Power Aware MOM for Telemetry-Oriented Applications—Levee Monitoring Use Case

Tomasz Szydlo, Piotr Nawrocki, Robert Brzoza-Woch and Krzysztof Zielinski

Abstract The paper discusses the problem of the message-oriented middleware utilization in telemetry systems. The authors provide a survey and practical measurements of common data transmission protocols for telemetry applications and wireless sensing. Based on that survey the authors propose concepts of message aggregation mechanisms to improve power consumption of the data transmission channel. As the entry point, the authors assume the utilization of the MQTT protocol. The concepts described in this paper have been successfully implemented in a smart levee monitoring system.

1 Introduction

The purpose of telemetry systems is to transparently convey measurement information from a remotely located sensor to receiving equipment for further processing and visualization. Development and miniaturization of electronic devices has allowed for the high penetration of telemetry solutions in the surrounding world in order to increase the quality of life. In typical telemetry solutions, remote stations are powered from external power sources and use industrial communication protocols such as Modbus to gather data from these devices to the central system. This work is an extended version of paper [1] presented at the Federated Conference on Computer Science and Information Systems, Warsaw, Poland, 2014.

T. Szydlo · P. Nawrocki (✉) · R. Brzoza-Woch · K. Zielinski
AGH University of Science and Technology, al. A. Mickiewicza 30,
30-059 Kraków, Poland
e-mail: piotr.nawrocki@agh.edu.pl

T. Szydlo
e-mail: tomasz.szydlo@agh.edu.pl

R. Brzoza-Woch
e-mail: robert.brzoza@agh.edu.pl

K. Zielinski
e-mail: kz@agh.edu.pl

© Springer International Publishing Switzerland 2017

M. Grzenda et al. (eds.), *Advances in Network Systems*, Advances in Intelligent
Systems and Computing 461, DOI 10.1007/978-3-319-44354-6_16

Currently, an increasing number of telemetry devices are designed to be powered by energy harvesting thus they must be power efficient and they might temporarily go asleep to preserve power [2–4]. Because of the differences between these types of devices, the legacy polling protocols for communication might not be effective. To achieve the desired functionality, we need two components: (1) an adequate communication channel and (2) a suitable communication protocol. The choice of the communication channel technology is described further in this article. In the case of the communication protocols, they should (1) leverage the power usage characteristic of the used communication technology, (2) handle the sleepy nodes and (3) provide high level addressing of nodes.

We think that the requirements for communication protocol might be fulfilled by the message oriented communication. Sending messages across channels decreases the complexity of the end application, thereby allowing the developer of the application to focus on true application functionality instead of the intricate needs of communication protocols. *Message-oriented middleware* (MOM) [5] allows application modules to be distributed over heterogeneous platforms and reduces the complexity of developing applications that span multiple operating systems and network protocols. The middleware creates a distributed communications layer that insulates the application developer from the details of the various operating systems and network interfaces. Message-oriented middleware may provide reliable asynchronous communication mechanisms that might be used to carry i.e. measurement data or other remote communication messages. We have studied and tested different communication technologies and methods. After analysis of the different MOM protocols such as AMQP (Advanced Message Queuing Protocol), MQTT or MQTT-SN (MQTT For Sensor Networks) we have decided to choose for further research the MQTT and MQTT-SN protocols.

In the current telemetry solutions we distinguish two categories of communication channels. The *external communication* concerns data transmission between any telemetry station and the Internet. The second category is the *internal communication* which may be utilized within the telemetry system, but may be unable to transmit data directly to the global network. To implement the internal communication mechanisms we utilize mesh networking hardware and protocols. In a situation where there is no possibility of the external communication, telemetry station can connect to the other stations through the internal communication. In case of no external communication availability, the data from sensor networks can be transmitted over the mesh network until a telemetry station with the external communication available is found.

In the paper we are analysing communication technologies for internal and external communication and then we are comparing the energy efficiency of XBee and GPRS (General Packet Radio Service) technology. Then we propose the concept of adaptive message aggregation method for MQTT-SN protocol that optimizes power used by GPRS wireless connection during data transmission for the external communication. The research (presented later in the paper) showed that sending data using short IP packets consumes much more energy than using longer packets. Because of the fact that messages containing measurements are relatively small, we propose the concept of adaptive data aggregation prior to sending via GPRS.

The research presented in this paper is a part of ISMOP [6] research project which objectives span construction of an artificial levee, design of wireless sensors for levee instrumentation, development of a sensor communication infrastructure, and a software platform for execution management, data management [7] and decision support [8]. Scientific and industrial consortium in the ISMOP project conducts research on a comprehensive monitoring system enabling evaluation of current and forecasted state of flood levees. This paper focuses on issues related to the organization of data acquired from the sensors located in the levees in order to optimize the power consumed by GPRS modem during data transmission to the central system for later analysis.

The paper is organized as follows. Section 2 discusses the motivating scenario, where Sect. 3 presents the related work. Section 4 contains a description of preliminary tests of the MOM (based on the MQTT protocol) power and energy requirements. Section 5 presents the concept of power-aware adaptive message aggregation, which is then evaluated in the use case in Sect. 6. Finally, the paper is summarized and further research steps are presented.

2 Motivating Scenario

Recently, the importance of sensor network for monitoring various areas, objects or devices, has significantly increased. One of the areas in which telemetry and sensor network begin to fulfill a major role is monitoring systems for hydrologic engineering facilities in particular dams and flood levees [9, 10]. The overall concept of hydrological monitoring facilities was the starting point for the assumptions and implementation of the ISMOP research project which will result in guidelines for creating a telemetry system that enables continuous monitoring of levees. Research addresses the collection of massive measurement data in continuous mode, optimized transmission methodology, interpretation and analysis of monitored data with computer simulation and finally providing visualized results for the relevant authorities.

The condition of a levee can be determined by measuring its internal temperature and pore pressure in multiple places using a large number of sensors. A threat of burst can be estimated from the gathered temperature data. A rapid change of temperature detected in one or more sensors may be a sign of a leak which can further become a dangerous burst. An example scenario of a telemetry system containing sensors placed inside the levees and the central data collection system is shown in Fig. 1.

The role of the telemetry station is to acquire data from sensor networks which contain information about levee condition and to transmit these data to the central station. The data is generated by the sensor network for an *epoch*, which results in bursts of data each time the epoch changes. Apart from that, telemetry station also sends periodically information about its current condition such as battery level, CPU

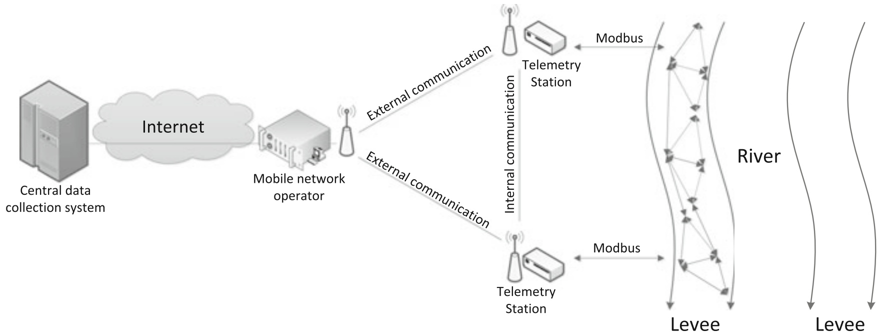


Fig. 1 Telemetry system for flood control levees

usage and others. The data from sensor networks, due to their importance, should be reliably delivered while the condition information may be transmitted on the best effort basis.

3 Related Work

Energy consumption of a wireless transmission device greatly depends on such factors as chosen communication standard, protocols used, and amount of transmitted data. Providing a medium-range or wide-area network connectivity requires a different approach. It is not a demanding task provided that a network infrastructure is available with appropriate SLA (Service Level Agreement) guaranties [11]. However, in remote areas a cellular connection is a common solution for industrial telemetry systems. Typical activities on a smartphone platform (sending a message, making a voice call, transmission over GPRS, etc.) are evaluated in [12]. An in-depth analysis o energy requirements for GPRS and UMTS services is provided in [13, 14].

High-level protocols over cellular network also have an impact on overall energy requirements of a system [15]. A review of various middleware protocols for telemetry applications can be found in [16]. Message-oriented Middleware is widely used as a communication layer for a variety of information systems which require event-driven message and data exchange, and more loose coupling than e.g. remote procedure calls. Examples of commonly utilized technologies for MOM are:

- Java Message Service (JMS) [17];
- Data Distribution Service [18];
- Extensible Messaging and Presence Protocol (XMPP) [19];
- MQTT and its variation, MQTT-SN [20].

These technologies provide several other functionalities such as transaction management, broker clustering, additional message paradigms including point-to-point, publish/subscribe and request-response. Nevertheless, only MQTT has been designed especially for transferring telemetry-style binary data from the pervasive devices with limited computational resources. It should be noted that utilizing the MQTT protocol over a standard TCP (Transmission Control Protocol) connection may provide redundant message delivery guaranties. As TCP is intended to provide a reliable link and has built-in retransmission mechanisms, setting the MQTT's QoS (Quality of Service) parameter to 1 or 2 provides another (redundant) layer of persistence. In contrast, those higher levels of QoS seem very useful in MQTT-SN variation which by design uses UDP (User Datagram Protocol) datagrams. In our research we have chosen MQTT-SN messaging protocol (formerly MQTT-S [21]) because it is promising due to its simplicity. MQTT-SN clients can be implemented in resource-constrained hardware (embedded systems), and there are available plenty of its implementations.

However, it is difficult to find any power efficiency considerations for MQTT-SN. By far many solutions dedicated to the MOM technology have been optimized to limit the data transfer and save energy. The MQTT-SN is an example of protocol that was optimized in terms of quantity of data to be transmitted. It results from using short-distance wireless protocols for sensor-based data transmission, including the IEEE 802.15.4 protocol. All these issues, not previously mentioned in MOM solutions, and particularly in the MQTT-SN protocol, have been analyzed in this article and relevant solutions have been suggested.

4 Power Aware MOM for Telemetry-Oriented System

There are several communication technologies that can be used in the telemetry-oriented applications. However monitoring levees in the hazardous weather conditions is not a trivial task and therefore needs technologies with particular properties. External communication should provide Internet connectivity with a good coverage in the rural areas. Internal communication should provide mesh connectivity as it will be used to transfer measured data to the other stations during the temporal failures of the external communication. Table 1 summarizes the available communication technologies that might be used.

Based on the aforementioned requirements, we have selected the XBee communication protocol for internal communication and GPRS for external communication. The tests described in this section concern power and energy requirements of the MOM based on the MQTT protocol implemented over these two technologies. Figure 2 shows the power consumed by the control-measurement station during four test *bursts*. Each burst consists of (a) reading data from the wireless sensor network edge router (b) processing the data, and (c) transmitting data in three different ways. The reading and processing procedures use the same underlying algorithms for each of the three cases. As can be noticed, data transmission is the main contributing factor for the overall power consumption footprint.

The XBee module's transmit operation required very little power ($\approx$0.2 W during transmission) compared to the GPRS connection ($\approx$3 W). It makes the XBee

Table 1 Communication technologies for telemetry applications

Technology	Topology	Range	Frequency	Applicability	Comments
GSM (GPRS/3G/LTE)	Star	omnipresent	850 MHz, 900 MHz, 1800 MHz, 1900 MHz	External	GPRS is a legacy technology with high area coverage
WiFi	Star	100 m to few km with high gain antenna	2.4 GHz	External/Internal	can be used for external communication when open HotSpots are available
XBee	Star, tree, mesh	Up to 10 km	2.4 GHz, 868 MHz, 900 MHz	Internal	proven technology for sensor networks
6lowPAN	Tree	100 m of direct link	868 MHz, 2.4 GHz	internal	IPv6 based technology
LoRA	Star	up to 20 km	868 MHz, 915 MHz	Internal	low power and low speed data transmissions over long distances

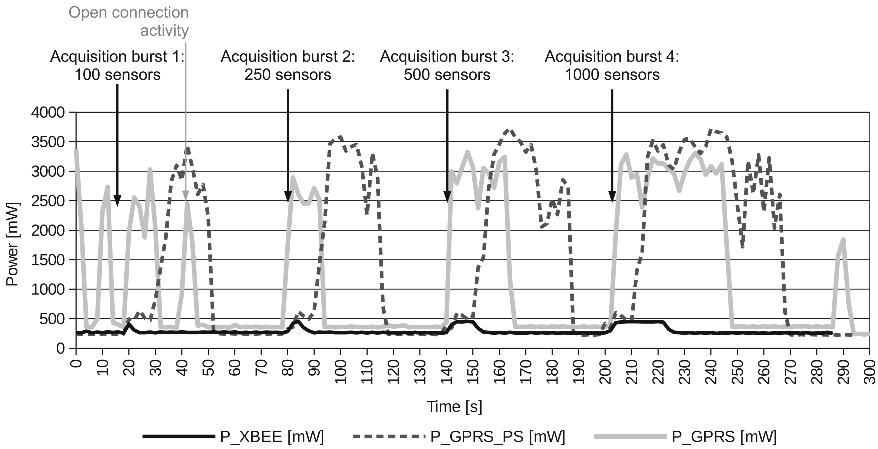


Fig. 2 Relation of power requirements in the function of time for four sample bursts

communication very well suited for the low-power data transmission in the control-measurement station and it does not require any sophisticated power saving mechanism. Despite XBee offers potentially economical station-to-station connectivity, it has no ability to transmit data to the Internet.

In contrast, the GPRS modem's power consumption is significant, but it has the advantage of providing connection to the Internet. In this case power saving mechanisms are justifiable. Further in this section we describe basic yet versatile method of implementing the GPRS power saving mechanisms at very low level. The mechanisms include (a) aggregating data to be transmitted (b) optionally completely disabling the power of the GPRS modem for a period of time. The latter requires the cold-start procedure to be performed each time the GPRS has to transmit data, but it saves the most power during the idle state. Then we compare the two methods.

The gathered information allowed us to determine the relationship between data aggregation time and energy that needs to be provided (harvested) in a period of time. To achieve this, we created a simplified energy model of the transmission subsystem with and without power saving (PS) feature.

First, we define the following symbols:

- T_{AG}—the data aggregation time (transmission interval), i.e. how long we wait and collect samples until transmission occurs (as one *sample* we mean one burst of data from all sensors);
- T_{SA}—sampling period, i.e. time interval between subsequent samples;
- T_1—one sample transmission time;
- T_{TOT}—total measurement period;
- E_{TOT_PS}—total energy required to transmit all data gathered during T_{TOT} with PS feature enabled (GPRS modem is powered down between transmissions);
- E_{TOT_ON}—total energy required to transmit all data gathered during T_{TOT} without the PS feature (GPRS modem is always powered and keeps the TCP connection);
- E_{TX_PS}, E_{TX_ON}—energy required for one transmission of aggregated samples with and without PS feature.

According to our model, we express total energy E_{TOT} for the two cases with the following equations:

$$E_{TOT\_PS} = E_{TX\_PS} \cdot N_{TX} \tag{1}$$

and

$$E_{TOT\_ON} = E_{TX\_ON} \cdot N_{TX}, \tag{2}$$

where N_{TX} is the number of transmissions that we need to perform during the total measurement period. It can be determined by the formula:

$$N_{TX} = \frac{T_{TOT}}{T_{AG}}. \tag{3}$$

The number N_{TX} decreases with the increasing aggregation period.

Formulas for the energy required to perform one aggregated transmission (E_{TX_PS}, E_{TX_ON}) depend on the selected power scheme.

Refer to Fig. 3a. For the power-saving scheme, the E_{TX_PS} can be expressed as a sum the energy required to perform the GPRS modem start-up procedure E_{START} (one time) and the energy required to transmit one sample (E_{SA}) multiplied by the number of samples (N_{SA}):

$$E_{TX\_PS} = E_{START} + N_{SA} \cdot E_{SA} \tag{4}$$

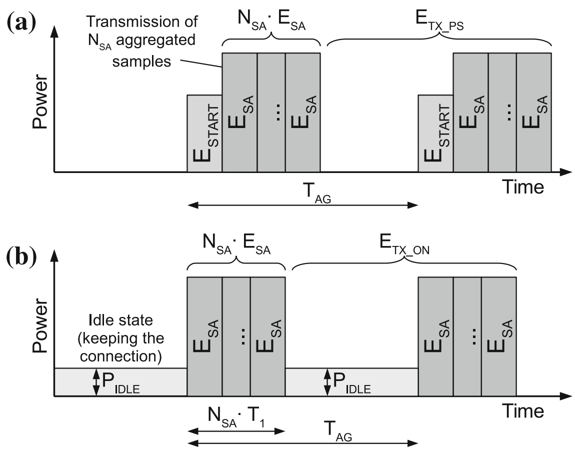


Fig. 3 Schematic representation of the two power schemes: power saving (**a**) and always-on (**b**)

The number of samples to transmit (N_{SA}) is equal to the ratio of the aggregation time to the one sample period:

$$N_{SA} = \frac{T_{AG}}{T_{SA}} \tag{5}$$

Using Eqs. 1, 3 and 4 we get:

$$E_{TOT\_PS} = E_{START} \cdot \frac{T_{TOT}}{T_{SA}} + E_{SA} \cdot \frac{T_{TOT}}{T_{SA}}. \tag{6}$$

Refer to Fig. 3b. For the always-on scheme, we express the energy needed for the one aggregated transmission as a sum of energy needed to keep the connection active and the no-overhead data transmission energy:

$$E_{TX\_ON} = (T_{AG} - N_{SA} \cdot T_1) \cdot P_{IDLE} + E_{SA} \cdot N_{SA} \tag{7}$$

The first element is expressed as the idle power (P_{IDLE}) multiplied by the time in which we need to keep the connection opened and we not yet transmit data.

Then, from (2), (3) and (7), we get:

$$E_{TOT\_ON} = P_{IDLE} \cdot T_{TOT}(1 - N_{SA} \cdot \frac{T_1}{T_{AG}}) + E_{SA} \cdot \frac{T_{TOT}}{T_{SA}} \tag{8}$$

Figure 4 shows a sample relationship between the aggregation time T_{AG} and total energy required in a period of time with and without the power saving feature (E_{TOT_PS} and E_{TOT_ON}). The input values are based on the actual working implementation: $T_{TOT} = 24$ h, $T_1 = 42$ s, $T_{SA} = 2$ min, $E_{START} = 47$ J, $E_{SA} = 113$ J, $P_{IDLE} = 0,273$ W. Depending on the input values which represent given system characteristics, the E_{TOT_PS} and E_{TOT_ON} traces may vary. The E_{TOT_ON} trace is constant,

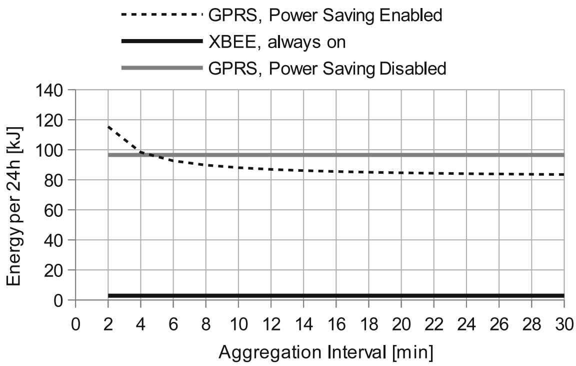


Fig. 4 Relation between the aggregation time and total energy required by the GPRS transmission subsystem of the prototype telemetry station

because the amount of data to be transmitted and idle time do not change, i.e. the transmission state to idle state ratio are constant and, depending on the aggregation interval (T_{AG}), transmission and idle periods are only differently organized. In contrast, the energy requirements with the PS feature enabled decreases with the growing number of the aggregated samples thanks to reduction of the number of cold start sequences of the GPRS modem.

5 Adaptive Message Aggregation for GPRS Connectivity

The main goal of the research was to decrease the amount of energy necessary to send the data using MOM over GPRS connectivity. The results of the base research aimed to analyze how much energy is used by GPRS modem as a function of packet data size is depicted in Fig. 5.

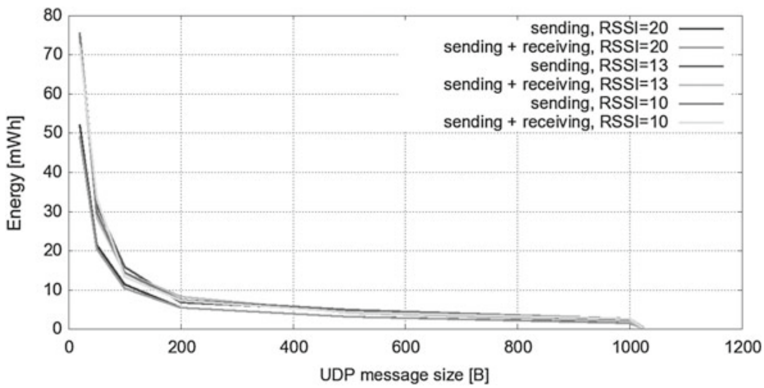


Fig. 5 Power necessary to send 10 kB of data using GPRS communication as a function of packet size

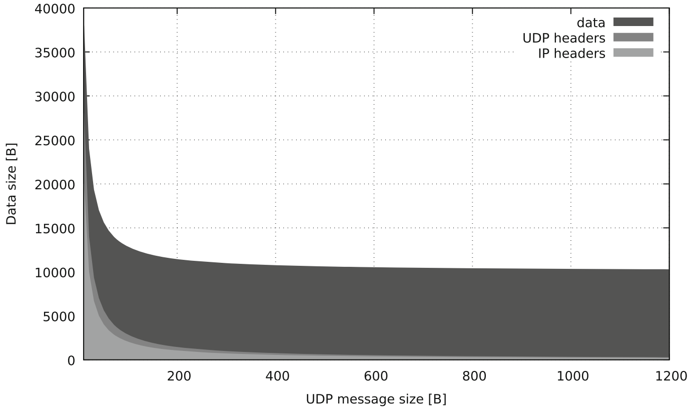


Fig. 6 Overall data necessary to send 10kB of data as a function of packet size

The nonlinearity in the power consumption is caused by two factors: overhead of the appropriate headers of TCP/IP protocol stack and purely technical considerations related to the physical communication with the GPRS modem in embedded devices (e.g. the time of data preparation, inter frame gaps and others). Figure 6 shows the overall data size that are necessary to send 10 KB of data payload. The overhead is related to the headers of UDP and IP protocols for each packet. Consequently, from an energy consumption point of view, it is much better for fixed amount of data to send it using as large packet size as possible.

On the other hand, the amount of measurement data that need to be transmitted is usually small i.e. typically 20 B. The previous measurements show that sending such small packets would be inefficient. Based on these data, we propose the concept of data aggregation before transmission.

Our concept, as presented in Fig. 7, can be applied to MQTT messages with QoS 0 and 1. In QoS 0, messages are not acknowledged, so client may aggregate several messages before sending them. In the second case, with QoS 1, all the messages had to be acknowledged so, we propose also aggregating the acknowledgment packets on the broker side.

We have assumed that larger, important data to send appear in the bursts, while less important data are sent on regular basis in small chunks. This is dictated by the fact that underlying sensor network (installed in the levee) wakes up in time intervals to preserve power. The naive approach to data aggregation is presented in Fig. 8. The main idea is that new messages are not send immediately but first copied to the buffer B with fixed length L and then sent as an aggregated packet. There are two conditions that decide of sending data: buffer is overflowed or buffer timeout T^I has ended. During the *period 1* and *3* aggregated messages are sent because of the timeout condition, while during the *period 2* aggregated message is sent because of the overflow condition. The drawback of the method is that in the *period 3*, the messages that belong to the burst are sent with longer delay then previous ones because

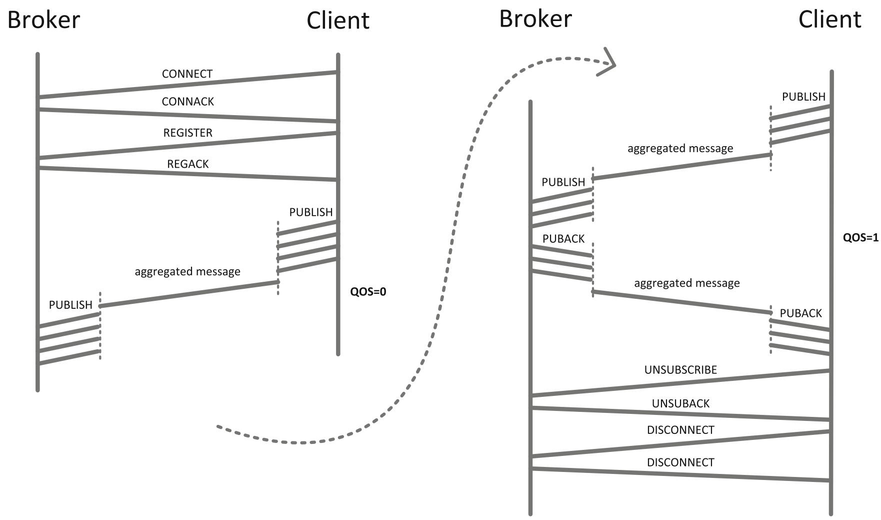


Fig. 7 Message aggregation concept for MQTT-SN

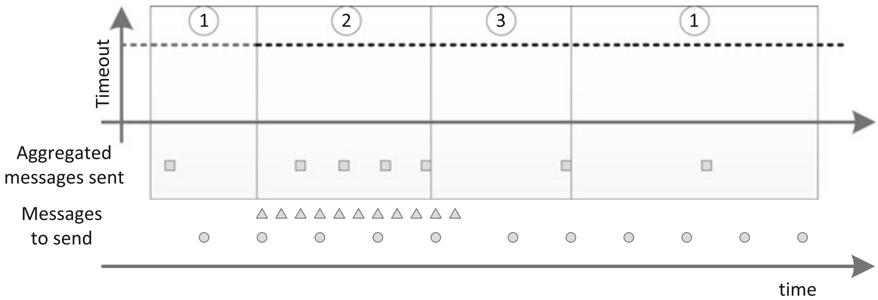


Fig. 8 Naive approach to data aggregation (time periods are marked with *numbers*, telemetry data are represented by *small grey circles*, *triangles* represent bursts of measurement data, packed and transmitted data are depicted as *squares*)

overflow condition did not occurred. Such a situation is unwanted if the data has to be analyzed in the real-time.

In our method, we propose adaptive timeout calculation that adjust itself to the frequency of incoming data. The main concept is that buffer overflow situation decreases the buffer timeout meaning that messages should be send faster, while decreasing the frequency of incoming new data recovers the timeout to its previous value. Such a policy results in the situation that messages belonging to the data burst are received by the broker in the burst as well. The concept is depicted in Fig. 9. During the *period 3*, the aggregated message is send earlier than in naive approach to the aggregation.

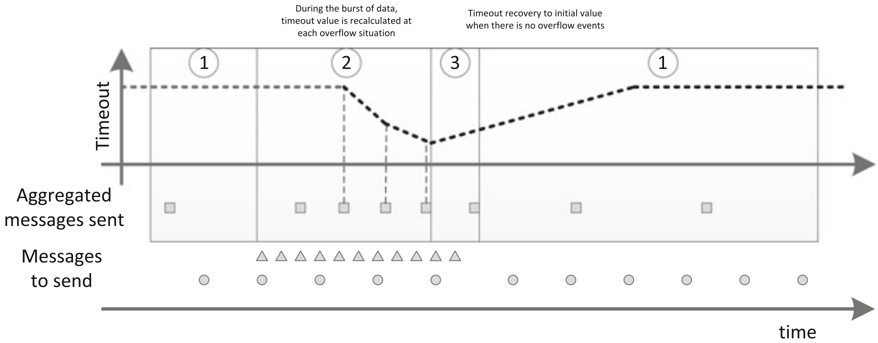


Fig. 9 Adaptive approach to data aggregation (time periods are marked with *numbers*, telemetry data are represented by *small grey circles*, *triangles* represent bursts of measurement data, packed and transmitted data are depicted as *squares*)

As can be noticed, the presented adaptive message aggregation does not generate any additional overhead. On the contrary, it contributes to reducing the overhead introduced by the TCP/IP protocol stack when transmitting small chunks of data. Another aspect is the latency of data arrival. It results directly from the fact, that data needs to be packed in a buffer. In our solution, the latency is controlled with the variable timeout value. The timeout automatically decreases as more data arrives. This can be utilized to keep latency and overhead ratio at reasonable levels.

More formally, the algorithm is composed of the two parts and might be presented as follows. The input to the algorithm is provided by four values: T^I is the initial timeout value, T^R is the recovery time to the initial value, the factor α, and a buffer length L. In the MOM client there is global timer T that represents actual timeout value—at time t this value is denoted as T_t. When the aggregated message buffer B of length L is created at time t, it has assigned timeout that equals T_t. Length of the buffers is constant.

First part of the algorithm is responsible for recovering (i.e. increasing) timeout T to the initial value of T^I and is formulated as follow: for each time k, the timeout T_{k+1} is calculated using the Eq. 9, where ΔT is the time step.

$$T_{k+1} = \min\left(T_k + \frac{T^I}{T^R}\Delta T, T^I\right) \tag{9}$$

The second part of the algorithm is responsible for decreasing timeout T to the value that is similar to the time of sending overflowed buffers when data burst is observed. The Eq. 10 is used only when the overflow of the buffer is observed. Value d in the equation is the time from the last overflow event.

$$T_{k+1} = \min\left(\alpha T_k + (1-\alpha)d, T^I\right) \tag{10}$$

Having in mind, that data usually comes from remote telemetry stations to the central point, we propose to use adaptive aggregation method on the client side to aggregate messages, and to use naive aggregation approach on the broker side to aggregate acknowledgments. The evaluation results of the proposed algorithm are presented in the next section.

6 Evaluation

We have evaluated the proposed concept on the scenario similar to the one presented in the previous section. We have assumed that data from levee monitoring sensors are gathered and sent in two stages:

- at the beginning, for QoS 1 in MQTT/MQTT-SN, 1000 PUBLISH messages with a length of 20 B are sent and received confirmation of these messages (PUBACK);
- later, for QoS 0 in MQTT/MQTT-SN, in 12 min epoch and for every 30 s PUBLISH messages with a length of 20 B are transmitted.

During tests we used a popular GPRS modem (SIM900D) and, in order to verify the results obtained, we also used an industrial GPRS Modem (Wavecom Fastrack Supreme 20). In order to develop test software we extend implementation of MQTT-SN— Eclipse Mosquitto [22] (which we call *A-MQTT-SN*) to support adaptation.

Above presented testing scenario was carried out for three cases using:

- MQTT protocol (Eclipse Mosquitto);
- MQTT-SN protocol (Eclipse Mosquitto);
- A-MQTT-SN protocol with adaptation for sent and received data (message type PUBLISH and PUBACK).

The adaptive aggregation algorithm for A-MQTT-SN was initiated with values: $TI = 120$ s, $TR = 240$ s, $\alpha = 0.5$, and $L = 1000$ B. The naive aggregation algorithm was initiated with values: $T_I = 2$ s and $L = 250$ B. The values are application-specific, and should be tailored for different conditions, such as: amount of transmitted data, real-time boundaries and the maximal accepted latency by the application.

The measurements were made with a custom multichannel current and voltage sensing module and tailored for energy measurements of various embedded devices. Data for all of the presented tests was acquired from the GPRS modems (Class 10) connecting to public GSM network with a throughput of 25 Kb/s (2 timeslots in uplink direction).

The result of these tests is shown in the following figures:

- for MQTT protocol (using TCP and PPP protocols)—Fig. 10;
- for MQTT-SN protocol (using UDP, PPP protocols and AT commands on the GPRS modem)—Fig. 11;
- for A-MQTT-SN protocol (using UDP, PPP protocols and AT commands on the GPRS modem)—Fig. 12.

The above results show that the power consumption of a GPRS modem for data transmission is higher for MQTT and MQTT-SN than A-MQTT-SN protocol. In our

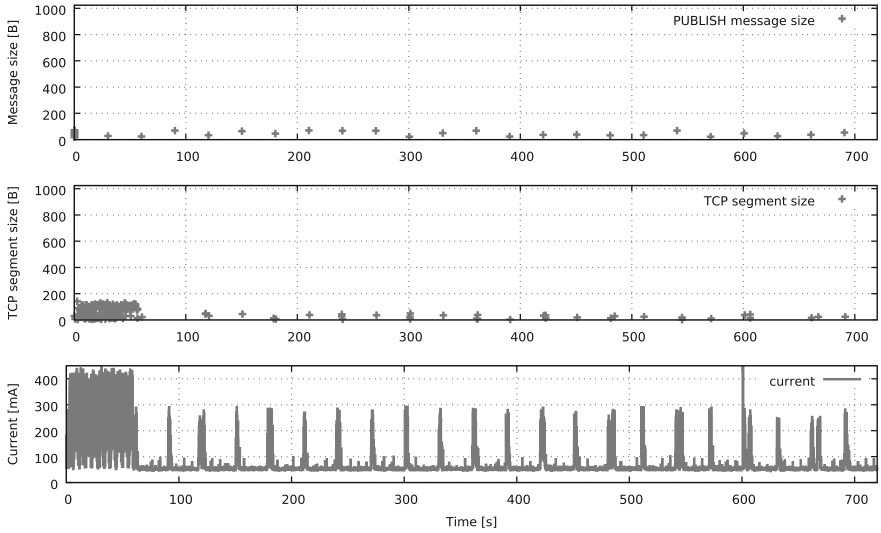


Fig. 10 The current consumption for the MQTT protocol

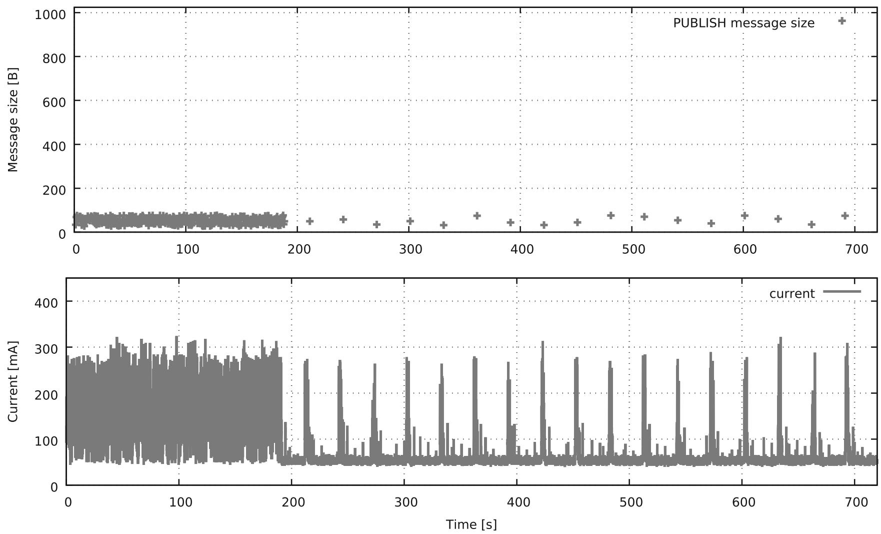


Fig. 11 The current consumption for the MQTT-SN protocol

opinion, the higher value of power consumption for MQTT protocol is the result of using TCP and its complexity (call setup, retransmissions). When we transmit MQTT messages with QoS 0, there should be no retransmissions at the MQTT protocol level. However, the retransmissions may actually occur at the TCP protocol level. In the second case the increased energy consumption of the GPRS modem is related

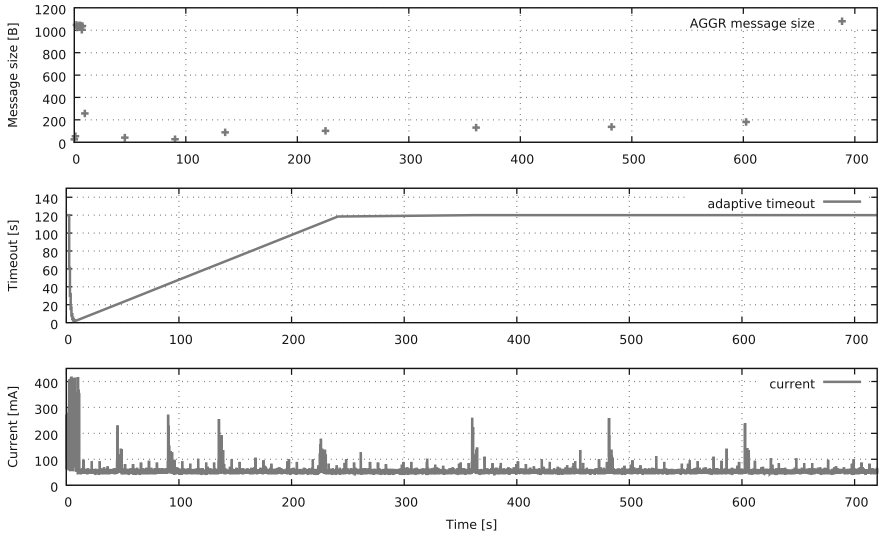
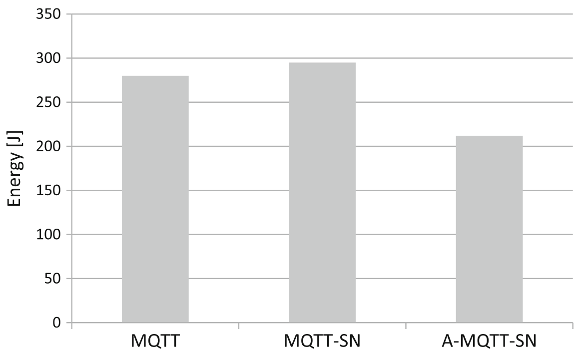


Fig. 12 The power consumption for the A-MQTT-SN protocol

Fig. 13 Energy consumption for MQTT, MQTT-SN and A-MQTT-SN protocols



to the size of transmitted data chunks with the MQTT-SN over UDP. Small chunks of data result in an increased overhead related to the packets' headers. The developed A-MQTT-SN protocol variation aggregates data and significantly decreases the number of headers that need to be transmitted. This, in turn, provides the best energy efficiency (Fig. 13).

7 Summary and Future Work

The paper discusses the problem of sending the sensor data from and between remote telemetry stations. We have analyzed various communication technologies and protocols for both internal and external communication and decided to use XBee and GPRS connectivity. The paper also proposed the extensions to the communication protocol that adjust its behavior to the GPRS connectivity profile in order to decrease the data transmission-related energy consumption.

The motivating scenario presented in the paper is only one of the possible applications of our concept. The solutions might be successfully applied to e.g. multilayer telemetry solutions where due to the sleepy nodes, data have to be pushed rarely but efficiently.

Acknowledgements The research presented in this paper was partially supported by the National Centre for Research and Development (NCBiR) under Grant No. PBS1/B9/18/2013 and by the Polish Ministry of Science and Higher Education under AGH University of Science and Technology Grant 11.11.230.124 (statutory project).

References

1. Szydlo, T., Nawrocki, P., Brzoza-Woch, R., Zielinski, K.: Power aware MOM for telemetry-oriented applications using gprs-enabled embedded devices - levee monitoring use case. In: Proceedings of the 2014 Federated Conference on Computer Science and Information Systems, Warsaw, Poland, 7–10 Sept 2014, pp. 1059–1064 (2014). doi:10.15439/2014F252
2. Szydlo, T., Brzoza-Woch, R.: Predictive power consumption adaptation for future generation embedded devices powered by energy harvesting sources. Microprocess. Microsyst. Embed. Hardw. Des. **39**(4–5), 250–258 (2015). doi:10.1016/j.micpro.2015.05.001
3. Szydlo, T., Gut, S., Puto, B.: Smart Applications: Discovering and interacting with constrained resources IPv6 enabled devices. Przeglad Elektrotechniczny, 221–226 (2013)
4. Szydlo, T., Suder, P., Bibro, J.: Message oriented communication for IPV6 enabled pervasive devices. Comput. Sci. **14**(4) (2013). doi:10.7494/csci.2013.14.4.667
5. Curry, E.: Message-oriented middleware. In: Mahmoud, Q.H. (ed.) Middleware for Communications, Chap. 1, pp. 1–28. Wiley, Chichester, England (2004). doi:10.1002/0470862084.ch1
6. ISMOP Project (2013). www.ismop.edu.pl. Accessed 30 Nov 2015
7. Piórkowski, A., Leśniak, A.: Using data stream management systems in the design of monitoring system for flood embankments. Studia Informatica **35**(2), 297–310 (2014)
8. Chuchro, M., Lupa, M., Pięta, A., Piórkowski, A., Leśniak, A.: A concept of time windows length selection in stream databases in the context of sensor networks monitoring. In: New Trends in Databases and Information Systems, Proceedings of 18th East-European Conference on Advances in Databases and Information Systems (in print). Advances in Intelligent Systems and Computing. Springer (2015)
9. Balis, B., Bartynski, T., Bubak, M., Dyk, G., Gubala, T., Kasztelnik, M.: A development and execution environment for early warning systems for natural disasters. In: 2013 13th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), pp. 575–582 (2013). doi:10.1109/CCGrid.2013.101
10. Balis, B., Kasztelnik, M., Bubak, M., Bartynski, T., Gubaa, T., Nowakowski, P., Broekhuijsen, J.: The urbanflood common information space for early warning systems. Proc. Comput. Sci. **4**(0), 96–105 (2011). http://dx.doi.org/10.1016/j.procs.2011.04.011. Proceedings of the International Conference on Computational Science, ICCS 2011
11. Kosinski, J., Nawrocki, P., Radziszowski, D., Zielinski, K., Zielinski, S., Przybylski, G., Wnek, P.: SLA monitoring and management framework for telecommunication services. In: Bi, J., Chin, K., Dini, C., Lehmann, L., Pheanis, D.C. (eds.) Fourth International Conference on

Networking and Services, 2008. ICNS 2008, pp. 170–175. IEEE Computer Society (2008). doi:10.1109/ICNS.2008.31
12. Perrucci, G.P., Fitzek, F.H., Widmer, J.: Survey on energy consumption entities on the smartphone platform. In: Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd, pp. 1–6. IEEE (2011). doi:10.1109/VETECS.2011.5956528
13. Sikora, A., Yunitasari, A., Dold, M.: GPRS and UMTS services for ultra low energy M2M-communication. In: 2013 IEEE 7th International Conference on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS), vol. 1, pp. 494–498. IEEE (2013). doi:10.1109/IDAACS.2013.6662734
14. Pauls, F., Krone, S., Nitzold, W., Fettweis, G., Flores, C.: Evaluation of efficient modes of operation of GSM/GPRS modules for M2M communications. In: Vehicular Technology Conference (VTC Fall), 2013 IEEE 78th, pp. 1–6. IEEE (2013). doi:10.1109/VTCFall.2013.6692200
15. Alonso, E.J.V.: Exploiting energy awareness in mobile. Communication (2013). doi:10.3384/lic.diva-98656
16. Azzara, A., Bocchino, S., Pagano, P., Pellerano, G., Petracca, M.: Middleware solutions in WSN: the IoT oriented approach in the ICSI project. In: 2013 21st International Conference on Software, Telecommunications and Computer Networks (SoftCOM), pp. 1–6. IEEE (2013). doi:10.1109/SoftCOM.2013.6671886
17. Java Message Service Specification. https://jcp.org/en/jsr/detail?id=343. Accessed 30 Nov 2015
18. Data Distribution Service ver. 1.2 documentation. http://www.omg.org/spec/DDS/1.2. Accessed 30 Nov 2015
19. Extensible Messaging and Presence Protocol documentation. http://xmpp.org/xsf/press/2004-10-04.shtml. Accessed 30 Nov 2015
20. MQ Telemetry Transport (MQTT) documentation. http://mqtt.org/documentation. Accessed 30 Nov 2015
21. Hunkeler, U., Truong, H.L., Stanford-Clark, A.: MQTT-SA publish/subscribe protocol for wireless sensor networks. In: 3rd International Conference on Communication Systems Software and Middleware and Workshops, 2008. COMSWARE 2008, pp. 791–798. IEEE (2008). doi:10.1109/COMSWA.2008.4554519
22. Mosquitto technology project. http://projects.eclipse.org/projects/technology.mosquitto. Accessed 30 Nov 2015