UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
Facultad de Ciencias Químicas e Ingeniería
Maestría y Doctorado en Ciencias e Ingeniería

Protocolo de Comunicación Ligero para la Integración de una Red de Sensores Inalámbricos a una Plataforma del Internet de las Cosas(IoT)

TESIS

que para obtener el grado de:

MAESTRO EN CIENCIAS

Presenta:

MARCOS CARRERA PERALTA

Director de Tesis:

DR. LEOCUNDO AGUILAR NORIEGA

Tijuana Baja California Junio 2018

Universidad Autónoma de Baja California
FACULTAD DE CIENCIAS QUÍMICAS E INGENIERÍA

FOLIO No. 255

Tijuana, B. C., a 7 de junio de 2018

C. Marcos Carrera Peralta
Pasante de: Maestro en Ciencias
Presente

El tema de trabajo y/o tesis para su examen profesional, en la Opción TESIS

Es propuesto, por el C. Dr. Leocundo Aguilar Noriega

Quienes serán los responsables de la calidad de trabajo que usted presente, referido al tema Protocolo de Comunicación Ligero para la Integración de una Red de Sensores Inalámbricos a una Plataforma del Internet de las Cosas (IoT). el cual deberá usted desarrollar, de acuerdo con el siguiente orden:

I.- INTRODUCCIÓN
II.- ANTECEDENTES Y MARCO TEORICO
III.- uIoTP: PROTOCOLO DE COMUNICACIÓN PROPUESTO
IV.- EXPERIMENTACIÓN Y RESULTADOS
V.- CONCLUSIONES Y TRABAJO FUTURO

UNIVERSIDAD AUTONOMA DE BAJA CALIFORNIA

Dr. José Luis González Vázquez
Sub-Director Secretario

Dr. Leocundo Aguilar Noriega
Director de Tesis

FACULTAD DE CIENCIAS QUIMICAS E INGENIERIA

Dr. Luis Enrique Palafox Maestre
Director

Agradecimientos

Agradesco al Dr. Leocundo Aguilar por la oportunidad de ingresar al laboratorio de STR donde pude aprender y hacer lo que de verdad me gusta, a él y al Profesor Jesus Garcia les agradezco sus enseñanzas, su paciencia, y su disposición para compartir el conocimiento con sus estudiantes.

A mi familia por todo el apoyo brindado durante todos estos años ya que sin ellos esto no hubera sido posible.

Marcos Carrera Peralta

Resumen

El Internet de las Cosas(IoT) es un paradigma que esta cobrando gran impulso en el ámbito tecnológico, por lo que Empresas e Instituciones Educativas estan dedicando esfuerzos para abordar esta nueva revolución tecnológica como muchos la llaman, el enfoque de esta tecnología consiste en conectar objetos de la vida cotidiana a internet para poder ser accedidos desde cualquier parte y en cualquier momento mediante una dirección única. Esta es una tarea que involucra multiples tecnológias partiendo desde los sensores y actuadores, redes de sensores, protocolos de comunicacion, almacenamiento de la información, procesamiento de datos, etc., en el presente trabajo se describen algunas de ellas como marco teórico pero el objetivo principal se centra en el diseño de un protocolo de comunicación que nos permita integrar dispositivos(sensores/ actuadores) de bajo poder computacional y una alimentación limitada de energía a una plataforma de IoT, ya que la tendencia es reducir el tamaño y el consumo energético de los dispositivos que seran integrados al IoT a la par es necesario mejorar los protocolos de comunicación que seran ejecutados en estos dispositivos con tal de hacer un uso eficiente de sus recursos, proveer de mayor independencia y reducir al maximo la intervención humana.

Introducción

El termino “Internet de las Cosas” (IoT; del ingles “Internet of Things”), es mencionano por primera vez en el año de 1999 por Kevin Ashton cofundador del Auto-ID Center del MIT en una presentacion para P&G con el fin de introducir el uso de RFID en la cadena de suministros de dicha empresa[1], 10 años mas tarde la idea de conectar las cosas de uso cotidiano a Internet para poder ser accedidas desde cualquier parte, en cualquier lugar y en cualquier momento mediante una dirección única cobra gran importancia debido al impacto económico, tecnológico y social que trae consigo.

Ya que IoT es un paradigma que se extiende por diversas areas entre ellas el hogar, la industria, el campo, las ciudades, entre otras, generalmente se requiere de un gran numero de dispositivos conectados entre si para abarcar areas extensas, es ahí donde entran en funcion las Redes de Sensores Inalámbricos (WSN; del ingles “Wireless Sensor Networks”) que son precisamente dispositivos(Nodos) sensores y actuadoses(SAs) que cuenta con un modulo sensor/ actuador, capacidad de computo, capacidad de comunicación y tipicamente alimentados por baterias. La mayoria de las veces los Nodos son dispositivos con recursos limitados en Hardware, en cuanto a Memoria de programa, Memoria Ram, poder de computo, ancho de banda, alimentacion limitada de baterias[2], y debido a estas circunstancias fundamental hace un uso conciente de estos recursos para prolongar su funcionamiento y sobre todo minimizar la intervencion humana en actividades de mantenimiento.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

1.1. Planteamiento del problema

En la actualidad ya existen diversos protocolos tanto para Internet como para dispositivos de bajos recursos, sin embargo no existe compatibilidad entre ellos, es decir los protocolos de Internet tales como HTTP, TCP fueron diseñados principalmente pensando en dispositivos de propósito general con recursos suficientes de hardware y alimentación electrica permanente, por otro lado los protocolos existentes para la conexión entre dispositivos de escasos recursos no están pensados para compartir datos a internet de forma directa, mucho menos tratandose de un gran número de dispositivos SAs como se espera en el IoT, observando esta necesidad se planteo el diseño de un protocolo ligero que permita minimizar el consumo de estos recursos tan limitados de hardware y energia, y ademas sea compatible con los protocolos de Internet ya existentes. La mayoria de estos protocolos fueron desarrollados sobre el protocolo TCP el cual se encuentra ubicado en la capa de transporte del modelo OSI (del ingles International Organization for Standardization, en español Modelo de Interconexión de Sistemas Abiertos)[3], TCP es un protocolo orientado a conexión, siendo un protocolo confiable que cuenta con ciertas politicas como una negociación de tres pasos, corrección de errores,

retransmision de paquetes, sin embargo estas cualidades hacen que sus encabezados sean grandes siendo muy pesado al momento de enviar y recibir información a través de la red, la propuesta es implementar un protocolo haciendo uso del protocolo UDP (del ingles User Datagram Protocol, en español Protocolo de Datagramas de Usuario)[4] el cual es un protocolo sin conexión, tambien ubicado en la capa de transporte siendo mucho mas ligero y simple que TCP, al ser un protocolo no orientado a conexión prácticamente no hace nada más que enviar paquetes entre aplicaciones, y deja que las aplicaciones construyan sus propios protocolos en la parte superior según sea necesario[5], siendo esta la función del protocolo que se propone.

1.2. Justificación

Ya que se estima un gran crecimiento en el número de dispositivos(Nodos) conectados al IoT donde cada uno de ellos es el responsable de obtiener los datos provenientes del ámbiente para posteriormente ser digitalizados y almacenados analizar la forma en la que los Nodos transportan los datos, se comunican y se agrupan entre ellos se vuelve una tarea igualmente importante, por tal razón el enfoque del presente trabajo se centra en el diseño de un protocolo de comunicación.

Para crear una WSN se requieren protocolos de comunicación ligeros, principalmente donde se tienen recursos limitados de hardware y energia por lo que un buen diseño permitira conservar dichos recursos y prolongar su tiempo de actividad, tomando en cuenta que los nodos son parte de una plataforma de IoT y contamos con cientos de ellos la intervencion humana es lo que menos se desea, otro factor muy importante es la estandarización de los protocolo para las diferentes clases de sensores [6] con el objetivo de poder integrar Nodos heterogeneos a una misma plataforma, haciendo un uso eficiente de los recursos de hardware, energia y ancho de banda.

1.3. Objetivos

General: Diseñar un protocolo de comunicación ligero que permita integrar una red de sensores inalámbricos de bajos recursos de hardware y energía a una plataforma de IoT.

Especificos: Reducir el tamaño de los encabezados. Reducir el consumo de energía mediante el protocolo. Diseñar un gateway el cual cumpla la funcion de nodo coordinador en la WSN y permita la conectividad desde y hacia internet.

1.4. Hipótesis

Mediante un protocolo de comunicación ligero se puede integrar una red de sensores inalámbricos a una plataforma de IoT con arquitectura tipo edge, logrando un mejor rendimiento de los recursos de hardware y energía de cada nodo en comparación con los protocolos ya existentes.

1.5. Estructura de la Tesis

Capítulo 2

Antecedentes y Marco teórico

En este capitulo estaremos abordando los temas relacionados al tema principal de investigación con el objetivo de tener un panorama general del mismo e irse familiarizando con los conceptos que se estaran mencionando en los siguientes capitulos.

2.1. IoT

Se estima que para el año 2020 el numero de dispositivos conectados a internet sea de 50 mil millones, este crecimiento exponencial debido a la integración de cosas de uso cotidiano mas que por el propio crecimiento poblacional, ya que dispositivos tales como refrigeradores, automoviles, luces, equipo industrial, etc., seran integrados a esta plataforma[7] como se muestra en la Figura 2.1.

2.1.1. Definiciones

Esta tendencia perfila al IoT como la siguiente Revolución Industrial, pero en realidad que es IoT, algunas de las definiciones que se encuentran en la literatura son las siguientes:

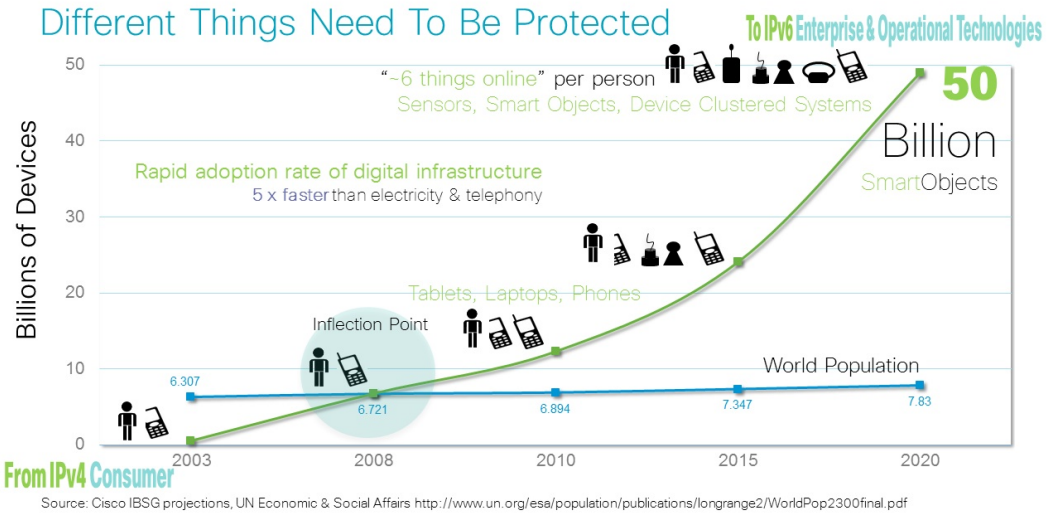

Figura 2.1: Dispositivos conectados a Internet (Fuente: Cisco).

- Una infraestructura global para la información de la sociedad, que permite servicios avanzados interconectando (físicos y virtuales) cosas basadas en tecnologías de la información, comunicadas, interoperables y en constante evolución [8].
  - A través de la explotación de capacidades de identificación, captura de datos, procesamiento y comunicación, IoT hace un uso completo de las cosas para ofrecer servicios a todo tipo de aplicaciones, al tiempo que garantiza que se cumplan los requisitos de seguridad y privacidad.
  - En una perspectiva amplia, el IoT puede percibirse como una visión con implicaciones tecnológicas y sociales.
- Tendremos conectividad desde cualquier momento, en cualquier lugar para cualquier persona, y ahora tendremos conectividad para cualquier cosa[9].
- Una infraestructura de red global dinámica con capacidades de autoconfiguración basadas en protocolos de comunicación estándar e interoperables donde las "cosas" físicas y virtuales, tienen identidades, atributos físicos y personalidades virtuales, usan interfaces inteligentes, y se integran a la perfección en la red de información[10].

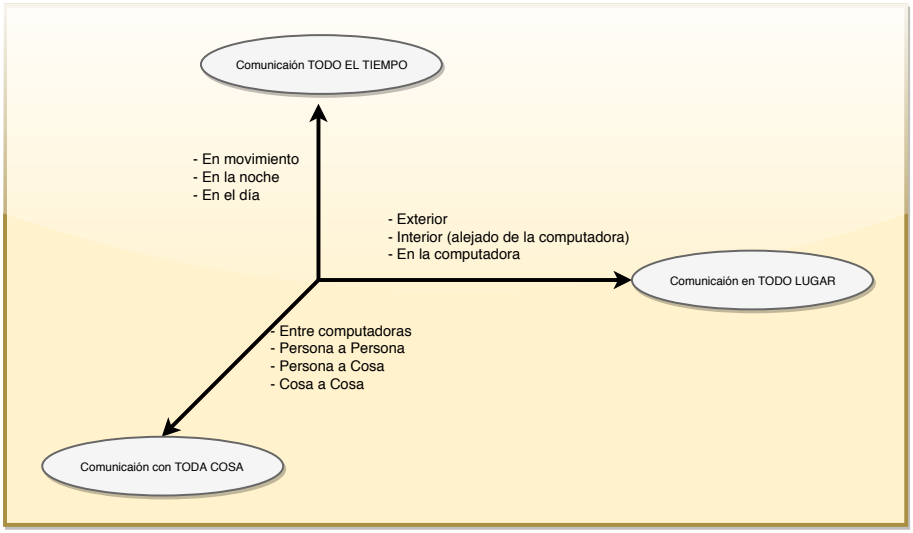

Figura 2.2: Nueva dimensión introducida en el IoT[b-ITU Report].

Como podemos darnos cuenta estas definiciones coinciden en que IoT integra dispositivos físicos y virtuales a las plataformas en Internet para poder ser administrados mediante una dirección única, con administrar nos referimos a que se tendra comunicación con toda “COSA”, “TODO EL TIEMPO” y en “TODO LUGAR”, como se muestra en la Figura 2.2[11].

2.1.2. Arquitectura

Una vez que hemos definido IoT es importante conocer su estructura, para ello nos guiamos del modelo de referencia mostrado en la Figura 2.3, la imagen a la izquierda muestra la estructura en modelo de capas[11], para hacerlo un poco mas comprensible lo trasladamos a un modelo mas visual, en el que se pueden observar los elementos que se encuentran en cada una de estas capas, a continuación se describe de una forma muy general cada una de las capas:

- **Capa de Dispositivos:**
  - Capacidades de Dispositivos: Tiene la capacidad de enviar y recibir datos a travez

del gateway o de forma directa, tambien soporta mecanismos para ahorro de energia.

- Capacidades de Gateway: Soporta conexiones mediante diferentes protocolos de comunicación alámbricos o inalámbricos.

- **Capa de Red:**
  - Capacidades de Red: Provee control sobre las funcionalidades de conectividad a la red, tales como Autentificación, Autorización y Contabilización (AAA)
  - Capacidades de Transporte: Se centran en proporcionar conectividad para el transporte de información de control y gestión relacionada con la aplicación IoT.

- **Capa de Soporte de Servicio y Soporte de Aplicación:**
  - Capacidades de Soporte Genericas: Contiene capacidades comunes usadas por diferentes aplicaciones IoT tal como procesamiento o almacenamiento de datos.
  - Capacidades de Soporte Especificas: Son capacidades particulares de cada aplicación, las cuales se agrupan y se integran para proveer soporte a funcionalidades diferentes.

- **Capa de Aplicación:** Contiene a la aplicación IoT, y a la Capa de Negocios como se muestra en el modelo gráfico de la Figura 2.3

2.1.3. Estandares

Ya que IoT es una tecnología emergente, existe un gran potencial de innovación en diversor sectores de la industria, y de la sociedad, como por ejemplo el cuidado de los recursos naturales y energeticos, las ciudades, el campo, etc. sin embargo este potencial solo puede ser explotado si los diseñadores de las plataformas IoT mantienen estas plataformas abiertas o en el mejor de los casos como codigo libre para permitir soluciones a la medida y la integracion

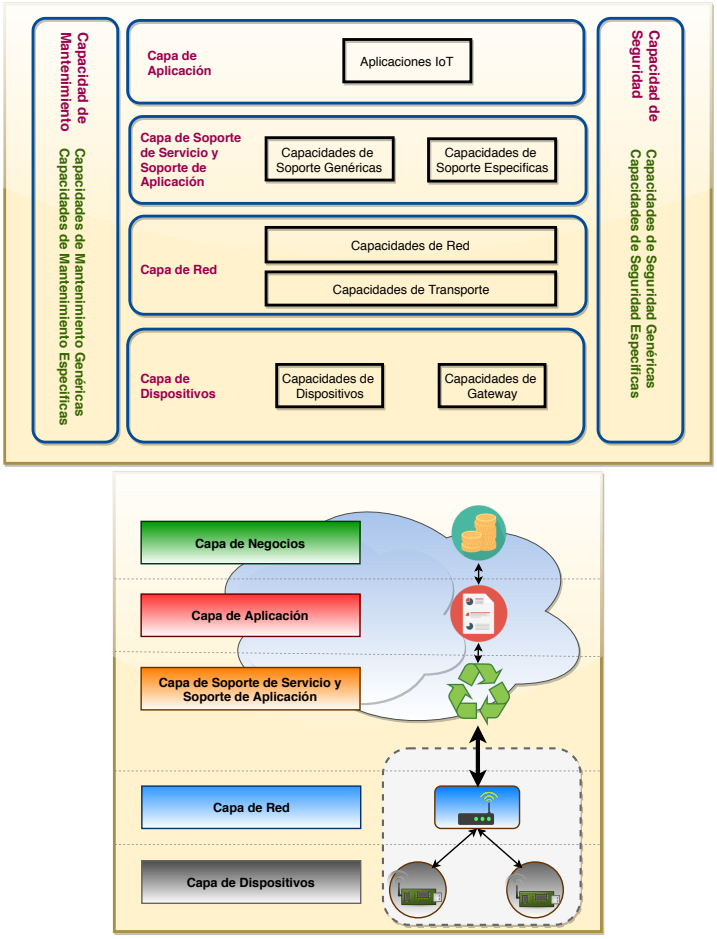

Figura 2.3: Modelo de Referencia IoT y modelo gráfico.

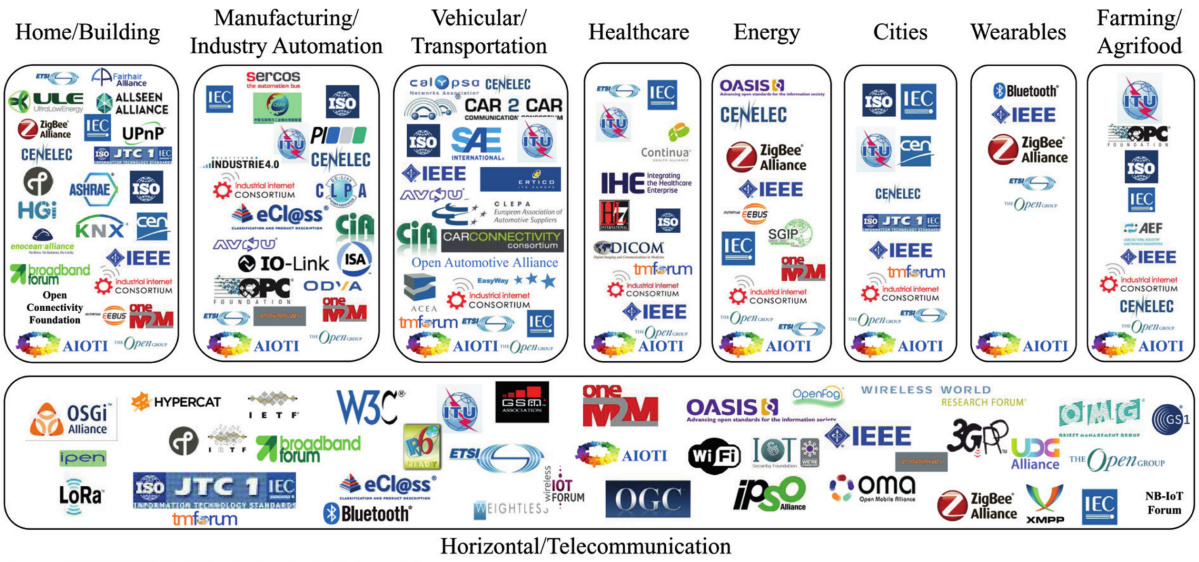

Figura 2.4: Iniciativas para la estandarizacion IoT agrupadas por dominio.

de aplicaciones y dispositivos heterogeneos, por tal razon ya han surgido iniciativas entre las empresas de manufactura y de servicios para trabajar en un estandar que permita esta integración, aunque esta interoperabilidad es todo un reto, la estandarización es el mejor enfoque para lograrlo. Actualmente existen iniciativas que trabajan de forma idividual o en alianza con organismos internacionales(e.g. ETSI SmartM2M, ITU-T, ISO, IEC, ISO/IEC JTC 1, oneM2M, W3C, IEEE, OASIS, IETF, etc.) asi tambien existen numerosas iniciativas industriales(e.g. AllSeen Alliance, Industrial Internet Consortium (IIC), Open Connectivity Foundation (OCF), Thread protocol, Platform Industrie 4.0, etc.). Las iniciativas de Estandarizacion en el panorama de IoT se pueden apreciar en dos dimensiones (dominio vertical y horizontal) como se muestra en la Figura 2.4 donde:

- Las aplicaciones IoT se representan en el dominio “vertical”.
- La infraestructura de telecomunicación se representa en el dominio “horizontal”.

2.1.4. Iniciativas de Plataformas

El objetivo de la Iniciativa de Plataformas Europeas de IoT (por sus siglas en ingles IoT-EPI: Internet of Things (IoT) European Platforms Initiative es crear un ecosistema mediante "Plataformas para conectar objetos inteligentes" para poder integrar la proxima generacion de dispositivos, dispositivos embebidos y tecnologias de red, con soporte para ciudades, negocios, con caracteristicas abiertas facilmente expandibles, interoperables con capacidades cognitivas, costo y eficiencia energetica, ergonomicos y amigables, a continuación describimos unas de estas plataformas:

- **Gateway Modular Adaptativo para IoT (AGILE Project: A Modular Adaptive Gateway for IoT)**

  AGILE es un gateway que permite integrar dispositivos IoT de forma modular y adaptativa, la modularidad se provee a nivel hardware, y soporta diferentes tecnologias de red (KNX, Z-Wave, ZigBee, Bluetooth Low Energy, etc.). A nivel software provee funcionalidades de colección de datos y mantenimiento en el gateway, interfaz intuitiva, lo que permite realizar prototipos IoT de forma rapida para diferentes dominios(hogar, automotriz, monitoreo ambiental, vestibles, etc.).

- **Reducción de la brecha de interoperabilidad del IoT (BIG IoT: Bridging the Interoperability Gap of the IoT)**

  Aun existen multiples barreras entre los desarrolladores y los proveedores de servicios ya que el mercado de las plataformas IoT esta fragmentado, esto es un problema para los desarrolladores al momento de tratar de integrar objetos que implementan tecnologias diferentes, lo cual provoca que se deba realizar la negociación individual para el acceso a cada plataforma, por lo que el objetivo de este proyecto es cerrar esa brecha mediante un mercado de servicios que provean APIs que contengan las funcionalidades de descubrimiento, acceso, control, mensajes y seguridad para cosas inteligentes.

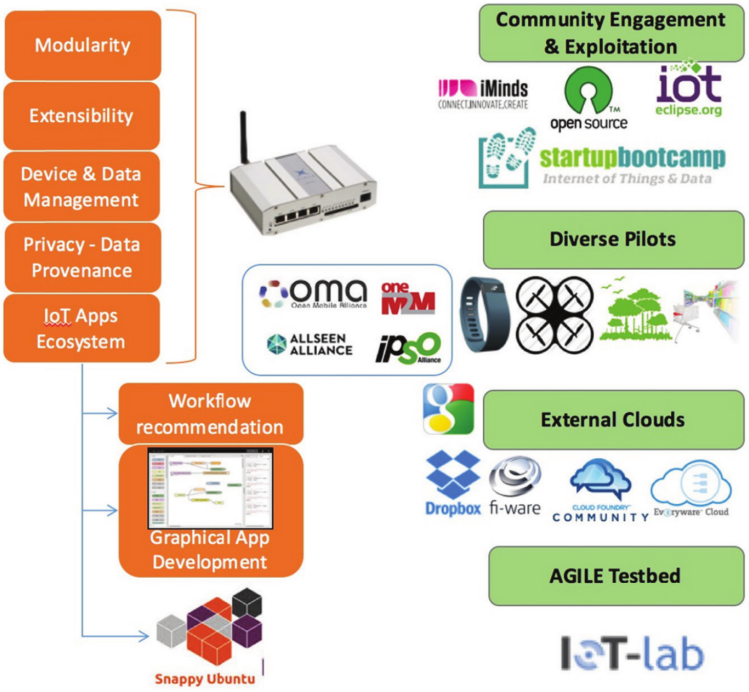

Figura 2.5: Proyecto AGILE.

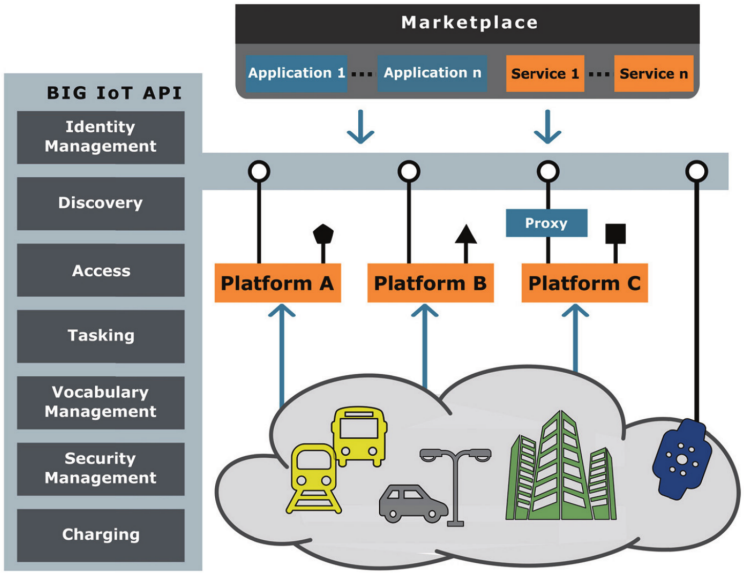

Figura 2.6: Enfoque BIG IoT.

2.2. Nodos

Para iniciar damos algunas definiciones que son de importancia para este tema[11].

- **Sensor:** Dispositivo electrónico que detecta una condición física o compuesto químico y lo transforma en una señal electrónica proporcional a la característica observada.
- **Actuador:** Un dispositivo que desencadena una acción física después de la estimulación por una señal de entrada. Por ejemplo podría actuar sobre el flujo de un gas o líquido, o electricidad, a través de una operación mecánica.
- **Redes de Sensores:** Red compuesta por nodos de Sensores interconectados que intercambian datos detectados mediante comunicación por cable o inalámbricas.
- **Nodo Sensor:** Un dispositivo que consta de uno o varios sensores o actuadores con capacidades de procesamiento, sensado, y conexión en red.
- **Nodo de bajo poder computacional ó Nodo Restringido:** Un nodo donde algunas de las características no estan disponibles para Internet, a menudo debido a restricciones de costos y / o restricciones físicas en características tales como tamaño, peso y disponibilidad de poder y energía, estos límites estrictos en energía, memoria y recursos de procesamiento conducen a límitar el espacio del código y los ciclos de procesamiento, haciendo que la optimización del consumo de energía y del ancho de banda de la red sea una consideración dominante en todos los requisitos de diseño. Además, pueden faltar algunos servicios de capa 2, como conectividad completa y difusión / multidifusión[12].
- **Red restringida:** Una red donde se carecen de algunas de las características dadas por supuestas con capas de enlace de uso común en Internet[12].

En el paradigma de IoT se tiene como objetivo interactuar con diferentes entornos tanto industriales, el hogar, el campo, las ciudades, etc., para ello se requieren multiples Nodos

sensores o Nodos Actuadores, los cuales deben tener ciertas capacidades para poder adquirir, procesar y transmitir las mediciones, el diseño del hardware del sensor depende de las necesidades de la aplicación, estos se pueden clasificar en 3 categorias[2]:

- **Computadora de propocito general:** Incluye a las computadoras de bajo poder de computo PCs, Pcs embebidas, Asistente Personal Digital(por sus siglas en ingles PDA: Personal Digital Assistans), tipicamente estos sistemas tiene un Sistema Operativo por ejemplo Linux, y gracias a su poder computacional son capaces de comunicarse mediante protocolos estandares como Bluetooth o IEEE 802.11, a su vez pueden tener varios sensores aunque esto provoca un mayor consumo de energía.

- **Nodo sensor embebido dedicado:** Uno de los ejemplos mas populares el es mote de Berkeley[13], estos son dispositivos de bajo costo, bajo poder computacional, bajo ancho de banda y una interfaz sencilla para conectar sensores, generalmente cuentan con poca memoria y algunas veces con un Sistema Operativo de Tiempo Real como TinyOS.

- **Nodo sistema en chip:** Mejor conocidos como SOC (por sus siglas en ingles System on Chip), Se trata de colocar el nodo sensor a niver del chip con el objetivo de tener un consumo de energia muy bajoy ademas tener dispositivos muy pequeños fisicamente.

Tipicamente los nodos de red son sistemas embebidos dedicados o SoC, con bajo poder de computo, normalmente su función es sensar el fenomeno fisico, procesar y enviar los datos a un gateway de forma asincrona, respondiendo a eventos, con esto se puede lograr un mejor aprovechamiento de los recursos de hardware y de energia ya que generalmente estos dispositivos son alimentados por baterias, en la Figura 2.7 podemos ver los elementos que componene un Nodo Sensor[14].

En la Figura 2.8 podemos ver la arquitectura tipica de un Nodo Sensor donde:

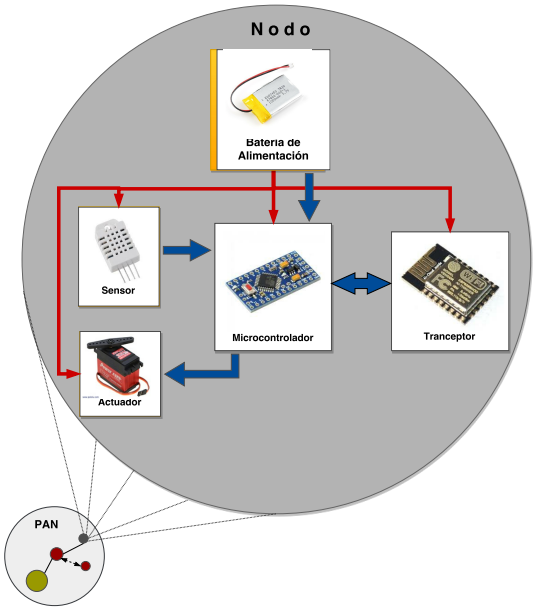

Figura 2.7: Elementos de un Nodo Sensor.

- **Microcontrolador:** Su función es procesar los datos y controlar el resto de los componenetes del nodo, y se compone de los siguientes modulos:
  - Unidad Central de Procesamiento (CPU)
  - Memoria
  - Convertidor Analogico a Digital (ADC)
  - Interfaces Digitales(i2S, UART, 1-wire, SPI, USB, GPIO, etc.)
  - Modulo de Cifrado
  - Convertidor Digital a Analogico (DAC)
  - Procesador Digital de Señales (DSP)
- **Procesador Digital de Señales(DSP):** Es un tipo de procesador que tiene como función el reconocimiento de patrones en señales de audio o video.
- **Arreglo de compuertas programables(FPGA):** Como su nombre lo dice es un arreglo de compuertas programable en un lenguaje descriptor de hardware donde se puede lograr procesamiento paralelo a frecuencias muy altas.

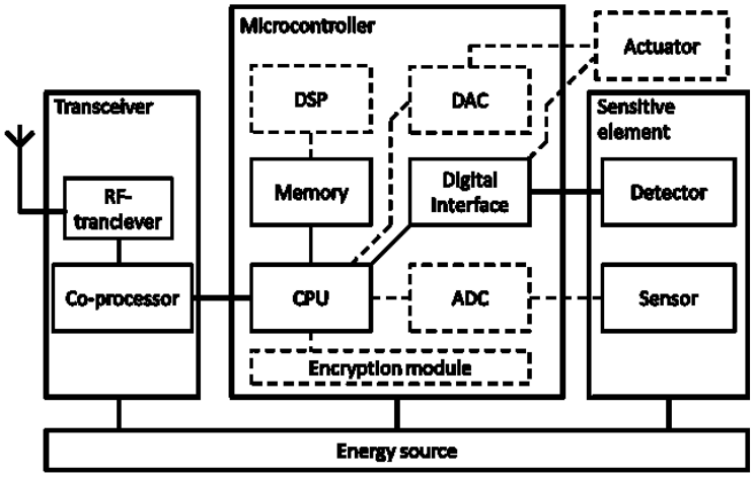

Figura 2.8: Elementos de un Nodo Sensor.

- **Radio Transmisor:** Es el encargado de transmitir y recibir los datos ya que tiene acceso al canal de transmisión.

Ya que IoT contendra Nodos de diversas caracteristicas, pero principalmente hacemos enfasis en los dispositivos de bajo poder de computo (en ingles “Constrained Devices”) en la Tabla 2.1 podemos ver una clasificacion de estos dispositivos de acuerdo a su capacidad de almacenamiento de datos[12].

Tabla 2.1: Clase de Dispositivos Restringidos (KiB = 1024 bytes)

Nombre	**Tamaño del datos (ej., RAM)**	**tamaño de programa (ej., Flash)**
Clase 0, C0	<< 10 KiB	<< 100 KiB
Clase 1, C1	$\sim$ 10 KiB	$\sim$ 100 KiB
Clase 2, C2	$\sim$ 50 KiB	$\sim$ 250 KiB

Otra limitante en los Nodos es la Energia diponible, en la Tabla 2.2 se proporciona una clasificación en base a ello[12].

Tabla 2.2: Clasificación por fuente de Alimentación de Energia

Nombre	Tipo de Energia Limitada	Ejemplo de Fuente de Alimentación
E0	Energia limitada por evento	Recoleccion bazada en eventos
E1	Energia limitada por periodo	Bateria reemplazada o recargada periodicamente
E2	Energia limitada de por vida	Bateria no reemplazable
E9	Sin limitante de energia	Alimentado por la Red Electrica

2.3. WSN

El desarrollo de Redes de Sensores Inalambricos(por sus siglas en Ingles WSN: Wireless Sensors Network) fue inspirado en aplicaciones militares, usadas en zonas de conflicto y vigilancia, esta investigación comenzo en los años 80s cuando la Agencia de Proyectos de Investigación Avanzados de Defensa (por sus siglas en Ingles DARPA: United States Defense Advanced Research Projects Agency) llevó a cabo el programa de redes de sensores distribuidos (DSN) para el ejército de los E.U. En ese momento, la Red de Agencias de Proyectos de Investigación Avanzada (ARPANET) había estado en operación durante varios años, con aproximadamente 200 anfitriones en universidades e institutos de investigación[15]. Para 1998 comienza nuevamente la ola de investigaciones y desarrollos de WSN con Nodos restringidos, del 2011 al 2016 los dispositivos inalámbricos instalados incremento en un 553 %, habiendo 24 millones de dispositivos activos[16].

Generalmente una WSN se describe como un conjunto de nodos que de forma cooperativa sensan y controlan el entorno, permitiendo la interacción entre pesonas y computadoras con el entorno, hoy en día las WSNs se componen de nodos sensores, nodos actuadores y gateways, donde los sensores son distribuidos a lo largo del área que deben monitorear siendo capaces de autoorganizarse para poder hacer llegar los datos hasta el gateway mas cercano como se ve en la Figura 2.9.

El rango de alcance de estas redes va de unos pocos metros a miles dependiendo de los

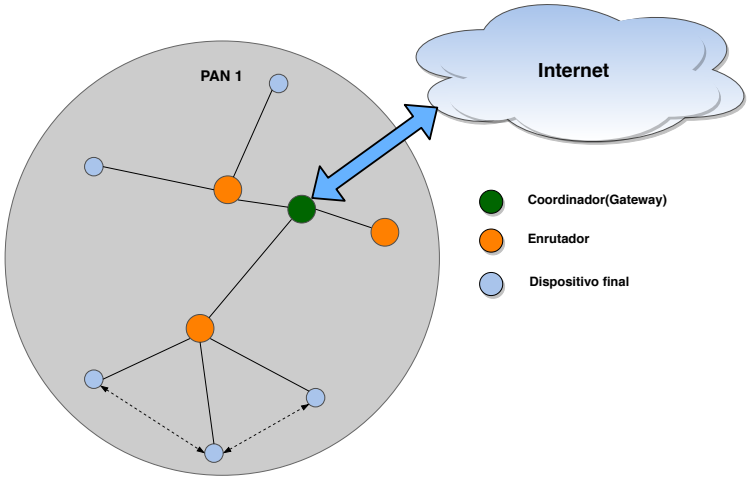

Figura 2.9: Red de Sensores Inalámbricos.

canales por los que se este transmitiendo, de acuerdo al alcance y a la velocidad de acceso las redes se pueden clasificar en cuatro categorias:

- Redes Inalámbricas de Área Local (WLAN: del ingles Wireless Local Area Network).
- Redes Inalámbricas de Área Metropolitana (WMAN: del ingles Wireless Metropolitan Area Network).
- Redes Inalámbricas de Área Personal (PAN: del ingles Wireless Personal Area Network).
- Redes Inalámbricas de Área Amplia (WWAN: del ingles Wireless Wide Area Network).

Las redes inalámbricas no son muy seguras ya que sufren interferencia por otras frecuencias, ruido en el canal, etc., aun asi son mas viables que las alambricas para espacios amplios, las tecnologias mas usadas hoy en dia son Bluetooth 4.0 para el area medica, IEEE 802.15.4e para la industria y IEEE 802.11 para IoT, debido a su facil integración a infraestructura ya existente[17].

2.3.1. Topologías

Las WSN se componen de cientos o miles de dispositivos, el diseño de estas redes se basa en las topologias que se muestran en la Figura 2.3.

Tabla 2.3: Ventajas y desventajas de las Topologías de Red

Topologia	Energia uasada	Rango de comunicación	Requiere sincronización
Estrella	Bajo	Corto	No
Arbol	Bajo	Largo	Si
Malla	Alto	Largo	No
Hibrido	Bajo tipicamente	Largo	Depende de la configuración

- **Estrella:** La principal caracteristica de esta topología es que se conectan directamente con el gateway, no existe comunicación directa entre los nodos como se muestra en la Figura 2.10 a), es ideal para conectar redes pequeñas ya que tiene un alcance de entre 10 y 100m dependiendo del radio de comunicación, y es una de las topologías que requiere menor consumo de energía.

- **Arbol:** Como podemos ver en la Figura 2.10 b) se tiene una estructura gerarquica donde el gateway es el nodo raiz ya que se encuentra en el primer nivel, a partir del segundo nivel los nodos cumplen funciones de raiz e hijos, esta topología es adecuada para un gran numero de nodos y asi poder cubrir areas amplias, se tiene un consumo moderado de energia, sin embargo se requieren operaciones de sincronización entre los nodos.

- **Malla:** En esta topología cada nodo tiene conexión con todos los nodos que se encuentran al alcance de su radio de comunicación, normalmente se usa para cubrir grandes distancias, es una topología altamente tolerante a fallos ya que existe mas de una ruta para llegar al gateway como se ve en la Figura 2.10 c).

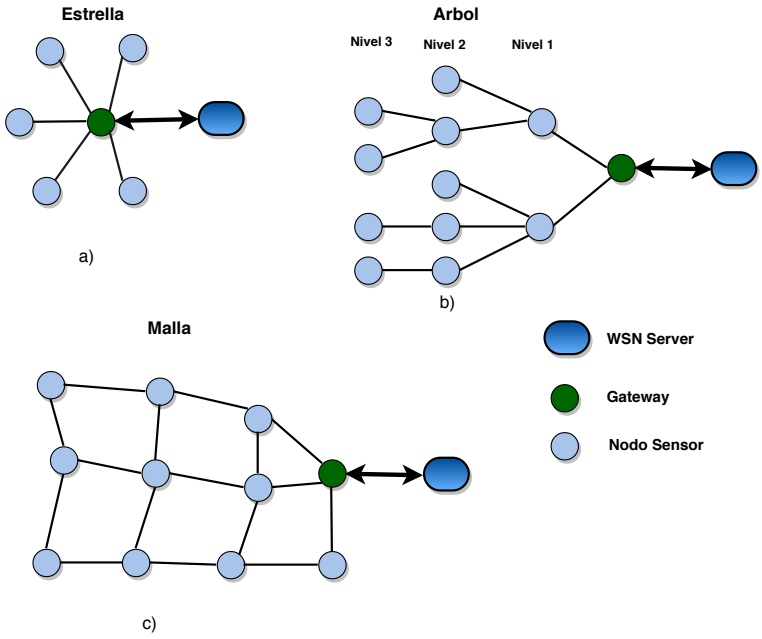

Figura 2.10: Topologías de las WSN.

- **Hibrido:** Algunas redes usan una combinación de las topologías anteriores logrando cubrir un área mas amplia y conectar cientos de nodos, logrando tener una red tolerante a fallos con un consumo moderado de energia en proporción al area cubierta, y nodos autoorganizados.

2.4. Modelos de referencia

Para reducir la complejidad del diseño del software de red la mayoría de las redes se organizan como una pila de capas o niveles, cada una construida a partir de la que está abajo. El propósito de cada capa es ofrecer ciertos servicios a las capas superiores, mientras les oculta los detalles relacionados con la forma en que se implementan los servicios ofrecidos. Estos modelos de capas son: el modelo de referencia OSI (Interconexión de Sistemas Abiertos, del inglés Open Systems Interconnection) desarrollada por la Organización Internacional de Normas (ISO) en 1995, y el modelo de referencia TCP/IP definido por primera vez en 1974, definido como estandar de internet en 1989, en la Figura 2.11 podemos ver las capas que

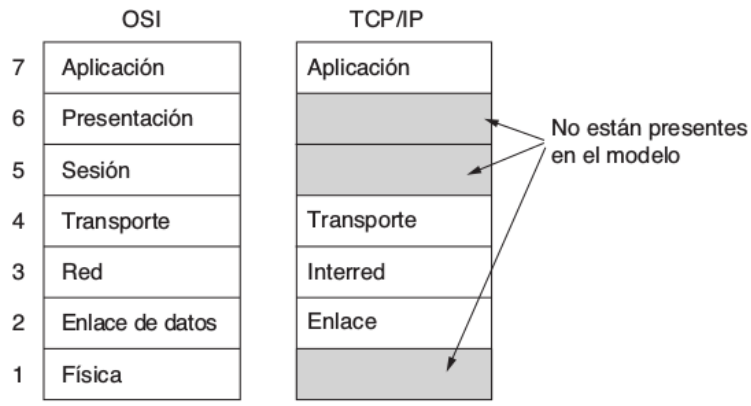

Figura 2.11: Modelo de Referencia OSI y TCP/IP

componene a los dos modelos [5].

Algunos autores consideran que las capas 5: Sesión y 7: Presentación del modelo OSI simplemente no existen, ya que su funcionaldad esta contenida en la capa 7: Aplicación, por lo que al igual que ellos utilizaremos el modelo hibrido de la Figura 2.12 donde se tiene cinco capas, empezando por la capa física, pasando por las capas de enlace, red y transporte hasta llegar a la capa de aplicación.

- La capa física especifica cómo transmitir bits a través de distintos tipos de medios como señales eléctricas (u otras señales analógicas).
- La capa de enlace trata sobre cómo enviar mensajes de longitud finita entre computadoras conectadas de manera directa con niveles específicos de confiabilidad. Ethernet y 802.11 son ejemplos de protocolos de capa de enlace.
- La capa de red se encarga de combinar varios enlaces múltiples en redes, de manera que podamos enviar paquetes entre computadoras distantes. Aquí se incluye la tarea de buscar la ruta por la cual enviarán los paquetes. IP es el principal protocolo de ejemplo para esta capa.
- La capa de transporte fortalece las garantías de entrega de la capa de Red, por lo general con una mayor confiabilidad, además provee abstracciones en la entrega, como un flujo

Figura 2.12: Modelo de referencia que usaremos en el presente trabajo.

de bytes confiable, que coincida con las necesidades de las distintas aplicaciones. TCP es un importante ejemplo de un protocolo de capa de transporte.

- La capa de aplicación contiene programas que hacen uso de la red. Por ejemplo el navegador web usa el protocolo HTTP. También hay programas de soporte importantes en la capa de aplicación, como el DNS, que muchas aplicaciones utilizan.

Para IoT no existe un modelo de referencia estandar ni adoptado por defacto aun, aunque uno de los mas referenciados en la literatura es el de la Figura 2.13 [18].

2.5. UDP

El Protocolo de Datagramas de Usuario (UDP: del ingles Use Datagram Protocol) es un protocolo de capa de transporte del modelo OSI, es un protocolo no orientado a conexión descrito en el RFC 768, lo único que hace es enviar paquetes entre aplicaciones. UDP transmite segmentos con un encabezado de 8 bytes seguidos de la carga útil como se muestra en la Figura 2.14, los campos puerto de origen y puerto de destino sirven para entregar el paquete a una aplicación en especifico, la longitud máxima de un paquete UDP es de 65 515 bytes, el campo Suma de verificación es opcional y proporciona una confiabilidad adicional al realizar una suma de verificación para el encabezado, los datos y un pseudoencabezado IP conceptual[5].

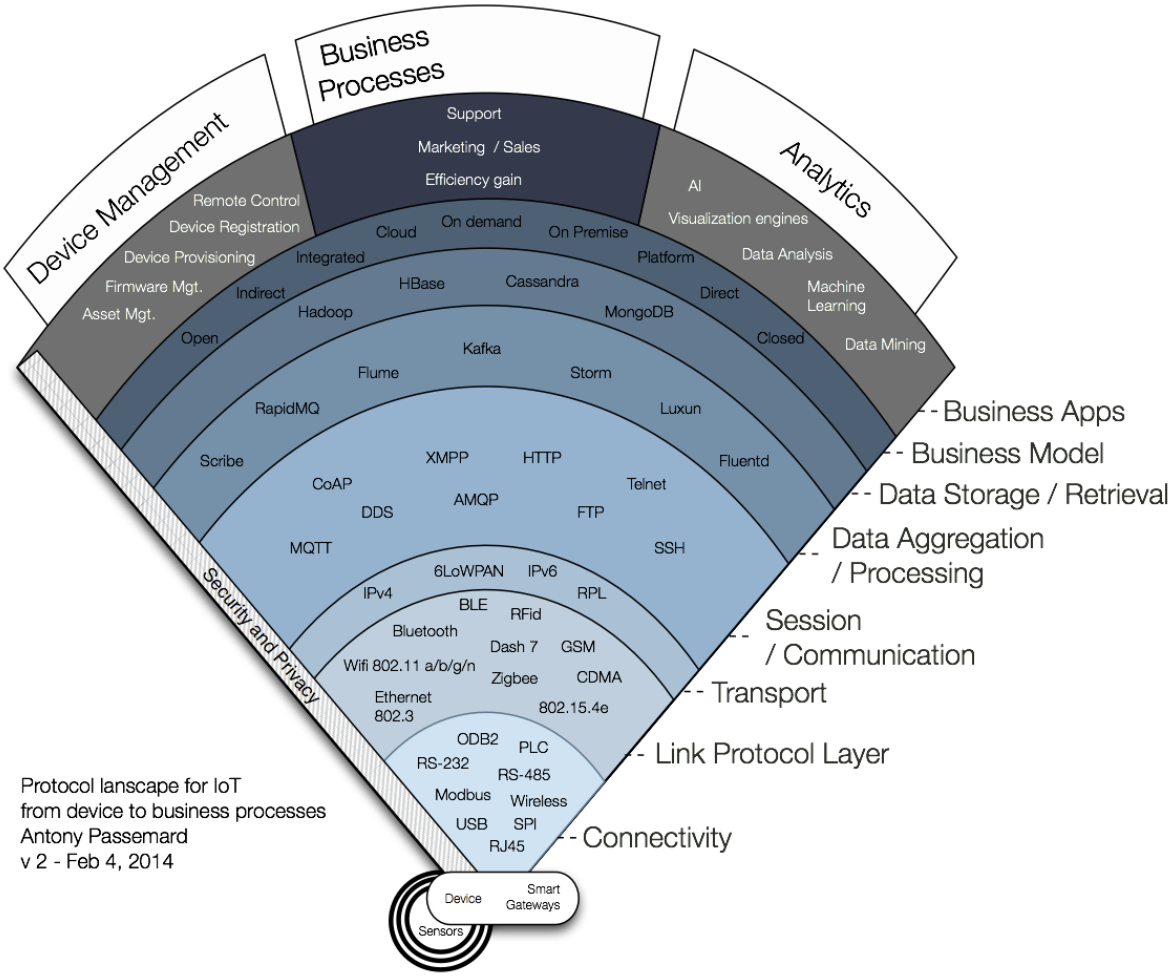

Figura 2.13: Model de Capas y Protocolos para IoT.

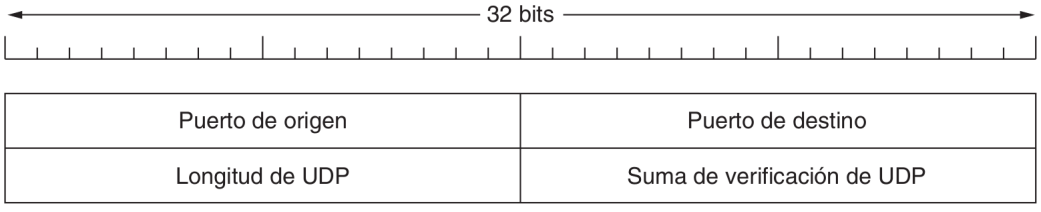

Figura 2.14: Encabezado UDP.

UDP es una interfaz entre la capa de Red a la Capa de Aplicación ya que como se menciono su función es entregar un segmento de datos, por tal razón tiene un encabezado tan corto, y no requiere un mecanismo tan complejo como TCP para entablar la comunicacion Cliente-Servidor, otras cosas que UDP no realiza son las siguientes:

- Control de flujo.
- Control de congestión.
- Retransmisión cuando se recibe un segmento erróneo.

Estos mecanismos son necesarios para tener una conexión confiable, pero UDP deja la tarea de implementarlas a la aplicación.

2.6. TCP

El Protocolo de Control de Transmisión (TCP: del ingles Transmission Control Protocol) fue diseñado para proporcionar un flujo confiable de bytes, es un protocolo de la capa de Transporte y se definio en el RFC 793 en Septiembre de 1981, durante el paso de los años se han hecho mejoras las cuales se han publicado en distintos RFC, el RFC 4614 es una guia de todos estos documentos. Ya que la capa de Red con IP no existen garantias de que los datagramas se entreguen de forma apropiada, corresponde a TCP manejar los mecanismos de:

- Control de flujo.
- Control de congestión.
- Retransmisión cuando se recibe un segmento erróneo.

En el servicio TCP tanto el cliente como el Servidor deben crear puntos terminales de conexión los cuales se conocen como sockets, cada socket tiene una direccion IP y un numero de puerto de 16 bits, los primeros 1024 puertos estan reservados para el sistema, y del 1024 al 49151 pueden ser usados para las aplicaciones. Todas las conexines en TCP pueden ir en ambas direcciones al mismo tiempo(full duplex), y debe tener exactamente dos puntos terminales, es decir no acepta multidifusión ni difusión. Una conexión TCP es un flujo de bytes, no un flujo de mensajes y cuando la aplicación pasa datos a TCP, éste decide entre entre enviarlos de inmediato o almacenarlos en el buffer, para tener mayor numero de datos que enviar, existe una señalización de datos urgentes que envia los datos directamente sin almacenarlos en buffer, sin embargo no se recomienda su utilización.

Las entidades TCP emisora y receptora intercambian datos en forma de segmentos, que consisten en en un encabezado fijo de 20 bytes, una parte opcional, seguidos de una carga util, el encabezado mas la carga util puede ser de un maximo de 65 520 bytes, aunque cada enlace tiene un Unidad Maxima de Transferencia (MTU: del ingles Maximum Transfer Unit), por lo que cada segmento debe caber en el MTU del emisor y receptor que normalmente es de 1500 bytes (el tamaño maximo de carga util de Ethernet). En implementaciones modernas existe un descubrimiento de MTU de la ruta, para ajustarlo al valos mas pequeño sin y no tener que fragmentar los segmentos.

En TCP cada dato enviado tiene un número de secuencia de 32 bits, cuando el emisor transmite un segmento se inicia un temporizador , y cuando llega al destino la entidad TCP receptora devuelve un segmento que contiene el numero de confirmación, si el temporizador del emisor expira antes de recibir la confirmación este reenvia de nuevo el segmento.

El encabezado de TCP como se muestra en la Figura 2.15 es de 20 bytes fijos para cada segmento transmitido. Los puntos terminales de origen y destino en conjunto identifican a la conexión, este identificador se denomina 5-tupla el cual consiste en el protocolo (TCP), IP de origen y puerto de origen, IP de destino y puerto de destino.

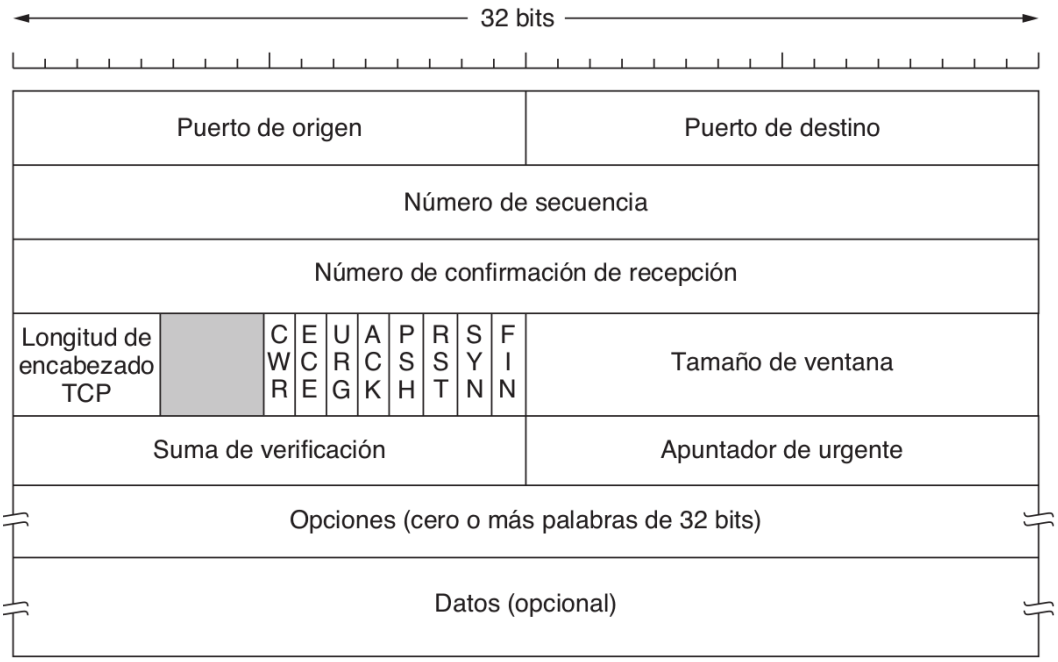

Figura 2.15: Encabezado TCP.

Puesto que TCP es un protocolo bastante complejo solo hacemos una breve descripción de el en este capitulo, mientras que en el Capitulo 3 y 4 tocamos algunos temas que fueron de gran importancia para el diseño de nuestro protocolo.

2.7. Protocolos de Comunicación para IoT

Los siguientes son algunos de los protocolos mas populares usados para IoT, todos ellos son protocolos de capa de Aplicación y estan montados sobre TCP o UDP.

2.7.1. MQTT(Message Queuing Telemetry Transport)

Protocolo desarrollado para la conexión maquina-maquina (M2M; del ingles machine-to-machine), la primer versión del protocolo fue desarrollada por Andy Stanford-Clark y Arlen Nipper en 1999, en 2013, IBM presenta MQTT v3.1. MQTT es un protocolo muy ligero, descrito en el estándar ISO (ISO/IEC PRF 20922), su funcionalidad se basa en mensajes

publish-subscribe, implementado sobre el protocolo TCP/IP, y requiere de un dispositivo central encargado de distribuir los mensajes al que se le llama broker, este protocolo requiere muy poco ancho de banda por lo que es ideal para el uso en comunicación de sensores, además de ser de codigo libre[19]. Algunas soluciones que ya implementan este protocolo son:

- Mosquitto: Broker MQTT v3.1
- IBM MessageSight

2.7.2. MQTT For Sensor Networks (MQTT-SN)

Este protocolo al igual que el anterior está optimizado para ser usado en conexiones de red con un ancho de banda muy bajo e intermitente, por tal razón es un protocolo ligero cuyo funcionamiento se basa en publish/subscribe para la comunicacion machine-to-machine y en aplicaciones móviles, sin embargo el protocolo MQTT requiere del protocolo de la capa de red TCP/IP el cual es complejo y requiere de muchos recursos de hardware de los cuales se carecen en ciertos sistemas pequeños y de bajo costo. Originalmente MQTT-SN fue desarrollado para correr en la parte superior de ZigBee definido en el estándar IEEE 802.15.4, para la comunicación entre la capa física y el control de acceso al medio, MQTT-SN fue diseñado para ser implementado en sistemas de bajo costo, con limitado uso de procesamiento y bajo consumo de batería, otras diferencias entre MQTT y MQTT-SN son las siguientes[19]: El mensaje CONNECT se divide en tres mensajes. El mensaje PUBLISH es sustituido por un mensaje corto de dos bytes, donde se envía un ID del dispositivo que desea publicar. No es necesario registrarse ya que de antemano se conocen los IDs de los clientes y el gateway/server por sus IDs cortos de dos bytes.

2.7.3. CoAP (Constrained Application Protocol)

Descrito en el RFC 7252, diseñado para la comunicación M2M entre nodos en una red (WNS), fácil integración a la web mediante HTTP, provee un modelo de integración request/response entre los dispositivos, soporta multicast. El objetivo de dicho protocolo es mantener bajo uso de recursos mediante la fragmentación de los paquetes. Su funcionamiento es muy similar al modelo Cliente/ Servidor en HTTP, sin embargo en en la interacción M2M los dos dispositivos cumplen los dos roles en cierto momento, a diferencia de HTTP este protocolo realiza las solicitudes de forma asíncrona mediante datagramas UDP[20].

2.7.4. Thread

Es un protocolo desarrollado para la industria emergente del Internet de las Cosas, por Thread Group en Julio del 2014, un grupo de trabajo conformado por siete compañias entre ellas Samsung, ARM, Qualcomm, NXP, OSRAM, entre otras, es una de las primeras iniciativas por hacer uso de un protocolo comun para el desarrollo de dispositivos domesticos e industriales para el IoT enfocandose en el bajo consumo de energia, implementando protocolos como 6LoWPAN[21] y el protocolo de comunicación inalámbrica IEEE 802.15.4[22] usado en dispositivos ZigBee. Thread soporta mas de 250 dispositivos conectados en una red de malla mediante el protocolo IPv6[23], de esta forma todos los dispositivos tienen acceso a Internet. Usando estándares ya probados Thread ofrece las siguientes ventajas:

- Redes fiables: Ofrece conexiones robustas de malla, auto reparables en caso de fallos.
- Redes seguras: cuenta con encriptación AES.
- Conectividad simple: Fácil instalación en teléfonos, tabletas, computadoras personales, que permiten la configuración de la red Thread.
- Bajo consumo de energía: Soporta bajo consumo de energía para dispositivos que usan

baterías, ya que implementa el estándar IEEE 802.15.4.

El protocolo Thread soporta conectividad con dispositivos pequeños de bajos recursos y conectividad hacia Internet, soporta múltiples capas de aplicación.

Otros protocolos de comunicación son los siguientes:

- SMCP
- STOMP
- XMPP
- XMPP-IoT
- Mihini/M3DA
- AMQP (Advanced Message Queuing Protocol)
- DDS
- LLAP
- LWM2M (Lightweight M2M)
- SSI (Simple Sensor Interface)
- Reactive Streams
- REST (Representational state transfer) - RESTful HTTP
- HTTP/2
- SOAP (Simple Object Access Protocol), JSON/XML, WebHooks, Jelastic, MongoDB

- Websocket
- JavaScript / Node.js IoT projects

Capítulo 3

μIoTP: Protocolo de Comunicación propuesto

3.1. Diseño del Protocolo

La motivación del presente trabajo fue diseñar un protocolo de comunicación que permita integrar dispositivos de bajo poder computaciónal a una plataforma de IoT, ya sea de forma directa si es que se cuentan con los recursos de hardware y energia necesarios o a traves de un Gateway(GW), las ventajas de entablar la comunicacion mediante un GW son principalmente que los nodos realizan menor cantidad de computo lo que les permite ahorrar energia, la otra ventaja es que en el GW se puede realizar procesamiento y analisis de los segmentos recibidos evitando el envio constante de datos a internet, teniendo asi una arquitectura tipo edge que consiste en traer capacidades de computo y almacenamiento a un dispositivo local, logrando una menor latencia en la comunicación y un menor consumo de ancho de banda.

El protocolo propuesto Micro Protocolo para Internet de las Cosas(μIoTP por sus siglas en ingles Micro Internet of Things Protocol) es un protocolo de comunicación entre Nodos Sensores/ Actuadores con un Gateway, μIoTP esta implementado sobre la capa de transporte

basado en el protocolo UDP descrito en el capitulo anterior, por esta razon podemos decir que es un protocolo de la capa de Aplicación como se muestra en la Figura **??**, por otro lado al ser un protocolo generico e independiente de la aplicación que provee servicios de transporte al administrar las conexiones de los dispositivos tambien parace un protocolo de capa de transporte en analogia con protocolo RTP con la diferencia que μIoTP maneja unidifusión y multidifusión y la comuncación es uno a muchos. En la Figura **??** se muestra el anidamiento y sobrecarga de bytes que es agregado en cada capa de red para poder enviar y recibir los segmentos, como se ha mencionado UDP únicamente entrega conjuntos de bytes llamados datagramas, con un encabezado de 8 bytes es un protocolo ligero pero poco confiable, estos datagramas son tomados por μIoTP quien los entrega a la aplicación, para asegurar que los datos sean entregados correctamente a su destino se implemento el mecanismo de retransmision de paquetes cuya recepción no ha sido confirmada, valiendose de temporizadores de expiración de paquetes que se detalla en secciones mas adelante al igual que otros aspectos de diseño.

3.1.1. Contexto de Aplicación

El protocolo fue diseñado para funcionar en una arquitectura tipo estrella, donde existe un nodo coordinador al que llamamos Gateway, (tipicamente mayor a tipo C2 e igual a E9) que se encuentra en el borde de la comunicación de la WSN hacia internet, los nodos hijos son dispositivos con capacidades reducidas los cuales son tipo C2 y E2 o menores, se puede abarcar un area local que depende de la infraestructura de red con la que se cuenta con un maximo de 254 Nodos mas 1 GW, lo equivalente a un Hogar, una nave Industrial, un pequeño campo o Invernadero, etc.

3.1.2. Eficiencia Energetica

Poniendo principal enfasis en reducir el consumo de energia de estos dispositivos el protocolo μIoTP cuenta con un encabezado reducido

tiene un encabezado fijo de solo 2 bytes, dependiendo de la palabra de control se tiene un subencabezado variable de 0 a 75 bytes. El diseño del protocolo se centra en lograr una mayor eficiencia de los recursos de hardware y energia de los nodos, ya que estos son dispositivos con recursos limitados de hardware y energia, los datos enviados y recibidos transportan lecturas de sensores como los descritos en los capitulos anteriores e intrucciones de control para los Actuadores, para este fin no se requieren gran capacidad en la carga útil por lo que cada segmento se limita a un maximo de 255 bytes, sin capacidad de fragmentacion o concatenacion de paquetes.

3.1.3. QoS

3.1.4. Timers

3.1.5. Buffers

3.1.6. Gateway

REF: T-TUT-SMARTCITY-2016-2-PDF-E.pdf, p. 96 Common requirements and capabilities of a gateway for Internet of Things applications

REF: T-TUT-NGN-2014-PDF-E, WSNs with the cluster structure, pag. 15

3.1.7. Mas Inteligencia

3.1.8. Seguridad

3.2. Palabras de Control μIoTP

Las palabras de control soportadas por el protocolo se muestran en la Tabla 3.1

Nombre	**Valor**	**Dirección**	**Descripción**
Reservado	0	Reservado	Reservado
CONNECT	1	Cliente a GW	Descubrimiento del Gateway y Conexión
CONFIG	2	Cliente a GW / GW a Cliente	Paquete de configuración
SEND	3	Cliente a GW / GW a Cliente	Envio de datos
RECEIVE	4	GW a Cliente	Esperando información
PING	5	Cliente a GW / GW a Cliente	Comprobación de dispositivos
DISCONNECT	6	Cliente a GW / GW a Cliente	Aviso de desconexión
Reservado	7	Reservado	Reservado
Reservado	8	Reservado	Reservado
ACKCONNECT	9	GW a Cliente	Confirmacion de Conexión
ACKCONFIG	10	GW a Cliente	Confirmacion de Configuración
ACKSEND	11	GW a Cliente / Cliente a GW	Confirmacion de Mensaje
Reservado	12	Reservado	Reservado

ACKPING	13	GW a Cliente / Cliente a GW	Confirmación de Mensaje PING
ACKDISCONNECT	14	GW a Cliente / Cliente a GW	Confirmacion de desconexión

Tabla 3.1: Palabras de control contenidas en cada segmento.

3.2.1. Encabezado

El protocolo cuenta con un encabezado fijo de 2 octetos, dependiendo de la intrucción se tiene un encabezado variable de(0 a X bytes) teniendo un segmento de 256 octetos incluyendo la parte fija del protocolo, la parte variable mas la carga util, la distribución de los 16 bits del encabezado fijo se muestran en la Tabla 3.2.

Tabla 3.2: Encabezado Fijo del protocolo

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
byte 1	PALABRA DE CONTROL				BANDERAS			
byte 2	ID DE SEGMENTO							

- **PALABRA DE CONTROL:** Palabras de control aceptadas por el protocolo descritas en la Tabla 3.2.
- **BANDERAS:** La función de cada una de ellas depende de la palabra de control.
  - **F**
  - **K**
  - **R**
- **ID DE SEGMENTO:** Número de secuencia de cada segmento enviado, es un dato de 8 bits.

3.2.2. CONNECT

La palabra de control CONNECT es enviada por el cliente en modo multidifusión (broadcast) con el fin de encontrar un dispositivo gateway, el segmento esta compuesto por los campos mostrados en la Tabla 3.3, y a continuacion se describe cada uno de ellos:

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
byte 1	CONNECT				F	K	R	
byte 2	ID DE SEGMENTO							
byte 3 - 7	NOMBRE DEL PROTOCOLO							
byte 8	VERSION DEL PROTOCOLO							
byte 9	CLASE DE DISPOSITIVO							
byte 10...	NOMBRE DEL DISPOSITIVO (maximo 64 bytes)							

Tabla 3.3: Segmendo de conexión CONNECT

Encabezado fijo CONNECT:

- **CONNECT:** Palabra de control de conexión.
- **BANDERAS:**
  - **F:** (1) Indica que el receptor debe confirmar la recepción del mensaje con un mensaje ACK.
  - **K:** (1) Indica que el Segmento va cifrado.
  - **R:** (1) Indica que el Segmento es una retransmisión.
- **ID DE SEGMENTO:** El primer mensaje es un número aleatorio de 8 bits.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
byte 1	CONNECT				F	K	R	

	0	0	0	1	1	1	0	0
byte 2	ID DE SEGMENTO							
	Número de secuencia inicial aleatorio							

Tabla 3.4: Encabezado fijo CONNECT.

Encabezado variable CONNECT:

- **NOMBRE DEL PROTOCOLO:** Contiene el nombre del protocolo de 5 bytes en codigo ASCII el cual sirve para ser identificado por dispositivos como firewalls.
- **VERSION DEL PROTOCOLO:** Version actual del protocolo de 1 byte.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
byte 3 - 7	NOMBRE DEL PROTOCOLO							
byte 3	"μ" (0x4B)							
byte 4	"I" (0x49)							
byte 5	"o" (0x4B)							
byte 6	"T" (0x6F)							
byte 7	"P" (0x50)							
byte 8	VERSION DEL PROTOCOLO							
	0x01							

Tabla 3.5: Encabezado variable CONNECT.

Carga útil CONNECT:

- **ALIMENTACIÓN:** Describe la fuente de alimentacion del nodo de acuerdo a la Tabla 2.2.

- **CLASE:** Describe la capacidad del nodo de acuerdo a la Tabla 2.1.
- **NOMBRE DEL DISPOSITIVO:** Nombre del nodo, maximo 64 bytes.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
byte 9	CLASE DE DISPOSITIVO							
	ALIMENTACIÓN: E0, E1, E2, E9				CLASE: C0, C1, C2			
byte 10...	NOMBRE DEl DISPOSITIVO (maximo 64 bytes)							
	ej. “Sensor n01”							

Tabla 3.6: Carga util CONNECT.

3.2.3. CONFIG

La palabra de control CONFIG puede ser enviada por el cliente o por el GW. En el segmento config el cliente le indica al GW los perfiles que estará manejando para enviar los datos, el encabezado se muestra en la Tabla 3.7:

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
byte 1	CONFIG				F	K	R	
byte 2	ID DE SEGMENTO							
byte 3	INFO PERFILES							
byte 4	CARGA ÚTIL OPCIONAL(de 0 a 45 bytes)							

Tabla 3.7: Segmendo de conexión

Encabezado fijo CONFIG:

- **CONFIG:** Palabra de control de configuración.
- **BANDERAS:** Mismo caso que en CONNECT.

- **ID DE SEGMENTO:** Numero de secuencia de segmento.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
byte 1	CONFIG				F	K	R	
	0	0	1	0	1	1	0	0
byte 2	ID DE SEGMENTO							
	Número de secuencia							

Tabla 3.8: Encabezado fijo CONFIG.

Encabezado variable CONFIG:

- **PERFIL:** Existen 3 tipos de perfiles, cada uno estructura los datos de forma diferente, los tipos de perfiles se muestran en la Tabla 3.10.
- **QoS:** (1) Se debe confirmar la recepción de los segmentos.
- **T:** Tipo de dispositivo (1: Actuador, 0: Sensor).
- **NUM PERFILES:** Numero de perfiles que seran registrados, maximo 15.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
byte 3	INFO PERFILES							
	PERFIL		QoS	T	NUM PERFILES (N)			

Tabla 3.9: Encabezado variable CONFIG.

PERFIL	**DESCRIPCIÓN**
00	Reservado
01	Caracteres de 8 bits
10	Personalizado

11	Definido en el Gateway

Tabla 3.10: Tipos de PERFIL para CONFIG.

Carga útil CONFIG: La carga útill del segmento de configuración depende del campo PERFIL para su codificación.

- **Caracteres de 8 bits:** Este tipo de perfil indica que los datos serán enviados en formato de cadena, dado que no es necesaria mayor explicación no hay carga útil.
- **Personalizado:** Este tipo de perfil es el más complejo, debido a que se debe describir el formato y orden de cada dato que puede ser enviado en el segmento como se describe en la Tabla 3.11, donde:
  - **ID PERFIL:** Corresponde al numero consecutivo identificador que se le ha asignando al perfil, comenzando por el número 1 y maximo el número 15.
  - **LONG. DE PERFIL:** Numero de elementos que componen el perfil + 1, es decir si la longitud de perfil tiene un valor 2, se tendran 2 + 1 elementos, con el fin de poder tener 16 elementos con los 4 bits disponibles para el campo LONG. DE PERFIL.
  - $dt_{i,j}$**:** Tipo de dato del elemento i, j en la lista de perfiles, donde i corresponde al ID PERFIL actual y j corresponde al elemento del perfil, los tipos de dato se muestran en la Tabla 3.12.
- **Definido en el GW:** Puesto que los perfiles ya estan descritos en el GW, en la carga útil de CONFIG solo es necesario listar los perfiles que seran usados por el dispositivo como se muestra en la Tabla 3.13.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

	CARGA ÚTIL			
byte 4	ID PERFIL(1)		LONG. DE PERFIL ($n1$)	
byte 5	$dt_{1,1}$	$dt_{1,2}$	...	$dt_{1,n1}$
byte ...	ID PERFIL(i)		LONG. DE PERFIL (ni)	
byte ...	$dt_{i,1}$	$dt_{i,2}$	...	$dt_{i,ni}$
byte ...	ID PERFIL(N)		LONG. DE PERFIL (nN)	
byte ...	$dt_{N,1}$	$dt_{N,2}$	...	$dt_{N,nN}$

Tabla 3.11: Carga útil CONFIG para PERFIL = 10b y N: NUM PERFILES.

TIPO DE DATO (dt)	**Significado**
00b	String con fin de cadena '\x0' máximo 253 caracteres de 8 bits
01b	Caracter / Booleano de 8 bits (char)
10b	Entero de 16 bits (int)
11b	Número flotante de 32 bits (float)

Tabla 3.12: Tipos de dato del perfil en CONFIG.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	CARGA ÚTIL							
byte 4	ID PERFIL(1)							
byte ...	ID PERFIL(i)							
byte ...	...							
byte ...	ID PERFIL(N)							

Tabla 3.13: Carga útil CONFIG para PERFIL = 11b y N: NUM PERFILES

3.2.4. ACK

3.2.5. SEND

La palabra de control SEND puede ser enviada en cualquiera de las dos direcciones, depende del formato como fue establecido en cada uno de los perfiles, como se describe en la Tabla 3.14.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
byte 1	SEND				F	K	R	
byte 2	ID DE SEGMENTO							
byte 3	CARGA ÚTIL(de 0 a 253 bytes)							

Tabla 3.14: Segmendo de envio de mensaje SEND.

Encabezado fijo SEND:

- **SEND:** Palabra de control de envio de segmento.
- **BANDERAS:** Mismo caso que en CONNECT.
- **ID DE SEGMENTO:** Numero de secuencia de segmento.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
byte 1	SEND				F	K	R	
	0	1	0	0	1	1	0	0
byte 2	ID DE SEGMENTO							
	Número de secuencia							

Tabla 3.15: Encabezado fijo SEND.

Carga útil SEND: La carga útill del segmento SEND depende del perfil que se esta enviando.

- **Caracteres de 8 bits:** Como se muestra en la Tabla 3.16.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	CARGA ÚTIL(de 0 a 253 bytes)							
byte 3...	CADENA							

Tabla 3.16: Segmendo de envio de mensaje SEND con PERFIL = 01b.

- **Personalizado:** Como se muestra en la Tabla 3.17 se envia el ID del perfil seguido de los bytes correspondientes de acuerdo a la codificación establecida en CONFIG para cada perfil.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	CARGA ÚTIL(de 0 a 253 bytes)							
byte 3	ID PERFIL(i)							
	CODIFICACIÓN DE DATOS							
byte 4...	$DATO_{i,1}$							
byte ...	$DATO_{i,j}$							
byte ...	$DATO_{i,ni}$							

Tabla 3.17: Segmendo de envio de mensaje SEND con PERFIL = 10b.

- **Definido en el GW:** La codificación de los bytes es parecida con la del perfil Personalizado, con la única diferencia de que los perfiles ya existen en el GW.

3.2.6. RECEIVE

La palabra de control RECEIVE es enviada del GW al cliente, como unicast o multicast con el objeto de solicitar los datos de lectura del nodo de acuerdo a cierto perfil, como se muestra en la Tabla 3.18, el nodo debe contestar con un segmento SEND.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
byte 1	RECEIVE				F	K	R	
byte 2	ID DE SEGMENTO							
byte 3	CARGA ÚTIL							

Tabla 3.18: Segmendo de envio de mensaje SEND.

Encabezado fijo RECEIVE:

- **RECEIVE:** Palabra de control de solicitud de segmento.
- **BANDERAS:**
  - **F:** (1) Mensaje Unicast, (0) Mensaje Multicast.
  - **K:** (1) Indica que el Segmento va cifrado.
  - **R:** (1) Indica que el Segmento es una retransmisión.
- **ID DE SEGMENTO:** Numero de secuencia de segmento.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
byte 1	RECEIVE				F	K	R	
	0	1	0	1	1	1	0	0
byte 2	ID DE SEGMENTO							
	Número de secuencia							

Tabla 3.19: Encabezado fijo RECEIVE.

Carga útil RECEIVE: La carga útill del segmento RECEIVE depende del perfil que fue definido por CONFIG, para PERFIL = 01b no se tiene carga útil, ver Tabla 3.20.

- **ID PERFIL:** ID del perfil del cual se solicitan los datos.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	CARGA ÚTIL(de 0 a 253 bytes)							
byte 3	ID PERFIL(i)							

Tabla 3.20: Segmendo RECEIVE para PERFIL = 10b ó PERFIL = 11b.

3.2.7. PING

La palabra de control PING permite verificar el estado activo tanto del cliente como del GW, proporcionando información del tiempo de ida y vuelta del mensaje, el segmento se muestra en la Tabla 3.21.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
byte 1	PING				F	K	R	
byte 2	ID DE SEGMENTO							
byte 3	CARGA ÚTIL							

Tabla 3.21: Segmendo PING.

Encabezado fijo PING:

- **PING:** Palabra de control de solicitud de estado.
- **BANDERAS:** Mismo caso que RECEIVE.
- **ID DE SEGMENTO:** Numero de secuencia de segmento.

Bit	7	6	5	4	3	2	1	0
byte 1	RECEIVE				F	K	R	
	0	1	1	0	1	1	0	0
byte 2	ID DE SEGMENTO							
	Número de secuencia							

Tabla 3.22: Encabezado fijo PING.

Carga útil PING: La carga útill del segmento PING contiene dos bytes que representan la estampa de tiempo del dispositivo que envia el segmento.

- **ESTAMPA DE TIEMPO:** Estampa de tiempo del dispositivo que envia el segmento.

Bit	7	6	5	4	3	2	1	0
	CARGA ÚTIL(opcional)							
byte 3	ESTAMPA DE TIEMPO (L)							
byte 4	ESTAMPA DE TIEMPO (H)							

Tabla 3.23: Carga útil de Segmento PING.

3.2.8. DISCONNECT

La palabra de control DESCONNECT tiene dos funcionalidades principales en la Tabla 3.24 tenemos el formato del encabezado.

Bit	7	6	5	4	3	2	1	0
byte 1	DISCONNECT				F	K	R	
byte 2	ID DE SEGMENTO							
byte 3	CARGA ÚTIL (0 A 2 Bytes)							

Tabla 3.24: Segmendo DISCONNECT.

Encabezado fijo DISCONNECT:

- **DISCONNECT:** Palabra de control para solicitud de desconección.
- **BANDERAS:**
  - **F:** (1) Sleep, (0) Desconección permanente.
  - **K:** (1) Indica que el Segmento va cifrado.
  - **R:** (1) Indica que el Segmento es una retransmisión.
- **ID DE SEGMENTO:** Numero de secuencia de segmento.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
byte 1	DISCONNECT				F	K	R	
	0	1	1	1	1	1	0	0
byte 2	ID DE SEGMENTO							
	Número de secuencia							

Tabla 3.25: Encabezado fijo DISCONNECT.

Carga útil DISCONNECT: Cuando F = 1 el segmento contiene carga útil de 2 Bytes, que representa el tiempo que el dispositivo entrara en suspención, para F = 0, es una desconexion permanente por lo que no hay carga útil.

- **TIEMPO DE SUSPENCIÓN:** Tiempo que el dispositivo entrara en suspención en milisegundos (ms).

Bit	7	6	5	4	3	2	1	0
	CARGA ÚTIL(opcional)							
byte 3	TIEMPO DE SUSPENCIÓN (L)							
byte 4	TIEMPO DE SUSPENCIÓN (H)							

Tabla 3.26: Carga útil de Segmento PING.

3.3. Modelado de administración de conexiones

Capítulo 4

Experimentación y Resultados

4.1. Escenarios de pruebas

4.2. Hardware

4.3. Experimentos

4.4. Resultados

4.5. Comparativa

Capítulo 5

Conclusiones y trabajo futuro

5.1. Trabajos Futuros

5.2. Conclusiones

Bibliografía

[1] K. Ashton, "That 'Internet of Things' Things," 2009. http://www.rfidjournal.com/articles/view?4986.

[2] F. Zhao and L. Guibas, *Wireless Sensor Networks: An Information Processing Approach*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2004.

[3] "Information technology-Open Systems Interconection-Basic Reference Model," ISO/IEC 7498, ISO, November 1994. https://www.ecma-international.org/activities/Communications/TG11/s020269e.pdf.

[4] "User Datagram Protocol," RFC 768, ISI, August 1980. https://www.ietf.org/rfc/rfc768.txt.

[5] A. S. Tanenbaum and D. J. Wetherall, *Computer Networks*. Upper Saddle River, NJ, USA: Prentice Hall Press, 5th ed., 2010.

[6] S. Dwars and T. Phinney, "Industrial Routing Requirements in Low-Power and Lossy Networks," 2009.

[7] Cisco, "Securing the Internet of Things: A Proposed Framework," 2017. https://www.cisco.com/c/en/us/about/security-center/secure-iot-proposed-framework.html#10.

[8] P. F. Ovidiu Vermesan, *Digitising the Industry - Internet of Things Connecting the Physical, Digital and Virtual Worlds*. Alsbjergvej 10 9260 Gistrup Denmark: River Publishers, 2016.

[9] ITU Internet Reports, The Internet of Things, November 2005.

[10] O. Vermesan, P. Friess, P. Guillemin, S. Gusmeroli, et al., "Internet of Things Strategic Research Agenda", Chapter 2 in Internet of Things – Global Technological and Societal Trends, River Publishers, 2011, ISBN 978-87-92329-67-7.

[11] ITU Internet Reports, Unleashing the potenti al of the Internet of Things, 2016.

[12] C. Bormann, M. Ersue, and A. Keränen, "Terminology for Constrained-Node Networks." RFC 7228, May 2014.

[13] J. Hill, R. Szewcyk, A. Woo, D. Culler, S. Hollar, and K. Pister. System architecture directions for networked sensors. In *Proc. 8th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS IV)*, Cambridge, MA, pages 93-104. 2000.

[14] ITU-T, "Applications of Wireless Sensor Networks in Next Generation Networks," *Series T.2000: Next Generation Networks*, no. February, pp. 1–94, 2014.

[15] CHONG, C.-Y. and KUMAR, S. P. *Sensor networks: Evolution, opportunities, and challenges*. Proceedings of the IEEE 91(8), 2003, pp. 1247-1256.

[16] HATLER, M., GURGANIOUS, D. and CHI, C. *Industrial wireless sensor networks. A market dynamics report*. ON World, 2012.

[17] "Internet of Things: Wireless Sensor Networks," *Internation Electronic Commision*, no. December, pp. 1–78, 2014.

[18] A. Passemard, *The Internet of Things Protocol stack – from sensors to business value*, online at http://entrepreneurshiptalk.wordpress.com/2014/01/29/the-internet-of-thing-protocol-stack-from-sensors-to-business-value/.

[19] Ibm, "MQTT For Sensor Networks (MQTT-SN) Protocol Specification," p. 28, 2013. `http://mqtt.org/new/wp-content/uploads/2009/06/MQTT-SN{_}spec{_}v1.2.pdf`.

[20] "The Constrained Application Protocol (CoAP)," RFC 7252, Internet Engineering Task Force (IETF), June 2014. `https://www.ietf.org/rfc/rfc768.txt`.

[21] "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks," RFC 6282, Internet Engineering Task Force (IETF), September 2011. `https://tools.ietf.org/pdf/rfc6282.pdf`.

[22] IEEE Computer Society, *802.15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)*, vol. 2011. 2011.

[23] "Internet Protocol, Version 6 (IPv6)," RFC 2460, Network Working Group, December 1998. `https://tools.ietf.org/pdf/rfc2460.pdf`.