Service Level Agreements for Wireless Sensor Networks: a WSN Operator's Point of View

Guillaume Gaillard, Dominique Barthel, Fabrice Theoleyre, Fabrice Valois

To cite this version:

Guillaume Gaillard, Dominique Barthel, Fabrice Theoleyre, Fabrice Valois. Service Level Agreements for Wireless Sensor Networks: a WSN Operator's Point of View. IEEE/IFIP NOMS - Network Operations and Management Symposium, May 2014, Krakow, Poland. hal-00998906

HAL Id: hal-00998906

https://inria.hal.science/hal-00998906

Submitted on 6 Jun 2014

Service Level Agreements for Wireless Sensor Networks: a WSN Operator's Point of View

Guillaume Gaillard[1,3], Dominique Barthel[1], Fabrice Theoleyre[2], and Fabrice Valois[3]

[1]Orange Labs R&D, Meylan, France. Email: firstname.lastname@orange.com
[2]ICube, Université de Strasbourg / CNRS. Email: theoleyre@unistra.fr
[3]Université de Lyon, INRIA, INSA-Lyon, Laboratoire CITI, F-69621.
Email:fabrice.valois@insa-lyon.fr

June 3, 2014

Abstract

The era of the Internet of Things brings complexity and deployment costs in smart cities, particularly in Wireless Sensor Networks (WSNs). Utilities such as gas or water providers are keen on delegating the management of the communications to specialized firms, namely WSN Operators, that will share the WSN resource among their various clients. WSN operators will use a functional architecture to manage the Service Level Agreements (SLAs), i.e. the Quality of Service (QoS) clauses they contract with their clients. WSN operators will need a robust and reliable technology in order to guarantee QoS constraints in a wireless environment, as in the industrial world. IEEE 802.15.4e Time Slotted Channel Hopping (TSCH) [2] is one good candidate. Moreover, the IETF experience in IP networks management is an important input for monitoring and QoS control over WSNs.

This article gives formal guidelines for the implementation of a SLA architecture for operated WSNs. It distinguishes the various formal algorithms that are necessary to operate a WSN according to SLAs, and determines which functional entities are necessarily technology-dependent. Detailed examples of such entities are developed in an IPv6 over IEEE 802.15.4e TSCH context, such as advocated in the IETF 6TiSCH Working Group [13].

1 Introduction

Implementation of Wireless Sensor Networks (WSNs) solutions with Quality of Service (QoS) becomes a necessity if a company aims at differentiating itself by providing guaranteed connectivity and performances. Moreover, because of the increasing traffic demands and the complexity of deploying multiple superimposed WSN, the use of shared infrastructure among multiple clients will be common sense. Thus, a WSN operator that provides wireless sensor connectivity as a resource to several clients, should be able to quantify the resource usage to establish contract-based business strategies.

In particular, smart cities radio environment is ever denser, and the use of non dedicated frequency bands makes it complex for clients and providers to contract guarantees on the quality of the communication process.

There exists a framework for this kind of contract: the Service Level Agreements (SLAs) [14]. SLAs settle the terms of breach, degradation, and compliance of the contracted clauses a client and an operator agree upon. They also include the financial and technical obligations (repayment, human interventions on the WSN) corresponding to all the situations that can occur.

We claim that a WSN operator will reduce cost by sharing its infrastructure among several clients. Yet, in order to manage all the clients applicative traffics, this sharing requires a SLA architecture that includes all the necessary functions for SLA Management. Clients must obtain guarantees in terms of network capacity, reliability, and delay.

The task is harder than in traditional wired networks because the radio medium is shared and no longer static, and because of the constraints of the sensor node (energy, memory, and processing capacity). Frequent changes in the network have to be monitored, and we must consider all layers at the same time, because they are strongly dependent. Management of SLAs requires features such as admission control or QoS monitoring, with more complex models and algorithms.

Moreover, sensor nodes are energy-constrained. The traffic load impacts the consumption of the nodes and their longevity. When a WSN operator connects new client nodes, it must be watchful in balancing the traffic load: it has to avoid the depletion of nodes for both itself and its clients.

We introduce a new SLA architecture, in which WSN operator monitors, manages, visualizes and grants its clients a personalized view on its network behavior. It may detect, anticipate, and correct the possible incidents. Non contracted usage of the resource can be controlled to preserve the others flows and the energy of the nodes.

In this paper, we go in depth in the architecture description and the functional entities. We highlight the scientific issues concerning our SLA architecture for WSNs. We also illustrate this architecture with examples based on IEEE 802.15.4e [2].

Wireless links in a changing environment are unreliable [3]. The technology must be robust to guarantee durable levels of service. We propose the use of IEEE802.15.4e Time Slotted Channel Hopping (TSCH) where it is easy to differentiate flows, and to have a relationship between the duty cycle of the nodes (i.e. energy consumption) and the usage of the network. The Internet Engineering Task Force (IETF), gathering expertise from academia and industry, is currently working on integrating this technology into the IP world, and take advantage of its experience (e.g. monitoring mechanisms). Our contribution fits in this context, defined by the IETF Working Group 6TiSCH [13].

The contribution of this paper consists in providing a formal view of the implementation of our new SLA architecture proposal:

- We define the basic entities and their interactions, that are requested to support SLA in the context of WSN;
- We explain how to deal with the admission of new flows on the WSN. We provide corresponding heuristics;
- We discuss the real-time monitoring design and we detail a framework for measures;
- We identify technology-dependent vs generic items. We provide implementation examples in the IETF 6TiSCH [13] open standard context (IPv6 over IEEE 802.15.4e TSCH).

Next Section will provide the State-of-the-Art in terms of SLAs, and especially on WSNs. We provide the scenario and challenges that we address in Section 3. We highlight the main features of our SLA architecture, and a recommended implementation in Section 4. Then, each functional entity is precisely detailed in Section 5 before we conclude and evoke future implementation perspectives.

2 Related work

2.1 Service Level Agreements in a nutshell

When a client buys services from a Service Provider, he expects them to be performed with the quality they have been sold for. This is particularly true in telecommunications, (e.g. on IP networks) where the services need to be precisely defined between the operator and the client. The contract describing technical and financial commitments of each party is named

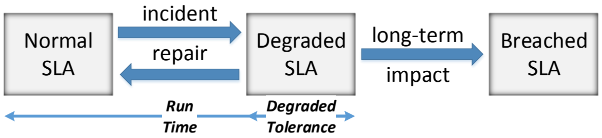

Figure 1: The lifecycle of Service Level Agreements.

a Service Level Agreement [14]. It also states the penalties the operator has to pay in case of a breach. Network operators, when signing SLAs, agree on collecting performance metrics that will prove that the contracted Quality of the Service is maintained. Different strategies exist [14]:

- Static SLA: the operator does not change its system; it takes a statistical risk of breaching the SLA in the future, as in the insurance prediction models;
- Provisioned SLA: the operator reserves resource for each SLA, and guarantees the resource will always have the requested characteristics;
- Adaptive SLA: the operator is able to change the network configuration. It handles both incidents and degradations of the network performance. SLAs are dynamically maintained.

This last approach implies that the SLA defines the maximum time of degradation of the service. This duration of degraded SLA [5] must be short: the operator must rapidly act on its infrastructure to restore its normal performance. The life cycle of the SLAs is illustrated in Fig. 1.

IBM has focused part of its research on the Web Services, and for this purpose, Keller and Ludwig have updated the notion of SLAs by proposing a formal framework [7]. They propose an XML-based decomposition of SLAs, that we have used as an inspiring model in the context of WSNs.

SLAs decompose themselves into Service Level Objectives (SLOs), namely logical sub-parts corresponding to a QoS requirement. For instance, *At least one gas index shall be collected each week for 99% of the client meters.* The SLOs are defined during a time frame, e.g. *each working day from 7 a.m. to 8 p.m.*.

2.2 QoS mechanisms for maintaining SLAs

In order to fulfill the requirements of the SLAs, the operator will necessarily use QoS mechanisms, and particularly on a wireless technology [16]:

- Call Admission Control (CAC);
- Resource Control (e.g. Radio Resource Management in radio networks);
- Network Monitoring;
- QoS Prediction;
- Configuration Management.

In wireless networks, the broadcasting nature of the channel mandates that the nodes control their transmissions in order to avoid collisions or interference. In WSNs, one can schedule the communications, according to algorithms such as the Traffic-Aware Scheduling Algorithm (TASA) [10], which uses matching and coloring heuristics to find a communication solution, given the nodes topology and the required traffic load.

In WSNs, radio link quality estimation is crucial to predict if the agreed-upon QoS will be met [3]. It permits, alongside with other raw measures (e.g. remaining energy of the nodes [8], traffic statistics), to elaborate the composite metrics defined in the SLOs. Some of these metrics prove that the QoS requirement of each SLO is met, and guide the real-time control actions on the network. Others are defined as Key Performance Indicators (KPIs), and sent to the clients and the WSN operator to show them high level statistics of the network.

The communication protocols for WSNs have for long been studied and implemented with the goal of meeting QoS constraints. A classification of the MAC layer mechanisms is proposed in [17]. At the routing layer, the IETF ROLL RPL [15] proposes the construction of the routing topology based on Objective Functions (OFs). The OFs include QoS constraints that the topology must meet at buildup. IETF CORE CoAP [11] is an application level protocol based on UDP and adapted for constrained nodes.

Finally, one may design WSNs applicative behavior in order to provide easier resource allocation for concurrent applications [8].

2.3 SLA management architectures

More than separate mechanisms, the SLA Management requires that all the above-listed functionalities be combined into a complete SLA architecture. [12] is an example of such an SLA architecture, dedicated to electricity distribution grids. Similarly, we intend to specify a SLA architecture for urban operated WSNs.

Some works have already considered similar ideas for generic WSNs. Del Cid et al. proposes a lightweight middleware platform, to manage flexible SLAs over WSNs [4], whereas the VITRO project [9] publicizes a software abstraction layer that enables virtualization in order to share sensor resource among various users. Finally, Octopus [6] is an open source, TinyOS-specific tool enabling monitoring (visualization) and human control of WSNs.

However, these approaches do not bridge the gap between the monitoring of network resources and node resources and the SLA Management: giving QoS guarantees for WSNs, at network level, is not their concern. We propose a complete and generic architecture for SLA Management over operated WSNs. The next section gives the general context on which we apply the proposed SLA Management architecture.

3 A general scenario for the SLA architecture

We describe the physical topology on which the proposed SLA architecture is based. We consider a running operated WSN. The network infrastructure is composed of two types of nodes (Fig. 2):

- client nodes that may be installed by the client himself (e.g. gas sensors);
- operator relay nodes owned by the WSN operator. They will forward all the client traffic between the source nodes and the sink(s).

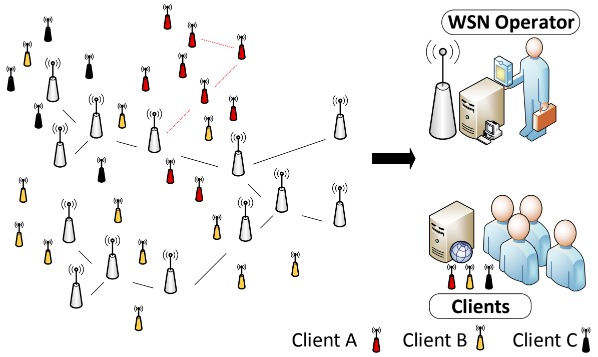

Figure 2: Nodes infrastructure of an operated WSN.

The implementation of the client firmware impacts the global energy consumption: this must be incorporated in the SLA. The amount of traffic the client nodes offer to the relays (and which ones) is a main part of the contract. This infrastructure permits flexibility: the WSN operator is able to manage the life cycle of the various SLAs, i.e. it controls the lifetime of the relay nodes by balancing the traffic on them, according to the SLAs setup times (when new flows appear) and terminations.

Our concerns are:

- what occurs when a new client comes with his nodes and asks to use the operator connectivity to access the data of his sensors?
- what amount of additional traffic is the relay network currently able to stand?

To answer this, the operator must have an infrastructure that:

- collects information about the network performance and current capacity;
- allows it to decide to admit or not the new client flows.

If this infrastructure was incorrectly implemented, the WSN operator could admit a new client without the necessary resource: the QoS would be degraded for all the clients, and the lifetime of the relays would decrease.

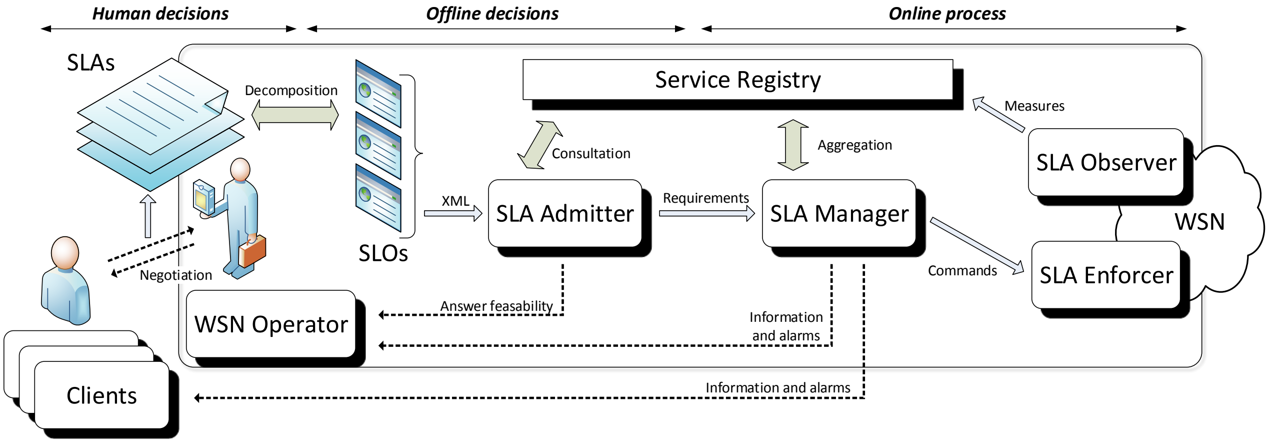

Figure 3: The SLA architecture for WSNs: main entities and their interactions.

When new relaying nodes are needed to accept a new client, the WSN operator must have access to all information that would help a human engineer to determine how many new nodes would have to be deployed, and where. The WSN operator must face two challenges:

- to cover the largest zone possible;
- to densely cover zones of high traffic.

Thus, the strategy of the WSN operator consists in a tradeoff between the number and the load of its clients. The initial deployment choices are crucial (re-deploying a lot of relays is onerous).

Finally, some new routes and configurations must be automatically computed to reflect the new requirements. Moreover, the WSN operator needs to monitor the energy remaining in its nodes in order to anticipate a failure (it is important to change depleted relays before a breach of service appears).

In this scenario, the architecture introduced in Fig. 3 allows the WSN operator to succeed in all previously described challenges.

4 Big picture of the SLA architecture for WSNs

We highlight the generic details of the SLA architecture illustrated in Fig. 3. Then, an overview of a possible implementation in a 6TiSCH context [13] is given.

4.1 Architecture steps and motivations

The SLA architecture is divided into three temporally-distinct parts :

- the human part, that consists in the (re-)negotiation of the terms of the SLAs;
- the offline part that deals with the admission of new services on the WSN;
- and the executive part (i.e. the online part) that manages the online processes of services on the WSN (this part must control and monitor the WSN parameters and maintain the QoS).

In this way, the SLA architecture encompasses all the aspects of the SLA Management: the left part of Fig. 3 supports week-scale or month-scale changes in human business decisions, whereas the right part promptly reacts to WSN changes.

4.1.1 Human processes

the negotiation of a SLA between the WSN operator and a client is the first step of the whole process. The client describes his high level applicative needs. He provides the location of the collection points, the characteristic of the traffic. The client selects the duration of the contract, and he specifies the KPIs. The clauses of the SLA are defined accordingly.

The WSN operator processes the SLA in its information system. The SLA decomposition into SLOs is described in Section 4.2. The set of SLOs is then transmitted to the SLA Admitter (See details in Section 5.3).

4.1.2 Admission Process

the SLA Admitter extracts the composite metrics from the submitted SLOs, their combination, their period of validity. It evaluates how much WSN resource the corresponding new SLA would use. Then, it consults the WSN state information in the Service Registry (See details in Section 5.2). It compares the current WSN availability with the global needed resource, including the requirements of the previously admitted SLAs. It concludes and informs the operator about the feasibility of the new submitted SLA.

If the current configuration of the WSN doesn't meet the new traffic requirements, the WSN can adopt new rules (self-adaptation), e.g. a new route setting, to make better use of the WSN resource and make the admission possible.

If the SLA Admitter rejects a new SLA request, we go back to the human part: a human manager should take appropriate business decisions (e.g. make the admission of strategic clients possible by deploying complementary devices). Else, it sends the new flow requirements to the SLA Manager (See details in Section 5.4).

4.1.3 Integration into the online process

the SLA Manager merges the requirements of the new flows with the information it already has in memory. It will use this input parameter as a base for its real-time analysis. This analysis leads to a set of instructions it will send to the SLA Enforcer (See details in Section 5.5). For instance, *Give node [25-35] 2 daily opportunities to transmit 200 B of data to sink C, in less than 300 s.*

The SLA Manager consults the Service Registry in real-time to get an up-to-date view of the network state. The Service Registry is fed by the SLA Observer (Section 5.1) whose role is to monitor the WSN. The analysis of the SLA Manager consists in:

- composing raw measures into composite metrics;
- comparing the composite metrics with the flows requirements;
- computing appropriate actions on the nodes configuration.

The SLA Manager reports some of the composite metrics, specified as KPIs in the SLOs, then in the SLA, to both the WSN operator and the clients. Finally, the SLA Enforcer updates in real-time the WSN nodes configuration, according to the SLA Manager's instructions.

We introduced five main functional entities, each one corresponding to a specific role, and interacting with the others via well defined interfaces. Future work will prove the efficiency of the contribution.

4.2 Proposal of SLA decomposition

A SLA is for the operator a logical composition of SLOs. A typical SLO is illustrated in Listing 1.

```
<ServiceLevelAgreement name="GasComp">
  <ServiceLevelObjective name="GasDayCR">
    <Validity>
      <Start> 2016-01-01 </Start>
      <End> 2030-12-31 </End>
    </Validity>
    <Expression>
      <Predicate xsi:type="Greater">
        <SLAParameter> DayCollectRatio </
    SLAParameter>
        <Value> 0.95 </Value>
      </Predicate>
    </Expression>
    <EvaluationEvent> Daily </EvaluationEvent>
    <DegradedTolerance> 2 </DegradedTolerance>
  </ServiceLevelObjective>
  <ServiceLevelObjective name="GasWeekCR">
    ...
```

```
</ServiceLevelAgreement>
```

Listing 1: XML source code of a Service Level Objective.

It expresses the requirement of GasComp, a gas company: 95% of its gas meter indexes must be collected each day. In this example, the SLO contains:

- A complex metric or a parameter: `DayCollectRatio`;
- A threshold (the required objective of service, e.g. 95%): the `Value` marker;
- An internal comparison operator (between bound and metric value): `Greater`;
- An external composition operator (how to combine this SLO with the other ones to form the full SLA);
- A temporal frame:
  - periods of applicability: `Validity`;
  - evaluation frequency and evaluation mode: `EvaluationEvent`;
  - maximum duration of the degradation of the SLO: `DegradedTolerance`.

Some SLOs should define the clients' obligations: the behavior of the applications must be contracted in the SLAs, and checked in real-time, in order to avoid abusing the WSN.

The complex metric is a high level view of the client requirement. It will be decomposed into a set of network composite metrics by the SLA Admitter (Section 5.3).

4.3 Application to a 6TiSCH environment

We argue that IEEE 802.15.4e technology is a good basis for the implementation of a SLA framework for WSNs. Indeed, the standard is designed for addressing QoS requirements of a large number of application schemes [2].

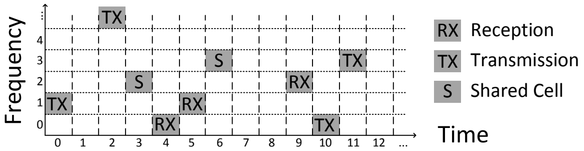

Figure 4: Duty cycle of a TSCH node.

4.3.1 Context

IEEE 802.15.4e has several defined modes, depending on the application requirements. We will here focus on the Time Slotted Channel Hopping (TSCH) mode. The industry world has already used channel hopping technologies such as WirelessHart, and ISA100.11a. TSCH mode of IEEE 802.15.4e MAC standard is newer, not application-specific, and more flexible than its ancestors [10]. It gives a reliable base for the implementation of QoS mechanisms.

6TiSCH [13] is working on integrating these WSN in the IP world. This context is advantageous because all the standard mechanisms and protocols of the IP stack may be adapted to our needs (e.g. monitoring, storing the information, etc.).

4.3.2 Definition

if we consider the Channel Hopping mode of IEEE 802.15.4e, the communication is divided into time-slots during which nodes can communicate on a defined frequency. Time-frequency blocks are named cells of communication. Nodes are scheduled to receive or transmit packets on dedicated cells. The schedule may also include shared cells where the nodes participate in a contention if they need to communicate. The schedule is applied on a certain quantity of time-slots, namely a slot-frame, and is periodically repeated. The nodes sleep during unscheduled cells (See Fig. 4).

In the centralized approach of 6TiSCH, the Path Computation Element (PCE) [13] can change the schedules of the nodes. Whereas in a distributed scheme, the nodes may run multiple instances of RPL [15], and autonomously update the topology.

4.3.3 Application

In 6TiSCH context, the monitoring function is distributed in each node: each node transmits the raw measures toward the Service Registry, using a protocol for the exchange of the monitoring information, e.g. the IETF CORE CoAP [11]. RPL routing metrics are also a useful information to monitor. The values can be passively collected by the system. The monitoring may be integrated in the communication schedule: some cells may be used to collect the measures.

The enforcement can be addressed in two ways:

- A central PCE may generate the cell allocation schedule (the scheduling algorithm being for instance the Traffic-Aware Scheduling Algorithm (TASA) [10]) according to the service requirements;

- inside the RPL routing protocol, the DODAG roots would initiate distributed updates of the routing topology.

Guaranteed flows would be assigned dedicated cells, but the WSN operator can give some permeability to non-guaranteed traffic by installing shared cells.

Finally, the 6TiSCH context simplifies the admission process. The knowledge of the schedules of the nodes simplifies the QoS predictions, since it gives a clear view on unoccupied remaining cells. Note that it is also easier to predict the node energy consumption.

We will now get into formal descriptions of the generic details of each functional entity, and their interfaces with the rest of the architecture.

5 Algorithms for the different building blocks

We explain here in details how the different processes work together, and we separate generic algorithms from technology-dependent implementations.

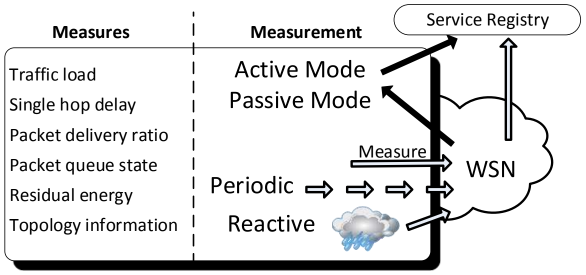

Figure 5: The SLA Observer.

5.1 SLA Observer: monitoring of WSNs and applications

In order to manage the network, the WSN operator needs to collect precise information about the performance of the nodes. The SLA Observer (Fig. 5) aims at measuring this information and to transmit it to the Service Registry.

When designing such a monitoring algorithm for WSNs, a tradeoff between the energy consumption and the precision of the measure has to be found. The client traffic is so heterogeneous, (the applications can generate from one transmission a day to several packets each hour [1]), that the measure payload can exceed the data payload at some relay nodes. Hence, the measure algorithms have to be well designed not to waste energy.

5.1.1 What we measure

raw measures are locally built on each node. The SLA Observer typically depends on the node technology, because the measurement will take place in the firmware (not all nodes can measure the same Link Quality Indicators (LQI)) or on the software (average packet delivery ratio (PDR) may be differently computed) [3]. Measures take place at different protocol layers (e.g. MAC single-hop delay, routing topology information, packet queue state of the nodes).

The SLA architecture requires the following end-to-end or local generic metrics:

- delay;

- packet loss rate;

- residual energy;
- traffic load.

5.1.2 How we measure

some measurements may be periodically driven. The period impacts the granularity of the measure: large sampling (e.g. one measure every hour) is more energy-efficient than short sampling but does not give precise information. Clients may need a specific measure precision, and that could be a reason for differentiation among providers: the WSN operator should monitor precisely enough its network performance, to satisfy the client requirements.

Other measurements should only be held in specific occasions (e.g. when the application layer requests a specific parameter) or depending on previous measurements (e.g. react when a previous value has reached a threshold, or when no change has been seen for long).

5.1.3 How measures are collected

the SLA Observer has to store the local measures in the Service Registry (See details in Section 5.2). Depending on the technology, it can:

- use a dedicated channel of communication (out-of-band) with independent protocol stack. This requires that the nodes support multiple technologies. For instance, the measures could be placed in dedicated CoAP messages [11];
- use passive monitoring: no explicit traffic is generated on the WSN, a server in the WSN operator core network centrally observes all the transiting packets;
- use active monitoring: either send dedicated monitoring packets, or piggyback the measures onto the data packets. When piggybacking, the WSN operator may need to inspect data packets of the client. Thus, confidentiality should be handled carefully;
- measures may be retrieved either on-demand (with a specific request message) or periodically. For instance, the DSME mode of IEEE 802.15.4e provides periodic link status reports to a central entity;
- use hybrid monitoring: in order to save bandwidth, the SLA Observer can use passive monitoring for normal activity, and reactive dedicated packets in case of incidents or specific applicative needs.

5.2 Service Registry: a database for raw measures

The Service Registry contains all the information about the network performance. The raw measures produced by the SLA Observer can't be stored in the WSN nodes, that are memory-constrained (they can only keep the current local information). Moreover, the raw measures must be consulted by the other entities of the architecture (SLA Manager, SLA Admitter), without occupying the WSN resource. Hence, the Service Registry must be centralized (e.g. a database in the WSN operator core network).

We keep the raw measures all along the SLA duration. Thus, if the WSN operator changes the algorithms of the SLA Manager and SLA Admitter, new policies can be applied taking into account the whole history of the WSN.

Given that we monitor various sensor nodes, during long periods, the measures have to be accurately identified and classified. We propose the following log format for the raw measures:

- timestamp: the time of the measure;
- measurer (id): the producer of the measure;
- entity (ids): which entities have been measured;
- type: abstract unit (See Section 5.1.1);
- mode: how it has been measured (See Section 5.1.2);
- value: the measure itself.

Producing the timestamp is challenging since measures are done locally, with local, distributed clocks.

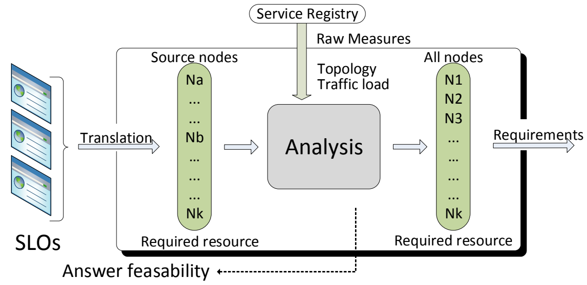

Figure 6: The SLA Admitter.

Nodes may be synchronized (e.g. in IEEE 802.15.4e TSCH). The impact of the clock drift on the precision of the measure is then limited, while the energy cost increases. Passive monitoring, held outside of the WSN, does not suffer synchronizing issues, but the measures are less precise.

Note that if the WSN operator updates the SLA Observer, new Measure Types can be directly written if needed. A more detailed example of how raw measures may be collected from the Service Registry is given in the following XML-based formalism:

```
<MeasureRequest ID="12478">
  <Timestamp>34567890</Timestamp>
  <Measurer>
    <RelayNode> 46 </RelayNode>
  </Measurer>
  <Measured>
    <RelayNode> 46 </RelayNode>
    <RelayNode> 47 </RelayNode>
  </Measured>
  <Type>
    <Layer> MAC </Layer>
    <Range> SingleHop </Range>
    <metric> Delay </metric>
  </Type>
  <Mode name ="Periodic">
    <Period> 60 </Period>
  </Mode>
  <Value>0.37</Value>
</MeasureRequest>
```

Listing 2: XML raw measure exchange format.

5.3 SLA Admitter: Controling the Admission of new clients

The SLA Admitter (Fig. 6) acts as the interface between the human decisions and the SLA architecture. It answers the new SLA installation requests by analyzing their feasibility, regarding current WSN state. Hence, the installation of unfeasible SLAs should be forbidden.

The SLA Admitter first manages the translation of the SLOs into resource needs for each relay node. The result of this translation includes:

- the set of relay nodes connecting the client (e.g. kind of *virtual source nodes*, owned by the operator);
- the amount of traffic the new SLA guarantees on the WSN operator source nodes;
- the delay requirements of the new client flows;
- the time frame of these guarantees (e.g. working hours) and their duration.

Once the traffic shape is determined, the SLA Admitter consults the Service Registry. Based on the raw measures, it computes the current estimation of the network state:

- the remaining capacity of the WSN operator source nodes;
- the physical topology information (e.g. the connectivity graph of the WSN);
- the delay each route induces;
- the variations of this information during the time frame of the composite metrics.

The analysis has to predict the impact of the new traffic flows on the relay nodes (energy, occupation, etc.). It must consider changes in the WSN configuration (e.g. new routing paths, load balancing on the relays, etc.).

These QoS predictions on WSNs are complex, given the variability of the parameters, and still represent a research challenge [8, 17].

Note that the granularity of the SLO may vary according to the cost strategy of the WSN operator. Moreover, the WSN technology impacts the precision and form of the raw measures. Hence, the SLA Admitter is technology-dependent.

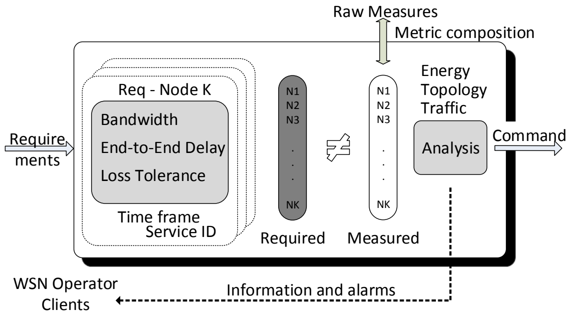

Figure 7: The SLA Manager.

In cases where the analysis fails to install the new SLA, the SLA Admitter must give a complete report of the reject to the WSN operator. The report permits to determine where are the hard points that have prevented the admission, and what can be done about it (e.g. deployment of new relays).

The SLA Admitter accepts the SLA request by giving the corresponding node requirements to the SLA Manager. Indeed, the output of the analysis is the set of requirements implied by the SLA installation, for all the relay nodes.

5.4 SLA Manager: Controlling and reporting the WSN state

The SLA Manager represents the intelligence of the system. It interfaces the offline part and the online part of the SLA architecture.

5.4.1 interface with the SLA Admitter

the SLA Manager stores for each relay node and each client flow the new incoming resource requirements:

- the expected traffic load;
- the requirements;
- the time frame of application (i.e. the time bounds during which the requirements apply).

5.4.2 analysis of performance

it collects the raw measures from the service Registry, and aggregates them into composite metrics that correspond to the set of requirements it has stored. For instance, it evaluates the end-to-end delay from the MAC single hop delay values on the path, taking into account the Packet Delivery Ratio (PDR) and the buffer state of the nodes. Then the SLA Manager compares the measured state of the network (i.e. the composite metrics) with the set of requirements: if the requirements are not met, the corresponding SLA is considered degraded.

With such analysis, the SLA Manager may autonomously change the configuration to resolve the incident. If no parameter can be thwarted to meet the contracted QoS, an alarm is raised for human decisions to be taken (manual intervention, renegotiation, etc.). QoS provisioning mechanisms in WSNs still constitute a research challenge [10].

The SLA Manager must also monitor the energy level of the relay nodes. It reports an alarm to the WSN operator when necessary (when a threshold is reached).

5.4.3 reporting

besides the alarms, the SLA Manager reports some of the composite metrics, denoted as KPIs, to the clients and the WSN operator to show them specific views on the behavior or the state of the WSN. For instance, a client may want to supervise the applicative delay for packet on specific nodes, in order to write its own commercial offer, or for troubleshooting purpose on specific sensors.

5.4.4 action command

the SLA Manager triggers changes in the technology-dependent configuration of the WSN operator relay nodes (e.g. *Give node [25-35] 2 daily opportunities to transmit 200 B of data to sink C, in less than 300 s*). The generic set of instructions includes:

- the IDs of the nodes concerned by the change;
- the generic characteristics of the change;

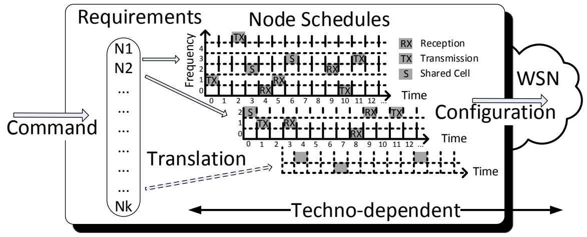

Figure 8: The SLA Enforcer.

The message exchange between the SLA Manager and the SLA Enforcer depends on the nature of the SLA Enforcer.

5.5 SLA Enforcer: set up networking configuration

The SLA Enforcer controls the actual configuration: it provides each node its routing and MAC instructions, translating the SLA Manager requirements. This entity is consequently technology dependent.

While the SLA Manager gives generic requirements, the SLA Enforcer translates them in protocol-dependent instructions. For instance, the SLA Enforcer will construct a DODAG with an objective function and some routing metrics directly translated from the requirements of the SLA Manager. Note that these changes may impact one specific relay, a group of nodes, or even the whole operated WSN.

Fig. 8 illustrates an implementation of the SLA Enforcer in a 6TiSCH context [13]. The node schedules are directly modified in the PCE, according to the generic instructions. Then the PCE transmits the schedules to the nodes.

Note that the SLA Enforcer can also behave independently of the SLA Manager instructions. For instance, it may decide to locally balance the traffic over some nodes, or to carry out local Admission Control on unexpected traffic flows. In this case, configuration loops between the SLA Manager and the SLA Enforcer have to be avoided.

6 Conclusion

Future increase in the Internet of Things area, particularly in urban environment, will dynamize the WSN market. Specific actors denoted in this paper as WSN operators will sell wireless connectivity that will be shared among various clients. They will gain profitability by positioning themselves on specific market sectors: this differentiation will be based on their business strategy, i.e the price policy, and on the QoS guarantees they offer. They will use the SLA framework to settle contractual clauses about these guarantees.

We propose a complete roadmap to build an architecture for SLA management on WSNs. We distinguish five functional entities: the SLA Observer, the Service Registry, the SLA Admitter, the SLA Manager and the SLA Enforcer. We argue that they are necessary mechanisms for guaranteeing QoS, thus maintaining contracted levels of service. Our SLA architecture defines efficient interactions between these entities, and provide the formal algorithms they apply / run. We focus our examples of implementation on existing standards, particularly on the robust IEEE 802.15.4e TSCH technology.

As a future work, we plan to provide an entire implementation. We will evaluate the efficiency of the admission and management heuristics, the composition of network measures and the monitoring energy cost.

References

[1] Spectrum Requirements for Short Range Device, Metropolitan Mesh Machine Networks (M3N) and Smart Metering (SM) applications. *ETSI TC ERM, TR 103 055, v1.1.1*, September 2011.

[2] IEEE Standard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer. *IEEE Std 802.15.4e-2012 (Amendment to IEEE Std 802.15.4-2011)*, pages 1–225, 2012.

[3] Nouha Baccour, Anis Koubâa, Luca Mottola, Marco Antonio Zúñiga, Habib Youssef, Carlo Alberto Boano, and Mário Alves. Radio Link Quality Estimation in Wireless Sensor Networks: A survey. *ACM Transactions on Sensor Networks*, 8(4):1–33, September 2012.

[4] Pedro Javier Del Cid et al. DARMA: adaptable service and resource management for wireless sensor networks. In *Proceedings of MidSens*, pages 1–6. ACM, 2009.

[5] W. Fawaz et al. Service Level Agreement and Provisioning in Optical Networks. *IEEE Communications Magazine*, 42(1), January 2004.

[6] Raja Jurdak et al. Octopus: monitoring, visualization, and control of sensor networks. *Wireless Communications and Mobile Computing*, 11(8):1073–1091, August 2011.

[7] Alexander Keller and Heiko Ludwig. The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services. *J. Netw. Syst. Manage.*, 11(1):57–81, March 2003.

[8] Wei Li, Flvia C. Delicato, Paulo F. Pires, and Albert Y. Zomaya. Energy-efficient task allocation with quality of service provisioning for concurrent applications in multi-functional wireless sensor network systems. *Concurrency and Computation: Practice and Experience*, pages n/a–n/a, 2013.

[9] Monica Navarro et al. VITRO Architecture: Bringing Virtualization to WSN World. In *2011 IEEE Eighth International Conference on Mobile Ad-Hoc and Sensor Systems*, pages 831–836. IEEE, October 2011.

[10] M.R. Palattella, N. Accettura, L.A. Grieco, G. Boggia, M. Dohler, and T. Engel. On Optimal Scheduling in Duty-Cycled Industrial IoT Applications Using IEEE802.15.4e TSCH. *Sensors Journal, IEEE*, 13(10):3655–3666, 2013.

[11] Zack Shelby, Kart Hartke, Carsten Bormann, and Bebeve Frank. Constrained Application Protocol (CoAP). IETF CoRE Working Group, feb 2011.

[12] Ward Snoeck. Quality of Service Monitoring in the Low Voltage Grid by using Automated Service Level Agreements. Master's thesis, KTH, Industrial Information and Control Systems, 2013.

[13] Pascal Thubert et al. An Architecture for IPv6 Over Time Slotted Channel Hopping. IETF ID draft-thubert-6tsch-architecture (Work In Progress), 2013.

[14] D.C. Verma. Service level agreements on IP networks. *Proceedings of the IEEE*, 92(9):1382–1388, 2004.

[15] T. Winter, P. Thubert, and al. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. RFC 6550, IETF, March 2012.

[16] Xu Yang, J. Bigham, and L. Cuthbert. Resource Management for Service Providers in Heterogeneous Wireless Networks. In *Wireless Communications and Networking Conference, 2005 IEEE, New Orleans, LA USA*, volume 3, 2005.

[17] M. Aykut Yigitel, Ozlem Durmaz Incel, and Cem Ersoy. QoS-aware MAC protocols for wireless sensor networks: A survey. *Computer Networks*, 55(8):1982–2004, June 2011.