, through the use of monitoring and traffic-intensity configurable data processing approaches fitted at user needs. Related work includes monitor, studies on packet lost and in-network data processing.

Monitoring tools are crucial to measure network-specific parameters and to control the performance of a wireless sensors network. Monitoring tools such as Sympathy (Ramanathan et al., 2005), Sensor Network Management System (SNMS) (Tolle & Culler, 2005), Sensor Network Inspection Framework (SNIF) (Ringwald & Römer, 2007) and Distributed Node Monitoring in Wireless Sensor Networks (DiMo) (Meier et al., 2008), can monitor necessary parameters to ensure that all functionalities are working as expected. Our work is related to these ones in what concerns monitoring, and their failure detection mechanisms can be integrated into our system to enhance the detected failure conditions. However, our monitoring component includes additional application-level, end-to-end message lost and message delivery tests.

To avoid congestion, we include in our prototype in-network data processing mechanisms. Some previous studies, such as (Jianbo et al., 2006) and (Wu & Tian, 2006) have shown that computing in-network significantly reduces the amount of communication and the energy consumed. We also use in-network data processing approaches to decrease the required amount of communication. These approaches are integrated in our system together with delivery guarantees of urgent messages, monitoring and configuration to reduce network traffic to acceptable levels.

Our approach addresses the impact of excess traffic from empirical results collected on a real deployed wireless sensor network with contention based protocols. These results were important for evaluating the delivery guarantees of urgent messages, and alternative data processing and configuration mechanisms provided to bring the traffic intensity to acceptable levels.

3 NetDP - IN-NETWORK STATUS AND DATA PROCESSING SOLUTION

NetDP is an approach designed to provide the necessary means to configure, test and handle high-rate sensor data using in-network approaches.

The system is divided into two major entities: a Manager application running in a workstation, which controls and manages WSN nodes actions, and an Executor application running at sensor and relay nodes, which processes and sends data, receives and performs actuation commands.

The Manager application includes a Data Processing Manager (DPmanager) and a Network Status Manager (NSmanager); Executors include a Data Processing Module (DPmodule) and a Network Status Module (NSmodule), as illustrated in figure 1. Each of these modules have a set of configurable parameters that can be adjusted by issuing

configuration commands to nodes (sensors and relays) with end-to-end acknowledge requirements, to activate different data processing strategies, run network status tests, or change just some of the existing configuration parameters.

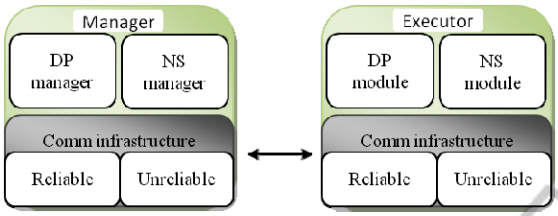

Figure 1: NetDP Modules.

The DPmanager sends reliable configuration commands to sensors and relays, and manages client-side data processing. When reliable commands are sent, consecutive retries may be tried until an acknowledgement (or a failure timeout) has been received. The Executor, when receiving a configuration command, forwards it to the executor DP module for processing and configuring the instructed DP parameters. Upon completion, the executor sends an acknowledgement message confirming the reception and the execution of the requested command.

The NSmanager sends network status related requests and gathers network status information from executors, in order to compute valuable network statistics indicators. The executors NSmodule gathers and sends the network status information to the NS Manager.

In the next subsections we discuss (re)configuration, application-level network status measures and tests, and we present data processing strategies and their configuration parameters.

3.1 (RE) Configuration for Guaranteed Delivery

After network status tests, reconfiguration may be necessary to guarantee that all information is delivered or to improve the performance of the system. The user can change configuration parameters until the desired characteristics are obtained. In order to do this, the reconfiguration requires a set of configurations that should be used in a successive test procedure, until the desired characteristics are obtained.

During a test, reconfiguration module collects information provided by network status, considering metrics such as message/sample loss ratio, delays and the number of failed acks. If any metric fails to provide desired guarantees, it is necessary to reconfigure the system. The users can change the sampling rate, opt for other data processing approach or change any parameter associated to the data processing approaches that are summarized in section 3.3.

3.2 Network Status Measures and Tests

Traffic-aware data processing approaches need to use traffic-reduction configuration parameters to avoid congestion and allow good network performance characteristics. One key component is to have a set of measures and tests that detect network performance problems and application-level message delivery tests are crucial. We define a set of measures and tests that can be used and that allow the user to conclude whether the network status is satisfactory or, otherwise, if it is necessary to change some configuration parameters and/or data processing approach. Given a deployment, these tests can be used to verify whether the traffic characteristics are as desirable, or to test different alternatives concerning data processing approaches and/or traffic-related configurations, in order to decide the most appropriate ones. We provide alternative data processing approaches and configuration parameters for traffic reduction. The actual choice of a user should depend on application context requirements and the results of these tests.

Our focus is on detecting excess traffic conditions that lead to excessive application-level message loss, so we assume link and node connection. In practice, the Network Status module also periodically tests connection between any node and the sink and report disconnection.

In the rest of this section we will discuss the tests used by our approaches to evaluate the network status.

Message Loss Ratio: The number or ratio of lost messages is an important measure of network performance, revealing problems that may be due to disconnection or other outside factors of the environment. In our approaches we use message loss ratio as an indicator of excess traffic conditions.

Number of Retries for Guaranteed Delivery of Messages: The number of retries is an application-level test that is based on sending messages with ack between sensor and sink node and between sink and sensor node. A significant number of retries is indicative of problems due to excess traffic intensity, providing useful information to the user, who can then conclude if it is necessary to change some configuration that decreases traffic intensity.

End-to-end Delay: We measure the end-to-end delay regarding a transmission of data messages. The end-to-end delay implies the time taken between messages is submitted by the source and when it is successfully received at the destination, and it accounts for the sum of all components of the delay, including queuing and the propagation delay of each message.

3.3 Data Processing Approaches

The overall objective of the Data Processing part is to provide flexibility for users to configure collection of data values in a way that brings traffic into acceptable levels to offer timely delivery guarantees for alarms and commands with end-to-end acknowledgement.

Different strategies have been studied concerning ways to extract useful data from the network, and provide a compact delivery of that information to the user. We consider three common types of in-network processing (aggregation, merging and compression), besides the basic alternative of sense-and-send. Each of these alternatives fits into different application needs – for instance, a sense-and-react system may require frequent detailed sensor data, while another application may tolerate a larger delay or accept statistically-summarized data every 2 seconds. Next we present some of the data processing strategies included in our system:

Synchronous Delivery of Sensed Data (SD-SD): this is the basic approach, where sensors periodically gather and send sensor data values to sink without further processing. Users can only adjust the sampling rate.

The next in-network processing alternatives trade information loss with delay: instead of reducing the sampling rate, they merge, aggregate or summarize several values into statistical measures.

Aggregated Delivery of Sensed Data (AD-SD): this approach aggregates continuous data readings within the sensor node (or at intermediate nodes), and sends the aggregated data to the sink. The user can configure the maximum delivery delay, which internally is translated into an adjustment in the size of the underlying window used to store the sensor values before computing the statistical information.

Merged Delivery of Sensed Data (MD-SD): this approach exploits the fact that in the basic approach most data packets could be stuffed with much more data. Ensuring that data packets accommodate multiple samples before being sending the packet reduces the overall number of in-network exchanged packets. For a maximum packet size we concatenate several consecutive sensor values together, while there is space available in the packet and a time limit is not met, thus sending a single packet instead of one packet per reading. Internally, we store sensor values into an array and only send them when the data packet is full. The configuration parameter for this is the window size.

Compressed Delivery of Sensed Data (CD-SD): this approach compresses the sensed data into an array to decrease the transmission rate. We selected run-length encoding (RLE) because introduces only a very low compression overhead to the nodes. RLE is used to compress sequences of values containing repetitions of the same value. The idea is to replace repeating values with just an instance of the value and a counter that counts the number of repetitions. Compressed data only needs to be sent to the sink when the array fills up or a maximum delay time is reached. Since very small signal variations may be insignificant for most applications, we also added quantization to RLE in our system, which increases compression rates in some sensitive sensors. Users may define a quantization interval so that similar values, within the boundaries of the quantization level, are considered equal (lossy compression). For instance, for a quantization level of 0.5, the measure values 23.1 and 23.2 are considered equal because they are within the same quantization level.

The workstation has to decompress the data messages to reconstruct the sensor values from the compressed stream. CD significantly reduces network traffic in scenarios with low variation of sensor readings. The configuration parameters are: the window size, the maximum delay and the quantization levels.

4 EXPERIMENTAL EVALUATION

The objective of this section is threefold: to characterize and compare alternative data processing approaches and configurations under different traffic conditions, to show that network status information given by the Network Status manager (NSManager) tests is useful to assess the processing status; and to show that excessive traffic intensity is promptly characterized by the tests.

The setup consists of a multi-hop network, as illustrated in figure 2, where the leaf nodes are set to collect sensor data with different sampling rates, and then to send these data values to a sink node, following a multi-hop path.

Intermediate nodes simply relay the data into upper nodes. All nodes implement the DP strategies (SD-SD, AD-SD, MD-SD, CD-SD), which are activated when required or instructed through a command instruction.

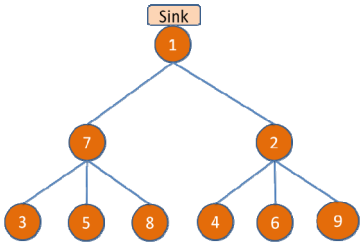

Figure 2: Node Hierarchy Diagram.

Nodes are telosB motes running Contiki 2.3 operating system with a uIPv6 stack over Contiki Rime for network communication, using a maximum packet size of 132 bits, with a 32 bits header.

Results presented below were obtained by running 10 times the experimental setup with a runtime of 30 minutes. In each run, sensor nodes are continuously sending sensor data values to the sink node at sampling rates of 50ms, 250ms, 500ms or 1s, while alarms and actuation commands were sent with end-to-end acknowledge and messages retries when necessary. Data items are processed using SD-SD, AD-SD, MD-SD or CD-SD (in any part of the path to the sink), and sent to the sink at moments according to the data processing strategy.

The discussion is organized as follows: section 4.1 discusses the effects of reliable communication applied to all messages in a high-rate setting. This result motivates the reason why we chose guaranteed delivery only to urgent messages; Section 4.2 compares measures considering different data processing strategies using the X-MAC protocol, and we analyze the differences between then.

4.1 Reliable Communication

We focused on only guaranteeing end-to-end acknowledgement for urgent messages. A natural question is why not just using reliable communication for all messages instead? Figure 3 compares the message loss ratios at high-traffic rates using two alternatives: a best-effort delivery protocol (mesh), and a reliable version (reliable multihop protocol), which requires acknowledgement (ack) on each hop, with up to 5 retries. Results show a much higher message loss ratio with the reliable protocol, due to a considerable increase in packets resulting from retries and packet drops. This was the major motivation for our dual approach of guaranteed delivery of urgent messages and best-effort delivery of the remaining sensor data.

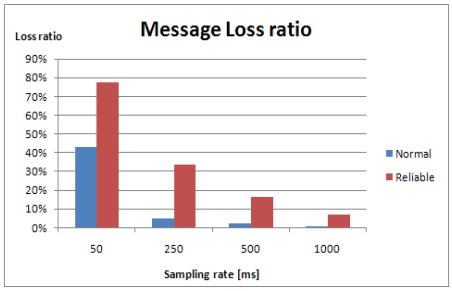

Figure 3: Message loss ratio for SD approach with reliable protocol.

4.2 Network Status and Data Processing Configurations

In this sub-section we used the Network Status test tool to study how different data processing configurations behave with traffic intensity.

Given an initial deployment, a user runs network status tests to assess whether the configuration had acceptable traffic intensity; if not, he changes some data processing configuration parameter and retries, or simply tries more than one configuration from the start to evaluate which best fits his application needs.

Message Loss Ratios: Figure 4 shows the message loss ratios obtained versus sampling rate, for varied processing strategies (SD-SD – direct sense-and-send, AD-SD – aggregate and send with window size 10, MD-SD – merge and send, with window size 10, and CD-SD – compress and send).

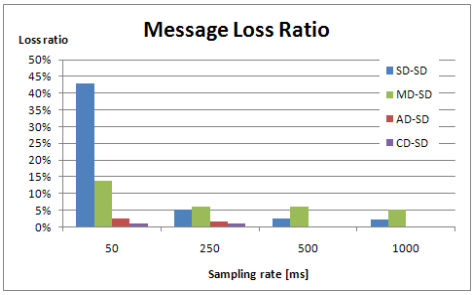

Figure 4: Evaluation of message loss ratio versus sampling rate.

SD-SD has a large loss ratio for sampling rates below 250ms. The most basic SD-SD strategy presents high message loss ratios as the sampling rate increases. With 50 ms rate almost 43% of the

messages sent by sensor nodes are lost, and this means that urgent messages will also have severe difficulty to arrive and to receive an acknowledgement, resulting in several retries.

The remaining approaches all reduce the message loss ratio significantly when compared with SD-SD. One interesting result is the enormous improvement obtained over SD-SD just by merging values and sending a single packet less frequently (MD-SD), instead of multiple packets with just one value (SD-SD). This means that, if some delay is admissible, it pays to get several pieces of data together and to send them less frequently than SD-SD. Aggregation (AD-SD) results in the same number but much smaller packets than MD-SD, since MD-SD maintains full data in large packets, while AD-SD summarizes the data into a small packet.

Compression-based CD-SD presents the lowest message loss ratios in the tested scenario, because it generates the smallest number of messages.

Window Size: in some approaches, such as AD-SD and MD-SD, the window size has an impact on message loss ratios, since it will vary the number and size of messages sent by the sensors. Figure 5 shows the influence of two different window sizes (5 and 10). An increase of the window size from 5 to 10 reduces to half the number of messages sent and consequently congestion. The figure shows that the window size had only a very moderate influence on message loss ratio for aggregation (since there is already no significant congestion), and a larger influence for MD-SD.

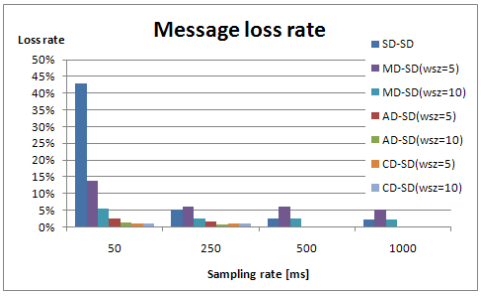

Figure 5: Evaluation of message loss ratio versus window size.

Number of Retries: the best way to evaluate whether urgent messages are deliverable in time is using a test that sends such messages (with end-to-end acknowledgement) while the system is in operation. The "Number of retries" test of the NSmanager module does exactly that. Table 1 shows the results of the test for the tested scenario (average, standard deviation and maximum number of retries over ten rounds). For each approach, the test measures the number of retransmissions that were required by a node to deliver the message correctly. The displayed results were obtained for a 5 seconds retry timeout (time between a node sends the data and waits for the ack), and a maximum of 10 retries (retries repeat until an ack is received). Low sampling rates (1 second) succeeded in sending the messages without retires for any of the approaches, while for sampling rates higher, SD-SD becomes unreliable, with a much larger number of retries than the remaining approaches.

Table 1: Number of retries required to deliver an urgent message with ack.

Sampling rate [ms]	SD-SD			AD-SD		
	Avg	Stdev	Max	Avg	Stdev	Max
50	5	3,72	10	1	1,21	3
250	3	2,1	5	0	0,52	1
500	1	0,41	1	0	0	0
1000	0	0	1	0	0	0
Sampling rate [ms]	MD-SD			CD-SD		
	Avg	Stdev	Max	Avg	Stdev	Max
50	1	0,42	2	1	1,17	3
250	0	0,2	1	0	0,43	1
500	0	0	0	0	0,04	0
1000	0	0	0	0	0	0

5 CONCLUSIONS

In this paper we have described our experience with studying data processing and network performance evaluation solutions for high-rate data in the context of an FP7 project – Ginseng, which aims at developing performance controlled WSNs to apply in critical scenarios. We have described the modules that we have developed for the industrial setting and our experimental data that tests both how variants of network-level protocols react to data processing needs, and how alternative data processing approaches can help resolve the high-rate congestion problem.

REFERENCES

Ginseng (2010). *The FP7 GINSENG project website. [Online]*. Available: http://www.ict-ginseng.eu/

Ramanathan, N., Chang, K., Kapur, R., Girod, L., Kohler, E. and Estrin, D. (2005). *Sympathy for the Sensor Network Debugger*. In ACM Sensys 2005.

Tolle, G. and Culler, D. (2005). *Design of an Application-Cooperative Management System for Wireless Sensor Networks*. In the EWSN 2005.

Ringwald, M. and Römer, K. (2007). *SNIF: A Comprehensive Tool for Passive Inspection of Sensor Networks*. In GI/ITG KuVS Fachgespräch "Drahtlose Sensornetze", pp. 59-62.

Meier, A., Motani, M., Siquan, H. and Künzli, S. (2008) *DiMo: Distributed Node Monitoring in Wireless Sensor Networks.* In MSWiM 2008, Vancouver.

Jianbo, X., Siliang, Z. and Fengjiao, Q. (2006). *A New In-network Data Aggregation Technology of Wireless Sensor Networks*. In IEEE SKG'06, Guilin, China.

Wu, X. and Tian, Z. (2006). *Optimized Data Fusion in Bandwidth and Energy Constrained Sensor Networks*. In IEEE ICASSP'06, Toulouse, France.