Universal Integration of the Internet of Things through an IPv6-based Service Oriented Architecture enabling heterogeneous components interoperability

Grant agreement for: Collaborative project
Grant agreement no.: 288445
Start date of project: October 1st, 2011 (36 months duration)

**Deliverable D6.4**

Innovative interactions between STIS and IPv6 through IoT6 architecture

<table>
<thead>
<tr>
<th>Contract Due Date</th>
<th>31/03/2014</th>
</tr>
</thead>
<tbody>
<tr>
<td>Submission Date</td>
<td>30/04/2014</td>
</tr>
<tr>
<td>Version</td>
<td>v1.0</td>
</tr>
<tr>
<td>Responsible Partner</td>
<td>KAIST</td>
</tr>
<tr>
<td>Author List</td>
<td>Daeyoung Kim, Seong Hoon Kim, Minkeun Ha, Giang Nam, Sehyeon Heo,</td>
</tr>
<tr>
<td>Dissemination level</td>
<td>PU</td>
</tr>
<tr>
<td>Keywords</td>
<td>Smart Thing Information Service, Smart Thing Identification, IPv6, EPC, IoT6</td>
</tr>
</tbody>
</table>

Project Coordinator: Mandat International (MI)
Sébastien Ziegler     sziegler@mandint.org
Table of Contents

Executive summary ................................................................................................................ 5
General overview ................................................................................................................... 5
Summary of the proposed advantages .................................................................................... 5

1. Overview ............................................................................................................................ 6
   1.1. Purpose and scope of the document ............................................................................ 6
   1.2. Document outline ....................................................................................................... 6

2. Introduction ....................................................................................................................... 6
   2.1. Background .................................................................................................................. 7
   2.2. Smart Thing Information Service (STIS) ................................................................... 8
   2.3. Active RFID Tags (ARTag) over 6LoWPAN ............................................................ 10
   2.4. A Survey of Related Work ........................................................................................ 11

3. Design of 6LoWPAN-based Active RFID Networks .................................................... 13
   3.1. Overview ................................................................................................................... 13
   3.2. System Architecture ................................................................................................ 13
       3.2.1. 6LoWPAN LLRP Reader (6LoL) ......................................................................... 14
       3.2.2. 6LoWPAN Router as an Antenna (6LoRA) ....................................................... 17
       3.2.3. Active RFID Tag (ARTag) ................................................................................ 17
   3.3. Functional Description .............................................................................................. 18
       3.3.1. STID and IPv6 address management scheme .................................................... 18
       3.3.2. Protocols for address registration and update of IoT devices .............................. 18
       3.3.3. Sensor information exchange between IoT6 and STIS with 6LoRNet ............... 22

4. Test cases .......................................................................................................................... 23
   4.1. LLRP Wrapper Conformance Test ............................................................................. 23
   4.2. LLRP Wrapper Delay (without 6LoWPAN) ............................................................... 24
       4.2.1. Environment Settings ....................................................................................... 24
       4.2.2. Test Scenario .................................................................................................... 24
       4.2.3. Test Results ...................................................................................................... 25
   4.3. Integrated System Delay (without 6LoWPAN) .......................................................... 26
       4.3.1. Environment Settings ....................................................................................... 26
       4.3.2. Test Scenario .................................................................................................... 26
       4.3.3. Test Results ...................................................................................................... 27
   4.4. Active Tag Network Construction Delay ................................................................. 28
D6.4 Innovative interactions between STIS and IPv6 through IoT6 architecture

4.4.1. Environment Settings ........................................................................................................ 28
4.4.2. Test Scenario .................................................................................................................. 28
4.4.3. Test Results ..................................................................................................................... 29
4.5. RFSurvey Delay .................................................................................................................... 29
4.5.1. Environment Settings ....................................................................................................... 29
4.5.2. Test Scenario ................................................................................................................... 30
4.5.3. Test Results ..................................................................................................................... 30
4.6. Test Case Scenarios ............................................................................................................. 30
5. Conclusion .......................................................................................................................... 34
6. References .......................................................................................................................... 35
7. List of Acronyms .................................................................................................................... 36

List of Figures

Figure 1. STIS architecture with three different interfaction points ...................................................... 8
Figure 2. Active RFID tags over 6LoWPAN .................................................................................... 10
Figure 3. The penetration of active RFID in different application sectors over the next ten years ...... 11
Figure 4. General RFID networks ................................................................................................. 13
Figure 5. 6LoWPAN-based Active RFID Networks ......................................................................... 13
Figure 6. 6LoWPAN System Architecture ...................................................................................... 14
Figure 7. Each Reader router is mapped into a single antenna of an RFID reader ............................. 14
Figure 8. LLRP Wrapper Structural Overview ................................................................................ 15
Figure 9. Simplified Class Diagram of 6LoL .................................................................................... 15
Figure 10. C1G2 Air Protocol Cross-reference Table [referred LLRP 1.1 spec] .................................... 16
Figure 11. Network Stack of the 6LoRA ......................................................................................... 17
Figure 12. Network Stack of the Active RFID Tag .......................................................................... 17
Figure 13. Inventory operation of conventional RFID device and the naïve approach for 6LoRNet ... 19
Figure 14. Inventory Operation ..................................................................................................... 20
Figure 15. RF Survey Operation ................................................................................................... 21
Figure 16. Access Operation ......................................................................................................... 22
Figure 17. Graph of tag processing time .......................................................................................... 25
Figure 18. Integrated System Delay Test Scenario ......................................................................... 26
Figure 19. Usage Model for the Integrated System Delay Test Scenario ........................................... 27
D6.4 Innovative interactions between STIS and IPv6 through IoT6 architecture

Figure 20. Test Results Graph of Average Integrated System Delay (T)................................................ 28
Figure 21. Active Tag Network Construction Delay Test Environment ................................................ 29
Figure 22. Test Result Graph of Active Tag Network Construction Delay............................................. 29
Figure 23. RFSurvey Delay Test Result Sample...................................................................................... 30
Figure 24: 6LoWPAN Border Router LLRP Wrapper.............................................................................. 31
Figure 25: F&C Server Logs.................................................................................................................... 31
Figure 26: Capturing Apps catching the events..................................................................................... 32
Figure 27: Live stream of events in STIS web interface......................................................................... 33
Executive summary

General overview

As a non-IP-based smart things, IEEE 802.15.4-based active RFID (Radio-frequency identification) tags with STID (Smart Things Identifier, i.e., EPC (Electronic Product Code) ID) will be introduced. To enable interaction between these smart things and STIS (Smart Things Information Service), 6LoWPAN (IPv6 over Low Power Wireless Personal Area Networks) is employed as a vehicle to integrate non-IP-based smart things with STIS. As a result, coverage of IEEE 802.15.4-based active RFID networks can be easily extended with the aid of 6LoWPAN networks, which are called 6LoWPAN-based active RFID networks. Also, by integrating 6LoWPAN gateways with LLRP (Low-level Reader Protocol) readers, smart things and STIS can communicate with each other without any modification through 6LoWPAN-based active RFID networks. This document will show the feasibility of the design and implementation of the proposed 6LoWPAN-based active RFID networks. Finally, it will test and present the result of the proposed architecture in the frame of the IoT6 architecture.

Summary of the proposed advantages

6LoWPAN-based active RFID networks bring many benefits to support interaction between non-IP-based smart things with STID (i.e., IEEE 802.15.4-based active RFID tags) and STIS. It enables the easy extension of active RFID network coverage for IEEE 802.15.4-based active RFID networks by leveraging the key features of 6LoWPAN. For instance, simply putting more routers acting as antennas, the dead zone of any smart space can be easily covered, making smart things easily detected anywhere with low cost. In the meanwhile, since smart things and STIS can communicate with each other without any modification through 6LoWPAN-based active RFID networks, IoT6 applications can benefit from information about those smart things through STIS.

In a summary, since many non-IP-based active RFID tags (IEEE 802.15.4-based active RFID tags) exist in the real world, the proposed STIS can broaden its spectrum to reach diverse types of smart things with its own STID. Accordingly, IoT6 applications can access static and dynamic information about those non-IP-based smart things and thus increase their usability thanks to the 6LoWPAN-based active RFID networks.
1. Overview

1.1. Purpose and scope of the document

The IoT6 research project aims at researching and exploiting the potential of IPv6 and IPv6-related technologies to develop an open service-oriented architecture overcoming the current IoT (Internet of Things) fragmentation.

This document is the result of Task 6.3 (T6.3), which explores innovative interactions between STID and IPv6 through IoT6. As described in the Description of Work (DoW), the aim of this task is to research and develop an innovative interaction and integration between IPv6 and STIS. Thus, this deliverable D6.4 reports a design specification describing innovative interaction between Smart Things Information System (STIS) using Smart Things ID (STID) and IPv6.

Unlike deliverable D6.3 which deals with CoAP/JSON/oBIX-based interaction between STIS and IoT6, this deliverable reports the development of 6LoWPAN-based active RFID networks, which enables smart things with STID to interact with STIS through the aid of 6LoWPAN and thus allows IoT6 applications (e.g., CMS or Control and Monitoring System) to discover and observe smart things of interest. To achieve this, an IPv6 and STID address management scheme as well as their registration and update operations are addressed with 6LoWPAN. Also, by allowing smart things to have their own STID in their tag memory, the proposed architecture enables non-IP-based as well as IP-based smart things to interact with STIS.

The following are the 3 key aspects of this deliverable:

- Firstly, this deliverable describes a novel architecture for interaction between smart things with STID and STIS via 6LoWPAN where smart things with STID can be either non-IP-based or IP-based.
- Secondly, the way of managing the IPv6 and STID for accessing smart things based on the architecture is described.
- Thirdly, this deliverable reports the results of test cases regarding the proposed architecture.

1.2. Document outline

Since this document is based on STIS, the Introduction in Section 2 provides the background of our research. In Section 3, the detailed design of 6LoWPAN-based active RFID networks including their operations is presented. Section 4 presents test cases for interaction between STIS and 6LoWPAN-based active RFID networks, which validates interaction between STIS using STID and IPv6. Section 5 concludes this document.

2. Introduction

This section provides the background which is necessary to understand the purpose of this deliverable and also introduces STIS and the concept of active RFID tags over 6LoWPAN,
followed by related works.

2.1. Background

The Internet of Things (IoT) is the vision that interconnects smart things, which are everyday objects empowered by communication and computing capabilities, and provides humans with more fine-grained real-world information. According to Gartner research, the IoT will include 26 billion units installed by 2020. Eventually, smart things in the IoT may scale up to trillions of networked objects. By being equipped with such smart things in the real-world, the IoT will enable real-time decision making and many comfort services like healthcare, connected home, and smart grid, etc. In particular, according to a recent survey from Zebra research (refer to the key summary in text box below), many companies consider that one of the valuable IoT components is RFID devices such as active tags which enable supply chain, transaction tracking, and asset location issues.

"Barcodes and RFID devices are valuable components of IoT solution implementations. Firms consider many different devices valuable to implementing IoT solutions. Specifically, 72% of enterprises state that barcodes are valuable to enable IoT solutions; 71% say the same thing about real-time location tracking technologies like active RFID and 58% of firms consider passive RFID devices as valuable IoT solution technologies."

"IoT solutions address supply chain, transaction tracking and asset location issues. Enterprises seek to address a variety of issues by implementing IoT solutions (see Figure 5). Between 50% and 56% of surveyed enterprises stated that IoT solutions would help their organization address supply chain visibility, transaction tracking, and asset location issues. Other issues that 44% to 48% of surveyed enterprises can address with IoT solutions include identifying assets, reducing inventory levels, and minimizing theft of and damage to objects or assets."

"Supply chain visibility and asset-tracking applications are a key focus for many firms. Supply chain visibility, asset location, and transaction tracking are key applications. Supply chain visibility applications help reduce working capital, improve fixed asset utilization, and improve customer service. Asset-tracking applications are used to reduce the time to locate assets, improve asset utilization, throughput, and customer service."

As stated, the most demanding IoT applications such as supply chain visibility, transaction tracking, and asset location require active and passive RFID technologies, and smart thing information services (STIS) like EPC networks which is a global architecture and interface standard to process RFID data from globally dispersed RFID reader infrastructure and which allows to share RFID data among associated stakeholders.

1 http://www.gs1.org/epcglobal
To enable these, every smart thing should have at least one ID and one IP address so that a single smart thing can be uniquely identified and ubiquitously communicated with other smart things. In this regard, a smart thing’s identifier (STID) like EPC ID allows smart things to be distinguished from each other. By using STID as a key, we can retrieve smart things static information like product description, homepage, etc., as well as dynamic historical information from smart thing information services (STIS). However, to reach these STISs or individual smart things, it is essential that scalable IP addressing scheme should be used. In this regime, IPv6 enables communication between any individual smart things with sufficient address space to cover all the smart things in the world.

In this literature, STIS, STID and IPv6 are potential key players to enable IoT. Therefore, we have developed three types of interactions as shown in Figure 1. First of all, the interaction Point 1 is a lightweight STIS web interface based on CoAP/JSON. This interaction point provides domain-specific applications including not only resource-constrained but also resourceful devices with IPv6 (e.g., 6LoWPAN device) to access STIS via IPv6 address. The interaction Point 2 is CoAP/JSON/oBIX interface at the bottom, which allows CoAP/JSON/oBIX-enabled devices with STID that can push sensors data to STIS. Finally, IEEE 802.15.4-based active RFID tags with STID (i.e., EPC ID) are introduced and communicate with 6LoWPAN networks so that 6LoWPAN is used as a vehicle to carry active RFID tags’ data to STIS. Therefore, the coverage of RFID networks can be easily extended with the aid of 6LoWPAN mesh networks. Regarding the first and second interaction points, D6.3 [1] addresses our innovative interaction. Thus, this report describes design and implemetation of the interaction Point 3 in detail.

2.2. **Smart Thing Information Service (STIS)**

Before going into the deeper details of our design, it is necessary to understand smart things...
information services (STIS) which is based on the EPC network architecture framework and interfaces. Thus, we provide a brief overview of each component in STIS, below.

- **RFID Readers** make multiple observations of RFID tags while they are in the read zone (i.e., smart self, smart room). RFID tags can be active or passive depending on its power-sources.

- **Low Level Reader Protocol (LLRP)** defines the control and delivery of raw tag reads from Readers to the Filtering & Collection role. Events at this interface say “Reader A saw EPC X at time T.”

- **Filtering & Collection (F&C) middleware** filters and collects raw tag reads, over time intervals delimited by events defined by the Capturing Application (e.g. notifying RFID events only if the number of item is changed). As an extension, we have developed CoAP/JSON/oBIX plug-ins here so that IoT6 devices can push their data to STIS, details of which appears in D6.3 [1].

- **Application Level Events (ALE)** interface defines the control and delivery of filtered and collected tag read data from Filtering & Collection role to the Capturing Application role. Events at this interface say “At Location L (e.g., John’s smart self), between time T1 and T2, the following EPCs (e.g., books) were observed,” where the list of EPCs has no duplicates and has been filtered by a criteria defined by the Capturing Application.

- **Capturing Application** supervises the operation of the lower EPC elements, and provides business context by coordinating with other sources of information involved in executing a particular step of a business process. In the groceries trace framework, it may be straightforward, as in an inventory process where there may be a smart fridge that generates periodic observations about objects that enter or leave the fridge. In this case, the Capturing Application merely configures and routes events from the ALE interface directly to an STIS Repository.

- **Capture Interface** is the interface through which RFID data is delivered to STIS Repositories or STIS Accessing Applications. Events at this interface say, for example, “At location X (e.g., John’s smart self), at time T, the following objects (e.g., book items) were observed as a set of object events.”

- **Query Interface** allows accessing applications to retrieve data stored at the STIS repository via STIS Query Callback and Query Control interfaces. Thus, applications needing smart things’ historical or current data from the STIS repository can use the STIS Query Control interface to subscribe or poll data in the STIS repository and receive the results via the STIS Query Callback interface. We call this type of data dynamic data, in terms of when the data is retrieved. On the other hand, smart things may have their own static profile that can be obtained regardless of their physical activity, which is called static data.

- **Accessing Application** is responsible for carrying out overall application processes, such as retrieving food status in the fridge, and so forth, aided by EPC-related data. These can be applications residing at IoT6 domain (e.g., a control and management system).

- **STIS Repository** is the software that records RFID events generated by one or more STIS Capturing Applications, and makes them available for later query by Accessing Applications.

- **Object Name Service (ONS)** is to provide pointers to authoritative information about an object. Examples of the authoritative information are multiple types of services which may include pointers to STIS, product specific web pages, web services and other data such as XML data about products. Thus, ONS clients (e.g., remote applications) can obtain the product information from ONS.
• **Discovery Service** enables applications to find out dynamic historical data. More importantly, applications also need to know where the smart things are now or where the smart things have been before. For instance, a user who wants to purchase milk may look for nearby grocery stores that sell milk. When the milk is purchased, the user may want to know its distribution history in order to confirm whether or not the milk was delivered under safe conditions. Technically, discovery services allow user applications to find a list of STISs nearby the user’s current location or along the distribution channel in a ubiquitous fashion.

### 2.3. **Active RFID Tags (ARTag) over 6LoWPAN**

According to IEEE TG4f Active RFID System, an ARTag can be defined as follows:

"An Active RFID tag is a device which is typically attached to an asset or person with a unique identification and the ability to produce its own radio signal not derived from an external radio signal. Active RFID tag applications include wireless sensor telemetry, control, and location determination. To generate a radio signal, Active RFID tags must employ some source of power. Traditionally this has been accomplished by integrated batteries, although designs exist for such devices that harvest ambient energy from the surrounding environment."

In this report, an active RFID tag is defined as a device which is capable of being uniquely identified by other entities through wireless medium, i.e., IEEE 802.15.4. In this regard, an ARTag is a device with a STID (i.e., EPC ID) that uses IEEE 802.15.4 as its wireless communication interface. 6LoWPAN is used to extend coverage of ARTags at low-cost and on the fly. Figure 2 depicts an overview of ARTags over 6LoWPAN.

![Figure 2. Active RFID tags over 6LoWPAN](image)

---

2 http://www.ieee802.org/15/pub/TG4f.html
By adopting this architecture, STIS can take advantages as follows: since 6LoWPAN routers or active RFID tags may have their own sensors, STIS is able to simultaneously identify smart things and sense their sensory states, enabling fine-grained simultaneous identification and sensing at an item level. Also, since a 6LoWPAN acts as a wireless mesh network, it enables easy construction of active RFID networks on the fly and very cost effectively. Finally, from the data integration point of view, all the RFID data from active RFID tags are translated into a standard-compatible LLRP protocol. Therefore, transparent interaction between smart things and STIS is possible.

With these advantages, many applications based on active RFID tags over 6LoWPAN could be possible as shown in Figure 3. For instance, one of the 6LoWPAN use case scenarios [2] considers a use case regarding hospital storage rooms in which each blood pack has a 6LoWPAN sensor tag in order to monitor the temperature during delivery. Also, a 6LoWPAN device can be installed in each container of a set of blood packs which contain an individual object (e.g., blood pack) and computing devices (e.g., 6LoWPAN node at each container).

2.4. A Survey of Related Work

The following presents the recent integrating developments: National Taitung University (NTTU) in Taiwan [3] presented a novel RFID/IPv6 (RFIPv6) network that allows RFID information to be obtained through the IPv6-based environment. The proposed RFIPv6 network can decrease queries to the Object Naming Service (ONS) system for an EPCIS address, because the RFIPv6 Gateway manages EPC network and sends a command to the EPCIS. In addition, RFID information including EPC code and RFID data is inserted to the option header portion of IPv6 header. ETRI [4] [5] in Korea proposed to combine IPv6 address with EPC code for end-to-end connectivity. The proposed unified address structure integrates identification code of object and location identification address of IPv6. 128-bit IPv6 address consists of 64-bit network prefix and 64-bit interface ID. EPC information such as the RFID object class and serial number can be mapped into the IPv6 interface ID using direct mapping for 64-bit EPC and hashing algorithm for 96-bit EPC. This mapping mechanism allows users to communicate interactively with the tag having the EPC code via the Internet. The National Technical University of Athens,
Greece [6] employed Virtual Mac Address generator (VMAG) that receives an RFID tag ID of various length and that generates an IP address. The VMAG acquires its IP address from a DHCP server by using the virtual MAC address. This direct mapping between RFID tag ID and IP address allows RFID tags to be transparently accessed by the Internet. Dong-A University in Korea [7], proposed IP-RFID systems. In IP-RFID systems, each RFID tag has one IPv6 address, and there is a special RFID reader called IP-RFID reader. The IP-RFID readers extract the IP addresses of the tags read and use the IP address as a source IP address of data packets in order to directly deliver the data packets to its users.
3. Design of 6LoWPAN-based Active RFID Networks

This section illustrates the design of 6LoWPAN-based active RFID networks in terms of system architecture and functional operations.

3.1. Overview

In a general passive or active RFID network, an RFID reader may have multiple antennas. The RFID reader is typically connected to antennas through wired cables as shown in Figure 4. Each antenna can detect RFID tags in a range of 1-5m. Therefore, if it is necessary to extend coverage of RFID networks, more antennas should be installed and whenever an antenna is deployed, they are always wired through a cable. Alternatively, if 6LoWPAN is used for relaying active RFID tag data, active RFID tags can take advantage of 6LoWPAN’s key features such as the simple installation of router antennas for extending network coverage on the fly and cost effectively. Furthermore, these 6LoWPAN-based active RFID tags are connected to LLRP readers acting as a gateway to STIS. STIS does not need to be modified at all, enabling transparent communication between STIS and 6LoWPAN-based active RFID networks. Importantly, a 6LoWPAN router antenna requires a single radio interface based on IEEE 802.15.4 for both 6LoWPAN router and active RFID tag roles, thus reducing deployment cost. In the next section, the system architecture is described in detail.

3.2. System Architecture

Enabling 6LoWPAN-based active RFID tag requires transparent operations between active RFID tags, 6LoWPAN and STIS. Therefore, as depicted in Figure 6, a 6LoWPAN-based active RFID network consists of three major components: an active RFID tag (ARTag) over IEEE 802.15.4, 6LoWPAN Router Antenna (6LoRA), and 6LoWPAN LLRP Reader (6LoL). An active RFID tag role is implemented right above IEEE 802.15.4. The 6LoWPAN Router Antenna is a 6LoWPAN router with the additional role of an active RFID reader. The 6LoWPAN LLRP Reader is an LLRP adaptor which makes the ARTag’s data fit into an LLRP reader and subsequently communicates with STIS through F&C middleware.
D6.4 Innovative interactions between STIS and IPv6 through IoT6 architecture

Figure 6. 6LoWPAN System Architecture

3.2.1. 6LoWPAN LLRP Reader (6LoL)

A 6LoWPAN LLRP Reader (6LoL) is implemented as an LLRP 1.1 standard compliance reader, which reads virtual tags in 6LoWPAN based network and communicates with higher layer of EPCglobal framework (e.g. F&C middleware) via LLRP messages.

The development of 6LoL is based on the wrapper implementation that conforms to the LLRP 1.1 specification and is named the LLRP Wrapper. The LLRP Wrapper targets the LLRP 1.1 standard compliance virtual reader with adaptation capability for various types of tag readers, from virtual tag generators to legacy non-LLRP RFID readers. It conforms to a majority of the mandatory requirements in the LLRP 1.1 specification, except for some of C1G2 air protocol parameters required to deal with low level details of RFID antennas. The LLRP Wrapper can be deployed in various kinds of embedded systems, and in this case in the 6LoWPAN gateway board because it is implemented using C++ and portable libraries.

6LoL is the LLRP Wrapper with adaptation capability of 6LoWPAN. 6LoL treats the 6LoWPAN gateway as a non-LLRP compliant RFID reader, and wraps its functionality to make it conform to the LLRP standard specification. With the aid of the LLRP Wrapper, the 6LoWPAN gateway can connect to F&C middleware and operate as a part of the EPCglobal framework.

The LLRP Wrapper is composed of the following components:
D6.4 Innovative interactions between STIS and IPv6 through IoT6 architecture

**LLRPImpl**: Implements the operations of LLRP 1.1 specification. This component includes operations for ROSpec and AccessSpec, RO/AO administrators, and other supplementary classes.

**PhysicalReader**: Abstracts the structure of the real reader. This component includes stubs such as antenna, GPI/GPO port, and RFIDTag and imitates the structure of the real reader to which the LLRP Wrapper tries to adapt.

**AbstractReader**: Abstracts the communication between the LLRP Wrapper and the real reader. This component includes the interface for various types of readers.

**CConnectionFnCMgr**: Abstracts the LLRP messaging layer using the LLRP Toolkit library.

---

**Figure 8. LLRP Wrapper Structural Overview**

**Figure 9. Simplified Class Diagram of 6LoL**

Figures 8 and 9 show the architectural overview of 6LoL. When the F&C middleware sends an LLRP message to the LLRP Wrapper, it is processed by the LLRPImpl component and especially, LLRPImpl runs ROSpecs and AccessSpecs, and generates the RO_ACCESS_REPORT based on the structure and data of the PhysicalReader. LLRPImpl does
not have to directly access the wrapped reader because PhysicalReader imitates the actual structure of wrapped reader. This accomplishes decoupling between the wrapped reader and the LLRP adaptation layer of the LLRP Wrapper, so this provides high level abstraction for each other.

### 16.1.1 Class-1 Generation-2 (C1G2) Air Protocol

The Class-1 Generation-2 (C1G2) Air Protocol is specified by the EPCglobal Class-1 Generation-2 UHF RFID Protocol v1.1.0 specification.

The following table cross-references LLRP parameters to C1G2 air protocol specific parameters.

<table>
<thead>
<tr>
<th>LLRP Parameter (Section #)</th>
<th>C1G2 Parameter (Section #)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AirProtocolLLRPCapabilities (10.2.3)</td>
<td>C1G2LLRPCapabilities (16.2.1.1.1)</td>
</tr>
<tr>
<td>UHF_RFModeTable (10.2.4.1)</td>
<td>UHFC1G2RFModeTable (16.2.1.1.2)</td>
</tr>
<tr>
<td>AirProtocolInventoryCommandSettings (13.2.6)</td>
<td>C1G2InventoryCommand (16.2.1.2.1)</td>
</tr>
<tr>
<td>TagSpec (12.2.1.2)</td>
<td>C1G2TagSpec (16.2.1.3.1)</td>
</tr>
<tr>
<td>OpSpec (12.2.1.2)</td>
<td>C1G2OpSpec (16.2.1.3.2)</td>
</tr>
<tr>
<td>AirProtocolOpSpecList (12.2.2)</td>
<td>C1G2OpSpec (16.2.1.3.2)</td>
</tr>
<tr>
<td>AirProtocolSpecificEPCMemorySelector (14.2.1.1)</td>
<td>C1G2EPCMemorySelector (16.2.1.5.1)</td>
</tr>
<tr>
<td>AirProtocolTagData (14.2.3)</td>
<td>C1G2PC, C1G2XPCW1, C1G2XPCW2 and C1G2CRC (16.2.1.5.2, 16.2.1.5.3, 16.2.1.5.4, 16.2.1.5.3)</td>
</tr>
<tr>
<td>AirProtocolSingulationDetails (14.2.6.9)</td>
<td>C1G2SingulationDetails (16.2.1.5.6)</td>
</tr>
<tr>
<td>OpSpecResults (14.2.3)</td>
<td>C1G2OpSpecResult (16.2.1.5.7)</td>
</tr>
</tbody>
</table>

**Figure 10. C1G2 Air Protocol Cross-reference Table [referred LLRP 1.1 spec]**

LLRP is designed to manage the low level details of RFID which was not covered by the Reader Protocol (RP). The biggest difference between LLRP and RP is that LLRP is “RFID air protocol aware” while RP provides higher level abstraction that hides many RFID details [referred LLRP 1.0.1 FAQ]. Figure 10 shows the air protocol specific parameters for the C1G2 protocol, but in the case of 6LoL, some of them are not appropriate for handling a 6LoWPAN-based active tag network. There are two options to solve this issue: 1) emulate the conventional C1G2 air protocol specification, or 2) create new air protocol specification for 6LoWPAN. And here, the former option was chosen for integration.

Emulating the C1G2 air protocol specification is straightforward. For example, most of the C1G2 parameters such as C1G2TagSpec or C1G2OpSpec are already compatible with 6LoL, because ARTag is emulating the C1G2 RFID tag. However, some parameters such as UHFC1G2RFModeTable or C1G2InventoryCommand are not appropriate, because the low level details of the antenna structure and the singulation mechanism of RFID are quite different from those of 6LoWPAN or other non-RFID environment. So, in 6LoL, those parameters are currently omitted or emulated using stub implementations.
3.2.2. 6LoWPAN Router as an Antenna (6LoRA)

A 6LoWPAN Router as an Antenna (6LoRA) follows a dual-stack mechanism to support both 6LoWPAN networking and the 6LoRNet service, which are run on IEEE 802.15.4 PHY/MAC. As shown in Figure 11, the 6LoWPAN Stack is used to establish IPv6 connectivity and the ARTag stack is used to emulate the functionalities of the RFID reader.

The 6LoWPAN stack consists of IP adaptation to support IPv6 over IEEE 802.15.4 (e.g. header compression, fragmentation / reassembly, mesh-under routing, neighbor discovery, etc.), TCP/IP, and CoAP, which is a lightweight application protocol that was designed for the communication between smart objects in resource-constrained environments.

The ARTag stack is comprised of ARTag Registration to proxy the registration of the details of the ARTag information to the 6LoL, ARTag Inventory to collect the list of ARTags which reside in its RF boundary, ARTag RFSurvey to obtain the link status between a 6LoRA and an ARTag, ARTag Access to manipulate the resource of ARTag (for example, to access sensor data, to control ARTag application, or to operate actuators), and Local Tag info which is a data storage to maintain the antenna-level tag information. The functional operations is described in detail in Section 3.3.

3.2.3. Active RFID Tag (ARTag)

An Active RFID Tag (ARTag) is implemented very easily: an ARTag stack resides between IEEE 802.15.4 PHY/MAC and the application layer, which includes three main components which play a role in the ARTag Stack. ARTag Registration is to register its information to the 6LoL (e.g. STID, IEEE 802.15.4 network address, attached sensors/actuators, etc.). ARTag PeriodicNot is a module to periodically broadcast its existence to nearby 6LoRAs. ARTag Access includes all the
functionality to manipulate its own resources.

3.3. **Functional Description**

This section illustrates the design of functional operations of 6LoRNet such as bootstrapping, IPv6 autoconfiguration, tag registration, inventory, RF survey, and access operation.

3.3.1. **STID and IPv6 address management scheme**

We use an EPC ID for the smart thing’s identifier and IPv6. Here, we explain the bootstrapping procedure of 6LoRAs, which includes an auto-configuration to obtain the IPv6 address and a 6LoWPAN node registration. The initial step of bootstrapping is an association with a router. After power-up, an active scan is requested over a set of logical channels by sending beacon requests. The node then enables its RF receiver for at most $a_{\text{BaseSuperframeDuration}} \times (2^n + 1)$ time period, where $0 \leq n \leq 14$, to obtain the information contained in all unique solicited beacons. Using this, the node performs association with one selected router by exchanging association request/response. This response contains a 16-bit short address assigned to the newcomer. After that, the newcomer performs stateless auto-configuration for its IPv6 link-local address from its EUI-64 (or derived IID from its short address). Then, it listens for and solicits Router Advertisements (RAs) from 6LoLs or other 6LoRAs to get the IPv6 prefix. The node may form an optimistic global unique IPv6 address by concatenating the prefix and its IID. The next step is a 6LoWPAN node registration. The node needs to register its information such as short address, a EUI-64, an IPv6 address, etc. with the 6LoL. These operations are part of neighbor discovery, which is an important feature offered by the IPv6 protocol.

3.3.2. **Protocols for address registration and update of IoT devices**

The functional requirements for ARTag are very simple. The main requirement for an ARTag is to transmit its unique ID number (STID), and a secondary requirement is to also provide an easy method to access the sensing/actuating resource of the ARTag.

The ARTag periodically sends a message that includes an ID number uniquely identifying the sending tag. This periodic message is termed a *hello*. For maximum battery life, the ARTag may sleep in a low-power state with its radio off, awaken briefly to send its *hello* frame, and then return to its sleep state. To minimize power consumption, the *hello* frame should be as short as possible. However, STID is not short enough to save power if the ARTag periodically sends it. Thus, our strategy is to send an ID number uniquely identifying the sending tag in the network, which is not global. In 6LoRNet, each ARTag periodically sends a *hello* frame which includes its 16 bits short address, instead of STID. For this, each ARTag registers its global unique ID (STID) to the network in advance, immediately after the association procedure.

The functional requirements of 6LoL and 6LoRA are to emulate the RFID Reader in 6LoWPAN which is multi-hop wireless networks. There are three main requirements: the first is an inventory operation to discover the list of tags which is located in the RF boundary of the reader’s antennas; the second is a RF survey to obtain the RF link status between antennas and the tag; the third is an access operation to collect the sensing data from active tags and actuate services of the tags. This section describes the 6LoRNet protocols to emulate these RFID operations.

Prior to explaining the 6LoRNet protocols, we need to focus on the difference between
conventional RFID devices and 6LoRNs. An RFID device consists of two modules: one reader and multiple antennas which are physically attached to the reader, whereas a 6LoRNet is comprised of one 6LoL and multiple 6LoRAs which are logically attached to the 6LoL over multi-hop wireless links. The limitations of multihop wireless communications are inherent to 6LoRNs, such as RF interferences, congestions, etc. In addition, the 6LoRNet protocol should be designed considering the 6LoWPAN characteristics, such as multi-hop topologies, low power, low bandwidth, and resource limitations.

**Inventory Operation:** An RFID device performs inventory operations to discover the list of tags which reside in the RF boundary of its antennas. For this, the reader module scans the tags using all or some specific local antennas. The naive approach to operate this inventory operation in 6LoRNet is to broadcast inventory requests to the entire network. However, it may suffer from huge network overhead caused by frequent flooding, as shown in Figure 13.

![Figure 13. Inventory operation of conventional RFID device and the naïve approach for 6LoRNet.](image)

To reduce network overhead, the 6LoRNet protocol follows a pre-registration and pre-inventory policy. When an ARTag is powered-on, it performs association with one of the 6LoRAs, as shown in Figure 14. Then, it registers its information to the 6LoL. The registration message contains the tuple (epc, saddr, paddr, sensor, actuator) signifying its STID, 16-bit short address, the antenna address which the ARTag associates with, and sensor/actuator information. On receiving the message, the 6LoL stores this tuple in its local memory. After completing the registration, the ARTag starts to periodically sends hello message to its one-hop neighbor 6LoRAs. When the 6LoRA receives the first hello from an ARTag (newcomer), it tentatively remembers the tag and waits for the next hellos from the tag in order to recognize the hello interval and make sure that the tag is located in its RF boundary. Then, the 6LoRA notifies the 6LoL of the tag sensing information with an inventory request message. As a result, the 6LoL can collect all the necessary information of the tags and antennas which can read the tags, and each 6LoRA knows the list of local tags in its RF boundary.
Here, we should consider the network overhead in 6LoWPAN. If every 6LoRA sends the inventory request message to its 6LoL every time it receives *hello*, it causes huge network overhead. Thus, in our 6LoRNet inventory operation, only two events are reported to the 6LoL: one is new-tag event and the other is delete-tag event. A new-tag event is when a 6LoRA detects a new tag which is not in its local tag list. A delete-tag event is when the 6LoRA can no longer detect the tag which is in its local tag list. To make sure the tag information coherent, delete-tag event is important. Therefore, the 6LoRNet inventory operation uses a detection algorithm, as follows. Before sending an inventory request to 6LoL, each 6LoRA waits to receive a couple of *hello* messages in order to recognize the *hello* interval of the tag. The 6LoRA detects delete-tag event using this information. If three consecutive *hello* messages are not delivered to the 6LoRA, it simply assumes that the tag has moved out of its RF boundary or it is turned off. Then, it sends a delete-tag event to the 6LoL to update the antenna information from the tag information in the 6LoL. By doing so, 6LoL can maintain the latest inventory information in its memory. When a 6LoL receives an inventory request from STIS, it simply responds with the tag information which is stored in its local memory without signaling.

**RF Survey:** RF Survey is the operation to obtain the RF link status between antennas and the tag. The link status (e.g. RSSI (Received Signal Strength Index)) is the value which is dynamically changed due to a tag’s location, RF environment, and so on. Thus, it is not possible to maintain the latest information following a similar method as the inventory operation. To maintain the latest RF survey information in 6LoL constantly, all the 6LoRAs should inform the latest RF survey information to the 6LoL every time they receive a *hello* message from the tags. However, this method significantly increases network overhead which may cause RF interference, congestions, or high power consumption. Alternatively, 6LoL can request an RF survey to its 6LoRAs when it is required. But this would also use expensive broadcasting to the entire network. This RF survey operation in 6LoRNet is described in Figure 15.
D6.4 Innovative interactions between STIS and IPv6 through IoT6 architecture

RF Survey Req.  
RF Survey Res.

RF Survey is done with unicast, not cost-expensive broadcast

Figure 15. RF Survey Operation

For the RF Survey operation in 6LoRNet, we can reuse the tag information stored in 6LoL thanks to our inventory method which maintains the tag information in 6LoL. In this protocol, 6LoRNet RF survey follows on-demand RF survey method. When a 6LoL received RF Survey request for a specific tag or a collection of tags from STIS, it searches the 6LoRA list related to the requested tags in its locally stored tag information. Using this information, 6LoL unicasts RF Survey requests to the searched 6LoRAs and receives the latest link status from them.

**Access Operation**: This operation collects the sensing date from active tags and actuate functional services of the tags. The access operation in 6LoRNet can be achieved very simply. As described in the other operations, the 6LoL already has enough tag information including tag ID (STID and short address), related antennas, sensors and actuators which attached to the tag, etc. For the access operation in 6LoRNet, when a 6LoL receives access request from STIS, it simply sends an access request to the tag in order to access the sensor/actuator resources of the tags. As shown in Figure 16, this access operation uses simple unicast. However, since a ARTag is built on IEEE 802.15.4 and it is an RFID, it can communicate only with one 6LoRA with which the tag is associated. Therefore, access operations in 6LoRNet protocol can be done through one specific 6LoRA with which the tag is associated with when it joined the network.
3.3.3. Sensor information exchange between IoT6 and STIS with 6LoRNet

IoT6 devices including sensors and actuators that are operated in a 6LoWPAN network can be transformed into Active RFID tags as described above. Briefly, each device is assigned an EPC number that is hard-coded into the device’s firmware. The device’s data is then embedded into LLRP packages with the EPC number which represents an RFID event that will be read by a LLRP reader, which is, in this case, the 6LoWPAN border router.

The process is described as follows:

Through the 6LoWPAN Router Antennas, which are 6LoWPAN routers, the devices’ Tag Memory is read and forwarded to the 6LoWPAN LLRP Reader. This sort of reader is in fact a 6LoWPAN border router but with an additional LLRP protocol stack so that it can communicate with the LLRP protocol.

From the 6LoWPAN LLRP Reader, the devices’ sensor data are treated as legacy RFID events and are pushed upward to the F&C middleware over legacy LLRP protocol.

In the F&C middleware, the device’s tag events are captured by the Capturing Application and finally will reach the STIS repository and be stored there.

Since the devices’ events fully conform with the LLRP protocol and RFID standards, the STIS, data can be queried normally in the same manner as conventional RFID events.

By adding a third integration point at the bottom layer, it makes the integration with IoT6 devices smooth with legacy RFID protocol. Since the ARTag fully conforms with the LLRP protocol and RFID standards, it can easily be discovered by the EPC Discovery Service. In addition, domain-specific applications see no difference on the device layer, thus making the transition seamless from IPv6 devices to be trackable by the RFID network.

Figure 16. Access Operation.
4. Test cases

Several test cases were designed and performed to verify the implementations of this document. The tests are composed of the following test cases:
- Conformance and performance test of LLRP Wrapper
- Performance test of integrated system (without 6LoWPAN)
- Performance test of 6LoWPAN based sensor network
- Overall integration test and confirmation of functionalities

4.1. LLRP Wrapper Conformance Test

To confirm that the LLRP Wrapper conforms to the LLRP 1.1 standard, conformance tests provided by EPCglobal were performed. Some of the tests were omitted because the requirements were not significantly relevant to the integration test scenario.

The test was performed with a LLRP conformance test tool, which is implemented with CxxTest framework. The test tool automatically executes and connects to the LLRP Wrapper for every test case requirement, and performs predefined actions and checks the response according to the LLRP 1.1 conformance test requirements. Some of the tests were confirmed by manually checking the log and LLRP message which was printed by the LLRP Wrapper. The detailed test results are shown in Table 1, as follows:

<table>
<thead>
<tr>
<th>TCR</th>
<th>Testcase Name</th>
<th>P/F</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCR-R1</td>
<td>TCP Connections</td>
<td>PASS</td>
<td></td>
</tr>
<tr>
<td>TCR-R2</td>
<td>Protocol Version Management</td>
<td>N/A</td>
<td>Omitted. LLRP Wrapper only accepts LLRP 1.1 messages from F&amp;C middleware, so there is no protocol version management currently.</td>
</tr>
<tr>
<td>TCR-R3</td>
<td>Get Reader Capabilities</td>
<td>PASS</td>
<td>This case was checked manually, by reading the log of the following message. GET_READER_CAPABILITY_RESPONSE</td>
</tr>
<tr>
<td>TCR-R4</td>
<td>Custom Messages and Custom Parameters</td>
<td>N/A</td>
<td>Omitted: There is no vendor extension for current scenario.</td>
</tr>
<tr>
<td>TCR-R5</td>
<td>Errors</td>
<td>PASS</td>
<td>Omitted some test steps, step 3~9, and 11. LTKCPP automatically ignores unsupported messages and some kinds of erroneous messages.</td>
</tr>
<tr>
<td>TCR-R6</td>
<td>Read Operations and Reporting</td>
<td>PASS</td>
<td></td>
</tr>
<tr>
<td>TCR-R7</td>
<td>Read Operations in Loop</td>
<td>PASS</td>
<td>The test tool does not programmatically compare the EPCID and memory bank contents. So this case was checked manually, by reading the log of following message. RO_ACCESSREPORT</td>
</tr>
<tr>
<td>TCR-R8</td>
<td>Access Operations and Reporting</td>
<td>PASS</td>
<td></td>
</tr>
<tr>
<td>TCR-R9</td>
<td>Tag Observations, Count-based</td>
<td>PASS</td>
<td></td>
</tr>
</tbody>
</table>
D6.4 Innovative interactions between STIS and IPv6 through IoT6 architecture

<table>
<thead>
<tr>
<th>Triggering</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>TCR-R10   Immediate Triggering</td>
<td>PASS</td>
</tr>
<tr>
<td>TCR-R11   AISpec Stop Trigger</td>
<td>N/A</td>
</tr>
<tr>
<td>TCR-R12   Omitted in LLRP 1.1</td>
<td>N/A</td>
</tr>
<tr>
<td>TCR-R13   Polled Reporting</td>
<td>N/A</td>
</tr>
<tr>
<td>TCR-R14   Keepalives</td>
<td>PASS</td>
</tr>
<tr>
<td>TCR-R15   Lock and Kill Access</td>
<td>N/A</td>
</tr>
</tbody>
</table>

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1. LLRP 1.1 Conformance Test Result

As shown in the table, the LLRP Wrapper validated most of mandatory test cases which are required for test case scenarios. For example, TCR-R1 (Test Connections) verified that the LLRP Wrapper supports both reader-initiated and F&C middleware-initiated connection sequence. TCR-R6 (Read Operations in Loop) verified that basic ROSpec successfully generated corresponding RO_ACCESS_REPORT. Also, TCR-R14 (Keepalives) verified the message exchange sequence of KEEPALIVE and KEEPALIVE_ACK between the LLRP Wrapper and F&C middleware.

4.2. LLRP Wrapper Delay (without 6LoWPAN)

When a reader operation is registered and started, the LLRP Wrapper collects tag data from the reader and generates a report to them. The more tags in the FoV of antennas, the longer time is required for this procedure. In this test case, we substituted reader to emulator and measured pure tag processing performance of 6LoL.

4.2.1. Environment Settings

- LLRP: LLRP Wrapper emulator on Raspberry PI model B, BCM2835 700Mhz ARM11, 512 MB RAM
- F&C: LLRP Commander provided by Fosstrak, on i7 860 2.80GHz, 8 GB RAM, Ubuntu 12.04 64-bits

4.2.2. Test Scenario

To measure pure tag processing performance of 6LoL, this test is performed in the 6LoL emulator mode. In emulator mode, 6LoL doesn’t communicate with 6LoWPAN gateway, but just creates the reader-antenna-tag structure on the PhysicalReader component using given test
In this test, we measured overall time between the arrival of START_ROSPEC message and the send of RO_ACCESS_REPO RT, for various configurations of antenna counts and number of virtual tags. Number of antennas is as follows, [1, 5, 10, 20, 50, 100]. Number of virtual tags per antenna is as follows, [1, 10, 20, 50, 100, 200, 500]. Test was performed 10 times per each configuration, and the average was recorded.

### 4.2.3. Test Results

<table>
<thead>
<tr>
<th>100 (tags)</th>
<th>1</th>
<th>5</th>
<th>10</th>
<th>20</th>
<th>50</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (antennas)</td>
<td>78.40 (ms)</td>
<td>133.00</td>
<td>78.40</td>
<td>20.00</td>
<td>105.80</td>
<td>77.20</td>
</tr>
<tr>
<td>200</td>
<td>150.40</td>
<td>266.00</td>
<td>146.60</td>
<td>168.20</td>
<td>131.60</td>
<td>154.40</td>
</tr>
<tr>
<td>250</td>
<td>188.00</td>
<td>195.00</td>
<td>183.25</td>
<td>164.50</td>
<td></td>
<td></td>
</tr>
<tr>
<td>400</td>
<td>291.84</td>
<td>256.80</td>
<td>260.80</td>
<td>265.80</td>
<td>263.20</td>
<td>270.40</td>
</tr>
<tr>
<td>500</td>
<td>364.80</td>
<td>321.00</td>
<td>326.00</td>
<td>293.90</td>
<td>329.00</td>
<td>338.00</td>
</tr>
<tr>
<td>1000</td>
<td>609.55</td>
<td>742.00</td>
<td>781.60</td>
<td>587.80</td>
<td>714.00</td>
<td>676.00</td>
</tr>
<tr>
<td>2000</td>
<td>1219.10</td>
<td>1306.08</td>
<td>1569.60</td>
<td>1390.40</td>
<td>1284.32</td>
<td>1239.80</td>
</tr>
<tr>
<td>2500</td>
<td>1523.88</td>
<td>1632.60</td>
<td>1502.50</td>
<td>1537.00</td>
<td>1605.40</td>
<td>1492.13</td>
</tr>
<tr>
<td>4000</td>
<td>2438.20</td>
<td>2476.20</td>
<td>2404.00</td>
<td>2459.20</td>
<td>2295.60</td>
<td>2387.40</td>
</tr>
<tr>
<td>5000</td>
<td>3273.20</td>
<td>3061.20</td>
<td>3213.00</td>
<td>3154.00</td>
<td>3210.20</td>
<td>2920.20</td>
</tr>
</tbody>
</table>

Table 2. Elapsed tag processing time of LLRP Wrapper, for the number of tags and antennas

![Graph of tag processing time](image_url)

**Figure 17. Gragh of tag processing time**

The test result in Table 2 was measured by setting the emulator reader of LLRP Wrapper. The row of the table shows each (number of antennas) * (number of tags per antenna). (20 antenna/250 tags) and (100 antenna/250 tags) cells remained vacant because 250 cannot be divided with 20 and 100.

\[
[\text{Elapsed Time (ms)}] = [\text{TagCount}] \times 0.612 + 21.13
\]

Figure 17 shows the graph of the contents of Table 2 and the equation above is linear regression.
graph based on the table. There were no relations between elapsed time and number of antennas, but only the number of overall tags. The performance of LLRP Wrapper was proportional to the number of tags the wrapper should process. The gradient of the graph (in this case, 0.612) depends on the processing power of the test machine.

4.3. **Integrated System Delay (without 6LoWPAN)**

Similar to the LLRP Wrapper Delay, it takes time to create useful information from raw sensor data using EPCglobal architecture.

4.3.1. **Environment Settings**

- LLRP: LLRP Wrapper emulator on i7 860 2.80GHz, 8 GB RAM, Ubuntu 12.04 64-bit, Cloudstack VM (4x2.0GHz CPU, 8GB RAM), Ubuntu Linux 10.04 64-bit, i5 3570K 3.40GHz, 8GB RAM, Ubuntu Linux 12.10.
- F&C: Tomcat 7.0.42, Oracle 64-bit JVM, Ubuntu 12.04, i7-2600, 8GB RAM
- Capturing Application: Intel core i7 2.8 GHz, 8 GHz RAM, Windows 8.1 64-bit
- STIS (EPCIS): Intel core i5 3.0 GHz, 8 GHz RAM, Ubuntu 12.04 64-bit

4.3.2. **Test Scenario**

![Figure 18. Integrated System Delay Test Scenario](image)

This test shows the overall system delay of current implementation except for the 6LoWPAN based sensor network. We used a virtual tag generator which is included in LLRP Wrapper due to the difficulty of building a test environment containing hundreds of ARTags. Figure 18 shows...
the scenario of this test. Capturing Application (component of EPCIS) sets up the LLRP reader operation command and subscribes itself while the F&C middleware sets the ROspec to LLRP Wrapper. Then the LLRP Wrapper generates RO_ACCESS_REPORT from its wrapping reader and sends it to F&C middleware periodically. The report is filtered based on the LLRP reader operation setup, and sent to the capturing application in the format of ECReport. Then the capture application generates a capture report to EPCIS and EPCIS stores it to database.

For this test, we made a simple usage model shown in Figure 19. We assumed that the start trigger of ROspec is invoked periodically with a random period from 37 seconds to 5 minutes. We selected 37 seconds at minimum because sensing with correction takes 37 seconds. We also assumed that ROspec starts at least every 5 minutes. Whenever it starts, it reports 200 tags to F&C middleware. The time between reader timestamp and end of capture operation (T) was measured. We will analyze this result, which shows the overall elapsed time between the send of data from reader and the arrival of data to EPCIS.

![Figure 19. Usage Model for the Integrated System Delay Test Scenario](image)

4.3.3. Test Results

The test results summary is shown in Table 3. There were some abnormally high delay during the test, so they were removed using Quartile or Fourth-Spread Method. Those delays are expected as the garbage collection or network problem. As shown in Figure 20, the delay increases linearly depending on the number of readers.

<table>
<thead>
<tr>
<th>Reader Count</th>
<th>100</th>
<th>500</th>
<th>1000</th>
<th>2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum</td>
<td>31721 (ms)</td>
<td>32317</td>
<td>8558</td>
<td>3374</td>
</tr>
<tr>
<td>Quartile(75)</td>
<td>943</td>
<td>965.25</td>
<td>1095</td>
<td>1413</td>
</tr>
<tr>
<td>Median</td>
<td>614</td>
<td>736</td>
<td>995</td>
<td>1206</td>
</tr>
<tr>
<td>Quartile(25)</td>
<td>579</td>
<td>622.75</td>
<td>865</td>
<td>1062</td>
</tr>
<tr>
<td>Minimum</td>
<td>334</td>
<td>351</td>
<td>609</td>
<td>725</td>
</tr>
<tr>
<td>Fourth Spread</td>
<td>364</td>
<td>342.5</td>
<td>230</td>
<td>351</td>
</tr>
<tr>
<td>Upper Outlier Boundary</td>
<td>2028.5</td>
<td>2183.9</td>
<td>2637.5</td>
<td>3325.5</td>
</tr>
<tr>
<td>Lower Outlier Boundary</td>
<td>-254.5</td>
<td>-198.1</td>
<td>-302.5</td>
<td>-387.0</td>
</tr>
<tr>
<td>Outlier %</td>
<td>1.96%</td>
<td>2.94%</td>
<td>1.89%</td>
<td>0.06%</td>
</tr>
<tr>
<td>Average (w/o outliers)</td>
<td>678.4</td>
<td>773.6</td>
<td>1017.1</td>
<td>1319.9</td>
</tr>
<tr>
<td>STDDEV (w/o outliers)</td>
<td>218.0</td>
<td>230.1</td>
<td>266.8</td>
<td>438.7</td>
</tr>
</tbody>
</table>

Table 3. Test Result Summary of Integrated System Delay Test (T)

To perform the Quartile or Fourth-Spread Method, we calculated quartiles and fourth spread of test results. Then the upper and lower outlier boundary was calculated, and the corresponding
outliers were removed from the results. The average and standard deviation of this result set is at the last rows of the Table 3.

![Figure 20. Test Results Graph of Average Integrated System Delay (T)](image)

4.4. Active Tag Network Construction Delay

In this test, we measured the delay of the active tag network construction, by measuring the time between the start of 6LoWPAN gateway initialization and the end of the first EPC ID registration from 6LoRA to 6LoL. Technically, this is not the tag network construction delay but the overall initialization delay of 6LoRA and 6LoWPAN gateway. But we named this period as active tag network construction delay because this is the delay for active tags to become available, which the 6LoL should wait for.

4.4.1. Environment Settings

- LLRP: LLRP Wrapper emulator on Raspberry PI model B, BCM2835 700Mhz ARM11, 512 MB RAM
- F&C: LLRP Commander provided by Fosstrak, on i7 860 2.80GHz, 8 GB RAM, Ubuntu 12.04 64-bit

4.4.2. Test Scenario

The test environment is described in Figure 21, 6LoRA1 is wired to 6LoL, 6LoRA2 is wirelessly connected to 6LoRA1, and ARTags are connected to each of 6LoRA. We performed 10 tests for each configuration and obtained the average of the results.
D6.4 Innovative interactions between STIS and IPv6 through IoT6 architecture

![Diagram of network setup](image)

**Figure 21. Active Tag Network Construction Delay Test Environment**

### 4.4.3. Test Results

The test results are shown in Figure 22. The construction delay of the network was not so relevant with the number of ARTags in this test. This implies that the time consumed by each ARTags is not so significant if there are only a few tags, but initialization time results in the majority of this delay.

![Graph of test results](image)

**Figure 22. Test Result Graph of Active Tag Network Construction Delay**

### 4.5. RFSurvey Delay

In this test, we’ve measured the delay of RFSurvey (refer to 3.3.2) operation for ARTag. This operation is used to measure the RSSI of target ARTag with the matching EPC.

#### 4.5.1. Environment Settings

- 6LoL: Raspberry Pi model B, BCM2835 700Mhz ARM11, 512 MB RAM
- 6LoRA: MSP430F5438, 16-Bit Ultra-Low-Power Microcontroller, 18 Mhz, 256KB Flash, 16KB RAM, 12-bit ADC, 4 USCI, 32-bit HW Multi
- ARTag: MSP430F5438, 16-Bit Ultra-Low-Power Microcontroller, 18 Mhz, 256KB Flash, 16KB RAM, 12-bit ADC, 4 USCI, 32-bit HW Multi
4.5.2. Test Scenario

The test environment is the same as the one in Figure 21, which was used in Active Tag Network Construction Delay test. But in this test, only single ARTag is connected to 6LoRAs. In this structure, we’ve measured the delay of RFSurvey for a single ARTag with given EPCID.

4.5.3. Test Results

![Image](image-url)

*Figure 23. RFSurvey Delay Test Result Sample*

The test result sample is shown in Figure 23, and it shows the single test result and updated RSSI value for each antenna. By performing 50 tests, we’ve obtained the average delay of 234.46ms and the standard derivation of 2.03. But because the RFSurvey is done for each 6LoRA, the result can be varied for the number of 6LoRAs and the structure of network, for example, number of 6LoRAs, number of hops, and distance to reach the target 6LoRA or ARTag.

4.6. Test Case Scenarios

In this test we verify our 6LoWPAN ARTag’s operation with 6LoWPAN-based LLRP reader and upper layer components such as F&C and STIS. The goal of this test is to verify if 6LoTag information is discoverable by STIS (EPCIS).

The below snapshot shows the log of our LLRP Wrapper, which is the 6LoWPAN border router:
D6.4 Innovative interactions between STIS and IPv6 through IoT6 architecture

![Figure 24: 6LoWPAN Border Router LLRP Wrapper](image)

And our F&C Server:

![Figure 25: F&C Server Logs](image)
The 6LoTag information is captured by Capturing App:

Handling incoming reports

Figure 26: Capturing Apps catching the events

And is stored finally in our STIS. The following snapshot shows the STIS livestream of events:
D6.4 Innovative interactions between STIS and IPv6 through IoT6 architecture

Figure 27: Live stream of events in STIS web interface
5. Conclusion

The objective of this deliverable (D6.4) is to report a design specification describing innovative interaction between Smart Things Information System (STIS) using Smart Things ID (STID) and IPv6 and also tested cases of interaction between STID and IPv6 through IoT6 architecture.

In WP6, the new types of interactions have been proposed in deliverable 6.3 (D6.3), the interaction point 1 is a lightweight STIS web interface based on CoAP/JSON. This interaction point provides domain-specific applications including not only resource-constrained but also resourceful devices to access STIS with IPv6 address after obtaining IPv6 address of STIS via discovery service or ONS. The second interaction point is CoAP/JSON/oBIX interface at the bottom, which allows CoAP/JSON/oBIX-enabled devices with STID can push sensory data to STIS. Lastly, IEEE 802.15.4-based active RFID tags with STID (i.e., EPC ID) are introduced and communicate with 6LoWPAN networks so that 6LoWPAN is used as a vehicle to carry active RFID tags’ data to STIS.

Since the first two interactions types are described in D6.3 this deliverable is devoted to reporting innovative interaction between STIS using STID and IPv6 by employing 6LoWPAN as a means to convey smart thing’s data like STID and sensory data and by adapting a 6LoWPAN gateway and routers to LLRP reader and RFID antennas. In this way, smart things with STID can benefit from IPv6 and interact with STIS. Through implementation and test cases, this report validated the proposed approach and also showed its feasibility.
6. References


### 7. List of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACR</td>
<td>Acronym</td>
</tr>
<tr>
<td>6LoWPAN</td>
<td>IPv6 over Low power Wireless Personal Area Networks</td>
</tr>
<tr>
<td>AA</td>
<td>Accessing Applications</td>
</tr>
<tr>
<td>ALE</td>
<td>Application Level Event</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>CMS</td>
<td>Control Monitoring System</td>
</tr>
<tr>
<td>CoAP</td>
<td>Constrained Application Protocol</td>
</tr>
<tr>
<td>DS</td>
<td>Discovery Service</td>
</tr>
<tr>
<td>EPC</td>
<td>Electronic Product Code</td>
</tr>
<tr>
<td>F&amp;C</td>
<td>Filtering and Collection</td>
</tr>
<tr>
<td>HTTP</td>
<td>Hyper Text Transfer Protocol</td>
</tr>
<tr>
<td>IoT</td>
<td>Internet of Things</td>
</tr>
<tr>
<td>IPv6</td>
<td>Internet Protocol Version 6</td>
</tr>
<tr>
<td>JSON</td>
<td>JavaScript Object Notation</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>LLRP</td>
<td>Low-level Reader Protocol</td>
</tr>
<tr>
<td>ONS</td>
<td>Object Name Service</td>
</tr>
<tr>
<td>RFID</td>
<td>Radio-frequency identification</td>
</tr>
<tr>
<td>STID</td>
<td>Start Things Identifier</td>
</tr>
<tr>
<td>STIS</td>
<td>Start Things Information Service</td>
</tr>
<tr>
<td>WP</td>
<td>Work Package</td>
</tr>
<tr>
<td>WSDL</td>
<td>Web Service Description Language</td>
</tr>
</tbody>
</table>