



<http://www.diva-portal.org>

Preprint

This is the submitted version of a paper presented at *Workshop on Protocols, Applications and Platforms for Enhanced Living Environments, PAPELE 2017, Held in conjunction with IFIP/IEEE International Symposium on Integrated Network Management IM 2017 (Lisbon // Portugal)*.

Citation for the original published paper:

Lindh, T. (2017)

Real-time Performance Management of Assisted Living Services for Bluetooth Low Energy Sensor Communication.

In:

N.B. When citing this work, cite the original published paper.

Permanent link to this version:

<http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-211034>

# Real-time Performance Management of Assisted Living Services for Bluetooth Low Energy Sensor Communication

Jonas Wåhslén

School of Technology and Health  
KTH  
Stockholm, Sweden  
Jonas.Wahslen@sth.se

Thomas Lindh

School of Technology and Health  
KTH  
Stockholm, Sweden  
Thomas.Lindh@sth.kth.se

**Abstract**— **PerfMon is a prototype implementation of a real-time performance management method for sensor data communication in assisted living applications. It is implemented in accordance with the specification for GATT services in Bluetooth low energy (BLE). PerfMon provides a tool for real-time performance monitoring and control for caregivers and service providers. Test results from monitoring and control of packet loss ratio related to alarm thresholds are presented. PerfMon is adapted to cloud-based web services using RESTful APIs and established object models. Performance management is a necessary component in an overall management system of IoT devices for healthcare and assisted living applications.**

**Keywords**—*performance monitoring, performance management, Internet of Things, Bluetooth low energy, Generic Attribute (GATT) service, ambient assisted living.*

## I. INTRODUCTION AND MOTIVATION

The rapid development of wireless sensor network technologies has in several ways revolutionized healthcare and different forms of ambient assisted living environments. Wearable wireless devices and sensors enable and facilitate measurements and monitoring of parameters related to health, movements, wellness and other personal conditions. The reasons for the increasing interest in ambient assisted living applications are twofold; the well-known demographic shift with growing share of elderly in the population, and the rapid development of wirelessly connected embedded sensor devices in combination with efficient IT and storage infrastructures. The European initiative Active and Assisted Living Joint Programme (AAL JP) supports applied research on ICT-enhanced services for ageing [1]. EIT Health is another European project for active ageing [2]. The European COST action IC1303 Algorithms, Architectures and Platforms for Enhanced Living Environments (AAPELE) is a third example of research in this field.

Deployment, operation and maintenance of wireless sensors and devices require appropriate management systems for reliable and efficient health and AAL services. There are several initiatives to standardize remote management of Internet of Things (IoT) as well as traditional machine-to-machine (M2M) devices. In this paper we present design and prototype implementation of performance management methods of Bluetooth low energy (BLE) IoT devices. It pursues previous research on Wi-Fi [3] and IEEE 802.15.4 [4].

The main motivation for the work is that performance feedback and control is of vital importance for healthcare and AAL services. The paper is organized as follows: a survey of IoT management in Section II; a brief overview of Bluetooth low energy in Section III; the overall design of the performance monitoring method in Section IV; some use cases in Section V; implementation of a Generic Attribute (GATT) service for performance monitoring, examples of control actions and web-based communication in Section VI; the results and discussion in Section VII; and, finally the conclusions in Section VIII.

## II. RELATED WORK ON MANAGEMENT OF IOT DEVICES

Wirelessly connected devices are becoming ubiquitous in our environments and daily lives. Different names and notations, such as wireless sensor nodes, machine-to-machine and Internet of things, are used in various application areas and scientific fields. Wireless connectivity for these devices can be provided by a wide range of network and radio techniques: wireless personal area networks (e.g. Bluetooth and IEEE 802.15.4); wireless local area networks (Wi-Fi and Wi-Fi Direct [5]), cellular low power wide area (LPWA) networks (e.g. EC-GSM, NB-IoT and LTE Cat M1 [6]); and, proprietary low-power wide area networks, LPWA (e.g. LoRa [7] and Sigfox [8]).

One or several of these connectivity solutions may constitute the communication infrastructure for enhanced assisted living environments. Operators and providers of care at home and related services need efficient and standardized methods to remotely operate, maintain and control their devices and services. There are several initiatives to enable end-to-end IP communication to devices or gateways. Lightweight Machine-to-Machine (LWM2M) [9] from Open Mobile Alliance is a candidate for a unified protocol for remote management of cellular M2M appliances as well as devices from the IoT industry as a whole. LWM2M servers and clients (M2M or IoT devices) utilize the Constrained Application Protocol (CoAP) [10], and the underlying DTLS/UDP for transport. The CoAP protocol, defined by the IETF Constrained RESTful Environment (CoRE) working group [11], creates an alternative to HTTP for devices with limited resources to apply the web methods GET, POST, PUT and DELETE. IPv6 communication for constrained IoT devices is enabled by 6LoWPAN [12], e.g. in IEEE 802.15.4, ZigBee and Bluetooth networks. In addition to a standardized communication protocol, an

information or data model is needed to describe the objects in the LWM2M clients (the IoT devices). A LWM2M client consists of one or several objects and related resources. These objects and resources in the LWM2M data model are mapped to an URI path hierarchy where the path components represent */ObjectID/Instance ID/ResourceID*. A client can e.g. read a temperature value (ID=5700) on an instance (ID=0) of an object (ID=3303) using the URI */3303/0/5700*. This request is then mapped to the GET method of the CoAP protocol. The data model comprises a set of data types, operations (read, write, set etc.) and content formats [9]. IPSO (IP for Smart Objects alliance) has released a list of Smart Object IDs and Resource IDs which conform to the OMA LWM2M data model [13]. The LWM2M communication protocol stack and related data model can be extended with objects and resources that are relevant for healthcare, wellness and other services in enhanced assisted living environments. Large-scale deployment of IoT devices will definitely require management systems to monitor, maintain and control equipment, performance and services. LWM2M is a feasible candidate for a building block in a SCADA (supervisory control and data acquisition) management system for IoT devices and services. An alternative to CoAP for LWM2M is MQTT (MQ telemetry transport) [14], which is a widely used protocol in M2M applications. Unlike CoAP, MQTT relies on TCP on the transport layer, which may exclude resource-constrained sensor nodes.

Several projects, such as UniversAAL, OpenAAL and PERSONA, have developed IT systems and platforms to support ambient assisted living. However, performance and mobility issues have been studied insufficiently [15].

### III. BLUETOOTH LOW ENERGY

Today, a large number of IoT devices related to healthcare and smart homes support Bluetooth or Wi-Fi connectivity. It means that mobile phones and tablets can act as hubs or gateways for these IoT devices. This paper presents and evaluates a method for performance monitoring and control of devices connected via Bluetooth low energy (BLE) to a mobile phone as gateway. BLE (version 4.0-4.2) [16], also branded as Bluetooth Smart by Bluetooth Special Interest Group (SIG), is targeted for the growing IoT and smartphone markets. The priority is to minimize energy consumption by reducing complexity, functionality and throughput. A brief overview of relevant BLE features is given in this section.

#### A. GATT Servers and Clients

On the application level, BLE devices use so called profiles to exchange data based on the upper layer protocol Generic Attribute Profile (GATT). The position of GATT in the BLE protocol stack is shown in Fig. 1. A *profile* describes and specifies a use case application. Bluetooth SIG has published a number of BLE profiles related to healthcare, such as the Glucose profile and the Heart rate profile [17]. Guidelines for development of custom profiles and interoperability with public profiles are also available [18]. A BLE IoT device acts as a GATT server and a BLE enabled gateway, such as a smartphone, acts as a GATT client. Profiles consist of *services* that group user data in so called *characteristics*. A characteristic contains at least two *attributes*: a declaration and a value. Attributes are the basic data unity defined in the ATT (attrib-

ute) protocol (Fig. 1), reused by the GATT protocol. Attributes can conceptually be seen as addressable data on a GATT server, which a GATT client can access, read and also modify. An attribute, the building block of a characteristic, has four fields: a *handle*, a 16-bit identifier; a *type*, which is a 128-bit universally unique identifier (UUID); *permissions*; and, a *value*, which is the actual data content declared in the type field. A service exposed by a GATT server consists of data organized as characteristics defined according to the attribute definition above. A GATT client can read and write characteristics from a BLE sensor's GATT server. In addition, a GATT server can, without being requested by a client, send server-initiated updates to a client; *indications*, which need to be confirmed by the client, or *notifications*, which are unacknowledged by the client.

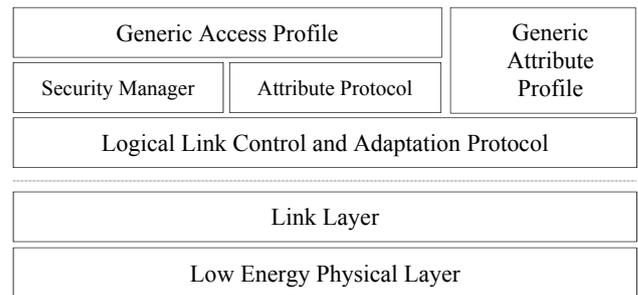


Fig. 1. Bluetooth low energy single mode protocol stack. The dotted line marks the Host Controller Interface (HCI).

The Generic Access Profile (GAP) [16], an upper layer protocol (Fig. 1), specifies how devices interact on the lower layers, such as discovery, connection establishment and security. A non-connected BLE device can act as a *broadcaster* and send *advertising* packets or, act as an *observer* and *scan* for advertising packets. A BLE observer device can act as a *central* (master) and initiate a connection with a *peripheral*, which corresponds to a slave on the link layer. There is no direct mapping between these GAP roles and the GATT server and GATT client roles.

The physical and link layers have been significantly reengineered to obtain low energy consumption, faster connection set-up times and extended range. Fewer and wider frequency bands, higher modulation index, fewer (three) advertising channels, shorter and uniform packet formats, less overhead, reduced error and flow control, no support for voice communication, and no need for continuous polling are some of the new features. As in Bluetooth BR (basic rate) and EDR (enhanced data rate), the stop and wait procedure and retransmissions after missing acknowledgements are still applied in the BLE link layer.

#### B. Internet Connectivity

BLE devices can be connected to Internet and web applications via gateways. Bluetooth SIG has published guidelines for gateways [19] and specifications for GATT REST APIs [20] and GAP REST API [21] that enable a gateway to expose a RESTful API for web services. An application can thereby use ordinary HTTP methods to connect and disconnect devices, and read and write data on the GATT server. The unified data model for IoT objects from IPSO Alliance can also be used to

retrieve information from BLE devices. IPSO has published a library for using smart objects on Bluetooth low energy devices. This BIPSO library defines a set of BLE characteristics that follows the IPSO Smart Object guideline for developers to build applications with a unified data model [22].

In versions 4.1 and 4.2, direct Internet connectivity has been added to the BLE specifications. The Internet support profile (IPSP) [23] defines a node role and a router role. A node (GATT server) shall implement an IP support service (IPSS) instance that enables a GATT client to discover the IP service. Notice that the upper left part of Fig. 2 is only used for discovery. The IPv6-based communication part in Fig. 2 is specified in RFC 7678 (IPv6 over Bluetooth Low Energy) [23]. It builds on the same protocol, 6LoWPAN, for IP communication in IEEE 802.15.4 networks.

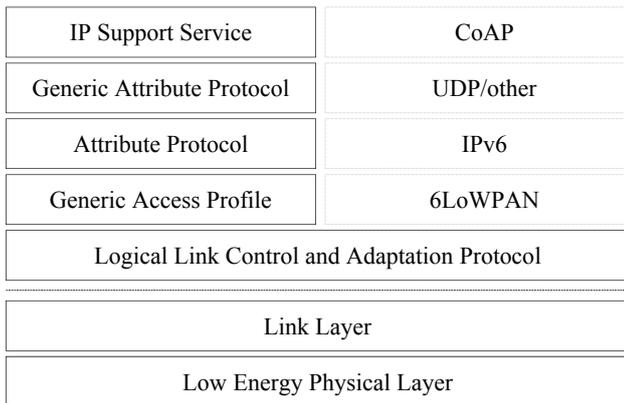


Fig. 2. Protocol stack for IP support profile node for Internet connectivity. The left upper part is specified by Bluetooth SIG. The right upper part is specified in RFC 7868, RFC 7252 and RFC 4944.

The native way to communicate between BLE devices is to use the GATT profiles and services. Even though the Internet support profile and IPv6 for BLE is introduced, the GATT profiles will still be a preferred choice by many vendors and operators, e.g. with a smartphone acting as BLE client and gateway to cloud-based applications.

#### IV. PERFORMANCE MONITORING METHOD

The online measurement method presented in Sections VI and VII is implemented as a GATT service called PerfMon. A service provider can include this method to monitor the performance of those GATT-based data profiles it finds relevant. The architecture and its wider context are described in Section VI and Fig. 4. The measurement method is explained in this section.

The performance monitoring function consists of lightweight traffic meters and monitoring packets. A traffic meter is implemented in both sending and receiving BLE devices, the smartphone and each of the sensor nodes. The traffic meters have two counters that keep track of the number of sent and received packets and bytes. The monitoring packets are implemented as a GATT service. These dedicated monitor packets are sent, using notifications, with a configurable time interval. The interval between the monitoring packets is the size of the monitoring block (Fig. 3). The monitoring packets

contain a sequence number, a timestamp and the current counter value of the cumulative number of packets and bytes transmitted by the sending node. Upon receiving a monitoring packet, the receiving node’s traffic meter stores a timestamp and the current value of the cumulative number of received packets and bytes. Observe that for  $n$  transmitting sensor nodes, the receiving smartphone maintains  $n$  separate monitoring functions, one for each transmitting sensor node.

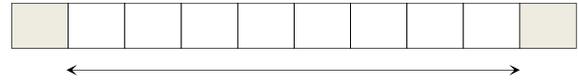


Fig. 3. A monitoring block consists of data packets sent from a GATT server to a GATT client using a GATT-based profile. The monitoring packets (shaded) are sent with a configurable time interval using the PerfMon GATT service (described in Section VI).

Synchronization of the clocks in the participating nodes is not required. The local timestamps are used to calculate the inter-sending and inter-arrival times between pairs of monitoring packets. The inter-arrival jitter can then be calculated in a similar way as for RTP timestamps [25]. This means that the arrival time variation is estimated based on the monitoring packets, which represent samples of the ordinary data packet inter-arrival variation. Packet loss is measured directly using the traffic meter counters. The following metrics can be calculated and estimated based on the measurements described above.

- Packet loss ratio: long-term average and average per monitoring block.
- The length of loss and loss-free periods defined as the number of consecutive monitoring blocks with or without losses.
- Inter-arrival jitter,  $J$ , is defined as  $J=(r_n-r_{n-1})-(s_n-s_{n-1})$ , where  $s$  is the sending time,  $r$  is the receiving time for timestamp sequence numbers  $n$  and  $n-1$ . The monitoring packets provide samples of delay variations.
- Data throughput between sender and receiver can be calculated as a long-term average and also per monitoring block.

The method, described in more detail in [4], enables real-time performance monitoring of sensor data communication. This paper focuses on monitoring and control of packet loss.

#### V. AAL USE CASES

There are many AAL and health-related services and use cases where performance monitoring and control is important, e.g. emergency alert systems, medical reminders, tracking daily activity, GPS (geo-fencing), periodic health check-ups at home etc. The use of BLE sensors for daily health tests at home without visiting a caregiver is described in [26] and [19]. These check-ups can include measurements of blood pressure, heart rate and activity, glucose, weight, medicine consumption from digital dossiers, and continuous data from IMUs (inertial measurement units) etc. Analysis of continuous accelerometer data can be used to detect significant changes of patterns in waking, eating less than usual, abnormal repetition of activities, and restless sleeping [25]. The data rate for such check-ups will vary depending on the use case. A long-term

ECG normally has a 10-bit resolution and sampling rates between 75 Hz and 200 Hz. A modern IMU has 10 degrees of freedom: a three-axis accelerometer, a gyroscope, a magnetometer and an altitude meter. Two bytes are needed for each degree of freedom of the IMU with 10-bit samples, which means that each sample requires a 20-byte payload. The sampling rate can vary between 5 Hz and 100 Hz. In summary, a 10-byte payload sent every 20 millisecond is in most daily check-up's cases sufficient to provide the information requested by a healthcare provider. Fig. 4 shows a case with a number of external sensors connected to a cloud service through a smartphone.

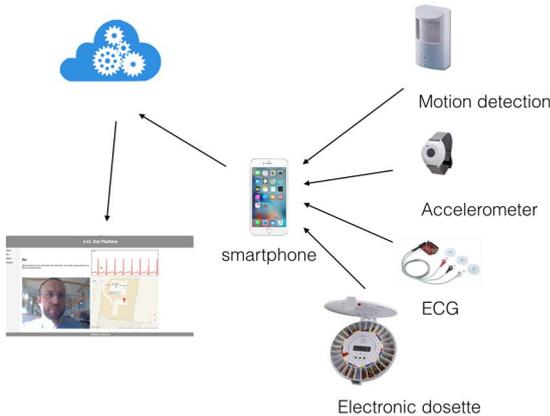


Fig. 4. A smartphone acts as a gateway for an IMU, an ECG, a motion detection device and an electronic dosette. The smartphone connects the devices to a cloud service that can be displayed in a web application (lower left part) and monitored by a healthcare provider.

## VI. BLUETOOTH LOW ENERGY PERFORMANCE MONITORING AND CONTROL

We have developed a GATT-based performance monitoring service (PerfMon) to be installed on GATT servers (BLE sensor devices) and included the GATT-based profiles that a BLE sensor device exposes. The architecture is shown in Fig. 5 below.

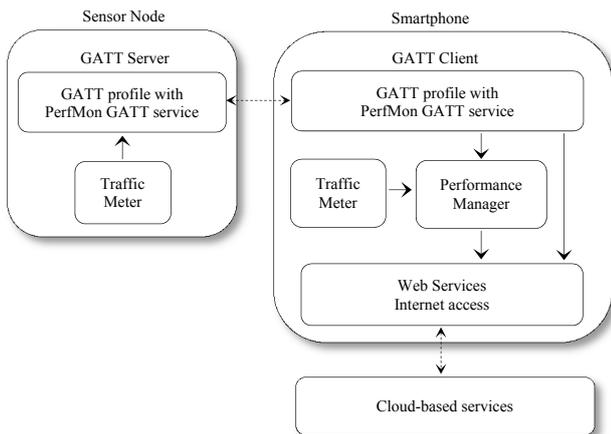


Fig. 5. The architecture of the BLE performance monitoring functions for traffic and services between a sensor node (GATT server) and a smartphone gateway (GATT client) connected to cloud-based services.

A GATT client (a smartphone gateway in our use case) can request performance metrics (Section IV) related to the data traffic and service provided by the application and profiles. This PerfMon GATT service requires a TrafficMeter, a small piece of code, implemented on the sending BLE sensor device and correspondingly on the receiving smartphone gateway. A Performance manager, implemented on the smartphone gateway, processes the information from TrafficMeter on the BLE sensor device provided by the PerfMon GATT service and the TrafficMeter on the receiving smartphone gateway. The resulting performance metrics can be presented on the smartphone or retrieved and stored by a remote client using Web services.

### A. GATT Performance Monitoring Service

An example of configuration of the PerfMon GATT service is shown in Table 1. It has been designed to transfer performance data, retrieved from the TrafficMeter on the sensor node, to the GATT client (Fig. 5). In addition to traffic data, information about signal strength (RSSI), battery capacity and local time are also included. The first three rows in Table 1 describe the declaration and data for this Logger data characteristic. The third row enables and disables notifications sent from the GATT server without being requested. This feature, in combination with PerfMon period (last row in Table 1) can implement a Heartbeat signal from the sensor node to the smartphone gateway. It allows a client to be informed whether a sensor is alive or not even though the data traffic service is disabled. The second characteristic, the PerfMon configuration, declared on row four and five, means that the service can be adjusted to the capacity of the sensor node. It is possible to configure PerfMon GATT service to enable a subset of the data in the first Logger characteristic.

TABLE I. THE PERFMON GATT SERVICE CHARACTERISTICS DESCRIPTION

GATT characteristic declaration	Read	PerfMon data
Logger data	Read, Notify	ByteLSB:ByteMSB:SeqLSB:SeqMSB:TimeLSB:TimeMSB, RSSI, battery
Client characteristic configuration	Read, Write	Write 1 to enable notifications, Write 0 to disable notifications
GATT characteristic declaration	Read	PerfMon configuration
PerfMon configuration	Read, Write	PerfMon enable bits:Byte=0,Seq=1,Time=2, RSSI = 3, Battery = 4
GATT characteristic declaration	Read	PerfMon Logger period
PerfMon period	Read, Write	Period = [Input*100] ms (lower limit 1000 ms), default 10000 ms

The third characteristic, PerfMon period, declared on the last two rows in Table 1, allows the GATT client to set the time interval between the monitoring packets (the PerfMon GATT service data packets) adapted to the requirements for the sensor data traffic and service that is monitored. The tuning of this parameter, the size of the monitoring block, is a

trade-off between, on the one hand the resolution and accuracy of performance metrics, and on the other hand, the extra load on the sensor nodes and network. If a PerfMon packet is inserted on average between a block of e.g. one thousand sensor data packets, the increase in traffic and battery consumption will be around 0.1 %. This is further analysed in [4].

### B. The Performance Manager

The Performance manager has several tasks: to calculate, compile and send performance reports and alarms; set policies and levels for performance thresholds and alarms; and, initiate control actions based on policies and thresholds. The Performance manager (Fig. 5) receives performance data in the PerfMon monitoring packets sent from the sensor node and from the TrafficMeter in the gateway (e.g. a smartphone) as well. The performance metrics, described in Section IV, can then be calculated. Reports and alarms are presented on the smartphone and can be retrieved via Web services as well (Section VI.C). The Performance manager may also be located on an external server that retrieves information from the PerfMon packets and the smartphone's TrafficMeter via Web services. Section VII.B describes an example of a control action where packet loss ratio above a certain threshold is followed by an instruction to reduce the packet frequency for the sensor data traffic being monitored.

### C. Web-Based Management for Caregivers and Service Providers

Caregivers and service providers may have a number of caretakers in assisted living homes using various wireless sensors. A service management system, e.g. using the LWM2M protocol (Section II), should be able to remotely configure devices, receive performance reports and alarms, and control the equipment and services as well. The PerfMon GATT service can be translated to a RESTful API that Web services can access. The API converts a GATT characteristic declaration according to the data model from IPSO (Section II and III). The IPSO smart objects and resources [13] are applied in the following way; the Generic sensor object (ID=3300) represents a sensor device; the Set point object (ID=3308) is used for configuration and the Analogue Input object (ID=3202) for setting the interval between PerfMon monitoring packets; and, the Resource IDs 5700-5703 represent the number of packets, bytes and local time. The CoAP request to retrieve performance data from the GATT server will then be: `https://IP_addr/1/3300/5700`, where IP\_addr is the IPv4 or IPv6 address to the smartphone. This can also be converted to a MQTT or any other preferred RESTful request.

## VII. RESULTS AND DISCUSSION

In this section results using the prototype implementation of the PerfMon GATT service are presented: firstly, monitoring of the packet loss ratio of a GATT-based data service; and secondly, control actions to decrease the packet loss ratio based on the feedback information. The PerfMon GATT service is implemented on an iPhone that acts as a BLE peripheral sensor device. A generic GATT-based data service is implemented and configured to send 10-byte data packets with a variable interval.

### A. Performance Monitoring Reports

The Performance manager receives data from the PerfMon GATT server on the sensor device and the TrafficMeter on the smartphone and presents statistics of the measured packet loss ratio in this case. In Fig. 6, a generic GATT data service sends packets with 100 milliseconds intervals during three minutes. The alarm threshold for the loss ratio in this case is set to 5 %. The Performance manager reports that packet loss ratio during time periods P<sub>1</sub>, P<sub>2</sub> and P<sub>3</sub> exceeds the threshold.

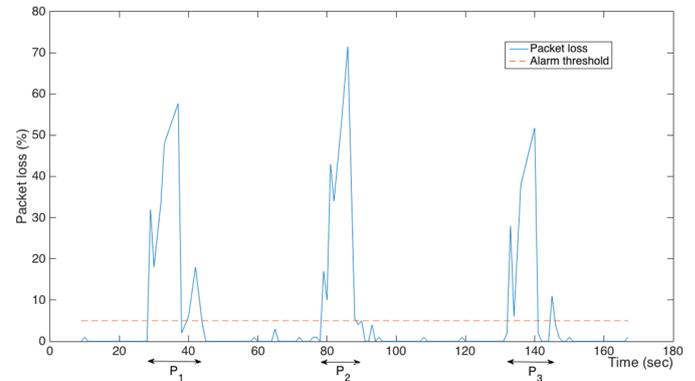


Fig. 6. The figure shows 10-byte packets transmitted with 100 millisecond intervals during three minutes. Data sent during time periods P<sub>1</sub>, P<sub>2</sub> and P<sub>3</sub> have a packet loss ratio that exceeds the alarm threshold (5 %).

### B. Performance Control

The Performance manager may implement control actions based on feedback information from the TrafficMeters and the PerfMon GATT service. The control policy can comprise priority between multiple sensors, inactivation or activation of sensors as well as algorithms for feedback control. A simple case is illustrated in Fig. 7 where a generic GATT-based data profile is monitored by the PerfMon GATT service.

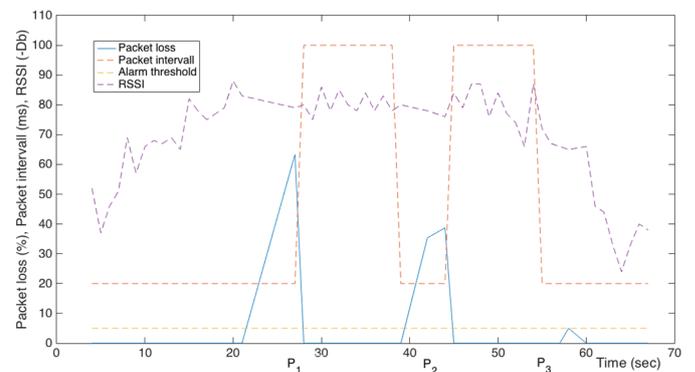


Fig. 7. The figure shows packet loss ratio and packet intervals for sensor data transmission during one minute. The Performance manager instructs the sensor node to increase the packet interval when the loss ratio exceeds the threshold (5 %). The dotted violet line shows the received signal strength.

The initial packet interval for the generic GATT service is set to 20 milliseconds. However, as the packet loss ratio exceeds the alarm threshold (5 %) at P<sub>1</sub>, the sensor node is instructed to increase the packet interval to 100 milliseconds during 10 seconds, which results in packet loss below the

threshold. At  $P_2$ , the Performance manager tries to decrease the packet interval, but resumes to a 100 millisecond packet interval for another 10 seconds. At  $P_3$ , the Performance manager successfully reduces the packet interval to 20 milliseconds. The relation between packet loss and the wireless channel's received signal strength (RSSI) in Fig. 7 is discussed below.

Packet loss ratio for increased packet frequency is shown in Fig. 8. Packets that are lost or damaged in poor wireless channels are retransmitted. Even though BLE applies retransmissions, packets attempted to be sent by applications may not reach the receiver. They are lost due to overflow in the transmission buffer, mainly for two reasons. Firstly, the application can deliver packets to the transmission buffer with a rate that exceeds the BLE radio capacity (Fig. 6). Secondly, when an ACK is missing, or a negative ACK arrives, the previous packet is resent and no new packets are transmitted. Packets that an application still submits to the buffer, before next ACK or a timeout, lead to overflow and are eventually discarded. If the packet interval is sufficiently long, or if the application adapts to the status of the link, buffer overflow can be avoided. This can be seen in Fig. 7 where the received signal strength (RSSI) is displayed. An impaired channel and lower RSSI will not lead to packet loss provided that the packet frequency is sufficiently low.

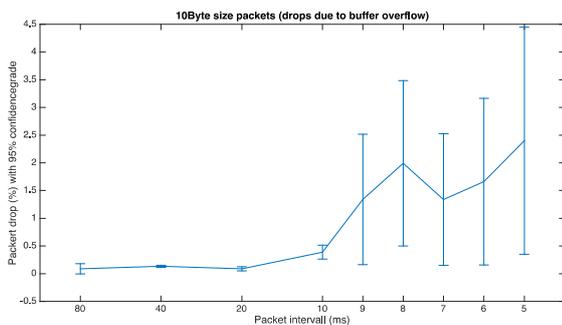


Fig. 8. Packet loss ratio for a sensor transmitting 10-byte packets with intervals decreasing from 80 milliseconds to 5 milliseconds. Confidence intervals for the 95 % confidence grade is indicated in the figure.

### C. Future Work

The PerfMon service is one component in an overall management system of IoT devices and services for providers of healthcare and assisted living environments. Research issues related to distributed systems, scalability and security are part of future work on management of IoT. Algorithms for feedback control of communication performance and sensor battery usage will be further developed and evaluated in this context.

## VIII. CONCLUSIONS

This paper has presented the design and a prototype implementation of a performance monitoring and control method for assisted living services in BLE sensor network communication. The method provides real-time performance monitoring and control capabilities for caregivers and service providers. These functions have been implemented as a PerfMon GATT service in accordance to the BLE specifications. Re-

sults from lab test scenarios are presented, firstly, performance monitoring of packet loss ratio and alarm thresholds, and secondly, control actions to decrease the packet loss ratio. The PerfMon GATT service and Performance manager are adapted to cloud-based Web services and more general management architectures for caregivers and providers of healthcare and assisted living services.

## ACKNOWLEDGEMENTS

We thank Marcus Anderson for his valuable contributions in the degree project report “Implementation of a Performance Monitoring Method of Sensor Communication with Bluetooth Low Energy”, KTH, June 2016 (TRITA-STH 2016:112).

## REFERENCES

- [1] European AAL Joint Programme, <http://www.aal-europe.eu/>
- [2] EIT Health, <https://eithealth.eu/>
- [3] J. Wåhslén and T. Lindh, “Smartphone-centric Wi-Fi device-to-device sensor communication for user mobility in AAL services”, IEEE 27th International Symposium on Personal, Indoor and Mobile Radio Communications, Workshop IoTAAL (Internet of things for ambient assisted living, September 2016, Valencia. Spain.
- [4] I. Orhan and T. Lindh, “Measurement-based performance and admission control in wireless sensor networks”, International Journal on Advances in Systems and Measurements. International Journal on Advances in Systems and Measurements, Vol. 4, No. 1, 2011.
- [5] D. Camps-Mur et al., “Device-to-device communications with Wi-Fi direct: overview and experimentation”, IEEE Wireless Communications, June 2013.
- [6] Release-13, 3GPP, 2016.
- [7] The LoRa Alliance, <http://www.lora-alliance.org>
- [8] SigFox, <https://www.sigfox.com>
- [9] OMA Lightweight M2M v1.0 LWM2M OMA, 2014.
- [10] The Constrained application protocol (CoAP), RFC 7252, June 2014.
- [11] Constrained RESTful Environments, <https://datatracker.ietf.org/wg/core/charter>
- [12] Transmission of IPv6 packets over IEEE 802.15.4 networks, RFC 4944, September 2007.
- [13] IPSO smart object guideline, Starter pack 1.0, IPSO, September 2014.
- [14] MQTT version 3.1.1, OASIS standard and ISO/IEC 20922:2016 standard.
- [15] M Memon et al., “Ambient Assisted Living Healthcare Frameworks, Platforms, Standards, and Quality Attributes”, Sensors, March 2014, Vol.14(3), pp.4312-4341.
- [16] Bluetooth low energy specifications for version 4.0, 4.1, 4.2, Bluetooth SIG, 2014.
- [17] Heart rate profile, Bluetooth SIG, July 2011.
- [18] <https://www.bluetooth.com/specifications/profiles-overview/creating-custom-profiles>
- [19] Internet gateways, White paper, Bluetooth SIG, February 2016.
- [20] GATT REST API, White paper, Bluetooth SIG, April 2014.
- [21] GAP REST API, White paper, Bluetooth SIG, April 2014.
- [22] BIPSO library for using IPSO smart objects on BLE devices, <http://bluetoothether.github.io/bipso/#/> (2016)
- [23] Internet protocol support profile, Specification, Bluetooth SIG, 2014.
- [24] IPv6 over Bluetooth low energy, RFC 7678, 2015.
- [25] RTP: A transport protocol for real-time applications, RFC 3550, July 2003.
- [26] CarePredic, <https://www.carepredic.com>