



Universidad Austral de Chile

Facultad de Ciencias de la Ingeniería
Escuela de Ingeniería Civil en Informática

USO DE 6LOWPAN PARA EL MONITOREO DE LA SALUD ESTRUCTURAL DE UN PUENTE

Proyecto para optar al título de
Ingeniero Civil en Informática

PROFESOR PATROCINANTE:
CHRISTIAN ALEXIS LAZO RAMÍREZ
DOCTOR EN INGENIERÍA TELEMÁTICA

PROFESOR CO-PATROCINANTE:
FRANK SCHANACK
DOCTOR INGENIERO DE CAMINOS, CANALES Y PUERTOS

PROFESOR INFORMANTE:
LUIS HERNÁN VIDAL VIDAL
M.B.A. INGENIERO CIVIL EN INFORMÁTICA

PAULO ANTONIO GALLARDO CASANOVA

VALDIVIA - CHILE
2013

Valdivia, 20 Noviembre del 2013

De: Christian Lazo Ramírez
Prof. Patrocinante

A: Mauricio Ruiz-Tagle M.
Director
Escuela de Ingeniería Civil en Informática

Ref: Calificación proyecto de título

De mi consideración:

Habiendo revisado el trabajo de titulación “Uso de 6LOWPAN para el monitoreo de la salud estructural de un puente”, presentado por el alumno Sr. **Paulo Antonio Gallardo Casanova**, mi evaluación del mismo es la siguiente:

Nota: 7,0 Siete, coma cero

Fundamento de la nota:

El trabajo de título realizado por el alumno cumple con todos los desafíos y objetivos planteados originalmente.

Aspecto	Evaluación
Cumplimiento de objetivos	7,0
Satisfacción de alguna necesidad	7,0
Aplicación de metodologías pertinentes	7,0
Interpretación de los datos y obtención de conclusiones	7,0
Originalidad	7,0
Aplicación de criterios de análisis y diseño	7,0
Perspectivas del trabajo	7,0
Coherencia y rigurosidad lógica	7,0
Precisión del lenguaje técnico	7,0

Sin otro particular, saluda atentamente a usted,



Dr. Ing. Christian Lazo Ramírez
Académico
Universidad Austral de Chile



Universidad Austral de Chile
Escuela Ingeniería Civil en Obras Civiles



Valdivia, noviembre 22 de 2013

DE : DR. ING. FRANK SCHANACK
A : ESCUELA DE ING. CIVIL INFORMATICA
SR. MAURICIO RUIZ TAGLE M.

MOTIVO:

Informar a usted, calificación del trabajo de tesis "Uso de GLOWPAN para el monitoreo de la salud estructural de un puente", del estudiante señor Paulo Antonio Gallardo Casanova.

Item	Nota
Cumplimiento del objetivo propuesto	7
Satisfacción de alguna necesidad	7
Aplicación del método científico	7
Interpretación de los datos y obtención de conclusiones	7
Originalidad	7
Aplicación de criterios de análisis y diseño	7
Perspectivas del trabajo	7
Coherencia y rigurosidad lógica	7
Precisión del lenguaje técnico en la exposición, composición, redacción e ilustración	7
NOTA FINAL	7

Saluda atentamente a usted,

DR. ING. FRANK SCHANACK
Profesor Co Patrocinante

FS/jasp
c.c.: archivo



Universidad Austral de Chile

Instituto de Informática

Valdivia, 20 de noviembre de 2013.

De : Luis Hernán Vidal Vidal.

A : Mauricio Ruiz-Tagle

Director de Escuela de Ingeniería Civil en Informática.

Ref. : Informa Calificación Trabajo de Titulación.

MOTIVO: Informar revisión y calificación del Proyecto de Título “**USO DE 6LOWPAN PARA EL MONITOREO DE LA SALUD ESTRUCTURAL DE UN PUENTE**”, presentado por el alumno Paulo Antonio Gallardo Casanova, que refleja lo siguiente:

Se logró el objetivo planteado que permitió: “Usar 6LoWPAN para monitorear la salud estructural de un puente a través de un sensor con una dirección IPv6 según la tendencia del Internet de las Cosas”.

Cumplimiento del objetivo propuesto.	7,0
Satisfacción de alguna necesidad.	7,0
Aplicación del método científico.	7,0
Interpretación de los datos y obtención de conclusiones.	7,0
Originalidad.	7,0
Aplicación de criterios de análisis y diseño.	7,0
Perspectivas del trabajo.	7,0
Coherencia y rigurosidad lógica.	7,0
Precisión del lenguaje técnico en la exposición, composición, redacción e ilustración.	7,0
Evaluación Tesis.	7,0

Por todo lo anterior expuesto califico el trabajo de titulación del alumno Paulo Antonio Gallardo Casanova con nota 7,0 (siete coma cero).

Sin otro particular, se despide atentamente.

M.B.A. ING. Luis Hernán Vidal Vidal.
Académico Instituto de Informática.
Facultad de Ciencias de la Ingeniería.
Universidad Austral de Chile.

AGRADECIMIENTOS

A mi madre, quien me ha brindado incondicionalmente su apoyo y cariño en cada etapa de mi vida.

A los profesores que me formaron, por contribuir con mi desarrollo tanto profesional como personal.

A los compañeros de clase y a todas las personas que me desearon bien y me expresaron palabras de sincero apoyo.

A todos ustedes, muchas gracias.

ÍNDICE

ÍNDICE	I
ÍNDICE DE TABLAS	III
ÍNDICE DE FIGURAS	IV
RESUMEN	V
ABSTRACT	VI
1 INTRODUCCIÓN	1
1.1 Antecedentes	2
1.2 Motivación	4
1.3 Impactos	5
1.4 Objetivos	5
1.4.1 Objetivo general	5
1.4.2 Objetivos específicos	5
1.5 Organización del documento	6
2 MARCO TEÓRICO	8
2.1 El Internet de las Cosas	8
2.2 El estándar IEEE 802.15.4	13
2.3 6LoWPAN	15
2.4 Sistemas embebidos	20
2.5 Wireless sensor networks	22
2.6 Aplicaciones existentes	24
2.6.1 Implementaciones del stack 6LoWPAN	25
2.6.2 Sistemas basados en 6LoWPAN	28
3 SOLUCIÓN PROPUESTA	33
3.1 Definición de la problemática	33
3.1.1 Sistema de Gestión de Puentes	33
3.1.2 Monitoreo continuo	36
3.2 Metodología de trabajo	37
3.3 Aplicación del método científico	38
3.3.1 Observación	39
3.3.2 Formulación de hipótesis	40
3.3.3 Experimentación	40
3.3.4 Emisión de conclusiones	44
3.4 Definición de la solución	44
3.4.1 Arquitectura de la solución	45
3.4.2 Tecnologías involucradas	45
4 ANÁLISIS Y DISEÑO	47
4.1 Metodología de software	47
4.2 Requisitos	49
4.3 Fase de Análisis	52
4.3.1 Descripción de las actividades del proceso	52
4.3.2 Actores del sistema	53
4.3.3 Casos de uso de alto nivel	54
4.3.4 Casos de uso expandidos	55
4.3.5 Diagramas de secuencia	56
4.4 Fase de Diseño	57
4.4.1 Casos de uso real	57
4.4.2 Diagramas de colaboración	59
4.4.3 Base de datos	60
5 IMPLEMENTACIÓN	62
5.1 Configuración de módulos XBee	62
5.2 Arduino IPv6 stack	64
5.2.1 Configuración del nodo	65
5.2.2 Configuración del router	68
5.2.3 Direccionamiento IPv6	69
5.2.4 Descubrimiento de vecinos	70
5.3 Configuración de la sonda	70
5.4 Configuración de Xively	72

5.5	Vista del estado del puente	73
6	RESULTADOS	75
6.1	Solicitud de router	75
6.2	Mediciones con acelerómetro	76
7	CONCLUSIONES	80
	BIBLIOGRAFÍA	82
	ANEXOS	86
	Anexo A: Artículo INFONOR	86
	Anexo B: Sketch del nodo	94
	Anexo C: Paquetes capturados	101
	Anexo D: JSON en Xively	102
	Anexo E: Lista de abreviaciones	104

ÍNDICE DE TABLAS

TABLA	PÁGINA
1 Clasificación de tecnologías de comunicación inalámbricas	13
2 Comparación de LR-WPAN con otras tecnologías inalámbricas	14
3 Los 2 bits más significativos del dispatch byte	19
4 Elementos principales de una WSN	23
5 Comparación entre TinyOS y Contiki	27
6 Alternativas para solución	41
7 Product Backlog	49
8 Sprint 1	50
9 Sprint 2	51
10 Sprint 3	51
11 Sprint 4	51
12 Sprint 5	52
13 Requisitos funcionales del sistema	52
14 Caso de uso de alto nivel “Capturar data”	54
15 Caso de uso de alto nivel “Ver estado de puente”	54
16 Caso de uso expandido “Capturar data”	55
17 Caso de uso expandido “Ver estado de puente”	56
18 Caso de uso real “Ver estado de puente”	58
19 Diccionario de datos de la tabla Sensor	61
20 Diccionario de datos de la tabla Muestreo	61
21 Características módulo XBee 1mW PCB Antenna	63
22 Direcciones IPv6 en notación hexadecimal	69

ÍNDICE DE FIGURAS

FIGURA	PÁGINA	
1	Visión del Internet de las Cosas	9
2	Agotamiento de direcciones IPv4	10
3	IoT para la creación de entornos inteligentes	11
4	Espacio de operación de WPAN y WLAN	14
5	Stack IP y Stack 6LoWPAN	15
6	Arquitectura de 6LoWPAN	17
7	Frame IEEE 802.15.4 con direccionamiento Full UDP/IPv6 (64-bit)	18
8	Frame IEEE 802.15.4 con direccionamiento mínimo UDP/6LoWPAN (16-bit)	18
9	Cabecera 6LoWPAN/UDP comprimida (6 Bytes)	18
10	Cabecera estándar IPv6/UDP	19
11	Dispositivo embebido	21
12	Estructura de una WSN	23
13	Pila de protocolos 6LoWPAN	25
14	La arquitectura de Contiki	26
15	La arquitectura de Idesco Cardea	30
16	Infraestructura de red inteligente de Elster Group	31
17	Diagrama de JenNet-IP	31
18	Módulos básicos de un Sistema de Gestión de Puentes	34
19	Login Sistema de Gestión de Puentes UACH	35
20	Módulos de Sistema de Gestión de Puentes de UACH	36
21	Modelo simplificado para el método científico	38
22	Simulador Cooja de Contiki	41
23	Consola OpenWRT	42
24	Router TP-LINK TL-MR3220	42
25	Placas Arduino	43
26	Arquitectura de la solución	45
27	Diagrama del ciclo iterativo Scrum	48
28	Diagrama de procesos	53
29	Diagrama de casos de uso	54
30	Diagrama de secuencia asociado a “Capturar data”	57
31	Diagrama de secuencia asociado a “Ver estado de puente”	57
32	Diagrama de colaboración del Nodo	59
33	Diagrama de colaboración del Router	59
34	Diagrama de colaboración de la Sonda	60
35	Subconjunto del modelo de datos del Sistema de Gestión de Puentes	60
36	Módulo XBee	62
37	Conexiones mínimas para módulo XBee	63
38	Configuración XBee con X-CTU	64
39	Frame UART	64
40	Arduino-IPv6 Stack de Telecom Bretagne	65
41	Nodo con sensor acelerómetro	66
42	Diagrama de flujo del nodo	67
43	Diagrama de bloques del acelerómetro	67
44	Diagrama de flujo del border router	68
45	Dirección IPv6 de enlace local	69
46	Dirección IPv6 unicast global	69
47	Proceso básico de descubrimiento de vecinos	70
48	Escenario de prueba con sonda	71
49	Diagrama de flujo de la sonda	71
50	Base de datos Xively	72
51	Ficha del puente en el Sistema de Gestión de Puentes	73
52	Vista para sensores en Sistema de Gestión de Puentes	74
53	Tiempo desde solicitud de router hasta envío de data	75
54	Muestreos de regla de medir	77
55	Muestreos del puente Calle Calle	78

RESUMEN

En la actualidad son cada vez más los dispositivos tecnológicos que se incorporan en el diario vivir, los cuales otorgan un inusitado grado de conectividad como, por ejemplo, *tablets*, *smartphones*, o incluso objetos de uso frecuente como un refrigerador. En este contexto, la humanidad comienza a convivir con una red de aparatos cotidianos interconectados entre sí, la cual se conoce como el “Internet de las Cosas”, el que sin lugar a dudas se convertirá en la próxima revolución tecnológica, generando nuevas oportunidades y desafíos para la comunidad, usuarios de tecnología, empresas y sociedad. En el actual escenario, lo que se persigue en el presente trabajo es comunicar los dispositivos a través de 6LoWPAN, un estándar basado en el protocolo IPv6 sobre redes de bajo consumo y pérdida para hacer realidad el Internet de las Cosas, utilizando para ello plataformas de hardware libre y dando énfasis a las particularidades de los sistemas embebidos, tales como un limitado número de operaciones, bajo consumo energético y baja capacidad del ancho de banda. Además, se usa 6LoWPAN en un entorno de red que hace posible el Internet de las Cosas para un Sistema de Gestión de Puentes a modo de optimizar las operaciones de mantenimiento de las estructuras.

ABSTRACT

Today, the technological devices are increasing and they are being incorporated in daily life. The technological devices give to us an unusual degree of connectivity, for example: tablets, smartphones, or even everyday things, such as a refrigerator. In this context, the humanity begins to live with a network of interconnected devices that is known as the “Internet of Things”, which will be the next technological revolution, bringing new opportunities and challenges for the users, business and society. In this scenario, the objective of this project is to communicate devices through 6LoWPAN, a standard based on the IPv6 protocol over low-power and lossy networks to support the Internet of Things, using open hardware platforms and emphasizing the singularities of embedded systems, such as a limited number of operations, low power consumption and low bandwidth capability. In addition, the prototype uses the standard 6LoWPAN in a network environment that brings the Internet of Things to a Bridge Management System for optimizing maintenance of structures.

1. INTRODUCCIÓN

Hoy en día los sistemas embebidos se han masificado enormemente, al punto de convertirse en parte esencial de nuestras vidas. Este hecho se ve reflejado, por ejemplo, en la penetración de teléfonos inteligentes que integran cada vez mayores funcionalidades que requieren conectividad hacia Internet. Sin embargo, esta necesidad de conectividad también está siendo requerida por artefactos de uso cotidiano, tales como electrodomésticos, prendas de vestir, dispositivos vehiculares, entre otros objetos. En este escenario, donde la conectividad de los objetos es el eje principal de desarrollo, surge un nuevo concepto llamado el “Internet de las Cosas” o IoT (*Internet of Things*), el cual describe una red de diferentes objetos de uso cotidiano interconectados entre sí.

Internet ha llegado a un punto de inflexión en su desarrollo, donde una red de computadores interconectados ha evolucionado hacia una red de objetos interconectados entre sí.

Este nuevo escenario de Internet ha sido evaluado, por ejemplo, por la Comisión Europea, la cual ha dispuesto de un plan de acción en torno a las perspectivas y retos que presenta el desarrollo del IoT, ya que la nueva Internet será una realidad dentro de los próximos años que modificará drásticamente la manera en que se organiza la sociedad [Com09].

La forma en que estos objetos se conectarán hacia Internet será a través de sistemas embebidos, los cuales en un principio realizaban un conjunto limitado de operaciones, que hoy día se han visto incrementadas con el paso de los años, y que dadas sus características no requieren un uso intensivo de la red para enviar información.

Por otra parte, el protocolo de red IPv6 (*Internet Protocol version 6*) posee una mayor capacidad de direccionamiento que su predecesor IPv4, pasando desde direcciones de 32 bits hasta los 128 bits, lo que posibilita la interconexión *end-to-end* de novedosos dispositivos inalámbricos de baja potencia con capacidad de procesamiento limitadas, llamados a formar parte de *The Wireless Embedded Internet* [She09].

De esta forma, se pretende en el presente proyecto implementar un escenario de red 6LoWPAN (*IPv6 over Low-Power Wireless Personal Area Networks*), el cual es un

estándar basado en IP sobre redes de bajo consumo y pérdida o LLN (*Low power and Lossy Networks*), para así, hacer realidad la tecnología que sustenta el Internet de las Cosas con el fin de construir un prototipo que permita aplicarla al monitoreo de la salud estructural de un puente bajo el contexto de un “Sistema de Gestión de Puentes”.

1.1. Antecedentes

El Internet de las Cosas (IoT), aunque aún no es una realidad tangible, se proyecta como tal para cambiar la forma en que se interactúa con los objetos. El IoT plantea cambios en los paradigmas de comunicación, donde, los objetos con capacidad autónoma de interacción, cómputo y organización se asocian para cumplir tareas específicas sin requerir la intervención directa del ser humano. Además, el IoT evidencia una visión tecnológica futura donde Internet se extiende dentro del mundo real incluyendo a los objetos de uso cotidiano como nuevos actores del ecosistema de conectividad digital, transformando el entorno hasta incluirlo de forma activa en el ciclo de vida de las personas [Cas11].

Los esfuerzos para convertir en realidad el IoT han sido adoptados también por múltiples empresas con soluciones propietarias como, por ejemplo, ZigBee, Z-Wave o HomePlug, cuyo objetivo va en dirección del desarrollo de comunicaciones seguras con bajas tasas de envío de datos y maximización de la vida útil de las baterías. Además, desde hace un tiempo, investigadores del MIT (*Massachusetts Institute of Technology*) como Neil Gershenfeld exponen acerca del Internet 0 [Del07], cuyas características apuntan hacia una lenta evolución para hacer más fácil su implementación y poder llegar a tener dispositivos de bajo consumo que permitan comunicar objetos del hogar.

De esta forma, se comienza a visualizar una gran cantidad de aplicaciones que puede ofrecer una red interconectada de objetos, provocando mejoras considerables en la calidad de vida de las personas. En este ámbito, Marc Torrent de la empresa multinacional Ficosa, dedicada a la investigación, desarrollo, producción de sistemas y componentes para la industria automotriz plantea que los automóviles pronto serán parte del IoT, lo que podría producir una disminución de 200000 muertes al año en carreteras europeas con la utilización de esta tecnología [Rev09], ya que la principal causa de accidentes es la mala visibilidad, donde la tecnología puede contribuir a superar esta problemática. Siguiendo en la línea, NXP Semiconductors presentó en el año 2011 el producto JenNet-IP [NXP11],

basado en 6LoWPAN, que permite el control de lámparas, ampollas LED, enchufes inteligentes y un monitor a través de una aplicación móvil diseñada para *smartphone* o *tablet*, para lograr mejoras de consumo energético en redes residenciales e industriales.

En este contexto, el *IoT European Research Cluster* (IERC) [Eur] surge como un grupo de investigación que hace frente al potencial de la nueva Internet, poniendo énfasis en los beneficios competitivos que representa, tanto socialmente como con respecto a la economía de la información, teniendo en mente las implicancias en seguridad y privacidad que se desprenden de la aplicación del Internet de las Cosas.

En cuanto a la estandarización de las comunicaciones que sustentan el IoT, en el año 2003 el IEEE (*Institute of Electrical and Electronics Engineers*) normó el 802.15.4, el cual se alza como el primer estándar de radio de bajo consumo, aunque en primera instancia presentó dificultades de escalabilidad, capacidad de evolución e integración con Internet [She09]. Por otra lado, el protocolo IPv6 posibilita nuevos paradigmas para la interconexión de dispositivos inalámbricos de baja potencia con capacidades de procesamiento limitadas, reflejándose este desarrollo en numerosos RFC (*Request for Comment*), como el RFC 2460 [Dee98], RFC 4291 [Hin06] y RFC 4861 [Nar07] referidos a IPv6, el RFC 4919 [Kus07] referido al uso de IPv6 sobre 6LoWPAN considerando la transmisión de paquetes IP sobre el IEEE 802.15.4, y el RFC 4944 [Mon07] que estandarizó una compresión de cabeceras IPv6 y UDP sobre IEEE 802.15.4 en 6 Bytes. Además, para establecer convergencia entre IPv6 y redes WHAN (*Wireless Home Area Network*) el estándar 6LoWPAN define las funcionalidades que debe presentar un *edge router* o router de frontera.

En lo estrictamente referente a la aplicabilidad del IoT para el monitoreo de estructuras como puentes, el desarrollo de un sensor que implementa 6LoWPAN puede entregar datos en línea a un Sistema de Gestión de Puentes. En 1981 un informe de la OECD (*Organisation for Economic Co-operation and Development*) definió a los Sistemas de Gestión de Puentes o BMS (*Bridge Management System*) como una herramienta para ayudar a las agencias de puentes y carreteras en la elección de las mejoras óptimas para la red de puentes, y que son compatibles con las políticas de las organizaciones, tales como objetivos a largo plazo y restricciones de presupuesto [Akg07]. Dentro de los alcances que un BMS posee cabe mencionar la importancia de su base de datos, dado que esta

información genera una destacable base de conocimiento para los futuros diseños, los componentes y los sistemas constructivos, así como también ayuda a planear y controlar de mejor manera las mantenciones de las estructuras. Para lograr este objetivo, los BMS requieren de procedimientos que aseguren que los puentes son inspeccionados y evaluados regularmente, de modo que se lleve a cabo una mantención apropiada y se mantenga una condición adecuada a lo largo de su vida útil.

Con respecto a los Sistemas de Gestión de Puentes en el mundo, PONTIS y BRIDGIT son los dos BMS más utilizados en Norteamérica, mientras que en Alemania los análisis y procedimientos de evaluación se hacen de acuerdo a los resultados de inspecciones según la norma DIN 1076 (*Deutsches Institut für Normung*), para luego optimizar los datos en el nivel de red e integrar a los programas de mantenimiento general; la base de datos existente SIB-Buawerke y las bases de datos de rutas comunes son integradas para proporcionar datos para los módulos posteriores [Mol12]. En Chile existen algunas propuestas acerca de sistemas privados de inventarios de puentes y otras como MAPRA (Mantenimiento de Puentes de la Red Vial Austral), el cual se encuentra aún inconcluso, posee componentes de un sistema de gestión moderno, pero está limitado a su zona geográfica y falta desarrollo en términos de evaluación y priorización de estructuras [Esh].

Por todo esto, el IoT será una realidad que introducirá un nuevo paradigma en la sociedad, por este motivo ha generado el interés de universidades, centros de investigación, empresas y gobiernos, lo que permite vislumbrar un gran desarrollo de aplicaciones que podrán elaborarse a partir de este concepto.

1.2. Motivación

Durante los últimos años se ha experimentado una notable masificación de los sistemas embebidos, como teléfonos celulares, reproductores de música, POS (*Point Of Service*), decodificadores y *set-top boxes* para televisores, cajeros automáticos, routers, entre otros. Los sistemas embebidos realizan un conjunto limitado de operaciones, usan un procesador relativamente pequeño y memoria limitada para reducir costes, enfrentando normalmente problemas en tiempo real. En este aspecto existen soluciones, pero propietarias, para un tema que actualmente es objeto de investigación para universidades y organismos especializados, que por cierto despierta el interés de gobiernos y empresas privadas, por

tanto es necesaria una alternativa abierta y alineada con los estándares definidos por la IETF (*Internet Engineering Task Force*) que potencie la investigación y desarrollo del área, el cual se encuentra directamente relacionado con la calidad de vida de la sociedad, a través de la convergencia de redes de área doméstica y comunicación IP.

1.3. Impactos

Socialmente, la irrupción del Internet de las Cosas implica una revolución, un cambio en la forma en que la sociedad interactúa con el entorno y con los pares. La comunicación con los objetos depende de la convergencia entre redes WHAN (*Wireless Home Area Network*) e IPv6, e implica una reivindicación de los objetos de uso cotidiano, los cuales se relacionarán con cada individuo de una forma hasta ahora insospechada.

Además, la utilización de sensores que implementen esta tecnología permite optimizar y/o automatizar procesos, así como también establecer canales de comunicación a través de la infraestructura de la red entre objetos y personas.

De esta forma se pone en evidencia que la tendencia de la nueva Internet tiene el potencial para transformarse prontamente en una realidad [Com09], por lo tanto, los esfuerzos invertidos en el área contribuyen como aporte científico-tecnológico en dirección de la investigación de nuevas tecnologías, siendo aplicada a ámbitos tales como el monitoreo de estructuras, monitoreo de medio ambiente, ahorro de energía, fábricas más eficientes, mejor logística, cuidado de la salud, hogares inteligentes, entre otros.

1.4. Objetivos

1.4.1. Objetivo general

Usar 6LoWPAN para monitorear la salud estructural de un puente a través de un sensor con una dirección IPv6 según la tendencia del Internet de las Cosas.

1.4.2. Objetivos específicos

1. Describir conceptos referentes al Internet de las Cosas y al estándar 6LoWPAN, y analizar comparativamente aplicaciones que han sido desarrolladas en ámbitos similares.

2. Analizar la problemática que plantea el monitoreo de estructuras en un Sistema de Gestión de Puentes, proponer una solución y definir la metodología de trabajo.
3. Modelar la solución propuesta haciendo uso de los artefactos de análisis y diseño correspondientes según la metodología, y examinar alternativas hardware y software.
4. Construir un prototipo trabajando sobre el hardware y software seleccionados, guiándose por los artefactos elaborados.
5. Establecer métricas y validar los resultados obtenidos del prototipo a través de la realización de pruebas.

1.5. Organización del documento

El contenido del documento se encuentra dividido en siete capítulos organizados de la siguiente manera:

- El primer capítulo aborda la “Introducción” del proyecto de título.
- El segundo capítulo, denominado “Marco teórico”, profundiza en los contenidos más importantes que son tratados a lo largo del documento, referentes al Internet de las Cosas y a 6LoWPAN, proporcionando definiciones en el contexto del trabajo realizado y brindando una revisión de aplicaciones desarrolladas en ámbitos similares de acuerdo con el primer objetivo específico.
- El tercer capítulo, llamado “Solución propuesta”, aborda la problemática que plantea el monitoreo de la salud estructural de un puente, de acuerdo al segundo objetivo específico, aplicando el Internet de las Cosas en este contexto en particular y proponiendo una solución a la situación expuesta, junto con la selección de una metodología de trabajo adecuada. Además se contempla la evaluación de las herramientas hardware y software pertinentes para implementar un prototipo funcional que de respuesta al problema expuesto, según el tercer objetivo específico.
- El cuarto capítulo, titulado “Análisis y diseño”, gira en torno al modelamiento de la solución propuesta según la metodología de software seleccionada de acuerdo con el tercer objetivo específico.

- El quinto capítulo, denominado “Implementación”, expone la construcción del prototipo de acuerdo con el cuarto objetivo específico, según el proceso de análisis y diseño llevado a cabo en el capítulo anterior con el hardware y software seleccionados.
- El sexto capítulo trata sobre los “Resultados” obtenidos del proyecto, estableciendo métricas que permiten validarlo y realizar pruebas sobre el prototipo según al quinto objetivo específico.
- Finalmente, el séptimo y último capítulo aborda las “Conclusiones” obtenidas del trabajo realizado, además de proyectar posibles áreas para llevar a cabo esfuerzos a futuro.

2. MARCO TEÓRICO

A lo largo del presente capítulo se pretende describir los conceptos en torno a los cuales se efectuará el siguiente trabajo, girando principalmente en torno al “Internet de las Cosas” y al estándar 6LoWPAN (*IPv6 over Low-Power Wireless Personal Area Networks*), prestando especial atención en la generación de definiciones lo suficientemente claras. Además se llevará a cabo una revisión de aplicaciones que han sido desarrolladas en ámbitos similares.

2.1. El Internet de las Cosas

El “Internet de las Cosas” o IoT (*Internet of Things*) incorpora a todos los sistemas embebidos y redes de dispositivos que se encuentran de forma nativa con el protocolo IP habilitado, con acceso a Internet, junto con los servicios de monitoreo y control de estos dispositivos. El IoT, a grandes rasgos, describe una red de diferentes objetos de uso cotidiano interconectados entre sí, cuyo origen se remonta a 1999, siendo propuesto el concepto por Kevin Ashton¹.

Esta tecnología será, por lo tanto, una parte integral de la futura Internet, permitiendo establecer una definición conceptual en base a una infraestructura de red global dinámica, con capacidad de autoconfigurarse, basada en la normalización existente y que involucre protocolos de comunicación interoperables donde las “cosas” pueden tener una identidad, al igual que atributos físicos y virtuales, utilizando interfaces inteligentes integradas a la red para así hacer posible un intercambio mundial de información. En este contexto, el *European Research Cluster on the Internet of Things* plantea que la investigación y desarrollo del IoT se vuelve cada vez más compleja debido al nivel avanzado de la tecnología existente, la colaboración global e interdisciplinaria necesaria para las crecientes demandas de la sociedad y el mercado [Eur], donde el desarrollo de áreas como: nanotecnología, comunicaciones, sensores, teléfonos inteligentes, sistemas embebidos, *cloud computing* y tecnologías de software se vuelven esenciales para apoyar innovaciones en esta dirección.

Dado este escenario, con el fin de contextualizar el surgimiento del IoT es pertinente

¹<http://www.rfidjournal.com/articles/view?4986>

señalar que la red se encuentra formado por el “*core Internet*”, el cual representa una malla interconectada de dispositivos de ruteo que permiten la conexión de los sistemas terminales, contando con una alta capacidad de transporte de tráfico y modificándose solo ocasionalmente. La mayoría de los nodos de Internet hoy se encuentran en “*fringe Internet*”, la cual incluye a los computadores personales, portátiles y la infraestructura de red local conectada a Internet; la *fringe Internet* cambia rápidamente y se estima que posee sobre un billón de nodos. En este punto se presenta el IoT, también llamado “*embedded fringe*”, el cual se compone de dispositivos con el protocolo IP habilitado y con conexión a Internet, sensores, RFID (*Radio Frequency IDentification*), equipos de automatización en edificios, entre otros [She09]. El tamaño potencial del IoT se estima en trillones de dispositivos debido a la adopción del protocolo IPv6, mientras que su visión como una red global que vincula objetos [Taf12] se ejemplifica en la Figura 1.

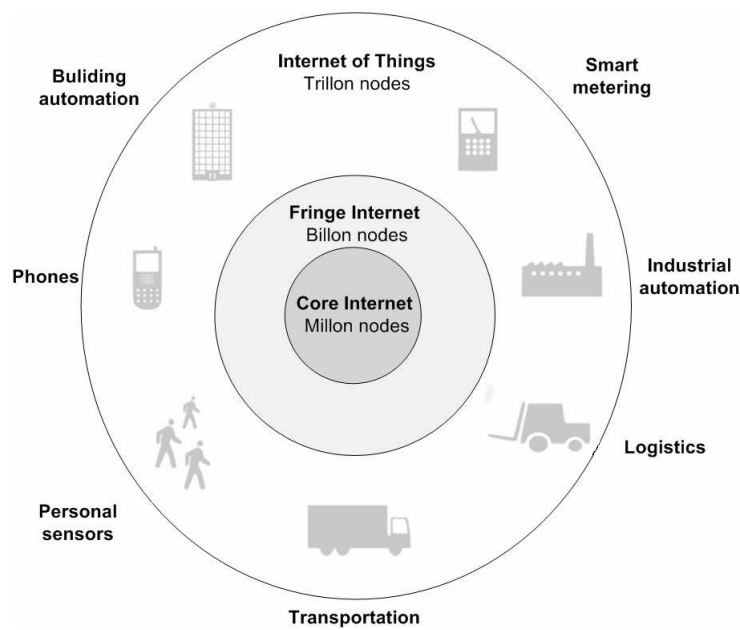


Figura 1. Visión del Internet de las Cosas

El Internet de las Cosas involucra a dispositivos embebidos o *smart objects* con el protocolo IP habilitado para ser parte integral de Internet. De esta forma, la adopción del protocolo IPv6 resuelve los problemas asociados al agotamiento de direcciones IPv4 debido a una mayor capacidad de direccionamiento, pasando desde los 32 bits hasta los 128 bits, lo que posibilita la interconexión *end-to-end* de dispositivos inalámbricos de baja potencia con capacidades de procesamiento limitadas.

Dada la problemática referente al agotamiento de direcciones IPv4, en la Figura 2 es

posible observar que en algunas zonas tales como LACNIC (*Latin America and Caribbean Network Information Centre*), de la cual Chile participa, se está pronto a entrar en una situación de “agotamiento virtual” bajo la cual los registros regionales no continúan asignando direcciones IPv4 en función de la necesidad, sino que se establece un límite máximo por organización [Lat12]. De esta forma, el direccionamiento a través del protocolo IPv6 permitirá dar conectividad a trillones de dispositivos, brindando soluciones en diversos ámbitos como: hogar, salud, logística, cuidado del medio ambiente, entre otros.

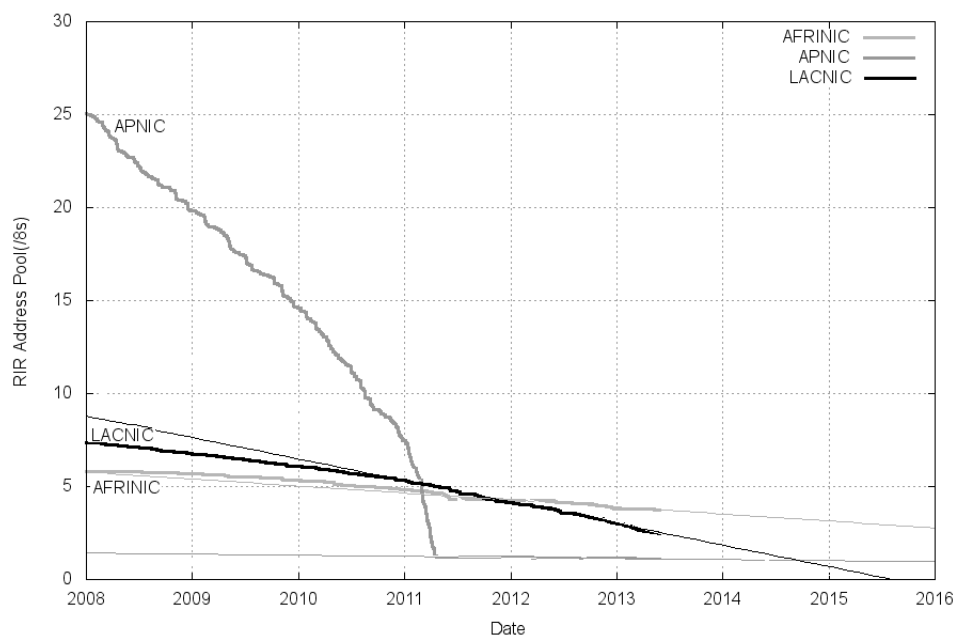


Figura 2. Agotamiento de direcciones IPv4

El crecimiento de Internet es un proceso continuo, que evoluciona desde una red de unos cuantos computadores interconectados hasta una red de objetos interconectados, donde el IoT puede hacer frente a los actuales desafíos de la sociedad: vigilancia, salud, deforestación, control de tráfico y reciclaje, dando lugar a un nuevo paradigma de conectividad. Este proceso se ve reflejado en los inusitados grados de conexión que es posible alcanzar en la actualidad, observándose este hecho claramente en la forma en que han evolucionado las relaciones sociales, permitiendo conectar no solo a personas con personas, sino que también a personas con objetos y a objetos entre sí.

El IoT representa un cambio en la forma en que cada individuo se relaciona con su entorno, por lo cual ha suscitado el interés de universidades, centros de investigación, empresas privadas y gobiernos, donde incluso la Unión Europea ha elaborado un plan de acción [Com09] para la obtención de beneficios en torno al bienestar individual para hacer

realidad un Internet de las Cosas dirigido a la población. Resulta pertinente mencionar que Internet nació conectando personas, para luego avanzar hacia máquinas capaces de comunicarse entre ellas sin necesitar al hombre; el IoT se mueve en el mundo de los objetos, donde éstos son capaces de interconectarse entre sí.

Cabe destacar que el Internet de las Cosas implica contar con métodos de almacenamiento masivo de grandes volúmenes de información, aludiendo al concepto de “*Big Data*”, el cual se produce como resultado de la democratización de la red y debido a factores del ecosistema digital tales como: proliferación de dispositivos móviles (*smartphones* y *tablets*), integración de sensores a la red, sistemas interconectados, redes sociales, cambios en los modelos de procesamiento de información, disponibilidad de sistemas de código abierto, la computación de propósito general y la integración de una red unificada [Cis11]. Además, en la actualidad se encuentran disponibles servicios como Xively², el cual proporciona almacenamiento en línea para la recepción de data proporcionada por sensores, fomentando ecosistemas digitales abiertos.

El principal objetivo del IoT es la creación de entornos inteligentes y “cosas” conscientes de sí mismas, en ambientes como: transporte, productos, ciudades (*smart city*), edificios, zonas rurales, suministro de energía y salud, para, a fin de cuentas, integrarse de manera natural a la vida de las personas, logrando llevar a cabo mejoras sustentables para la población, reflejadas en aplicaciones como, por ejemplo, para el clima, la alimentación, la energía, la movilidad, la sociedad digital y la salud, tal como se aprecia en la Figura 3 [Eur].

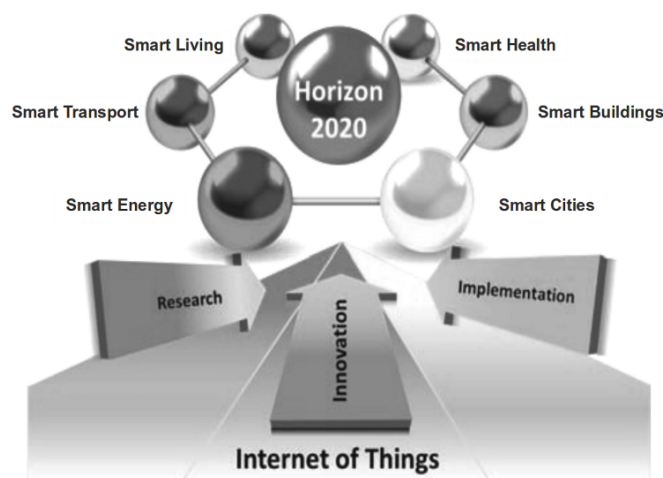


Figura 3. IoT para la creación de entornos inteligentes

²<https://xively.com/>

En esta nueva realidad, el número de dispositivos con capacidad para conectarse a Internet se incrementa año tras año, lo cual es corroborado por IMS Research, donde, según datos proporcionados en octubre del año 2012 [IMS12] la compañía muestra que la ola de dispositivos conectados a la red se está acelerando, lo cual impulsaría un fuerte crecimiento del mercado de los aparatos con capacidad de conexión, previendo un total de 28 billones de dispositivos conectados a Internet a finales del año 2020.

También, IMS Research (recientemente adquirida por IHS Inc.) estimó que a finales del año 2012 los dispositivos conectados e Internet llegarían a un total de 9.6 billones, donde es importante destacar que aproximadamente el 75 % de los artefactos se encuentra referido a segmentos de mercado considerados como “dispositivos conectados” existentes hoy en día, como es el caso de comunicaciones fijas y móviles, ordenadores y electrónica de consumo. Para el año 2020, se prevé que el segmento descrito anteriormente representará la mitad de los artefactos con conectividad a Internet; se incluirá a la infraestructura de la red a sectores médicos (cuidado de la salud de las personas), industriales y automotrices. En este sentido, se alzan sectores que representan oportunidades de crecimiento hoy en día, relacionados a aquellos dispositivos que hoy no integran todavía el protocolo IP para obtener conectividad y una dirección de red, junto con la tecnología de procesamiento subyacente para lograr estar integrado como un nodo final.

De acuerdo con Bill Morelli, director asociado de IMS Research [IMS12], hoy existen una serie de factores que ejercen presión para obtener dispositivos direccionables a través de la red, entre los cuales se incluyen los avances en cuanto a la tecnología de los procesadores, especialmente aquellos de bajo consumo de energía, los cuales resultan críticos para dispositivos industriales que se encuentran desconectados de una fuente de energía, dependiendo en gran medida de baterías, lo cual implica que es necesaria una alta eficiencia energética.

En este contexto se proyecta el IoT como una oportunidad de desarrollo para organizaciones, con un mercado que apunta hacia dispositivos que se conectan a Internet con una dirección IP única. La futura Internet contará con nodos que van desde sensores básicos y dispositivos de telemetría, incluyendo a objetos que actualmente juegan un rol omnipresente en la vida diaria, hasta dispositivos informáticos de gran alcance con un sistema operativo completo y con una interfaz de usuario (UI).

2.2. El estándar IEEE 802.15.4

En cuanto a la estandarización de las comunicaciones que sustentan el Internet de las Cosas (IoT), en el año 2003 el IEEE 802.15.4 [IEE03] se posiciona como el primer estándar de radio de bajo consumo, constituyendo el primer paso hacia la estandarización de 6LoWPAN (*IPv6 over Low-Power Wireless Personal Area Networks*) en el año 2007. Sin embargo, cabe destacar que en primera instancia presentó dificultades de escalabilidad, capacidad de evolución e integración con Internet, las cuales han sido tratadas durante los últimos años [She09].

Este estándar define los niveles de red básicos para dar servicio a redes inalámbricas de área personal o WPAN (*Wireless Personal Area Networks*), cuyo principal uso radica en aplicaciones de monitoreo para brindar comunicación entre dispositivos embebidos de bajo coste, bajo consumo de energía y bajo uso del ancho de banda como, por ejemplo, en redes de sensores inalámbricas o WSN (*Wireless Sensor Networks*).

En la Tabla 1 se presenta una clasificación de las diferentes tecnologías de comunicación inalámbricas [Kuo07], señalando sus principales características y aplicaciones típicas, a fin de observar las condiciones que presenta el 802.15.4, donde se concentrará la atención del presente trabajo.

Tabla 1. Clasificación de tecnologías de comunicación inalámbricas

Clase	Velocidad de datos	Radio de cobertura	Aplicaciones típicas	Ejemplo de tecnologías
WWAN	<10 Mbps	>10 km	Telefonía, Internet móvil,	GSM, UMTS, satélite
WMAN	<100 Mbps	<10 km	Internet banda ancha	IEEE 802.16, HIPERMAN
WLAN	<100 Mbps	<100 m	reemplazo LAN alámbrica	IEEE 802.11, HIPERLAN/2
WPAN	<10 Mbps	<10 m	transferencia de datos personales	Bluetooth, IEEE 802.15.3
WSN	<1 Mbps	<1 km	monitoreo, control	redes propietarias, IEEE 802.15.4, RFID

El estándar IEEE 802.15.4 tiene como principales atributos su bajo coste en términos de requerimientos y bajo consumo de energía, destacando en aplicaciones domóticas e industriales. Con una trama que no debe exceder los 127 Bytes de longitud, el estándar ofrece tasas de transmisión de datos de hasta 250 Kb/s, brindando soporte para aplicaciones que no requieran gran velocidad de transmisión, o para nodos alimentados por baterías donde el energía sea un factor crítico. En la Figura 4 se muestra el espacio de operación de los estándares WLAN Y WPAN [Gut04]; cabe destacar que el 802.15.4 no está diseñado para solaparse con estándares de redes inalámbricas de gama alta.

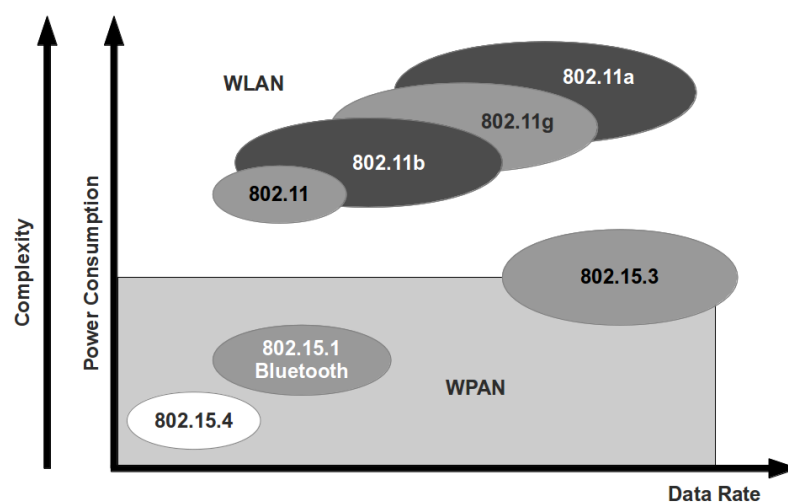


Figura 4. Espacio de operación de WPAN y WLAN

La Tabla 2 [Gut04] presenta un resumen de las principales características de una LR-WPAN (*Low-rate Wireless Personal Area Network*) usando el estándar IEEE 802.15.4, comparado con el estándar IEEE 802.11b y el estándar 802.15.1 (WPAN/Bluetooth).

Tabla 2. Comparación de LR-WPAN con otras tecnologías inalámbricas

	802.11b WLAN	Bluetooth WPAN	Low Rate WPAN
Rango	~100 m	~10 - 100 m	10 m
Velocidad de datos	~2 - 11 Mb/s	1 Mb/s	≤ 0.25 Mb/s
Consumo de energía	Medio	Bajo	Ultra bajo
Tamaño	Largo	Pequeño	Pequeñísimo
Costo/Complejidad	Alto	Medio	Muy bajo

Es importante destacar que el 802.15.4 no es sinónimo de la especificación ZigBee, ya que el 802.15.4 es un protocolo de capas Física y de Enlace (capas OSI 1 y 2), mientras que ZigBee es un protocolo de capa de red (capa OSI 3) que se sitúa sobre el IEEE 802.15.4. Además, ZigBee es una marca comercial de la ZigBee Alliance³, grupo que crea y mantiene el estándar.

De esta forma, es posible manifestar con determinación que la tecnología LR-WPAN busca brindar solución en ámbitos donde una WPAN resulta demasiado costosa y donde se requiere un funcionamiento con un consumo mínimo de energía.

2.3. 6LoWPAN

El término 6LoWPAN (*IPv6 over Low-Power Wireless Personal Area Networks*) hace referencia a un conjunto de estándares de Internet definidos por la IETF (*Internet Engineering Task Force*), que permiten el uso del protocolo IPv6 en redes de área personal inalámbricas de baja potencia o LR-WPAN, convirtiéndose en la clave para hacer realidad el Internet de las Cosas. En la Figura 5 se comparan los *stack* IP y 6LoWPAN respectivamente, introduciéndose en este último una capa de adaptación y optimización [She09], la cual define la forma de llevar a cabo la comunicación IPv6 sobre tramas 802.15.4, principalmente a través de elementos como: la compresión de cabeceras, fragmentación y reenvío de paquetes sobre la capa de enlace.

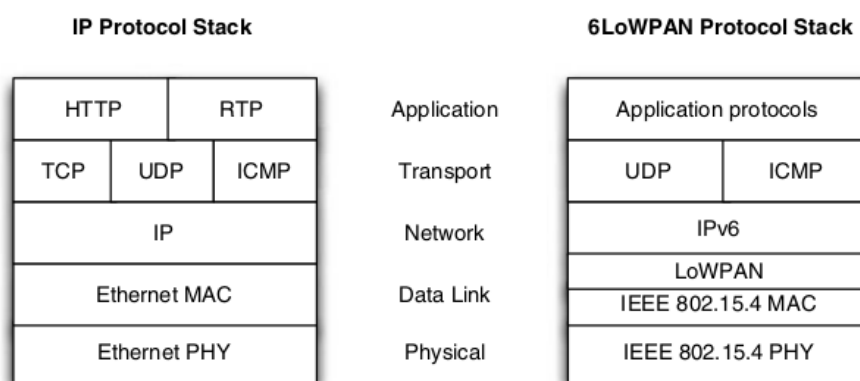


Figura 5. Stack IP y Stack 6LoWPAN

6LoWPAN es un estándar publicado en el año 2007 por la IETF, el cual optimiza IPv6 para el uso de tecnologías de bajo ancho de banda y bajo consumo de energía.

³<http://www.zigbee.org/>

6LoWPAN rompe las barreras a la utilización de IPv6 en baja potencia, en dispositivos con procesamiento limitado a través de redes inalámbricas con una baja utilización del ancho de banda, donde los objetos se transforman en nodos de Internet con una dirección IPv6, contando a priori con conectividad extremo a extremo con cualquier otro nodo de la red. Este estándar, definido en el RFC 2460 [Dee98], es la nueva versión del protocolo de Internet, el cual se desarrolló en la década de 1990 como una solución al rápido crecimiento y los desafíos que enfrenta Internet; el futuro crecimiento del Internet de las Cosas va a ser posible gracias a IPv6.

En este escenario, dada la utilización de 6LoWPAN, son los nodos finales u objetos los que ofrecen directamente los servicios Web (a través de SensorML, XML o JSON). Por este motivo, se dice que 6LoWPAN da pie al paradigma de la “Web de las Cosas” o “*Web of Things*” [Peñ13].

Es importante recalcar que 6LoWPAN es un estándar que hace referencia a tecnologías abiertas que son posibles de implementar por cualquier individuo u organización, sin infringir patentes o asumir costos de propiedad intelectual. En estas circunstancias, son evitados protocolos y pasarelas propietarias que introducen retardos, limitaciones y costes por licenciamiento. De esta manera los beneficios del uso de protocolos de Internet, y por lo tanto, la integración de los objetos con la red, implica que los dispositivos se puedan conectar fácilmente con redes más grandes o con servicios basados en Internet sin necesidad de *gateways* o *proxies*.

La nueva Internet se creará mediante la conexión de islas de dispositivos inalámbricos embebidos, donde cada isla posee conexión con la red. En este escenario, la arquitectura de 6LoWPAN se compone de redes de área local con acceso inalámbrico y de baja potencia (LoWPANs), las cuales tienen acceso a Internet, tal como se observa en la Figura 6. De esta manera, una LoWPAN (*Low-Power Wireless Personal Area Networks*) es entendida como una colección de nodos 6LoWPAN que comparten un prefijo de dirección IPv6 común (primeros 64 bits), pudiendo identificar a una red LoWPAN simple como una red que se conecta a través de un “*edge router*”, o router de borde, a otra red IP; por otro lado una red LoWPAN ad-hoc no está conectada a Internet, pero opera sin una infraestructura subyacente, mientras que una red LoWPAN extendida abarca múltiples router de borde a lo largo de un enlace [She09].

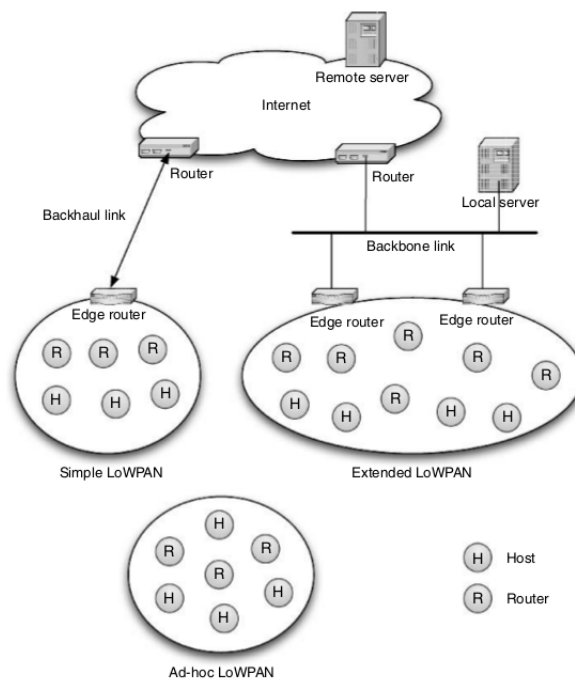


Figura 6. Arquitectura de 6LoWPAN

6LoWPAN permite la utilización de IPv6 sobre redes de bajo consumo y pérdida o LLN (*Low power and Lossy networks*), donde se resuelven los siguientes desafíos presentados por el uso del protocolo IPv6:

- Las versiones del protocolo IP (IPv4/IPv6) asumen que los dispositivos siempre están conectados a la red, a pesar de que no estén transmitiendo. En 6LoWPAN, los dispositivos sólo se conectan a la red cuando deben transmitir información, lo que permite un uso eficiente de la energía necesaria para dicha operación.
- IPv6 requiere de multicast. 6LoWPAN incorpora el uso de este método de transmisión en comunicaciones inalámbricas de baja potencia, lo que anteriormente no era soportado por las tecnologías de comunicación de radio existentes.
- La topología de malla es más eficiente en el uso de la energía. 6LoWPAN permite utilizar de manera más fácil este tipo de topología de red con IPv6.
- 6LoWPAN comprime el tamaño de las cabeceras a 6 Bytes, disminuyendo el tamaño del frame necesario para transmitir sobre LLNs, el cual se encuentra definido por el estándar IEEE 802.15.4 en 127 Bytes en capa de enlace, con una carga de al menos 72 Bytes.
- 6LoWPAN optimiza el uso de los estándares de Internet sobre redes inalámbricas de baja potencia.

6LoWPAN se ha definido para trabajar sobre el estándar de redes de radio IEEE 802.15.4. En este sentido el RFC 4919 [Kus07] define el uso de IPv6 sobre 6LoWPAN, considerando la transmisión de paquetes IP sobre el 802.15.4. En este ámbito existen ciertos elementos a considerar, tales como el tamaño del frame, el cual para IPv6 puede ser de 1280 Bytes, mientras que 6LoWPAN define un tamaño de 127 Bytes, por lo que se estandarizó en el RFC 4944 [Mon07] la transmisión de paquetes IPv6 sobre el 802.15.4, mientras que el RFC 6282 [Hui11] trata sobre la compresión de cabeceras IPv6 y UDP para nivel de transporte (no orientado a la conexión y no fiable) en 6 Bytes; además, el RFC 6775 [She12] incorpora aspectos sobre el descubrimiento de vecinos (*Neighbor Discovery*) en este tipo de redes. En la Figura 7 se muestra un paquete con la cabecera IPv6 completa sobre un frame 802.15.4, mientras que en la Figura 8 la compresión de cabeceras 6LoWPAN/UDP.

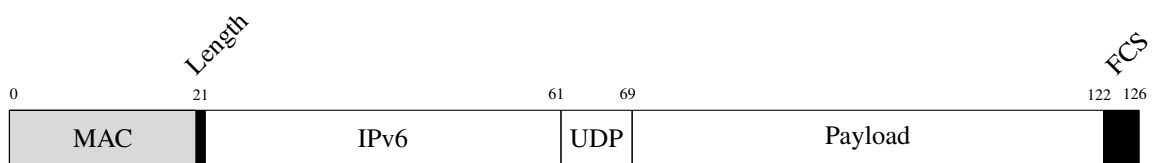


Figura 7. Frame IEEE 802.15.4 con direccionamiento Full UDP/IPv6 (64-bit)

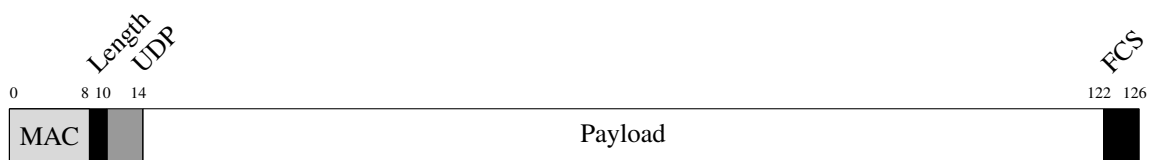


Figura 8. Frame IEEE 802.15.4 con direccionamiento mínimo UDP/6LoWPAN (16-bit)

Dados los antecedentes expuestos acerca de la arquitectura y el tamaño del frame 6LoWPAN, es posible mencionar que el formato de la cabecera, en su forma más simple, posee los campos expuestos en la Figura 9, con un *dispatch value*, donde los 2 bits más significativos representan lo expuesto en la Tabla 3, y la cabecera IPv6 comprimida (LOWPAN_IPHC) en 2 Bytes; posteriormente, le sigue la cabecera UDP (LOWPAN_NHC) proporcionando información acerca del puerto de origen (Src), puerto de destino (Dst) y UDP checksum en 4 Bytes [She09].

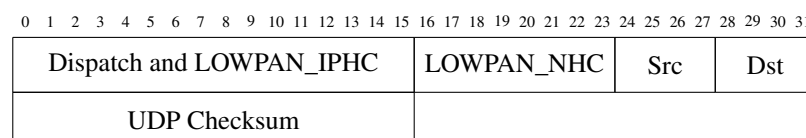


Figura 9. Cabecera 6LoWPAN/UDP comprimida (6 Bytes)

Tabla 3. Los 2 bits más significativos del dispatch byte

Patrón	Tipo de Cabecera
00	No es un paquete LoWPAN (NALP)
01	Envío normal
10	Cabecera de ruteo en capa de enlace
11	Cabecera de fragmentación

Por lo tanto, en el mejor de los casos la cabecera 6LoWPAN/UDP se encontrará en 6 Bytes de longitud; para efectos de comparación, una cabecera IPv6/UDP estándar utilizará 48 Bytes de longitud, como se muestra en la Figura 10. En el peor de los casos, se debe tener en cuenta que un frame IEEE 802.15.4 tiene solo 72 Bytes de carga disponibles luego de las cabeceras de la capa de enlace.

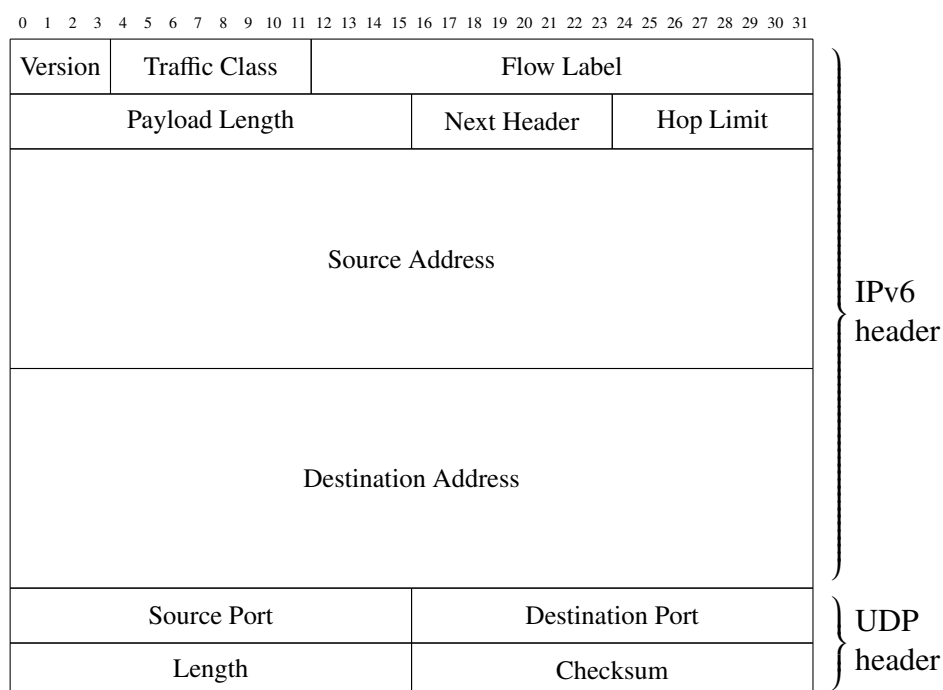


Figura 10. Cabecera estándar IPv6/UDP

En resumidas cuentas, actualmente se cuenta con dispositivos de corto alcance y baja potencia de radio, lo que limita las velocidades de transferencia de datos, tamaño del frame y ciclos de trabajo (energía). En este sentido, la normalización del IEEE 802.15.4 fue el primer paso para la estandarización de 6LoWPAN, dados los RFC 4919 [Kus07] y RFC 4944 [Mon07] en el año 2007, los cuales fueron completados por el grupo de

trabajo (*Working Group*) de la IETF llamado “*IPv6 over Low power WPAN*”⁴, quienes se aseguran de generar los documentos necesarios para garantizar implementaciones interoperables de 6LoWPAN, definiendo los protocolos de seguridad y gestión necesarios para la construcción de estas redes.

No obstante, cabe mencionar que el desarrollo a continuado a lo largo de estos años en materias de compresión y descompresión de cabeceras, *Neighbor Discovery*, y ruteo. En este último punto en el año 2008 se forma el grupo de la IETF “*Routing over Low-power and Lossy Networks*” o ROLL⁵, donde también es reconocida la necesidad de estandarizar el ruteo IPv6 sobre redes de bajo consumo y pérdida o LLN (*Low power and Lossy networks*), obteniendo como resultado a RPL (*Ripple routing protocol*), definido en el RFC 6550 [Win12].

Además, entre los beneficios de la utilización de comunicación IP sobre LLN, como lo provee 6LoWPAN, es posible destacar: el uso de la infraestructura actual de la red, 6LoWPAN es un estándar abierto, las tecnologías basadas en IP han existido durante décadas, por lo tanto se ha demostrado su funcionamiento y escalabilidad, mientras que ya existen herramientas de gestión, puesta en marcha y diagnóstico de redes basadas en IP.

2.4. Sistemas embebidos

La forma en que los objetos se conectarán hacia Internet será a través de sistemas embebidos, los cuales realizan un conjunto limitado de operaciones y no requieren un uso intensivo de la red para enviar información.

Últimamente los dispositivos electrónicos se han vuelto cada vez más baratos, jugando un rol omnipresente en la vida diaria. De esta manera, hoy es posible construir un sistema integrado del tamaño de una billetera con la capacidad equivalente a la de un computador de la década de los 90. Este tipo de soluciones embebidas puede ser apoyado con sistemas operativos de funcionalidades reducidas. Desde esta perspectiva, la aparición de redes de sensores inalámbricas o WSN (*Wireless Sensor Networks*) es esencialmente la última tendencia de la Ley de Moore hacia la miniaturización y la ubicuidad de los dispositivos

⁴<http://datatracker.ietf.org/wg/6lowpan/charter/>

⁵<http://datatracker.ietf.org/wg/roll/charter/>

informáticos [Yu06], lo que permitiría un intercambio mundial de información.

Un sistema embebido o empotrado se refiere a un sistema diseñado para realizar un conjunto específico de tareas, de uso único. Generalmente se encuentran incluidos en una única placa o SBC (*Single Board Computer*) y funcionan con un sistema operativo en la memoria interna de la placa, siendo normalmente desarrollados por los fabricantes de la misma.

Los sistemas embebidos suelen usar un procesador relativamente pequeño y una memoria limitada para reducir los costes. Además, los programas para sistemas embebidos se enfrentan normalmente a problemas de tiempo real.

Además, dado el conjunto limitado de operaciones que realizan estos sistemas, se permite al usuario modificar determinados parámetros para adaptarlos a su uso, viéndose de esta forma limitado por las posibilidades que ofrece el sistema para el cual fueron desarrollados.

Entre las aplicaciones que pueden ser mencionadas a modo de ejemplo, es posible destacar: POS (*Point Of Service*) en supermercados, decodificadores y *set-top boxes* para televisores, cajeros automáticos, routers, switches, *firewalls*, teléfonos celulares, reproductores de video, reproductores portátiles de música y últimamente, para redes de sensores de todo tipo, entre otros.

En este contexto, los dispositivos embebidos (Figura 11) involucran un nuevo paradigma con capacidades de procesamiento limitadas que participan intercambiando información en el Internet de las Cosas (IoT) [She09], formando una red embebida de aparatos conectados facilitada por el gran número de nodos que es posible comunicar utilizando el protocolo IPv6.

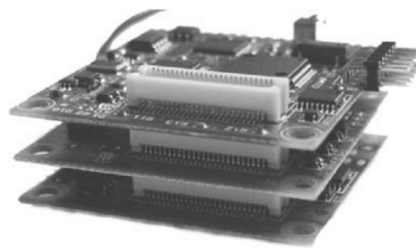


Figura 11. Dispositivo embebido

De la mano de los dispositivos empotrados, o embebidos, se comienza a hacer tangible el IoT. Sin embargo, en lo que respecta a los objetos de la nueva Internet, con el paso de los años, las capacidades del procesador de los dispositivos han aumentado, y también la complejidad de las normas de comunicación, servicios y protocolos, algunos de ellos propietarios, tales como ZigBee, Z-Wave o HomePlug, lo cual dificulta la integración con redes más grandes o con servicios basados en Internet. Los beneficios que se desprenden de la utilización de protocolos abiertos para Internet radican principalmente en que es posible conectar dispositivos fácilmente a otras redes usando la infraestructura actual.

La especificación de 6LoWPAN desarrollada en el año 2007 alienta a una serie de organismos a conducir esfuerzos hacia la implementación de la norma. En este sentido, en el año 2008 la ISA⁶ (*International Society of Automation*) comenzó la estandarización de un sistema inalámbrico de automatización industrial llamado SP100.11a (también conocido como ISA100), que se basa en 6LoWPAN. Otras actividades relacionadas incluyen al *IP for Smart Objects (IPSO) Alliance*⁷, alianza fundada en 2008 para promover el uso de IP en los objetos inteligentes y la adopción del IoT por parte del mercado. Finalmente la *IP500 Alliance*⁸ desarrolla recomendaciones para 6LoWPAN sobre subseñales de radio del IEEE 802.15.4 [She09].

2.5. Wireless sensor networks

Uno de los elementos más importantes del Internet de las Cosas (IoT) corresponde a las redes de sensores inalámbricos o WSN (*Wireless Sensor Network*), las cuales entregan información capturada desde el mundo físico para luego poder ser accedida desde la red y/o algún sistema computacional, posibilitando la colaboración y prestación de servicios a través del intercambio de información.

Una WSN se puede definir básicamente como un conjunto de elementos autónomos o nodos que se encuentran interconectados de manera inalámbrica, colaborando entre sí para satisfacer las necesidades requeridas, con un objetivo en común. Las WSN se encuentran conformadas por un grupo de sensores que poseen capacidades sensitivas y de comunicación inalámbrica, permitiendo la formación de redes de tipo ad-hoc sin

⁶<http://www.isa.org/>

⁷<http://www.ipso-alliance.org/>

⁸<http://www.ip500alliance.org/>

una infraestructura física preestablecida ni administración central. Además, se caracterizan por su facilidad de despliegue y por ser autoconfigurables, pudiendo convertirse en todo momento en emisor, receptor, ofrecer servicios de encaminamiento entre nodos con visión directa, así como registrar datos referentes a los sensores locales de cada nodo. La Tabla 4 resume los elementos básicos para el desarrollo de una WSN [Jim12].

Típicamente, una WSN puede conectar cientos, e incluso miles, de dispositivos sensores, los cuales cuentan con la capacidad de comunicarse entre sí e intercambiar información, contando con una o más estaciones base que actúan como sumideros de toda la información generada y recogida por la red.

Tabla 4. Elementos principales de una WSN

Elemento	Función
Sensor	Toma información del entorno y las convierte en señales eléctricas
Nodo (Mote)	Toma los datos del sensor y los envía a la estación base
Gateway	Elemento para la interconexión entre la red de sensores y una red de datos (TCP/IP)
Estación base	Colector de datos basado en una computadora común o un sistema embebido

Dada la estructura de una red inalámbrica de sensores y las limitaciones existentes, considerando la potencia de transmisión de los nodos, no todos los dispositivos podrán enviar o recibir directamente paquetes de una estación base, lo que hace necesario un mecanismo de retransmisión o encaminamiento de los diferentes paquetes que circulan por la red. En la Figura 12, se observa la estructura típica de una WSN [Kas13].

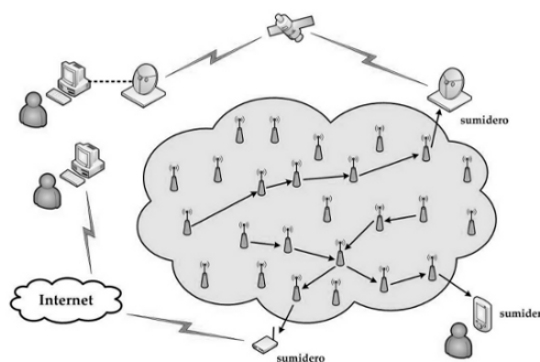


Figura 12. Estructura de una WSN

Un nodo de la WSN esta integrado en una o varias *boards*, y empaquetado en unos pocos centímetros cúbicos. De esta forma, los sensores se transforman en objetos inteligentes o *smart objects*, los cuales permiten una amplia gama de aplicaciones en áreas tales como: automatización del hogar, automatización de edificios, control de fábricas, ciudades inteligentes (*smart city*), monitoreo de la salud estructural, monitoreo de la salud de las personas, redes inteligentes, gestión de energía y transporte, seguridad, entre otros. Los *smart objects* pueden ser operados por baterías, pudiendo identificar típicamente a tres componentes: una CPU (un microcontrolador de 8, 16 o 32 bits), memoria de unas pocas decenas de kilobytes y un dispositivo de comunicación inalámbrica de baja potencia (desde unos pocos kbit/s a unos pocos cientos de kbit/s).

Sin embargo, resulta importante reiterar que las fuertes restricciones en términos de recursos que suponen las WSN, lo cual repercute en un gran reto tecnológico a superar debido a las características intrínsecas de los dispositivos que constituyen este tipo de redes, sumando además la variabilidad en la topología de la red y el ancho de banda disponible del canal, el movimiento de los nodos y la ausencia de infraestructura predeterminada.

Cabe destacar que el desarrollo técnico de sensores de bajo coste y actuadores, en combinación con las tecnologías de comunicación de baja potencia, tales como el IEEE 802.15.4, se ha visto influido y/o afectado por la irrupción de sistemas propietarios o semi-cerrados que han dado lugar a soluciones parciales y no interoperables entre sí. Muchas de las arquitecturas no IP actuales están moviéndose hacia arquitecturas totalmente IP (en un proceso de convergencia), debido a la complejidad inherente del desarrollo de puertas de enlace difíciles de gestionar [Dun10].

2.6. Aplicaciones existentes

Llegado este punto, resulta de gran importancia revisar algunas implementaciones que consideran la integración de 6LoWPAN (*IPv6 over Low-Power Wireless Personal Area Networks*) para la nueva Internet, con el fin de contribuir a desarrollar una solución para la problemática a enfrentar, la cual involucra, en el presente proyecto, el monitoreo de la salud estructural de un puente.

Integrar la comunicación IP en un dispositivo embebido se convierte en una tarea complicada debido a la existencia de un hardware altamente integrado y diseños de software que a menudo no permiten una abstracción completa del hardware subyacente. Además, los dispositivos de la nueva Internet presentan características que repercuten en consideraciones especiales de diseño a modo de maximizar la eficiencia de los recursos disponibles.

2.6.1. Implementaciones del stack 6LoWPAN

Una forma de integrar 6LoWPAN en un dispositivo embebido con capacidad de comunicación inalámbrica es mediante la integración de una pila de protocolos (*protocol stack*) existente, la cual se incluye en un sistema operativo de funcionalidades reducidas o en un proyecto de software embebido.

Una pila de protocolos para 6LoWPAN (Figura 13) incluye, como mínimo, los siguientes componentes básicos:

- Controladores de radio
- Acceso a capa de enlace (IEEE 802.15.4)
- IPv6 [Dee98] con 6LoWPAN [Mon07] a nivel de red
- UDP [Pos80] a nivel de transporte
- Descubrimiento de vecinos (*Neighbor Discovery*)
- API para la pila de protocolos

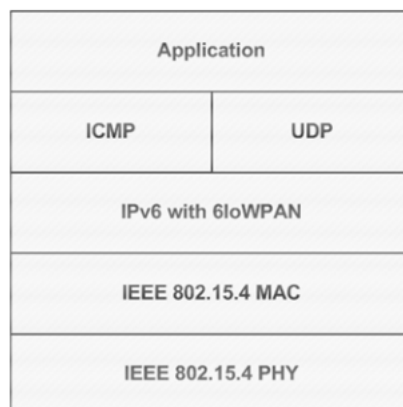


Figura 13. Pila de protocolos 6LoWPAN

También, de manera opcional, una pila de protocolos o *stack* podría incluir uno o más protocolos de enrutamiento; además, cabe señalar que un *stack* 6LoWPAN es muy compacto, tratándose a menudo de un módulo entre unos 15-20 kilobytes. De esta manera, a continuación se expondrán 5 *stack* disponibles actualmente.

Contiki

Contiki⁹ es un sistema operativo de código abierto para sistemas en red que poseen restricciones de memoria, donde se cuentan arquitecturas como AVR y MSP430, y cuyo énfasis recae en la comunicación inalámbrica con un bajo consumo de energía para dispositivos. Fue creado por Adam Dunkels¹⁰ en el año 2002, siendo mantenido por una red de desarrolladores a lo largo del mundo que abarca, por ejemplo, a Cisco, Atmel, SICS (*Swedish Institute for Computer Science*), Sensinode, ST Microelectronics, Zolertia, entre otros.

Contiki incluye una pequeña implementación IP llamada μ IP, junto con una implementación de IPv6 con soporte 6LoWPAN llamada μ IPv6. Su arquitectura se muestra en la Figura 14, donde es posible observar que la abstracción de hardware de bajo nivel se divide en plataforma y CPU para la portabilidad, incluyendo controladores de hardware. Además, cuenta con un núcleo orientado a eventos y procesos que usan protohilos (*protothreads*), los cuales representan una abstracción para proporcionar un estilo de programación secuencial sobre el núcleo orientado a eventos.

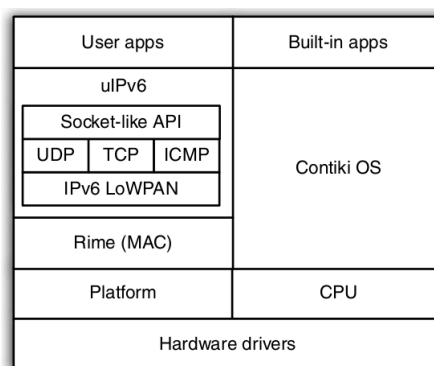


Figura 14. La arquitectura de Contiki

TinyOS

TinyOS¹¹ es un sistema operativo de código abierto desarrollado para redes de sensores

⁹<http://www.contiki-os.org/>

¹⁰<http://dunkels.com/adam/>

¹¹<http://www.tinyos.net/>

inalámbricas (WSN), diseñado para dispositivos embebidos de bajo consumo de energía y con capacidad limitada de memoria flash y RAM. El sistema operativo utiliza una estructura basada en componentes y un modelo de ejecución conducido por eventos. Además, algunas de las características de orientación a objetos que posee TinyOS son realizadas mediante la utilización de NesC (*Network Embedded Systems C*).

Una implementación 6LoWPAN se encuentra disponible para TinyOS, la cual fue desarrollada por la Universidad de California en Berkeley, siendo conocida como BLIP (*Berkeley IP Implementation*), el cual es un stack IPv6 que incluye compresión de cabeceras para 6LoWPAN, descubrimiento de vecinos, enrutamiento y soporte para programación en red; los protocolos de transporte TCP y UDP han sido implementados, junto con varios protocolos de aplicación. En la Tabla 5 se presenta una comparación entre los sistemas operativos para WSN TinyOS y Contiki [Gar09].

Tabla 5. Comparación entre TinyOS y Contiki

	TinyOS	Contiki
Licencia	BSD	BSD
Requisitos de RAM	512 Bytes	2 kilobytes
Requisitos de ROM	8 kilobytes	40 kilobytes
Características de diseño	Modularidad, Abstracción de componentes, Concurrencia, Planeamiento (Scheduling)	Alta portabilidad, Multitarea, Carga dinámica de servicios
Tipo de kernel	Manejado por eventos	Manejado por eventos y protothreads
Stack implementado	6LoWPAN, ZigBee	μ IP, μ IPv6, Rime
Lenguaje de programación	NesC	C, C++
Herramientas de simulación	TOSSIM	COOJA
Hardware implementado	MICAz, TelosB	Sky/TelosB, ATMEL y AVR

NanoStack

NanoStack 2.0¹² corresponde a un producto comercial basado en el stack 6LoWPAN de la compañía finlandesa Sensinode, la cual fue adquirida en Agosto del año 2013 por ARM.

NanoStack 2.0 entrega independencia a fabricantes de chips e integradores de sistemas a través de una solución compacta y optimizada específicamente para redes 6LoWPAN en sistemas de radio en un chip. Utiliza aproximadamente 32 kilobytes de memoria flash, presenta soporte para 2.4 GHz (IEEE 802.15.4) y Sub-1GHz (IEEE 802.15.4g), junto con la posibilidad de soportar aplicaciones y mecanismos de transporte como UDPv6, ICMPv6 y TCP.

Jennic 6LoWPAN

Jennic¹³ es un fabricante de chips inalámbricos especializado en la tecnología de radio IEEE 802.15.4, adquirido por NXP; destacan por ser los primeros fabricantes en ofrecer un stack 6LoWPAN para la norma 802.15.4 en sus productos, además de su stack propietario JenNet y soporte para ZigBee [She09]. También se cuenta con un acceso a programación de aplicaciones a través de una API que provee abstracción de alto nivel para acceder a cualquiera de las pilas de protocolo que ofrece Jennic. Además, es provisto un protocolo propietario similar a SNMP, llamado SNAP (*Simple Network Access Protocol*) para la configuración y administración de nodos.

Nivis ISA100

Nivis ISA (NISA) corresponde a un protocolo del estándar ISA100.11a que incluye componentes para dispositivos de campo, puertas de enlace, sistema de administración y seguridad. Es compatible con y sin una infraestructura de red troncal, presenta soporte para IEEE 802.15.4 y para el stack NISA100.11a, además de estar disponible para plataformas Freescale y Texas Instruments.

2.6.2. Sistemas basados en 6LoWPAN

Para el uso de 6LoWPAN en Internet, normalmente una aplicación embebida usa una pila de protocolos para configurar la red, junto con enviar y recibir paquetes. De esta manera,

¹²<http://www.sensinode.com/EN/products/software.html>

¹³<http://www.jennic.com/>

para la puesta en marcha deben considerarse aspectos como el canal de radio, frecuencia de envío, modo de la capa de enlace (MAC) y seguridad en capa de enlace. Además, se debe prestar atención al rol de los dispositivos dentro de la red (*host*, router o router de borde), direccionamiento, movilidad, fiabilidad de los datos y seguridad de la red.

En este contexto, el Internet de la Cosas ha comenzado a materializarse en los últimos años a través de sistemas que, integrando el *stack* 6LoWPAN, satisfacen las necesidades por medio de una red conectada de objetos.

Thingsquare

En la dirección de los esfuerzos realizados para conectar cosas, resulta pertinente destacar a Thingsquare¹⁴, una compañía fundada por Adam Dunkels, creador de Contiki. Thingsquare vende soluciones para dispositivos conectados para monitoreo de las condiciones climáticas, iluminación, ciudades inteligentes y hogares conectados.

ISA100 automatización industrial

Dado este escenario, cabe mencionar un sistema basado en ISA100, un estándar para automatización industrial inalámbrico con respaldo de la Sociedad Internacional de Automatización o ISA (*International Society of Automation*). ISA100.11a ha sido diseñado para apoyar tareas de monitoreo no crítico, alerta y supervisión, donde se puedan tolerar latencias del orden de 100 milisegundos. La norma define una pila de protocolos, la gestión del sistema y funciones de seguridad para su uso en redes inalámbricas de baja potencia y baja velocidad de transmisión de datos, donde se utiliza el estándar 6LoWPAN.

Idesco Cardea

Idesco¹⁵ junto a Sensinode desarrollaron un sistema para la implementación de un lector RFID (*Radio Frequency IDentification*) para la identificación y control de accesos utilizando 6LoWPAN (Figura 15).

El sistema Cardea de Idesco ofrece una línea de productos RFID que incluye: lectores, puertas con control de acceso y una unidad de control. El sistema usa la tecnología de red inalámbrica de Sensinode en base a 2.4 GHz, IEEE 802.15.4 y 6LoWPAN, utilizando una

¹⁴<http://thingsquare.com/>

¹⁵<http://www.idesco.fi/>

red *mesh* para interconectar los dispositivos.

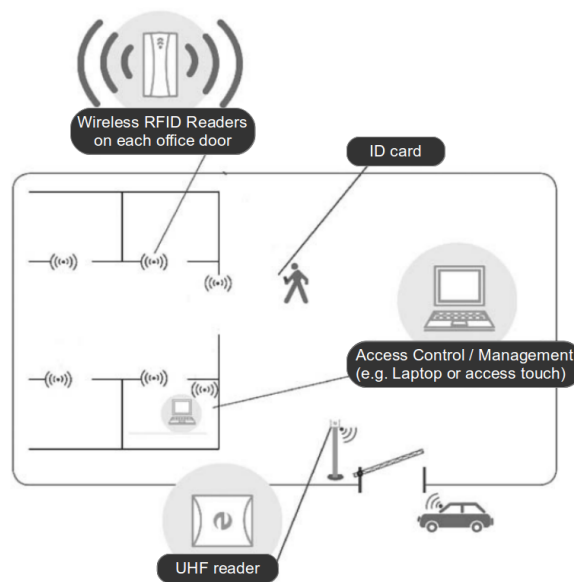


Figura 15. La arquitectura de Idesco Cardea

Grupo Elster

Considerando la aplicación de IPv6 en una *Smart Grid*, o red inteligente, para el control y monitoreo de elementos conectados a redes eléctricas, es importante destacar que el Instituto Nacional de Estándares y Tecnología de Estados Unidos o NIST (*National Institute of Standards and Technology*) desarrolló un estándar aplicado a *Smart Grids* que establece al protocolo IP como el protocolo que deben utilizar los dispositivos como *smart meters*, aplicaciones inteligentes en el hogar y concentradores de datos, además de fomentar el uso de IPv6 en su desarrollo y despliegue [Nat10].

El grupo Elster¹⁶ definió una arquitectura para redes de área vecina o FAN (*Field Area Network*) utilizando IPv6 para la distribución eficiente de energía. La “Arquitectura IPv6 Estandarizada y Flexible para Redes de Área de Campo” de la Figura 16, presenta aplicaciones no solo en la gestión del consumo de la electricidad, sino que también provee aplicaciones avanzadas como la oportunidad para el usuario de optimizar su consumo de energía basado en el uso de la información en tiempo real de los precios de la electricidad, la monitorización del estado y el control de la red eléctrica, detección y aislamiento de errores que sirven para las futuras centrales de energía virtuales [IPv11].

La arquitectura esta compuesta por una colección de *smart meters*, distribuidos e interconectados en una red de bajo consumo y de pérdida, un conjunto de *edge routers*

¹⁶<http://www.elster.com/>

para comunicación bidireccional desde y hacia los *smart meters* y una WAN (*Wide Area Network*) que permite la comunicación con los proveedores de servicio. Además, la infraestructura IPv6 permite gran cantidad de direcciones para el despliegue de los medidores, capacidad de incorporar nuevos elementos, flexibilidad en la configuración, es un protocolo abierto que utiliza una adaptación de 6LoWPAN, junto con considerar aspectos de ruteo para LLNs (*Low Power and Lossy Network*) [Els11].

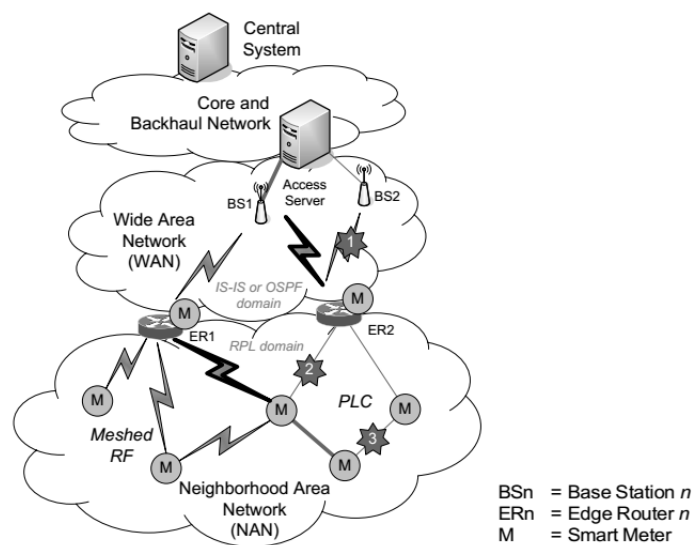


Figura 16. Infraestructura de red inteligente de Elster Group

JenNet-IP

La empresa NXP Semiconductors en el año 2011 realizó una demostración de control doméstico utilizando un protocolo básico 6LoWPAN, donde un conjunto de objetos de uso cotidiano, como lámparas fluorescentes, ampolletas de LED, enchufes inteligentes y un monitor, cada uno con su propia dirección IP, son monitoreados y controlados a través de una aplicación móvil diseñada para *tablet*, encontrándose conectados a través de una *gateway* [NXP11]. De esta forma, el producto JenNet-IP representa una solución que utiliza una capa de red 6LoWPAN optimizada, dirigida a ultra bajo consumo de energía, para redes residenciales e industriales, tal como se observa en la Figura 17.

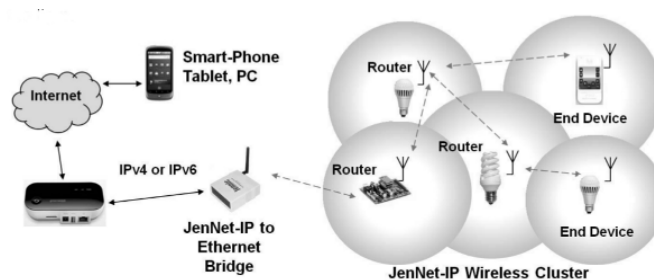


Figura 17. Diagrama de JenNet-IP

Flexibity

Flexibity Internet Sensors¹⁷ usa los protocolos definidos para el Internet de las Cosas, con énfasis en productos de uso libre y software de código abierto como Linux / OpenWRT y Contiki OS. En este sentido, son provistas plataformas para redes domésticas y vigilancia ambiental, donde cada sensor tiene una dirección IPv6 única, pudiendo accederse desde cualquier lugar utilizando el protocolo HTTP (*Hypertext Transfer Protocol*) convencional, o bien, integrar con otros servicios web, como Twitter.

¹⁷<http://www.flexibity.com/>

3. SOLUCIÓN PROPUESTA

En la extensión del capítulo se aborda la problemática que el presente proyecto pretende dar solución, estableciendo un contexto frente al cual hacer realidad el Internet de las Cosas implementando 6LoWPAN y proponiendo una solución para la situación dada, que contempla hardware y software, además de establecer la metodología de trabajo adecuada para llevarla a cabo.

3.1. Definición de la problemática

El Internet de las Cosas hace referencia a una red de objetos de uso cotidiano interconectados entre sí. En este sentido, la convivencia con cosas conscientes de sí mismas posibilita el desarrollo de ciudades inteligentes, monitoreo de la salud de las personas, uso eficiente de la energía, aplicaciones para el hogar, entre otras. Sin embargo, lo que suscita la atención del presente proyecto se encuentra relacionado con el monitoreo de la salud estructural de puentes, de la mano con un “Sistema de Gestión de Puentes” o BMS (*Bridge Management System*), para así alimentarlo con información sobre las condiciones de la estructura.

De esta forma, a través de la implementación de un escenario de red 6LoWPAN se busca convertir en una realidad al IoT por medio de un prototipo que posea la pila de protocolos necesaria para establecer comunicación IP, proporcionando información al Sistema de Gestión de Puentes.

3.1.1. Sistema de Gestión de Puentes

Dado el contexto, resulta pertinente mencionar que según un informe de la OECD (*Organisation for Economic Co-operation and Development*) de 1981, se define a los Sistemas de Gestión de Puentes o BMS como una herramienta para ayudar a las agencias de puentes y carreteras en la elección de las mejoras óptimas para la red de puentes, y que son compatibles con las políticas de la organización, objetivos a largo plazo y restricciones de presupuesto [Akg07].

Un BMS también puede ser entendido como un proceso sistemático para la realización

de mantenencias, rehabilitación y decisiones de reemplazo en un determinado conjunto de puentes, junto con el análisis de la información de los mismos, y sujeto a los fondos disponibles para preservar dicho sistema de puentes [Mol12].

Dentro de los alcances que posee un BMS, es importante destacar la importancia de la información contenida en la base de datos (inventario de puentes), ya que esta, genera una importante retroalimentación para futuros diseños, componentes y sistemas constructivos de las estructuras analizadas. Además, la información es de gran utilidad para la planificación y control de las mantenencias de los puentes.

Los BMS van más allá de inventarios con información básica de las estructuras, incluyendo también algoritmos que procesan y generan resultados útiles para los administradores de puentes, como sistemas con modelos de deterioro y planes de información financiera.

Idealmente, un Sistema de Gestión de Puentes esta compuesto por distintos módulos que pueden variar de acuerdo a los requerimientos de las agencias encargadas de la administración y pueden o no estar integrados entre sí, no obstante, un sistema básico se maneja bajo el esquema de la Figura 18, donde los módulos primordiales son: Inventario, Inspección, Condición del puente, Costos y Opciones de mantención [Mol12].

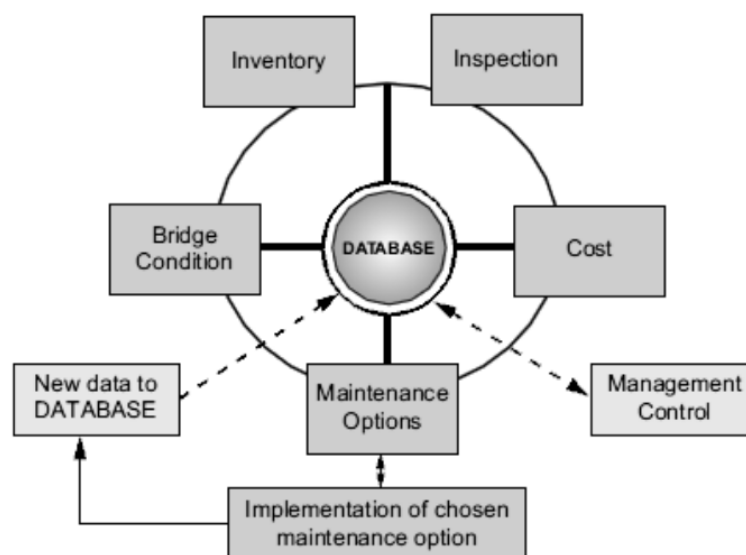


Figura 18. Módulos básicos de un Sistema de Gestión de Puentes

Un BMS contiene este registro inicial con datos básicos acerca de los puentes en una base de datos actualizada de forma regular, con el fin de también recolectar información sobre

las inspecciones realizadas en orden de evaluar las condiciones de las estructuras y su tasa de deterioro. Además, la información que es recabada según los procedimientos descritos con anterioridad junto con otros datos son usados para evaluar y priorizar el financiamiento de los puentes para su reparación y/o mantenimiento [Rya01].

Al revisar los Sistemas de Gestión de Puentes implementados en la actualidad, PONTIS y BRIDGIT son los dos BMS más utilizados en Norteamérica, mientras que en Alemania los análisis y procedimientos de evaluación se hacen de acuerdo a los resultados de inspecciones según la norma DIN 1076 (*Deutsches Institut für Normung*). En Chile existen propuestas de sistemas privados de inventarios de puentes y otras como MAPRA (Mantenimiento de Puentes de la Red Vial Austral), el cual se encuentra aún inconcluso y falta desarrollo en términos de evaluación y priorización de estructuras [Esh].

Sin embargo, para alcanzar su objetivo, los Sistemas de Gestión de Puentes requieren de procedimientos que aseguren que los puentes son inspeccionados y evaluados regularmente, de modo que se lleve una mantención apropiada y se mantenga una condición adecuada a lo largo de la vida útil de la estructura.

En el contexto local, el Instituto de Informática en conjunto con Obras Civiles en la Universidad Austral de Chile (UACH), se encuentran desarrollando un Sistema de Gestión de Puentes (Figura 19).



Figura 19. Login Sistema de Gestión de Puentes UACH

El BMS desarrollado por la Universidad Austral se compone de cinco módulos básicos:

- Ver Puentes: inventario
- Evaluar Estado de Conservación: inspección y condición del puente
- Determinar Acciones de Conservación: opciones de reparación y costos
- Gestión de Conservación: priorización
- Análisis y Estadística: control de información

Esquemáticamente, el BMS se representa en la Figura 20, donde el núcleo del sistema está compuesto por la base de datos, alrededor de la cual funcionan los demás componentes.



Figura 20. Módulos de Sistema de Gestión de Puentes de UACH

De esta forma, dada la problemática que plantea la inspección y evaluación de las estructuras para un Sistema de Gestión de Puentes, el Internet de las Cosas puede contribuir a proporcionar información en tiempo real sobre las condiciones de los puentes a través de un monitoreo continuo.

3.1.2. Monitoreo continuo

Dado un contexto, como un Sistema de Gestión de Puentes, sobre el cual un dispositivo pueda brindar información que contribuya a la toma de decisiones, surgen situaciones que ameritan atención, para así amortiguar posibles efectos no deseados.

En este sentido, el monitoreo continuo llevado a cabo por dispositivos embebidos con sensores para determinadas variables acarrea restricciones concernientes a las características de dichos dispositivos en términos de: ejecución de un número limitado de operaciones, bajo consumo energético y bajo uso del ancho de banda. Para este escenario, la arquitectura de la red trae implicancias para el ruteo de los dispositivos y descubrimiento de vecinos, considerando también la convergencia que debe existir entre los protocolos de comunicación de los dispositivos (IEEE 802.15.4) y el protocolo IP a través de esquemas de compresión y descompresión de cabeceras normados por la IETF (*Internet Engineering Task Force*), como 6LoWPAN.

Sin embargo, no solo los dispositivos finales reclaman atención, sino que también los dispositivos de comunicación y de almacenamiento, debido a los grandes volúmenes de datos que proporcionaría una red de sensores distribuida en una red de puentes, por ejemplo. En este contexto, el término “*Big Data*” hace referencia a dicha problemática, la que si bien no es el objetivo del presente proyecto, se alza como una tangente a considerar de la mano del Internet de las Cosas.

3.2. Metodología de trabajo

La elección adecuada de la metodología de trabajo contribuye a orientar el proyecto y asegurar que cada paso se acerque más a la consecución de los objetivos planteados.

En este escenario, la investigación, desarrollo e implementación de tecnologías nuevas plantea desafíos a considerar. De esta forma, los requerimientos dependen del tipo de hardware y software a utilizar, siendo necesario evaluar diferentes alternativas para conseguir un resultado adecuado. Dado este sentido, el problema se puede dividir en dos partes:

- Investigación: aplicación del método científico para abordar tecnologías que sustentan la nueva Internet
- Desarrollo: adopción de una metodología ágil como Scrum para el desarrollo de la solución

Para cumplir con el logro de los objetivos establecidos, en el proyecto se aborda el problema desde dos perspectivas, persiguiendo en primera instancia la búsqueda de una

solución adecuada que logre satisfacer las demandas, por medio de la experimentación con las tecnologías que se verán involucradas en la nueva Internet. Posteriormente, una vez encontrada la solución, se procederá a trabajar con una metodología ágil de software, en este caso Scrum, la cual proporciona resultados en el corto plazo y favorece un ambiente donde los requisitos del proyecto cambien a menudo, tal como es el caso de la prueba de nuevas tecnologías.

3.3. Aplicación del método científico

Dada la componente de investigación que se remarca en el presente proyecto, la adopción de una actitud acorde a los desafíos planteados por la nueva Internet resulta plasmada en el desarrollo de una solución a través del método científico para encontrarla, siguiendo el modelo de la Figura 21.

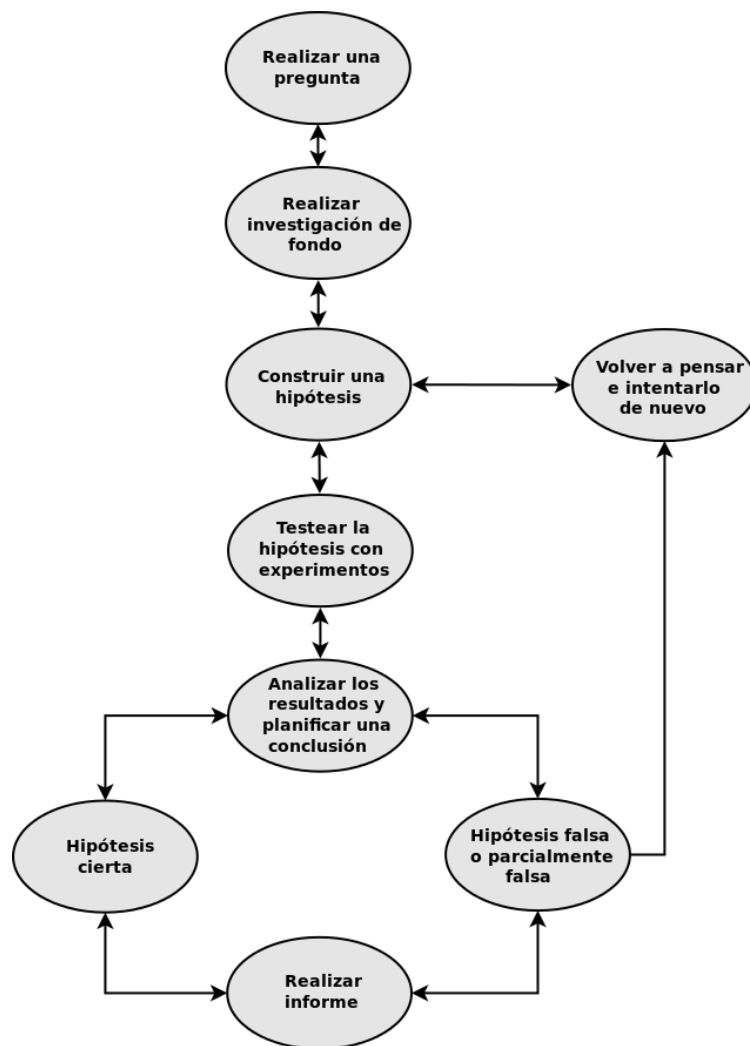


Figura 21. Modelo simplificado para el método científico

El establecimiento de un método de investigación pretende generar conocimiento con

respecto a visiones que se alzan para convertirse en una realidad para la concreción de la nueva red, desde una formada por ordenadores a otra formada por objetos. Por este motivo, el trabajo se orienta hacia la práctica y hacia la medición, sujeto a pruebas de razonamiento. Para abordar la problemática planteada se hace uso de métodos que se aproximan al conocimiento del objeto de estudio mediante su conocimiento directo y uso a partir de la experiencia, a través de la observación y la experimentación [Och].

De esta forma el proceso de búsqueda de una solución se puede dividir, a grandes rasgos, en las etapas que se exponen en los apartados siguientes.

3.3.1. Observación

En primera instancia, el proceso de observación apunta a examinar atentamente los hechos y fenómenos que tienen lugar en el escenario del Internet de las Cosas y que pueden ser percibidos por los dispositivos de comunicación, almacenamiento y sensores.

En este contexto, el objeto de estudio del presente proyecto lo constituye la aplicación de 6LoWPAN (*IPv6 over Low-Power Wireless Personal Area Networks*) a un contexto en específico, tal como lo es un Sistema de Gestión de Puentes; cabe destacar la existencia de protocolos propietarios que resuelven problemáticas en determinados contextos, como ZigBee, Z-Wave o HomePlug. Por este motivo, resulta importante observar los actores del actual ecosistema de red, los protocolos que se ven involucrados en el traspaso de información, tales como IPv6 y UDP, y los requerimientos en términos de recursos que exige este nuevo escenario, junto con las problemáticas que actualmente se plantean, como el ruteo en redes de bajo consumo y pérdida, el almacenamiento masivo de grandes volúmenes de data y la convergencia entre el protocolo IP y estándares como el IEEE 802.15.4, con bajas tasas de transmisión de datos y poco uso de energía, lo cual suscita la atención del presente proyecto.

De esta forma, si se observa la inminente adopción de IPv6 debido al agotamiento de su predecesor IPv4, pasando desde los 32 bits hasta los 128 bits, se instaura la posibilidad de conectar trillones de dispositivos que estarán bajo las condiciones descritas en el párrafo anterior. Por lo tanto, si son medidos tiempos de respuesta de los dispositivos de comunicación y revisadas las implementaciones existentes (totales y parciales),

considerando sus limitaciones en términos de recursos, licenciamiento y disponibilidad, es posible comprender la importancia de la adopción de protocolos de Internet como 6LoWPAN, debido a la integración con redes más grandes o con servicios existentes, la estabilidad y escalabilidad del protocolo IP, además de su arquitectura en capas que proporciona un amplio nivel de flexibilidad para la innovación.

3.3.2. Formulación de hipótesis

Luego de las observaciones efectuadas, se debe plantear el cómo y el porqué de lo ocurrido, para así formular una hipótesis. Lo que se busca es elaborar una explicación provisional de los hechos observados junto con sus posibles causas.

Por este motivo, dada la existencia de protocolos propietarios y los beneficios subyacentes de protocolos abiertos, y considerando la gran cantidad de limitaciones y problemas planteados hasta la actualidad, resulta pertinente manifestar que: “Una solución para el Internet de las Cosas exige el uso de estándares como 6LoWPAN para explorar las distintas características del esquema e introducir cambios, además de utilizar también alternativas hardware y software abiertas y apegadas a la normalización existente debido a la flexibilidad que entregan”.

3.3.3. Experimentación

El paso siguiente se orienta hacia comprobar si el uso de alternativas abiertas entrega efectivamente o no la flexibilidad y estabilidad planteada, realizando para ello pruebas sobre diversas opciones.

Para experimentar es pertinente modificar las circunstancias para observar el comportamiento del hardware y software probados.

En este caso, la existencia de sistemas operativos abiertos para redes de sensores como Contiki o TinyOS aún se encuentra limitado a un número pequeño de plataformas hardware, mientras que la adopción masiva de 6LoWPAN aún no se concreta, tal como aún no se posiciona con fuerza el uso de IPv6. Para este contexto, en el actual proyecto son tratadas las alternativas de la Tabla 6 referentes al uso de simuladores como Cooja de

Contiki, la implementación del stack en un dispositivo de comunicación Linux con una distribución como DD-WRT u OpenWRT, y el uso de sensores en plataformas abiertas como Arduino¹⁸ para implementar directamente la pila de protocolos requerida en ellos.

Tabla 6. Alternativas para solución

	Alternativa 1	Alternativa 2	Alternativa 3
Características	Uso de Simulador Cooja	Uso de un router con una distribución de Linux	Uso de librerías de Arduino
Hardware	Ninguno	router Linksys WRT54G / router TP-LINK TL-MR3220 / sensor ZigBee	Arduino Mega / antena XBee / sensor para Arduino
Software	Máquina Virtual Contiki 2.6	DD-WRT u OpenWRT	IDE Arduino

Para el primer caso la opción del simulador Cooja, que se muestra en la Figura 22, permite realizar pruebas de red emulando nodos y posibilitando la inspección del comportamiento del sistema. Sin embargo, el simulador no satisface los objetivos planteados ya que no hace posible el uso de 6LoWPAN en una estructura como un puente directamente, aunque si permite posicionar a Contiki como una alternativa altamente confiable, debido a que permite la experimentación de diferentes escenarios y la modificación de su código fuente para diferentes plataformas.

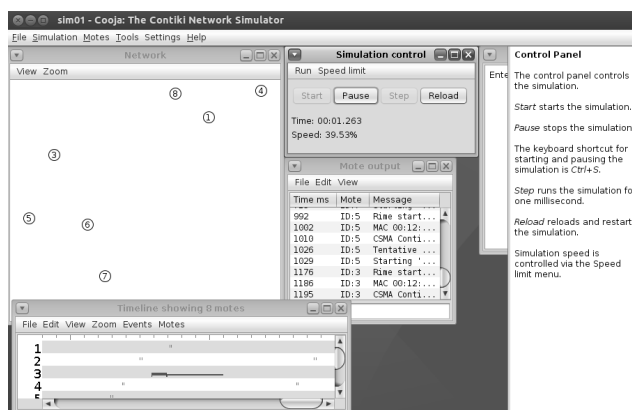


Figura 22. Simulador Cooja de Contiki

¹⁸<http://www.arduino.cc/>

Otra alternativa puesta sobre la palestra la constituye la utilización de plataformas hardware abiertas Arduino como las de la Figura 25, para no solo implementar los sensores, sino que también al dispositivo de comunicación. En este sentido, el uso de aparatos de comunicación XBee de Digi¹⁹ es común en plataformas Arduino, presentando la particularidad de operar con el estándar IEEE 802.15.4 necesario para la implementación de 6LoWPAN. Entonces, las dificultades radican en los recursos disponibles (memoria y procesador) para llevar a cabo la tarea, aunque, destaca notablemente la existencia de una librería provista por Telecom Bretagne de Francia [Tel13] que fusiona la versión 2.5 de Contiki con una librería para el dispositivo XBee junto con modificaciones introducidas por los autores, permitiendo contar con un verdadero sistema operativo en nodos y dispositivos de comunicación, y que además posee la implementación para IPv6 denominada μ IPv6 con soporte para 6LoWPAN.

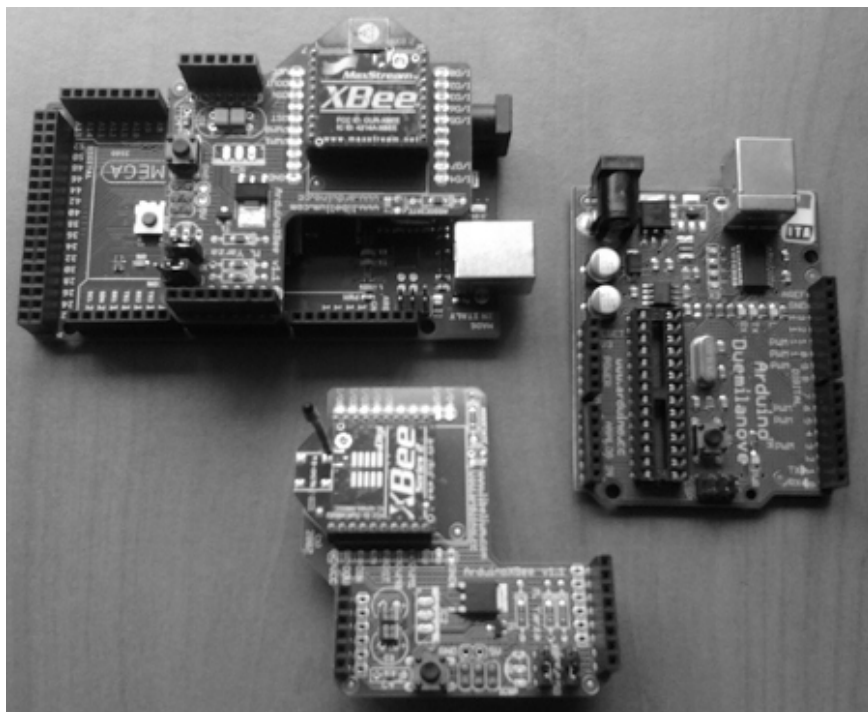


Figura 25. Placas Arduino

De cara a las pruebas realizadas, se recorre un espectro acotado por disponibilidad de recursos hardware y por las mismas condiciones del tema a investigar dado su actual estado de desarrollo, abarcando de esta manera a las alternativas que pueden dar pie al desarrollo de la solución final acorde a los objetivos, experimentando de primera mano con ellas.

¹⁹<http://www.digi.com/>

3.3.4. Emisión de conclusiones

Dado el desarrollo actual de las tecnologías que harán posible el Internet de las Cosas o IoT, y considerando la normalización existente, el uso de alternativas abiertas favorece la experimentación, sin embargo, al trabajar con tecnologías nuevas se requiere alta comprensión de los protocolos implicados en la nueva Internet. En este escenario, 6LoWPAN es el componente clave para el Internet del futuro, para que así los objetos cuenten con el protocolo IP habilitado para ser parte integral de la red.

En este sentido, valiéndose de las pruebas efectuadas, destaca la alternativa provista por Arduino debido a la facilidad para conseguir los módulos necesarios para construir la solución. Además, entre las tres alternativas, su utilización se aproxima fehacientemente a la proyectada por el IoT, dado el desarrollo sobre aparatos que presentan restricciones en término de recursos, pero que, sin embargo, pueden gozar de cierta autonomía logrando establecer monitoreo de puentes en terreno para el Sistema de Gestión de Puentes a nivel de prototipo.

De esta manera, la utilización de sensores con restricciones de recursos que implementen directamente 6LoWPAN brinda una aproximación real del ecosistema de red que se apronta para ser una realidad en los próximos años, permitiendo automatizar y/u optimizar ciertos procesos que lleva a cabo el Sistema de Gestión de Puentes actualmente.

3.4. Definición de la solución

La inspección y evaluación de las estructuras se puede mejorar a través del monitoreo continuo de diversos aspectos de las mismas, velando por la mantención de la salud estructural de los puentes a través de la implementación de un escenario de red 6LoWPAN, alimentando con volúmenes de datos al Sistema de Gestión de Puentes desarrollado por la Universidad Austral de Chile, y haciendo posible la puesta en marcha de un prototipo con la tecnología que sustenta el Internet de las Cosas.

De esta manera, el uso de dispositivos Arduino como nodos finales provistos de sensores, y como dispositivos de comunicación, se alza como una alternativa rentable a nivel de prototipo, ya que emula las condiciones reales de los objetos de la nueva Internet en

cuanto a memoria y capacidad de procesamiento limitada, bajo consumo de energía y baja utilización del ancho de banda, para así lograr realizar pruebas en terreno con el fin de capturar variables que contribuyan a la mantención de las estructuras.

3.4.1. Arquitectura de la solución

El escenario de red que se pretende implementar convive directamente con el Sistema de Gestión de Puentes de la UACH descrito en los apartados anteriores. La Figura 26 muestra la arquitectura del sistema, compuesto por un Servidor Web y una Base de Datos que aloja información acerca de todos los puentes de la Región de los Ríos; un sensor posicionado en un puente del sistema se comunican de forma inalámbrica con un router frontera también ubicado en el puente y con conectividad IP, que actúa como *gateway*, permitiéndo comunicación entre redes IP y el IEEE 802.15.4.

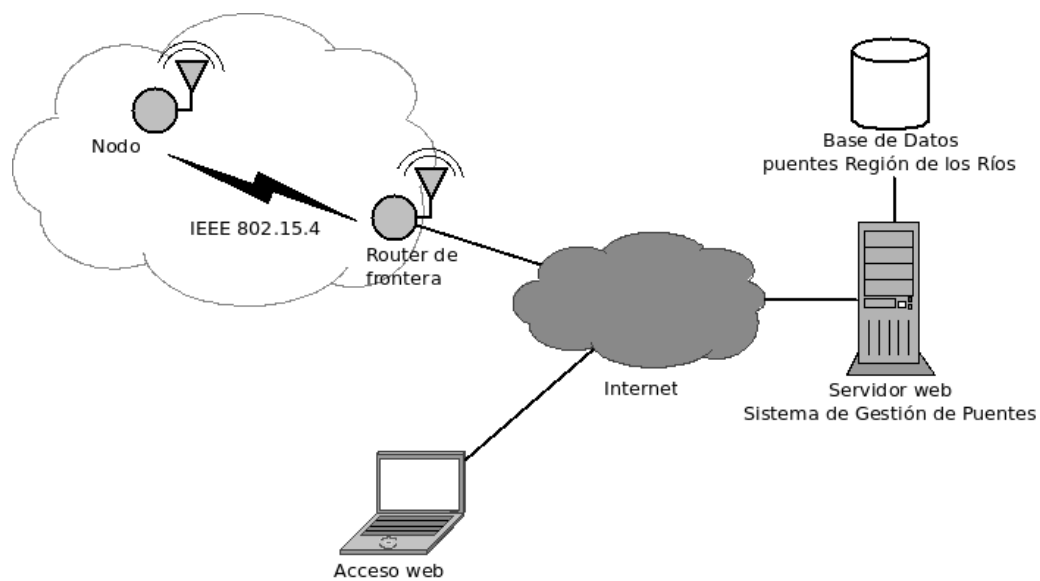


Figura 26. Arquitectura de la solución

3.4.2. Tecnologías involucradas

La infraestructura del Servidor de desarrollo del Sistema de Gestión de Puentes perteneciente al Ministerio de Obras Públicas (MOP) alojado en dependencias del Instituto de Informática de la UACH, posee el Sistema Operativo Ubuntu 11.04 (GNU/Linux 2.6.38-8-generic-pae i686), donde se encuentra instalada la infraestructura web LAMP (Linux + Apache + MySQL + PHP) con las siguientes versiones de software:

- Apache Server: Apache/2.2.17

- MySQL: Ver 14.14 Distrib 5.1.63
- PHP: 5.3.5-1ubuntu7.11

La solución planteada requiere de un sistema de almacenamiento que conviva con la información de los puentes. De esta manera se cuenta con un sistema de gestión de Base de Datos relacional MySQL, la cual posee características multihilo, multiusuario y de código abierto; es utilizado el mecanismo de almacenamiento InnoDB para soportar transacciones ACID (*Atomicity, Consistency, Isolation, Durability*), bloqueo de registros e integridad referencial. Cabe destacar, que si bien es cierto, el almacenamiento masivo de grandes volúmenes de data trae implicancias sobre los dispositivos de comunicación y de almacenamiento, una base de datos relacional puede no ser la solución adecuada, debiendo recurrir a Bases de Datos NoSQL o a algún *framework* que soporte aplicaciones distribuidas como Hadoop [Cis11]; sin embargo, el objetivo del presente proyecto radica en el uso de 6LoWPAN para establecer monitoreo de la salud estructural de puentes a nivel de prototipo, por lo que la utilización de MySQL se deriva de su uso en el Sistema de Gestión de Puentes, presentándose la oportunidad de evaluar el comportamiento de un esquema relacional sometido a las condiciones anteriormente expuestas.

Desde la perspectiva del escenario de red 6LoWPAN, el Nodo sensor y dispositivo de comunicación (Router) será construido a partir de la plataforma Arduino, siendo requerido el IDE de Arduino 1.0 y la librería Arduino-IPv6 proporcionada por Telecom Bretagne [Tel13], programando un *sketch* para ser cargado en el microprocesador de la placa, utilizando el lenguaje de programación de Arduino basado en C/C++. Los módulos XBee Series 1 proporcionan comunicación bajo la norma IEEE 802.15.4 para Arduino.

Además, se contempla la utilización del lenguaje de programación Python para interactuar a bajo nivel con los puertos que se conectan a los dispositivos Arduino, así como también con la base de datos, capturar paquetes de data y enviar solicitudes HTTP, desarrollando una “Sonda” para la red; Python hace incapié en una sintaxis limpia y elegante [Zia08].

También se plantea la utilización de la base de datos en línea “Xively” para almacenar información del muestreo de sensores en la web por medio de la sonda mencionada anteriormente.

4. ANÁLISIS Y DISEÑO

El capítulo que se extiende a continuación gira en torno al modelamiento de la solución propuesta según la metodología de software seleccionada, utilizando los artefactos de software pertinentes para documentar y brindar mantenibilidad al sistema, con el fin de implementar posteriormente un prototipo funcional que de respuesta al problema expuesto de acuerdo a los objetivos planteados.

Para este escenario se hará uso de UML (*Unified Modeling Language*), una notación esquemática en su mayor parte, con que se construyen sistemas por medio de conceptos. De esta forma para crear una aplicación de software hay que describir el problema y las necesidades o requerimientos (en qué consiste el problema y qué debe hacerse), centrando el análisis en una investigación del problema y no en la manera de definir la solución; también, para la etapa de diseño, es necesario contar con descripciones detalladas de alto nivel de la solución lógica, junto con saber cómo satisface los requerimientos [Lar99].

4.1. Metodología de software

Una metodología de desarrollo de software se puede comprender como un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo de sistemas. Una metodología tiene como objetivo presentar un conjunto de técnicas de modelado de sistemas que posibiliten desarrollar software de calidad; para lograr su cometido se describen herramientas de análisis y diseño UML: diagramas, especificaciones y criterios de aplicación.

Para el desarrollo de la solución del presente proyecto se utilizará el marco de trabajo Scrum, tal como ha sido señalado en el apartado acerca de la metodología de trabajo expuesto en el capítulo anterior.

Dadas las características de la problemática planteada, referentes principalmente a la investigación y experimentación con nuevas tecnologías, se ha optado por un entorno de desarrollo ágil de software, considerando la inestabilidad del entorno bajo el cual se trabaja. El proyecto se construye al mismo tiempo que se modifica y aparecen nuevos requisitos. Por este motivo, se tiene en cuenta el valor de la innovación que se necesita,

junto con la velocidad de desarrollo requerida.

De esta manera, a través de Scrum se plantea que tal vez no existen los productos finales, sino que productos en evolución, mejora o incremento continuo desde la primera versión beta. La gestión de proyectos ágil no se formula sobre la necesidad de anticipación, sino sobre la de adaptación continua [Scr13].

Siguiendo en esta línea, un implantación estándar de Scrum:

- Elementos: pila del producto, pila del *sprint* e incremento.
- Reuniones: de inicio del *sprint*, revisión diaria, revisión del *sprint* y retrospectiva.
- Roles: propietario del producto, equipo, *scrum master* y resto de implicados.

La pieza clave la constituye el *sprint*, el cual es entendido como cada ciclo o iteración de trabajo que produce una parte del producto terminada y funcionalmente operativa (incremento). Además, el equipo de desarrollo actualiza a diario un gráfico de avance o *burn down* para comprobar su estado. La Figura 27 ilustra el ciclo iterativo Scrum basado en pulsos de tiempo (*time boxing*) [Scr13].

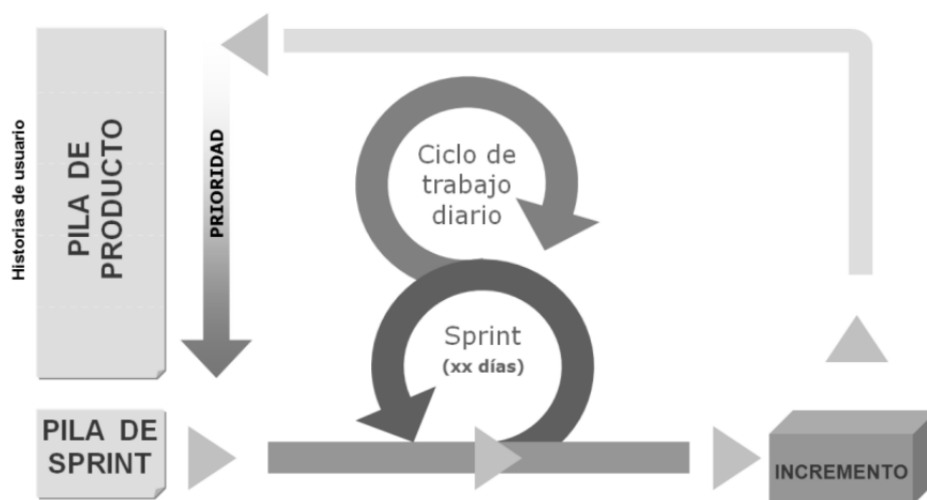


Figura 27. Diagrama del ciclo iterativo Scrum

Considerando los elementos del marco estándar de Scrum, es pertinente definir:

- Pila del producto (*product backlog*): lista de requisitos de usuario que a partir de la visión inicial del producto crece y evoluciona durante el desarrollo.

- Pila del sprint (*sprint backlog*): lista de los trabajos que debe realizar el equipo durante el *sprint* para generar el incremento previsto.
- *Sprint*: nombre que recibe cada iteración de desarrollo. Es el núcleo central que genera el pulso por tiempos prefijados (*time boxing*).
- Incremento: resultado de cada *sprint*.

4.2. Requisitos

En los proyectos predictivos los requisitos del sistema suelen especificarse en documentos formales, mientras que en los proyectos ágiles toman forma de la pila del producto o lista de historias de usuario. De esta manera, Scrum emplea dos formatos para registrar los requisitos: pila del producto (*Product Backlog*) y pila del *sprint* (*Sprint Backlog*).

La pila del producto es el inventario de funcionalidades, mejoras, tecnología y corrección de errores que deben incorporarse al producto a través de los sucesivos *sprints*; representa todo aquello que esperan el cliente, los usuarios y en general todos los interesados, encontrándose en continuo crecimiento y evolución [Scr13]. En la Tabla 7 se puede observar la pila de producto del proyecto, la cual incluye: un identificador único de la funcionalidad, sistema de priorización, descripción (historia de usuario) y una estimación del esfuerzo en horas; los *sprints* de 40 horas se ubican en las Tablas 8, 9, 10, 11 y 12.

Tabla 7. Product Backlog

ID	Prioridad	Descripción	Est.(horas)
r1	Muy alta	Instalar el ambiente de desarrollo	4
r2	Muy alta	Configurar módulos XBee	4
r3	Alta	Establecer comunicación entre dispositivos Arduino con módulos XBee	8
r4	Muy alta	Configurar librería Arduino-IPv6	24
r5	Alta	Cargar librería Arduino-IPv6 en dispositivos Arduino	4
r6	Media	Establecer comunicación con librería Arduino-IPv6 en dispositivos Arduino con módulos XBee	24
r7	Media	Integrar un sensor en Nodo	12

Tabla 7 (continuación)

ID	Prioridad	Descripción	Est.(horas)
r8	Muy alta	Desarrollar <i>sketch</i> de Arduino para el dispositivo de comunicación (Router)	16
r9	Muy alta	Desarrollar <i>sketch</i> de Arduino para el Nodo con sensor integrado	20
r10	Muy alta	Establecer comunicación entre Nodo y Router con sensor integrado	4
r11	Alta	Desarrollar <i>script</i> para Sonda que capture paquetes	24
r12	Alta	Desarrollar modelo de datos para Sensor en Sistema de Gestión de Puentes	4
r13	Alta	Desarrollar módulo para Sonda que envíe información a la Base de Datos del Sistema de Gestión de Puentes	8
r14	Alta	Establecer comunicación de Nodo con la Base de Datos del Sistema de Gestión de Puentes	4
r15	Baja	Configurar Xively	4
r16	Baja	Desarrollar módulo para Sonda que envíe información a Xively	8
r17	Baja	Establecer comunicación de Nodo con Xively	4
r18	Media	Desarrollar vista en Sistema de Gestión de Puentes para observar resultados	16

Tabla 8. Sprint 1

ID	Tarea	Estimado (horas)	Real (horas)
r1	Instalar el ambiente de desarrollo	4	2
r2	Configurar módulos XBee	4	3
r3	Establecer comunicación entre dispositivos Arduino con módulos XBee	8	6
r4	Configurar librería Arduino-IPv6	24	32
		TOTAL	43

Tabla 9. Sprint 2

ID	Tarea	Estimado (horas)	Real (horas)
r5	Cargar librería Arduino-IPv6 en dispositivos Arduino	4	2
r6	Establecer comunicación con librería Arduino-IPv6 en dispositivos Arduino con módulos XBee	24	32
r7	Integrar un sensor en Nodo	12	8
		TOTAL	42

Tabla 10. Sprint 3

ID	Tarea	Estimado (horas)	Real (horas)
r8	Desarrollar <i>sketch</i> de Arduino para el dispositivo de comunicación (Router)	16	8
r9	Desarrollar <i>sketch</i> de Arduino para el Nodo con sensor integrado	20	24
r10	Establecer comunicación entre Nodo y Router con sensor integrado	4	4
		TOTAL	36

Tabla 11. Sprint 4

ID	Tarea	Estimado (horas)	Real (horas)
r11	Desarrollar <i>script</i> para Sonda que capture paquetes	24	20
r12	Desarrollar modelo de datos para Sensor en Sistema de Gestión de Puentes	4	4
r13	Desarrollar módulo para Sonda que envíe información a la Base de Datos del Sistema de Gestión de Puentes	8	8
r14	Establecer comunicación de Nodo con la Base de Datos del Sistema de Gestión de Puentes	4	3
		TOTAL	35

Tabla 12. Sprint 5

ID	Tarea	Estimado (horas)	Real (horas)
r15	Configurar Xively	4	2
r16	Desarrollar módulo para Sonda que envíe información a Xively	8	8
r17	Establecer comunicación de Nodo con Xively	4	2
r18	Desarrollar vista en Sistema de Gestión de Puentes para observar resultados	16	24
		TOTAL	36

Adicionalmente, la Tabla 13 presenta los requisitos funcionales del sistema.

Tabla 13. Requisitos funcionales del sistema

ID	Función	Categoría
rf1	Capturar información relevante para monitorear la salud estructural de un puente	Evidente
rf2	Utilizar comunicación inalámbrica para los dispositivos	Evidente
rf3	Registrar la información capturada en una Base de Datos	Oculto
rf4	Asignar a los dispositivos una dirección IPv6	Oculto
rf5	Utilizar el esquema 6LoWPAN para establecer comunicación IP en los dispositivos	Oculto
rf6	Desplegar información en el Sistema de Gestión de Puentes de la UACH	Evidente

4.3. Fase de Análisis

4.3.1. Descripción de las actividades del proceso

La descripción de las actividades del proceso de captura y almacenamiento de data efectuado por el Nodo (sensor) se lleva a cabo a través del diagrama de proceso de la Figura 28, donde es posible observar de manera gráfica las actividades referentes al sensor.

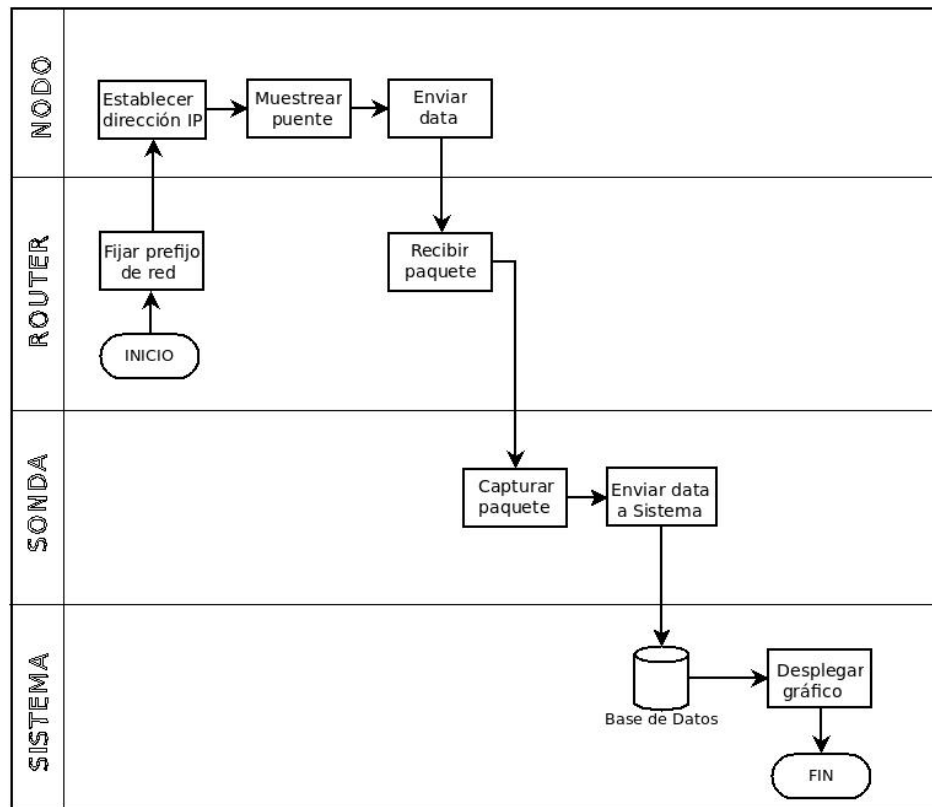


Figura 28. Diagrama de procesos

De esta forma, se establece un entorno de red IPv6 entre el Nodo y el Router, donde se utiliza el esquema 6LoWPAN, recurriendo a una Sonda para establecer la interacción con el Sistema de Gestión de Puentes, dadas las condiciones de prueba sobre la infraestructura actual de la red y la experimentación con tecnologías con el nivel de desarrollo planteado por la nueva Internet. El sensor que se ubica en el Nodo corresponde a un acelerómetro para medir vibraciones en la estructura.

4.3.2. Actores del sistema

A partir de las actividades efectuadas por los dispositivos involucrados se procederá a describir la secuencia de eventos de los actores (agentes externos) que utilizan el sistema para completar un proceso. Los actores identificados que estimulan al sistema con eventos son:

- Sensor: hace referencia al dispositivo que alimenta con data al Sistema de Gestión de Puentes.
- Usuario: hace referencia al usuario del Sistema de Gestión de Puentes, el cual podrá observar los valores de las variables registradas por el sensor.

En la Figura 29 se observa el diagrama de casos de uso del sistema.

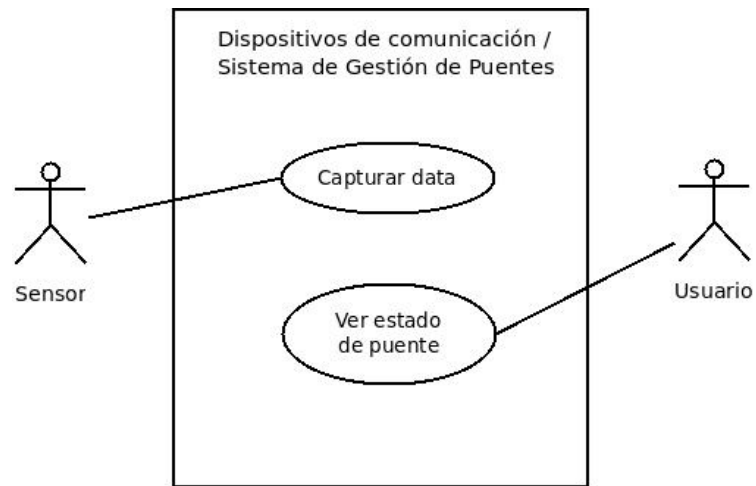


Figura 29. Diagrama de casos de uso

4.3.3. Casos de uso de alto nivel

Los procesos “Capturar data” y “Ver estado de puente” son descritos brevemente a través de casos de uso de alto nivel en las Tablas 14 y 15 respectivamente.

Tabla 14. Caso de uso de alto nivel “Capturar data”

Caso de uso:	Capturar data (CU01)
Actores:	Sensor
Tipo:	Primario y esencial
Descripción:	El Sensor posicionado en el puente, con una dirección IPv6, muestrea las variables requeridas para monitorear su salud estructural y enviar la data al Sistema de Gestión de Puentes.

Tabla 15. Caso de uso de alto nivel “Ver estado de puente”

Caso de uso:	Ver estado de puente (CU02)
Actores:	Usuario
Tipo:	Primario y esencial
Descripción:	El usuario con acceso al Sistema de Gestión de Puentes observa el estado del puente a través de gráficas en la ficha del puente.

4.3.4. Casos de uso expandidos

Los casos de uso expandidos “Capturar data” y “Ver estado de puente” son presentados en las Tablas 16 y 17 correspondientemente.

Tabla 16. Caso de uso expandido “Capturar data”

Caso de uso:	Capturar data (CU01)	
Actores:	Sensor	
Propósito:	Capturar información del estado del puente	
Resumen:	El Sensor en el puente captura data acerca de sus vibraciones y envía la información al Sistema de Gestión de Puentes	
Tipo:	Primario y esencial	
Referencias cruzadas:	rf1, rf2, rf3, rf4, rf5	
Precondiciones:	<ul style="list-style-type: none"> ■ El Sensor posee una dirección IPv6 ■ El Sensor posee implementado el esquema 6LoWPAN 	
Curso normal de los eventos		
Acción de los actores	Respuesta del sistema	
<ol style="list-style-type: none"> 1. Este caso de uso comienza cuando el Sensor crea una dirección de enlace local en la interface IPv6 habilitada (<i>Neighbor Discovery</i>) 3. El Sensor muestrea las vibraciones del puente en la componente vertical 4. El Sensor envía el tren de data capturada 	<ol style="list-style-type: none"> 2. El Sistema (Router) responde la solicitud asignando una subred 5. El Sistema almacena la información en la Base de Datos del Sistema de Gestión de Puentes 	
Cursos alternativos		
<ol style="list-style-type: none"> 2. El Sistema no responde la solicitud de Router del Sensor 4. El Sensor pierde información y envía el tren de la data rescatada 5. El Sistema pierde paquetes y almacena la información recepcionada en la Base de Datos del Sistema de Gestión de Puentes 		
Post condiciones:	Sistema queda en estado de espera para nuevo proceso	

Tabla 17. Caso de uso expandido “Ver estado de puente”

Caso de uso:	Ver estado de puente (CU02)	
Actores:	Usuario	
Propósito:	Observar el estado de conservación del puente	
Resumen:	El usuario ingresa al Sistema de Gestión de Puentes y observa gráficamente el estado de la estructura	
Tipo:	Primario y esencial	
Referencias cruzadas:	rf6	
Precondiciones:	<ul style="list-style-type: none"> ■ El Usuario posee credenciales de acceso para el Sistema ■ El puente posee sensores en su estructura 	
Curso normal de los eventos		
Acción de los actores	Respuesta del sistema	
<p>1. Este caso de uso comienza cuando el Usuario inicia sesión</p> <p>3. El Usuario ingresa al módulo “Ver puentes”, busca un puente e ingresa a su descripción</p> <p>5. El Usuario ingresa al apartado para observar la data capturada y ver el estado de conservación del puente</p>	<p>2. El Sistema despliega la vista del home del Sistema de Gestión de Puentes</p> <p>4. El Sistema despliega la ficha de identificación del puente</p> <p>6. El Sistema despliega gráficamente la data capturada por los sensores</p>	
Cursos alternativos		
3. El Usuario ingresa al módulo “Ver puentes”, busca incorrectamente un puente y el Sistema despliega el error		
Post condiciones:	Sistema queda en estado de espera para nuevo proceso	

4.3.5. Diagramas de secuencia

Los diagramas de secuencia de las Figuras 30 y 31 muestran gráficamente los eventos que fluyen desde los actores al sistema.

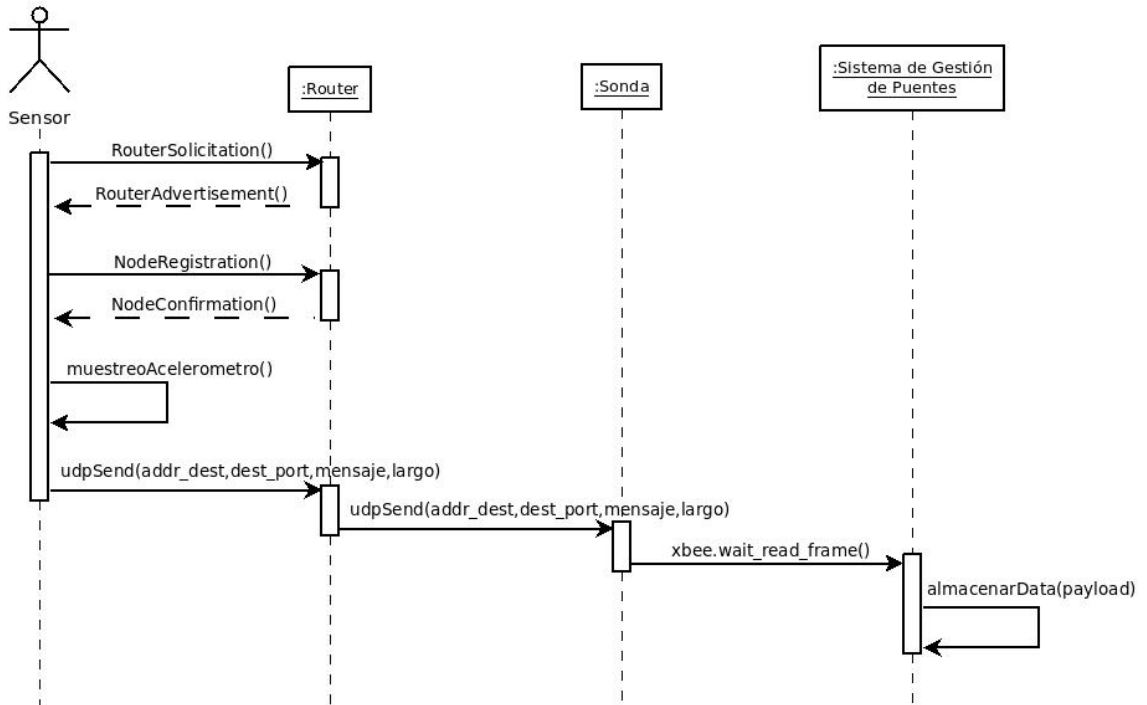


Figura 30. Diagrama de secuencia asociado a “Capturar data”

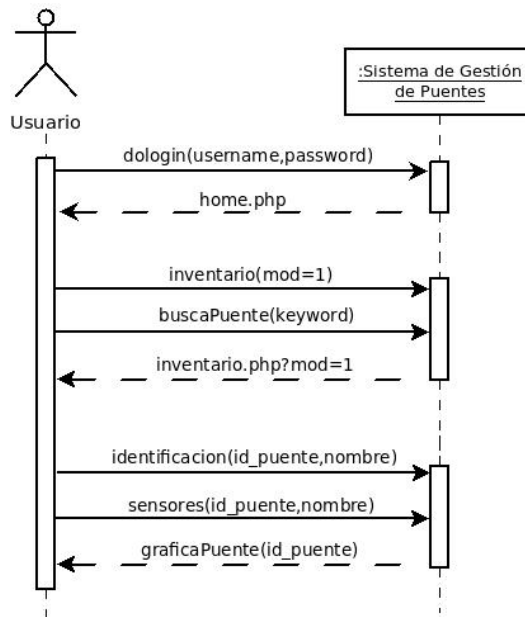


Figura 31. Diagrama de secuencia asociado a “Ver estado de puente”

4.4. Fase de Diseño

4.4.1. Casos de uso real

El diseño concreto del caso de uso “Ver estado de puente” es descrito en la Tabla 18 a partir de una tecnología particular de entrada y salida, así como de su implementación global.

Tabla 18. Caso de uso real “Ver estado de puente”

Caso de uso:	Ver estado de puente (CU02)
Actores:	Usuario
Propósito:	Observar el estado de conservación del puente
Resumen:	El usuario ingresa al Sistema de Gestión de Puentes y observa gráficamente el estado de la estructura
Tipo:	Primario y esencial
Referencias cruzadas:	rf6

Ventanas



Acción de los actores

- Este caso de uso comienza cuando el Usuario inicia sesión
- El Usuario ingresa al módulo “Ver Puentes” de la ventana 1, busca un puente en A e ingresa a su descripción en B
- El Usuario ingresa al apartado para observar la data capturada y ver el estado de conservación del puente en C en la ventana 2

Respuesta del sistema

- El Sistema despliega la vista del home del Sistema de Gestión de Puentes
- El Sistema despliega la ficha de identificación del puente
- El Sistema despliega gráficamente la data capturada por los sensores en D en la ventana 2

Cursos alternativos

- El Usuario ingresa al módulo “Ver puentes”, busca incorrectamente un puente y el Sistema despliega el error

4.4.2. Diagramas de colaboración

Las interacciones de las instancias del Nodo, al cual se le monta un sensor acelerómetro, son presentadas en la Figura 32.

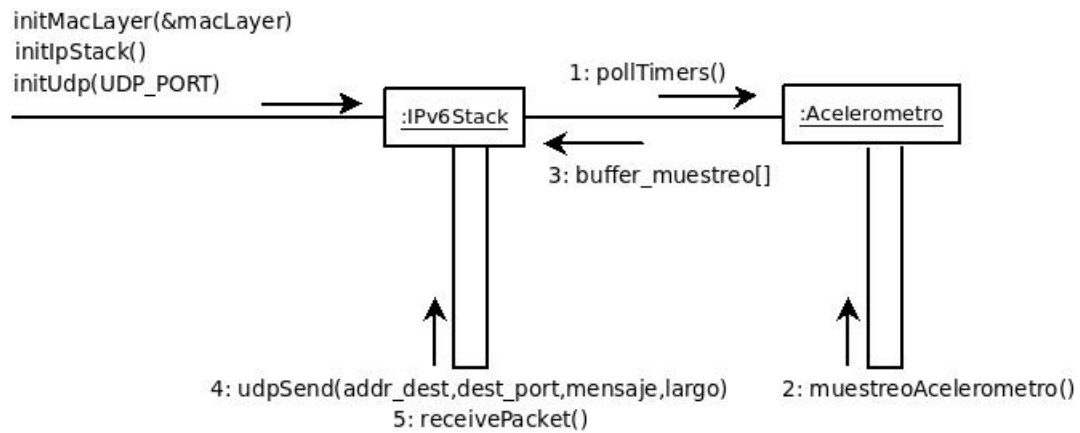


Figura 32. Diagrama de colaboración del Nodo

El Router posee las interacciones que se observan en la Figura 33, donde es presentada la implementación de las operaciones que lleva a cabo el dispositivo de comunicación en el escenario de red propuesto.

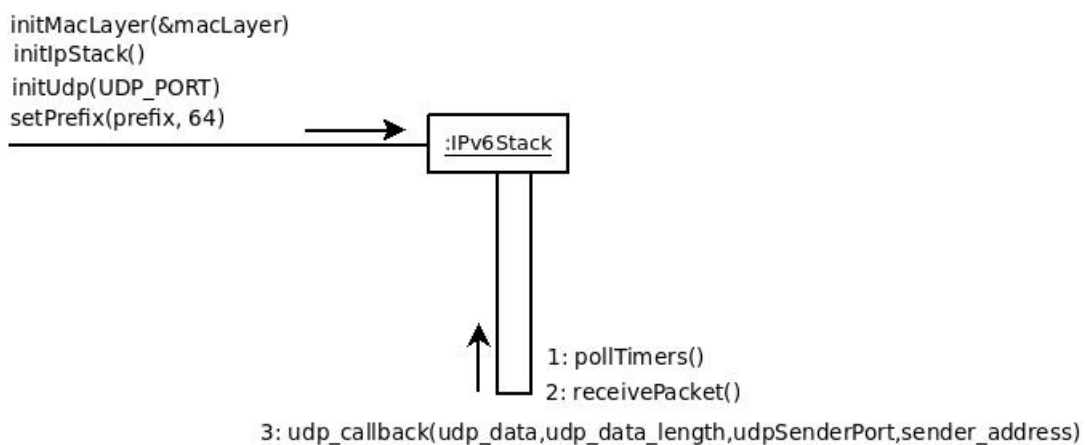


Figura 33. Diagrama de colaboración del Router

Dado este contexto, el Nodo y el Router componen un escenario de red IPv6-6LoWPAN, sin embargo, para el presente proyecto también es planteada comunicación IP entre el Nodo final y el Sistema de Gestión de Puentes, por lo tanto es habilitada una Sonda que permite el intercambio de información entre las tecnologías involucradas en el Internet de las Cosas, la cual posee las interacciones de la Figura 34, capturando paquetes que son intercambiados entre el Nodo y el Router.

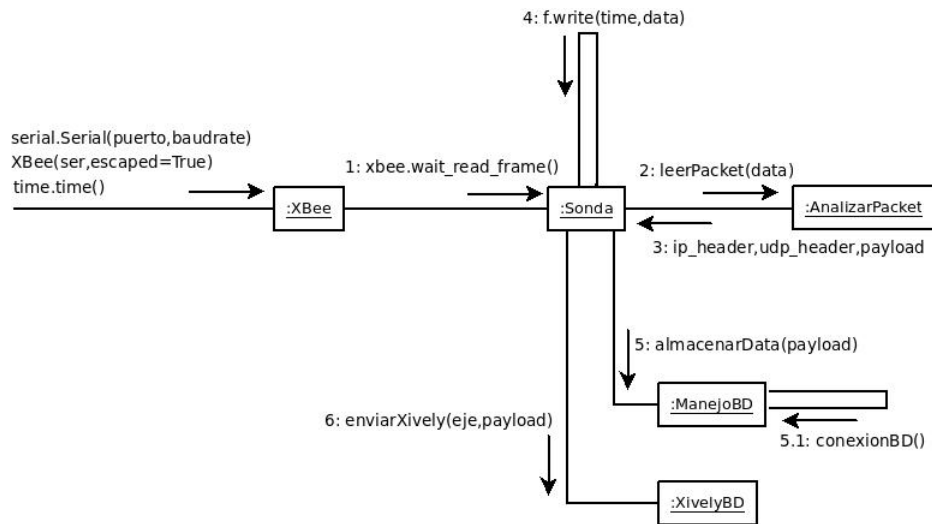


Figura 34. Diagrama de colaboración de la Sonda

4.4.3. Base de datos

Dado el modelo de datos del Sistema de Gestión de Puentes, son añadidas dos nuevas tablas asociadas a Puente: “Sensor” y “Muestreo”. La Figura 35 presenta un extracto del modelo de datos del Sistema de Gestión de Puentes.

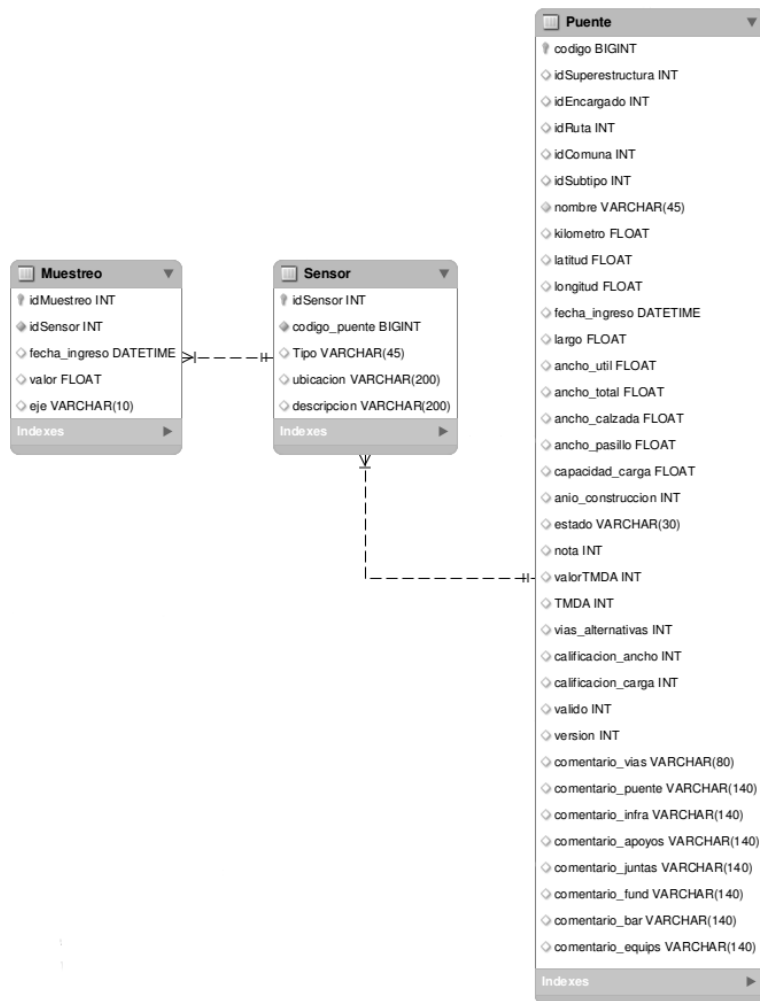


Figura 35. Subconjunto del modelo de datos del Sistema de Gestión de Puentes

Con el fin de satisfacer los requerimientos planteados, y considerando las condiciones del entorno de red de prueba implementado, son añadidas tablas manteniendo la estructura de la Base de Datos del Sistema. De esta forma, un puente puede contener varios sensores, y cada sensor varias tuplas con muestreos; cabe destacar que cada sensor puede alimentar al Sistema con grandes volúmenes de data. Para “Sensor” y “Muestreo” se desglosan sus respectivos diccionario de datos en las Tablas 19 y 20.

Tabla 19. Diccionario de datos de la tabla Sensor

Atributo	Tipo dato	Llave primaria	Descripción	Entidad foránea
idSensor	INT	PK	Identificador del sensor	
codigo_puente	BIGINT	FK	Puente asociado al sensor	Puente (codigo)
Tipo	VARCHAR (45)		Tipo de sensor (por ejemplo: Acelerometro)	
ubicacion	VARCHAR (200)		Ubicacion del sensor en la estructura	
descripcion	VARCHAR (200)		Descripción del sensor	

Tabla 20. Diccionario de datos de la tabla Muestreo

Atributo	Tipo dato	Llave primaria	Descripción	Entidad foránea
idMuestreo	INT	PK	Identificador del muestreo	
idSensor	INT	FK	Identificador del sensor	Sensor (idSensor)
fecha_ingreso	DATETIME		Fecha en que se ingresa muestreo	
valor	FLOAT		Valor de la variable muestreada	
eje	VARCHAR (10)		Eje coordenado muestreado	

5. IMPLEMENTACIÓN

El presente capítulo, apoyado en las fases de análisis y diseño anteriores, expone la construcción del prototipo basado en el hardware y software seleccionados. De esta manera, para alcanzar los fines planteados, fue creado un escenario de red 6LoWPAN (*IPv6 over Low-Power Wireless Personal Area Networks*) con un Nodo (sensor) y un Router a través de la configuración del *stack* de código abierto Arduino-IPv6 de Telecom Bretagne [Tel13] en dispositivos Arduino Mega 2560 y módulos XBee Series 1; además, el desarrollo de una Sonda permite analizar el tráfico de red en el escenario 6LoWPAN, junto con establecer un canal de comunicación hacia el Sistema de Gestión de Puentes.

Acompañado al proceso descrito en las siguientes páginas, el Anexo A contiene el artículo “Implementación Arduino-6LoWPAN para el Internet de las Cosas” presentado en el IV Congreso Internacional de Computación e Informática del Norte de Chile INFONOR-CHILE 2013.

5.1. Configuración de módulos XBee

Los módulos XBee de Digi son requeridos para brindar un medio inalámbrico de comunicación entre los dispositivos. En este contexto, los módulos XBee de la Figura 36 emplean el protocolo IEEE 802.15.4 y soportan las necesidades de bajo costo y bajo consumo de energía de las WSN (*Wireless Sensor Networks*).



Figura 36. Módulo XBee

Los módulos requieren un mínimo de energía y proporcionan entrega fiable de datos entre dispositivos, funcionando en la banda de frecuencia ISM (*Industrial, Scientific and Medical*) de 2.4 GHz [Max06]. La Figura 37 muestra las conexiones mínimas que necesita el módulo para poder ser utilizado, siendo requerida alimentación desde 2.8 a 3.4 Volt, la conexión a tierra y las líneas de transmisión de datos por medio de UART

(*Universal Asynchronous Receiver-Transmitter*), pines TXD Y RXD, para comunicarse con un microcontrolador o directamente al puerto serial [Oya10].

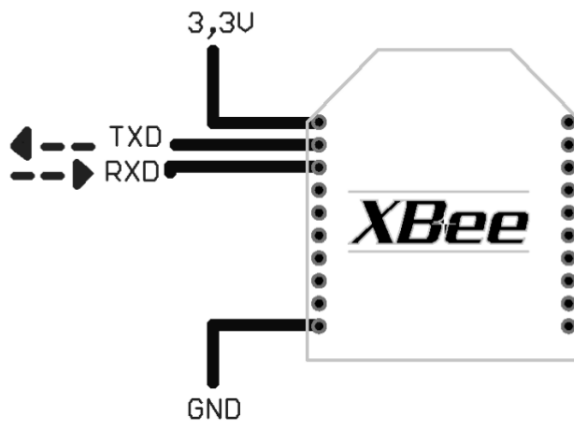


Figura 37. Conexiones mínimas para módulo XBee

Cabe destacar que para cumplir con los requerimientos del proyecto se debe trabajar con módulos XBee Series 1, ya que este proporciona IEEE 802.15.4 PHY y MAC *layer*, a diferencia del XBee series 2, el cual también brinda un *stack* ZigBee que no es compatible con la pila de protocolos de Telecom Bretagne. La Tabla 21 muestra las características de los módulos XBee utilizados.

Tabla 21. Características módulo XBee 1mW PCB Antenna

Velocidad de datos máxima:	115.2 kbP
Banda de frecuencia:	2.4 GHz
Potencia de transmisión:	1 mW (+0 dBm)
Rango:	100 m

Para configurar los módulos es utilizado el software X-CTU, como se observa en la Figura 38, estableciendo los siguientes parámetros:

- ID - PAN ID 0xABCD
- MM - MAC Mode 2 - 802.15.4 WITH ACKS
- AP - API Enable 2 - API ENABLED W/PPP
- MY - 16-bit Source Address 0xFFFF

- DH - Destination Address High 0x00
- DL - Destination Address Low 0xFFFF

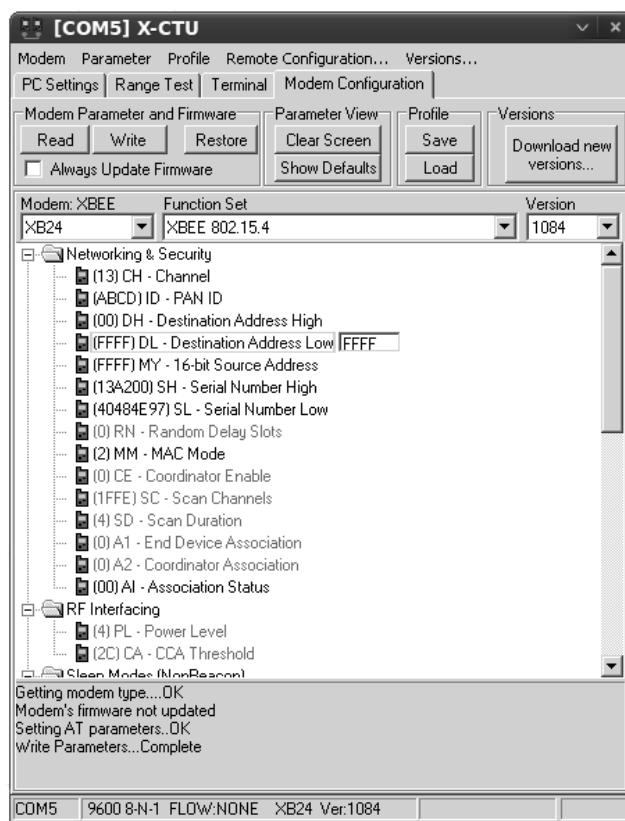


Figura 38. Configuración XBee con X-CTU

Al utilizar los módulos XBee es necesario trabajar en API mode (AP=2), donde la estructura del *frame* acepta caracteres de escape (0x7E, 0x7D, 0x11, 0x13), de modo que no interfieran con la operación de la trama de datos. La Figura 39 permite observar la estructura del *frame* UART en los módulos [Max06].

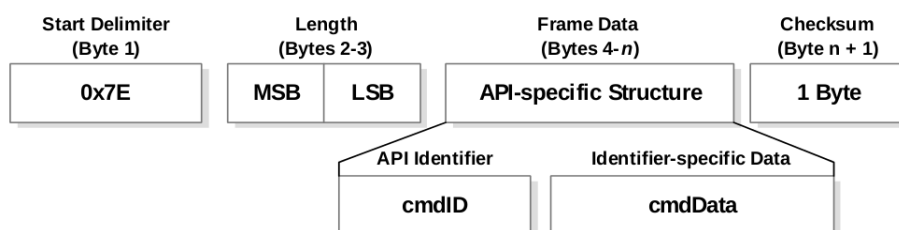


Figura 39. Frame UART

5.2. Arduino IPv6 stack

La pila de protocolos a implementar en los dispositivos Arduino Mega 2560 R3, desarrollada por Telecom Bretagne [Tel13], se encuentra basado en el sistema operativo

para redes de sensores Contiki OS. Entre las características del *stack* destaca la pila IP con normas de bajo consumo, como 6LoWPAN, RPL (*Routing Protocol for Low Power and Lossy Networks*), además de CoAP (*Constrained Application Protocol*) [She13], un protocolo de la capa de aplicación orientado a dispositivos con recursos limitados y con acceso a Internet. La Figura 40 muestra el *stack* de Telecom Bretagne.

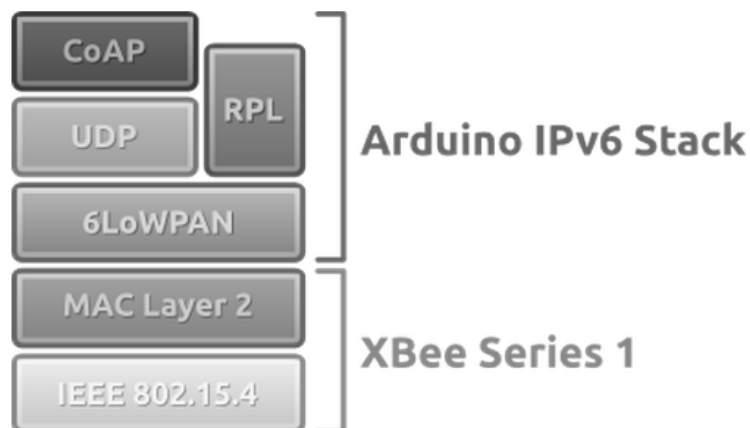


Figura 40. Arduino-IPv6 Stack de Telecom Bretagne

Para configurar el *stack* en los dispositivos, es utilizado el Arduino IDE 1.0 en un entorno Linux, donde es cargado como una librería en el directorio del IDE `/libraries/IPv6Stack/`, el cual contiene una adaptación del sistema operativo Contiki OS 2.5 y la librería XBee-Arduino²⁰ junto con modificaciones introducidas por Telecom Bretagne. Con este fin, se debe cambiar el valor del `SERIAL_BUFFER_SIZE` de Arduino a 128 en el archivo `/arduino-directory/hardware/arduino/cores/arduino/HardwareSerial.cpp`.

- `#define SERIAL_BUFFER_SIZE 128`

5.2.1. Configuración del nodo

Para la configuración del dispositivo Arduino como Nodo se debe acceder al archivo `contiki_conf.h` dentro del directorio de la librería `IPv6Stack/`, estableciendo los siguientes parámetros:

- `#define UIP_CONF_ROUTER 0` (ser un router o no)
- `#define UIP_CONF_IPV6_RPL 1` (con o sin RPL)
- `#define UIP_CONF_ND6_SEND_RA 0` (enviar anuncio de router o no)

²⁰<http://code.google.com/p/xbee-arduino/>

De esta forma, el Nodo envía mensajes multicast de modo que si la red posee un prefijo cambia el propio por el de la dirección IP de origen del mensaje.

Sin embargo, además de configurar la pila de protocolos y establecer comunicación inalámbrica, también es de interés del proyecto montar un sensor en el Nodo de manera de monitorear la salud estructural de un puente, en este caso, un acelerómetro para captar las vibraciones en la estructura, tal como se aprecia en la Figura 41.

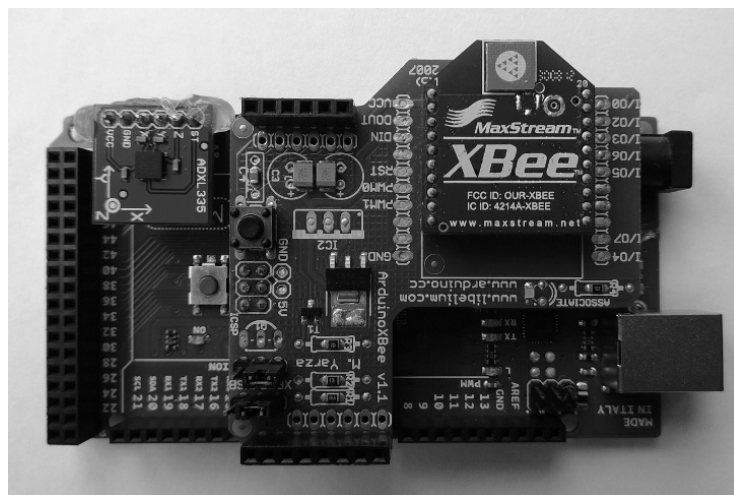


Figura 41. Nodo con sensor acelerómetro

Al montar el sensor a la placa Arduino son inicializados los pines para la lectura de los puertos análogos según el siguiente circuito:

- analog 8: self test
- analog 9: z-eje
- analog 10: y-eje
- analog 11: x-eje
- analog 12: ground
- analog 13: vcc

El Nodo con una dirección IPv6 envía mensajes a un Router utilizando el esquema 6LoWPAN, inicializando los componentes necesarios mediante el proceso que se observa en la Figura 42. El sensor muestrea a 60 Hz por 30 segundos, almacenando la data en un *array* de enteros de 2 Bytes de capacidad para cada INT, para luego enviar la data en

paquetes de nueve valores cada 1 segundo. Adicionalmente el Anexo B contiene el *sketch* del Nodo.

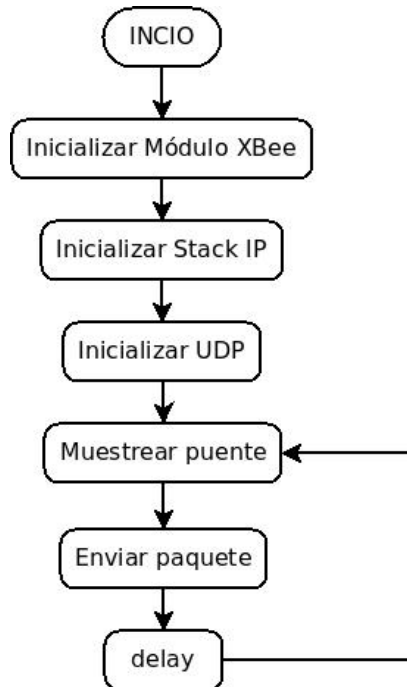


Figura 42. Diagrama de flujo del nodo

El sensor acelerómetro ADXL335 muestrea los tres ejes a 10 bits, obteniendo valores entre 0 y 1023, en un rango entre $-3g$ a $+3g$, siendo posible observar su respectivo diagrama de bloques en la Figura 43 [Ana09].

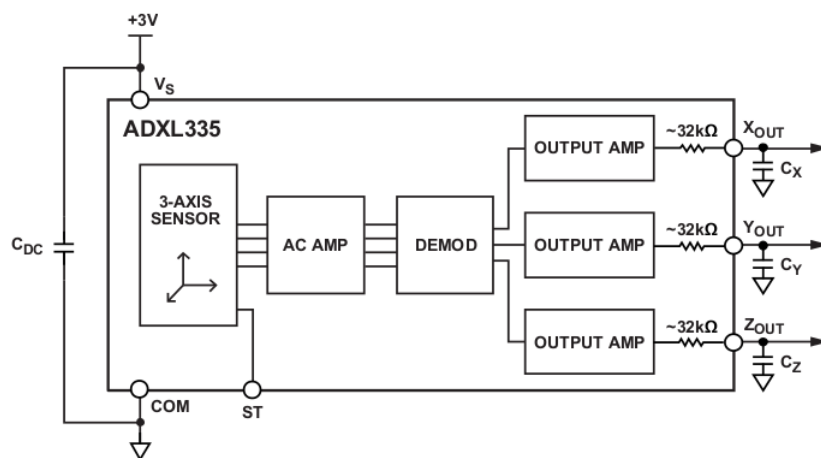


Figura 43. Diagrama de bloques del acelerómetro

De esta manera, la carga del paquete enviado por el Nodo contiene nueve muestras de la componente vertical separados por el carácter ‘a’, siendo concluida la trama por los caracteres ‘aa’, tal como es posible observar en el ejemplo siguiente:

- 590a596a591a594a591a590a594a594a594aa

Obteniendo los valores binarios del sensor en decimales, es realizado un proceso de conversión a fuerzas G, dado por la fórmula:

$$FuerzaG = \left(\frac{6}{1024} * valor_arduino \right) - 3 \quad (1)$$

De la misma manera, se sigue un proceso similar de conversión para obtener el resultado en m/s^2 , aplicando la siguiente fórmula:

$$Aceleracion = \left(\left(\frac{6}{1024} * valor_arduino \right) - 3 \right) * 9,8 \quad (2)$$

5.2.2. Configuración del router

Para el escenario de red 6LoWPAN es configurado un *border router*, el cual no envía ni responde a los mensajes, sino que establece un prefijo para la red.

Un Arduino Mega 2560 R3 junto a un módulo XBee constituyen su hardware, mientras que los procesos que lleva a cabo se muestran en la Figura 44. El software del Router lo conforma el *sketch* de prueba del Arduino-IPv6 *stack* de Telecom Bretagne, siendo establecido como prefijo `aaaa::/64`, con un tiempo de envío entre paquetes de 1 segundo.

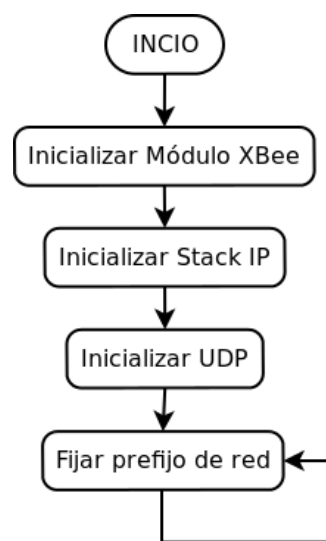


Figura 44. Diagrama de flujo del border router

Al acceder al archivo `contiki_conf.h` en el directorio de la librería `IPv6Stack/`, son establecidos los siguientes parámetros:

- `#define UIP_CONF_ROUTER 1` (ser un router o no)
- `#define UIP_CONF_IPV6_RPL 1` (con o sin RPL)

- #define UIP_CONF_ND6_SEND_RA 0 (enviar anuncio de router o no)

5.2.3. Direccionamiento IPv6

Contando con un escenario de red compuesto por un Nodo y un Router, cada uno con una dirección IPv6, resulta pertinente destacar aspectos referentes a la notación utilizada para los 128 bits, la cual se basa en ocho valores de 16 bits separados por dos puntos y representados por un máximo de cuatro dígitos hexadecimales, donde los ceros iniciales suelen omitirse al ser reemplazados por dos puntos dobles (::). La Tabla 22 ilustra algunas direcciones IPv6 [She09].

Tabla 22. Direcciones IPv6 en notación hexadecimal

Formato largo	Formato abreviado	Explicación
2001:DB8:0:0:8:800:200C:417A	2001:DB8::8:800:200C:417A	dirección unicast
FF01:0:0:0:0:0:101	FF01::101	dirección multicast
0:0:0:0:0:0:0:1	::1	dirección de loopback (unicast del localhost)
0:0:0:0:0:0:0:0	::	dirección no especificada

A modo de ejemplo, las Figuras 45 y 46 permiten apreciar el *frame* de una dirección IPv6 de enlace local y unicast global respectivamente.

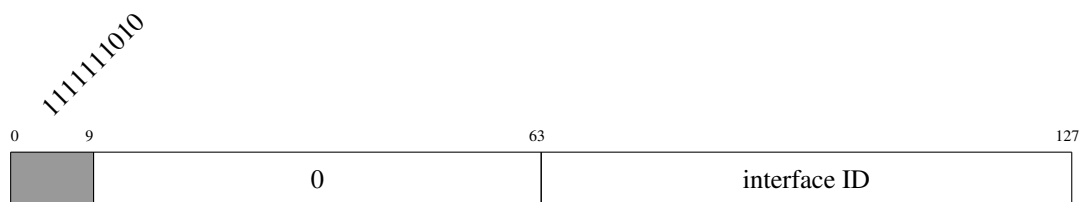


Figura 45. Dirección IPv6 de enlace local

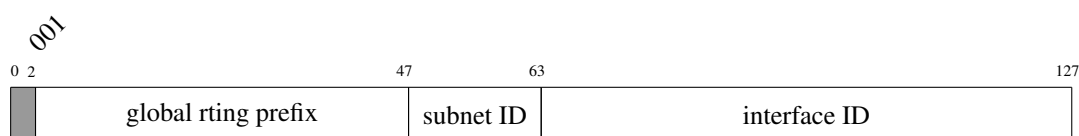


Figura 46. Dirección IPv6 unicast global

Además, cabe destacar que IPv6 ha sido definido para permitir la configuración automática basado en el concepto de un identificador de interfaz o *interface identifier* (IID) a nivel de enlace. Luego, este IID se combina con un prefijo para formar una dirección [She09].

5.2.4. Descubrimiento de vecinos

Para el escenario propuesto, un nodo utiliza el descubrimiento de vecinos (*Neighbor Discovery*) para encontrar otros nodos en el mismo enlace, para determinar sus direcciones a nivel de enlace (*link-layer*), para encontrar routers y para mantener la accesibilidad a la información acerca de las rutas de acceso a los vecinos con que el nodo se comunica activamente.

Al examinar los paquetes que intercambian Nodo y Router en la red, es posible identificar la dirección multicast ff02::1/128 a todos los nodos y la dirección multicast ff02::2/128 a todos los routers, identificando el proceso de registro de nodo y confirmación del mensaje (Figura 47) [She09].

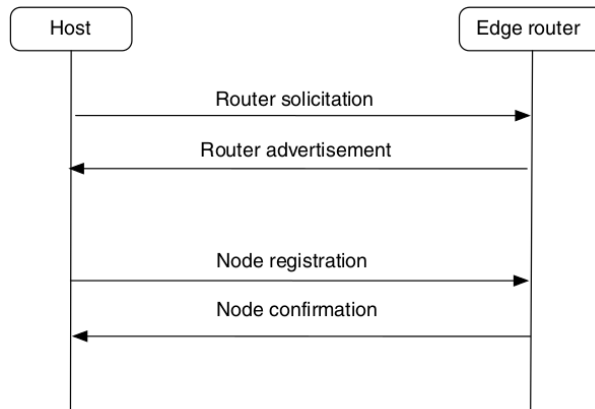


Figura 47. Proceso básico de descubrimiento de vecinos

5.3. Configuración de la sonda

En un escenario de una red de sensores con direccionamiento IPv6 que tiene por objeto la utilización del esquema 6LoWPAN en el contexto de la nueva Internet, una Sonda en la red permite capturar paquetes para su análisis junto con brindar un canal de comunicación hacia el Sistema de Gestión de Puentes con el fin de alimentarlo con data referente a, en este caso, las vibraciones de la estructura captadas por el sensor acelerómetro montado en el Nodo.

Con tal objetivo fijado, un módulo XBee configurado con los mismos parámetros del Nodo y del Router captura todos los paquetes de la red y los envía a través del puerto serial a un computador que ejecuta un *script*. El escenario de prueba se estableció de acuerdo al diagrama de la Figura 48.

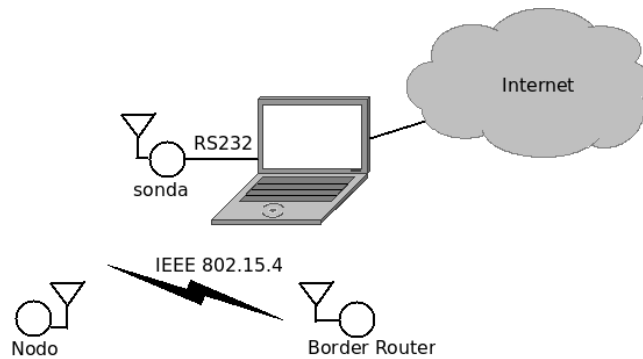


Figura 48. Escenario de prueba con sonda

El *script*, desarrollado en Python, utiliza la librería `python-xxbee`²¹ para trabajar en modo API con los módulos desde la consola. Además, extrae la data de los paquetes 802.15.4, IP y UDP para luego enviar la información a la base de datos del Sistema de Gestión de Puentes, tal como lo describe el diagrama de la Figura 49.

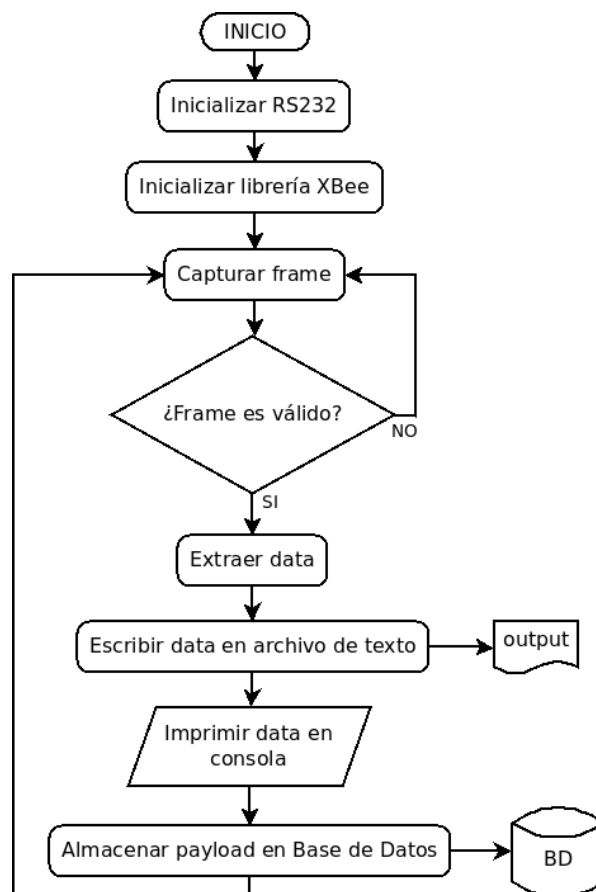


Figura 49. Diagrama de flujo de la sonda

²¹<http://code.google.com/p/python-xxbee/>

El Anexo C contiene el desglose de un paquete capturado por la Sonda.

5.4. Configuración de Xively

Junto con almacenar la información acerca de las condiciones del puente en la base de datos del Sistema de Gestión de Puentes se propone la utilización de la base de datos *on-line* Xively (antes llamada Cosm y Pachube).

Xively provee herramientas basadas en la web, permitiendo la conexión de dispositivos a una “nube pública” de información para así almacenar data obtenida por objetos que forman parte del Internet de las Cosas.

Para el presente proyecto se busca respaldar la información capturada por el Nodo tanto en el Sistema de Gestión de Puentes como en Xively; el último aspecto se logra por medio de la creación de un “sensor”, al cual se accede a través de la librería *xively-python*²² utilizando la *API key* provista y creando un *feed*. La Figura 50 muestra la gráfica creada en la web a partir de la data enviada.

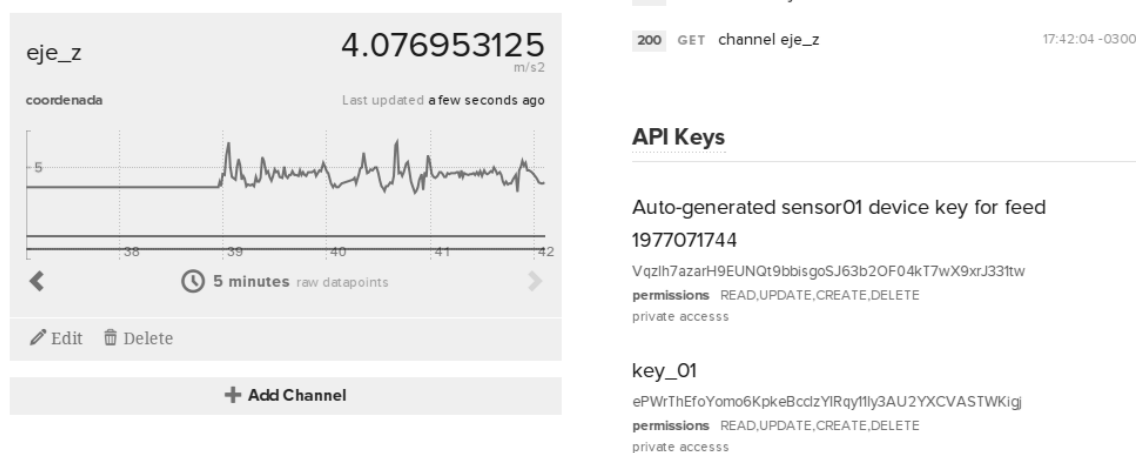


Figura 50. Base de datos Xively

La API de Xively soporta tres formatos de datos: JSON, XML y CSV. En la versión 2 de la API, la cual es utilizada por el *script* de la Sonda, si no se especifica ningún formato se utiliza JSON. Dado este escenario, cabe destacar que JSON se puede analizar fácilmente utilizando JavaScript, siendo un excelente formato de serialización de datos genéricos ya que los gastos de procesamiento son menores que XML, utilizando menos ancho de banda

²²<https://github.com/xively/xively-python>

para transmitir.

El Anexo D muestra como lucen las representaciones de JSON en Xively.

5.5. Vista del estado del puente

Una vez almacenada la información de la estructura provista por el sensor en la Base de Datos del Sistema de Gestión de Puentes, el usuario requiere conocer el estado del puente de manera gráfica en el sistema. En este sentido, en la Ficha de identificación del Puente (Figura 51) se añade una opción para visualizar los sensores que posee.

The screenshot displays the 'Sistema de Gestión de Puentes' web interface. At the top, there is a navigation bar with 'Inicio' and 'Módulos' menus, and a user profile for 'Bienvenido Paulo Gallardo'. On the left, a sidebar menu lists various system functions, with 'Identificación' selected. The main content area is titled 'Módulo Ver Puentes' and shows the details for 'Puente Calle Calle'. The details are organized into a table-like structure with a large photograph of the bridge on the right. The data includes:

Región: de los Ríos
Provincia: Valdivia
Comuna: Valdivia
Código: 39490373134001
Ruta: 74A202 / RUTA 202
Nombre del camino: CRUCE RUTA 5 (MARIQUINA) - CRUCE RUTA 206 (VALDIVIA)
Kilómetro: 48,238 km
Tipo de estructura: Viga continua, cajón de HA, multicelular, canto variable
Longitud total: 223,00 m
Ancho útil: No disponible
Ancho total: 12,00 m
Ancho calzada: 9,00 m
Ancho pasillo: 1,50 m
Capacidad de carga: No disponible
TMDA: 6000 Veh/día (estimado)
Año de construcción: 1944
Estado: En uso
Responsable: MOP
Nota de estado de conservación: 4

Figura 51. Ficha del puente en el Sistema de Gestión de Puentes

A nivel de prototipo, para el escenario de red dispuesto, un usuario registrado en el sistema debe iniciar sesión para acceder al módulo “Ver Puentes”, donde es posible realizar búsquedas sobre los puentes en la Región de los Ríos bajo diferentes criterios, como el nombre de la estructura, el código del camino en el que se ubica y el rol o nombre de dicho camino. Una vez que el usuario acceda a la ficha de identificación del puente se ofrece la opción de ver los sensores de la estructura, donde un gráfico representa las condiciones de la misma según la variable que se esté muestreando.

Para desplegar la data gráficamente es añadida la vista *sensores.php*, siendo utilizada la librería Highcharts²³, escrita en HTML5/JavaScript; Highcharts ofrece la creación de gráficos intuitivos e interactivos.

Una vez que se accede a la vista creada especialmente para el prototipo construido se logran apreciar las vibraciones de la estructura en la componente vertical, siendo desplegado un ciclo de muestreo completo, el cual consta de 1800 datos tomados durante 30 segundos a una frecuencia de muestreo de 60 Hz. La Figura 52 permite apreciar algunos detalles del gráfico propuesto, considerando condiciones de laboratorio en la mediciones expuestas.

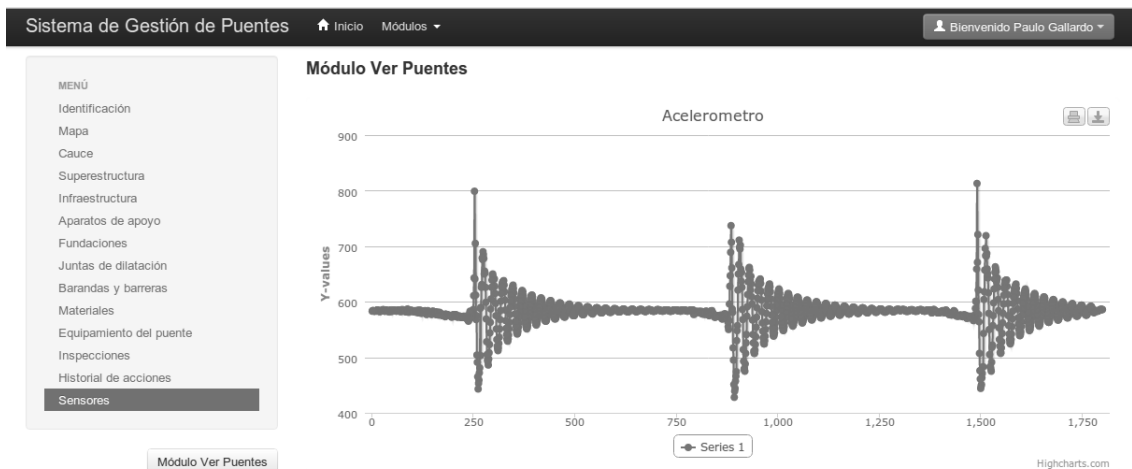


Figura 52. Vista para sensores en Sistema de Gestión de Puentes

²³<http://www.highcharts.com/>

6. RESULTADOS

El capítulo que se extiende en las siguientes páginas tiene por objeto profundizar en los resultados obtenidos a partir de prototipo construido, estableciendo métricas y realizando pruebas para validar el producto conseguido.

Al tratarse de tecnologías que formarán parte del Internet de las Cosas (IoT) en un futuro, y considerando la utilización de dispositivos de hardware abierto como Arduino, se estableció como métrica la medición del tiempo que tarda un nodo en enviar información desde que efectúa la solicitud de router. Además, se considera el análisis de los registros del sensor capturados en terreno.

6.1. Solicitud de router

Teniendo en cuenta las limitaciones en términos de capacidad de procesamiento, memoria y bajo uso del ancho de banda que impone la red de sensores, se procedió a medir el tiempo que tarda un nodo en enviar datos desde que se realiza la solicitud de router (ff02::2/128). Cabe destacar que la librería de Arduino que permite la implementación de la pila de protocolos para 6LoWPAN (*IPv6 over Low-Power Wireless Personal Area Networks*) constituye un factor a considerar, ya que la eficiencia de su código trae repercusiones en los tiempos contemplados en el proceso de envío.

Al registrar paquetes con una sonda en el escenario dispuesto por un nodo y un *border router*, se obtuvieron los resultados de la Figura 53.

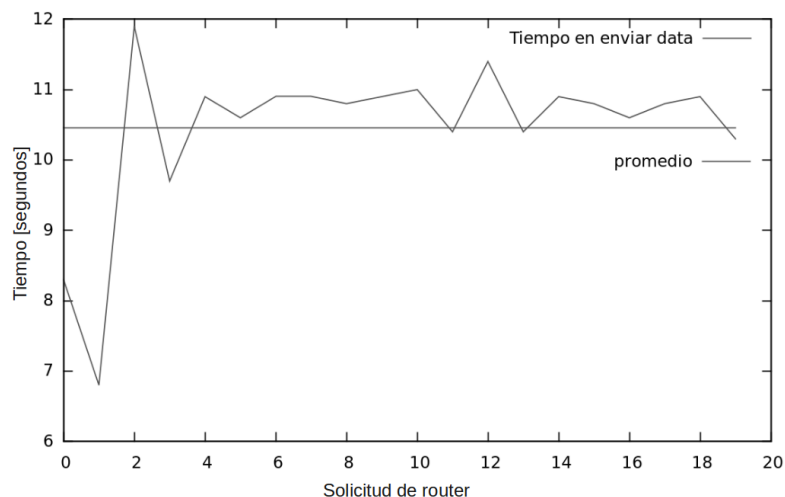


Figura 53. Tiempo desde solicitud de router hasta envío de data

Para esta red, con un tiempo de envío entre paquetes de 1 segundo, transcurren en promedio 10.46 segundos desde que el nodo realiza la solicitud de router hasta que envía el primer paquete con data (adquiere prefijo de la red).

Sin embargo, es importante mencionar que el comportamiento de la red se vería notablemente influido si se contara con un gran número de nodos enviando datos al mismo tiempo. Para esta situación, el impacto en los dispositivos de comunicación resultaría notable, tanto como el impacto en la base de datos, donde esquemas relacionales tal vez no sean los indicados para recibir estos grandes flujos de información; de esta manera el Internet de las Cosas trata tópicos referidos a los grandes volúmenes de datos que generaría la gran cantidad de objetos conectados a la red, aludiendo al concepto de *Big Data*, con un potencial de trillones de artefactos debido a la inminente adopción de IPv6.

En este sentido, también influyen aspectos como el tiempo de envío entre paquetes y el tamaño de los paquetes en la red dispuesta; para el escenario de red del prototipo se consideró un tiempo de envío entre paquetes de 1 segundo, y un paquete de 9 muestras, de manera de generar una carga lo suficientemente grande como para producir el proceso de compresión de las cabeceras, pero sin saturar el canal ni los dispositivos involucrados.

6.2. Mediciones con acelerómetro

La inclusión de un sensor en el nodo de la red permite enviar datos referente al entorno en el cual se enmarca el dispositivo. En este contexto, para el prototipo construido, un sensor acelerómetro permitiría captar variables con respecto a las vibraciones de una estructura, alimentando al Sistema de Gestión de Puentes con información de interés para conocer el estado del puente.

De esta forma, en primera instancia se recurrió a la realización de pruebas con el sensor acelerómetro en un ambiente conformado por una regla de medir de 60 centímetros fijada en el extremo de una mesa, quedando 35 centímetros suspendidos en el aire. El nodo se posiciona en el extremo de la regla y es sujeto a estimulaciones para captar el comportamiento del sensor en la componente vertical. Posteriormente se utiliza *gnuplot* para graficar los datos obtenidos en archivos de texto; en la Figura 54 se observan los resultados de la vibración de la componente vertical, siendo excitada con un golpe en a),

con 3 golpes en b) y con 5 golpes en c).

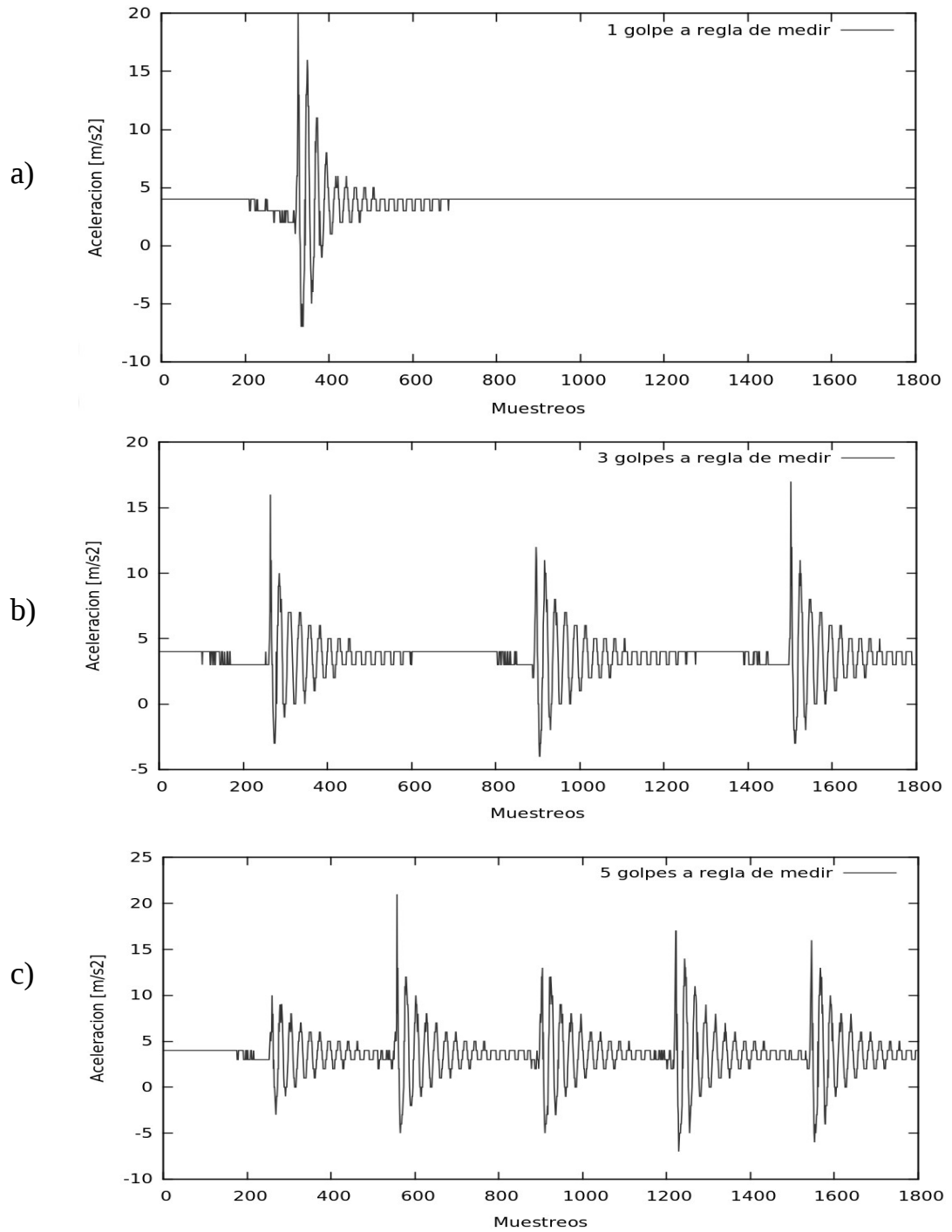


Figura 54. Muestras de regla de medir

En la línea del proceso de medición del prototipo, es probada la comunicación entre el nodo y el sensor por medio de transmisión inalámbrica utilizando el estándar 6LoWPAN para la compresión de cabeceras de los mensajes IPv6/UDP.

En este sentido, es llevado el nodo-sensor y el router a un puente en la localidad de

Valdivia, específicamente al puente Calle Calle (tránsito hacia Las Animas), donde se fija el sensor en medio del tramo más largo, entre dos puntos de apoyo del puente. De esta manera son registradas vibraciones en la estructura, realizando mediciones a 60 Hz durante 30 segundos, según se observa en la Figura 55, donde el eje Y representa las medidas según las capturas del sensor en la placa Arduino, en 10 bits, con valores decimales entre 0 y 1023 (-+3g).

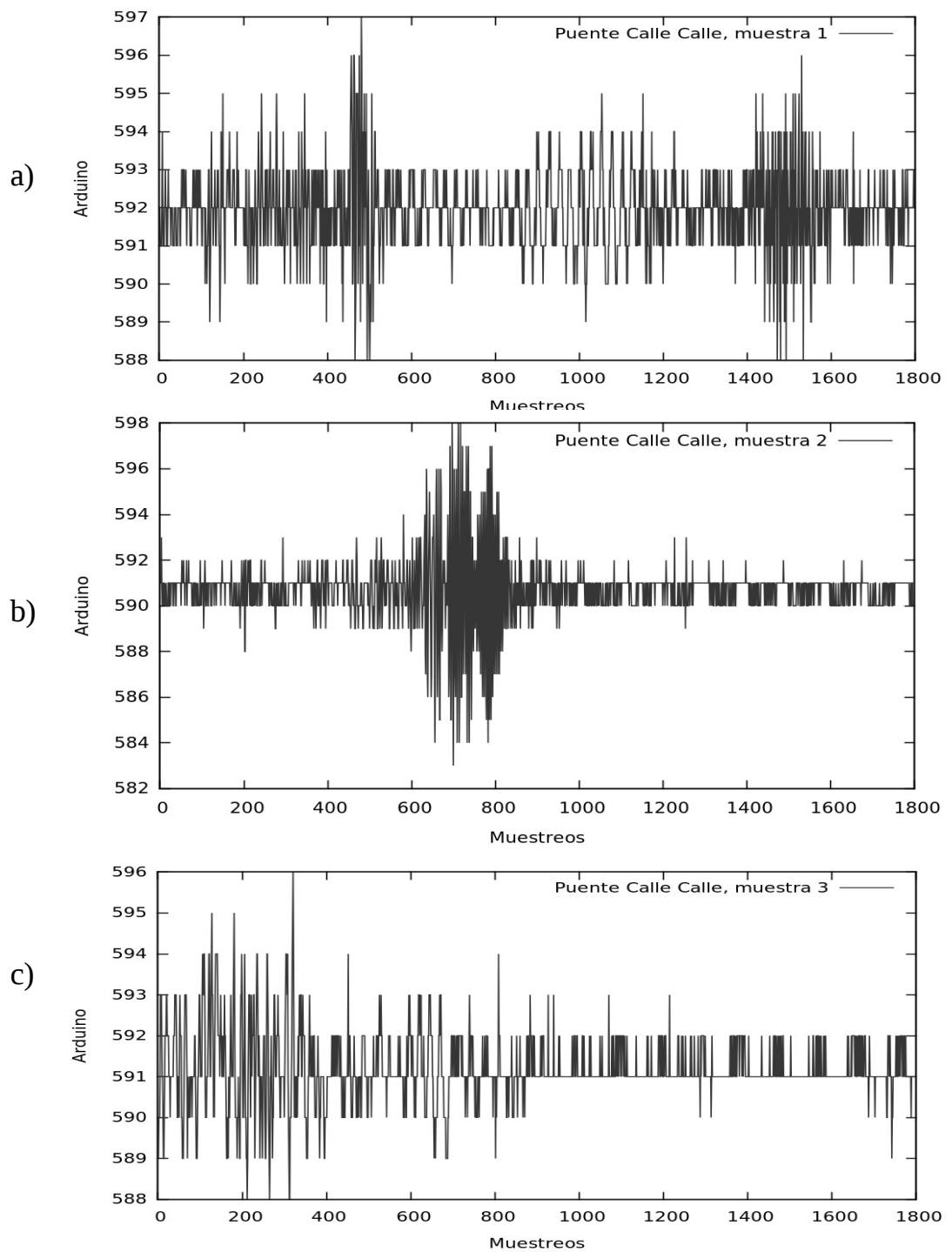


Figura 55. Muestras del puente Calle Calle

Las vibraciones de la estructura son registradas por el nodo y enviadas mediante un tren de data en paquetes de 9 muestras cada 1 segundo al *border router*. En este proceso, la sonda captura los paquetes en la red y los envía a la base de datos del Sistema de Gestión de Puentes para que posteriormente un usuario registrado observe las condiciones de la estructura en la web.

En la estructura pueden posicionarse varios sensores e interactuar con el sistema, estableciendo una red de monitoreo que permita entregar información de importancia para el desarrollo y mantención de la red vial, haciendo realidad en este contexto al IoT.

Además, dadas las características del sensor utilizado, resulta imprescindible señalar que las pruebas efectuadas permiten probar los mecanismos de transmisión de datos en excelentes condiciones a nivel de prototipo, sin embargo, dada la resolución del sensor acelerómetro dispuesto en la placa Arduino (ADXL335)[Ana09], los datos no pueden ser utilizados para efectuar ningún análisis estructural del puente. De esta forma, para contribuir satisfactoriamente en el monitoreo de las condiciones de las estructuras en un escenario real se debe contar con sensores de mayor capacidad y optimizados para las condiciones a las cuales serán expuestos, tales como los utilizados por la compañía de servicios de monitoreo estructural y geotécnico Ambher Ingeniería²⁴, entre los cuales es posible destacar:

- Miron Optics os7100²⁵, el cual se encuentra optimizado para grandes estructuras y medidas a largo plazo.
- Acelerómetro de Movimientos Fuertes TITAN²⁶, un sensor triaxial de balance de fuerzas que provee rendimiento en un rango de frecuencia que va desde DC (corriente directa) hasta 430 Hz. Diseñado para aplicaciones de monitoreo estructural y de movimientos fuertes en campo libre.

También se debe tener en consideración el uso de una plataforma de desarrollo hardware que provea mayores prestaciones que Arduino para un escenario real, tal como BeagleBone Black²⁷, una placa de hardware abierto basado en un procesador ARM Sitara AM335x Cortex-A8 de Texas Instruments con una velocidad de reloj de 1 GHz.

²⁴<http://www.ambher.com/>

²⁵<http://www.ambher.com/content/Micron%20Optics%20os7100.pdf>

²⁶<http://www.ambher.com/content/NMX-Titan.pdf>

²⁷<http://beagleboard.org/>

7. CONCLUSIONES

Dado el nivel actual de las tecnologías que harán posible el Internet de las Cosas (IoT), y considerando la normalización existente, el uso de alternativas abiertas favorece la experimentación, sin embargo, al trabajar con tecnologías en desarrollo se requiere alta comprensión de los protocolos implicados en la nueva Internet. En este escenario, 6LoWPAN (*IPv6 over Low-Power Wireless Personal Area Networks*) es el componente clave para el Internet del futuro.

6LoWPAN hace posible el uso del IEEE 802.15.4 para el Internet de las Cosas, logrando establecer un enlace con el protocolo IP habilitado para los objetos, lo cual proporciona interoperabilidad para sistemas abiertos de dispositivos de baja potencia.

Además, se obtiene gran capacidad para trabajar dentro de las limitaciones que plantean los sistemas embebidos en cuanto a recursos: baja potencia, memoria limitada y bajo ancho de banda para este tipo de nodos.

En este sentido, destaca la alternativa provista por Arduino debido a la facilidad para conseguir los módulos necesarios para construir soluciones. Además, su utilización se aproxima fehacientemente a la proyectada por el IoT, dado el desarrollo sobre aparatos que presentan restricciones en término de recursos, pero que, sin embargo, pueden gozar de cierta autonomía logrando establecer monitoreo continuo en terreno a nivel de prototipo.

De esta forma, El IoT se considera la próxima gran oportunidad y desafío para la comunidad, usuarios de tecnología, empresas y sociedad, mientras que 6LoWPAN es la tecnología que hace posible esta pronta realidad.

Socialmente, la irrupción del Internet de las Cosas implica una revolución, un cambio en la forma en las personas interactúan con su entorno, con sus pares y consigo mismas. La comunicación con los objetos depende de la convergencia entre redes de bajo consumo y pérdida e IPv6, e implica una reivindicación de las cosas de uso cotidiano, las cuales se relacionarán con cada individuo de una forma hasta ahora insospechada.

La utilización de sensores que implementen 6LoWPAN permitiría optimizar y/o

automatizar procesos, así como también establecer canales de comunicación a través de la infraestructura actual de la red entre objetos y personas, o entre objetos.

También, se ha evidenciado que la tendencia de la nueva Internet tiene la capacidad de transformarse prontamente en una realidad con el potencial para ser aplicada a ámbitos tales como el monitoreo de estructuras, monitoreo de medio ambiente, ahorro de energía, fábricas más eficientes, mejor logística, cuidado de la salud y hogares inteligentes.

En el contexto del proyecto en particular, el IoT se vuelve una realidad a través del monitoreo continuo de las condiciones de las estructuras, conviviendo y brindando información de interés a un Sistema de Gestión de Puentes, contribuyendo a la mantención oportuna de la red vial y permitiendo automatizar y/u optimizar procesos que lleva a cabo el sistema.

BIBLIOGRAFÍA

- [Akg07] Akgül F. (2007). Criteria for a bridge management system based on inspection, monitoring and maintenance practices en Cho H., Frangopol D. & Ang A. ed. Life-Cycle Cost and Performance of Civil Infrastructure Systems, Taylor & Francis Group, Londres, 79 - 84.
- [Ana09] Analog Devices (2009). Accelerometer ADXL335. Disponible en <https://www.sparkfun.com/datasheets/Components/SMD/adxl335.pdf>. Consultado el 02 de Octubre de 2013.
- [Cas11] Casado D., Vega M. & López M. (2011). Infraestructuras Inteligentes en el Internet del Futuro. Universidad de Alcalá, Grupo de Ingeniería en Servicios Telemáticos, Escuela Politécnica. Disponible en http://paginaspersonales.deusto.es/dcasado/docs/ei3_2011.pdf. Consultado el 01 de Agosto de 2013.
- [Cis11] Cisco (2011). Big Data in the Enterprise: Network Design Considerations. Disponible en http://www.cisco.com/en/US/prod/collateral/switches/ps9441/ps9670/white_paper_c11-690561.pdf. Consultado el 20 de Agosto de 2013.
- [Com09] Commission of the European Communities (2009). Internet of Things - An action plan for Europe. Disponible en <http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=COM:2009:0278:FIN:EN:PDF>. Consultado el 01 de Agosto de 2013.
- [Dee98] Deering, S. & Hinden, R. (1998). RFC 2460. Internet Protocol, Version 6 (IPv6) Specification. Internet Engineering Task Force. Disponible en <http://www.ietf.org/rfc/rfc2460.txt>. Consultado el 07 de Agosto de 2013.
- [Del07] Delclós T. (2007). El reto del 'Internet de las Cosas'. Diario El País. Disponible en http://elpais.com/diario/2007/05/17/ciberpais/1179368665_850215.html. Consultado el 01 de Agosto de 2013.
- [Dun10] Dunkels A. & Vasseur JP. (2010). White paper #1: Why IP. IP for Smart Objects. Internet Protocol for Smart Objects (IPSO) Alliance. Disponible en <http://www.ipso-alliance.org/white-papers>. Consultado el 06 de Septiembre de 2013.
- [Els11] Elster Group (2011). A Standardized and Flexible IPv6 Architecture for Field Area Networks. Smart Grid Last Mile Infrastructure. Disponible en <http://www.elster.com/assets/downloads/IP-arch-SG-WP-clean-final-112211.pdf>. Consultado el 09 de Septiembre de 2013.
- [Esh] Echaveguren T., Subiabre M., Echaveguren E., & León C. (n.d.). Proposición de un Subsistema de Información para el Sistema de Gestión de Puentes MAPRA. Disponible en http://www2.udec.cl/~provincial/trabajos_pdf/11TomasEchavegurenSistemapuentesMapra.pdf. Consultado el 07 de Agosto de 2013.
- [Eur] European Research Cluster on the Internet of Things (n.d.). Disponible en <http://www.internet-of-things-research.eu/>. Consultado el 07 de Agosto de 2013.

- [Gar09] Gracia E. (2009). Implementación de Protocolos de Transporte en Redes de Sensores. Escuela Técnica Superior de Ingeniería de Telecomunicación de Barcelona, Universidad Politécnica de Cataluña.
- [Gut04] Gutiérrez J., Barrett R. & Callaway E. (2004). Low-rate Wireless Personal Area Networks: Enabling Wireless Sensors with IEEE 802.15.4. Institute of Electrical and Electronics Engineers, New York.
- [Hin06] Hinden R. & Deering S. (2006). RFC 4291. IP Version 6 Addressing Architecture. Internet Engineering Task Force. Disponible en <http://www.ietf.org/rfc/rfc4291.txt>. Consultado el 07 de Agosto de 2013.
- [Hui11] Hui J. & Thubert P. (2011). RFC 6282. Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks. Internet Engineering Task Force. Disponible en <http://www.ietf.org/rfc/rfc6282.txt>. Consultado el 24 de Septiembre de 2013.
- [IEE03] IEEE Standards Association (2003). IEEE Standard 802.15.4-2003. Disponible en <http://standards.ieee.org/getieee802/download/802.15.4-2003.pdf>. Consultado el 22 de Agosto de 2013.
- [IMS12] IMS Research (2012). Internet Connected Devices Approaching 10 Billion, to exceed 28 Billion by 2020. Disponible en http://www.imsresearch.com/press-release/Internet_Connected_Devices_Approaching_10_Billion_to_exceed_28_Billion_by_2020. Consultado el 28 de Agosto de 2013.
- [IPv11] IPv6 para Chile (2011). Fase de Inteligencia de Mercados y Competitiva. Informe de Tendencias N° 6. Disponible en <http://www.ipv6.cl/system/files/Informe-de-Tendencias-enero-2011.pdf>. Consultado el 09 de Septiembre de 2013.
- [Jim12] Jiménez A., Jiménez S., Lozada P. & Jiménez C. (2012). Wireless Sensors Network in the Efficient Management of Greenhouse Crops. 2012 Ninth International Conference on Information Technology - New Generations. Las Vegas, 680 - 685.
- [Kas13] Kaschel H. & Iturralde D. (2013). Análisis de Mejoras en la Agricultura Aplicando WSN: Cultivo de Rosas. XIV Congreso Internacional de Telecomunicaciones SENACITEL 2013, Valdivia.
- [Kuo07] Kuorilehto M., Kohvakka M., Suhonen J., Hämäläinen P., Hännikäinen M. & Hämäläinen TD. (2007). Ultra-Low Energy Wireless Sensor Networks in Practice. Wiley, Great Britain.
- [Kus07] Kushalnagar N., Montenegro G. & Schumacher C. (2007). RFC 4919. IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals. Internet Engineering Task Force. Disponible en <http://www.ietf.org/rfc/rfc4919.txt>. Consultado el 07 de Agosto de 2013.
- [Lar99] Larman C. (1999). UML y Patrones: Introducción Al Análisis y Diseño Orientado a Objetos. Prentice-Hall, México.
- [Lat12] Latin America and Caribbean Network Information Centre (2012). Estado de IPv4 a fin de 2012. Disponible en: <http://portalipv6.lacnic.net/estado-de-ipv4-a-fin-de-2012-es/>. Consultado el 20 de Agosto de 2013.

- [Max06] MaxStream (2006). XBeeTM/XBee-PROTM OEM RF Modules. Product Manual v1.xAx - 802.15.4 Protocol. MaxStream, Inc., London.
- [Mol12] Molina N. (2012). Diseño de un Sistema de Gestión de Puentes bajo Enfoque de Priorización de la Inversión. Escuela de Ingeniería Civil en Obras Civiles, Universidad Austral de Chile, Valdivia.
- [Mon07] Montenegro G., Kushalnagar N., Hui J. & Culler D. (2007). RFC 4944. Transmission of IPv6 Packets over IEEE 802.15.4 Networks. Internet Engineering Task Force. Disponible en <http://www.ietf.org/rfc/rfc4944.txt>. Consultado el 07 de Agosto de 2013.
- [Nar07] Narten T., Nordmark E., Simpson W. & Soliman H. (2007). RFC 4861. Neighbor Discovery for IP version 6 (IPv6). Internet Engineering Task Force. Disponible en <http://www.ietf.org/rfc/rfc4861.txt>. Consultado el 07 de Agosto de 2013.
- [Nat10] National Institute of Standards and Technology (2010). NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 1.0. Disponible en http://www.nist.gov/public_affairs/releases/upload/smartgrid_interoperability_final.pdf. Consultado el 09 de Septiembre de 2013.
- [NXP11] NXP Laboratories UK Ltd (2011). JenNet-IP Network Protocol Stack. Low-Power Wireless IP Networking for the ‘Internet of Things’. Disponible en http://www.jennic.com/files/product_briefs/JenNet-IP-PBv1.2docx.pdf. Consultado el 01 de Agosto de 2013.
- [Och] Ochoa A. (n.d.). Métodos científicos. Disponible en <http://www.monografias.com/trabajos11/metods/metods.shtml>. Consultado el 23 de Septiembre de 2013.
- [Oya10] Oyarce A. (2010). Guía del Usuario XBEE Series 1. Ingeniería MCI Ltda., Santiago.
- [Peñ13] Peña C. & Ralli C. (2013). IPv6: El motor de “La WEB de las Cosas”. Blog Think Big. Disponible en <http://blogthinkbig.com/ipv6-motor-internet-de-las-cosas-iot/>. Consultado el 27 de Agosto de 2013.
- [Pos80] Postel J. (1980). RFC 768. User Datagram Protocol. Internet Engineering Task Force. Disponible en <http://www.ietf.org/rfc/rfc768.txt>. Consultado el 06 de Septiembre de 2013.
- [Rev09] Reventós L. (2009). El ‘Internet de las Cosas’ ahorraría 200000 muertes anuales en las carreteras europeas. Diario El País, Tecnología. Disponible en http://tecnologia.elpais.com/tecnologia/2009/05/20/actualidad/1242810061_850215.html. Consultado el 01 de Agosto de 2013.
- [Rya01] Ryall M. (2001). Bridge Management. Elsevier, Great Britain.
- [Scr13] Scrum Manager (2013). Scrum Manager BoK (SMBoK). Disponible en: http://www.scrummanager.net/bok/index.php?title=Main_Page. Consultado el 25 de Septiembre de 2013.
- [She09] Shelby Z. & Bormann C. (2009). 6LoWPAN: The Wireless Embedded Internet. Wiley, Great Britain.

- [She12] Shelby Z., Chakrabarti S., Nordmark E. & Bormann C. (2012). RFC 6775. Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs). Internet Engineering Task Force. Disponible en <http://www.ietf.org/rfc/rfc6775.txt>. Consultado el 24 de Septiembre de 2013.
- [She13] Shelby Z., Hartke K. & Bormann C. (2013). draft-ietf-core-coap-18. Constrained Application Protocol (CoAP). Internet Engineering Task Force. Disponible en <http://tools.ietf.org/id/draft-ietf-core-coap-18.txt>. Consultado el 01 de Octubre de 2013.
- [Taf12] Taffernaberry C. (2012). 6LoWPAN: IPv6 for Wireless Sensor Network. Simposio Argentino de Sistemas Embebidos (SASE). Disponible en <http://www.sase.com.ar/2012/files/2012/09/4-2012-SASE-6lowpan.pdf>. Consultado el 21 de Agosto de 2013.
- [Tel13] Telecom Bretagne (2013). Arduino μ IPv6 Stack. Disponible en <https://github.com/telecombretagne/Arduino-IPv6Stack/wiki>. Consultado el 24 de Septiembre de 2013.
- [Win12] Winter T., Thubert P., Brandt A., Hui J., Kelsey R., Levis P., Pister K., Struik R., Vasseur JP & Alexander R. (2012). RFC 6550. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks . Internet Engineering Task Force. Available: <http://www.ietf.org/rfc/rfc6550.txt>. Consultado el 29 de Agosto de 2013.
- [Yu06] Yu Y., Prasanna VK., & Krishnamachari B. (2006). Information Processing and Routing in Wireless Sensor Networks. World Scientific Publishing Co. Pte. Ltd., Singapore.
- [Zia08] Ziadé T. (2008). Expert Python Programming. Best practices for designing, coding, and distributing your Python software. Packt Publishing, Birmingham.

ANEXOS

Anexo A: Artículo INFONOR

IV Congreso Internacional de Computación e Informática del Norte de Chile

INFONOR-CHILE 2013

Implementación Arduino-6LoWPAN para el Internet de las Cosas

Paulo A. Gallardo Casanova
Instituto de Informática
Universidad Austral de Chile
General Lagos 2086, Valdivia, Chile
gallardo.casanova@gmail.com

Christian A. Lazo Ramírez
Instituto de Informática
Universidad Austral de Chile
General Lagos 2086, Valdivia, Chile
clazo@inf.uach.cl

Resumen—En la actualidad son cada vez más los dispositivos tecnológicos que se incorporan en el diario vivir, los cuales otorgan un inusitado grado de conectividad como, por ejemplo, *tablets*, *smartphones*, o incluso objetos de uso frecuente, tales como un refrigerador. En este contexto, la humanidad comienza a convivir con una red de aparatos cotidianos interconectados entre sí, la cual se conoce como el “Internet de las Cosas”, el que sin lugar a dudas se convertirá en la próxima revolución tecnológica, generando nuevas oportunidades y desafíos para la comunidad, usuarios de tecnología, empresas y sociedad.

Lo que se persigue en el presente trabajo es comunicar los dispositivos a través de 6LoWPAN, un estándar basado en IPv6 sobre redes de bajo consumo y pérdida para hacer realidad el Internet de las Cosas, utilizando para ello plataformas de hardware libre, dando énfasis a las particularidades de los sistemas embebidos, tales como un limitado número de operaciones, bajo consumo energético y baja capacidad del ancho de banda.

Abstract—Today, the technological devices are increasingly and they are incorporated in daily life. The technological devices give to us an unusual degree of connectivity, for example: *tablets*, *smartphones*, or even everyday items, such as a refrigerator. In this context, the humanity begins to live with a network of interconnected devices that is known as the “Internet of Things”, which will be the next technological revolution, bringing new opportunities and challenges for the users of technology, business and society.

The objective of this work is to communicate the

devices through 6LoWPAN, a standard based on IPv6 over low-power and lossy networks for realizing the Internet of Things, using for it open hardware platforms, emphasizing the peculiarities of embedded systems, such as a limited number of operations, low power consumption and low capability of the band width.

Keywords— 6LoWPAN; Arduino; IEEE 802.15.4; Internet de las Cosas; IPv6; sistemas embebidos; Wireless Sensor Network; XBee

I. INTRODUCCIÓN

Hoy en día los sistemas embebidos se han masificado enormemente, al punto de convertirse en parte esencial de nuestras vidas. Este hecho se ve reflejado, por ejemplo, en los teléfonos inteligentes que integran cada vez mayores funcionalidades que requieren conectividad a Internet. Sin embargo, esta necesidad también está siendo adquirida por aparatos de uso cotidiano, tales como electrodomésticos, prendas de vestir, dispositivos vehiculares, entre otros objetos. En este escenario de alta conectividad surge un nuevo concepto llamado el “Internet de las Cosas” o IoT (*Internet of Things*), el cual describe una red de diferentes objetos de uso cotidiano interconectados entre sí.

Este nuevo escenario ha sido evidenciado por la Comisión Europea, la cual ha dispuesto de un

plan de acción, ya que el Internet de las Cosas será una realidad dentro de los próximos años que modificará drásticamente la manera en que se mueve nuestra sociedad [1].

La forma en que los objetos se conectarán hacia Internet será a través de sistemas embebidos, los cuales realizan un conjunto limitado de operaciones, habiendo visto incrementadas sus capacidades con el paso de los años, y dadas sus características no requieren un uso intensivo de la red para enviar información.

Por otra parte, el protocolo IPv6 posee una mayor capacidad de direccionamiento que su predecesor IPv4, pasando desde los 32 bits hasta los 128 bits, lo que posibilita la interconexión end-to-end de novedosos dispositivos inalámbricos de baja potencia con capacidades de procesamiento limitadas, llamados *The Wireless Embedded Internet*.

De esta forma, lo que se pretende en el presente trabajo es implementar un escenario de red 6LoWPAN (*IPv6 over Low-Power Wireless Personal Area Networks*), el cual es un estándar basado en IP sobre redes de bajo consumo y pérdida o LLN (*Low power and Lossy Networks*), para así hacer realidad la tecnología que sustenta el Internet de las Cosas. La evolución de 6LoWPAN parte cuando se estandariza el 802.15.4 en el año 2003, el cual se alza como el primer estándar de radio de bajo consumo, aunque en primera instancia presentó dificultades de escalabilidad, capacidad de evolución e integración con Internet que han sido tratadas durante los últimos años [2].

La irrupción del Internet de las Cosas implica mejoras en los procesos de captura de datos, los que se verán reflejados en ámbitos tan diversos como: sistema de monitoreo del medio ambiente, fábricas eficientes, logística, ahorro de energía, cuidado de la salud, hogares inteligentes, entre otras aplicaciones que contribuyen a aumentar la calidad de vida de las personas.

II. PROBLEMÁTICA

La visión del Internet de las Cosas involucra a dispositivos embebidos o *smart objects* que cuentan con el protocolo IP habilitado para ser parte integral de Internet. En este escenario, el protocolo IPv6 resuelve los problemas asociados al agotamiento de direcciones IPv4, tal como se observa en la Figura 1 donde en algunas zonas, tales como LACNIC (*Latin America and Caribbean Network Information Centre*) se está pronto a entrar en una situación de “agotamiento virtual”, bajo la cual los registros regionales no continúan asignando direcciones IPv4 en función de la necesidad, sino que se establece

un límite máximo por organización (/22) [3]. En este contexto, el direccionamiento a través de IPv6 permitirá dar conectividad a trillones de dispositivos, brindando soluciones en diversos ámbitos [4], tales como:

- Automatización de edificios
- Monitoreo de las condiciones climáticas
- Monitoreo de la salud estructural de puentes
- Automatización industrial
- Sensores personales
- Control logístico

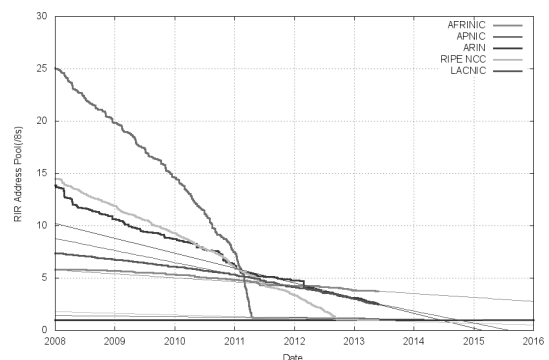


Figura 1. Agotamiento de direcciones IPv4

En lo que respecta a los objetos de la nueva Internet, con el paso de los años, las capacidades del procesador de los dispositivos embebidos han aumentado, y también la complejidad de las normas de comunicación, servicios y protocolos, algunos de ellos propietarios, tales como ZigBee, Z-Wave o HomePlug, lo cual dificulta la integración con redes más grandes o con servicios basados en Internet. Los beneficios que se desprenden de la utilización de protocolos abiertos para Internet radican en que es posible conectar dispositivos fácilmente a otras redes usando la infraestructura actual.

Sin embargo, dada la situación actual, la comunicación directa con redes IP tradicionales requiere muchos protocolos y a menudo un Sistema Operativo para hacer frente a la complejidad y mantenibilidad planteada, lo que no siempre está disponible en los sistemas embebidos desarrollados hasta la fecha.

Últimamente los dispositivos electrónicos se han vuelto cada vez más baratos, jugando un rol omnipresente en la vida diaria. De esta manera, hoy es posible construir un sistema integrado del tamaño de una billetera con la capacidad equivalente a la de un computador de la década de los 90. Este tipo de soluciones embebidas puede ser apoyado con Sistemas Operativos de funcionalidades reducidas. Desde esta perspectiva, la aparición de redes de sensores inalámbricas o WSN (*Wireless Sensor Networks*) es esencialmente la última tendencia de la Ley de Moore hacia la

miniaturización y la ubicuidad de los dispositivos informáticos [5], lo que permitiría un intercambio mundial de información.

En relación a una WSN, por lo general un sensor inalámbrico cuenta con un detector, posee capacidad de cómputo, esta provisto de mecanismos de comunicación, tiene un actuador y componentes para energizarlo. Estos elementos están integrados en una o varias *boards*, y empaquetados en unos pocos centímetros cúbicos. De esta forma, los sensores se transforman en objetos inteligentes o *smart objects*, los cuales permiten una amplia gama de aplicaciones en áreas tales como: automatización del hogar, automatización de edificios, control de fábricas, ciudades inteligentes (*smart city*), monitoreo de la salud estructural, monitoreo de la salud de las personas, redes inteligentes, gestión de energía y transporte. Los *smart objects* pueden ser operados por baterías, si las condiciones lo permiten, y típicamente podemos identificar tres componentes: una CPU (un microcontrolador de 8, 16 o 32 bits), memoria de unas pocas decenas de kilobytes y un dispositivo de comunicación inalámbrica de baja potencia (desde unos pocos kbit/s a unos pocos cientos de kbit/s).

El desarrollo técnico de sensores de bajo coste y actuadores, en combinación con las tecnologías de comunicación de baja potencia, tales como el IEEE 802.15.4, se ha visto influido y/o afectado por la irrupción de sistemas propietarios o semi-cerrados que han dado lugar a soluciones parciales y no interoperables. Muchas de las arquitecturas no IP actuales están moviéndose hacia arquitecturas totalmente IP, debido a la complejidad inherente del desarrollo de puertas de enlace difíciles de gestionar [6].

En la Tabla I se presenta una clasificación de las tecnologías de comunicación inalámbricas [7], señalando sus principales características y aplicaciones típicas, a fin de observar las condiciones que presentan las WSN, las cuales concentrarán la atención del presente trabajo.

TABLA I. CLASIFICACIÓN DE LAS TECNOLOGÍAS DE COMUNICACIÓN INALÁMBRICAS

Clase	Data rate	Radio de cobertura	Aplicaciones típicas	Ejemplo de tecnologías
WWAN	<10Mbps	>10 km	Telefonía, Internet móvil,	GSM, UMTS, satélite
WMAN	<100Mbps	<10 km	Internet banda ancha	IEEE802.16, HIPERMAN
WLAN	<100Mbps	<100 m	LAN alámbrica	IEEE802.11, HIPERLAN/2
WPAN	<10Mbps	<10 m	transferencia de datos personales	Bluetooth, IEEE802.15.3
WSN	<1 Mbps	<1 km	Monitoreo, control	propietarias, IEEE802.15.4, RFID

Para soportar la gran cantidad de nuevas aplicaciones para *smart objects*, la tecnología de red subyacente debe ser inherentemente escalable, interoperable y con una sólida base en la normalización para apoyar la innovación a futuro. El protocolo IP ha demostrado ser estable y escalable, mientras que soporta una amplia gama de aplicaciones; además, la arquitectura en capas de IP proporciona un amplio nivel de flexibilidad para la innovación.

III. 6LoWPAN

El término 6LoWPAN hace referencia a un conjunto de estándares de Internet definidos por la IETF (*Internet Engineering Task Force*), que permiten el uso de IPv6 en redes de área personal inalámbricas de baja potencia, convirtiéndose en la clave para hacer realidad el Internet de las Cosas. En la Figura 2 son comparados los Stack IP y 6LoWPAN, introduciéndose en este último una capa de adaptación y optimización.

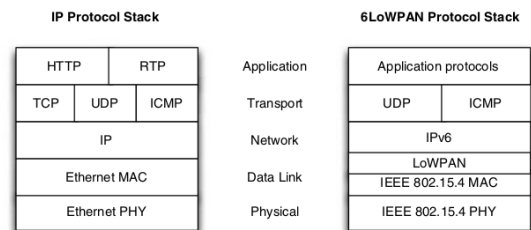


Figura 2. Stack IP y Stack 6LoWPAN

6LoWPAN rompe las barreras a la utilización de IPv6 en baja potencia, en dispositivos con procesamiento limitado a través de redes inalámbricas con un bajo ancho de banda. IPv6, definido en el RFC2460 [8], es la nueva versión del protocolo de Internet, el cual se desarrolló en la década de 1990 como una solución al rápido crecimiento y los desafíos que enfrenta Internet. El futuro crecimiento del Internet de las Cosas va a ser posible gracias a IPv6.

Los beneficios del uso de protocolos de Internet, y por lo tanto, la integración de los objetos con Internet, implica que los dispositivos se puedan conectar fácilmente con redes más grandes o con servicios basados en Internet sin necesidad de *gateways* o *proxies*. De esta forma el RFC4919 [9] define el uso de IPv6 sobre 6LoWPAN, considerando la transmisión de paquetes IP sobre el IEEE 802.15.4. En este ámbito existen ciertas cuestiones a considerar: los ciclos de duración de la batería (porcentaje de tiempo activo), el 802.15.4 no suele soportar multicast, la topología embebida de radio se beneficia de topologías tipo malla donde el enrutamiento IP no es fácilmente aplicable, el 802.15.4 posee un ancho de banda y tamaño del frame limitado. El tamaño del frame

para IPv6 puede ser de 1280 bytes, mientras que 6LoWPAN define un tamaño de 127 bytes, por lo que se estandarizó en el RFC4944 [10] una compresión de cabeceras IPv6 y UDP sobre IEEE 802.15.4 en 6 bytes, tal como se observa en la Figura 3.

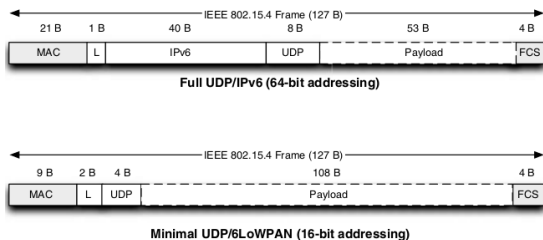


Figura 3. Compresión de la cabecera con 6LoWPAN

En cuanto al 802.15.4, en el año 2003 se alza como el primer estándar de radio de bajo consumo. En este escenario, ZigBee [11] desarrolló soluciones para el estándar, las cuales resuelven parte de la problemática, presentando problemas de escalabilidad, capacidad de evolución e integración con Internet. En la Tabla II se presenta una comparación del 802.15.4 con otras tecnologías inalámbricas.

TABLA II. COMPARACIÓN DE TECNOLOGÍAS INALÁMBRICAS

Tecnología	Rango	Velocidad	Potencia	Costo
WiFi	100 mts.	10-100 Mbps	Alto	\$\$\$
Bluetooth	10-100 mts.	1-3 Mbps	Medio	\$\$
802.15.4	10-100 mts.	0.25 Mbps	Bajo	\$

Actualmente contamos con dispositivos de corto alcance y baja potencia de radio, lo que limita las velocidades de transferencia de datos, tamaño del frame y ciclos de trabajo (energía). La estandarización del IEEE 802.15.4 fue el primer paso para la estandarización de 6LoWPAN, dados los RFC4919 y RFC4944 en el año 2007.

Cabe destacar que el desarrollo a continuado a lo largo de estos años en materias de compresión y descompresión de cabeceras, *Neighbor Discovery*, y ruteo. En este último punto en el año 2008 se forma el grupo *Routing over Low-power and Lossy Networks* o ROLL, donde también es reconocida la necesidad de estandarizar el ruteo IPv6 sobre redes de bajo consumo y pérdida [12]. El resultado del ROLL es RPL (*Ripple routing protocol*), definido en el RFC6550 [13].

IV. SISTEMAS OPERATIVOS LIGEROS PARA WSN

El Internet de las Cosas, sin lugar a dudas, es un desafío importante para la industria y para el mundo académico. En este contexto, un Stack IP implementado en los dispositivos

es esencial para hacerlos parte de la red [14]. El Stack IP debe mantener las características de interoperabilidad del protocolo, resguardando las limitaciones en términos de recursos que imponen los dispositivos embebidos. En este sentido, serán mencionadas a continuación algunas implementaciones existentes.

A. Contiki

Contiki es un Sistema Operativo multitarea para sistemas integrados y WSN (*Wireless Sensor Network*), open source y altamente portable, el cual realiza un uso eficiente de la memoria [15]. El kernel de Contiki contiene varios componentes, entre ellos μ IP, un pequeño "*RFC-compliant TCP/IP stack*" que permite la comunicación hacia Internet, junto con una implementación de IPv6 con soporte 6LoWPAN llamada μ IPv6. Contiki es implementado en una estructura en capas, donde son consideradas diferentes plataformas de hardware para implementar controladores para ellas. También provee un simulador de red denominado "Cooja", a través de una máquina virtual; Cooja permite emular nodos a nivel de hardware, con el fin de obtener una inspección del comportamiento del sistema.

B. TinyOS

El Sistema Operativo de código abierto TinyOS posee BLIP, una implementación de 6LoWPAN para WSN desarrollada por la Universidad de California, Berkeley [16]. BLIP es un Stack IPv6 que incluye compresión de cabeceras 6LoWPAN, *Neighbor Discovery*, *routing* y *network programming support*. Ha sido implementado TCP y UDP con varios protocolos de aplicación.

C. Atmel RUM

RUM o Atmel *Route Under Mac* es un pequeño protocolo IEEE 802.15.4, el cual rutea paquetes a través de capa 2 (*MAC Layer*) para microcontroladores Atmel. Presenta soporte para IPv6 y 6LoWPAN, permitiéndole a un coordinador de RUM actuar como *Border Router* [17].

V. IMPLEMENTACIÓN

Con el fin de trabajar en un escenario de red 6LoWPAN, se optó por hacer uso de un Stack IPv6 en un microcontrolador de una *board* Arduino. El Stack a implementar es el "Arduino-IPv6Stack", desarrollado por Telecom Bretagne de Francia [18], el cual portó Contiki OS a la plataforma Arduino (Figura 4). Junto a 6LoWPAN, IPv6 y RPL, se cuenta también con funcionalidades limitadas de CoAP, un protocolo de la capa de aplicación

orientado a dispositivos con recursos limitados y con acceso a Internet, tales como nodos de WSN (*Wireless Sensor Networks*) [19].



Figura 4. Arduino-IPv6Stack de Telecom Bretagne

Los recursos hardware a utilizar lo constituyen *boards* Arduino Mega 2560 R3, junto con XBee series 1, el cual proporciona IEEE 802.15.4 PHY y MAC layer, a diferencia del XBee series 2, el cual también brinda un Stack ZigBee que no es compatible con el Stack de Telecom Bretagne. La Tabla III muestra las características de los módulos XBee utilizados.

TABLA III. CARACTERÍSTICAS MÓDULO XBEE 1mW PCB ANTENNA

Velocidad de datos máxima:	115.2 kbP
Banda de frecuencia:	2.4 GHz
Potencia de transmisión:	1 mW (+0 dBm)
Rango:	100 m

Los módulos XBee fueron diseñados para cumplir con los estándares IEEE 802.15.4 y soportar las necesidades de bajo coste y bajo consumo de energía para WSN. Los módulos requieren un mínimo de energía, proporcionan entrega de datos fiable entre dispositivos, y funcionan en la banda de frecuencia de 2,4 GHz [21].

Al utilizar los módulos XBee es necesario trabajar en API mode ($AP = 2$), donde la estructura del frame acepta caracteres de escape (0x7E, 0x7D, 0x11, 0x13), de modo que no interfieran con la operación de la trama de datos.

En este escenario, el esquema dispuesto por el Telecom Bretagne prueba la interoperabilidad de dos nodos, un *Intermediate Router* y un *Border Router*. El *sketch* de Arduino considera a nodos que envían mensajes multicast y responden con la data invertida. Si la red tiene un prefijo, cambian el prefijo de la dirección IP de origen del mensaje. En caso de que el dispositivo es un *Intermediate Router*, no va a enviar mensajes multicast, sino que responderá a los mensajes recibidos (sólo si ya tienen un prefijo). En caso de que el dispositivo es un *Border Router*, nunca va a enviar mensajes o responder a ellos, pero establece un prefijo

codificado utilizado para la red.

En el nodo es cargada data de prueba, y el Stack construye el paquete, donde son comprimidas las cabeceras IPv6 y UDP del frame. La data de prueba que se envía entre nodos puede ser un mensaje cargado directamente en la aplicación, o la información capturada por un sensor conectado a la *board*.

VI. RESULTADOS

Para observar los resultados obtenidos al utilizar el Arduino-IPv6Stack, se dispone del entorno de pruebas de la Figura 5, donde la data es capturada mediante un cable RS232 conectado a un módulo XBee que actúa como “sonda” capturando paquetes, lo cual permite realizar un análisis del tráfico de paquetes en la red con una computadora que se encuentra ejecutando un script.

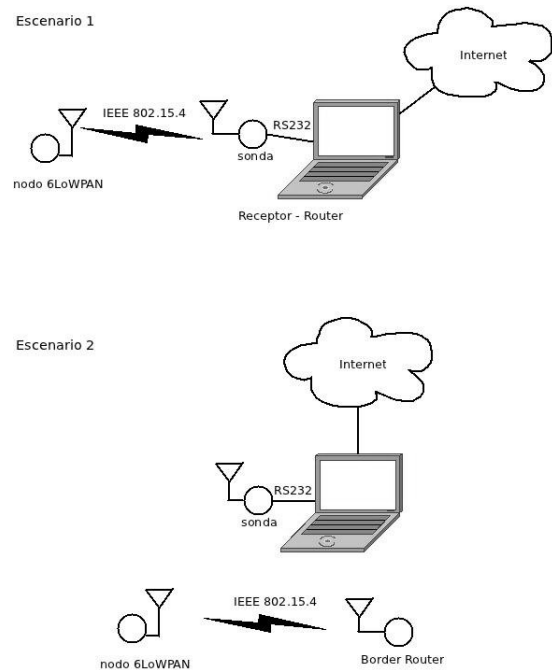


Figura 5. Entorno de pruebas

Inicialmente se considera un escenario con un nodo 6LoWPAN, siendo capturada la data que emite, con el fin de analizar los paquetes que envía y procesar la información. Luego se trabaja con un nodo 6LoWPAN y un *Border Router*, contando con una sonda para capturar los paquetes enviados entre ellos. De esta manera es posible observar el proceso de *Neighbor Discovery* de la forma en que se describe en la Figura 6.

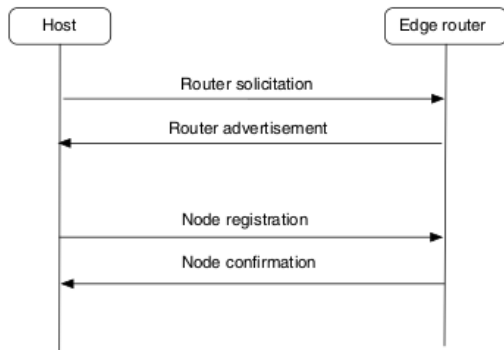


Figura 6. Neighbor Discovery en una LoWPAN

6LoWPAN comprime las cabeceras IPv6 y UDP en 6 bytes, para luego procesar la trama según el RFC4944 con el objeto de obtener la data de las cabeceras y payload. En la Figura 7 es posible observar una comparativa entre las cabeceras IPv6, UDP y 6LoWPAN comprimida. Con el fin de brindar una aproximación a grandes rasgos del formato de la cabecera 6LoWPAN/UDP comprimida, es pertinente mencionar que los dos bits más significativos del primer byte (*Dispatch Type*) indican si la cabecera corresponde a un *Normal dispatch* (01), *Mesh header* (10), *Fragmentation header* (11), o simplemente no se trata de un frame 6LoWPAN (00).

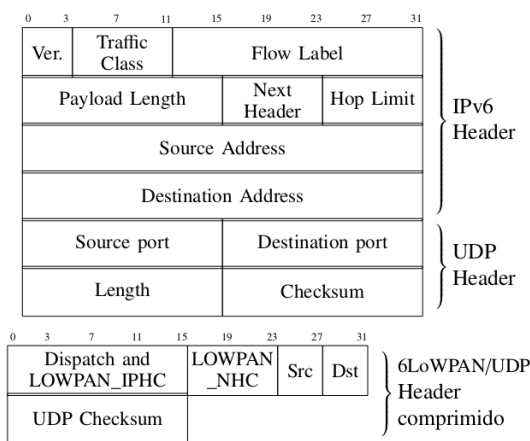


Figura 7. Header IPv6, UDP y 6LoWPAN

La sonda actúa a través de un script en Python que procesa la data capturada según la secuencia descrita en el diagrama de flujo de la Figura 8. Desde un archivo de configuración (.cfg) es leída información referente al puerto que se encuentra en uso, *baudrate*, entre otros parámetros.

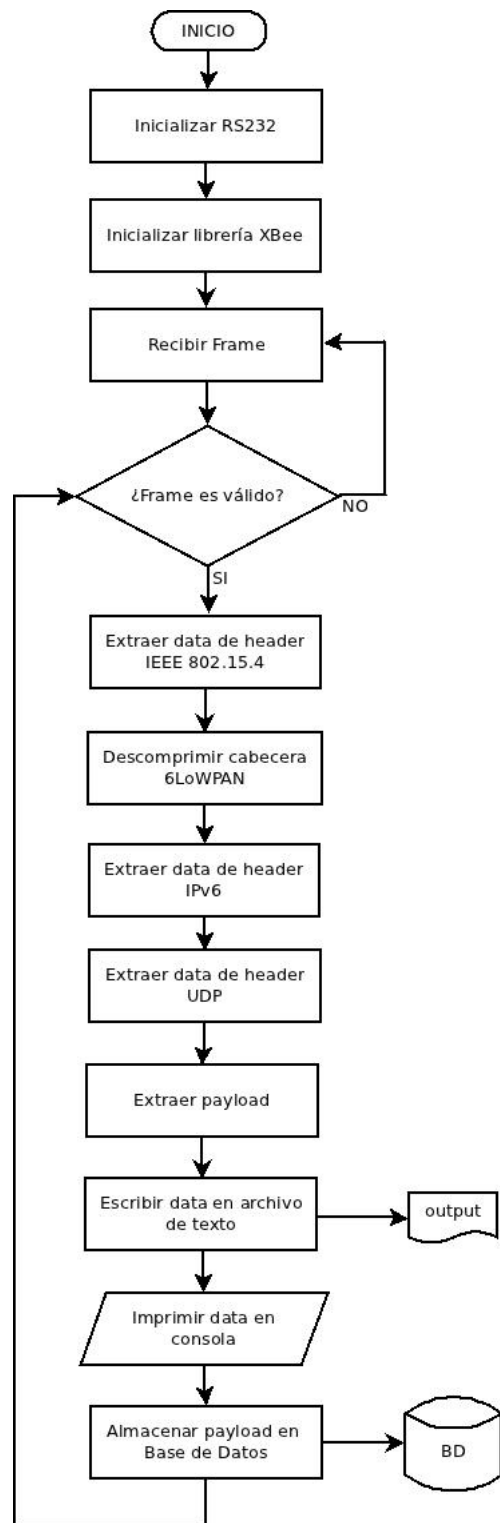


Figura 8. Diagrama de flujo de script en Python

La data capturada por la sonda en el entorno de pruebas es procesada por el script, el cual extrae la información contenida en la cabecera del IEEE 802.15.4, IPv6, y UDP respectivamente, para en última instancia obtener la carga del paquete. El script guarda la carga en una base de datos MySQL, genera un archivo con data de las cabeceras y tiempos de llegada entre paquetes, además de desplegar en consola la data deseada. Una salida del script en consola es mostrada a continuación:

```

---- 802154 Header ----
{'rssi': '4',
'id': 'rx_long_addr',
'options': '\x02'}
---- IP Header ----
{'Source Address': 'fe800000000000000213a20040484e97'
'Destination Address': 'ff020000000000000000000000000001',
'Payload Length': '0012',
'Next Header': '11'}
---- UDP Header ----
{'Destination port': '223d',
'Length': '0012',
'Source port': '223d',
'Checksum': '85ce'}
---- Payload ----
0123456789

```

Al analizar los archivos de texto generados por el programa, trabajando con un tiempo de envío entre paquetes de 1 segundo entre nodos, es posible determinar que, desde la Solicitud de Router del nodo hasta que se envía la data, transcurren en promedio 10.46 segundos. Los tiempos obtenidos en las salidas son presentados en el gráfico de la Figura 9.

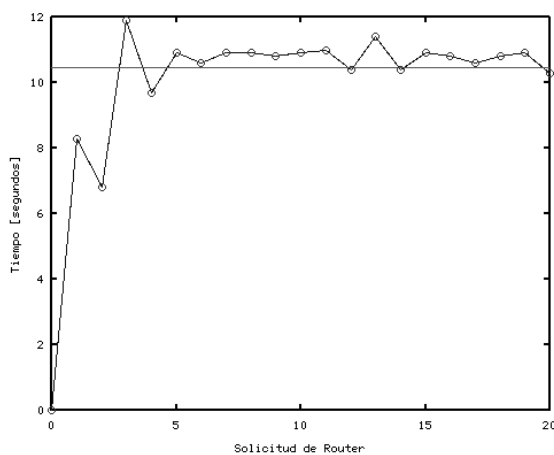


Figura 9. Tiempo desde Solicitud de Router hasta envío de data

VII. APLICACIÓN

Con el fin de acreditar el funcionamiento del escenario de pruebas construido, se posicionó un sensor acelerómetro ADXL335 en un nodo para registrar las vibraciones de una estructura, tal como se puede observar en la Figura 10. De esta manera el sensor puede contribuir al monitoreo de la salud estructural de un puente, realizando muestreos y enviando valores promedio, mínimos y máximos de cada eje.

Además, según la tasa de muestreo del sensor, podemos contar con almacenamiento masivo de un gran volumen de data (*“Big Data”*), lo cual sugiere la apertura de nuevos horizontes a investigar, planteando nuevos desafíos en torno a la gran cantidad de información que se genera, e intenta acceder, de forma paralela desde distintas fuentes (al contar con un mayor número de sensores), lo que puede provocar problemas en los canales de comunicación, en los dispositivos que procesan la data, y a nivel de Base de Datos.

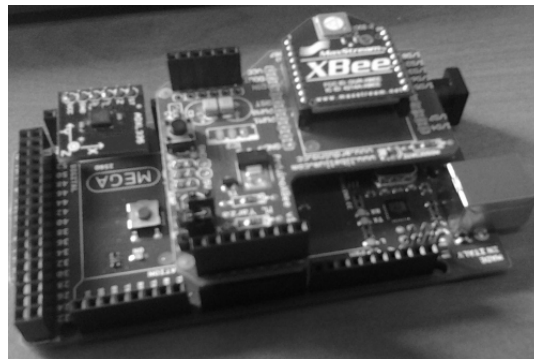


Figura 10. Nodo Arduino con Acelerómetro

VIII. CONCLUSIONES

6LoWPAN hace posible el uso del IEEE 802.15.4 para el Internet de las Cosas, logrando establecer un enlace con el protocolo IP habilitado, lo cual proporciona interoperabilidad para sistemas abiertos de dispositivos de baja potencia.

Además, se obtiene gran capacidad para trabajar dentro de las limitaciones que plantean los sistemas embebidos en cuanto a recursos: baja potencia, memoria limitada y bajo ancho de banda para *Wireless Sensor Networks*.

El Internet de las Cosas se considera la próxima gran oportunidad y desafío para la comunidad, usuarios de tecnología, empresas y sociedad, mientras que 6LoWPAN es la tecnología que hace posible esta pronta realidad.

Socialmente, la irrupción del Internet de las Cosas implica una revolución, un cambio en la forma en que interactuamos con nuestro entorno, con nuestros pares y con nosotros mismos. La comunicación con los objetos depende de la convergencia entre redes de bajo consumo y pérdida e IPv6, e implica una reivindicación de las cosas que nos rodean, las cuales se relacionarán con nosotros de una forma hasta ahora insospechada.

La utilización de sensores que implementen 6LoWPAN permitiría optimizar y/o automatizar procesos, así como también establecer canales de comunicación a través de la infraestructura actual de la red entre objetos y personas, o entre objetos.

También, se ha evidenciado que la tendencia de la nueva Internet tiene la capacidad de transformarse prontamente en una realidad con el potencial para ser aplicada a ámbitos tales como el monitoreo de estructuras, monitoreo de medio ambiente, ahorro de energía, fábricas más eficientes, mejor logística, cuidado de la salud y hogares inteligentes.

AGRADECIMIENTOS

Al profesor Christian Lazo del Instituto de Informática de la Universidad Austral de Chile por su apoyo y guía al llevar a cabo la investigación, y a Baptiste Gaultier de Telecom Bretagne por responder amablemente las consultas referentes al Arduino-IPv6Stack.

REFERENCIAS

- [1] Commission of the European Communities, “*Internet of Things - An action plan for Europe*”. Bruselas, June 2009.
- [2] Z. Shelby and C. Bormann, “*6LoWPAN: The Wireless Embedded Internet*”. Great Britain: Wiley, 2009.
- [3] Latin America and Caribbean Network Information Centre (2012). “*Estado de IPv4 a fin de 2012*” [Online]. Available: <http://portalipv6.lacnic.net/estado-de-ipv4-a-fin-de-2012-es/>
- [4] C. Taffernaberry, “*6LoWPAN: IPv6 for Wireless Sensor Network*”. SASE, Mendoza, 2012.
- [5] Y. Yu, VK. Prasanna and B. Krishnamachari, “*Information Processing and Routing in Wireless Sensor Networks*”. Singapore: World Scientific Publishing Co. Pte. Ltd., 2006.
- [6] A. Dunkels and JP. Vasseur, “*IP for Smart Objects*”. Internet Protocol for Smart Objects Alliance, July 2010.
- [7] M. Kuorilehto, M. Kohvakka, J. Suhonen, P. Hamalainen, M. Hannikainen and TD. Hamalainen, “*Ultra-Low Energy Wireless Sensor Networks in Practice*”. Great Britain: Wiley, 2007.
- [8] S. Deering and R. Hinden, “*Internet Protocol, Version 6 (IPv6) Specification*”. RFC 2460, Internet Engineering Task Force, December 1998 [Online]. Available: <http://www.ietf.org/rfc/rfc2460.txt>
- [9] N. Kushalnagar, G. Montenegro and C. Schumacher, “*IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals*”. RFC 4919, Internet Engineering Task Force, August 2007 [Online]. Available: <http://www.ietf.org/rfc/rfc4919.txt>
- [10] G. Montenegro, N. Kushalnagar, J. Hui and D. Culler, “*Transmission of IPv6 Packets over IEEE 802.15.4 Networks*”. RFC 4944, Internet Engineering Task Force, September 2007 [Online]. Available: <http://www.ietf.org/rfc/rfc4944.txt>
- [11] ZigBee Alliance, *ZigBee Specification* [Online]. Available: <http://www.zigbee.org/>
- [12] JP. Vasseur, N. Agarwal, J. Hui, Z. Shelby, P. Bertrand and C. Chauvenet, “*RPL: The IP routing protocol designed for low-power and lossy networks*”. Internet Protocol for Smart Objects Alliance, April 2011.
- [13] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, JP. Vasseur and R. Alexander, “*RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks*”, RFC 6550, Internet Engineering Task Force, March 2012 [Online]. Available: <http://www.ietf.org/rfc/rfc6550.txt>
- [14] YC. Ma, X. He, Z. Cao and H. Deng, “*Stack Analysis for Lightweight IP Implementation*”, draft-ma-lwip-stack-analysis-00, Internet Engineering Task Force, October 2010 [Online]. Available: <http://tools.ietf.org/id/draft-ma-lwip-stack-analysis-00.txt>
- [15] Contiki, “*The Open Source OS for the Internet of Things*” [Online]. Available: <http://www.contiki-os.org/>
- [16] TinyOS, “*BLIP Tutorial*” [Online]. Available: <http://www.tinyos.net/>
- [17] Atmel, “*AVR2070: Route Under MAC (RUM) with IPv6 and 6LoWPAN*”, Atmel, 2009.
- [18] Telecom Bretagne, “*Arduino-IPv6Stack*” [Online]. Available: <http://github.com/telecombretagne/Arduino-IPv6Stack/wiki>
- [19] Z. Shelby, K. Hartke and C. Bormann, “*Constrained Application Protocol (CoAP)*”, draft-ietf-core-coap-17, Internet Engineering Task Force, May 2013 [Online]. Available: <http://tools.ietf.org/id/draft-ietf-core-coap-17.txt>
- [20] A. Oyarce, P. Aguayo and E. Martin, “*Guía del Usuario XBEE Series 1*”. Santiago: Ingeniería MCI LTDA., July 2010,
- [21] Digi, “*XBeeTM/XBee-PROTM OEM RF Modules*”. XBee Series 1, October 2006.
- [22] M. Margolis, “*Arduino Cookbook*”. O’Reilly, 2011.
- [23] J. Hui, D. Culler and S. Chakrabarti, “*6LoWPAN: Incorporating IEEE 802.15.4 into the IP architecture*”. Internet Protocol for Smart Objects Alliance, January 2009.
- [24] R. Faludi, “*Building Wireless Sensor Networks, with ZigBee, XBee, Arduino, and Processing*”. O’Reilly, 2010.
- [25] T. Ziadé, “*Expert Python Programming: Best practices for designing, coding, and distributing your Python software*”. Packt Publishing, 2008.

Anexo B: Sketch del nodo

```
/*  
  Arduino IPv6 stack example
```

```
This sketch connects two Arduino Mega with wireless shield and Xbee  
mounted. This sketch demonstrates use of the IPv6 stack library.
```

```
Circuit:
```

```
* Arduino Mega 2560  
* Wireless shield w/ Xbee Series 1
```

```
created 29 June 2012
```

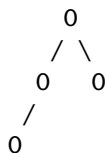
```
by Alejandro Lampropulos (alejandro.lampropulos@telecom-bretagne.eu)  
Telecom Bretagne Rennes, France
```

```
Redistribution and use in source and binary forms, with or without  
modification, are permitted provided that the following conditions  
are met:
```

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the Institute nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

```
THIS SOFTWARE IS PROVIDED BY THE INSTITUTE AND CONTRIBUTORS ‘‘AS IS’’ AND  
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE  
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE  
ARE DISCLAIMED. IN NO EVENT SHALL THE INSTITUTE OR CONTRIBUTORS BE LIABLE  
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL  
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS  
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)  
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT  
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY  
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF  
SUCH DAMAGE.
```

```
/*-----  
This example test the interoperability of different Arduino Mega + Xbee S1  
to send messages and respond them inverted. The example was tested within a  
scenario of 4 nodes (a root, an intermediate router node, and two leaves):
```



```
In this case, the start sending multicast messages and respond to received  
messages with the data inverted. If they get a prefix from a DIO, they  
change the prefix of the source ip address of the message received and they  
send it to that address, inverted.
```

```
This allows to see the routing of the router nodes.
```

```
In case the device is an intermediate router, it will not send broadcast  
messages but it will answer to received messages (only if it has already  
got a prefix)
```

```
In case the device is a border router (root of the DoDAG), it will never  
send messages or answer to them, but it sets a hardcoded prefix used for
```

the network.

This work has been partly funded by French Agence Nationale de la Recherche
under contract ANR-09-VERS-017 ARESA2

Modificaciones introducidas para conectar un sensor acelerometro en el nodo
para monitorear la salud estructural de un puente
con: Arduino Mega 2560 R3, XBee series 1

Autor: Paulo A. Gallardo Casanova, gallardo.casanova@gmail.com
Universidad Austral de Chile - Valdivia
Septiembre de 2013

-----*/

```
#include <IPv6Stack.h>
#include <XBeeMACLayer.h>

// FOR INTERMEDIATE ROUTERS: 1, FOR NODES: 0 -> UIP_CONF_ROUTER MUST BE 1
#define IS_INTERMEDIATE_ROUTER (UIP_CONF_ROUTER && 0)

#define IS_BORDER_ROUTER (UIP_CONF_ROUTER && !IS_INTERMEDIATE_ROUTER)

#define UDP_PORT 8765

#define SEND_INTERVAL (1 * 1000) // enviar mensaje cada 1 segundo

/* inicializar pines para lectura del puerto analogo - acelerometro
El circuito:
analog 8: self test
analog 9: z-eje
analog 10: y-eje
analog 11: x-eje
analog 12: ground
analog 13: vcc
*/
const int groundpin = A12;
const int powerpin = A13;
const int xpin = A11; //eje x del acelerometro en analog 11
const int ypin = A10; //eje y del acelerometro en analog 10
const int zpin = A9; //eje z del acelerometro en analog 9
char temporal[4]; // variable temporal para construccion de mensaje
char mensaje[60]; // mensaje para ser enviado
char separador[]="a"; // caracteres separadores en mensaje

// variables globales para muestreo en un eje
int num_muestreos = 1800; // numero de muestreos a 60Hz en 30 segundos
int buffer_muestreo[1800]; // arreglo con valores muestreados
int valor_indice = -30; // variable temporal
int debug = 1; // activacion de debug=1 / cancelar debug=0

//This function calculates the RAM memory left
int mem(){
    uint8_t * heapptr, * stackptr;
    stackptr = (uint8_t *)malloc(4);
```

```

    heapptr = stackptr;
    free(stackptr);
    stackptr = (uint8_t *) (SP);
    return stackptr - heapptr;
}

/*
We need a specific object to implement the MACLayer interface (the virtual
methods). In this case we use XBee but could be anyone capable of
implementing that interface
*/
XBeeMACLayer macLayer;

//We use a buffer in order to put the data that we will send
#define UDP_MAX_DATA_LEN 100
char udp_send[UDP_MAX_DATA_LEN];

//This will be the destination IP address
IPv6Address addr_dest;

#if IS_BORDER_ROUTER
/*
If we are configured to be a router, this is our prefix. Up to now we
hardcode it here. Note: The aaaa::/64 prefix is the only known by the
6LoWPAN context so it will achieve a better compression.
*/
IPv6Address prefix(0xaa, 0xaa, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0);
#endif

/*
We use a timer in order to send data if we do not have to respond to
a message received.
*/
IPv6Timer send_timer;

char udp_data[UDP_MAX_DATA_LEN];
int udp_data_length = 0;
IPv6Address sender_address;
uint16_t dest_port = UDP_PORT;

/*
changes the /64 prefix of the given address to the value of the prefix
that was given to us by a router (if any). Returns true if we have an
address with a prefix like this and false if we did not get a prefix
and thus a global address
*/
bool change_addr_prefix(IPv6Address &address){
    IPv6Address globalAddress;
    /*
we get our global address' prefix, given by our router, to send
the response to the sender using global addresses
(this will allow us to see routing)
*/
    if (IPv6Stack::getGlobalPreferredAddress(globalAddress)){
        address.setAddressValue(globalAddress.getAddressValue(0), 0);
        address.setAddressValue(globalAddress.getAddressValue(1), 1);
        address.setAddressValue(globalAddress.getAddressValue(2), 2);
        address.setAddressValue(globalAddress.getAddressValue(3), 3);
        return true;
    }
}

```

```

    }
    return false;
}

/*
This is what we will run as a result of receiving a UDP message.
We first use the serial port to show what we received, wait half
of the sending time and send the response to the sender that is,
actually, the same message inverted
*/
void udp_callback(char *data, int datalen, int sender_port,
IPv6Address &sender_addr){
    Serial.println(mem());
    delay(100);
    //Show received data
    data[datalen] = 0;
    Serial.println();
    Serial.println();
    Serial.print("Data received from ip: ");
    sender_addr.print();
    Serial.print(" port: ");
    Serial.print(sender_port);
    Serial.print(", data: ");
    //could do it this way too, but we try the other way just to test
    //Serial.println(data);
    //Show data
    while (IPv6Stack::udpDataAvailable()){
        Serial.print(IPv6Stack::readUdpData());
    }
    Serial.println();

/*
    In case we have been given a prefix by a router and we have validated
    our global address, change the address of the sender by changing its
    prefix. It is related to the same router, messages will go through it.
    Routers do not respond until they have their prefix
    (Border Routers never respond)
*/
    if (!change_addr_prefix(sender_addr)){
        #if IS_INTERMEDIATE_ROUTER
            return;
        #endif
    }

    delay(SEND_INTERVAL/2);//take SEND_INTERVAL/2 to respond

    Serial.println("Sending response..");
    Serial.println();
    delay(50);
    int j;
    for(j=0; j<datalen; ++j){
        udp_send[datalen-1-j] = data[j];
    }
/*
    now if our message is not responded, it will be resent from
    the main loop to the same destination
*/
    addr_dest = sender_addr;
    dest_port = sender_port;
    IPv6Stack::udpSend(sender_addr, sender_port, udp_send, datalen);

```

```

    delay(50);
    /*
    each time we receive something we reset this timer so we should
    not send in the main loop as long as our message is responded
    */
    send_timer.restart();
}

// funcion para realizar muestreos de un eje del sensor acelerometro
void muestreoAcelerometro(){
    // realizar muestreos
    for(int i=0; i<num_muestreos; i++){
        buffer_muestreo[i] = analogRead(zpin);
        if(debug == 1) Serial.println(buffer_muestreo[i]);
        delay(16);
    }
}

void setup(){
    Serial.begin(9600);
    delay(1000);
    Serial.println();
    Serial.print("MEMORY LEFT:");
    Serial.println(mem());
    delay(100);

    // init network-device
    if (!IPv6Stack::initMacLayer(&macLayer)){
        Serial.println("CANNOT INITIALIZE XBEE MODULE.. CANNOT CONTINUE");
        while (true){};
    }

    //init IP Stack
    IPv6Stack::initIpStack();
    Serial.println("IPV6 INITIALIZED");
    delay(100);

    //init UDP
    IPv6Stack::initUdp(UDP_PORT);
    Serial.println("UDP INITIALIZED");
    delay(100);

    //If Border Router, set prefix. If not, set a timer to send data
    #if !IS_BORDER_ROUTER
        send_timer.set(SEND_INTERVAL);
        Serial.println("SEND TIMER SET");
        delay(50);
    #else
        //This line is added to specify our prefix
        IPv6Stack::setPrefix(prefix, 64);
    #endif /*IS_BORDER_ROUTER*/

    Serial.println("SETUP FINISHED!");
    delay(100);

    // ground y power usando pines analogos
    pinMode(groundpin, OUTPUT);
}

```

```

    pinMode(powerpin, OUTPUT);
    digitalWrite(groundpin, LOW);
    digitalWrite(powerpin, HIGH);
    delay(100);
}

void loop(){
    // funcio de muestreo de acelerometro
    if(valor_indice==0) muestreoAcelerometro();

    //Always need to poll timers in order to make the IPv6 Stack work
    IPv6Stack::pollTimers();

    //If we are not a router (any kind), we also send messages
    #if !UIP_CONF_ROUTER
        if (send_timer.expired()){
            send_timer.reset();

            Serial.println();
            Serial.println("Sending data..");
            delay(50);

            // ----- construccion del mensaje a enviar -----
            // se envia tren de datos: paquetes de 9 valores cada 1 segundo
            if(valor_indice<0){
                if(debug == 1) Serial.print("delay inicio: ");
                if(debug == 1) Serial.println(valor_indice);
                valor_indice = valor_indice + 1;
            }else if(valor_indice>=0 && valor_indice<1800){
                if(debug == 1) Serial.print("Construte paquete: ");
                if(debug == 1) Serial.println(valor_indice);
                itoa(buffer_muestreo[valor_indice], mensaje, 10);
                strcat(mensaje,separador);
                for(int i=valor_indice+1; i<valor_indice+9; i++){
                    itoa(buffer_muestreo[i], temporal, 10);
                    strcat(mensaje,temporal);
                    strcat(mensaje,separador);
                }
                strcat(mensaje,separador);

                valor_indice = valor_indice + 9;
            }else if(valor_indice==1800){
                if(debug == 1) Serial.print("Fin de tren de envio: ");
                if(debug == 1) Serial.println(valor_indice);
                memset(mensaje,'\0',60); // limpiar mensaje
                valor_indice = -120;
            }

            // Enviar mensaje
            IPv6Stack::udpSend(addr_dest, dest_port, mensaje, 37);

        }
    #endif
    /*
    We always check if we got anything. If we did, process that with
    the IPv6 Stack
    */
    if (IPv6Stack::receivePacket()){
    #if !IS_BORDER_ROUTER

```

```
/*
If we are not configured as border router, check if udp data
is available and run the callback with it
*/
if (IPv6Stack::udpDataAvailable()){
    udp_data_length = IPv6Stack::getUdpDataLength();
    IPv6Stack::getUdpData(udp_data);
    IPv6Stack::getUdpSenderId(sender_address);
    udp_callback(udp_data, udp_data_length, IPv6Stack::getUdpSenderId(),
    sender_address);
}
#endif
}
}

/*-----*/
```

Anexo C: Paquetes capturados

La sonda permite capturar paquetes en el escenario de red compuesto por un nodo y un router, con direccionamiento IPv6 y utilizando el esquema 6LoWPAN para la compresión de cabeceras.

A modo de ejemplo se muestran a continuación salidas del *script*, en referencia a la información contenida en los paquetes capturados.

De esta forma, inicialmente se logra observar un paquete del nodo hacia todos los routers:

```
{'Destination Address': 'ff020000000000000000000000000002',  
  'Compress': 'No',  
  'Hop Limit': 'ff',  
  'Traffic Class': '00',  
  'Version': '6',  
  'Payload Length': '0008',  
  'Flow Label': '00000',  
  'Next Header': '3a'}
```

Una vez que la comunicación se ha establecido comienza el proceso de envío de data mediante paquetes del siguiente estilo:

```
{'Source Address': '00020010020000aaaa0000000000000002',  
  'Destination Address': '13a20040557fdc020607040002008004',  
  'Compress': 'No',  
  'Hop Limit': '00',  
  'Payload Length': '0101',  
  'Next Header': '38'}  
  
{'Destination port': '223d',  
  'Source port': '223d',  
  'Checksum': '4f3d'}
```

Payload: 574a572a569a572a572a570a565a567a571aa

Anexo D: JSON en Xively

Una solicitud (*request-json*) del eje Z configurado en m/s^2 desde la web luce:

```
{
  "id": "eje_z",
  "tags": "coordenada",
  "unit": {
    "symbol": "m/s2",
    "label": "Metros/segundo2"
  }
}
```

Realizando pruebas con un sensor acelerómetro y registrando las vibraciones de los tres ejes coordinados en fuerzas G, una respuesta (*response-json*) desde la API de Python tiene la siguiente estructura:

```
{
  "id": 1977071744,
  "title": "sensor01",
  "private": "true",
  "feed": "https://api.xively.com/v2/feeds/1977071744.json",
  "status": "frozen",
  "updated": "2013-06-17T22:24:11.131298Z",
  "created": "2013-06-17T16:27:16.177468Z",
  "creator": "https://xively.com/users/pgallardo",
  "version": "1.0.0",
  "datastreams": [
    {
      "id": "eje_x",
      "current_value": "0",
      "at": "2013-06-17T21:52:59.902191Z",
      "max_value": "0.0",
      "min_value": "0.0",
      "tags": [
        "coordenada"
      ],
      "unit": {
        "symbol": "G",
        "label": "aceleración gravitatoria"
      }
    },
    {
      "id": "eje_y",
      "current_value": "0",
      "at": "2013-06-17T22:13:57.222256Z",
      "max_value": "0.0",
      "min_value": "0.0",
      "tags": [
        "coordenada"
      ],
      "unit": {
        "symbol": "G",
        "label": "aceleración gravitatoria"
      }
    },
    {
      "id": "eje_z",
      "current_value": "0",
      "at": "2013-06-17T22:14:24.934343Z",
```

```
    "max_value": "0.0",
    "min_value": "0.0",
    "tags": [
      "coordenada"
    ],
    "unit": {
      "symbol": "G",
      "label": "aceleración gravitatoria"
    }
  }
],
"location": {
  "name": "valdivia",
  "domain": "physical"
},
"product_id": "3D05AocitJ83MFuoPqym",
"device_serial": "7NW9GMAMMA97"
}
```

Anexo E: Lista de abreviaciones

6LoWPAN: IPv6 over Low-Power Wireless Personal Area Networks

ACID: Atomicity, Consistency, Isolation, Durability

API: Application Programming Interface

BLIP: Berkeley IP Implementation

BMS: Bridge Management System

CoAP: Constrained Application Protocol

CPU: Central Processing Unit

DC: Corriente directa (direct current)

DIN: Deutsches Institut für Normung

FAN: Field Area Network

FCS: Frame Check Sequence

HTML: HyperText Markup Language

HTTP: Hypertext Transfer Protocol

ICMP: Internet Control Message Protocol

IDE: Integrated Development Environment

IEEE: Institute of Electrical and Electronics Engineers

IERC: European Research Cluster on the Internet of Things

IETF: Internet Engineering Task Force

IID: Interface IDentifier

IoT: Internet of Things

IP: Internet Protocol

IPv4: Internet Protocol version 4

IPv6: Internet Protocol version 6

IPSO: IP for Smart Objects (Alliance)

ISA: International Society of Automation

ISM: Industrial, Scientific and Medical

JSON: JavaScript Object Notation

LACNIC: Latin America and Caribbean Network Information Centre

LAMP: Linux, Apache, MySQL, PHP

LLN: Low power and Lossy Networks

LoWPAN: Low-Power Wireless Area Network

LR-WPAN: Low-rate Wireless Personal Area Network

MAC: Media Access Control layer

MAPRA: Mantenimiento de Puentes de la Red Vial Austral

MIT: Massachusetts Institute of Technology

MOP: Ministerio de Obras Públicas

NIST: National Institute of Standards and Technology

ND: Neighbor Discovery

OECD: Organisation for Economic Co-operation and Development

OS: Operating System

OSI: Open System Interconnection
PHY: Physical layer
POS: Point Of Service
RFC: Request for Comments
RFID: Radio Frequency IDentification
ROLL: Routing over Low-power and Lossy Networks
RPL: Ripple routing protocol
RSSI: Received Signal Strength Indicator
SBC: Single Board Computer
SICS: Swedish Institute for Computer Science
SNAP: Simple Network Access Protocol
SNMP: Simple Network Management Protocol
TCP: Transmission Control Protocol
UACH: Universidad Austral de Chile
UART: Universal Asynchronous Receiver-Transmitter
UDP: User Datagram Protocol
UI: User Interface
UML: Unified Modeling Language
WAN: Wide Area Network
WHAN: Wireless Home Area Network
WLAN: Wireless Local Area Network
WMAN: Wireless Metropolitan Area Network
WPAN: Wireless Personal Area Network
WSN: Wireless Sensor Network
WWAN: Wireless Wide Area Network
XML: eXtensible Markup Language