Run-Time Assurance for the E-care@home System

Mobyen Uddin Ahmed[1(✉)], Hossein Fotouhi[1], Uwe Köckemann[3], Maria Lindén[1], Ivan Tomasic[1], Nicolas Tsiftes[2], and Thiemo Voigt[2]

[1] Mälardalen University, Västerås, Sweden
mobyen.ahmed@mdh.se
[2] RISE SICS, Stockholm, Sweden
[3] Örebro University, Örebro, Sweden

Abstract. This paper presents the design and implementation of the software for a run-time assurance infrastructure in the E-care@home system. An experimental evaluation is conducted to verify that the run-time assurance infrastructure is functioning correctly, and to enable detecting performance degradation in experimental IoT network deployments within the context of E-care@home.

1 Introduction

Making IoT networks more dependable is critical in applications such as assisted living facilities [1]. One way towards achieving this goal is to increase the visibility into the operation of the network to developers, researchers, and system operators. In essence, it should be possible to quickly get warnings about issues that require attention in order for the system to operate correctly and with acceptable performance. Such issues can be, for example, when a node's Battery Level (BL) is low, or when the network is experiencing high packet loss rates. Another objective is that we wish to keep long-term records of the historical performance in various metrics for each IoT device to find anomalies and changes in trends of system performance.

To this end, we present a software infrastructure for *Run-Time Assurance* (RTA) in the E-care@home system [2]. We define *run-time assurance* as a service that continuously runs to discover and report system errors and performance problems. At a high level, one can summarize the core functionality of the run-time assurance infrastructure as the following three actions: (a) monitor a variety of internal and external operating conditions periodically, (b) analyze the collected data to find current performance degradations, or changes in the environment that might affect future performance, and (c) report important information to a system operator.

The challenges of building an infrastructure for RTA are threefold. First, we must identify which protocols and parameters to monitor, with low overlapping of data that describe the same condition. Second, the monitoring must be conducted with low overhead and with minimum interruption of regular

M. U. Ahmed et al. (Eds.): HealthyIoT 2017, LNICST 225, pp. 107–110, 2018.
https://doi.org/10.1007/978-3-319-76213-5_15

E-care@home application-layer data packets. Third, since the monitoring relies on a potentially faulty communication link to transmit the results, the RTA must comprise parallel monitoring efforts at the server-side.

Our run-time assurance infrastructure consists of four different components that address these challenges: (a) RTA for sensor platforms, (b) database storage of RTA information, (c) RTA at the server-side, and (d) a graphical user interface for RTA. Together, these components provide an RTA service for all parts of an e-health system, including static sensor nodes for environmental monitoring, mobile sensor nodes for health parameter monitoring, and the data collection server for the aforementioned sensor nodes.

2 Run-Time Assurance Infrastructure

The E-care@home system uses a database as a central point of storage for data collection from heterogeneous sources. The inclusion of a relational database in a remote monitoring system has been shown before not to introduce a significant latency in the system [3]. We extended the database for RTA data, including received and lost packets, timestamps between messages, as well as several statistics related to the IoT network protocols used in the E-care@home system (i.e., RPL and TSCH). Data is entered into the database from a server application that receives data from variety of sensors. We distinguish between RTA data collected on the client side and the server side.

The power consumption of a node can reveal insights as to whether it is operating correctly and efficiently. For E-care@home's environmental monitoring nodes, which use the Contiki OS, we rely on software-based power profiling [4], which provides visibility into the power consumption of various system components and software modules. We periodically measure the amount of time that (1) the radio has been in transmission mode (TX), (2) the radio has been in reception mode (RX), (3) the micro-controller has been in active mode (CPU), and (4) the micro-controller has been in Low Power Mode (LPM). The CPU time and the LPM time are both needed to calculate the percentage of time that the CPU is in active mode.

For medical health monitoring, we use Shimmer sensors that communicate using Bluetooth radio. Each Shimmer device has the ability to measure low-noise accelerometer, wide-range accelerometer, gyroscope, magnetometer, pressure, temperature, battery voltage, external expansion ADCs, and ECG/EMG. For our experiments with the RTA infrastructure, we collect sensor id, various physiological measurements, number of received packets, number of lost packets, BL (calculated at the Shimmer device), and local timestamps.

RTA packets are collected by the server together with regular sensor data packets. For the Contiki server software, we have a Python script that receives and parses messages from the IoT network's border router. For the Shimmer server software, we use Shimmer firmware, which is an application that can receive data from a Shimmer device using Bluetooth. This software acts as an API for the server-side to collect data. In order to get the raw values of BL

together with medical parameters, we modified the firmware source code to be able to send these readings to an E-care@home database, where they can be obtained for analysis and presentation to a network operator.

At the server side, we are keeping track of the following metrics related to packet reception. The *signal strength* is collected to find if the radio environment of a node deteriorates. If the deteriorated condition persists over a long time, this may indicate that the physical topology of the network should be adjusted. The *Packet Reception Rate* (PRR) shows whether the application-layer packets get through from each sensor node to the gateway server. The PRR will be affected by the quality of the radio environment, network routing, traffic load, packet buffer space, and other system-level conditions. The *communication interruption time* is collected to warn when we have not received data from a specific node in a selected time period. The PRR is insufficient to warn about communication interruption because the PRR can take a long time to change significantly, depending on the time range for packet statistics that is used to calculate the PRR.

3 Evaluation

For brevity, we show only excerpts from our experimental evaluation of RTA within the E-care@home system, including both environmental and medical sensors. Figure 1 shows the radio duty cycles for nine Contiki nodes used for environmental monitoring. The TX duty cycle is on average below 0.5% across all nodes. The amount of time in the RX mode is below 2.5% on average. The data was collected by the RTA infrastructure over 1.68 days.

To evaluate the RTA of Shimmer devices, we deployed three sensors in three different rooms, where one of them was located closer to the destination (server). Two sensors sending measurements with 512 Hz, while having low BL (average of 1778 and 1801 mV) and one sending with 32 Hz with 3761 mV average BL. Figure 2 shows the PRR fluctuation over a period of around 12 h. There are some fluctuations at the beginning of the experiment, which are due to the existence of high environmental noise during the working hours. Shimmer 1 and Shimmer 2 with higher sampling frequency were depleted faster, resulting in some fluctuations at the end of the experiment, and eventually sudden PRR reductions.

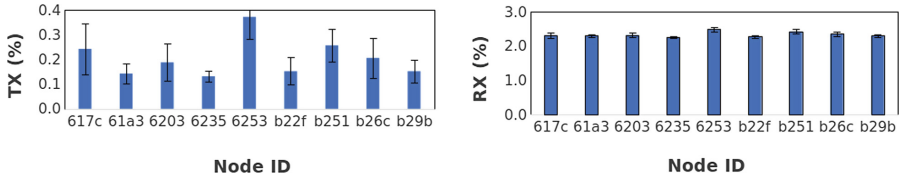

Fig. 1. TX (left) and RX (right) radio duty cycles for nine Contiki nodes. Note that the figures have different Y-axis scales.

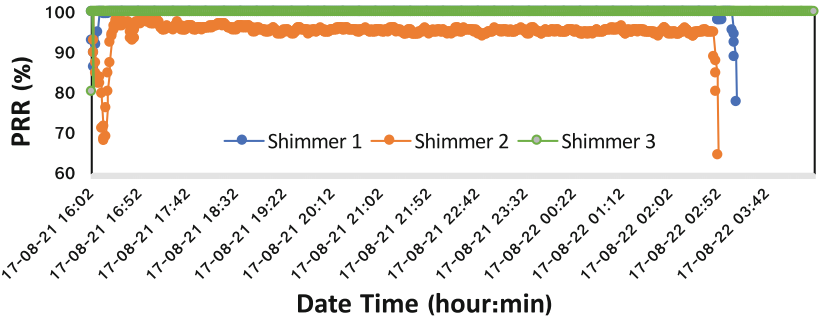

Fig. 2. Packet reception rates of three Shimmer sensors located in different rooms.

4 Conclusion

This paper describes the run-time assurance of the E-care@home system. We conducted an experimental evaluation that includes environmental sensor nodes running Contiki, and medical sensor nodes based on the Shimmer platform. We observed the packet reception rate and battery level in the Shimmer experiments. According to the experimental results, our RTA software shows the ability to monitor the PRR and battery level in real-time. Moreover, our evaluation with Contiki nodes shows that we can effectively collect important data for RTA while keeping the radio duty cycles at the same level as a system without an RTA service.

Acknowledgments. This work and the authors are supported by the distributed environment E-care@Home, funded by the Swedish Knowledge Foundation 2015–2019; and partially by the Embedded Sensor System for Health (ESS-H) research profile.

References

1. Fairbairn, M.L., Bate, I., Stankovic, J.: Improving the dependability of sensornets. In: Proceedings of Distributed Computing in Sensor Systems (DCOSS), Cambridge, Massachusetts, USA (2013)
2. Loutfi, A., Jönsson, A., Karlsson, L., Lind, L., Linden, M., Pecora, F., Voigt, T.: Ecare@Home: a distributed research environment on semantic interoperability. In: Ahmed, M.U., Begum, S., Raad, W. (eds.) HealthyIoT 2016. LNICST, vol. 187, pp. 3–8. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-51234-1_1
3. Tomasic, I., Petrović, N., Fotouhi, H., Lindén, M., Björkman, M.: Data flow and collection for remote patients monitoring: from wireless sensors through a relational database to a web interface in real time. In: Eskola, H., Väisänen, O., Viik, J., Hyttinen, J. (eds.) EMBEC 2017, NBC 2017. IFMBE, vol. 65, pp. 89–92. Springer, Singapore (2018). https://doi.org/10.1007/978-981-10-5122-7_23
4. Dunkels, A., Österlind, F., Tsiftes, N., He, Z.: Software-based on-line energy estimation for sensor nodes. In: Proceedings of the IEEE Workshop on Embedded Networked Sensor Systems (IEEE Emnets), Cork, Ireland, June 2007