

Cross-Layer Admission Control to Enhance the Support of Real-Time Applications in WSN

Pedro Pinto, António Pinto, and Manuel Ricardo, *Member, IEEE*

Abstract—Real-time monitoring applications may be used in a wireless sensor network (WSN) and may generate packet flows with strict quality of service requirements in terms of delay, jitter, or packet loss. When strict delays are imposed from source to destination, the packets must be delivered at the destination within an end-to-end delay (EED) hard limit in order to be considered useful. Since the WSN nodes are scarce both in processing and energy resources, it is desirable that they only transport useful data, as this contributes to enhance the overall network performance and to improve energy efficiency. In this paper, we propose a novel cross-layer admission control (CLAC) mechanism to enhance the network performance and increase energy efficiency of a WSN, by avoiding the transmission of potentially useless packets. The CLAC mechanism uses an estimation technique to preview packets EED, and decides to forward a packet only if it is expected to meet the EED deadline defined by the application, dropping it otherwise. The results obtained show that CLAC enhances the network performance by increasing the useful packet delivery ratio in high network loads and improves the energy efficiency in every network load.

Index Terms—WSN, admission control, end-to-end delay, estimation.

I. INTRODUCTION

MULTIPLE types of applications are expected to be supported by Wireless Sensor Networks (WSNs). This work addresses the topology scenario depicted by SELF-PVP project [1] that aims to increase the efficiency of a photovoltaic power plant where solar panels communicate with each other using a WSN in a grid topology and considers that a real-time multimedia application should also be supported by the

network in order to monitor the power plant infrastructure. These applications expect from the network quality characteristics regarding delay, packet loss or bitrate. In particular, it is assumed that the multimedia application requires a maximum End-to-End Delay (EED) for its data packets. A packet will be considered useful if delivered at the destination within the expected maximum EED and useless otherwise.

The nodes in a WSN are known to have limited processing and communications capabilities so we assume that the elimination of useless packets - those that will be of no use at the receiver - contributes to enhance network performance and the end user quality of experience. At same time, energy resources can be used more efficiently, transmitting only useful data.

A node consumes energy while processing packets and using the radio for packet transmission or reception, but a packet's usefulness is determined at the destination, when these processing and transmission resources may already have been expended to transport it. Since the destination application may not consider all received packets as useful, if we are able to identify, as soon as possible, which packets will likely miss the application delay deadlines and avoid their transmission to the network, an increase in network performance and energy efficiency is expected. In order to implement such an approach, two mechanisms are required: one mechanism to perform the EED estimation and another to perform packet admission control. The EED estimation mechanism will estimate an EED per packet, and the admission control mechanism will decide if a packet should progress to its destination based on that estimation.

The main contribution of this paper is a novel Cross-Layer Admission Control (CLAC) mechanism intended to enhance the support for real-time applications in a WSN. CLAC is a distributed mechanism deployed in each WSN node which intercepts packets when they are generated (in the application layer) and when they are forwarded (in the network layer) in a cross-layer operation. CLAC is also responsible for a decision to send or drop an intercepted packet according to the requirements defined by the application and the input of an EED estimation mechanism proposed in a previous work [2]. This EED estimation mechanism provides a per packet delay estimation while using Routing Protocol for Low-power and Lossy Networks (RPL) [3] messages to feedback the EED delays of previously sent packets to the source nodes and thus, it avoids out-of-band traffic in the WSN, that would contribute to use more energy.

The structure of this paper is as follows. Section II surveys the work related to our proposal. Section III describes the

Manuscript received May 22, 2015; revised July 25, 2015; accepted July 28, 2015. Date of publication August 12, 2015; date of current version October 8, 2015. The work was supported in part by the BEST CASE Project under Project NORTE-07-0124-FEDER-000056 through the North Portugal Regional Operational Program ON.2 - O Novo Norte, under the National Strategic Reference Framework within the European Regional Development Fund, and in part by the National Funds through the Fundação para a Ciência e a Tecnologia. The associate editor coordinating the review of this paper and approving it for publication was Prof. Elena Gaura.

P. Pinto is with the Instituto Politécnico de Viana do Castelo, 4900-347 Viana do Castelo, Portugal, and also with the Instituto de Engenharia de Sistemas and Computadores, Technology and Science, 4200-465 Porto, Portugal (e-mail: pedropinto@estg.ipv.vc.pt).

A. Pinto is with GCC, Centro de Inovação e Investigação em Ciências Empresariais e Sistemas de Informação, Escola Superior de Tecnologia e Gestão de Felgueiras, Politécnico do Porto, 4200-465 Porto, Portugal, and also with the Instituto de Engenharia de Sistemas and Computadores, Technology and Science, 4200-465 Porto, Portugal (e-mail: apinto@inesctec.pt).

M. Ricardo is with the Instituto de Engenharia de Sistemas and Computadores, Technology and Science, 4200-465 Porto, Portugal and also with the Faculdade de Engenharia, Universidade do Porto, 4200-465 Porto, Portugal (e-mail: mricardo@inesctec.pt).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/JSEN.2015.2467329

operation and configuration of the EED estimation mechanism. Section IV describes the cross-layer admission control mechanism. Section V describes the hardware and the simulation environment used to validate the admission control proposal. Section VI presents and discusses the results obtained. Section VII concludes our work and addresses future work.

II. RELATED WORK

Current research efforts on EED estimation focus on probabilistic estimation or on the use of routing protocols. The use of out-of-band packets as a way to feedback experienced delays imposes extra traffic on the WSN, and demands more processing, transmission and energy resources. Our EED estimation mechanism, proposed in [2], provides a per packet EED estimation using RPL packets to feedback the EED to the source nodes. In order to enhance the EED estimation accuracy of the mechanism proposed in [2], a set of RPL modifications were proposed in [4]. In [5] the estimation mechanism was adapted in order to provide the best estimation results for multiple network loads.

The Admission Control (AC) mechanisms can be categorized as centralized or distributed [6]. The centralized AC assumes that only one entity performs the AC decisions and that such entity exchanges signaling packets with ingress nodes when new flows arrive. The distributed AC assumes that the decision may be performed in multiple points within the network. The distributed AC mechanisms avoid the single point of failure and the scalability concerns of the centralized approach.

The distributed AC mechanisms can be organized in two groups: Edge-to-Edge and Hop-by-Hop. In the Edge-to-Edge mechanisms, only the ingress and egress nodes participate in the AC mechanism; the AC decision is taken on the egress node based on measurements and the decision is transported back to the ingress node. In the Hop-by-Hop mechanism, all nodes participate; each one of the routers in AC deployment area take a local decision about new flows. Considering the Hop-by-Hop mechanisms, three types of proposals can be found in literature: the Parameter Based Admission Control (PBAC), the Measurement Based Admission Control (MBAC), or a combination of both.

PBAC decisions are based on the traffic characteristics of the new flows, which must be known prior to their establishment. Each node must have a complete knowledge of the currently admitted requests and of the currently available resources. The major disadvantage of these mechanisms is that it is difficult to have an accurate knowledge of each flow before its establishment. Mushtaq and Ahmed [7] present a PBAC mechanism implemented in a peer-to-peer network for real-time video streaming applications; the decision of the AC mechanism is performed by a service provider, and is based on traffic descriptors that characterize the applications and their contract with the service provider.

MBAC decisions are based on real-time network measurements. The AC mechanism attempts to capture the characteristics and requirements of the admitted flows and bases its decisions on this knowledge. The major disadvantage of MBAC is that its decision depends on measurements that

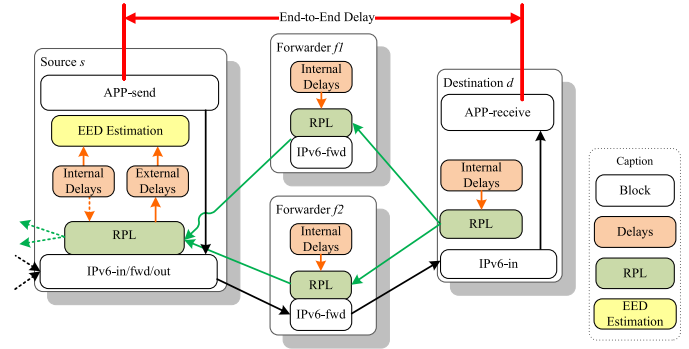


Fig. 1. Overview of the EED estimation mechanism overview.

have errors and may lead to false negatives or false positives. A MBAC mechanism that only uses the aggregated bandwidth measurement and does not need to keep flow state information in each node is proposed in [8]. A MBAC mechanism for WSN based on direct measures of packet loss ratio, inter-arrival jitter and throughput is proposed in [9] to be used by real-time applications; the authors estimate these performance parameters by using probing packets. Brewer and Ayyagari [10] implemented two AC mechanisms, one using PBAC and the other using MBAC, and evaluated the efficiency both. When tested with bursty traffic, the authors concluded that MBAC provided a more efficient network utilization. In [11] is proposed a MBAC mechanism implemented over a reactive routing protocol, that computes the required bandwidth and estimates the available bandwidth, and based on these inputs decides to admit/deny a flow in a per-hop basis. Jiao *et al.* [12] proposed a analytical model for node delay distribution in IEEE 802.11 wireless networks and developed an admission control mechanism scheme for traffic with stochastic quality of service guarantees.

The hybrid proposals use both knowledge of existing traffic and measurements taken from the network to predict future service levels required by a flow. Davy *et al.* [13] present a hybrid proposal that directly estimates bandwidth usage from the available traces and uses these estimates, combined with peak rate values, decide on a new flow. A hybrid admission control mechanism for real-time traffic was proposed in [14] and takes delay and reliability into account, plus a fairness-aware rate control algorithm for non-real-time traffic, to be used in WSN. The admission control mechanism is deployed in the source node and the delay estimation is not addressed.

III. END-TO-END DELAY ESTIMATION MECHANISM OPERATION AND CONFIGURATION

The CLAC mechanism proposed in Section IV depends on the estimation of EED, what lead the authors to develop the EED estimation mechanism proposed in [2]. An overview of the EED estimation mechanism is shown in Fig. 1. For simplicity, the figure shows only functions above network layer. The EED estimation mechanism outputs a per-packet EED estimate in each node based on the EED experienced by previous data packets sent along the path from the source node to the destination node. The EED estimation is performed

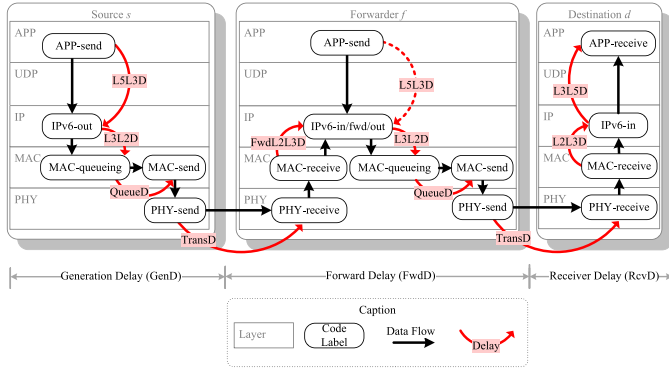


Fig. 2. Interaction of WSN nodes in EED estimation.

using two components: the Internal Delays (delays accounted inside the node) and the External Delays (other nodes' delays transported by RPL in the opposite direction of the data flow).

Internal delays consist of time intervals when the packet is processed within the stack of the source node, the packet is waiting in the MAC layer queue, or the packet is being transmitted. Delay accounting is accomplished by using timers that measure delays between labels inserted into code where the data flow passes through, ranging from the source application node to the application in the destination node. This EED estimation mechanism assumes that the WSN nodes run Contiki [15] and thus, the labels above were inserted in the Contiki code files, according to the proposal in [2]. These internal paths and associated delays are shown in Fig. 2. The $LxLyD$ format represents the delays between layer x and layer y , the MAC queuing delay (QueueD) is the time interval a packet spent in MAC queue, and the Transmission Delay (TransD), is the time interval required for the successful transmission of a packet.

According to Fig. 2, from the packet generation up to the moment the packet reaches the destination, three types of Internal Delays are considered: the Generation Delay (GenD), the Forward Delay (FwdD), and the Receiver Delay (RcvD). From a source s with a parent f and a destination d , GenD^{sf} , FwdD^{fd} , and RcvD^d are obtained as follows:

$$\text{GenD}^{sf} = \text{L5L3D}^s + \text{L3L2D}^s + \text{QueueD}^s + \text{TransD}^{sf} \quad (1)$$

$$\text{FwdD}^{fd} = \text{FwdL2L3D}^f + \text{L3L2D}^f + \text{QueueD}^f + \text{TransD}^{fd} \quad (2)$$

$$\text{RcvD}^d = \text{L2L3D}^d + \text{L3L5D}^d \quad (3)$$

All the delays accounted in the Internal Delays are obtained by using Exponential Weighted Moving Averages (EWMA). Whenever a new delay item is accounted (Delay_i), it is obtained a delay estimation for a future packet p (Delay_p) using all delay history items as follows:

$$\text{Delay}_p = \beta \cdot \text{Delay}_i + (1 - \beta) \cdot \text{Delay}_{p-1} \quad (4)$$

The EED estimation mechanism should provide an accurate estimation for multiple network loads and thus, our proposal in [5] is used to obtain the best β value according to the network load.

External Delays are the delays transported back to the source nodes by using RPL messages with a set of RPL metrics. An RPL instance is organized in a tree topology named Directed Acyclic Graph (DAG), where the DAG root is the node where all paths terminate. Within the DAG, the Objective Function (OF) defines how to the metrics/constraints above are converted into a rank value, i.e. a value representing the distance/cost to the DAG root. The OF also defines how a node selects its parent. In the current proposal, is used the Minimum Rank with Hysteresis Objective Function (MRHOF) [16] with a variable Hysteresis Value (HystV) according to our proposal in [4], in order to avoid parent selection instability. The lower the preferred parent delay metric is, the higher the HystV will be, making the parent change less probable.

To create and maintain routing information in each node, the RPL standard defines a set of control messages. The DAG Information Solicitation (DIS) message is used when a node wants to join the network; the DAG Information Object (DIO) message is used to carry information of the RPL instance and DAG configuration parameters; the DAO message is used to support downward routes in the DAG. DIS and DAO are sent in the same direction of the data flow so they are not suitable to transport the delay feedback required for the EED estimation mechanism. DIO conveys the metrics/constraints using the DAG Metric Container option and it is the single RPL control message that can be used to convey the EED feedback in a direction reverse to the data flow. The current scenario uses two RPL metrics: the Delay Metric (DMetric) which is used to obtain the rank, representing the cumulative delays up to the DAG root, and the Hop count Metric (HopMetric) which counts the hops up to the DAG root in order to obtain HystV. Both metrics used are additive and set to provide a routing decision in order to return the lowest delay route from a source node towards the DAG root. According to Fig. 2, a node s with an RPL preferred parent f and destination d , will advertise the following DMetric and HopMetric:

$$\text{DMetric}^{sd} = \text{FwdD}_p^{sf} + \text{DMetric}^{fd} \quad (5)$$

$$\text{HopMetric}^{sd} = \text{HopMetric}^{fd} + 1 \quad (6)$$

The destination node d advertises, to its neighbors, the following metrics:

$$\text{DMetric}^{dd} = \text{RcvD}_p^{dd} \quad (7)$$

$$\text{HopMetric}^{dd} = 0 \quad (8)$$

The EED estimation is obtained by using the Internal and External Delays. When a node s has to send a data packet p , the EED estimation mechanism provides an estimated EED (EstEED) towards destination d obtained as follows:

$$\text{EstEED}_p^{sd} = \text{GenD}_p^{sf} + \text{DMetric}^{fd} \quad (9)$$

The procedure to obtain the EED estimation is implemented in all nodes of the WSN, except at the DAG root.

IV. CROSS-LAYER ADMISSION CONTROL MECHANISM

CLAC is a distributed mechanism running in each WSN node that intercepts packets; it requests EED estimations

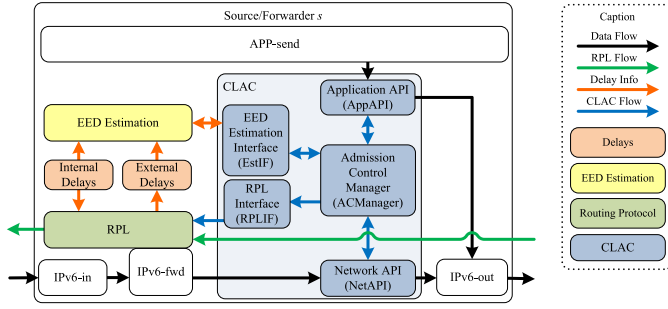


Fig. 3. CLAC internal overview.

and decides if the node should accept or reject the packets, according to the preview of their usefulness at the application layer of the destination node. CLAC is designed to enhance network performance while fostering energy efficiency in a grid WSN. Since not all the received packets are useful for the destination, in alternative to the common Packet Reception Ratio (PRR), we also define Packet Usefulness Ratio (PUR) as the Number of Useful Packets (#UsefulPkts) per Number of Received Packets (#RcvdPkts) as follows:

$$\text{Packet Usefulness Ratio (PUR)} = \frac{\#UsefulPkts}{\#RcvdPkts} \quad (10)$$

The #UsefulPkts can be accounted as follows:

$$\#UsefulPkts = \sum_{p=1}^{\#RcvdPkts} U(p) \quad (11)$$

where $U(p)$ is 1 if packet p is useful, or 0 otherwise, expressed as follows:

$$U(p) = \begin{cases} 1 \rightarrow EED_p \leq \text{MaxEED} \\ 0 \rightarrow EED_p > \text{MaxEED} \end{cases} \quad (12)$$

In order to preview $U(p)$, CLAC uses the Usefulness Preview (UP) function for packet p expressed as follows:

$$UP(p) = \begin{cases} \text{progress} \rightarrow \text{EstEED}_p \leq \text{MaxEED} \\ \text{drop} \rightarrow \text{EstEED}_p > \text{MaxEED} \end{cases} \quad (13)$$

where EstEED_p is the estimated EED for each packet p , i.e. the result of Eq. 9. A packet having EstEED below MaxEED is an in-profile packet and should progress, and above MaxEED it is an out-of-profile packet and should be dropped.

Fig. 3 shows the integration of the CLAC mechanism with the EED estimation mechanism and the internal building blocks of CLAC. CLAC assumes two types of nodes: the source/forwarder node that generates packets or forwards packets from other nodes, and the destination node which consumes the packets. CLAC is deployed only in the source/forwarder nodes with the support of the EED estimation mechanism that, in turn, depends on delay information conveyed by RPL control messages.

In order for CLAC mechanism to intercept and evaluate the usefulness of each packet, as earlier as possible, it defines two stages for packet interception: after the packet generation at the application layer, and when a packet is

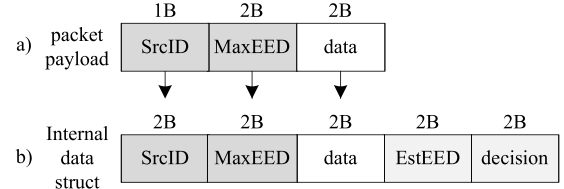


Fig. 4. Data packet payload internal mapping performed by CLAC. a) Packet payload. b) Internal data struct.

being forwarded at the network layer. Two Application Programming Interfaces (APIs) were defined for this purpose: the Application API (AppAPI) and the Network API (NetAPI). Additionally, two interfaces were also defined: the EED Estimation Interface (EstIF), and the RPL Interface (RPLIF). EstIF handles the delay estimation requests issued to the EED estimation mechanism. The RPLIF is used to trigger the sending of RPL control messages when necessary. The core block, named Admission Control Manager (ACManager), receives requests from AppAPI and NetAPI and, according to the caller API, issues requests for delay estimations to the EstIF. Then, the ACManager provides a decision (accept or drop) regarding the progress of a packet and communicates this decision back to the caller API (AppAPI or NetAPI). If the caller API is the NetAPI, the ACManager can also request the sending of RPL control messages to the RPLIF.

In order to support the operation of CLAC, data packets must have a payload format that contains a set of fields that can either be generated by the application or added later by a middle layer. In the adopted scenario, it is assumed that the data packets are generated with the fields shown in Fig. 4a. The data packet payload includes a Source ID (SrcID) which enables the per packet EED registering, and a MaxEED which indicates the maximum amount of delay allowed by the application. The MaxEED is defined when the packet is generated and updated while in progress to its destination. In each data packet interception performed at the application layer or at the network layer, the data packet payload (see Fig. 4a) is mapped to an internal data structure representing the packet according to Fig. 4b. For each instance of such data structure, the ACManager requests an EstEED to the EstIF. The returned value will be evaluated by the ACManager that will provide the decision whether the packet is to be accepted (value 1) or to be dropped (value 0). The EstEED and the decision fields added to the packet payload are saved in the internal data structure and, according to the decision field returned by the ACManager, the APIs will map the internal data structure into the packet payload again and allow the data packet to proceed, or not, according to the decision value.

A flow diagram of the detailed interaction between the application layer and the CLAC mechanism is shown in Fig. 5. The application layer requests a decision to the AppAPI where the packet is mapped into the internal data structure of CLAC. The ACManager requests an EstEED to the EstIF and stores it in the EstEED value. The ACManager checks if EstEED is less than the value defined for MaxEED. If EstEED does not meet this condition, the ACManager writes a zero in the decision field and the packet is later dropped by

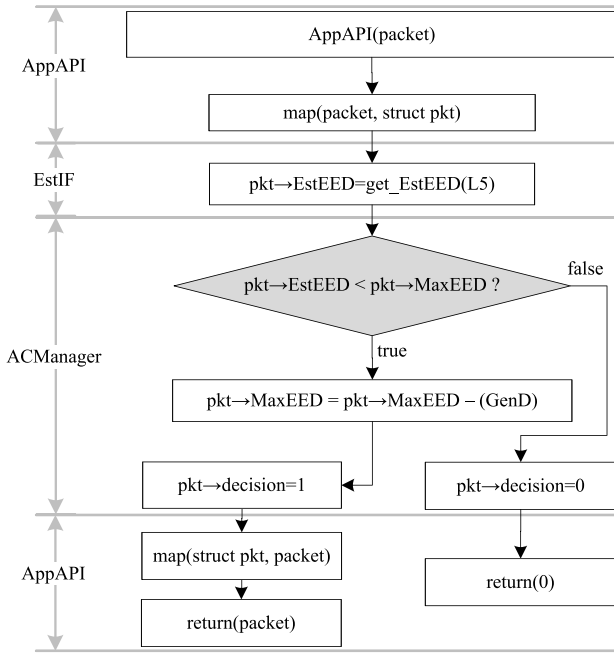


Fig. 5. CLAC interaction with the application layer (using AppAPI).

the AppAPI. If the EstEED is below the defined MaxEED, the MaxEED is updated by subtracting the GenD of the current node, and decision of one is written in the struct. The AppAPI will then return the packet towards its destination.

Fig. 6 presents a flow diagram of the interactions within the CLAC mechanism when a packet is forwarded at the network layer. The NetAPI maps the packet into an internal data structure and then the ACManager checks if the packet has been generated in the current node or if the destination address is a multicast address. If true, the packet decision is set to one and later, NetAPI will map the data structure back into the packet to be forwarded. If not, the ACManager updates the MaxEED by subtracting the FwdL2L3D value. The ACManager then requests an EstEED to EstIF and checks if the returned value is less than the current MaxEED. If true, the MaxEED value is updated again by subtracting the values L3L2D, QueueD and TransD. The ACManager sets packet decision to one and NetAPI will map the data structure to the packet in order to be forwarded towards destination. If false, the ACManager sets the decision value to 0 to order NetAPI to drop the packet. Right after the decision taken by the ACManager, the RPLIF is ordered to send an RPL DIO message to feedback delay information to the previous nodes (forwarders or generators), forcing them to update their delay estimation. Ideally, nodes would not have to discard any packets at the NetAPI, as they should all be discarded at the generation (at the AppAPI).

V. HARDWARE AND SIMULATION ENVIRONMENT

A test scenario was deployed using the open source Cooja Simulator [17] in order to validate our proposal. A grid network topology, as shown in Fig. 7, and the simulation parameters shown in Table I were used. The scenario consists of 16 source/forwarder nodes, placed within a distance of

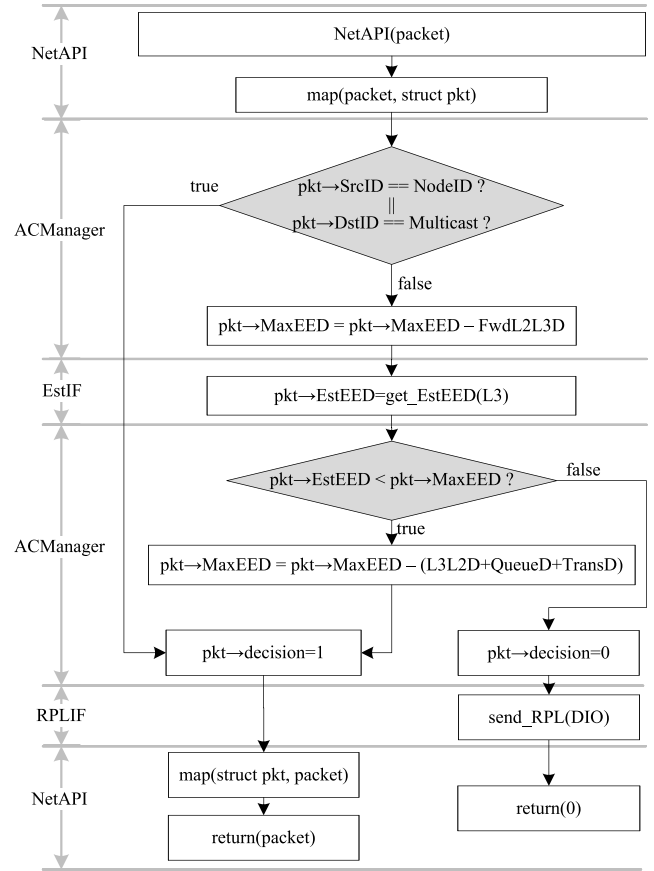


Fig. 6. CLAC interaction with the network layer (using NetAPI).

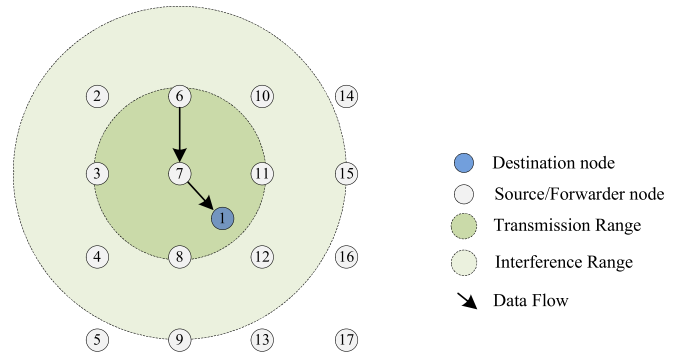


Fig. 7. Simulation Topology.

25 m from each other, plus a destination node, all deployed in a WSN area of 100 m². Each node was simulated as a Tmote Sky [18] using a transmission range of 30 m, an interference range of 60 m, and the Unit Disk Graph Medium as the physical channel model. The nodes ran the Contiki OS 2.5 [15] and extra code was inserted to implement the EED estimation and the CLAC mechanisms. The application layer uses UDP as transport and generates packets of 100 Bytes in a Constant Bit Rate (CBR) by using constant Inter-packet Generation Intervals (IGIs). The simulations were repeated 10 times using different seeds, and each was configured to stop whenever the source had sent 100 packets. CLAC was tested against a scenario using no admission control mechanism (CLAC *off*).

TABLE I
SIMULATION PARAMETERS

Parameter	Value
Number of nodes	16 + sink node
Deployment area	100 m x 100 m
Transmission range	30 m
Interference range	60 m
Channel	Unit Disk Graph Medium
Packet size	100 Bytes
# of sent packets per node	100 packets
Total # of sent packets (#SentPkts)	1600 packets
Transport/Application	UDP/CBR

The accuracy of the EED estimation was assessed and CLAC was evaluated regarding a set of network performance items and energy savings.

In order to assess the EED estimation accuracy, the EstEED for each packet was collected and later compared with the real EED (RealEED) of that packet. When the simulation ended, the estimation accuracy was evaluated using the EED Error (EEDError) for a #RcvdPkts samples obtained using the difference between EstEED and RealEED using the Mean Absolute Percentage Error (MAPE) according to Eq. 14.

$$\text{EEDError} (\%) = \frac{1}{\#RcvdPkts} \sum_{p=1}^{\#RcvdPkts} \frac{|\text{EstEED}_p - \text{RealEED}_p|}{\text{RealEED}_p} \quad (14)$$

CLAC's network performance was evaluated regarding the following items: average EED, PRR, in-profile versus out-of-profile packets, and PUR. The Average EED was obtained using the RealEED for each packet and, at the end of the simulation, using Eq. 15.

$$\text{Average EED (ms)} = \frac{1}{\#RcvdPkts} \sum_{p=1}^{\#RcvdPkts} \text{RealEED}_p \quad (15)$$

$$\text{Average PRR} (\%) = \frac{\#RcvdPkts}{\#SentPkts} \quad (16)$$

It was defined In-profile Packets Ratio (IPR) and Out-of-profile Packets Ratio (OPR), respectively obtained using number of in-profile packets and out-of-profile packets, per #SentPkts (constant in all cases), according to Eq. 17 and Eq. 18.

$$\text{IPR} (\%) = \frac{\text{Number of in-profile packets}}{\#SentPkts} \quad (17)$$

$$\text{OPR} (\%) = \frac{\text{Number of out-of-profile packets}}{\#SentPkts} \quad (18)$$

The PUR was obtained using Eq. 10 where the #UsefulPkts equals to the number of in-profile packets.

CLAC was evaluated also for energy savings. The energy in Joules consumed by a device can be obtained using the power in Watts multiplied by time. Since a Tmote Sky device was used, three different power constants were defined using the values shown in Table II, obtained from the Tmote Sky datasheet [19].

The Powertracker plugin for Cooja was used to collect the time that each node n was in the monitored state ($\text{Time}_{Monitored}^n$), the time it was in the *on* state (Time_{on}^n), and

TABLE II
DEFINED POWER CONSTANTS

Operating Conditions	Voltage (V)	Current Nom. (mA)	Power Constant	
			Name	Value (mW)
MCU on Radio RX	3	21.8	Power_{Rx}	65.4
MCU on Radio TX	3	19.5	Power_{Tx}	58.5
MCU on Radio off	3	0.18	Power_{MCUon}	0.54

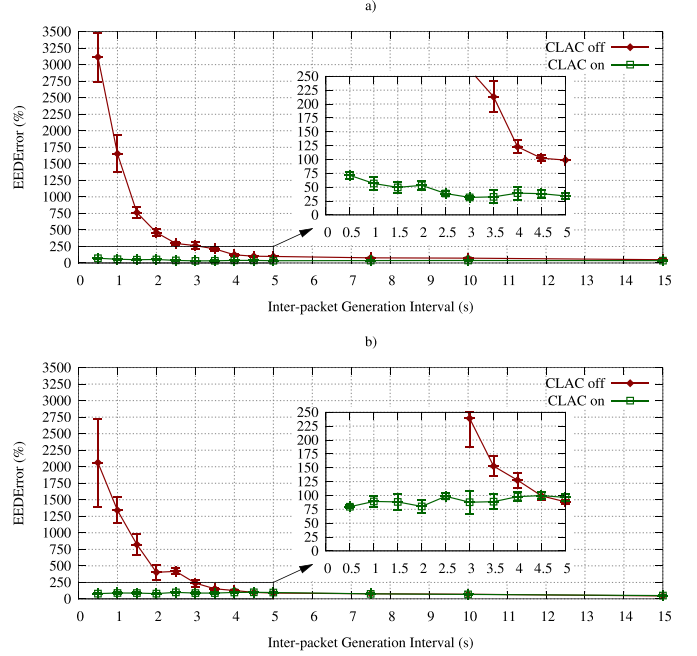


Fig. 8. EEDError for app1 and app2. a) EEDError for app1 (MaxEED=500 ms). b) EEDError for app2 (MaxEED=2000 ms).

the time it was either transmitting (Time_{Tx}^n), receiving (Time_{Rx}^n) or interfered (Time_{Int}^n). Using these times and the power constants defined in Table II, three types of energy components were calculated per each node n : the reception component (Energy_{Rx}^n), the transmission component (Energy_{Tx}^n), and the radio *off* component (Energy_{off}^n):

$$\text{Energy}_{Rx}^n = \text{Power}_{Rx} \times (\text{Time}_{Rx}^n + \text{Time}_{Int}^n) \quad (19)$$

$$\text{Energy}_{Tx}^n = \text{Power}_{Tx} \times \text{Time}_{Tx}^n \quad (20)$$

$$\text{Energy}_{off}^n = \text{Power}_{off} \times (\text{Time}_{Monitored}^n - \text{Time}_{on}^n) \quad (21)$$

The total energy spent in Joules (J) for the Number of Nodes (#Nodes) used per simulation was calculated as follows:

$$\text{Total Energy} (J) = \sum_{n=1}^{\#Nodes} (\text{Energy}_{Rx}^n + \text{Energy}_{Tx}^n + \text{Energy}_{off}^n) \quad (22)$$

VI. RESULTS AND ANALYSIS

In order to test the CLAC performance for different network loads, two types of applications were defined. The application 1 (*app1*) was defined with a MaxEED of 500 ms, and the application 2 (*app2*) was defined with a MaxEED of 2000 ms.

Fig. 8 presents the EEDError for *app1* and *app2*. The graphics show the results obtained as a function of the network load.

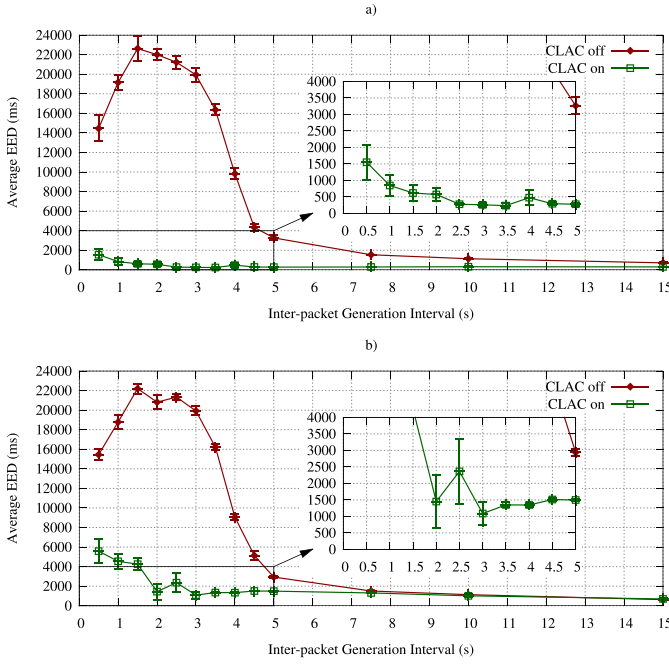


Fig. 9. Average End-to-End Delay. a) Average EED for app1 (MaxEED=500 ms). b) Average EED for app2 (MaxEED=2000 ms).

For high network loads (lower IGIs), the EEDError when using *app1* with CLAC *off* is higher than the error obtained with CLAC *on*. With CLAC *on* and for high network loads, the estimation error is significantly reduced (below 100%).

Fig. 9 shows the average EED with either CLAC *on* or *off* for both applications and as a function of network load; their standard deviations are also represented. The graphics show the same results for low network loads. For high network loads (IGIs below 5 s), the average EED is lower when CLAC is on. Despite the average values being above the MaxEED defined by each application for high network loads (IGIs below 2 s), when using CLAC *on* they are always lower those obtained with CLAC *off*.

Fig. 10 presents the average PRR for both applications and as a function of the network load. The results show that the average PRR with CLAC *on* is always smaller than the one obtained with CLAC *off*. The difference of results is more pronounced for low network loads and for short MaxEED (*app1*). A lower PRR is expected assuming that only the useful packets are delivered at the end application, discarding useless packets and saving energy along the way. The following results will help to attain this conclusion.

Fig. 11 shows the IPR (left side) and the OPR (right side) with either CLAC *on* or *off*, for both applications, and as a function of the network load. The results show that, for both applications and for high network loads, the IPR with CLAC *on* is higher than the IPR with CLAC *off*, meaning that a higher number of useful packets was delivered. Considering the OPR, for both applications and for any network load, the results show lower values with CLAC *on* than those obtained with CLAC *off*, meaning that a lower number of useless packets was transmitted.

Fig. 12 presents the PUR for both applications and as a function of the network load. The results show that the PUR is

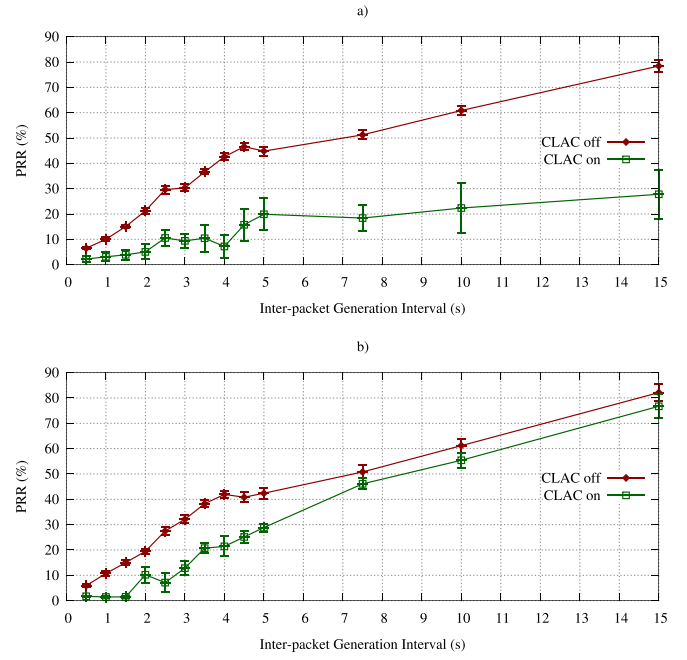


Fig. 10. Packet Reception Ratio. a) PRR for app1 (MaxEED=500 ms). b) PRR for app2 (MaxEED=2000 ms).

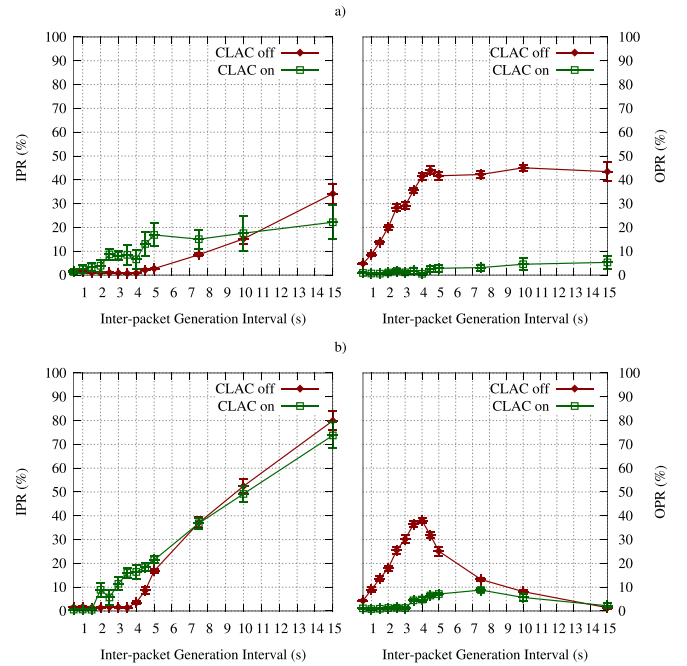


Fig. 11. In-profile Packets Ratio (left) and Out-of-profile Packets Ratio (right). (a) IPR and OPR for app1 (MaxEED=500 ms). (b) IPR and OPR for app2 (MaxEED=2000 ms).

always higher when CLAC is turned *on*. In other words, the network performance is improved when CLAC is turned *on*.

Fig. 13 shows the total energy consumed, per simulation, considering that the number of sent packets is constant for all simulations, using both applications and for different network loads. The results show that the energy spent with CLAC *on* is lower than the energy spent with CLAC *off*. This difference is higher in the case of the *app1*; greater energy savings are

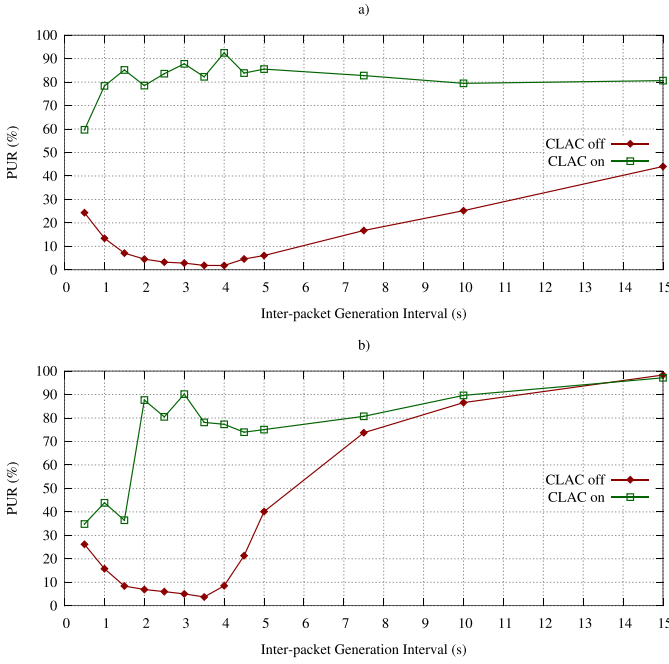


Fig. 12. Packet Usefulness Ratio. (a) PUR for app1 (MaxEED=500 ms). (b) PUR for app2 (MaxEED=2000 ms).

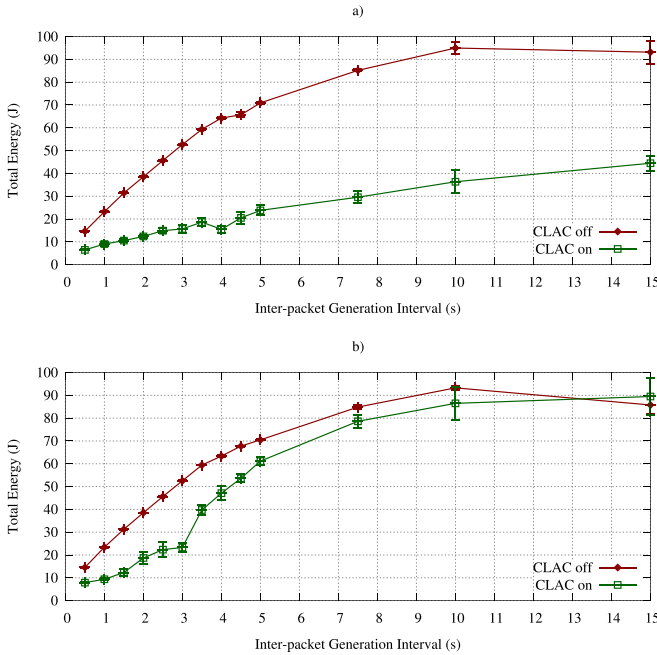


Fig. 13. Total energy consumed. (a) Total Energy for app1 (MaxEED=500 ms). (b) Total Energy for app2 (MaxEED=2000 ms).

obtained for lower MaxEED values because, in this case, a higher number of out-of-profile packet is expected.

A combined analysis of the results, shown in Fig. 11, Fig. 12, and Fig. 13, leads to the conclusion that when CLAC is *on* a higher number of in-profile packets is obtained and a lesser number of out-of-profile packets is found, while using less energy.

Fig. 14 shows the energy consumed by each node for both applications and as a function of the network load. Node 1 (central point of the graphic) consumes the most energy due to being the destination node, constantly receiving data packets

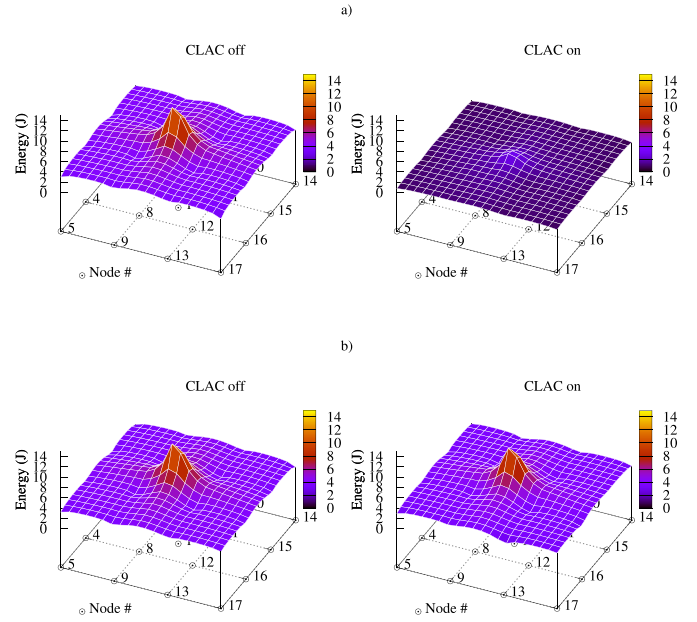


Fig. 14. Energy consumed mapped in each node when IGI equals to 5 s. (a) Energy for app1 (MaxEED=500 ms). (b) Energy for app2 (MaxEED=2000 ms).

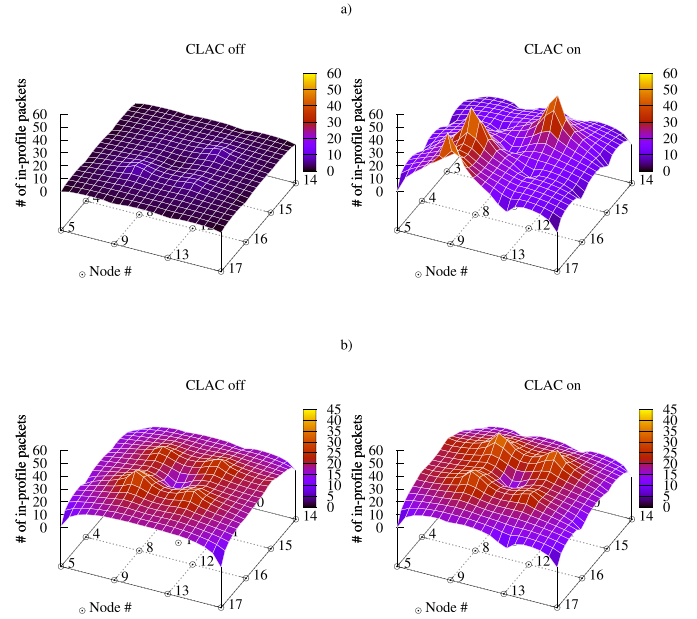


Fig. 15. Number of in-profile packets mapped in each node when IGI equals to 5 s. (a) Number of in-profile packets for app1 (MaxEED=500 ms). (b) Number of in-profile packets for app2 (MaxEED=2000 ms).

from the others nodes and sending routing packets in the reverse direction at a high rate. Apart from node 1, the nodes that consume more energy are those closer to the destination (node 7, 8, 11, and 12) and they can be identified by their position in the grid topology shown in Fig. 7. The results show that the energy savings for all nodes is greater when CLAC is *on*.

The Fig. 15 presents the number of in-profile packets identified at each node mapped onto the grid topology for both applications and as a function of the network load. The results show that, when CLAC is turned *on*, the number of in-profile

packets is higher. This effect is verified mainly on the nodes closer to the destination (central point) as their packets have a lower EED and can, more easily, meet the imposed deadline.

VII. CONCLUSION

A real-time monitoring multimedia application demanding strict EEDs is envisioned to be deployed in a WSN. Since the WSN devices have limited processing and communications capabilities, the network performance can be enhanced if we avoid at the sources the transmission of packets that will be useless for the receiver. At the same time, if the WSN devices only generate/transport useful data, the energy resources can be used more efficiency.

In this paper we propose a novel Cross-Layer Admission Control mechanism designed to enhance network performance and to increase the energy efficiency of a WSN in a grid topology, by avoiding the transmission of the potentially useless packets. The CLAC mechanism uses an estimation mechanism to preview packets EED and performs a decision to accept or drop a packet if it is expected to comply or miss the EED applications defined deadline. The CLAC mechanism was tested using different network loads and the results show that the CLAC enhances the overall network performance by increasing the number of useful packets and, as a side effect, improves the WSN energy efficiency, particularly in high network loads.

As future work, CLAC can be extended to support multiple applications. Per packet priorities can be defined according to their remaining time to reach the destination and an additional distributed packet scheduling mechanism can also be implemented to boost the network performance.

REFERENCES

- [1] *SELF-PVP: Self-Organizing Power Management for Photo-Voltaic Power Plants*. [Online]. Available: <http://www.cmuportugal.org/tiercontent.aspx?id=3374>, accessed May 4, 2015.
- [2] P. Pinto, A. Pinto, and M. Ricardo, "End-to-end delay estimation using RPL metrics in WSN," in *Proc. IFIP Wireless Days (WD)*, Nov. 2013, pp. 1–6.
- [3] T. Winter *et al.*, *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks*, IETF Standard RFC 6550, Mar. 2012. [Online]. Available: <http://www.ietf.org/rfc/rfc6550.txt>
- [4] P. Pinto, A. Pinto, and M. Ricardo, "RPL modifications to improve the end-to-end delay estimation in WSN," in *Proc. 11th Int. Symp. Wireless Commun. Syst. (ISWCS)*, Aug. 2014, pp. 868–872.
- [5] P. Pinto, A. Pinto, and M. Ricardo, "Delay accounting optimization procedure to enhance end-to-end delay estimation in WSNs," in *Proc. 8th Int. Wireless Internet Conf. (WICON) Symp. Wireless Veh. Commun.*, Lisbon, Portugal, 2014, pp. 14–19.
- [6] L. Fàbrega and T. Jové, "A review of the architecture of admission control schemes in the Internet," *J. Netw. Protocols Algorithms*, vol. 5, no. 3, pp. 1–32, 2013.
- [7] M. Mushtaq and T. Ahmed, "End-to-end QoS provisioning for real-time video streaming over SP-driven P2P networks using admission control," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2009, pp. 1–5.
- [8] S. Georgoulas, P. Trimintzios, G. Pavlou, and K. H. Ho, "Heterogeneous real-time traffic admission control in differentiated services domains," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, vol. 1, Nov./Dec. 2005, p. 6.
- [9] I. Orhan and T. Lindh, "Measurement-based admission control in wireless sensor networks," in *Proc. 4th Int. Conf. Sensor Technol. Appl. (SENSORCOMM)*, Jul. 2010, pp. 447–452.
- [10] O. T. Brewer and A. Ayyagari, "Comparison and analysis of measurement and parameter based admission control methods for quality of service (QoS) provisioning," in *Proc. Military Commun. Conf. (MILCOM)*, Oct./Nov. 2010, pp. 184–188.
- [11] J. Youn, S. Pack, and Y.-G. Hong, "Distributed admission control protocol for end-to-end QoS assurance in ad hoc wireless networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2011, no. 1, pp. 1–18, Nov. 2011. [Online]. Available: <http://link.springer.com/article/10.1186/1687-1499-2011-163>
- [12] W. Jiao, M. Sheng, K.-S. Lui, and Y. Shi, "End-to-end delay distribution analysis for stochastic admission control in multi-hop wireless networks," *IEEE Trans. Wireless Commun.*, vol. 13, no. 3, pp. 1308–1320, Mar. 2014.
- [13] A. Davy, D. Botvich, and B. Jennings, "Empirical effective bandwidth estimation for IPTV admission control," in *Real-Time Mobile Multimedia Services* (Lecture Notes in Computer Science), vol. 4787, D. Krishnaswamy, T. Pfeifer, and D. Raz, Eds. Berlin, Germany, Springer-Verlag, pp. 125–137, 2007.
- [14] X. Yin, X. Zhou, M. Pan, and S. Li, "Admission control with multi-constrained QoS providing in wireless sensor networks," in *Proc. Int. Conf. Netw. Sens. Control (ICNSC)*, Apr. 2010, pp. 524–529.
- [15] *Contiki OS*. [Online]. Available: <http://www.contiki-os.org>, accessed May 4, 2015.
- [16] O. Gnawali and P. Levis, *The Minimum Rank With Hysteresis Objective Function*, IETF Proposed Standard RFC 6719, Sep. 2012. [Online]. Available: <http://www.ietf.org/rfc/rfc6719.txt>
- [17] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-level sensor network simulation with COOJA," in *Proc. 31st IEEE Conf. Local Comput. Netw.*, Nov. 2006, pp. 641–648.
- [18] *Tmote Sky Project*. [Online]. Available: <http://www.snm.ethz.ch/Projects/TmoteSky>, accessed May 4, 2015.
- [19] Moteiv Corporation. (Feb. 2006). *Tmote Sky Datasheet*. [Online]. Available: <http://www.eecs.harvard.edu/~konrad/projects/shimmer/references/tmote-sky-datasheet.pdf>



Pedro Pinto received the degree in electrotechnical and computer engineering and the M.Sc. degree in communication networks and services from Porto University. He is currently pursuing the Ph.D. degree with the MAP-Tele Doctoral Program. He is an Invited Assistant with the Instituto Politécnico de Viana do Castelo, where he lectures telecommunications and computer networks related courses. He is also a Researcher with the Centre for Telecommunications and Multimedia, INESC TEC Research Institute.



António Pinto received the Ph.D. degree in electrical and computers engineering from Porto University. He is currently an Assistant Professor with the Escola Superior de Tecnologia e Gestão de Felgueiras, Politécnico do Porto, where he gives courses on computer networks, operating systems, network security, and digital forensics. He is also a Researcher with the Centre for Telecommunications and Multimedia, INESC TEC Research Institute.



Manuel Ricardo received the Ph.D. degree in electrical and computer engineering from Porto University in 2000. He is currently an Associate Professor with the Faculty of Engineering, Universidade do Porto, where he gives courses on mobile communications and computer networks. He also coordinates the Centre for Telecommunications and Multimedia, INESC TEC Research Institute.