



FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

IMPLEMENTACIÓN DE UN PROTOTIPO IOT CON
SEMÁFOROS INTELIGENTES A TRAVÉS DE UNA
INFRAESTRUCTURA DE COMPUTACIÓN EN LA NIEBLA

AUTOR

SEBASTIAN VALLE PAZMIÑO

AÑO

2019



FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

IMPLEMENTACIÓN DE UN PROTOTIPO IOT CON SEMÁFOROS
INTELIGENTES A TRAVÉS DE UNA INFRAESTRUCTURA DE
COMPUTACIÓN EN LA NIEBLA.

Trabajo de Titulación presentado en conformidad a los requisitos establecidos
para optar por el título de Ingeniero en Electrónica y Redes de Información.

Profesor Guía

MSc. Luis Santiago Criollo Caizaguano

Autor

Sebastián Valle Pazmiño

Año

2019

DECLARACIÓN DEL PROFESOR GUÍA

“Declaro haber dirigido este trabajo, Implementación de un prototipo IoT con semáforos inteligentes a través de una infraestructura de una infraestructura de computación en la niebla, a través de reuniones periódicas con el estudiante, Sebastián Valle Pazmiño, en el semestre 201910, orientando sus conocimientos y competencias para un eficiente desarrollo del tema escogido y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación”

Luis Santiago Criollo Caizaguano

Magister en Redes de Comunicaciones

C.I.: 1717112955

DECLARACIÓN DEL PROFESOR CORRECTOR

“Declaro haber revisado este trabajo, Implementación de un prototipo IoT con semáforos inteligentes a través de una infraestructura de computación en la niebla, del estudiante, Sebastián Valle Pazmiño, en el semestre 201910, dando cumplimiento a todas las disposiciones vigentes que regulan los trabajos de Trabajos de Titulación”

Iván Patricio Ortiz Garcés

Magister en Redes de Comunicaciones

C.I.: 0602356776

DECLARACIÓN DE AUTORIA DEL ESTUDIANTE

“Declaro que este trabajo es original, de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes”

Sebastian Valle Pazmiño

C.I.: 1715759369

AGRADECIMIENTOS

A Dios por ser el pilar fundamental en cada etapa de mi vida. A mi hija quien es y será siempre mi mayor motivación para poder superarme cada día, también a cada uno de mis profesores quienes aportaron en mi formación profesional con su guía y experiencia.

DEDICATORIA

A mi familia, pero sobre todo a mis papás que siempre están conmigo apoyándome y que han dado su mayor esfuerzo para poder alcanzar cada una de mis metas.

RESUMEN

El Internet de las cosas es un tipo de tecnología emergente que se está adentrando en varias áreas de la industria tales como transporte, ciudades, agricultura, casas, edificios inteligentes, redes eléctricas, entre muchas otras. Consecuentemente esto desencadenará una cantidad desenfrenada de objetos y dispositivos conectados a la red, que pueden saturar los enlaces de core hacia los servidores de la computación en la nube. Por otra parte las aplicaciones IoT que son sensibles al tiempo de respuesta, también se ven afectadas en la distancia a la que se envían los datos para ser procesadas con el fin de obtener una acción y resultado.

En el presente proyecto de titulación se propone realizar el prototipo de una aplicación IoT enfocada para los vehículos de emergencia a través de una infraestructura de computación en la niebla. Esta nueva tecnología permite reducir los tiempos de respuesta y enviar solamente los datos que son necesarios a la computación en la nube. El vehículo de emergencia contiene un dispositivo inalámbrico que envía mensajes de alerta periódicos, al cual se lo conoce como *beacon* y se encuentra a bordo del vehículo. Mediante los mensajes de alerta se podrá habilitar los semáforos en verde hacia el destino final.

El ambiente de simulación contiene nodos de la computación en la niebla que se encuentran interconectados lo más cercano al vehículo mediante el protocolo de red de área extendida de bajo consumo denominado LoRaWAN. De la misma forma los nodos de la computación en la niebla ejecutan una aplicación con interfaz gráfica para poder administrar los nodos y saber cuando se recibe un mensaje de alerta.

Finalmente se procederá a realizar una comparación de la computación en la niebla y la computación en la nube tomando en cuenta como parámetro el tiempo de respuesta en una aplicación u otra.

ABSTRACT

The Internet of Things is a type of emerging technology that is showing up in some areas of the industry such as transportation, cities, agriculture, houses, smart buildings, smart grids, among many others. Consequently, this will unleash a huge number of things and devices connected to the internet, which can saturate the core links to the data centers of the cloud computing. On the other hand, the IoT applications that are time sensitive also has the disadvantage to send the data to the data centers that are far away from the IoT devices.

In this work it is develop an IoT application prototype focused on emergency vehicles with a fog computing infrastructure. This new technology allows to reduce response times and send only the data that is necessary to be analyze in the cloud computing. The emergency vehicle contains a wireless device that sends warning messages. This device is known as a beacon and is on board the vehicle. Through the warning messages, the green traffic lights can be enabled towards its final destination.

The simulation environment contains the fog nodes that are interconnected in the edge of the network near to the vehicle through the low power wide area network protocol LoRaWAN. In the same way, the fog nodes execute an application with graphical interface to be able to manage the nodes and know when an alert message is received.

Finally, we will make a comparison about the response time with the fog computing and cloud computing technologies.

ÍNDICE

1	Introducción	1
1.1	Antecedentes	1
1.2	Justificación	2
1.3	Alcance	3
1.4	Objetivo General	3
1.5	Objetivos Específicos	3
2	Tecnologías Existentes.....	4
2.1	Introducción	4
2.2	Computación en la niebla	4
2.2.1	El rol de la computación en la niebla en el Internet de las Cosas .	6
2.2.2	Principales características	9
2.2.3	Pilares de la arquitectura referencial OpenFog.....	10
2.2.4	Arquitectura	16
2.3	La computación en la niebla y la computación en la nube	19
2.4	Tecnología Inalámbrica	24
2.4.1	Redes de área extendida de bajo consumo.....	24
2.4.2	LoRa/LoRaWAN.....	25
2.4.3	Sigfox.....	29
2.4.4	WiFi Halow.....	31
3	Sistemas de Transporte Inteligentes	33
3.1	Introducción	33
3.2	Redes Ad-Hoc.....	33
3.2.1	MANET	35
3.2.2	VANET	35

3.3	Tecnologías inalámbricas para STI	38
3.3.1	Comunicaciones dedicadas de corto alcance	38
3.3.2	Identificación por radiofrecuencia	40
3.3.3	Tecnologías 802.15.....	43
3.4	Urbes y semaforización con STI.....	47
3.4.1	Quito	48
3.4.2	Pittsburgh.....	51
3.4.3	Nueva York.....	57
3.5	Comparación de STI entre urbes.....	61
4	Metodología e implementación.....	63
4.1	Estudios previos.....	63
4.2	Metodología	63
4.3	Selección de tecnologías	64
4.3.1	Comunicación vehículo a infraestructura	64
4.3.2	Comunicación entre nodos de la computación en la niebla	65
4.4	Diagrama de red	67
4.5	Desarrollo de prototipo	71
4.5.1	Framework.....	72
4.5.2	Nodos de la computación en la niebla.....	75
4.5.3	Implementación de Vehículo a Infraestructura.....	78
4.5.4	Red malla de LoRa.....	82
4.5.5	Aplicativo C#.....	85
4.5.6	Instancia Windows en Amazon Web Services	87
4.5.7	Ambiente de simulación	91
4.6	Análisis y resultados.....	93
4.6.1	Comparación de tiempos de respuesta entre la computación en la nube y la computación en la niebla	94
5	Conclusiones y Recomendaciones.....	103

5.1 Conclusiones	103
5.2 Recomendaciones	104
REFERENCIAS	105
ANEXOS	114

1 Introducción

1.1 Antecedentes

En la actualidad las principales urbes lidian con resolver problemas como lo son el transporte, la movilidad, seguridad entre muchas otras y en el esfuerzo de solventarlas con el gran avance y desarrollo tecnológico, ha surgido un nuevo concepto denominado *Smart Cities* o ciudades inteligentes. Este nuevo concepto tiene como objetivo principal lograr la sustentabilidad en temas económicos, sociales y ambientales a través de las Tecnologías de Información y Comunicación (TIC). (Höller et al., 2014)

Se proyecta que para el año 2050 el 68% de la población mundial se concentrará en áreas urbanas, por lo que está latente la tendencia de establecer ciudades inteligentes y el de abordar soluciones en transporte y movilidad. (Naciones Unidas, 2018)

Para la disminución de los problemas del tráfico vehicular, el municipio del Distrito Metropolitano de Quito en su última alcaldía cuenta con un plan de visión estratégica y movilidad en donde no se evidencian aun soluciones que incluyan a las Tecnologías de Información (TI). Un ejemplo de esto es el Internet de las Cosas, cuyas siglas en inglés son (IoT), en donde esta tecnología es indispensable para cualquier urbe.

Ante la necesidad de regularizar con éxito el tránsito vehicular y todo lo que ello implica, existe un plan de sincronización y funcionamiento general del sistema de semáforos que permita optimizar la distribución del tiempo de estos en las intersecciones de la ciudad de Quito. Lo que contribuye ligeramente como solución al tráfico vehicular que crece aceleradamente; pero sigue siendo inefectivo en la circulación de vehículos de emergencia tales como ambulancias, vehículos de policías y bomberos que pierden tiempo vital al momento de atender una emergencia por el irrespeto de ciertos conductores a la norma.

1.2 Justificación

Según una publicación de diario el Comercio en el año 2015, las ambulancias al responder algún llamado de emergencia triplican su tiempo de traslado del paciente, debido a la congestión, irrespeto y falta de conciencia de parte de los conductores. (El Comercio, 2015)

Debido a que la Industria 4.0 se está instaurando en la actualidad con soluciones basadas en el Internet de las Cosas, 5G, Big Data, Software etc. Se tendrá miles de millones de dispositivos conectados al internet como lo son: sensores, actuadores, celulares, laptops y tablets para ser utilizados en distintas áreas como la movilidad, transporte, Smart grids, alumbrado inteligente, casas y edificios inteligente; por lo que Cisco estima que para el año 2020 existirán 50 mil millones de dispositivos lo que implicaría una posible saturación del ancho de banda en el transporte de datos desde el perímetro de las redes de área local (LAN), hacia los centros de datos. (Cisco, 2017)

El análisis de datos en una arquitectura IoT, se lo hace en su gran mayoría con soluciones puramente basadas a través de una infraestructura de computación en la nube, lo que tiene como consecuencia altos tiempos de respuesta y termina siendo no la mejor solución hacia aplicaciones que necesiten respuesta en tiempo real.

Debido a estas limitantes en futuras implementaciones en el Internet de las Cosas, es que surge con fuerza el término de computación en la niebla o como se lo conoce en inglés como fog computing y al encontrarse en el perímetro de la red; específicamente entre los dispositivos físicos y el backbone de la red de datos, esta nueva tecnología permitirá que no exista saturación en el transporte de los datos, mejoren los tiempos de respuesta e inclusive brindar una mayor seguridad, ya que la gran mayoría de información se manejará dentro de la red interna.

1.3 Alcance

Este trabajo de titulación consiste en un prototipo simulando una solución IoT a través de una red de sensores con los siguientes componentes:

- Sensores de proximidad o beacons instalados en los vehículos de emergencia e intersecciones.
- Red de sensores mesh entre semáforos y gateways.
- Implementación del framework DSA (Distributed Services Architecture) en los gateways lo que permite actuar a estos como dispositivos de la computación en la niebla.

Los sensores estarán comunicados entre sí con el protocolo LoRa, el cual está clasificado dentro de los protocolos LPWAN (Low Power Wide Area Network) que están orientados a tener gran alcance de cobertura y un bajo consumo de energía lo que permite tener varios nodos distribuidos a lo largo de una amplia zona geográfica, salvaguardando el consumo energético del dispositivo.

1.4 Objetivo General

Implementar un prototipo de semáforos inteligentes a través de una infraestructura de computación en la niebla para el paso de vehículos de emergencia.

1.5 Objetivos Específicos

- Investigar el funcionamiento de los sistemas de semaforización en una urbe.
- Realizar una comparación entre la computación en la nube y computación en la niebla tomando en cuenta latencia y tiempos de respuesta.
- Realizar la automatización de semáforos para habilitarlos según sea el camino o destino del vehículo de emergencia.

2 Tecnologías Existentes

2.1 Introducción

Actualmente, el IoT, permite que cualquier clase de objeto como equipos electrónicos, dispositivos médicos, electrodomésticos, vehículos y todo tipo de sensores o cosas pasen a formar parte del internet. Esto ha incidido en la interacción entre objetos y personas, también denominado comunicaciones máquina a persona (M2P – *Machine to Person*) e inclusive entre objetos también conocido por las siglas M2M (*Machine to Machine*).

Uno de los factores primordiales en el IoT es la simplificación de procesos e innovación en varias áreas de la industria como lo son: agricultura, ciudades inteligentes, transporte, salud, hogares digitales etc. Además de conducir a una transformación digital en todos los aspectos de nuestras vidas. Esto se debe a la creación de nuevos modelos de negocios mediante la conexión entre objetos o dispositivos cotidianos a través de servicios que están alojados en la nube.

En consecuencia, a toda esta transformación digital, las cosas u objetos están generando datos a una velocidad exponencial; por consiguiente, hay congestión y saturación de datos en el backbone de la red. Esto se debe a que modelos y arquitecturas actuales del IoT, están basadas solamente en la computación en la nube; por tanto, en la transmisión de grandes flujos de datos se llega a tener alta latencia. Por lo que no es factible para aplicaciones que requieran respuesta en tiempo real; como por ejemplo la interacción entre vehículo a vehículo, vehículo a plataforma, redes eléctricas inteligentes (Smart grids) y alarmas de prevención contra desastres.

2.2 Computación en la niebla

El término computación en la niebla lo propuso Cisco por primera vez en el año 2012. En un principio se catalogaba a esta tecnología emergente como una extensión de la computación de la nube ubicada en el borde de una red de área local. Definiendo como principal objetivo el de habilitar un servicio de procesamiento para los objetos con conexión a Internet. (Mouradian et al., 2018).

Este nuevo concepto tomó interés y se lo propuso también como un nuevo sistema computacional geográficamente distribuido para aplicaciones IoT. Por tanto, debido a su vertiginoso interés, es que dicha tecnología está siendo impulsada por el consorcio OpenFog, el cual está fundado por empresas tecnológicas de alto renombre e instituciones académicas como: ARM, Cisco, Dell, Intel, Microsoft y la Universidad de Princeton.

Además, está conformado por: AT&T, FOXCONN, GE, Hitachi, Zakura Internet, ZTE y la Universidad ShanghaiTech. Como se puede observar en la figura 1.



Figura 1. Miembros actuales del Consorcio OpenFog

Tomado de (The OpenFog Consortium, 2018).

Este consorcio es el encargado en la estandarización de la computación en la niebla. Lo que ha permitido desarrollar una arquitectura abierta para garantizar principalmente la interoperabilidad y escalabilidad en la implementación de esta tecnología emergente.

Debido al esfuerzo que este consorcio ha realizado para la estandarización de una arquitectura y sistema abierto. El IEEE (Instituto de Ingenieros Eléctricos y Electrónicos) que es una organización global responsable en la estandarización de distintas áreas técnicas; adoptó a la arquitectura referencial del consorcio OpenFog, como estándar oficial para la computación en la niebla. (IEEE Standards Association, 2018)

En el documento técnico de la computación en la niebla se la define como una arquitectura horizontal a nivel de sistema, que distribuye funciones de cómputo, control, almacenamiento y conectividad; a lo largo de la computación en la nube hacia los objetos. (OpenFog Consortium Architecture Working Group, 2017)

Para un mejor entendimiento del concepto; a continuación, se detalla de mejor manera las palabras claves que engloban esta terminología:

- **Arquitectura Horizontal:** Abarca múltiples industrias (médica, agricultura, transporte etc.) y mercados verticales, otorgando inteligencia y servicios a las aplicaciones de los usuarios y negocios.
- **A través de la nube hacia los objetos:** Distribuye y habilita tanto aplicaciones como servicios cerca de las cosas u objetos, inclusive en toda la infraestructura que cruza o hace puente entre los dispositivos finales y la computación en la nube.
- **Nivel de sistema:** Es un tipo de computación a nivel de sistema, debido a que se extiende desde los objetos físicos a través de la red hasta la computación en la nube. lo que permite intervenir en múltiples capas de protocolos que pueden empezar desde no basados en IP como Zigbee y Wifi; redes de área amplia de bajo consumo por ejemplo LoRaWAN y Sigfox; y por último protocolos basados en IP teniendo a los más conocidos IPv4 e IPv6.

2.2.1 El rol de la computación en la niebla en el Internet de las Cosas

En años recientes el IoT ha surgido como un nuevo paradigma, en el que un sin fin de dispositivos y objetos son interconectados con una gran variedad de protocolos de comunicación; como lo pueden ser: Bluetooth, Wifi, Zigbee, LoRaWAN, Sigfox, Z-Wave, 6LowPAN; por nombrar algunos. Por lo que actualmente se está experimentando el crecimiento desmesurado de dispositivos conectados al internet. (Yaqoob et al., 2017)

Empresas tanto como Gartner y Cisco anticipan que aproximadamente para el año 2020 existirán entre 20 a 30 mil millones de dispositivos conectados a

Internet; lo que supone una gran agregación de datos hacia los servidores y servicios de la computación en la nube que ponen en operación las aplicaciones IoT. (Nordrum, 2016)

Como se puede observar en la tabla 1, una refinería puede llegar a generar 1 Terabyte de información; de igual manera una turbina de avión solamente en treinta minutos puede alcanzar los 10 Terabytes de datos. Por consiguiente, el gran reto de la computación en la niebla es el de procesar la información en los nodos más cercanos que estarán distribuidos a lo largo desde el borde de la red hasta la computación en la nube.

Por consiguiente, una de las principales funcionalidades de la computación en la niebla es asistir a la computación en la nube con cómputo, almacenamiento y conectividad. Esto se lo realiza mediante nodos geográficamente distribuidos para que los centros de datos no se saturen con el desmedido flujo de datos. Por tanto, es indispensable que la computación en la nube solamente reciba información estrictamente necesaria que no se pueda procesar en el borde de la red.

Tabla 1.

Estadísticas y datos generados por objetos.

El desafío de la agregación de datos	
En una semana: 500 gigabytes. Datos generados por una plataforma petrolera.	En un día: 1.1 billón. Puntos de datos generados por sensores.
Cada 30 minutos: 10 terabytes. Datos generados por una turbina de avión.	En un día: 1 terabyte. Datos generados por refinería de petróleo.

En los últimos dos años:

Se ha generado el 90% de los datos.

En un día:

2500 petabytes.

Datos generados a nivel mundial.

Tomado de (Cisco, 2016).

Años atrás, el IoT tradicionalmente presentaba en su arquitectura una capa de sensores, puertas de enlace y servicios en la nube con el objetivo de proveer de cómputo, conectividad y almacenamiento a estas aplicaciones.

Sin embargo, la vasta cantidad de datos que se produce en días, semanas y años es motivo por el cual la computación en la niebla ha empezado a introducirse en la arquitectura del IoT. En consecuencia, la arquitectura de una aplicación IoT ahora consta de una capa adicional.

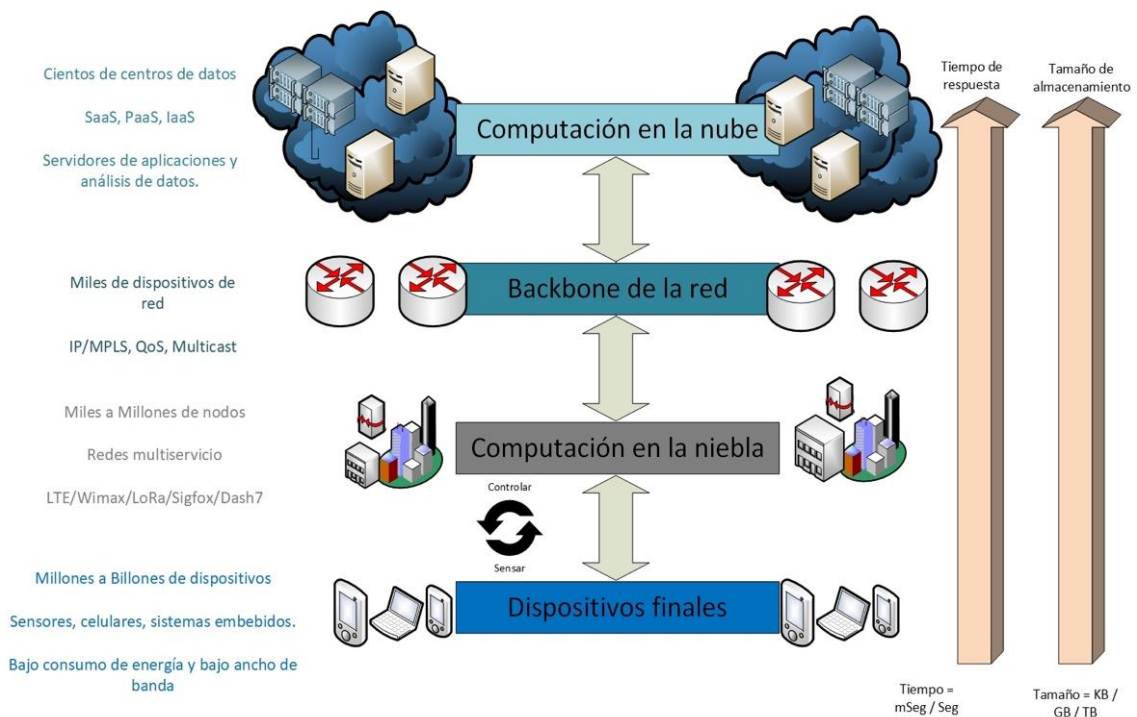


Figura 2. Arquitectura IoT con la computación en la niebla.

Adaptado de (Cisco, 2016).

Como se puede apreciar en la figura 2, la computación en la niebla se ubica entre los dispositivos físicos y el núcleo de la red; además, otro aspecto es que entre el primer y segundo nivel se forma un ciclo o bucle repetitivo. Este ciclo empieza en los dispositivos físicos que envían datos a los nodos de la niebla y estos se encargan de procesar la información y si es que estos se encuentran en la capacidad de tomar alguna decisión, estos lo realizan.

Sin embargo, el procesamiento y almacenamiento en la computación en la nube es más robusto; por lo que, en caso de requerir un mejor rendimiento, los nodos de la niebla o fog se encargan de remitir dicha información a los centros de datos.

De igual manera se puede apreciar que los nodos de la computación en la niebla se encuentran cerca de los dispositivos finales, lo que influye en que el tiempo de respuesta esté en el orden de los milisegundos. No obstante, en la computación en la nube esto llega a tardar varios segundos, teniendo como consecuencia una lenta respuesta hacia un actuador en los dispositivos físicos.

2.2.2 Principales características

La computación en la niebla presenta características únicas que son definidas por el consorcio OpenFog; las cuales están agrupadas en un acrónimo denominado "SCALE".

- **Seguridad:** Ofrece seguridad adicional, brindando confiabilidad en transacciones y transmisión de datos.
- **Conocimiento:** Conocer los objetivos y requerimientos del usuario, para determinar en qué nodos será mejor utilizar el cómputo, almacenamiento y control a lo largo de la nube hasta los dispositivos.
- **Agilidad:** Permite una ágil innovación y fácil escalamiento. Esto se debe a que el consorcio OpenFog está brindando las facilidades para crear una plataforma abierta; así los usuarios pueden utilizar interfaces de programación de aplicaciones (APIs) y kits de desarrollo de software (SDKs) para la operación de nuevos servicios.

- **Latencia:** Respuesta y procesamiento en tiempo real para el control de sistemas ciber-físicos (CPS); como lo pueden ser smart grids, vehículos autónomos, monitoreo médico, domótica etc.
- **Eficiencia:** Permite la agrupación de recursos que tienen los nodos y debido a que estos se encuentran geográficamente distribuidos; es posible distribuir el cómputo, almacenamiento, y control en cualquiera de los nodos disponibles de la red.

2.2.3 Pilares de la arquitectura referencial OpenFog

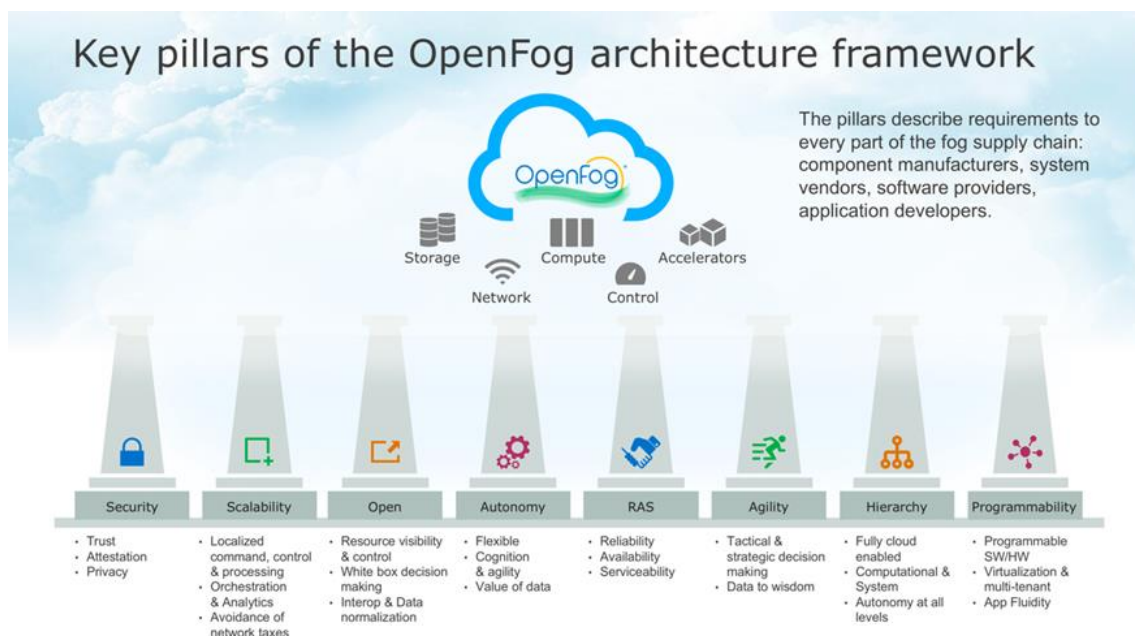


Figura 3. Pilares de la arquitectura referencial de la computación en la niebla.

Tomado de (OpenFog Consortium Architecture Working Group, 2017).

La arquitectura que se encuentra definida por el consorcio OpenFog, cuenta con ciertos lineamientos que deben cumplirse para su implementación.

A continuación, se describe cada pilar que compone esta arquitectura:

2.2.3.1 Escalabilidad

La escalabilidad es uno de los pilares que puede ser aplicado en toda la infraestructura de la computación en la niebla tanto en nodos individuales como en la red propia.

- Los nodos pueden escalar internamente, es decir a través de la adición o mejora de software y hardware; ya que estos pueden ser en un principio un simple servidor con conectividad, router o switch y escalar hacia una mejor infraestructura pudiendo llegar a ser un microcentro de datos.
- La red de la computación en la niebla va a escalar según se agreguen más nodos; en otras palabras, se pueden adicionar más nodos fog en el mismo grado de jerarquía (este-oeste) o en niveles adyacentes (norte-sur); por lo tanto, se asiste a los dispositivos de la niebla que se encuentren más sobrecargados.
- Las principales características para garantizar el escalamiento de esta tecnología, es el almacenamiento, conectividad (con uno o varios nodos) y funcionalidades para el análisis de datos que generalmente lo realizan los nodos fog de niveles superiores.

2.2.3.2 Seguridad

Las aplicaciones IoT que contengan en su arquitectura una infraestructura de computación en la niebla, deberán incluir privacidad y respuesta crítica; ya que varias aplicaciones del Internet de las Cosas son sensibles ante esto, como por ejemplo en el campo de la salud y ciudades inteligentes.

En el campo médico este tipo de aplicaciones se han visto envueltas en vulnerabilidades preocupantes; debido a que en un caso en particular se ha llegado a poner en riesgo la vida de medio millón de personas. Este hecho se dio específicamente en implantes de marcapasos de la empresa St Jude Medical' (SJM), con un potencial muy alto de que puedan ser manipulados por intrusos. De igual manera existen casos en bombas de insulina que pueden ser alteradas en su dosis. (Abdollah & Perrone, 2017; Alvarez, 2016, 2017)

Por otra parte, en aplicaciones como las ciudades inteligentes que se definen como la integración de muchos nodos con varias funcionalidades, tienen el fin de comunicarse entre unas y otras. Esta colección de nodos puede ser: casas inteligentes, oficinas o edificios, transporte, recolección de basura, fuentes eléctricas inteligentes etc. Por lo que, esta información que cruza esta red de nodos también es sensible y crítica en temas de seguridad. (Latif & Zafar, 2017).

En efecto la implementación de un sistema de seguridad puede contener distintos aspectos a considerar como lo son el anonimato, privacidad, integridad, confiabilidad y verificación. Atributos que son claves para la implementación de una capa de seguridad, en una arquitectura de la computación en la niebla. (OpenFog Consortium Architecture Working Group, 2017).

2.2.3.3 Sistema abierto

Tener un sistema abierto es uno de los pilares fundamentales para habilitar distintos nodos de la niebla distribuidos a lo largo de cualquier red.

Además, este pilar también permite que los nodos puedan ser descubiertos automáticamente; esto con el fin para suplir con rapidez alguna aplicación específica y dar soporte a otros nodos.

- Una de las principales características de este sistema abierto es que deba permitir portabilidad, fluidez de aplicaciones, servicios entre distintos nodos y redes que puedan interconectarse.
- Para que exista portabilidad y fluidez durante el emparejamiento de distintos nodos para la prestación de cómputo, conectividad y almacenamiento; es primordial que primero exista la interoperabilidad entre dispositivos mediante una arquitectura común que se maneje tanto en hardware como en software.
- La comunicación abierta que permite la agrupación de recursos de los nodos fog, tanto en cómputo, almacenamiento y conectividad. Por ejemplo, una aplicación que se encuentre implementada a través de una arquitectura de computación en la niebla y requiera una alta demanda de

recursos. Podrá hacer uso de cientos y miles de Gigabytes de procesamiento mediante los nodos fog que se encuentren repartidos en el borde la red. Inclusive el consorcio OpenFog asume que se podrá utilizar los recursos de ordenadores portátiles, escritorios, servidores o todo equipo que se encuentre inactivo. (OpenFog Consortium Architecture Working Group, 2017).

2.2.3.4 Autonomía

Autonomía ante fallas externas, en la que ciertas veces ante la inesperada indisponibilidad en los centros de datos, la computación en la niebla debe continuar entregando el servicio a los usuarios finales a través de los distintos niveles jerárquicos.

Generalmente los dispositivos IoT tienen sus servicios o aplicaciones alojados en la nube; por lo que, si existe un fallo en un enlace uplink, el dispositivo se encontrará fuera de línea; no obstante, los nodos fogs con características de autonomía pueden actuar como un proxy, para el registro del dispositivo y así evitar que el sensor o dispositivo necesite reportarse en el backend de la nube.

También es importante destacar la autonomía en operación; es decir que en una arquitectura IoT no solamente debe existir la toma decisiones en un punto centralizado o en la nube; por lo tanto, los nodos de la niebla deben satisfacer las peticiones de los usuarios aun cuando la nube pueda estar inoperativa.

Por último, la autonomía igualmente tiene de ventaja el ahorro en costos de conectividad, ya que mientras más datos son enviados hacia la computación en la nube, el costo de este enlace será mayor debido al transporte de la información por varios proveedores de servicios (ISP). Lo que inclina aún más en realizar el procesamiento en el borde de la red.

Por ejemplo muchas de las refinerías de petróleo poseen conectividad mediante enlaces satelitales debido a que generalmente se encuentran en lugares inhóspitos. Por lo tanto si en aquellas plantas se genera en promedio 1 Terabyte

de datos al día, no es prudente el envío de toda esa información a través de un enlace satelital, ya que resultaría altamente costoso. Por el contrario, los nodos fog se verán en la obligación de procesar esta información en el borde de la red y en ciertos casos específicos enrutar la información crítica y estrictamente necesaria para ser analizada en un centro de control de red.

2.2.3.5 Programabilidad

La capacidad de que estos nodos puedan ser programados y permite implementaciones altamente adaptables; en otras palabras, esto significa que un nodo de la niebla o un grupo de ellos, puedan ser reasignados dinámicamente para suplir alguna otra tarea o asistiendo a otro sistema que requiera de más recursos para el procesamiento de la información.

2.2.3.6 Confiabilidad, Disponibilidad y Servicio

Principalmente en el hardware y software es donde residen estas tres características.

- Una implementación confiable que asegure la continua entrega de recursos y funcionalidades ante operaciones normales y adversas. Esto se asegura mediante la operación confiable del hardware y del software en la que nodo reside.
- La disponibilidad que se garantiza mediante la administración continua de los nodos en los distintos niveles de jerarquía mediante las siguientes funciones: acceso seguro y remoto desde varios dispositivos, redundancia de los servicios y capacidad de acceso como un nodo mesh hacia cualquier dispositivo IoT.
- El servicio en el que se asegurara la correcta operación de todos los recursos y capacidades que puede brindar un nodo de la niebla a las diferentes aplicaciones del Internet de las Cosas

2.2.3.7 Agilidad

Las personas no están en la completa capacidad de analizar la impredecible y vasta cantidad de datos que generan los sensores y sistemas IoT. Por lo que estos datos generalmente son analizados y procesados en servidores dedicados en la nube; consecuentemente esto conlleva altos tiempos de respuesta. Es por eso que es indispensable que esta capacidad de análisis de datos y procesamiento se realice lo más cercano de su origen.

Como se mencionó anteriormente, esto también evita la dependencia de que los datos tengan que enrutarse hacia el núcleo de la red, provocando un alto costo de transporte de datos hacia la nube. Por otra parte, un aspecto importante de la computación en la niebla es el acrónimo DIKW, la cual está descrita en inglés: *“Data gathered becomes Information when stored and retrievable becomes Knowledge. Knowledge enables Wisdom for autonomous IoT”*. Lo que quiere decir que los datos almacenados inmediatamente se convierten en información, la información que se procesa y se extrae es conocimiento; por tanto, el conocimiento provee de sabiduría en la toma de decisiones en un ambiente IoT. (OpenFog Consortium Architecture Working Group, 2017)

Este principio es fundamental para el análisis y procesamiento de datos al borde la red debido a que en la computación en la niebla es posible la creación de varios niveles jerárquicos. Esto implica que se puede dotar de mas procesamiento a los equipos en niveles superiores por lo que la información puede ir escalando en estos niveles inclusive hasta llegar a la computación en la nube para que estos datos sean convertidos en sabiduría. Esta transformación se puede apreciar a continuación en la figura 4.

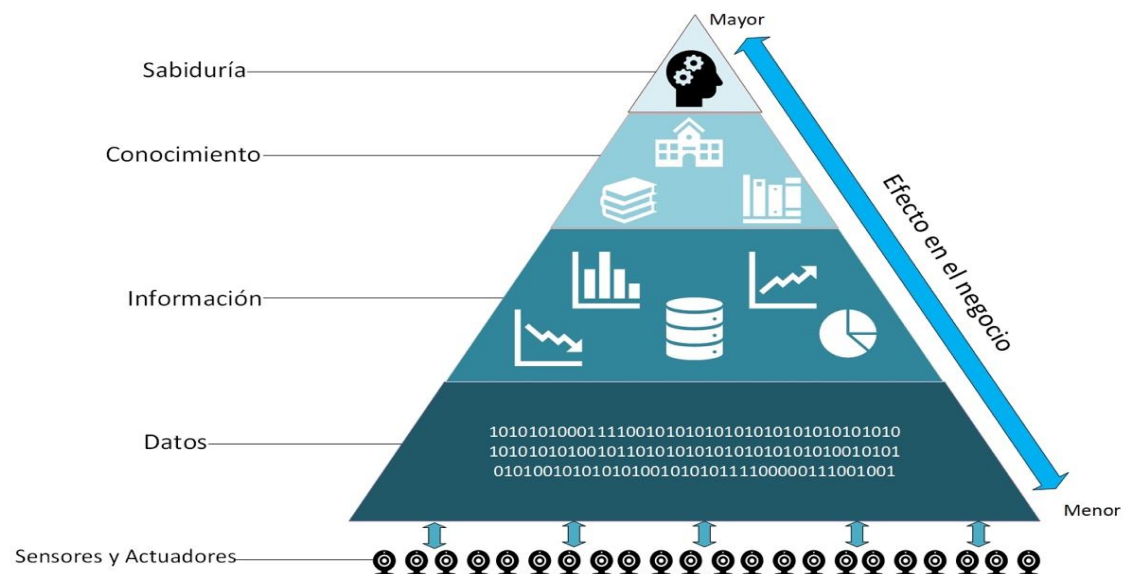


Figura 4. Transformación de datos en sabiduría

Adaptado de (Cisco, 2016)

2.2.3.8 Jerarquización

La jerarquización es un pilar que no es requerido en todas las arquitecturas de la computación en la niebla; pero aun así es tomado en la mayoría de las soluciones IoT. Para que se jerarquice por niveles dichas soluciones, la implementación debe ser una aplicación que requiera requisitos funcionales de una aplicación IoT extremo a extremo.

Este pilar podemos encontrarlo desde edificios hasta una ciudad inteligente, tomando como ejemplo una pequeña oficina que posiblemente requiera una arquitectura de computación en la niebla de forma local; por el contrario, una empresa grande con varias sucursales ubicadas en distintas ciudades debería tener distribuido esta arquitectura a niveles locales y regionales.

2.2.4 Arquitectura

La arquitectura referencial de la computación en la niebla es una representación abstracta del funcionamiento de los nodos fog y ciertamente de cómo manejar varios escenarios para satisfacer una aplicación IoT en específico.

La manera en que el software y hardware es implementado en un sistema de computación en la niebla es importante según el escenario en el que se vaya a desarrollar. En la mayoría de los casos en los que se implementa esta arquitectura se puede apreciar que existen distintos niveles en los que se encuentran distribuidos los nodos como se puede observar en la figura 5.

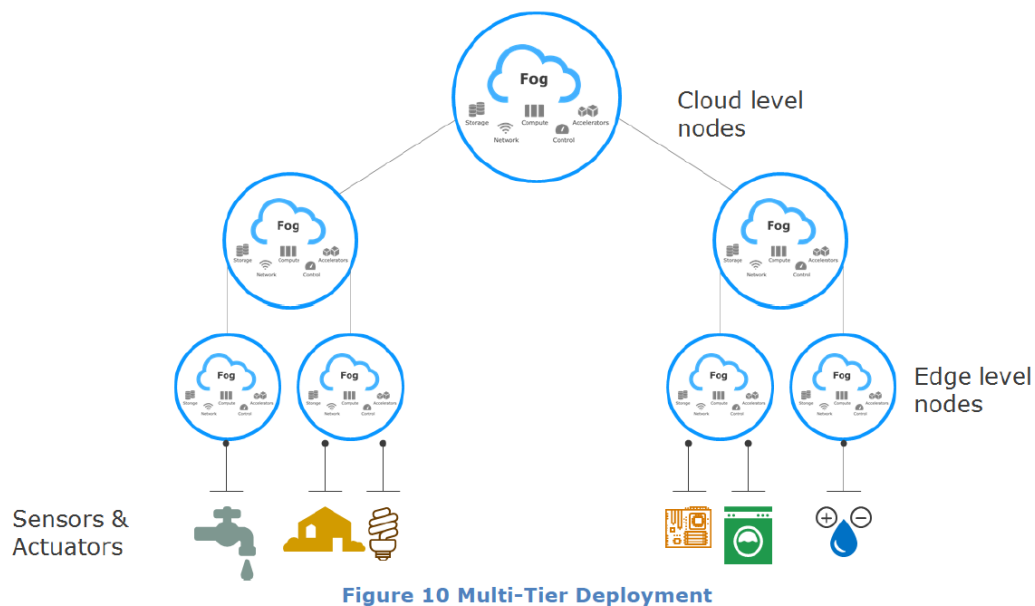


Figura 5. Arquitectura de computación en la niebla.

Tomado de (OpenFog, 2017).

Para una mejor comprensión de esta arquitectura, a continuación, se desarrolla un ejemplo que describe como funciona y en que determinada acción se utiliza las jerarquías en la computación en la niebla para una planta de procesamiento de alimentos.

Los nodos en el borde de la red están enfocados en atender los datos que son recolectados por los sensores.

En una planta de alimentos los nodos más cercanos a los usuarios finales o en este caso a los sensores, atenderán a la banda transportadora de alimentos con un tiempo de respuesta bajo los milisegundos; por tanto, actúan en la

determinación de separar los alimentos dañados o contaminados con los saludables para la siguiente fase de producción.

Los nodos en el siguiente nivel tienen como principal tarea el de filtrar y transformar los datos en información. En este nivel puede haber la posibilidad de que exista análisis de datos para la toma de decisiones críticas en tiempo real; por consiguiente, que mientras más se alejen los nodos de los dispositivos finales, mejores prestaciones tendrán hasta el punto de llegar a tener aprendizaje automático o como comúnmente más se lo conoce “Machine Learning”.

Retomando el ejemplo anterior, este nivel opera más alejado de los sensores; por lo que está encargado de controlar tanto la cinta transportadora de alimentos como la envoltura de alimentos y sistemas para que exista una fluidez y un tiempo de respuesta rápido en la producción.

Los nodos que se encuentren en el núcleo de la red ya próximos al backend de la nube, tienen como objetivo el almacenamiento indefinido de la información para transformarla en conocimiento y posteriormente en sabiduría. Mientras estos nodos se encuentren más alejados de los usuarios finales, mejores prestaciones tienen para el procesamiento de la información.

Por otra parte, en otras aplicaciones el análisis más profundo de los datos se encontrará en el borde de la red como por ejemplo las cámaras de vigilancia; en realidad a la cantidad descomunal de información que pueden generar estos dispositivos a comparación de otros.

En cambio, en un modelo de ciudad inteligente, podemos incluir *road side units* (RSUs), edificios, postes, semáforos entre muchos otros como nodos de la computación en la niebla; consecutivamente habrá aún más nodos que abarquen barrios, luego parroquias y por último varios nodos a nivel cantonal, provincial e inclusive regional.

La determinación para diseñar e implementar una arquitectura será según el nivel de escalabilidad que necesite la aplicación:

- Número de sensores.
- Tipo de red requerida en cada nivel.
- Prestaciones de los nodos según su jerarquía.
- Latencia entre nodos y latencia entre sensores y actuadores.
- Confiabilidad y disponibilidad de los nodos.

Además, como se puede observar en la figura 6, los nodos de la computación en la niebla tienen la capacidad de comunicarse de manera horizontal (este-oeste) y de igual manera de arriba hacia abajo (norte-sur).

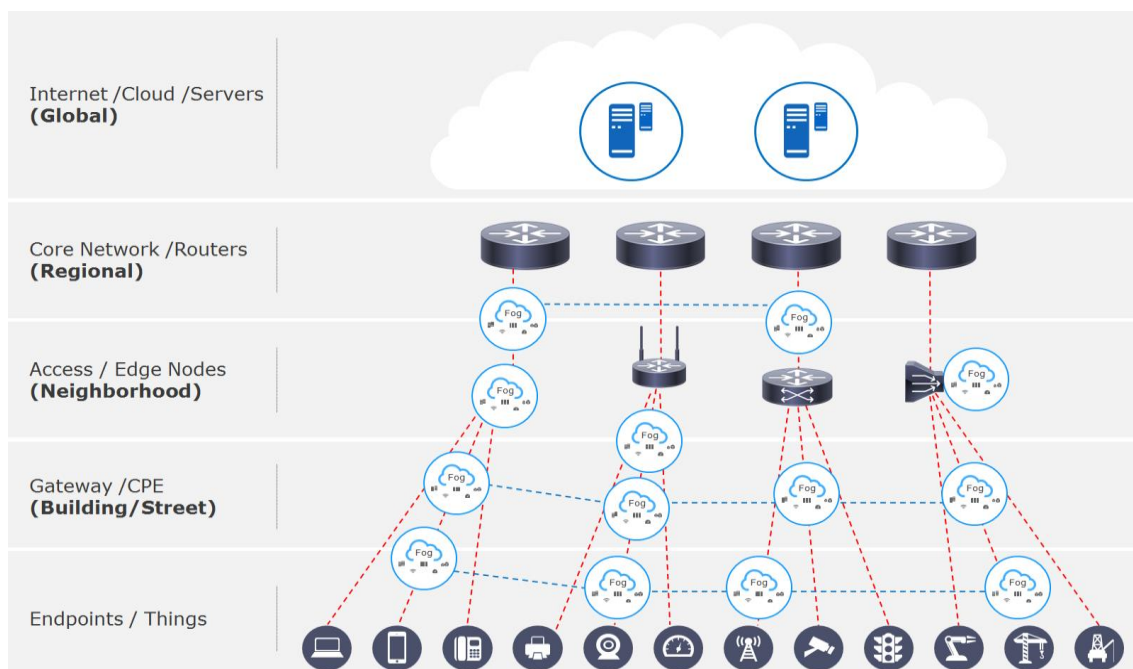


Figura 6. Arquitectura de computación en la niebla para una ciudad inteligente.

Tomado de (OpenFog, 2017).

2.3 La computación en la niebla y la computación en la nube

En la actualidad la computación en la nube es globalmente utilizada por los internautas; ya que proporciona una plataforma con acceso bajo demanda hacia recursos de almacenamiento, procesamiento y aplicaciones, en todo momento y en cualquier lugar.

Por eso la nube provee tres servicios esenciales que son denominados como:

- Infraestructura como servicio (IaaS): El usuario accede a recursos virtualizados como lo son el cómputo, almacenamiento y conectividad.
- Plataforma como servicio (PaaS): Provee de entornos de desarrollos integrados (IDE), para el manejo y desarrollo de aplicaciones.
- Software como servicio (SaaS): Provee todo tipo de aplicaciones software para usuarios finales.

Se reporta que al menos el 90% de los usuarios de internet utiliza servicios basados en la computación en la nube, ya sea contratando directamente el servicio bajo demanda o indirectamente a través de proveedores que alojan sus servicios en los centros de datos. (Ahmad et al., 2015)

Los usuarios cada vez solicitan más servicios que van desde almacenamiento en la nube, atención médica, dispositivos wearables, navegación, Internet de las Cosas, mensajería instantánea, realidad aumentada, entre muchas otras.

Por lo que al manejar esta gran cantidad de servicios que son demandados por billones de usuarios, es que de poco a poco empieza a convertirse en una tarea pesada y difícil para la nube; por tanto, se genera una necesidad de liberar ciertas tareas de procesamiento de la nube (Centros de Datos) hacia la computación en la niebla.

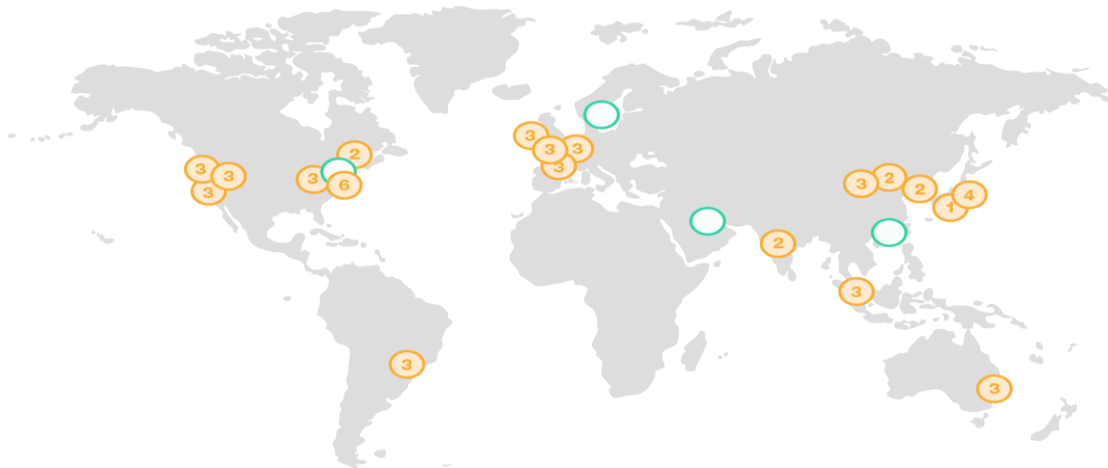


Figura 7. Computación en la nube de AWS que opera en 42 zonas alrededor del mundo.

Tomado de (Amazon Web Services, 2018).



Figura 8. La nube de Google que opera en 8 regiones con 23 zonas y alrededor de 100 puntos interconectados a través de miles de kilómetros de fibra óptica.

Tomado de (Google Cloud, 2018).

En las figuras 7 y 8, se puede observar que la mayoría de centros de datos de AWS y Google Cloud están geográficamente distantes varios kilómetros entre unos y otros. Por lo tanto se generan altos retardos tanto en *uplink* como

downlink, congestión en el núcleo de la red y degradación de la calidad de servicio (QoS) mientras se trata de satisfacer las peticiones y requerimientos de los usuarios finales. (Mukherjee, Shu, & Wang, 2018)

Para hacer frente a esta situación debido a la gran distancia entre dispositivos IoT y la computación en la nube, se ofrece los siguientes beneficios mediante la computación en la niebla a aplicaciones altamente sensibles al tiempo de respuesta:

- Provee de pronta respuesta ante aplicaciones que requieran asistencia en tiempo real, como lo son alarmas contra prevención a desastres y redes ad-hoc vehiculares.
- La posibilidad de la agregación de información en dispositivos interoperables. Como por ejemplo la recepción de datos de varios sensores o dispositivos médicos de varios fabricantes. Por otra parte, en la computación en la nube esto es difícil de alcanzar, ya que solamente se ofrece un servicio individual por cada marca de dispositivo que el usuario tenga.
- Brinda seguridad y protección de información sensible como registros médicos y localización de usuario; por lo tanto, evita la transmisión de datos innecesarios mediante filtros para determinar qué información se envía al core de la red y posteriormente a la nube.

Tabla 2.

Diferencias entre la computación en la nube y la computación en la niebla.

Características	Computación en la nube	Computación en la niebla
Infraestructura	Centros de Datos	Desde un simple servidor hasta un microcentro de Datos

Costo de despliegue	Alto	Medio - Bajo
Cobertura geográfica y proximidad	Global	Local (Edificio a Ciudad)
Almacenamiento de datos	Días o Años	Almacenamiento temporal o transitorio
Comunicación horizontal	No	Si
Latencia	Alto	Bajo
Confiabilidad	Alto	Bajo
Nodos	Centros de Datos	Servidores, routers, switches, gateways.

Adaptado de (Aazam, Zeadally, & Harras, 2018; Mukherjee et al., 2018).

Además es importante recalcar que en las diferencias descritas en la tabla 2, la computación en la niebla se caracteriza por tener una comunicación este-oeste u horizontal; es decir que los nodos de la niebla debido a su arquitectura distribuida, van a poder comunicarse y asistirse entre ellos para procesar información y no sobrecargar solamente a uno; sin embargo en la computación en la nube sigue habiendo más confiabilidad debido a las altas prestaciones en infraestructura y redundancia que ofrecen los centros de datos; que por el contrario en la computación en la niebla no se presentan.

En fin, es evidente la diferencia entre estas dos tecnologías, pero esto no debe ser un punto de quiebre para la elección de una de ellas en una aplicación IoT; por lo que justamente es importante recalcar que ninguna se sustituye entre sí. Por el contrario, las dos se interrelacionan formando una sola arquitectura para satisfacer de mejor manera al Internet de las Cosas.

2.4 Tecnología Inalámbrica

Con el éxito que ha supuesto la tecnología inalámbrica en dispositivos electrónicos de consumo comercial. Esta también se ha ido adentrando de poco a poco en el ámbito industrial, debido a la facilidad de poder enlazar sensores y dispositivos a la red de manera más fácil que una conexión cableada. Los dispositivos tradicionales con capacidades inalámbricas como Bluetooth, infrarrojo entre otras.

2.4.1 Redes de área extendida de bajo consumo

Las redes de área amplia de bajo consumo o comúnmente más conocidas como (LPWAN), surgen ante la necesidad de poder conectar cada objeto o dispositivo con prestaciones o recursos muy limitados. Generalmente estos objetos que son de IoT trabajan con tasas de transferencias bajas que van desde los bps (bits por segundo) hasta los Mbps. Los rangos de operación también pueden extenderse desde 1 metro hasta varios kilómetros. Por lo que las LPWAN generalmente utilizan las bandas sub-GHz o ISM (Industrial, Scientific and Medical) con 915 Mhz en Estados Unidos, 868 Mhz en Europa y 433 Mhz en Asia.

Las LPWAN están diseñadas para cubrir una vasta cantidad de aplicaciones específicas mediante distintos estándares y protocolos diseñados para cada fin. Sin embargo, los distintos protocolos de LPWAN deben cumplir los mismos requerimientos para el diseño de una red LPWAN. Las características principales que deben presentar son las siguientes:

- El bajo consumo de energía para evitar el cambio de baterías recurrente en el objeto. El impacto ambiental del cambio de batería de miles de millones de dispositivos y los altos costos de mantenimiento en sustituirlas supone un efecto muy negativo a tomar en cuenta en una aplicación IoT.
- Las restricciones económicas son un factor primordial al momento de desplegar cualquier solución tecnológica. La conectividad de cada sensor a la red comúnmente como se o realiza mediante redes móviles supone

un alto costo en la aplicación IoT. Por el contrario, en la tecnología LPWAN no se requiere de tarjetas SIM, son fáciles de instalar y necesitan poco mantenimiento.

- El nivel de actividad del nodo puede variar según la aplicación que se esté implementando. Por lo que la red LPWAN como tal debe mantener a los nodos en estado de reposo mientras no necesiten enviar o recibir tráfico.
- La infraestructura de red debe ser fácil de implementar a nivel local y nacional, que provea una buena cobertura y brindar movilidad entre distintos países.
- La transferencia de datos entre el objeto y el usuario final debe estar complementemente asegurada y cifrada. El operador de red no debe de poder acceder a la información transportada y además el enlace debe ser robusto ante inhibidores de señal. (Bardyn, Melly, Seller, & Sornin, 2016)

2.4.2 LoRa/LoRaWAN

La arquitectura de LoRa fue originalmente desarrollada por Cycleo en Francia, pero después fue adquirida por Semtech en 2012 por 5 millones de dólares. La alianza de LoRa fue creada en marzo de 2015. Esta alianza es la encargada de la estandarización de LoRaWAN, sus especificaciones y tecnología. Dicha alianza está conformada principalmente por IBM, Cisco y otros 160 miembros.

LoRa representa la capa física de una red LoRaWAN. Es la encargada de la modulación, potencia de transmisión y recepción de la información. Su arquitectura está basada en las siguientes bandas ISM.

- 915 Mhz: En los Estados Unidos establecido con límites de potencia, pero sin límite de ciclo de trabajo.
- 868 Mhz: Banda asignada en Europa con 1% y 10% de ciclo de trabajo.
- 433 Mhz: Frecuencia Asignada para Asia.

La técnica de modulación utilizada en LoRa es la denominada Chirp Spread Spectrum (CSS). CSS fue utilizado por primera vez en 1940 en la milicia para

comunicaciones de largo alcance mediante el uso de los pulsos de ruido (chirp) para la codificación de los datos. Además, que es robusta y resistente a interferencias de efectos Doppler y multitrayecto. Los pulsos de ruido son ondas sinusoidales que aumentan o disminuyen con el tiempo, por lo que la tasa de bits de LoRa es una función de la tasa de pulsos de ruido y de símbolos. Esta expresión puede ser representada de la siguiente manera:

$$R_b = S \times \frac{1}{\frac{2^s}{B}} \quad (\text{Ecuación 1})$$

Donde:

R_b: Tasa de bit (bps).

S: Factor de dispersión (chips/s).

B: Ancho de banda (Hz).

Esta forma de modulación es la que permite la transmisión de información a largas distancias con bajas potencias. Mediante la ecuación 1 se puede determinar que mientras el factor de dispersión es mayor, la transmisión será más lenta pero también un mayor rango de alcance y cobertura.

Además de su modulación, LoRa posee una característica especial llamada Adaptive Data Rate (ADR) la cual es una función dinámica y escalable en función del número de nodos en una red LoRa. Lo que permite configurar la velocidad de datos según la aproximación de los nodos. También, los nodos que se encuentren más cerca de una estación base podrán transmitir su información, liberar el ancho de banda del canal y entrar rápidamente a un estado de reposo en comparación con los nodos distantes que tomará más tiempo su transmisión.

Por otra parte, LoRaWAN representa la capa de enlace mediante direcciones MAC que está por encima de la capa física de LoRa. Cabe recalcar que LoRaWAN es un protocolo abierto mientras que LoRa no lo es. Existen tres

protocolos MAC que son parte de la capa de enlace y según uno de estos se balancea la latencia y el consumo de energía.

Los dispositivos con clase A, generalmente son sensores basan principalmente su alimentación en baterías. Cabe mencionar que todo nodo que se une a una red de LoRAWAN se identifica como un nodo clase A. La clase A optimiza el consumo de energía mediante el establecimiento de un delay. El sensor transmite la información y tiene que cumplir este tiempo de retraso para volver a transmitir los datos que recoge. Mientras el tiempo de retraso se cumple el nodo siempre entra en modo de reposo y se levanta del modo de reposo solamente para enviar la información.

En el protocolo de clase B, se prioriza el tiempo de retraso y consumo de energía. Estos dispositivos actúan como beacons a los cuales se les asigna un tiempo para poder transmitir su información. Por lo que se sincroniza con todos los nodos de la red para poder transmitir y receptar en ese lapso cualquier información.

En los dispositivos de clase C se prioriza la latencia con la posibilidad de transmisiones continuas y con la mayor potencia de transmisión. Generalmente estos dispositivos son actuadores o dispositivos energizados. A continuación en la figura 9 se puede observar con mas claridad todos los protocolos que involucran a LoRa desde su capa física hasta la capa de aplicación del modelo OSI.

LoRa / LoRaWAN Protocol Stack			Simplified OSI Model
Application Layer			7. Application Layer
LoRaWAN Layer			2. Data Link Layer
Class-A (Baseline)	Class-B (Baseline)	Class-C (Continuous)	
LoRa PHY Modulation			1. Physical Layer
LoRa PHY Regional ISM Band			
Lora PHY EU Band 868 MHz	Lora PHY EU Band 433 MHz	Lora PHY US Band 915 MHz	

Figura 9. El stack de protocolos de LoRa y LoRaWAN en contraste con el modelo OSI.

Tomado de (Lea, 2018)

LoRaWAN está basado en una topología de red tipo estrella. Los nodos de LoRa tienen la capacidad de conectarse y asociarse a múltiples gateways para el envío de su información. Los gateways de LoRa pueden comunicarse a los servidores de la computación de la nube mediante LTE, fibra óptica, satélite o cualquier red de agregación disponible como se puede observar en la figura 10.

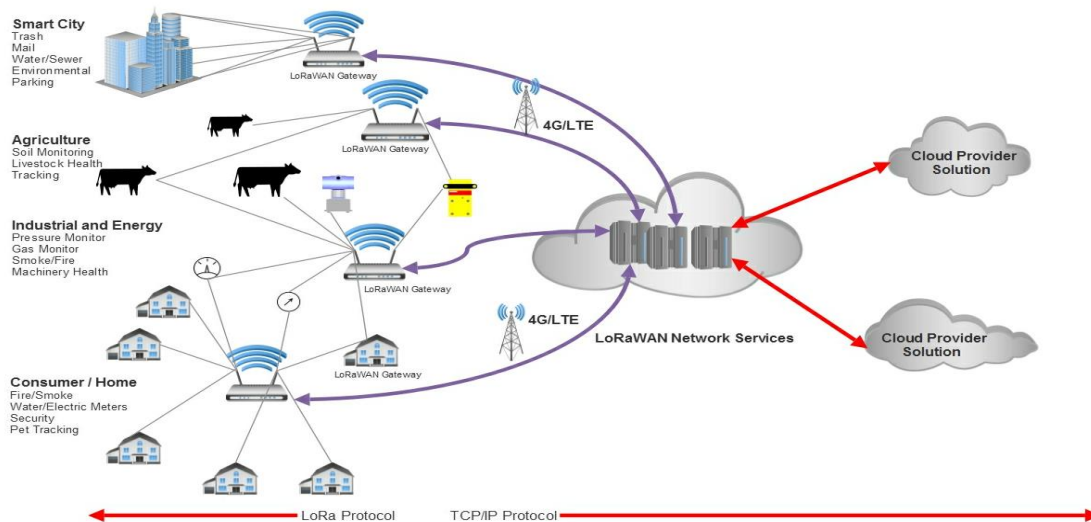


Figura 10. Topología de una red LoRaWAN.

Tomado de (Lea, 2018)

2.4.3 Sigfox

Sigfox es un tipo de red LPWAN pensada para ser utilizada en el IoT. Su funcionamiento es similar al de las operadoras móviles en telefonía celular debido que opera a partir de la ubicación de radio bases receptoras y transmisoras en las bandas ISM. En Europa opera en los 686 Mhz, y en Norte América en los 902 Mhz. Los mensajes que se envían mediante Sigfox no solo se receptan en una radio base; por lo contrario, en promedio 3 radio bases reciben los mensajes de un dispositivo que se encuentre en el rango de cobertura.

Sigfox se caracteriza por utilizar la tecnología UNB (Ultra Narrow Band) mediante la cual se puede alcanzar enlaces de larga distancia desde los 10 km hasta los 50 km en áreas rurales

Utiliza un canal de 200 KHz en las bandas de frecuencia de uso libre en la que cada mensaje de bajada tiene un ancho de banda de 100 Hz lo cual son de pequeñas dimensiones. Las tasas de transferencia que puede alcanzar Sigfox son entre 100 y 600 bps.

La modulación que utiliza es la Orthogonal Sequence Spread Spectrum (OSSS) para el enlace de uplink y Gaussian Frequency Shift Keying (GFSK) para el enlace descendente con un ancho de banda de 600 Hz. Sigfox envía los mensajes escogiendo un canal y tiempo de retraso aleatorio entre los 500 y 525 ms. Este tipo de codificación de los datos se la denomina Random Frequency and Time Division Multiple Access (RFTDMA).

Tabla 3.

Especificaciones principales de Sigfox.

	Enlace ascendente	Enlace descendente
Límite de payload	12 bytes	8 bytes

Throughput (bps)	100	600
Mensajes máximos por día	140	4
Esquema de modulación	DBPSK	GFSK
Sensibilidad (dBm)	<14	<27

En la tabla 3 se puede observar las características más importantes de Sigfox. Como por ejemplo que los mensajes son muy pequeños ya que tienen 12 bytes de payload en el enlace ascendente y 8 bytes en el enlace descendente. Lo que le convierte en una de las tecnologías LPWAN con menos consumo de energía y con mayor probabilidad de duración de batería en los sensores y dispositivos que la utilizan.

La topología de Sigfox consiste en una red tipo estrella como se puede apreciar en la figura 11, en la que varios nodos según su proximidad a una estación base estarán enlazados a esta. Cuando la estación recepta los mensajes, los enruta hacia el usuario final por el núcleo de la red hacia un servidor de la computación de la nube de Sigfox para así verificar que no haya duplicación de mensajes y enviarlo al usuario final correspondiente.

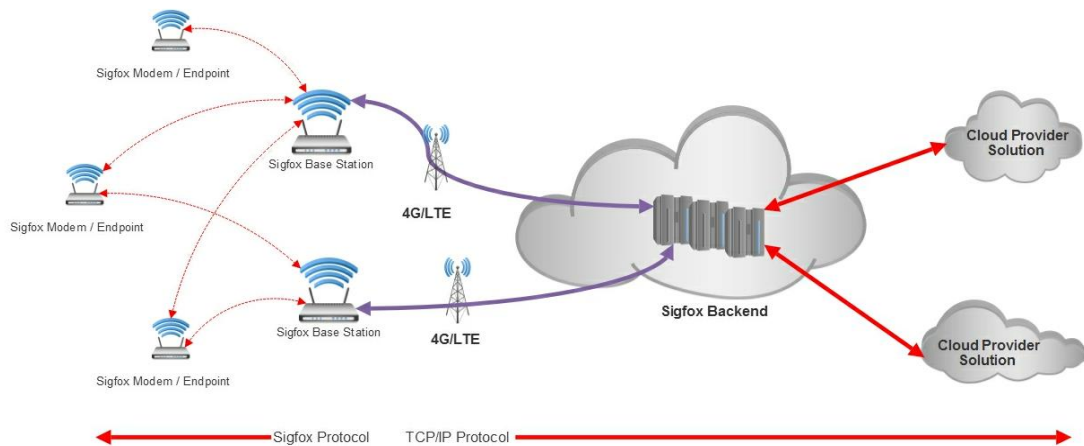


Figura 11. Topología de una red Sigfox.

Tomado de (Lea, 2018)

2.4.4 WiFi Halow

Wifi Halow o también como se lo conoce por su estandarización IEEE 802.11ah, es una tecnología de la familia IEEE 802.11 que tiene como principal característica la utilización de las bandas de frecuencia de uso libre. Está basado en la arquitectura de la capa física de la estandarización 802.11 ac.

El objetivo de la creación de IEEE 802.11 fue en primera instancia el dotar de cobertura a las áreas rurales y descargar el tráfico de datos a las redes celulares. El segundo propósito fue el de utilizar una familia del protocolo IEEE 802.11 en las bandas sub-Ghz. La especificación de este estándar fue publicada el 31 de diciembre de 2016 y esta se diferencia en las siguientes características de los otros estándares de WiFi.

- Opera en el espectro de los 900 Mhz lo que permite una mejor propagación y penetración ante materiales que absorben las ondas electromagnéticas.
- El ancho del canal puede variar y establecerse en 2, 4, 6, 8 o 16 Mhz de ancho de canal.

- Los métodos de modulación son diversos los cuales figuran el BPSK, QPSK, 16-QAM, 64-WAM y 256-QAM de técnicas de modulación.
- Soporta SU-MIMO y beamforming.
- La posibilidad de enlazar miles de dispositivos en un solo punto de acceso.
- Permite el manejo avanzado del consumo de energía de cada nodo conectado.
- Sectorización lo que permite el agrupamiento de antenas para cubrir áreas específicas.

El throughput mínimo que logra WiFi halow es de 150 kbps utilizando a BPSK como técnica de modulación en una canal de 1 Mhz con un solo flujo de MIMO. Mientras que el throughput máximo teórico es de 347 Mbps con una modulación 256-WAM utilizando 4 flujos de MIMO en canales de 16 Mhz.

Es importante tomar en cuenta que el ancho del canal dependerá de la región en la que se está implementando. Ciertas variaciones de canal entre 1 y 16 Mhz no pueden ser implementadas en algunas regiones debido a los marcos regulatorios de cada país como se puede observar en la figura 12.

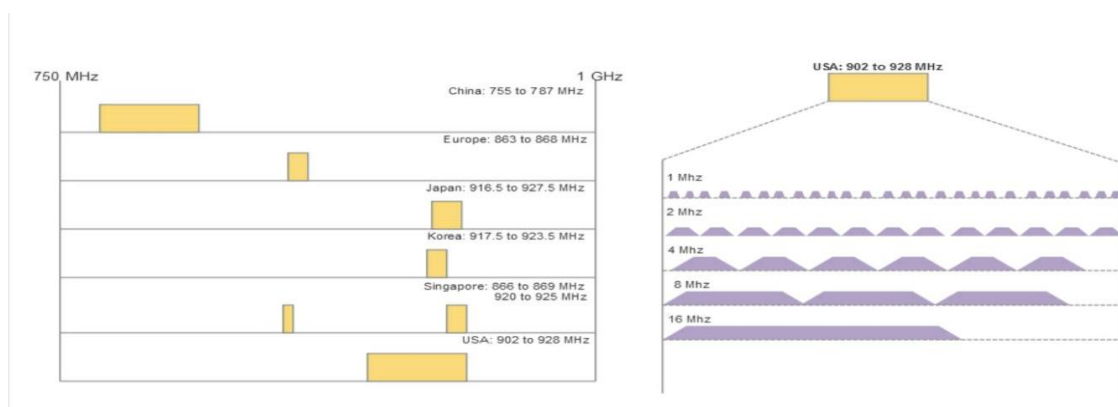


Figura 12. Frecuencias asignadas según el marco regulatorio de cada país.

Tomado de (Lea, 2018).

3 Sistemas de Transporte Inteligentes

3.1 Introducción

La creciente demanda en movilidad de las últimas décadas ha provocado que las ciudades se encuentren con un parque automotor caótico e insostenible. Además, esto causa embotellamientos, tráfico pesado, accidentes y emergencias impredecibles que deben ser atendidas oportunamente. Por otra parte, la ineficiencia en los sistemas de transporte actuales también ha causado la pérdida de tiempo, altos niveles de contaminación e inseguridad en las vías tanto para vehículos como peatones. (Dimitrakopoulos & Demestichas, 2010)

En la década de los años 1980 un pequeño grupo de transportistas profesionales, empiezan a reconocer el impacto que tendrían las tecnologías de la información en futuras décadas. Por tanto, se propone el término de sistemas de transporte inteligente o como se lo conoce por sus siglas STI.

STI es un sistema integral de gestión y servicio para los usuarios de las vías, que tiene como objetivo proporcionar servicios innovadores que estén relacionados con distintos modos de operación de transporte. STI combina las TI (Tecnologías de la Información), comunicación, sensores y métodos matemáticos con la infraestructura de transporte convencional actualmente implementada. (Lin, Wang, & Ma, 2017).

3.2 Redes Ad-Hoc

Las redes ad-hoc son redes inalámbricas descentralizadas; es decir que son capaces de formar una topología de red sin depender de una infraestructura fija, como por ejemplo un router o switch. Los nodos en una red ad-hoc se encuentra equipados con un radio transmisor y receptor, que permite a estos comunicarse con otros nodos mediante enlaces inalámbricos.

La comunicación entre dos nodos no siempre será directa, debido a que puede ser mediante múltiples saltos. Por lo que se generan rutas dinámicas que permiten a cada nodo actuar como un router; en otras palabras, cada nodo puede reenviar la información hasta el destino requerido.

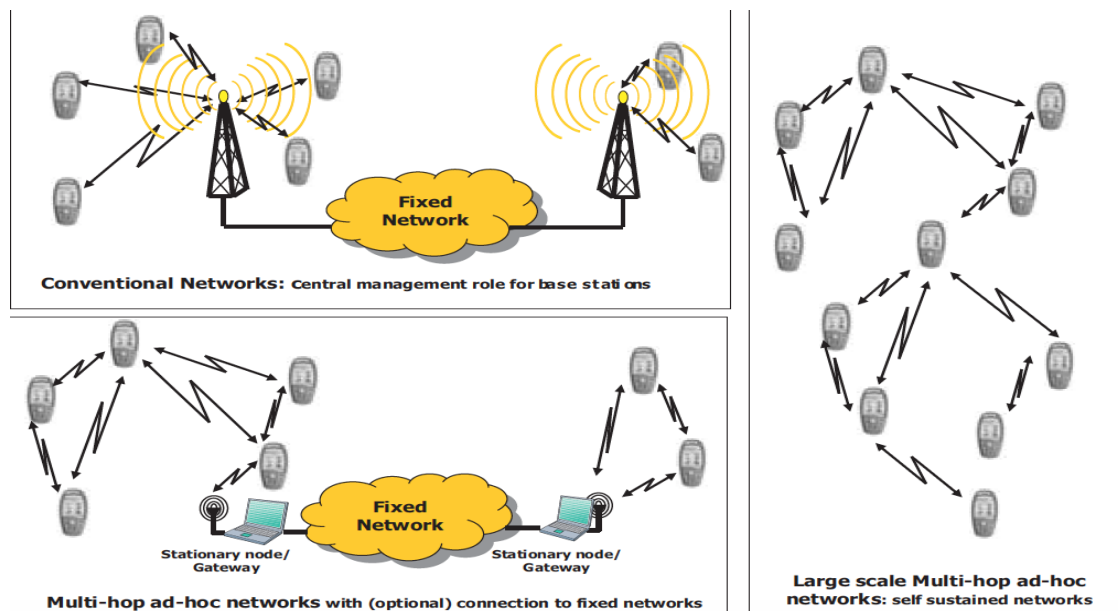


Figura 13. Diferentes topologías de redes ad-hoc.

Tomado de (Hekmat, 2006)

Como se puede observar en la figura 13, una red convencional tiene una topología de red tipo estrella. Mientras que una red ad-hoc tiene conectividad o enlaces directos con nodos vecinos que se encuentren en el rango de alcance del nodo. También hay la posibilidad de que se incorpore un gateway en la topología de red.

Consecuentemente, las redes ad-hoc generan ciertos beneficios tales como la movilidad, flexibilidad, ahorro en medios físicos de transmisión y equipos centralizados como routers y mejoras en la escalabilidad de la red.

A continuación, se detallará los tipos de redes ad-hoc que existen:

3.2.1 MANET

Redes móviles ad-hoc o como se lo conoce por sus siglas en inglés MANET. La principal característica en este tipo de redes es la movilidad y el enrutamiento dinámico que se genera entre los nodos dentro de esta topología de red.

La implementación y utilización de este tipo de redes puede suponer en varios inconvenientes. Como por ejemplo la pérdida de paquetes, degradación de potencia recibida, bajas tasas de transmisión y un alto consumo de energía.

Para transportar información se utiliza protocolos de enrutamiento dinámico, lo que favorece al envío de información según la ubicación de los nodos que se encuentren cercanos.

3.2.2 VANET

Las redes vehiculares ad-hoc (VANET) se derivan de las MANET; sin embargo, en este tipo de red, los nodos serán vehículos o cualquier tipo de infraestructura relacionada con los sistemas de transporte.

El objetivo principal de las VANET es el de ayudar a un vehículo o a un grupo de ellos para establecer y mantener una comunicación entre los distintos nodos sin la utilización de un Gateway. Debido a la falta de infraestructura que esto implica es que en una VANET habrá mayor responsabilidad por parte de los vehículos. Por lo que, cada uno pasa a formar parte de la red pudiendo establecer distintos tipos de comunicación. (Sabih ur, Khan, Zia, & Zheng, 2013).

Los componentes principales que son responsables de establecer los distintos tipos de comunicación son los siguientes:

- Vehículos: Los vehículos tienen conectividad mediante las unidades de comunicación a bordo, además de poder tener sensores o dispositivos de proximidad tales como LIDAR, GPS entre otros. Si un vehículo se encuentra en el rango de conexión de un nodo se puede establecer una

comunicación. Estas pueden ser V2V (Vehículo a Vehículo) V2I (Vehículo a Infraestructura) y V2X (Vehículo a Cualquier objeto).

- Unidad de comunicación abordo: Consideramos que cada vehículo contiene unidades de comunicación abordo o como más se lo conoce por sus siglas en inglés OBU (On-Board Unit). Cada OBU será la responsable de que los vehículos puedan enlazarse con otro nodo mediante distintos tipos de tecnologías inalámbricas como por ejemplo DSRC (Dedicated Short Range Communications), IEEE 802.15 (Bluetooth o Zigbee), identificación por radiofrecuencia (RFID), entre muchas otras.
- Infraestructuras: Las infraestructuras o comúnmente más conocidas como RSU, del inglés Road-Side Unit. Son nodos de las VANET que pueden ser emplazamientos, postes de luz e inclusive semáforos con conectividad inalámbrica que contienen equipos de cómputo y procesamiento. Este tipo de infraestructura ya puede considerarse también como nodos de la computación en la niebla. Por lo tanto, estos nodos pueden encontrarse geográficamente distribuidos para atender las solicitudes de los vehículos y peatones en los STI.

A continuación, se detalla de mejor manera los distintos tipos de comunicación entre los nodos:

- V2V (Vehicle to Vehicle - Vehículo a Vehículo): En este tipo de red, los vehículos envían y reciben mensajes de otros vehículos que se encuentren dentro de su rango de cobertura como se lo puede observar en la figura 14. Los mensajes que se intercambian tienen como objetivo: evitar colisiones, otorgar información sobre el estado de la vía, reportar un vehículo de emergencia en aproximación, entre otras.



Figura 14. Ejemplo de comunicación vehículo a vehículo.

Tomado de (iStock, 2016).

- V2I (Vehicle to Infrastructure – Vehículo a Infraestructura): Los vehículos se comunican con los nodos o infraestructura de telecomunicaciones que se encuentran desplegadas a lo largo de las vías y autopistas como se puede observar en la figura 15. Esto con el fin de evitar posibles atascos, conocer el estado de la vía, disponibilidad de parqueaderos e inclusive el reporte de un vehículo de emergencia para actuar sobre los semáforos.

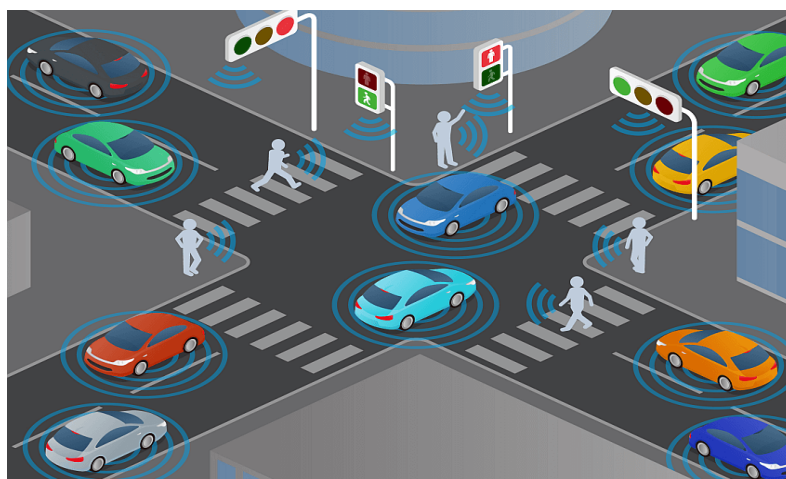


Figura 15. Ejemplo de comunicación vehículo a infraestructura entre vehículos y semáforos.

Tomado de (iStock, 2016).

- V2X (Vehicle to Everything – Vehículo a Cualquier Objeto): Los vehículos enlazan una comunicación a cualquier cosa u objeto móvil que se pueda involucrarse con los STI como, por ejemplo: ciclistas, peatones y buses.

3.3 Tecnologías inalámbricas para STI

Como se pudo evidenciar anteriormente, los STI basan su comunicación en tecnología inalámbrica y redes ad-hoc; especialmente en VANETs. Por lo que su arquitectura debe estar compuesta por redes inalámbricas que ofrezcan un tiempo de respuesta inmediato, ante las diferentes aplicaciones en STI. Un ejemplo de esto son las alertas o mensajes un vehículo puede generar para atender alguna emergencia en la vía. Además de poder verificar los sensores de un vehículo y comunicar mediante el OBU al cloud del fabricante del vehículo la orden de nuevas respuestas en la parte afectada.

Los STI tienen dos décadas de que hayan sido definidas y en el lapso de ese tiempo hasta la actualidad se han propuesto distintas tecnologías inalámbricas. Algunas se encuentran estandarizadas específicamente para STI y otras no; lo que conlleva a analizar entre estas, cual es la más idónea para su implementación.

3.3.1 Comunicaciones dedicadas de corto alcance

Las comunicaciones dedicadas de corto alcance o como se lo conoce por sus siglas en inglés DSRC, es un conjunto de estándares de los STI para la comunicación e intercambio de mensajes en una VANET.

Estados Unidos fue uno de los países pioneros en STI, ya que han desarrollado varios proyectos basados en esta tecnología. Por lo que en 1999 la Comisión Federal de Comunicaciones decidió asignar 75 Mhz de ancho de banda para los STI en la frecuencia de 5.850 – 5.925 Ghz. (Comisión Federal de Comunicaciones, 1999)

Consecuentemente en 2004 se forma un grupo de trabajo IEEE TFp (Task Force p) para estandarizar una arquitectura de comunicación para vehículos en STI. Esta arquitectura se basaría en redes inalámbricas de área local IEEE 802.11, que posteriormente fue definida como IEEE 802.11p. (Universidad Militar Nueva Granada & Llano Ramírez, 2014)

Wireless Access in Vehicular Environment (WAVE), es un término que describe el conjunto de normas y estándares de IEEE 1609 e IEEE 802.11p, que definen el acceso inalámbrico en redes ad hoc vehiculares.

IEEE 802.11p se encuentra definido en capa física y capa de enlace o MAC para el ambiente vehicular que desea implementar.

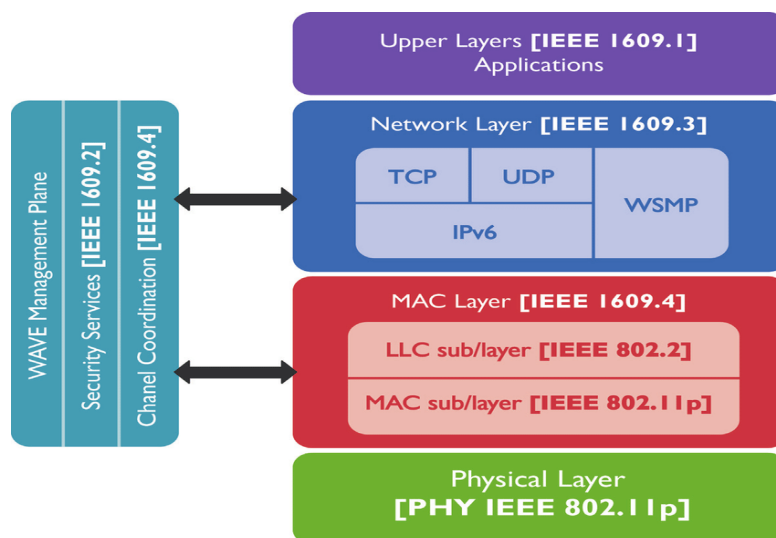


Figura 16. Arquitectura IEEE 802.11p e IEEE 1609 para WAVE.

Tomado de (Orozco, Michoud, & Llano, 2013)

En la capa física de la arquitectura WAVE, el estándar IEEE 802.11p tiene como técnica de multiplexación a OFDM (Acceso Múltiple por División de Frecuencia Ortogonal). Que de igual manera se utiliza en IEEE 802.11a para redes Wifi.

En IEEE 1609 se define como opera la capa de red y capa de transporte mediante los protocolos IP, TCP, UDP y WSMP. Lo que explica el modelo de

comunicación mediante varios mecanismos de acceso desde los OBU y RSU hacia el cloud.

Por otra parte, Europa desarrolla su propio estándar DSRC para STI en el que existen ciertas diferencias y otras similitudes en comparación con el estándar de IEEE. Las diferencias radican en la utilización de protocolos europeos que son comúnmente utilizados en la capa de enlace. Este estándar europeo es desarrollado por varias entidades como la ETSI TC ITS y el Comité de Estandarización Europeo.

Actualmente, a pesar del gran interés que se ha dado a esta tecnología. La Comisión Federal de Comunicaciones (FCC) mediante su grupo especializado en Wifi, ha propuesto que el espectro de 5.9 y 6 GHz sea compartido con la tecnología IEEE 802.11. Esta propuesta se justifica en que no existe una amplia utilización de DSRC en este espectro; concluyendo que ese ancho de banda asignado se encuentra subutilizado. (Alleven, 2018; Hetting, 2018).

3.3.2 Identificación por radiofrecuencia

La tecnología de identificación por radiofrecuencia o comúnmente más conocida por sus siglas en inglés como RFID (Radio Frequency Identification) es una tecnología inalámbrica que permite la identificación de objetos. Los componentes principales de los sistemas RFID son etiquetas o transpondedores que están incorporados en objetos para su comunicación con lectores RFID. (Bolić, Simplot-Ryl, & Stojmenović, 2010).

La implementación más simple en este sistema consiste en un lector estático que puede ser implementado en un RSU y una etiqueta RFID en un objeto móvil (vehículo) o fijo. Su aplicación es ampliamente utilizada en STI para identificar vehículos robados, pago de peajes automáticos y posicionamiento. (Saad, Zouka, & Al-Soufi, 2016).

Los componentes de un sistema RFID son los siguientes:

- Etiquetas: Está incorporado a cualquier objeto para su rastreo. La etiqueta está compuesta por un circuito integrado y este a su vez se compone de una antena para la comunicación. No necesita de una fuente de alimentación externa y estos pueden ser detectados en ciertos casos a una distancia de 10 metros sin línea de vista directa. La etiqueta o chip esta preprogramada con un identificador único el cual es asignado por su fabricante. Dicho identificador único esta almacenado dentro del circuito integrado y se lo conoce como EPC (Electronic Product Code) que se encuentra conformado por una cadena de caracteres de 96 bits.
- Antena: Se encarga de recolectar la energía y dirigirla hacia el circuito integrado para energizarlo. Generalmente, mientras más grande es la antena, más energía se podrá recolectar y canalizar hacia el circuito integrado de la etiqueta para obtener un mayor rango de cobertura.
- Lectores: Son dispositivos que se encuentran energizados generalmente mediante baterías y se comunican mediante ondas electromagnéticas hacia las etiquetas para obtener datos y el identificador único. Esto con el fin de procesarlo en un sistema operativo para la aplicación que se desee llevar a cabo.

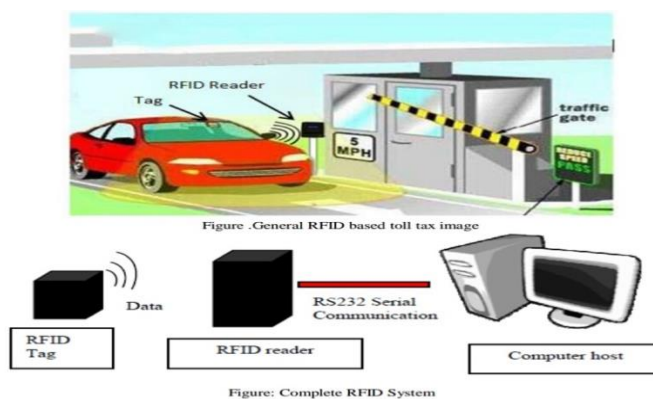


Figura 17. Ejemplo de funcionamiento del sistema RFID.

Tomado de (Orozco, Michoud, & Llano, 2013)

Los componentes descritos anteriormente se pueden apreciar en al figura 17 y además pueden operar mediante dos modos que se detallan a continuación:

- RFID pasivo: En este modo el lector envía una onda de radio hacia la etiqueta. Consecuentemente la etiqueta RFID utiliza la señal transmitida para energizarse y enviar de vuelta la información requerida. Este sistema opera en las bandas LF (low frequency), HF (high frequency) y UHF (ultra-high frequency).
- RFID activo: Este sistema se caracteriza en que sus elementos tengan su propio transmisor y fuente de alimentación. Los lectores son dispositivos RFID activos; pero muy pocas veces las etiquetas o transpondedores trabajan como RFID activos. Este modo generalmente opera en la banda UHF y ofrece un rango de cobertura de hasta 100 metros.

Tabla 4.

Diferencias entre RFID activo y RFID pasivo

	RFID activo	RFID pasivo
Fuente de energía	Tiene una fuente de energía interna.	Su fuente de energía es mediante el lector a través de radiofrecuencia.
Uso de batería	Si	No
Disponibilidad	Disponibilidad continúa debido a que se encuentra energizado.	Solamente cuando está en el rango del lector y pueda energizarse.
Energía requerida	Alta debido a que necesita energizar a la etiqueta RFID pasiva.	Baja ya que solamente utiliza energía para transmitir datos y alcance es corto
Rango de comunicación	Largo alcance de 100 metros o más.	Corto alcance de hasta 10 metros

Tomado de (IMPINJ, 2017).

Los sistemas RFID pueden ser también clasificados mediante la frecuencia en la que operan. Los sistemas RFID pueden transmitir en frecuencias que van desde los 30 KHz hasta 300 KHz considerada como banda LF. Generalmente en esta banda operan a 125 KHz y 134 KHz. Por otra parte, también tienen un ancho de banda que va desde los 3 hasta los 30 Mhz en la banda HF y por último en el rango de los 300 MHz hasta los 3 GHz. (IMPINJ, 2017).

3.3.3 Tecnologías 802.15

El 802.15 es un conjunto de grupos especializados en ciertas áreas de la IEEE, que inicialmente fue formado para enfocarse en dispositivos wearables y redes de área personal (PAN). Su trabajo ha sido ampliamente aceptado por lo que actualmente concentran su investigación en protocolos con altas tasas de transmisión de datos y rangos de cobertura que van desde los metros hasta los kilómetros.

A continuación, se profundiza en dos estándares que están siendo más utilizados en STI que son Bluetooth y 802.15.7 correspondiente a la tecnología de comunicación con luz visible

3.3.3.1 Bluetooth

Bluetooth es una tecnología inalámbrica de bajo consumo de energía que es utilizada masivamente en dispositivos que van desde celulares, teclados y mandos de videojuegos. Bluetooth fue creado en 1994 en Ericsson para contrarrestar la letanía de utilizar cables para cada periférico hacia las computadoras mediante la utilización de radiofrecuencia. (Sairam, Gunasekaran, & Redd, 2002)

Consecuentemente, se conforma el grupo de interés especial por Bluetooth (SIG), el cual tuvo miembros como Nokia, Intel, Toshiba, IBM entre otros. En el año 1998 se lanza la versión 1.0 de Bluetooth, posteriormente se ratifica y actualiza este estándar en diferentes versiones. Por lo que, se llega a proponer

a Bluetooth de bajo consumo de energía o como más se lo conoce por sus siglas en inglés BLE (Bluetooth Low Energy).

BLE introdujo un nuevo segmento de mercado en los dispositivos; puesto que permite a estos comunicarse y energizarse con una simple batería de moneda. Y actualmente es utilizado extensivamente en IoT como beacons, sensores inalámbricos, control remoto, monitoreo de salud, entre otras. (Jeon, She, Soonsawad, & Ng, 2018)

Los nodos en Bluetooth pueden tener varios modos:

- Modo anunciante: El dispositivo realiza un broadcast hacia los nodos que se encuentren en modo de escaneo. Esto con el fin de alertarlos o anunciarlos de su presencia hacia el dispositivo que se desee emparejar o solamente enviar un mensaje.
- Modo escaneo: Los dispositivos que reciben paquetes del anunciante sin la intención de establecer un enlace punto a punto.
- Iniciador: Dispositivos que tienen la intención de establecer un enlace o conexión.

El momento en que un nodo anunciante decide emparejarse con un dispositivo de escaneo; este último se convierte en maestro y el anunciante en esclavo. Este tipo de conexión se la conoce como piconet en Bluetooth. (Lea, 2018)

De igual manera los dispositivos en modo escaneo pueden manejarse de dos maneras:

- Escaneo activo: El enlace se encuentra escuchando protocolos de unidad de datos (PDU). Y dependiendo del PDU recibido el dispositivo de escaneo puede solicitar información adicional al nodo anunciante.
- Escaneo pasivo: El dispositivo de escaneo solamente recibe paquetes y la transmisión se encuentra desactivada.

Los dispositivos Bluetooth operan entre los 2.4000 hasta 2.4835 GHz en la banda no licenciada industrial, científica y médica (ISM). Por lo tanto, este ancho de banda en particular se encuentra congestionado con dispositivos inalámbricos que de igual manera operan en esta frecuencia como también lo hace 802.11 Wifi.

Para manejar y contrarrestar este tipo de interferencia; Bluetooth soporta espectro ensanchado por salto de frecuencia o como se lo conoce por sus siglas en inglés FHSS (Frequency-hopping spread spectrum). (Lea, 2018)

Una de las más importantes características de Bluetooth y del porque también es ampliamente utilizado, es debido al beaconing. El Bluetooth beaconing es una característica secundaria de BLE; sin embargo, es una tecnología muy utilizada en IoT y empieza a ser de gran importancia en STI.

Los beacons nunca se emparejan con un nodo en específico. Utilizan los mensajes de anuncio para poder transmitir un broadcast con su UUID y un mensaje; adicionalmente pueden transmitir su indicador de fuerza de la señal recibida (RSSI). Por tanto, el beacon lector puede tomar los datos cuanto más esté aproximado; lo que podría suponer una comunicación V2I entre un vehículo (anunciante) y una infraestructura (beacon de escaneo).

3.3.3.2 Objetivos específicos

Comunicación por luz visible o comúnmente más conocida por su acrónimo en inglés VLC (Visible Light Communication), se refiere a un tipo de comunicación óptica inalámbrica de corto alcance empleando luz visible. VLC transmite datos mediante la utilización del diodo emisor de luz (LED) y diodo láser entre los 380 y 780 nm del espectro electromagnético. En la figura 18 se puede observar los distintos colores según la longitud de onda en el espectro visible. (Rajagopal, Roberts, & Lim, 2012).

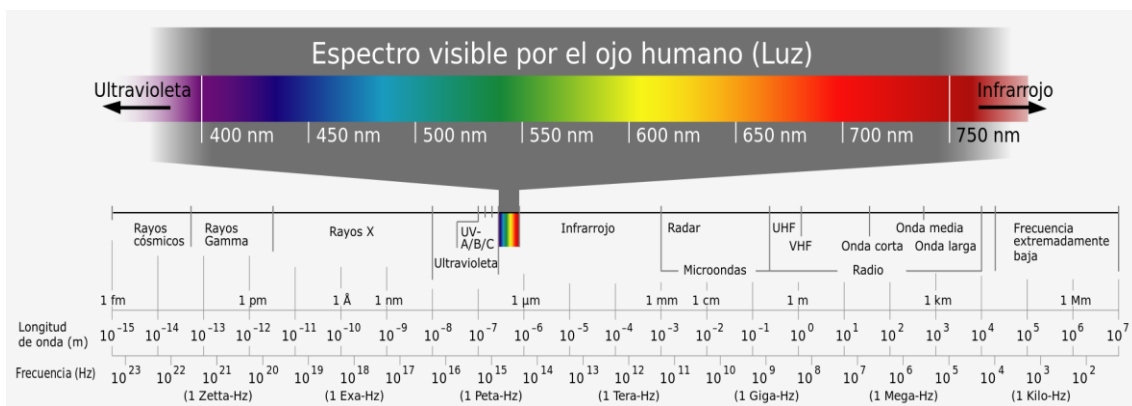


Figura 18. Rango de luz visible entre los 400 y 700 nm en el espectro electromagnético.

Tomado de (Pastor, 2018)

Como se puede evidenciar en la actualidad, la comunicación mediante radio frecuencia debajo de los 6 GHz está siendo saturada por las distintas tecnologías inalámbricas que están siendo desplegadas. VLC nace como una alternativa inalámbrica que puede ofrecer aproximadamente 300 THz de ancho de banda disponible con velocidades de transmisión en el orden de los Gbps en cortas distancias.

A esta tecnología se la conoce comercialmente como Lifi (Light fidelity) o también Optical WLAN. La estandarización IEEE 802.15.7 ofrece tres niveles de capa física (PHY). PHY I opera en transmisiones de velocidad desde los 11.67 hasta los 266.6 kb/s, PHY II entre 1.25 hasta 96 Mb/s y por último PHY III 12 y 96 Mb/s respectivamente. En PHY I y II se utiliza una única fuente de luz con modulación on-off (OOK). Mientras que PHY III utiliza múltiples fuentes ópticas con diferentes frecuencias; es decir distintos colores y utiliza un tipo de modulación denominada en inglés como color shift keying (CSK). (Rajagopal et al., 2012)

La tecnología LED es altamente confiable, energéticamente eficiente y su vida útil supera a las fuentes de luz incandescentes. Por lo que, estas características únicas han promovido a que los fabricantes de automóviles consideren utilizar este tipo de fuente de luz antes que las halógenas. (Long et al., 2015)

También se ha promovido la implementación de semáforos y señales vehiculares, basadas en LED en las distintas urbes alrededor del mundo. El estándar para el tamaño de estas va desde los 100 a 200mm. (Comité Europeo de Normalización, 2015)

Las miras de VLC en STI son prometedoras ya que las infraestructuras o RSU, podrían ser semáforos y postes de alumbrado público que fácilmente podrían transmitir datos mediante la luz visible que emiten. Además, también pueden cumplir con las comunicaciones V2V, debido a que mediante los faros de los vehículos también transmitir datos mediante VLC con tecnología LED hacia otros nodos. (Mare, Luiz Marte, & Cugnasca, 2016).

3.4 Urbes y semaforización con STI

El impacto y consecuencias del tráfico como la contaminación, entorpecimiento para la atención de emergencias, tiempo improductivo en atascos, entre otros, ha originado que los gobiernos y municipios de las distintas urbes del mundo analicen la implementación de las tecnologías de la información en las infraestructuras viales.

Los STI se han convertido en una de las aplicaciones más importantes del IoT; debido a que es una de las tecnologías que permite alcanzar las ciudades inteligentes. A pesar de que los STI basan sus comunicaciones en V2V y V2I; actualmente solo existen proyectos pilotos para que su implementación esté lista después de 2020. No obstante, los proyectos actuales de STI constan de automatización solamente en los RSU, sin existir una conexión directa con los vehículos. Sin embargo, estos últimos cuentan con una gran cantidad de sensores a bordo como GPS, cámaras, LIDAR etc. Pero recientemente se ha empezado a equipar con OBU y tecnología inalámbrica a vehículos de distintas marcas como Cadillac, Toyota, Mercedes y Audi. (Conner, 2018; United States Department of Transportation, 2017).

3.4.1 Quito

Quito es la capital de la República del Ecuador la cual se ubica en las montañas de los Andes a 2800 m.s.n.m. con una demanda vehicular de 60.000 vehículos diarios. Actualmente esta urbe cuenta con un sistema adaptativo de semáforos que fue suscrito por la alcaldía del Distrito Metropolitano de Quito con la empresa española TELVENT. Dicho sistema se lo denomina Optimus. (Martinez, 2017).

3.4.1.1 Gestión de tráfico Optimus

La adjudicación del proyecto se la realizó en el mes de noviembre del año 2011. Este sistema de STI cuenta con dispositivos que censan y miden el tráfico a través de cámaras y contadores de presión.

Este sistema fue implementado por TELVENT, una empresa española dedicada a las tecnologías de información en las áreas de transporte, energía y administración pública. Este sistema también cuenta con paneles de mensajes variables que permiten mantener informados a los conductores acerca de cierres, obras o accidentes en las vías.

3.4.1.1.1 Características

La topología de red de este sistema es centralizada y ofrece ciertas características:

- Las cámaras de video detectan el número de autos que circulan por la infraestructura vial. El sistema también obtiene datos mediante sensores de presión como se puede observar en la figura 19.
- Realiza el cálculo lo más pronto posible en el nodo central de procesamiento. Por tanto, se mejora la movilidad en los ejes longitudinales y transversales.
- La prioridad no solo son los vehículos, sino que el sistema también puede censar los cruces peatonales en las vías de donde se encuentran las paradas del transporte público.

- Además de medir el tráfico, las cámaras tienen la función de captar la infracción a vehículos que crucen las intersecciones en rojo o invadan las vías exclusivas del transporte público.



Figura 19. Sensor de presión y cámara de video para contabilizar los vehículos que circulan por la vía.

Adaptado de (Metro Ecuador, 2016)

3.4.1.1.2 Funcionamiento e implementación

La implementación de este sistema adaptativo de semaforización se lo realizó en dos fases:

- La fase 1 comenzó con la instalación de los primeros semáforos adaptativos a lo largo del corredor de la Av. Mariscal Sucre. Las cámaras que fueron instaladas se ubican entre 75 y 125 metros antes de cada intersección.

- Luego se realizó la instalación de estas en el sur y centro norte de la ciudad. Contabilizando en total 1511 cámaras distribuidas en 660 intersecciones bajo esta modalidad.

Las 1511 cámaras y los distintos sensores o contadores de presión que se encuentran implementados están interconectados a un sistema de control centralizado. El medio físico por el que se comunican los datos a dicho nodo central es mediante fibra óptica. Por tanto, la infraestructura que censa el nivel de tráfico se traslada mediante un anillo de red metropolitana de fibra óptica que interconecta todos los nodos con el Centro de Gestión y Movilidad. El cual contiene software de STI para que automáticamente regule los semáforos en las intersecciones más congestionadas. (Martinez, 2017)

Además, contienen personal el cual también puede modificar el tiempo de semaforización en dichas intersecciones y protocolos establecidos para dar prioridad cuando circulen vehículos de emergencia.

3.4.1.1.3 Ventajas

- La cantidad de nodos implementados a lo largo de la ciudad permiten una agregación de datos más confiable para ser procesada.
- Permite la asistencia y toma de decisiones para vehículos de emergencia.
- La posibilidad de modificar manualmente intersecciones que no cuentan con infraestructura para medir el tráfico. Se lo realiza mediante operadores que se encuentran monitoreando las 24 horas desde el Centro de Gestión de Movilidad.

3.4.1.1.4 Desventajas

- En el sistema no hay la posibilidad de realizar comunicaciones de V2V y V2I debido a que no cuenta con tecnología inalámbrica.
- Cuellos de botella ya que todos los datos viajan a un nodo centralizado y ante el fallo de este todo el sistema se vería afectado.

- El tiempo de respuesta se ve afectado ya que el procesamiento no se realiza lo más cercano a la infraestructura que censa el tráfico.

3.4.2 Pittsburgh

Pittsburgh es una ciudad de los Estados Unidos de América, ubicada en el estado de Pensilvania. Últimamente esta urbe se ha destacado por tener uno de los STI más avanzados e innovadores de su país. Por lo que inclusive ha atraído la atención de Uber; una de las compañías más grandes del mundo. En efecto, Uber escogió a Pittsburgh como una de las ciudades piloto para la prueba de sus vehículos autónomos. (Kang, 2017).

3.4.2.1 SURTRAC

El sistema STI que se encuentra implementado en la ciudad de Pittsburgh, se lo denomina por su acrónimo en inglés SURTRAC (Scalable URban TRAffic Control). Fue desarrollado por investigadores del Instituto Robótico, el cual es una división de la Universidad Carnegie Mellon. Recientemente también fue implementado en Atlanta GA y Portland ME.

SURTRAC es una iniciativa que permite el control de las señales de tránsito y semáforos en tiempo real, mediante la utilización de inteligencia artificial y teoría de tráfico. Esto con el objetivo de optimizar el tiempo y rendimiento de los semáforos en la infraestructura vial, la mejora del flujo de carros en el carril más congestionado y viajes más cortos. (Rapid Flow, 2016)

3.4.2.1.1 Características

Cabe destacar que la toma de decisiones en el sistema SURTRAC es de manera descentralizada. En otras palabras, el control individual se lo realiza en cada una de las intersecciones; lo que permite un mejor tiempo de respuesta a las condiciones locales de tráfico. Además, la descentralización facilita la escalabilidad al permitir la adición de nuevos nodos en las intersecciones.

También este STI se caracteriza por su respuesta en tiempo real a las condiciones cambiantes del tráfico según las horas de circulación. Esto lo realiza mediante su modelo denominado “*second by second*” que posee de inteligencia artificial; con el objetivo de elaborar un plan de automatización y así brindar una mejor circulación tanto para los vehículos, bicicletas, buses y transeúntes.

La gestión de las vía o carreteras urbanas la realiza en forma de cuadrícula; es decir que gestiona un número de nodos dentro de esta área para intercambiar información entre otros nodos.

Resumidamente SURTRAC comprende de estas características específicas en el control de semaforización:

- Respuesta en tiempo real debido a que SURTRAC analiza cada segundo las condiciones de tráfico.
- SURTRAC es descentralizado y escalable; independientemente del tipo de red (estrella, malla, etc.) y tamaño que se implemente.
- Optimizado para ambientes urbanos y suburbanos; es decir que es capaz de controlar autopistas, calles urbanas en varias intersecciones y flujos grandes de vehículos.
- Es multi-operacional debido a puede conectarse a vehículos, peatones, buses y ciclistas mediante la integración de varios tipos de sensores y según la prioridad que se presente.
- La integración con vehículos que poseen conectividad; en otras palabras, mediante la utilización de DSRC, tecnología mediante la cual se puede intercambiar datos entre RSU y vehículos.
- Tiene la funcionalidad de poder interconectarse con otros sistemas como una central de gestión de tráfico, dashboards o smartphones.

3.4.2.1.2 Funcionamiento

SURTRAC considera a esta optimización de tráfico como un conjunto de nodos que trabajan como uno solo. Esto significa que cada intersección agrega datos y

al mismo tiempo se comunica hacia intersecciones vecinas para la coordinación y optimización del tráfico. En base a la información que todas las intersecciones recogen, se da prioridad al flujo de vehículos que más congestionado se encuentre en tiempo real. En la figura 20 se puede observar todo este procedimiento descrito con mayor detalle.

Esto permite trabajar a SURTRAC en todo tipo de entornos ya sean suburbanos o urbanos y en ambientes completamente caóticos gracias al software de inteligencia artificial que incorpora.

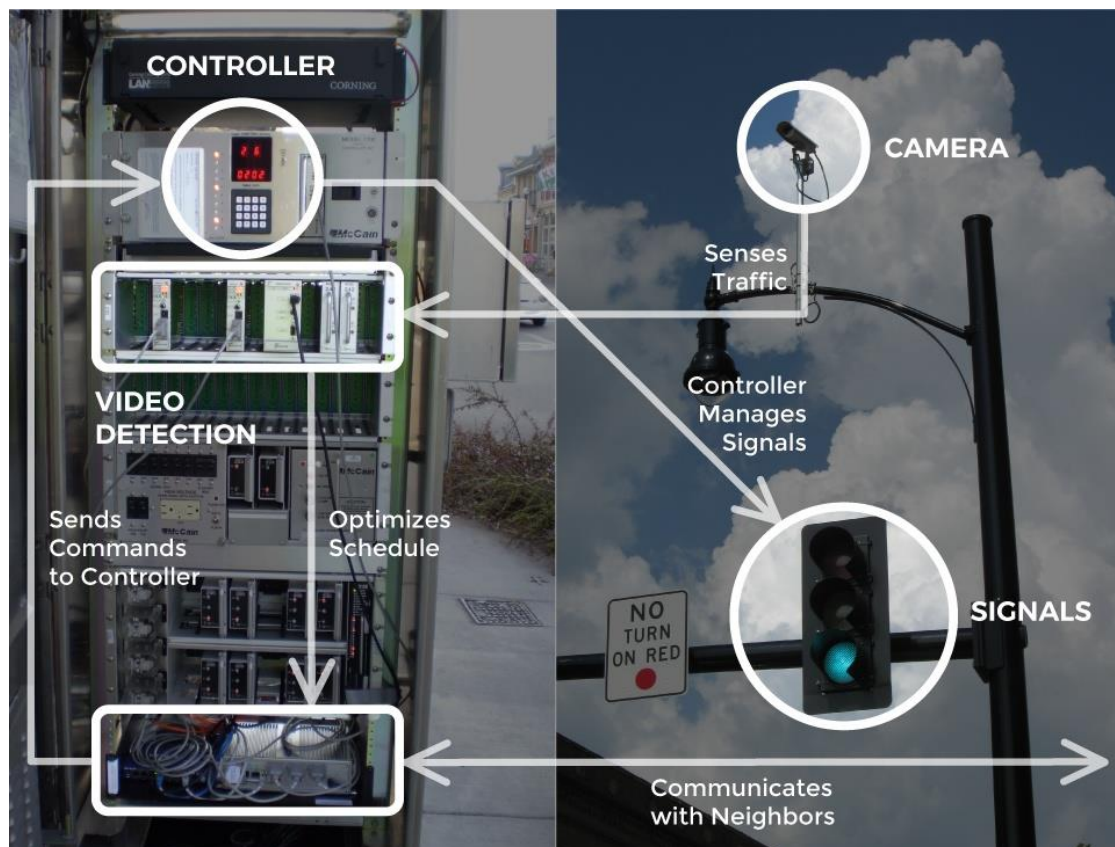


Figura 20. Funcionamiento de SURTRAC tomando como ejemplo a una cámara que mide el tráfico en una intersección y que posteriormente la información es enviada al software para un plan de optimización de tráfico en el que finalmente se envían comandos al semáforo.

Tomado de (Rapid Flow, 2016)

El procedimiento que realiza este sistema es el siguiente:

- 1) Primero agrega todos los datos en tiempo real de lo que sucede en cada intersección. La información es recopilada mediante un software que se integra con los sensores e infraestructura mediante una API. Estos sensores pueden ser cámaras, radares e incluso detectores de bucles de inducción.
- 2) Luego procesa la información mediante su software de eventos, en el que crea un plan de optimización del movimiento existente en la intersección para que la misma fluya de la forma más eficiente.
- 3) A continuación, y una vez obtenido los patrones de tráfico existentes. Mediante el software y su integración con los sensores mediante la API, se envía el plan de optimización al controlador para la coordinación de la señalización del semáforo.
- 4) Finalmente, SURTRAC envía la información de su intersección y como va fluyendo el tráfico en ese punto por las siguientes razones:
 - a. Con la finalidad de que los nodos incorporen datos e información relevante de sus intersecciones vecinas, para que la toma de decisiones entre estas sea autónoma y coordinada.
 - b. La posibilidad de informar a vehículos conectados, pasajeros, peatones o cualquier otro nodo que pueda utilizar dicha información de manera útil en los STI.

3.4.2.1.3 Implementación

Lo que SURTRAC necesita para su implementación es un software que gestione la conexión entre la infraestructura de detección (generalmente cámaras de video, radares o bucles de inducción) y el gabinete donde se encuentra el procesamiento y controlador. El software también debe gestionar la conexión con las intersecciones aledañas mediante enlaces inalámbricos.

Usualmente SURTRAC se maneja en una pequeña computadora contenida en el gabinete descrito anteriormente; pero también puede ser implementado en hardware ya existente dentro de estos.

SURTRAC está diseñado para ser instalado fácilmente. Primero se comprueba que la infraestructura para detectar el tráfico y gabinete se encuentren correctamente instalados y operativos. Luego se instala el procesador de SURTRAC en cada intersección (puede ser una computadora o equipo ya existente); a continuación, se realizan las pruebas de todos los componentes dentro del gabinete y fuera de él.

Las primeras semanas el sistema se encuentra en la recolección de datos y el manejo de este desde una central para que tenga un patrón base de la afluencia vehicular en cada intersección. Finalmente, y una vez que el sistema de STI se encuentre estable, se da la libertad al sistema para que opere y pongan en marcha los planes de optimización de tráfico mediante el procesamiento de su software que incorpora inteligencia artificial.

Como se puede observar en la figura 21 los componentes principales como lo son la comunicación tanto en el emplazamiento de la intersección como el gabinete. La detección mediante cámaras de video las cuales están directamente conectadas hacia el gabinete, procesamiento mediante un software con inteligencia artificial y el controlador el cual es el responsable de enviar cualquier acción hacia la intersección.

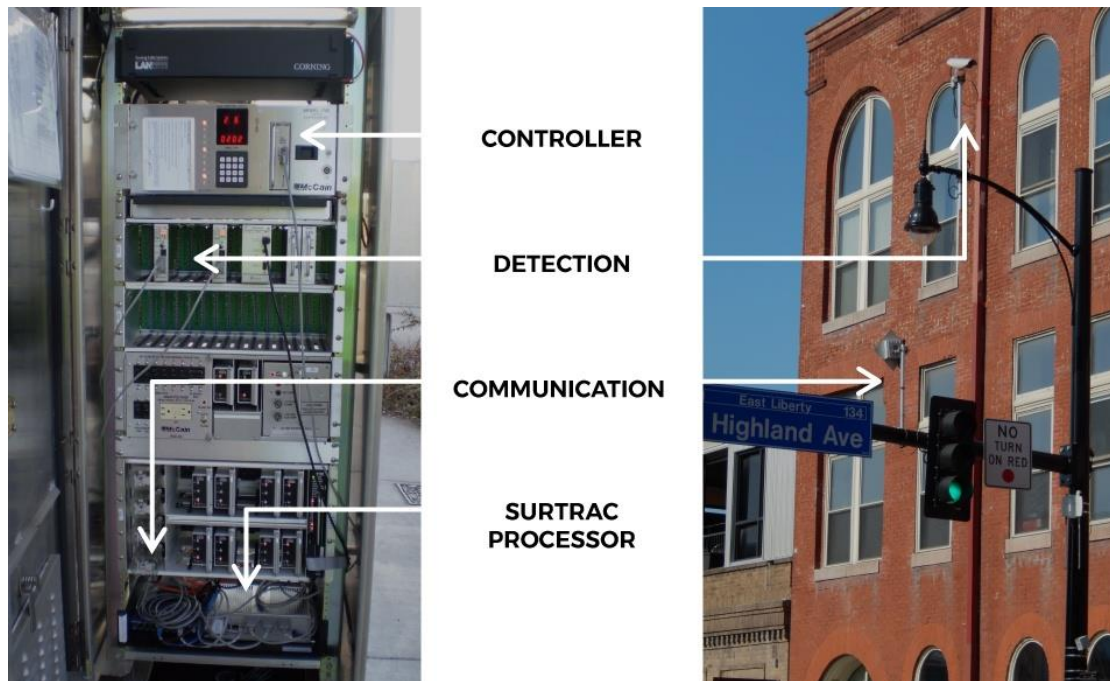


Figura 21. Los componentes que permiten la implementación del sistema SURTRAC. Consta de un controlador, detección mediante una cámara de video, comunicación mediante antenas y una computadora en donde se ejecuta el software.

Tomado de (Rapid Flow, 2016)

3.4.2.1.4 Ventajas

- Gracias a que el sistema es descentralizado se reduce la posibilidad de un cuello de botella y evita un solo punto de falla en el sistema.
- Permite conexiones V2I mediante la tecnología inalámbrica DSRC; además de la integración de peatones, ciclistas y buses.
- Una vez desplegado este sistema, los resultados arrojaron un 25% menos de tiempo en atascos y 81 horas más al año para los conductores de Pittsburgh. (Snow, 2017)
- Cumple con uno de los pilares fundamentales de la computación en la niebla como la descentralización y comunicación entre nodos vecinos para brindar respuestas en tiempo real.

3.4.2.1.5 Desventajas

- A pesar de que tiene conectividad V2I mediante DSRC, no direccionan este sistema para vehículos de emergencia.
- Los RSU solamente están en intersecciones y al tener una cámara como infraestructura principal para censar, la confiabilidad de los datos pueden verse afectados al existir lluvia, neblina u otras condiciones climáticas adversas.

3.4.3 Nueva York

La ciudad de Nueva York, capital del Estado de Nueva York ubicada en la costa este de los Estados Unidos. Esta metrópoli se encuentra entre una de las ciudades más aglomeradas del mundo con más de 22 millones de habitantes. Además de estar en el tercer puesto de las ciudades en las que más tiempo se pierde en tráfico vehicular, con un promedio de 91 horas al año. (INRIX, 2017)

Actualmente debido a las alarmantes cifras de tráfico y accidentes que ocurren en Estados Unidos. El organismo estatal el cual es el departamento de transporte (USDOT), ha escogido a tres ciudades piloto para la implementación de un novedoso sistema denominado tecnología CV (Connected Vehicles) que será desarrollado en Nueva York, Wyoming y Tampa.

3.4.3.1 NYC DOT Pilot

El proyecto piloto del departamento de transporte de la ciudad de Nueva York, o como lo conoce de manera abreviada NYC DOT Pilot. Tiene como objetivo reducir la congestión y los accidentes de tránsito en dicha urbe. Además, que está enfocada principalmente en aplicaciones de seguridad para conductores y peatones.

La tecnología que se implementa está completamente basada en conexiones V2V (vehículo a vehículo), V2I (vehículo a infraestructura) e inclusive IVP (infraestructura a peatón). Este sistema provee alertas de emergencia a los

conductores para que el mismo pueda tomar las acciones ante una posible colisión contra otro vehículo o infraestructura vial. (New York City DOT, 2018).

3.4.3.1.1 Características

Debido a que el proyecto abarca numerosas aplicaciones en cuanto a la seguridad, solo se detallarán las más importantes.

- V2I

Se notificará al conductor y al RSU de un cruce en rojo en la intersección para generar una infracción mediante el OBU instalado en el vehículo.

Alertas al vehículo que se no cumpla con los límites de velocidad asignados en la vía. La alerta puede ser mediante mensajes o inclusive audios de voz que serán reproducidos dentro del vehículo.

Alertas de sobrepeso en el vehículo, enfocado principalmente a camiones de carga que debido al peso y velocidad pueden causar accidentes de tránsito.

Avisos de emergencia y evacuación ante algún desastre o accidente.

- V2V

Notificación de una posible colisión con otro vehículo por detrás en distintas situaciones. Cuando un vehículo se encuentra detenido, cuando el vehículo de adelante está frenando y por último si el vehículo se encuentra en movimiento.

Prevé al conductor cuando otro vehículo se aproxima en el carril adyacente y se encuentran en el punto ciego de los retrovisores para evitar un choque.

- IVP

Se notifica al RSU implementado en los semáforos o intersecciones cuando los transeúntes estén cruzando un paso cebra. Consecuentemente el RSU notifica a los vehículos que se aproximen con precaución. Esta aplicación también se enfoca en personas no videntes.

3.4.3.1.2 Funcionamiento e implementación

El proyecto se encuentra dividido en tres fases. La primera fase constó en la planificación del proyecto que tuvo duración de un año y dando por terminado en septiembre del 2016. Luego la segunda fase tuvo una duración de 20 meses en la que se implementó 300 RSUs en Manhattan, Brooklyn y en puntos estratégicos como paradas de autobuses, puentes y aeropuertos. Por último, la tercera fase se encuentra en desarrollo, la cual empezó en julio de 2018 con miras a finalizar en febrero del 2020. En esta última fase se realizan pruebas del funcionamiento del proyecto ya implementado; por consiguiente, ya se envían alertas a los conductores. (New York City DOT, 2018)



Figura 22. Elementos que componen el NYC DOT Pilot en la ciudad de Nueva York.

Tomado de (New York City DOT, 2018)

Uno de los componentes más importante son los RSU ya que serán los responsables de comunicarse y detectar a vehículos, peatones y dispositivos

móviles. Al OBU en este proyecto se lo denomina ASD (Aftermark Safety Device), el cual se encuentra incorporado en vehículos livianos, buses y camiones. Puede transmitir y recibir señales de comunicación mediante DSRC en la frecuencia de 5.9 GHz.

El controlador de tráfico se encuentra específicamente instalado en las intersecciones y cumple con la función de administrar y comunicarse con las RSUs aledañas mediante DSRC. Tiene la capacidad de realizar cierto procesamiento; pero principalmente este nodo se encarga de enviar toda la información recolectada de los RSUs a los operadores del Centro de Gestión de tráfico de Nueva York. La información enviada a este nodo central la realiza mediante la red inalámbrica de Nueva York. La cual es una infraestructura dedicada de banda ancha que interconecta centros de operaciones, atención de emergencias entre agencias de servicios y provee de conectividad y movilidad a usuarios.

3.4.3.1.3 Ventajas

- Implementación con base en comunicaciones V2V, V2I e IVP. Lo que promueve un ambiente IoT; justamente en el que interconecta RSUs con vehículos e inclusive peatones.
- Una aplicación que está más enfocada en brindar seguridad en las vías antes que priorizar el tráfico.
- Cuenta con varios RSUs por intersección; es decir que existirá conectividad a lo largo de la vía y no solamente en los semáforos.
- Prioriza a peatones e inclusive a transeúntes no videntes cuando crucen en intersecciones.

3.4.3.1.4 Desventajas

- No contempla un OBU para la conectividad V2I en los vehículos de emergencia.

- Si bien se puede realizar cierto procesamiento en los controladores de tráfico. El mayor procesamiento se lo realiza en el centro de gestión de tráfico lo cual puede recurrir en retardos en tomar alguna acción.

3.5 Comparación de STI entre urbes

Tabla 5.

Cuadro comparativo entre Pittsburgh, Quito, Nueva York

	Pittsburgh	Quito	Nueva York
Sensores	Cámaras de video, radares y lazos de inducción	Cámaras de video y sensores de presión	Utiliza los sensores de los vehículos.
Infraestructura	Su infraestructura se basa principalmente en RSU instalados en las intersecciones y en ciertos casos vehículos conectados mediante OBUs.	No Consta de RSU ya que no soporta conectividad inalámbrica y solamente contiene cámaras de video en las intersecciones.	Contiene una gran cantidad de RSUs a lo largo de las vías como semáforos, postes o emplazamientos ya que recolecta la información de los vehículos.
Tipo de comunicación	Entre RSUs realizando un procesamiento local en las intersecciones. También es posible la comunicación V2I.	Las 1511 cámaras de video se conectan a un nodo central para el procesamiento de la información agregada.	Su comunicación se basa principalmente en V2V, V2I e IVP.

Tecnología inalámbrica	DSRC en la frecuencia de 5.9 GHz.	No posee DSRC. Se internecta F.O para CCTV.	DSRC en la frecuencia de 5.9 GHz.
Tiempo de respuesta	Bajo tiempo de respuesta ya que el procesamiento es local.	Alto tiempo de respuesta ya que se envía la información a una nube privada para que sea procesada.	Bajo tiempo de respuesta ya que los RSUs tienen procesamiento local y además se apoya de una nube privada.
Asistencia a vehículos de emergencia	No	Si	No

La tabla 5 nos permite determinar de que las aplicaciones críticas en los STI como lo son el descongestionamiento de vías en tiempo real o habilitación de semáforos para vehículos de emergencia, no cuentan aún con una infraestructura de computación en la niebla. El escenario o implementación en la ciudad de Pittsburgh es el que mas se acerca a una implementación con estas características. Sin embargo a esta se la puede catalogar como un escenario de *edge computing*, el cual comparte el mismo principio que la computación en la niebla pero con un cómputo dedicado y con altas prestaciones como el de la computación en la nube. Por otra parte es notable que cada vez es mas importante y decisivo el uso de tecnologías inalámbricas para ciudades inteligentes; debido a la facilidad de interconectar los nodos que pueden estar alejados varios kilómetros. Es por eso que las implementaciones de Pittsburgh y Nueva York basan su conectividad principalmente a traves de ondas electromagnéticas y a la ciudad de Quito le ha resultado costosa su implementación debido a la interconexión mediante un medio físico como fibra óptica.

4 Metodología e implementación

4.1 Estudios previos

Los STI se han convertido en una de las aplicaciones más importantes de IoT para el desarrollo de las ciudades inteligentes. Por lo que se han realizado varios estudios para el mejoramiento del tráfico vehicular y la optimización del tiempo para los vehículos de emergencia.

Una de estas técnicas se la denomina Green Wave en el que un vehículo de emergencia puede habilitar los semáforos en luz verde mientras el mismo se encuentra cruzando la intersección. Este estudio se basa en una implementación mediante tecnología RFID, en la cual los RSUs disponen de lectores para receptar la señal de las etiquetas instaladas en los vehículos. La desventaja de este sistema es que al tener una etiqueta RFID pasiva incapaz de energizarse por sí sola, recurre a una posible mala utilización del sistema debido a que, si el vehículo no se encuentra en una emergencia los semáforos de igual manera serán activados. (Saad et al., 2016)

También se ha propuesto la detección de vehículos de emergencia mediante los sistemas CCTV que se encuentran instalados en distintas urbes con la técnica de procesamiento de imágenes. La desventaja de este sistema conlleva que, ante situaciones climáticas adversas como lluvia o neblina, podría no haber una respuesta apropiada o precisa para los vehículos de emergencia. (Mittal & Bhandari, 2013)

4.2 Metodología

Para el desarrollo del prototipo se utiliza el método inductivo experimental ya que, en base al levantamiento y análisis de la información recabada en previos capítulos, se escogerá las herramientas más aptas para su implementación. Dicho prototipo consiste en una infraestructura mediante la computación en la niebla para mejorar los tiempos de respuesta en ambientes IoT enfocado principalmente a vehículos de emergencia.

4.3 Selección de tecnologías

La implementación de un prototipo IoT conlleva a que se utilice una gran variedad de tecnologías; debido a que este gran sistema se lo tiene que tener presente desde protocolos de capa física hasta capa de aplicación del modelo OSI. Además de que actualmente la tendencia para la selección de tecnología tanto en protocolos y dispositivos IoT, se orienta a que promuevan un bajo consumo de energía y un mayor tiempo de respuesta.

4.3.1 Comunicación vehículo a infraestructura

Para la comunicación entre un vehículo y el RSU se toma a consideración un tipo de comunicación inalámbrica que no requiera emparejamiento entre dispositivos como, por ejemplo: maestro – esclavo. Además de que el dispositivo que requiera ser instalado en el vehículo pueda emitir pulsos electromagnéticos con información de su RSSI y mensajes de alerta.

Tanto Bluetooth, DSRC y VLC pueden emitir mensajes de alerta sin requerir el emparejamiento con un dispositivo maestro. VLC sería la tecnología óptima teniendo en cuenta que tiene un ancho de banda de 300 THz en comparación con los 83 Mhz de Bluetooth y 75 Mhz de DSRC respectivamente; sin embargo, VLC se encuentra aún en desarrollo para su implementación en vehículos. (Shen & Tsai, 2017)

A pesar de que DSRC y Bluetooth fueron estandarizadas hace más de una década, esta última ha sido la que ha logrado significativos avances. Estas características diferenciadoras son el bajo consumo de energía que es una característica primordial en IoT y además de ser la tecnología inalámbrica pionera en manejar beacons para el broadcast de mensajes sin emparejamiento.

Como principales características de Bluetooth 4.0 es que su rango o alcance es de hasta 100 m con un consumo de 100 mW, su velocidad de transmisión se encuentra entre 25 Mbps y 32 Mbps. Tiene la funcionalidad de BLE (Bluetooth

Low Energy), el cual es un modo de bajo consumo de energía y que es esencialmente factible para aplicaciones IoT.

En su modo de bajo consumo de energía utiliza como multiplexación al acceso múltiple por división de frecuencia (FDMA) y acceso múltiple por división de tiempo (TDMA) con 40 canales de transmisión y recepción. Cada canal se encuentra separado por 2 Mhz en los que tres son utilizados exclusivamente como anunciante y los 37 restantes para datos y de igual manera mensajes anunciantes.

Como se puede observar en la figura 23 los canales de Bluetooth son elegidos de forma aleatoria y mediante el espectro ensanchado por frecuencia (FHSS) ya que mediante esta técnica también se reduce la interferencia entre canales.

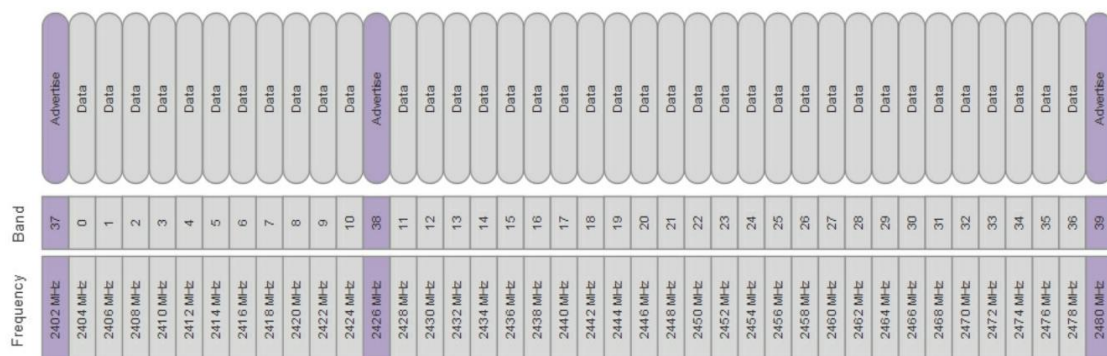


Figura 23. Los 40 canales seccionados a un ancho de banda de 78 Mhz correspondiente a la tecnología Bluetooth.

Tomado de (Lea, 2018)

4.3.2 Comunicación entre nodos de la computación en la niebla

La comunicación entre nodos de la computación en la niebla o también I2I también se realiza mediante tecnología inalámbrica ya que al ser RSUs que se encuentran ubicados en distintas intersecciones de una urbe, resultaría muy costosa su interconexión alámbrica

La cantidad de nodos de la computación en la niebla dependerá del rango de cobertura y área de la ciudad a cubrir. Por lo que para grandes urbes se requerirían cientos o miles de nodos de la computación en la niebla para satisfacer distintos servicios (medición de calidad de aire, contaminación auditiva, gestión de tráfico, vehículos de emergencia etc.). Tanto WiMAX como las redes móviles han sido los tipos de tecnología inalámbrica para dotar de conectividad a nodos en áreas metropolitanas. Pero con el advenimiento del IoT es que estas ahora no son factibles ya que representarían un alto costo para los miles y millones de nodos conectados a la red. Es por esto que actualmente surgen con fuerza las LPWAN, en las que una vez más se promueve un bajo consumo de energía y bajas tasas de transmisión de datos por nodo.

Para la selección de una de estas tecnologías que fueron ya descritas en el capítulo 1, se las detalla de manera más concisa a continuación en la tabla 6:

Tabla 6.

Comparación entre tecnologías LPWAN para aplicaciones IoT

Tecnología LPWAN	LoRaWAN	IEEE 802.11ah	Sigfox	Dash7
Cobertura Urbana	3 – 6 km	1 km	3 – 10 km	5 km
Cobertura Rural	15 km	1 km	30 – 50 km	5 km
Tasa de transferencia	Hasta 300 kbps	346.66 Mbps	300 bps	166.67 kbps
Com. Bidireccional	Si	Si	Limitada	Si
Frecuencia	No licenciada: 125, 433, 868 y 915 Mhz.	No licenciada Sub-bandas 1 Ghz.	Licenciada 868 y 902 Mhz	No licenciada 433, 868 y 915 Mhz.

Seguridad	Media	Alta	Muy alta.	Media
Movilidad	Si	Si	Limitada	Si
Tecnología Propietaria	No	No	Si	No

Adaptado de (Santos, Leroux, Wauters, Volckaert, & De Turck, 2018)

Al considerar una aplicación IoT con bajos tiempos de respuesta, la tecnología que más se ajustaría a este requerimiento es IEEE 802.11ah debido a su alta tasa de transferencia respecto a las otras. Sin embargo, su desventaja es su bajo rango de cobertura lo que incurriría en un gasto excesivo al dotar de cobertura a toda una urbe con varios gateways IEEE 802.11ah.

Por otra parte, Sigfox es el que ofrece un mayor rango de cobertura, pero a su vez es la tecnología LPWAN que tiene la tasa de transferencia más baja con apenas 300 bps, lo cual no es elegible para aplicaciones en tiempo real. Además de que, al ser una tecnología propietaria, el envío y recepción de mensajes supone un costo para el usuario. Por ende, las tecnologías que mejor se adaptan a este tipo de ambientes son LoRa y Dash7, debido a que poseen un gran alcance de cobertura y un bit rate aceptable para el envío de mensajes. Decididamente la tecnología a implementar es LoRa debido que a diferencia de Dash7 se logra sustancialmente una mejor tasa de transferencia.

4.4 Diagrama de red

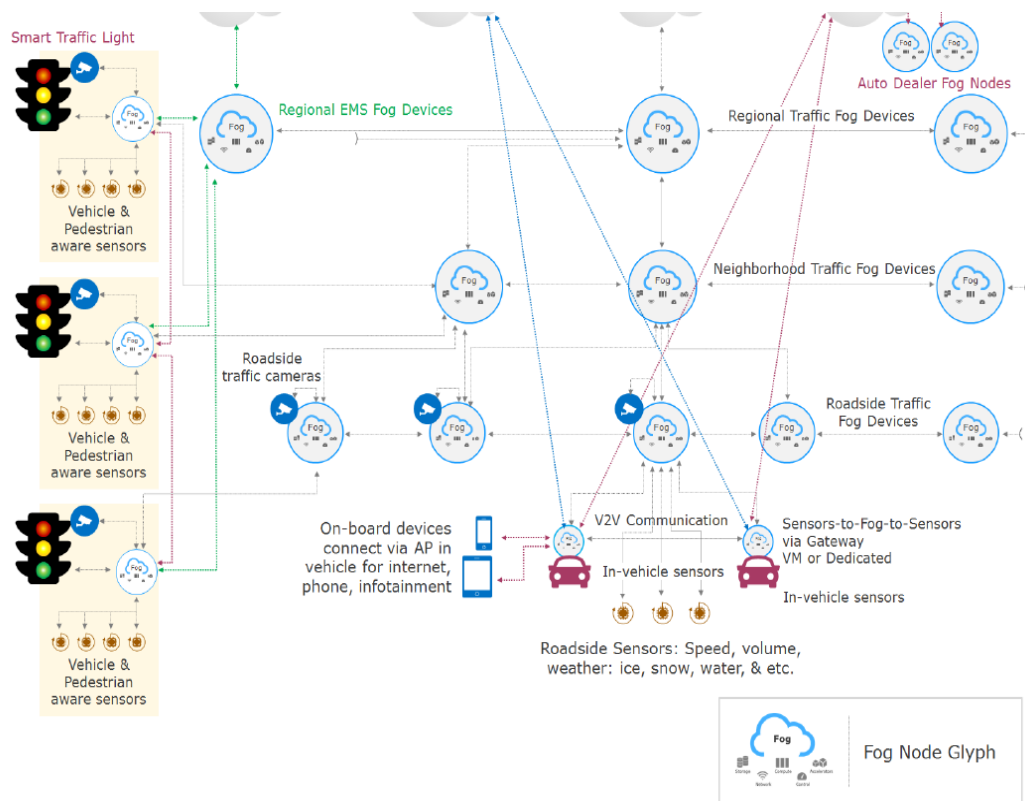


Figura 24. Diagrama de red de un ambiente IoT vehicular.

Tomado de (The OpenFog Consortium, 2018).

Como se puede observar en la figura 24 en el diagrama de red, los nodos de la computación en la niebla pueden enlazarse directamente con otro nodo que se encuentre en el rango de cobertura. Los RSU contienen procesamiento y conectividad para poder detectar vehículos, además que las intersecciones también cuentan con nodos de computación en la niebla. Se puede notar que este diagrama contiene nodos jerárquicos de la computación en la niebla mediante el cual en el primer nivel se encuentran emplazamientos o RSU que detectan los vehículos en las distintas intersecciones. Luego se puede observar también de nodos a nivel barrial o parroquial y por último nodos regionales que asisten a los nodos inferiores y enrutan la información a la computación en la nube.

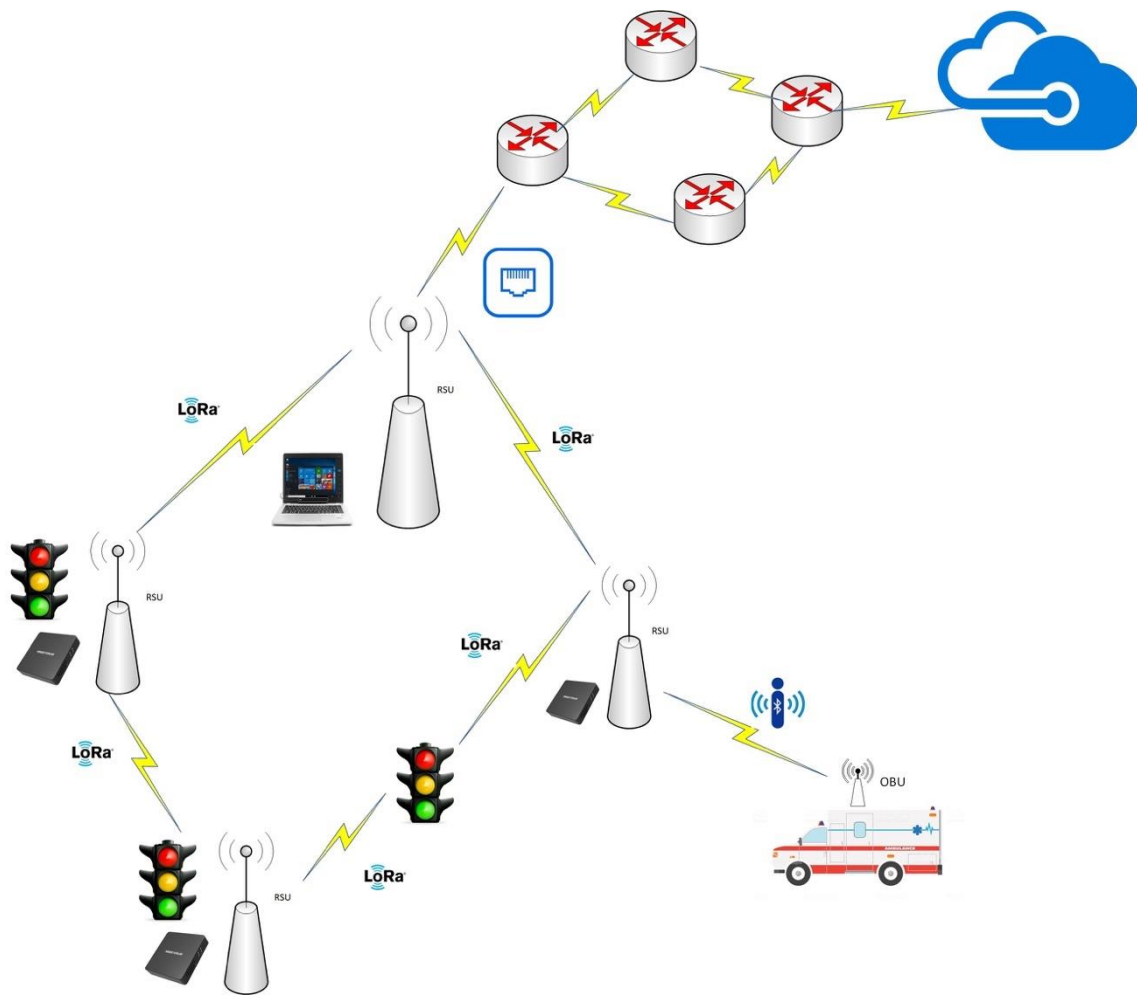


Figura 25. Diagrama de red de prototipo.

El diagrama de red del prototipo de la figura 25, consiste en cuatro nodos de la computación en la niebla. Uno de ellos es un RSU con emplazamiento para poder activar el primer semáforo lo mas pronto posible, mientras que los tres nodos restantes se encuentran interconectados directamente con los semáforos de la intersección. Se implementa una red malla de LoRa lo que tiene como ventaja de que el primer RSU que detecte la señal del OBU, tendrá la tarea de enrutar el mensaje de alerta a lo largo del camino hasta el destino final del vehículo de emergencia. Además de que todos los nodos logran conectividad entre si gracias a este tipo de topología de red. El procesamiento se realiza en las minicomputadoras Z83 - F y se apoyan en el nodo jerárquico que contiene mas procesamiento al poseer una PC y conectividad WAN hacia un servidor en

AWS. El OBU consiste de un módulo beacon HM-10 que opera en las bandas ISM con tecnología Bluetooth 4.0.

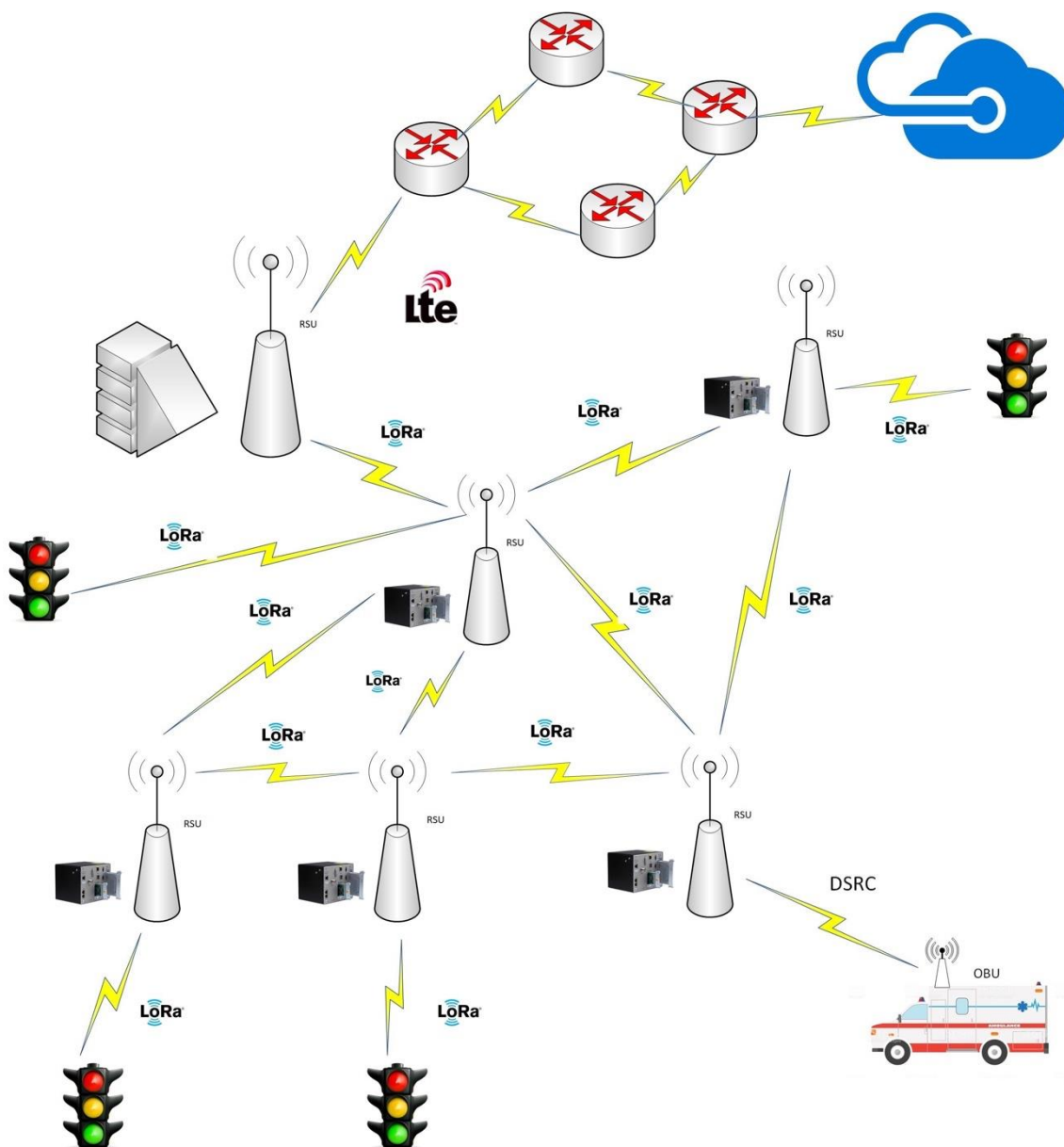


Figura 26. Propuesta de diagrama de red en un ambiente IoT vehicular real.

En la figura 26 se puede observar una topología de red parecida a la de la figura 25, con la diferencia de que hay mas nodos de la computación en la niebla distribuidos entre las intersecciones para activar mediante LoRa los semáforos.

Se puede apreciar tres niveles de la computación en la niebla que administran los nodos inferiores y un nodo parroquial o metropolitano con conectividad WAN a través de LTE o LTE CAT M1 el cual este último es un protocolo de IoT para aplicaciones en tiempo real. Los nodos de la computación en la niebla son routers industriales Cisco IR 910. Estos dispositivos tienen la capacidad de ejecutar cualquier distribución de Linux y agregar módulos LoRa para su conectividad. La comunicación V2I es mediante DSRC la cual se encuentra estandarizada en el protocolo IEEE 802.11p en la frecuencia de 5.9 Ghz lo cual tiene como ventaja de ser una frecuencia licenciada y no sufrir de interferencia con las bandas ISM.

4.5 Desarrollo de prototipo

El desarrollo e implementación del prototipo constará de nodos que se encuentren distribuidos en una maqueta que simula ser una urbe con varias intersecciones. Los nodos de la computación en la niebla correrán un aplicativo el cual les permita detectar mediante Bluetooth el beacon que será instalado en el vehículo de emergencia y a su vez poder enviar los mensajes mediante un transceptor LoRa. Esto con el fin de que la alerta llegue la próxima intersección para la habilitación del semáforo en luz verde y así sucesivamente en las siguientes intersecciones hasta que el vehículo de emergencia llegue a su destino.

Los materiales y herramientas para utilizar son las siguientes:

- Mini PC.
- Módulo HM-10 Bluetooth 4.0. DS-Tech.
- Transceptor LoRa sx 1278.
- Bluetooth 4.0 Integrado en Z83 - F Mini PC.
- Microcontrolador ATmega 2560 y SAM D-21.
- IDE Visual Studio.
- Aplicativo con lenguaje de programación en C#.
- Instancia Windows en Amazon Web Services.

4.5.1 Framework

El framework es uno de los componentes mas importantes a la hora de implementar una aplicación IoT. Se debe tomar en cuenta de que el framework abarque las herramientas y entorno de desarrollo necesario para incluir a las LPWAN, comunicación por sockets e interfaces gráficas de usuario personalizables para un mejor manejo de los nodos.

4.5.1.1 DSA

Distributed Services Architecture o simplemente conocido por sus siglas (DSA), es un framework de sistema abierto orientado a IoT con el objetivo de poder enlazar cualquier objeto sin importar del fabricante que sea. Por lo que DSA sobre todo asegura la interoperabilidad en un ambiente IoT en el que existen dispositivos que son elaborados por distintos fabricantes.

DSA utiliza tres componentes principales para su funcionamiento:

- **DS Broker:** Es un software basado en Apache 2.0 que se ejecuta en un dispositivo con mejores prestaciones para manejar los atributos de los nodos, administrar los DS Links, permitir y denegar permisos, agregar los datos de los nodos entre muchas otras.
- **DS Link:** El DS Link es el que permite enlazar a los distintos nodos de IoT con los servicios de suscripción en el broker, sin importar el fabricante que se elija.
- **nodeAPI:** Permite la comunicación y envío de mensajes entre los distintos nodos en el ambiente de IoT.

El funcionamiento de DSA consta como el de un servicio de suscripción mediante el cual los nodos se suscriben a un DS Broker referente a la información que están agregando. En otras palabras, los nodos que censan temperatura,

mediante un DS Link se suscriben a un DS Broker que solamente agrega y procesa los datos de temperatura como se lo puede notar en la figura 27.



Figura 27. Arquitectura DSA con distintos nodos que se suscriben a sus brokers respectivos para la agregación de datos.

Tomado de (DSA, 2018)

4.5.1.2 .NET

El framework de .NET es una plataforma de software desarrollada por Microsoft. En un principio el objetivo de este framework fue el de desarrollar aplicaciones tanto de escritorio como web; pero actualmente ofrecen también el desarrollo en ambientes móviles y aplicaciones Windows universales.

El lenguaje de programación que se utiliza en este framework es el de C#, pero también hay la posibilidad de programar las aplicaciones en Python, Javascript, PHP entre otros como se lo puede observar en la figura 28.

El framework de .NET tiene la posibilidad del uso de librerías para la utilización de comunicación serial e incluir redes LPWAN, además de poder incluir *web sockets* para la comunicación con la computación en la nube mediante TCP o UDP. También es posible la vinculación de protocolos de aplicación de IoT como

MQTT y CoAP. Además, las aplicaciones permiten la personalización de aplicaciones mediante interfaz gráfica para la administración de los nodos de IoT.

Las aplicaciones mas idóneas para un ambiente en computación en la niebla son las aplicaciones de consola o Windows forms para el desarrollo con interfaz gráfica.



Figura 28. Diagrama de la arquitectura del framework de .NET.

Tomado de (Guru99, 2013)

4.5.1.3 Comparación y elección de framework

Para el prototipo a implementar se escoge el framework de .NET debido a que se diseñan aplicaciones de escritorio o Windows Forms para que sean implementadas tanto en los nodos de la computación en la niebla como en la computación en la nube.

Las ventajas que ponen como diferenciador para escoger este framework es que mediante la comunicación serial es posible anexar los módulos de LoRa a los nodos de la computación en la niebla. Mientras que en DSA se depende de los DS Links que deben tener compatibilidad con este tipo de tecnología y que aun no lo tiene. También es importante resaltar que .NET es compatible con sistemas operativos Windows y cualquier distribución de linux mediante el framework mono project.

4.5.2 Nodos de la computación en la niebla

Los nodos de la computación en la niebla son los que se caracterizan por tener cómputo, almacenamiento y conectividad como se lo describió anteriormente en el capítulo 1. Los dispositivos o computadores que se ajustan a un prototipo reducido son las Mini PC con procesador Intel Atom y los Raspberry Pi 3B +. Por otra parte para la utilización de un nodo de la computación en la niebla en un ambiente real, se utilizan equipos industriales como los Cisco IR 910 y 809. Estos equipos de Cisco permiten correr sistema operativo Linux para el procesamiento y la posibilidad de comunicarse mediante LPWAN específicamente con LoRa.

4.5.2.1 Z83 - F Mini PC

La Z83 - F es un computador dentro de la categoría de minicomputadores, pero con prestaciones y recursos al nivel de un ordenador de escritorio.

Esta minicomputadora ejecuta el sistema operativo Windows 10 Pro por defecto, pero también es compatible con sistemas operativos de Linux. Se lo puede observar en la figura número 29.



Figura 29. Minicomputadora Z83 - F.

Tomado de (Verge, 2018)

Sus especificaciones principales son las siguientes:

- Procesador Intel Atom x5-z8350.
- Memoria ram de 4 GB.
- Gráficos Intel HD 500.
- HDD 32 GB.
- Puerto Gigabit LAN.
- Bluetooth 4.0 integrado.

4.5.2.2 Raspberry Pi 3B+

El Raspberry pi es un microcomputador de bajo costo del tamaño de una tarjeta de crédito que se conecta a un monitor o televisión y utiliza periféricos tales como un teclado o ratón. (Raspberry Org, 2018)

Este microcomputador puede correr el sistema operativo raspbian el cual es una distribución de Linux basada en Debian se puede observar ciertas especificaciones en la figura 30. El hardware contiene las siguientes características:

- Procesador ARM C rtex A-53 Quad Core de 64 bit con una frecuencia de reloj de 1.4 GHz.
- 1 GB LPDDR2 SRAM.
- Wireless LAN con banda dual de 2.4 y 5 GHz.
- Bluetooth 4.2 y Bluetooth Low Energy (BLE).
- Posibilidad de energizarse mediante Power over Ethernet.

Este microcomputador ser  capaz de recibir el mensaje que emita el veh culo mediante Bluetooth, esto lo realizar  mediante un Script programado a trav s de Python. A continuaci n, para el procesamiento del mensaje si se encuentra en alerta el veh culo, se determina que intersecci n o sem foro debe activarse en verde y tambi n a cu l pr ximo nodo se tiene que alertar para la habilitaci n de pr ximas intersecciones.

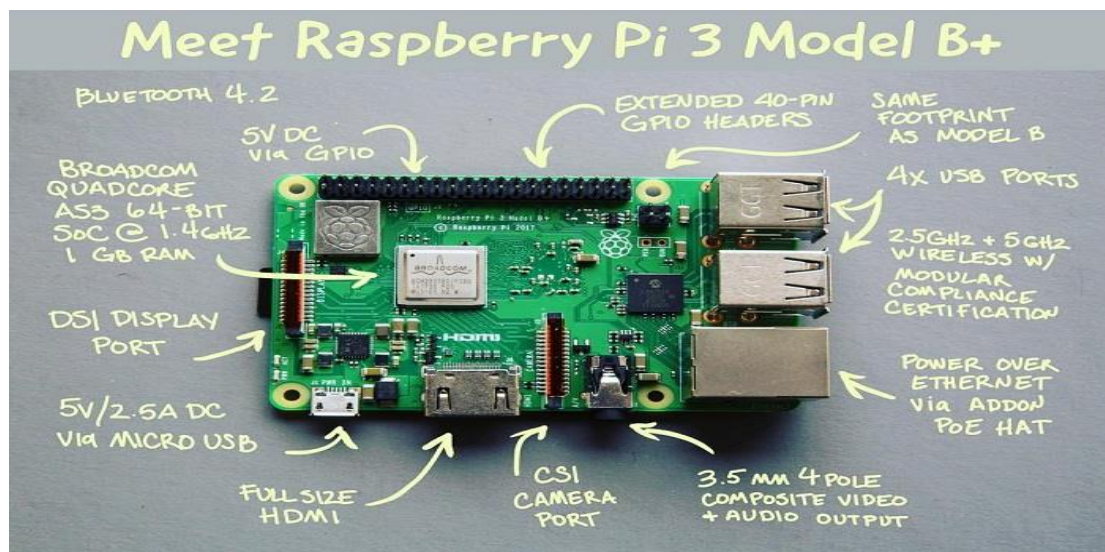


Figura 30. Especificaciones de hardware de Raspberry Pi 3 B+.

Tomado de (Santos Sara, 2018)

El programa o procesamiento se lo realiza en una aplicación que está escrita en el lenguaje de programación C#. Específicamente es una aplicación de Windows Forms la cual se ejecuta de manera nativa en sistemas operativos Windows; por lo que será necesario instalar una plataforma con características *cross-platform* mediante un software en raspbian. Dicha plataforma es Mono la cual es patrocinada por Microsoft que permite ejecutar y compilar el código del framework de .NET basada en estándares ECMA y C#.

Para la instalación de esta plataforma se ejecuta los siguientes comandos en el terminal de raspbian que se pueden apreciar en la figura 31. Tomar en cuenta que los siguientes comandos son para el sistema operativo raspbian en su versión 9.0.

```
sudo apt install apt-transport-https dirmngr
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80
--recv-keys 3FA7E0328081BFF6A14DA29AA6A19B38D3D831EF
echo "deb https://download.mono-project.com/repo/debian
stable-raspbianstretch main" | sudo tee
/etc/apt/sources.list.d/mono-official-stable.list
sudo apt update
sudo apt install mono-devel
```

Figura 31. Comandos para la instalación de *mono project* en Raspbian.

Se realiza la instalación de Mono en modo de desarrollador, el cual funciona para compilar y ejecutar el código. En caso de utilizar alguna librería adicional descargada de Nugget, es recomendable instalar la plataforma completa mediante el comando “`sudo apt install mono-complete`”.

Para la interconexión de los nodos, se utiliza un microcontrolador que maneja los módulos LoRa.

4.5.3 Implementación de Vehículo a Infraestructura

Como ya se mencionó anteriormente, la comunicación entre el vehículo de emergencia y el RSU o nodo de la computación en la niebla, se la realiza mediante Bluetooth. El módulo que actuará como beacon es el Bluetooth HM-10 en el vehículo de emergencia. Este módulo viene por defecto configurado como esclavo para enlazarse con otro módulo que sea maestro. Por lo que mediante comunicación serial y comandos AT es indispensable configurar a dicho módulo para que se encuentre en modo de beacon y que solamente realice un anuncio o broadcast de mensajes con su RSSI.

Por tanto, para la configuración del módulo HM-10 se requerirá un conversor USB a Serial TTL o un adaptador FTDI. Es importante tomar en cuenta que el RX (receptor) del HM-10 se conecta con el TX (transmisor) del conversor USB a TTL. La conexión entre estos dos módulos se la puede ver a continuación en la siguiente figura.

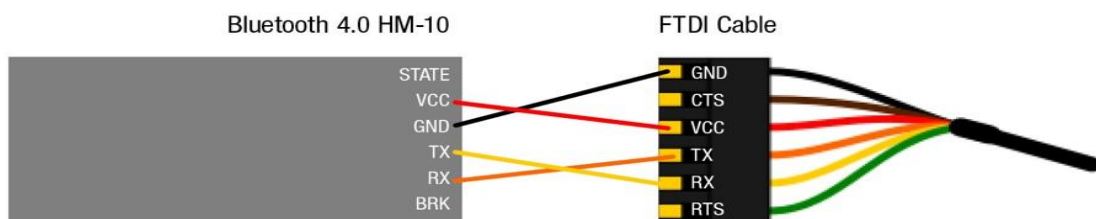


Figura 32. Conexión HM-10 con adaptador FTDI

Tomado de (Chen, 2016)

Una vez obtenida la conexión entre pines; según el sistema operativo se abre el programa de comunicación serial Putty para Windows o CoolTerm para dispositivos Mac. Se establece la comunicación en 9600 baudios y el puerto COM al que se encuentre conectado en el computador. Una vez establecida la comunicación en el puerto correspondiente, se procede a enviar el comando AT; por consiguiente, el módulo HM-10 tiene que responder con mensaje "Ok".

Una vez que el módulo responde, se puede introducir varios comandos AT para consultar la dirección MAC, nombre, comunicación etc. Por ejemplo, en la siguiente figura se puede observar como el módulo responde según el comando ingresado; en este caso se consulta la dirección MAC y la velocidad de comunicación en 0 lo que representa 9600 baudios. Este paso se lo puede apreciar en la figura 33.

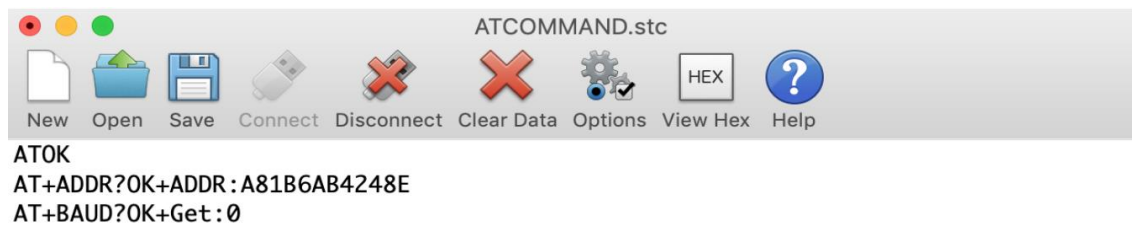


Figura 33. Conexión HM-10 con adaptador FTDI

Ahora se procede con los siguientes comandos para la configuración del módulo HM-10 como beacon introduciendo los siguientes comandos:

- AT+RENEW: Borra toda configuración y reestablece al módulo a su configuración por defecto.
- AT+RESET: Reinicia el dispositivo.
- AT+ADVI0: Establece el intervalo de mensajes cada 100 milisegundos.

- AT+NAMEvehiculoEmergencia: Se personaliza un nombre para que se lo pueda distinguir, en este caso su nombre es “vehiculoEmergencia”
- AT+ADTY3: Establece al módulo HM-10 como anunciante para transmitir mensajes, deshabilitar las funciones de escaneo y que otro dispositivo se empareje al módulo Bluetooth
- AT+IBEA1: Establece oficialmente al módulo como un dispositivo beacon.
- AT+DELO2: Configura que los mensajes sean transmitidos como broadcast
- AT+PWRM0: Coloca al dispositivo como dormido o auto-sleep para reducir el consumo de energía
- AT+RESET: Se reinicia una última vez al dispositivo para que se inicialice con las nuevas configuraciones.

Una vez configurado el módulo HM-10 como beacon, cualquier otro dispositivo que se encuentre en modo de escaneo puede recibir el mensaje con su RSSI que posee una ventana de 100 milisegundos que transmite el HM-10. Para poder escuchar el mensaje en los nodos de la computación de niebla se utiliza el módulo Bluetooth 4.0 que tiene incorporado el Z83 - F o la Raspberry Pi 3 B+.

En el caso de la Raspberry Pi 3 B+, para que los mensajes sean descifrados y así obtener el RSSI del beacon es necesario instalar la librería bluepy disponible en el lenguaje de programación Python para este microcomputador. Para proceder con la instalación de en el sistema operativo de Raspbian abrir el terminal de GNU/Linux e ingresar los comandos descritos en la figura 34.

```
$ sudo apt-get install python3-pip libglib2.0-dev  
$ sudo pip3 install bluepy
```

Figura 34. Comandos para la instalación de Bluepy en Raspbian.

En los comandos descritos en su primera línea se instala el compilador de Python en su versión más reciente y posteriormente se instala la librería de bluepy.

Ya obtenida la librería y el compilador de Python, se procede a crear un script con el nombre “ble_RSSI.py” para obtener la dirección MAC y RSSI del beacon HM-10 como se puede observar en la siguiente figura:

```

for ii in devices:
    print(ii.addr)
    if ii.addr == u'a8:1b:6a:b4:24:8e':

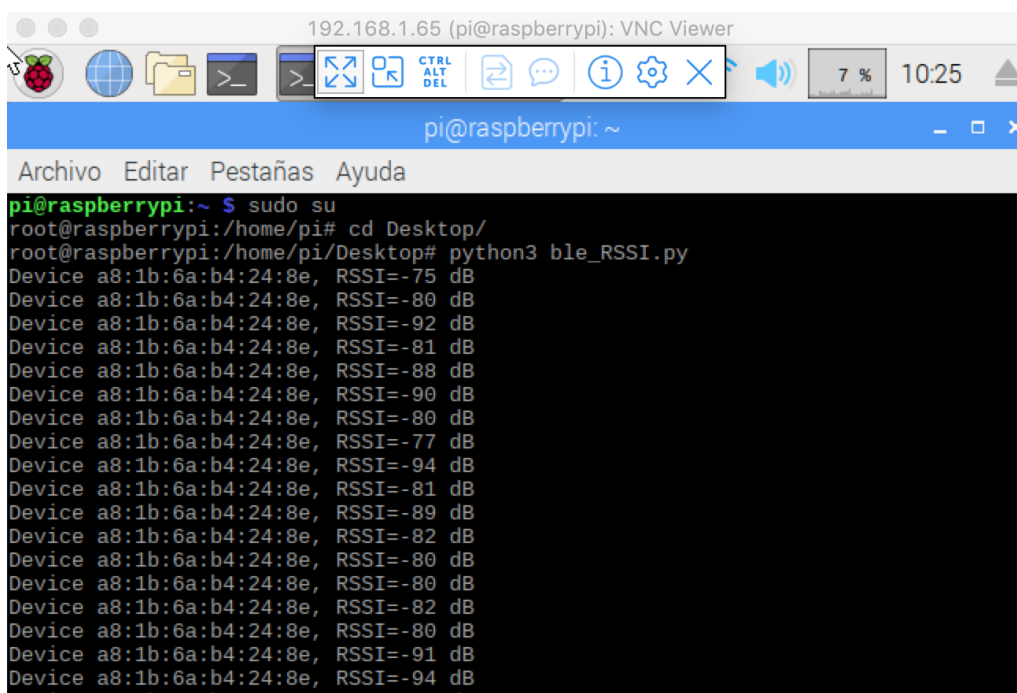
        print("Device %s, RSSI=%d dB" % (ii.addr,ii.rssi))
        message = "%s %d" % (ii.addr,ii.rssi)
        msg = message.encode('utf-8')
        sock.sendto(msg, ("127.0.0.1", 2019))

```

Figura 35. Python script para obtener la dirección MAC y RSSI de beacon hm-10 con librería de bluepy.

El script de la figura 35 instancia un objeto scanner para acceder a las funciones de la librería y obtener valores como el UUID, dirección MAC, RSSI y otros parámetros según los requerimientos del programador. Se crea un bucle infinito con la función while para escanear beacons con un tiempo definido cada 0.05 segundos; es decir 50 milisegundos. Con la función for es posible acceder a cada dispositivo escaneado y posteriormente con la siguiente función if, solamente recolectar e imprimir la dirección MAC y RSSI del beacon HM-10.

Para ejecutar el script se abre la terminal de raspbian y con el comando “sudo python3 ble_RSSI.py” se ejecuta el comando obteniendo los primeros resultados desplegados en la figura 36.



```

192.168.1.65 (pi@raspberrypi): VNC Viewer
pi@raspberrypi: ~
Archivo Editar Pestañas Ayuda
pi@raspberrypi:~ $ sudo su
root@raspberrypi:/home/pi# cd Desktop/
root@raspberrypi:/home/pi/Desktop# python3 ble_RSSI.py
Device a8:1b:6a:b4:24:8e, RSSI=-75 dB
Device a8:1b:6a:b4:24:8e, RSSI=-80 dB
Device a8:1b:6a:b4:24:8e, RSSI=-92 dB
Device a8:1b:6a:b4:24:8e, RSSI=-81 dB
Device a8:1b:6a:b4:24:8e, RSSI=-88 dB
Device a8:1b:6a:b4:24:8e, RSSI=-90 dB
Device a8:1b:6a:b4:24:8e, RSSI=-80 dB
Device a8:1b:6a:b4:24:8e, RSSI=-77 dB
Device a8:1b:6a:b4:24:8e, RSSI=-94 dB
Device a8:1b:6a:b4:24:8e, RSSI=-81 dB
Device a8:1b:6a:b4:24:8e, RSSI=-89 dB
Device a8:1b:6a:b4:24:8e, RSSI=-82 dB
Device a8:1b:6a:b4:24:8e, RSSI=-80 dB
Device a8:1b:6a:b4:24:8e, RSSI=-80 dB
Device a8:1b:6a:b4:24:8e, RSSI=-82 dB
Device a8:1b:6a:b4:24:8e, RSSI=-80 dB
Device a8:1b:6a:b4:24:8e, RSSI=-91 dB
Device a8:1b:6a:b4:24:8e, RSSI=-94 dB

```

Figura 36. Obtención de mensaje transmitido de beacon HM-10 e imprimiendo su dirección MAC y RSSI.

Una vez obtenido el RSSI podemos enviar al aplicativo en C# que se detalla a continuación, mediante el uso de sockets. Como se pudo observar en la figura 19 se utiliza la librería socket para posteriormente enviar a través de la dirección de loopback en el puerto 2019, enviar el datagrama de transporte UDP para la recepción de este mensaje.

En el caso del nodo Z83 - F este proceso es mucho mas sencillo debido que gracias al framework de .NET es posible mediante la instalación de librerías de *beacons* mediante el repositorio *NuGet package* para poder establecer al dispositivo en modo de escaneo y así poder recibir los mensajes del beacon.

4.5.4 Red malla de LoRa

LoRa se caracteriza por tener una topología de red tipo estrella en la que los nodos se comunican entre sí mediante un LoRa Gateway el cual tiene la funcionalidad de enrutar los datos hacia la red WAN u otros nodos LoRa adyacentes. Sin embargo, la implementación de una red tipo estrella en una

ciudad supondría un alto costo debido a la gran cantidad de nodos LoRa Gateway que tendrían que invertirse para cubrir toda un área urbana.

Es por lo que, en este prototipo la conectividad entre nodos se la realiza mediante una topología de red tipo malla. Los beneficios de utilizar esta topología de red es que brinda una mayor área de cobertura y el aseguramiento de que los paquetes lleguen a su destino sin incurrir en la colocación de gateways. En efecto, un estudio realizado en una universidad de China se realizó pruebas de conectividad inalámbrica en un área de 800m por 600m con 19 dispositivos LoRa para medir el rendimiento de topologías de red tipo estrella y malla. La topología de red tipo malla alcanza un PDR (Packet Delivery Ratio) de 88.49% y de la misma manera en las mismas condiciones la topología de red tipo estrella solamente alcanza un 58.7%. (Lee & Ke, 2018)

El módulo que se utiliza es el SX1278 del fabricante Semtech de la figura 37, el cual dispone de las siguientes características:

- Frecuencia de operación: 433 Mhz.
- Modulación: FSK/GFSK/MSK.
- Sensibilidad de recepción: -139 dBm.
- Potencia de transmisión: 20 dBm.
- Consumo energético: < 1uA.
- Tasa de transferencia: < 300 kbps.



Figura 37. Módulo LoRa SX1278 del fabricante Semtech con una frecuencia de operación de 433 Mhz.

Para la implementación del módulo LoRa con los microcontroladores ATMEGA 2560 y SAM D-21, es necesario la librería Radiohead para la comunicación entre dichos dispositivos y el establecimiento de la red malla.

```

#include <RH_RF95.h>
#include <RMesh.h>
#define RH_MESH_MAX_MESSAGE_LEN 50
#define RH_HAVE_SERIAL Serial

#define MESH1_ADDRESS 1
#define MESH2_ADDRESS 2
#define MESH3_ADDRESS 3
#define MESH4_ADDRESS 4

RH_RF95 driver(53,21); // SS y INT0 de arduino

RMesh manager(driver, 1);

void setup()
{
  Serial.begin(9600);
  Serial.println("Inicializar modulo LoRa");
  if (!manager.init())
    Serial.println("init failed");
}

```

Figura 38. Establecimiento del módulo LoRa en el IDE Arduino.

En la figura 38 se puede observar cómo se incluyen las librerías RH_RF95.H y RMesh.H, siendo la primera para la comunicación de módulos LoRa SX1276 y SX1278. Mientras que la segunda librería es la que permite habilitar a estos transceptores como dispositivos de tipo malla. Se establece el pin SPI de comunicación síncrona entre el microcontrolador y el transceptor. Dicho pin es el 53 de la placa de Arduino y también se establece la interrupción externa 0 ubicada en el pin 23.

La instancia para controlar el transceptor en estos dos pines se la denomina driver la cual nos sirve para indicar en la librería de la red tipo malla que este módulo es el número 1 en su red para enlazarse con los otros nodos adyacentes

4.5.5 Aplicativo C#

El procesamiento de las alertas y mensajes que se reciben del vehículo se las realiza en aplicaciones Windows Forms en el lenguaje de programación C#. Estas aplicaciones son ejecutadas en los distintos nodos de la computación en la niebla; ya que cada uno de estos tiene la posibilidad de recibir la alerta y procesar dicha información. Esto se lo realiza con el fin de que indistintamente en donde el vehículo se encuentre, la arquitectura distribuida de la computación en la niebla permitirá recibir los mensajes para atender la emergencia en todo momento.

Los nodos que se encuentran más próximos a los vehículos, es decir los RSU que en este caso pueden ser emplazamientos o semáforos ejecutan un programa con interfaz gráfica que se puede observar en la figura 39. En este programa se puede observar el medidor de señal en dbm y el código que se ejecuta es el responsable de determinar si el mensaje se encuentra dentro de un valor mayor a -48 dbm para la activación de semáforos en verde hasta el destino.

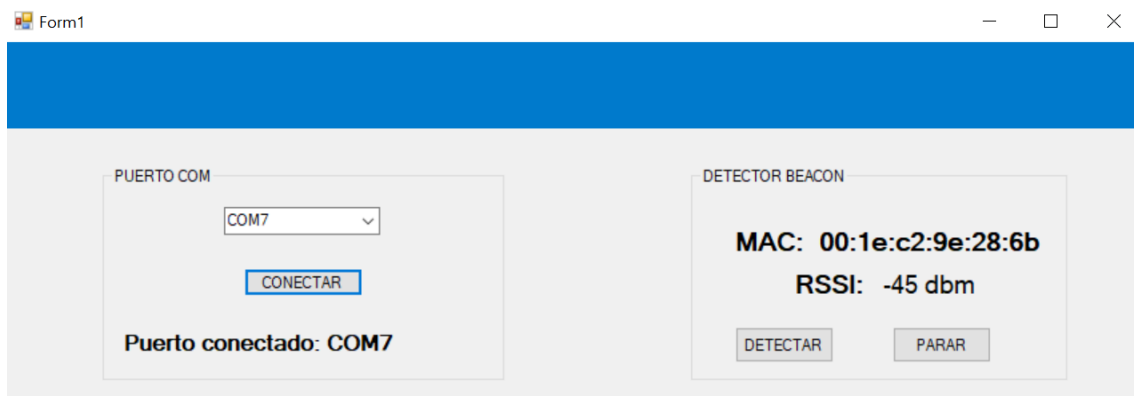


Figura 39. Interfaz gráfica de aplicación de los nodos de la computación en la niebla

Por otra parte, en la computación en la niebla al existir nodos jerárquicos, se considera un nodo administrativo que verifica la actividad o inactividad de los nodos en una zona. Este nodo posee de un programa con interfaz gráfica que se puede notar en la figura 40 y tiene la capacidad de administrar estos nodos.

Es decir que mediante esta interfaz es posible observar el funcionamiento de estos y además de dotar de procesamiento en caso de que un nodo se encuentre inactivo. Además, este nodo tiene la capacidad de conectividad WAN al servicio de computación en la nube para que todo el sistema pueda ser monitoreado constantemente cumpliendo con la interacción M2P (Máquina a Persona).

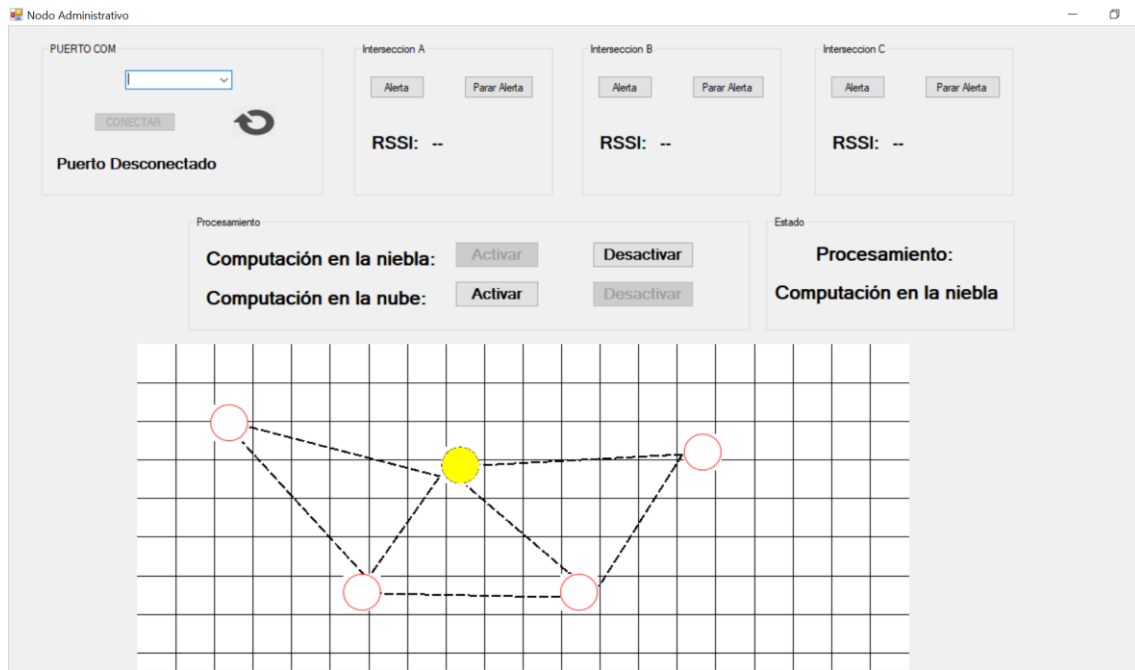


Figura 40. Aplicación que se ejecuta en el nodo de administración de la computación en la niebla.

Por último, en la computación en la nube, mediante máquina virtual se ejecuta de igual manera otra aplicación de Windows Forms para revisar el estado de todos los nodos de la computación en la niebla que se la puede observar con mayor detalle en la figura 41. Al contar con servidores dedicados con altas capacidades de procesamiento en la computación en la nube, es posible administrar los nodos y además los semáforos que se encuentran en verde o rojo e inclusive saber si los nodos que están en la niebla se encuentran atendiendo la emergencia de los vehículos.



Figura 41. Aplicación que se ejecuta en AWS.

4.5.6 Instancia Windows en Amazon Web Services

Para la habilitación de un servicio de la computación en la nube, se cuenta con grandes proveedores como Amazon Web Services, Google Cloud y Azure. Para este prototipo se utilizará Amazon Web Services conocido por sus siglas en inglés AWS. Este proveedor de computación en la nube se encuentra como líder respecto al servicio de IaaS según el cuadrante mágico de Gartner. (Shiff, 2018)

Lo que se necesita primordialmente para habilitar una máquina virtual en AWS entre otras, es la creación de una cuenta y la verificación de esta. Una vez obtenida una cuenta verificada se puede optar por los distintos servicios que van desde máquinas virtuales, bases de datos entre muchas otras. El segmento en el que se habilita la máquina virtual se la denomina EC2. Dentro de dicho segmento se puede visualizar la opción de crear una nueva instancia y en la cual se desplegará un listado de los distintos sistemas operativos. Para este caso en específico se utiliza un sistema operativo Windows Server 2016.

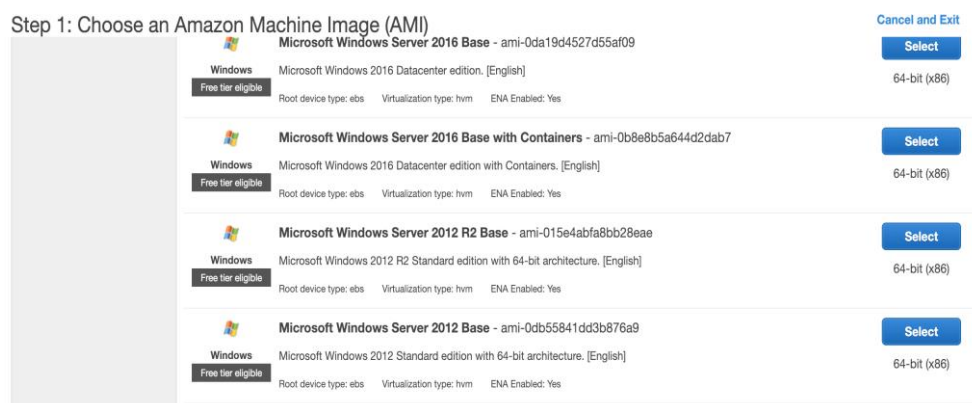


Figura 42. Selección del sistema operativo en una máquina virtual en AWS.

Una vez escogido el sistema operativo, se genera una tabla que se puede ver en la figura 42 con las especificaciones de la máquina que se utilizará. Las características son el número de procesadores, la cantidad de memoria RAM y el soporte de IPV6 como las más importantes. Se decide escoger una máquina virtual con 1 CPU, 1 GB de memoria RAM y sin soporte de IPV6 que para ejecutar la aplicación de Windows Forms es suficiente.

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ	Description ⓘ
Custom UDP Rule	UDP	2019	186.4.147.35/32	
All ICMP - IPv4	All	N/A	186.4.147.35/32	

Figura 43. Listas de control de acceso para la instancia Windows

Por último, se puede configurar las conexiones entrantes y salientes de la máquina virtual mediante listas de control de acceso (ACL) como se puede observar en la figura 43. Tanto como las conexiones entrantes y salientes se aceptará una conexión ICMP hacia y desde la dirección IP pública que maneje el nodo administrativo de la computación en la niebla. De igual manera el protocolo UDP mediante el puerto 2019 el cual está predeterminado para la comunicación entre las aplicaciones de la computación en la niebla y la aplicación que se ejecutará en la computación en la nube.

Una vez ya creada la máquina virtual se debe generar una llave o certificado para ingresar como administrador al sistema operativo. Este certificado es un

documento que se lo descargada directamente en el panel de EC2, por lo que cada intento de conexión se debe subir el archivo para descryptar la contraseña y así poder ingresar a la máquina virtual.

En las figuras 44 y 45 respectivamente se puede observar el proceso de autenticación para el ingreso a la instancia de Windows en AWS.

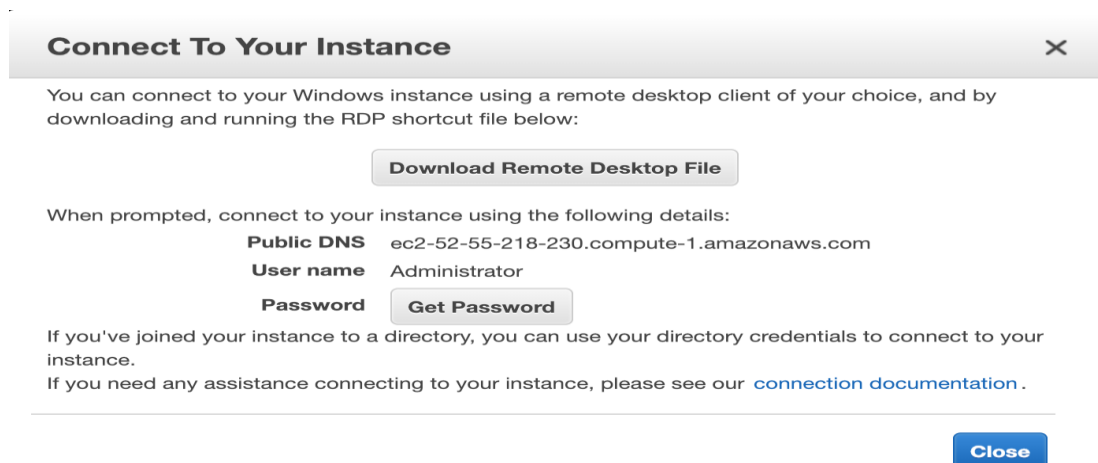


Figura 44. Panel de conexión a máquina virtual como administrador.

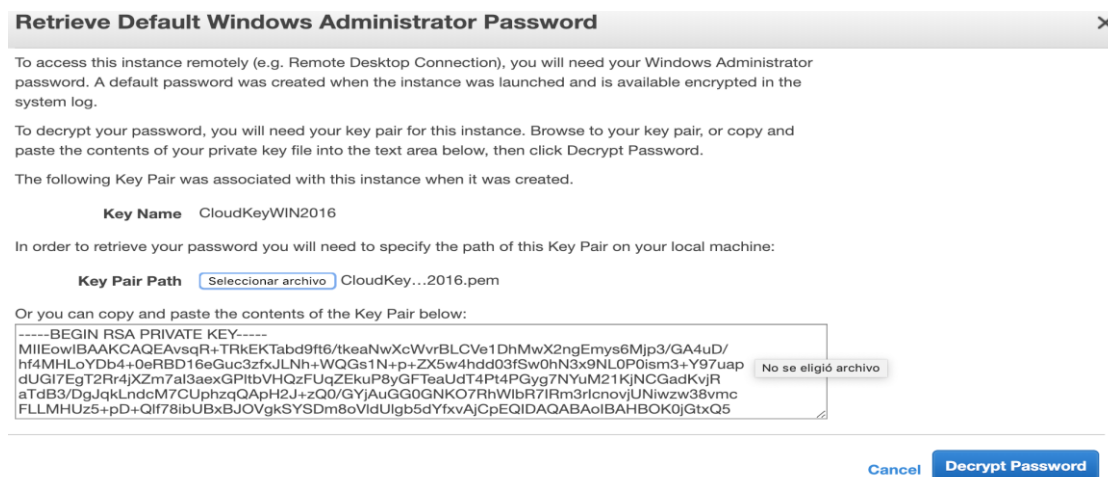


Figura 45. Descripción de contraseña mediante certificado.

Con dicho certificado se obtiene la contraseña y mediante el software Microsoft remote desktop es posible el control de la máquina virtual. En la figura 46 se puede observar que al momento de entablar la conexión con la máquina se

solicita la contraseña descifrada anteriormente. Una vez entablada la conexión con la instancia Windows, se puede observar tanto en el escritorio o en el panel EC2 la información concerniente a esta máquina virtual.

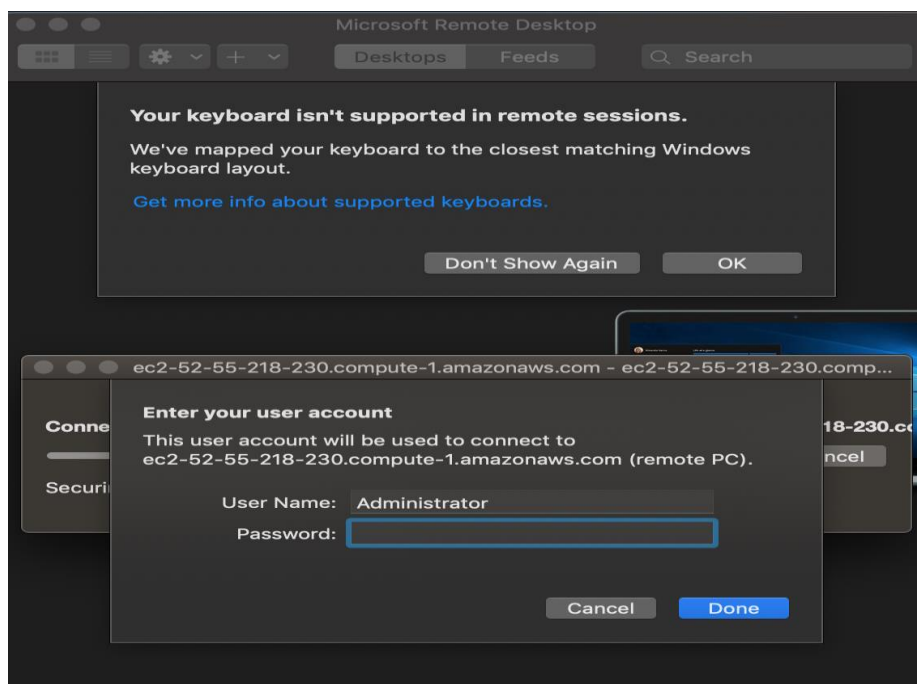


Figura 46. Pantalla de autenticación para el acceso a instancia Windows.

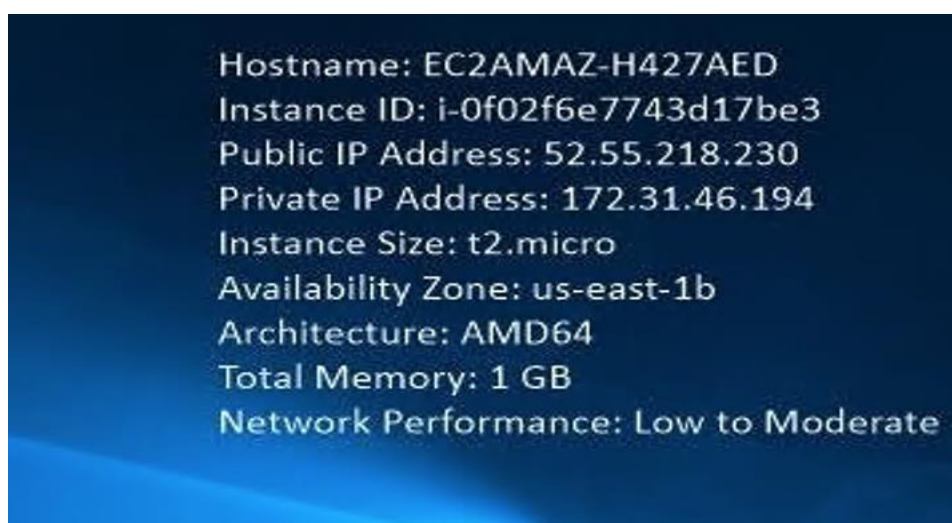


Figura 47. Direcciones IP pública y privada, ubicación y especificaciones.

Como se puede observar en la figura 47 se conoce cuál es la IP pública, IP privada, las especificaciones de la máquina virtual tanto como procesamiento, memoria entre otras y su localización. La ubicación de la máquina que corresponde a us-east-1b es el estado de Virginia en los Estados Unidos de América. Para una mayor información de las otras zonas disponibles de AWS se las adjunta en el anexo.

```
sh-3.2# ping 52.55.218.230
PING 52.55.218.230 (52.55.218.230): 56 data bytes
64 bytes from 52.55.218.230: icmp_seq=0 ttl=111 time=596.048 ms
64 bytes from 52.55.218.230: icmp_seq=1 ttl=111 time=228.803 ms
64 bytes from 52.55.218.230: icmp_seq=2 ttl=111 time=304.047 ms
```

Figura 48. ICMP hacia la IP pública de la instancia Windows en AWS.

Para realizar la prueba de conectividad se tiene que realizar un ICMP desde la IP pública que se permitió en la lista de control de acceso. Como se puede observar en la figura 48 la máquina virtual con dirección IP pública 52.55.218.230 responden satisfactoriamente con un retardo que oscila entre los 200 y 500 ms.

4.5.7 Ambiente de simulación

El ambiente de simulación se lo realiza en una maqueta de 90 cm de ancho por 50 cm de largo, en el que consta un camino con tres intersecciones semaforizadas hasta el destino como se lo puede observar en la figura 49. Cada intersección cuenta con cuatro sentidos; es decir que las calles son de doble vía. En la simulación se encuentran cuatro nodos de la computación en la niebla de los cuales tres son RSU con la capacidad de escanear y receptar los mensajes de alerta mediante beacons. Por otra parte, el nodo restante, se encarga de administrar los RSU de dicha zona y redirigir la información que es prioritaria hacia la computación en la nube. Este nodo de la niebla administrativo ejecuta una interfaz gráfica en la que se puede administrar los nodos y en caso de que algún nodo falle puede distribuir el procesamiento entre los otros RSU.

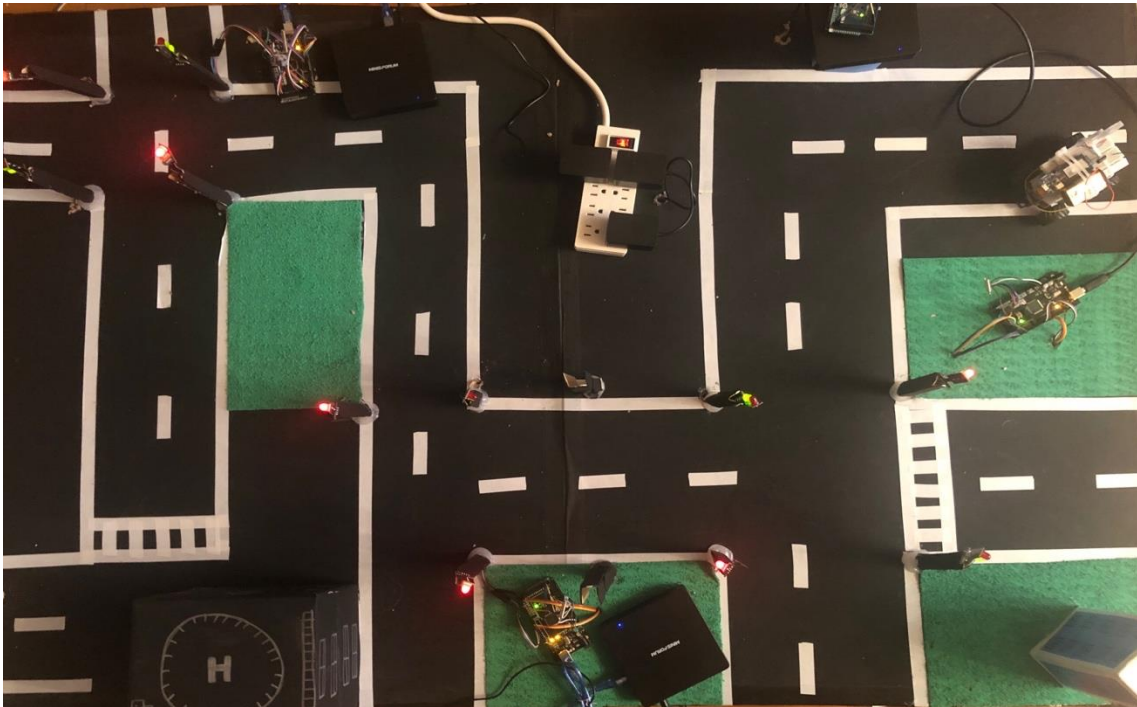


Figura 49. Ambiente de simulación con tres intersecciones, 4 nodos de la computación en la niebla y red malla de LoRa.

El OBU que se encuentra instalado en el vehículo de emergencia es un módulo HM-10 correspondiente a la tecnología de acceso Bluetooth 4.0, que se encuentra programado como beacon. Su consumo en modo BLE está en el rango de 50 y 200 uA por lo que puede ser energizado fácilmente mediante una batería de tipo moneda con capacidad de 1000 mAh. Mediante la siguiente fórmula se puede calcular que aproximadamente la duración es de entre 6 meses y 1 año y 5 meses. El factor de 0.7 se multiplica ante condiciones externas que puedan deteriorar la vida útil de la batería.

$$t = \frac{\text{capacidad de batería (mAh)}}{\text{carga del dispositivo (mA)}} \times 0.7 \quad (\text{Ecuación 2})$$

Cabe resaltar que en el cálculo de la descarga de la batería se toma en cuenta un escenario en el que el dispositivo se encuentra energizado continuamente. Para este caso en particular, el beacon solamente se energiza cuando el

vehículo se encuentra en una emergencia para alcanzar el nodo de la computación en la niebla más cercano. Por lo tanto, el dispositivo debe permanecer apagado debido a que se puede generar falsas alertas y los semáforos se habilitarán en verde cuando no exista una emergencia real. Debido a esto es que se puede deducir que la batería del dispositivo puede inclusive durar más de 2 años.

El escenario que se presenta en esta simulación es que el vehículo de emergencia pueda emitir una señal de alerta indistintamente del lugar en donde se encuentre en la urbe. Es por eso que, tanto en el ambiente de simulación como en un escenario real, los nodos de la computación en la niebla pueden estar distribuidos entre las intersecciones y también en los cruces de calles. Con esto se puede lograr que el vehículo pueda enviar la emergencia al nodo más cercano y habilitar los semáforos en verde hasta que el mismo llegue a su destino.

Los nodos de la computación en la niebla tienen la capacidad de habilitar todo el camino hacia el destino con semáforos en verde o también tienen la capacidad de ir habilitando la intersección en verde según el vehículo sigue avanzando.

El otro escenario de simulación es una solución IoT basada solamente en la computación en la nube. Esto se realiza con el fin de poder comparar el round trip delay time conocido por sus siglas RTT. Esto nos permite saber el tiempo de extremo a extremo; es decir el envío de la información hasta el nodo de procesamiento y el regreso hacia el actuador para la habilitación de los semáforos en verde

4.6 Análisis y resultados

Mediante el ambiente de simulación se pudo evidenciar notablemente que el procesamiento de los datos lo más cercano a los nodos permite reducir considerablemente los tiempos de respuesta. Para la justificación y comprobación de los resultados en el ambiente de simulación, se propuso

realizar una comparación del tiempo de respuesta entre una infraestructura de computación en la niebla y computación en la nube.

4.6.1 Comparación de tiempos de respuesta entre la computación en la nube y la computación en la niebla

Para la comparación de tiempos de respuesta entre estas dos tecnologías se utiliza de manera principal el concepto de RTT. El RTT se lo define como el tiempo que demora un paquete en alcanzar un host de destino y sumado el tiempo que toma en regresar al origen. En un escenario de computación en la niebla, esta situación puede variar ligeramente; debido a que el paquete que se envía para el procesamiento no necesariamente tiene que regresar al origen (vehículo). Por el contrario, el nodo que procesa la información envía la señal a los actuadores para modificar la señal de semaforización en las intersecciones.

Por otra parte, en la computación en la nube el mismo nodo que envía la alerta tiene que recibir las instrucciones del procesamiento en la computación en la nube que procesa la información para saber sobre cuales actuadores influye esta respuesta.

Para el cálculo de RTT existen cuatro fuentes de retardo de paquetes las cuales son: tiempo de transmisión, tiempo de propagación, tiempo de procesamiento del nodo y tiempo de encolamiento.

El tiempo de transmisión es el tiempo que le toma a un paquete en recorrer el enlace en donde influye la capacidad y velocidad máxima del medio. Por consiguiente este tipo de retardo puede variar según el medio de transmisión de las redes de acceso, agregación y *core*.

De igual manera, el tiempo de propagación es concerniente al medio de transmisión; sin embargo toma como factor principal a la distancia del enlace y la velocidad de propagación de las ondas que viajan por el medio.

El tiempo de procesamiento es el retardo que le toma a un servidor o aplicación de *backend* en procesar la información para tomar una acción. Y por último el tiempo de encolamiento es el nivel de congestión que tiene un equipo intermediario como lo puede ser un router. Cabe recalcar que para la computación en la niebla no es necesario medir o calcular el retardo de encolamiento debido a que se maneja el protocolo Bluetooth y LoRa en LPWAN las cuales manejan el reenvío de la información en capa física y capa de enlace de datos.

Para el cálculo de retardo de transmisión de la computación en la niebla se tiene que tomar en cuenta que pueden existir uno o más saltos para que el semáforo se coloque en verde. Además de que los enlaces y medios transmisión cambian debido a que nivel de capa de enlace se utilizan dos protocolos distintos.

Para el cálculo del enlace mediante el protocolo Bluetooth se toma en cuenta que el paquete que se envía del vehículo al RSU es un paquete de anuncio de BLE con datos de beacon como la dirección MAC, el ID de la compañía fabricante, UUID, números mayor, menor que sirven como identificador de los beacons entre otros.

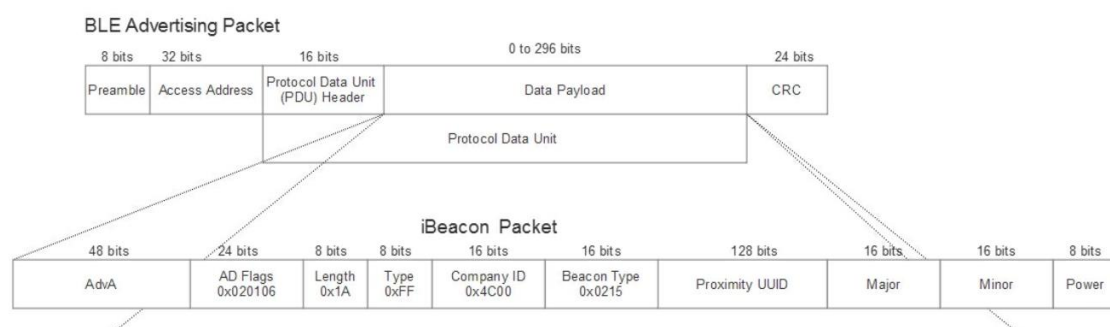


Figura 50. Paquete anunciante iBeacon.

Como se puede observar en la figura 50, el paquete que se envía puede tener un máximo de 376 bits. Por lo que para el cálculo del retardo de transmisión tomamos en cuenta este dato y la máxima velocidad del enlace que es de 32 Mbps en Bluetooth 4.0.

$$\text{tiempo de transmisión: } \frac{\text{Payload o longitud de paquete (bits)}}{\text{velocidad de transmisión } \left(\frac{b}{s}\right)} \quad (\text{Ecuación 3})$$

El resultado de la velocidad de transmisión entre el vehículo y el RSU en condiciones ideales es de $1,17 \times 10^{-5}$ segundos. Una vez que el RSU recibe el paquete del beacon, se descomprime este mismo para verificar la dirección MAC si corresponde a un vehículo de emergencia y se obtiene el RSSI. Al encontrarse estas dos aplicaciones en el mismo chasis o nodo, él envió de esta información a la aplicación en Windows Forms se lo realiza mediante el localhost; es decir la tarjeta de red.

El nodo de la computación en la niebla en este caso el Raspberry Pi B+ recoge el paquete beacon mediante script de bluepy que escanea beacons; por lo que es necesario enviarlo a la aplicación de Visual Studio a través de sockets mediante la tarjeta de red que de 330 Mbps. El paquete que se direcciona hacia la aplicación contiene la MAC, RSSI y mensaje. Se contabiliza 48 bits de dirección física, 4 bytes de RSSI y 6 bytes de mensaje debido a que contiene 6 caracteres, obteniendo una trama de datos de 128 bits.

Una vez obtenido estos datos se aplica el retardo de transmisión de la interfaz ethernet de 330 Mbps para él envió de información a Windows Forms, dando como resultado $3,9 \times 10^{-7}$ segundos. Este paso no es necesario realizarlo para el nodo Z83 - F.

A continuación, se realiza el procesamiento en la aplicación mediante la cual existe la facilidad de calcular el tiempo que demora el procesamiento en la aplicación. En este proceso para la Raspberry Pi 3B+ se obtiene los últimos 10 valores de RSSI para promediarlos y así obtener un valor más exacto del RSSI. Esto se realiza debido a que en el script de Python donde se recogen los RSSI se puede ver que el proceso de obtención de estos valores no son exactos y oscilan entre -60 dB y -70 dB. Se verifica la dirección MAC y el mensaje de alerta para determinar la acción a realizar. El tiempo que toma es de 50 milisegundos

para la Raspberry Pi 3B+ y para el nodo Z83 - F de 20 milisegundos. Este cálculo se lo puede realizar mediante cualquiera de los dos métodos que se pueden observar en la figura 51. Cabe recalcar que estos dos tiempos de 50 milisegundos para la Raspberry PI 3B+ y de 20 milisegundos para la minicomputadora Z83 - F corresponden al tiempo de procesamiento y no al tiempo de transmisión.

```

Método 1:
1 // Inicia el contador:
2 DateTime tiempo1 = DateTime.Now;
3
4 // Código del programa...
5
6 // Para el contador e imprime el resultado:
7 DateTime tiempo2 = DateTime.Now;
8 TimeSpan total = new TimeSpan(tiempo2.Ticks - tiempo1.Ticks);
9 Console.Write( "TIEMPO: " + total.ToString() );

Método 2:
1 // Inicia el contador:
2 Stopwatch tiempo = Stopwatch.StartNew();
3
4 // Código del programa...
5
6 // Para el contador e imprime el resultado:
7 Console.Write( "TIEMPO: " + tiempo.Elapsed.Seconds.ToString() );

```

Figura 51. Métodos para el cálculo de tiempo de procesamiento en IDE Visual Studio.

Una vez procesada la información y sabiendo el destino del vehículo de emergencia el último paso es el de enviar la alerta mediante LoRa a los otros nodos y actuadores para activar los semáforos en verde. En este caso en particular LoRa contiene una trama de datos con longitud máxima de 250 bytes y una tasa de transferencia hasta los 300 kbps en la que aplicando la misma ecuación 2 se obtiene un retardo de transmisión de $6,6 \times 10^{-3}$ segundos.

$$\text{tiempo total de transmisión con Raspberry PI} = TTb + TTr + TTL \quad (\text{Ecuación 4})$$

Donde:

TTb: Tiempo de transmisión bluetooth 4.0 beacon = $1,17 \times 10^{-5}$ (s).

TTr: Tiempo de transmisión tarjeta de red interna de script de Python a Visual Studio = $3,9 \times 10^{-7}$ (s).

TTI: Tiempo de transmisión de LoRa = $6,6 \times 10^{-3}$ (s).

tiempo total de transmisión con Z83 – F = $TTb + TTI$ (Ecuación 5)

Donde:

TTb: Tiempo de transmisión bluetooth 4.0 beacon = $1,17 \times 10^{-5}$ (s)

TTI: Tiempo de transmisión de LoRa = $6,6 \times 10^{-3}$ (s).

Una vez obtenido los distintos valores de retardo de transmisión se procede a sumar cada uno de ellos para obtener un total. El total de los tres valores en la Raspberry Pi 3B+ es de $6,61209 \times 10^{-3}$ segundos, mientras que para el nodo Z83 - F es de $6,617 \times 10^{-3}$

Por último, es necesario el cálculo de retardo de propagación mediante la siguiente ecuación.

$$\text{tiempo de propagación} = \frac{\text{distancia del enlace (m)}}{\text{velocidad de propagación} \left(\frac{\text{m}}{\text{s}}\right)} \quad (\text{Ecuación 6})$$

Para el cálculo del retardo de propagación es importante notar que el medio físico de transmisión es el aire con frecuencias dentro del espectro radioeléctrico. Por lo que para obtener la velocidad de propagación se utilizará el índice de refracción del aire de las leyes de Snell.

$$\text{velocidad de propagación} = \frac{\text{velocidad de la luz} \left(\frac{\text{m}}{\text{s}}\right)}{\text{índice de refracción}} \quad (\text{Ecuación 7})$$

Tomando como índice de refracción el aire ya que es el medio en el que se transmite las ondas, el resultado es de 299.704.764 m/s. Una vez obtenido la velocidad de propagación de las ondas en el medio en el que se los transmite se procede a calcular el retardo de propagación. En primera instancia, desde donde se recoge la alerta de emergencia es en el vehículo, por lo que el mensaje puede ser receptado hasta 100 metros de distancia. Con estos datos el resultado en el

enlace V2I es de $3,3 \times 10^{-7}$. Por otra parte, en LoRa los enlaces y distancia entre nodos de computación en la niebla estarán entre 2 o 3 kilómetros debido a que los RSU no tienen que estar tan apartados para la recepción de mensajes. Mediante la ecuación 3 y los datos de distancia de LoRa se obtiene 1×10^{-5} . El resultado total del retardo de propagación es de $1,03 \times 10^{-5}$. Cabe indicar que tanto para la Raspberry Pi 3B+ y la minicomputadora Z83 - F aplican los mismos cálculos en el tiempo de propagación como se puede apreciar en la ecuación 8.

tiempo total de propagación con Raspberri PI y Z83 – F = TPb + TPl

Donde: (Ecuación 8)

TPb: Tiempo de propagación bluetooth 4.0 beacon = $3,3 \times 10^{-7}$.

TPl: Tiempo de propagación de LoRa = 1×10^{-5} (s).

Una vez obtenido el retardo de transmisión, retardo de propagación y retardo de procesamiento de nodo, se suman los tiempos calculados para cada nodo y se obtiene el total dando como resultado 0,08664 segundos o 86,64 ms y para la minicomputadora Z83 - F es de 0,02662 segundos o 26,62 ms como se puede apreciar en la ecuación 9 y 10 respectivamente.

tiempo total de retardo con Raspberri PI = Tt + Tp + Tpr (Ecuación 9)

Donde:

Tt: Tiempo de transmisión total = $6,61209 \times 10^{-3}$ (s).

Tp: Tiempo de propagación total = $1,03 \times 10^{-5}$ (s).

Tpr: Tiempo de procesamiento total = 0,05 (s).

tiempo total de retardo Z83 – F = Tt + Tp + Tpr (Ecuación 10)

Donde:

Tt: Tiempo de transmisión total = $6,617 \times 10^{-3}$ (s).

Tp: Tiempo de propagación total = $1,03 \times 10^{-5}$ (s).

Tpr: Tiempo de procesamiento total = 0,02 (s).

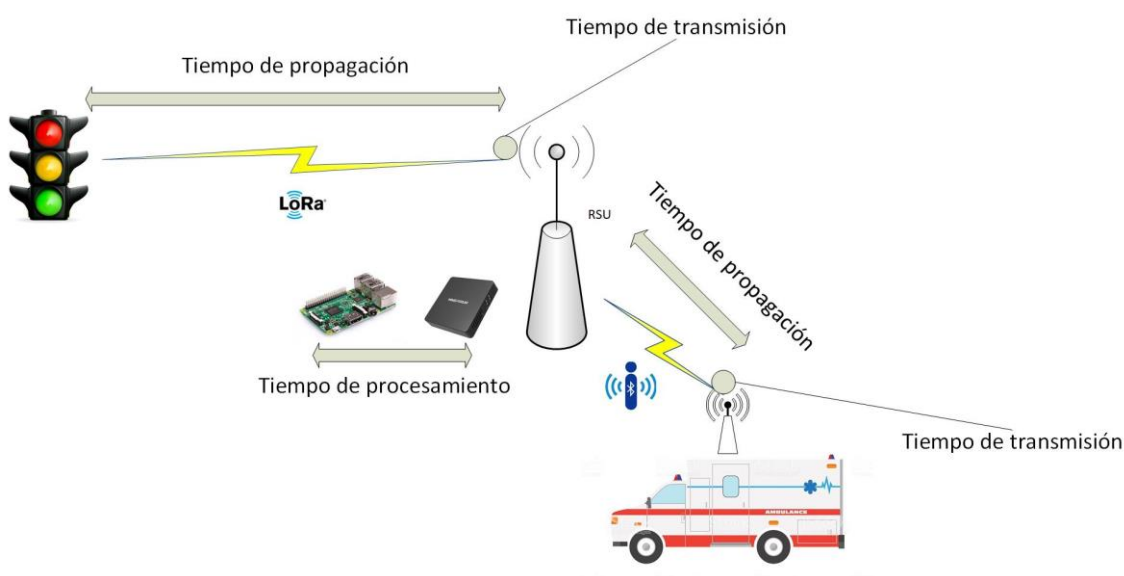


Figura 52. Cálculos de tiempo de retardo en topología de red de prototipo.

Como se puede observar en la figura 52, se calculó el tiempo de transmisión y de propagación en el enlace de V2I con tecnología Bluetooth 4.0. Así mismo el tiempo de transmisión y propagación del enlace de LoRa y el tiempo de procesamiento en los nodos de la computación en la niebla, siendo este en dos distintos casos si se utilizó un Raspberry Pi 3B+ o una mini PC Z83 - F.

Para el cálculo de computación en la nube el procedimiento es mas complejo debido a que se debe realizarlo mediante software que nos permita saber con cuanto retardo regresa el paquete. Debido a que no se puede saber con certeza cual será la tasa de transferencia de los enlaces por donde se transporte los datos. Tampoco la distancia y sobre todo que al transportar estos datos por una red de agregación y de core, no se puede saber si esta recorre un enlace

WDM, DWDM, SDH, entre muchas otras que conforman el core de la red que forma Internet.

Para este escenario de solución IoT basado en computación en la nube, el nodo de Z83 - F no realizará ningún procesamiento y reenviará la información lo más pronto a la WAN. Esto con el fin de que la información sea procesada en la computación en la nube. Para determinar los datos se utilizar el terminal en sistemas operativos basados en UNIX o CMD para computadoras Windows y se ejecuta el comando traceroute o traceroute.

Los resultados que arroja el comando traceroute son que el paquete para que solamente llegue a su destino es de aproximadamente 32 saltos.

```

sh-3.2# traceroute 52.55.218.230
traceroute to 52.55.218.230 (52.55.218.230), 64 hops max, 52 byte packets
 1  router.asus.com (192.168.1.1)  2.328 ms  1.217 ms  1.354 ms
 2  192.168.100.1 (192.168.100.1)  2.141 ms  2.421 ms  1.532 ms
 3  host-186-4-146-1.netlife.ec (186.4.146.1)  5.324 ms  4.151 ms  5.147 ms
 4  10.201.232.66 (10.201.232.66)  4.243 ms  4.681 ms  4.029 ms
 5  10.201.232.33 (10.201.232.33)  9.474 ms  7.401 ms  6.449 ms
 6  10.201.232.29 (10.201.232.29)  10.066 ms  29.726 ms  5.715 ms
 7  10.201.232.25 (10.201.232.25)  11.532 ms  5.912 ms  7.454 ms
 8  10.201.232.21 (10.201.232.21)  12.814 ms  7.598 ms  7.236 ms
 9  10.201.232.17 (10.201.232.17)  11.856 ms  6.640 ms  6.055 ms
10  10.201.232.1 (10.201.232.1)  4.145 ms  3.997 ms  6.290 ms
11  * * *
12  * * *
13  * * *
14  jax-b1-link.telia.net (62.115.58.205)  71.826 ms  71.250 ms
    ae53.bar1.orlando1.level3.net (4.35.250.141)  80.922 ms
15  mai-b1-link.telia.net (62.115.118.225)  73.148 ms  72.960 ms *
16  a100-ic-303623-mai-bl.c.telia.net (80.239.130.30)  73.949 ms
    4.16.98.206 (4.16.98.206)  100.014 ms  99.745 ms
17  52.93.37.117 (52.93.37.117)  81.982 ms
    52.93.37.115 (52.93.37.115)  73.587 ms
    52.93.37.105 (52.93.37.105)  83.596 ms
18  52.93.37.18 (52.93.37.18)  71.590 ms
    52.93.37.24 (52.93.37.24)  77.457 ms
    176.32.125.241 (176.32.125.241)  103.343 ms
19  * 54.239.42.192 (54.239.42.192)  97.453 ms *
    52.93.129.251 (52.93.129.251)  109.858 ms
    54.239.46.14 (54.239.46.14)  112.971 ms *
21  54.239.42.188 (54.239.42.188)  114.275 ms
    54.239.109.30 (54.239.109.30)  123.209 ms *
22  * 54.239.110.164 (54.239.110.164)  97.114 ms
    54.239.110.140 (54.239.110.140)  98.245 ms
23  54.239.110.161 (54.239.110.161)  99.613 ms
    54.239.110.171 (54.239.110.171)  106.135 ms
    54.239.108.216 (54.239.108.216)  124.958 ms
24  54.239.111.17 (54.239.111.17)  100.694 ms
    52.93.26.76 (52.93.26.76)  104.670 ms *
25  52.93.27.215 (52.93.27.215)  126.563 ms
    54.239.110.181 (54.239.110.181)  125.468 ms
    54.239.110.203 (54.239.110.203)  128.451 ms
26  52.93.24.29 (52.93.24.29)  99.647 ms
    52.93.24.13 (52.93.24.13)  99.731 ms
    205.251.244.89 (205.251.244.89)  98.072 ms
27  52.93.27.209 (52.93.27.209)  135.099 ms *
    52.93.27.213 (52.93.27.213)  139.326 ms
28  52.93.24.3 (52.93.24.3)  113.147 ms *
    72.21.197.17 (72.21.197.17)  113.007 ms
29  * * *
30  * * *
31  * * *
32  ec2-52-55-218-230.compute-1.amazonaws.com (52.55.218.230)  103.091 ms * 100.455 ms

```

Figura 54. Comando traceroute para determinar los saltos que realiza el paquete desde el nodo que recoge la alerta hasta la máquina virtual alojada en AWS.

Una vez determinado los saltos también se puede saber entre que valores oscila el tiempo de respuesta de extremo a extremo mediante el comando ping

enviando un paquete ICMP hacia el servidor. Se recibirá la respuesta mediante un ACK (*acknowledgment*) con el RTT.

```
[sh-3.2# ping 52.55.218.230
PING 52.55.218.230 (52.55.218.230): 56 data bytes
64 bytes from 52.55.218.230: icmp_seq=0 ttl=104 time=111.823 ms
64 bytes from 52.55.218.230: icmp_seq=1 ttl=104 time=111.526 ms
64 bytes from 52.55.218.230: icmp_seq=2 ttl=104 time=111.769 ms
64 bytes from 52.55.218.230: icmp_seq=3 ttl=104 time=114.121 ms
64 bytes from 52.55.218.230: icmp_seq=4 ttl=104 time=111.475 ms
64 bytes from 52.55.218.230: icmp_seq=5 ttl=104 time=111.441 ms
64 bytes from 52.55.218.230: icmp_seq=6 ttl=104 time=196.188 ms
64 bytes from 52.55.218.230: icmp_seq=7 ttl=104 time=113.008 ms
64 bytes from 52.55.218.230: icmp_seq=8 ttl=104 time=112.276 ms
64 bytes from 52.55.218.230: icmp_seq=9 ttl=104 time=113.841 ms
64 bytes from 52.55.218.230: icmp_seq=10 ttl=104 time=112.280 ms
64 bytes from 52.55.218.230: icmp_seq=11 ttl=104 time=117.729 ms
64 bytes from 52.55.218.230: icmp_seq=12 ttl=104 time=116.469 ms
64 bytes from 52.55.218.230: icmp_seq=13 ttl=104 time=113.335 ms
64 bytes from 52.55.218.230: icmp_seq=14 ttl=104 time=111.702 ms
64 bytes from 52.55.218.230: icmp_seq=15 ttl=104 time=111.439 ms
64 bytes from 52.55.218.230: icmp_seq=16 ttl=104 time=114.539 ms
64 bytes from 52.55.218.230: icmp_seq=17 ttl=104 time=111.464 ms
64 bytes from 52.55.218.230: icmp_seq=18 ttl=104 time=146.431 ms
64 bytes from 52.55.218.230: icmp_seq=19 ttl=104 time=111.249 ms
64 bytes from 52.55.218.230: icmp_seq=20 ttl=104 time=111.257 ms
```

Figura 55. Comando ICMP para determinar si al host al que se quiere alcanzar se encuentra activo con una respuesta mediante un ACK y su RTT.

Como se puede ver en la figura 34 el valor promedio es de 111 ms, lo cual en comparación con la computación en la niebla es notable y decididamente provechoso para la implementación de esta tecnología en soluciones IoT con bajo tiempo de respuesta.

Es importante determinar que uno de los factores que hace que el tiempo de respuesta se incremente al transportar datos a la computación en la nube, se debe al retardo de procesamiento en el nodo. Si bien el retardo de procesamiento en el nodo es el procesamiento de la información, no hay que olvidar que también se realiza ejecuciones en el plano de control y datos de los routers por los que fluye la información. Los routers de Core deben decidir y enrutar los datos por la mejor, por lo que en 32 saltos más el procesamiento de cada nodo es determinante en que se incremente el tiempo de respuesta.

5 Conclusiones y Recomendaciones

5.1 Conclusiones

Se concluye que la computación en la niebla no es una tecnología que reemplazará a la computación en la nube; por el contrario, es una extensión de esta para que las dos puedan trabajar conjuntamente y reduzcan los tiempos de respuesta al procesar localmente la información crítica que requiera acción temprana.

El procesamiento más cercano a la capa de sensores en IoT tiene como consecuencia la reducción de los tiempos de respuesta y sobre todo no saturar innecesariamente el enlace de red hacia la computación en la nube; debido a que solamente se envía la información que necesite ser analizada de manera más meticulosa.

Es evidente que las urbes con STI realizan esfuerzos para la automatización de semáforos en sus intersecciones con el objetivo de disminuir el tráfico vehicular en las zonas urbanas. Sin embargo, aún no existen soluciones concretas para que los vehículos tanto como ambulancias, bomberos y policía puedan reducir su tiempo en atender una emergencia.

En el levantamiento de información de las urbes con STI, solo la ciudad de Pittsburgh realiza el procesamiento localmente en las intersecciones semaforizadas; no obstante, solo cumple ciertos requerimientos para poder catalogarla como una tecnología de computación en la niebla.

Muchas de las aplicaciones IoT que se implementa en las ciudades inteligentes, no todas cuentan aun con una infraestructura de computación en la niebla y tampoco redes de área extendida de bajo consumo; las cuales ahora son imprescindibles en una arquitectura IoT.

Los resultados obtenidos en los tiempos de respuesta de la computación en la niebla y computación en la nube son una aproximación debido a que se realiza en un ambiente de simulación y principalmente también se debe a que los nodos

solamente procesan una aplicación en específico cuando en realidad un nodo de la computación en la niebla tiene la capacidad de procesar distintas solicitudes de aplicaciones IoT.

Es de mucha importancia los RSU en una aplicación IoT vehicular para la recolección de datos de los sensores que se encuentran equipados en los vehículos y así poder atender los distintos requerimientos de una aplicación IoT.

5.2 Recomendaciones

Para la obtención de mejores resultados se recomienda realizar las pruebas en un ambiente de simulación distinto en el que se pueda experimentar con vehículos e instalar los nodos de la computación en la niebla en intersecciones, emplazamientos y con una separación de varios kilómetros.

A este tipo de aplicaciones IoT en los STI, es importante tomar en cuenta si es necesaria la implementación de *machine learning* en los nodos de la computación en la niebla para que los mismos puedan aprender cual será el mejor camino según datos estadísticos de tráfico en tiempo real que puedan ser recabados y analizados a través de herramientas de Big Data.

Se recomienda que los nodos de la computación en la niebla que se encarguen de enviar la información a la a través del backbone de la red hacia la computación en la nube, lo realicen mediante telefonía celular con las nuevas tecnologías propietarias de LPWAN denominadas LTE M o NB-IoT.

Para el envío de la información a la computación en la nube es necesario tomar en cuenta utilizar protocolos de IoT en la capa de aplicación como lo son MQTT o CoAP para mejorar aún más los tiempos de respuesta hacia los actuadores.

REFERENCIAS

- Aazam, M., Zeadally, S., & Harras, K. A. (2018). Fog Computing Architecture, Evaluation, and Future Research Directions. *IEEE Communications Magazine*, 56(5), 46–52. <https://doi.org/10.1109/MCOM.2018.1700707>
- Abdollah, T., & Perrone, M. (2017). Hackers could remotely control defibrillator or pacemaker, U.S. warns. Recuperado el 20 de octubre de 2018 de <https://www.cbc.ca/news/world/cybersecurity-heart-devices-implantable-1.3930997>
- Ahmad, R. W., Gani, A., Hamid, S. H. A., Shiraz, M., Yousafzai, A., & Xia, F. (2015). A survey on virtual machine migration and server consolidation frameworks for cloud data centers. *Journal of Network and Computer Applications*, 52, 11–25. <https://doi.org/10.1016/j.jnca.2015.02.002>
- Alleven, M. (2018). FCC tests show Wi-Fi can share with DSRC in 5.9 GHz band. Recuperado el 8 de noviembre de 2018 de <https://www.fiercewireless.com/wireless/fcc-tests-show-wi-fi-can-share-dsrc-5-9-ghz-band>
- Alvarez, R. (2016). El hackeo no tiene límites y el nuevo blanco son las bombas de insulina. Recuperado el 20 de octubre de 2018 de <https://www.xataka.com/seguridad/el-hackeo-no-tiene-limites-y-el-nuevo-blanco-son-las-bombas-de-insulina>
- Alvarez, R. (2017). Casi medio millón de personas en peligro debido a una vulnerabilidad de seguridad en marcapasos “St Jude Medical”. Recuperado el 20 de octubre de 2018 de <https://www.xataka.com/seguridad/casi-medio-millon-de-personas-en-peligro-debido-a-una-vulnerabilidad-de-seguridad-en-marcapasos-st-jude-medical>

- Amazon Web Services. (2018). Infraestructura en la nube global | Regiones y zonas de disponibilidad. Recuperado el 17 de octubre de 2018 de <https://aws.amazon.com/es/about-aws/global-infrastructure/>
- Bardyn, J.-P., Melly, T., Seller, O., & Sornin, N. (2016). IoT: The era of LPWAN is starting now. En ESSCIRC Conference 2016: 42nd European Solid-State Circuits Conference (pp. 25–30). IEEE. <https://doi.org/10.1109/ESSCIRC.2016.7598235>
- Bolić, M., Simplot-Ryl, D., & Stojmenović, I. (Eds.). (2010). RFID Systems (1.a ed.). Chichester: John Wiley & Sons, Ltd. <https://doi.org/10.1002/9780470665251>
- Chen, D. (2016). How to Use Bluetooth 4.0 HM10: 4 Steps (with Pictures). Recuperado el 4 de diciembre de 2018 de <https://www.instructables.com/id/How-to-Use-Bluetooth-40-HM10/>
- Cisco. (2016). IoE and Fog Computing. Mexico. Recuperado de [https://ciscolive.cisco.com/on-demand-library/?search=fog computing#/video/1533848757581001u81H](https://ciscolive.cisco.com/on-demand-library/?search=fog%20computing#/video/1533848757581001u81H)
- Comisión Federal de Comunicaciones. (1999). FCC Allocates Spectrum 5.9 GHz Range for Intelligent Transportation Systems Uses. Recuperado el 5 de noviembre de 2018 de https://transition.fcc.gov/Bureaus/Engineering_Technology/News_Releases/1999/nret9006.html
- Comité Europeo de Normalización. (2015). Traffic control equipment - Signal heads (Standard EN 12368).
- Conner, F. (2018). The X-factor in our driverless future: V2V and V2I. Recuperado el 11 de noviembre de 2018 de <https://www.zdnet.com/article/the-x-factor-in-our-driverless-future-v2v-and-v2i/>

- Dimitrakopoulos, G., & Demestichas, P. (2010). Intelligent transportation systems: Systems based on cognitive networking principles and management functionality. *IEEE Vehicular Technology Magazine*, 5(1), 77–84. <https://doi.org/10.1109/MVT.2009.935537>
- DSA. (2018). Open Source IoT Platform & Toolkit. Recuperado el 4 de enero de 2019 de <http://iot-dsa.org/get-started/how-dsa-works>
- Google Cloud. (2018). Ubicaciones mundiales: regiones y zonas. Recuperado el 18 de octubre de 2018 de <https://cloud.google.com/about/locations/>
- Guru99. (2013). What is .NET Framework? Complete Architecture Tutorial. Recuperado el 4 de enero de 2019 de <https://www.guru99.com/net-framework.html>
- Hekmat, R. (2006). *Ad-hoc Networks: Fundamental Properties and Network Topologies*. Springer Netherlands. <https://doi.org/10.1007/1-4020-5166-2>
- Hetting, C. (2018). FCC's Jessica Rosenworcel reopens case for Wi-Fi in 5.9 GHz band. Recuperado el 8 de noviembre de 2018 de <https://wifinowevents.com/news-and-blog/fccs-jessica-rosenworcel-reopens-case-for-wi-fi-in-5-9-ghz-band/>
- IEEE Standards Association. (2018). New IEEE 1934™ Standard Delivers Framework for Developing Applications and Business Models Enabled by Fog Computing. Recuperado el 20 de septiembre de 2018 de <https://standards.ieee.org/news/2018/ieee1934-standard-fog-computing.html>
- IMPINJ. (2017). Different Types of RFID Systems. Recuperado el 8 de noviembre de 2018 de <https://www.impinj.com/about-rfid/types-of-rfid-systems/>
- INRIX. (2017). INRIX Global Traffic Scorecard. Recuperado el 13 de noviembre de 2018 de <http://inrix.com/scorecard/>

- iStock. (2016). Recuperado el 6 de noviembre de 2018 de <https://www.istockphoto.com/es/vector/comunicación-inalámbrica-de-los-vehículos-y-las-señales-peatones-sistema-de-gm690704442-127313235>
- Jeon, K. E., She, J., Soonsawad, P., & Ng, P. C. (2018). BLE Beacons for Internet of Things Applications: Survey, Challenges, and Opportunities. *IEEE Internet of Things Journal*, 5(2), 811–828. <https://doi.org/10.1109/JIOT.2017.2788449>
- Kang, C. (2017). Pittsburgh Welcomed Uber's Driverless Car Experiment. Not Anymore. Recuperado el 11 de noviembre de 2018 de <https://www.nytimes.com/2017/05/21/technology/pittsburgh-ubers-driverless-car-experiment.html>
- Latif, S., & Zafar, N. A. (2017). A survey of security and privacy issues in IoT for smart cities. En 2017 Fifth International Conference on Aerospace Science & Engineering (ICASE) (pp. 1–5). IEEE. <https://doi.org/10.1109/ICASE.2017.8374288>
- Lea, P. (2018). *Internet of Things for Architects* (1.a ed.). Birmingham: Packt Publishing.
- Lee, H.-C., & Ke, K.-H. (2018). Monitoring of Large-Area IoT Sensors Using a LoRa Wireless Mesh Network System: Design and Evaluation. *IEEE Transactions on Instrumentation and Measurement*, 67(9), 2177–2187. <https://doi.org/10.1109/TIM.2018.2814082>
- Lin, Y., Wang, P., & Ma, M. (2017). Intelligent Transportation System(ITS): Concept, Challenge and Opportunity. En 2017 IEEE 3rd International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing, (HPSC) and IEEE International Conference on Intelligent Data and

Security (IDS) (pp. 167–172). IEEE.
<https://doi.org/10.1109/BigDataSecurity.2017.50>

Long, X., He, J., Zhou, J., Fang, L., Zhou, X., Ren, F., & Xu, T. (2015). A review on light-emitting diode based automotive headlamps. *Renewable and Sustainable Energy Reviews*, 41, 29–41.
<https://doi.org/10.1016/j.rser.2014.08.028>

Mare, R. M., Luiz Marte, C., & Cugnasca, C. E. (2016). Visible Light Communication Applied to Intelligent Transport Systems: an Overview. *IEEE Latin America Transactions*, 14(7), 3199–3207.
<https://doi.org/10.1109/TLA.2016.7587621>

Martinez, A. (2017). ¿Semáforos de Quito están sincronizados? Recuperado el 12 de noviembre de 2018 de <https://www.metroecuador.com.ec/ec/noticias/2017/06/02/semaforos-quito-estan-sincronizados.html>

Metro Ecuador. (2016). Infracciones de tránsito en Quito han reducido en 41%. Recuperado el 12 de noviembre de 2018 de <https://www.metroecuador.com.ec/ec/noticias/2016/12/21/infracciones-transito-quito-reducido-41.html>

Mittal, A. K., & Bhandari, D. (2013). A novel approach to implement green wave system and detection of stolen vehicles. En 2013 3rd IEEE International Advance Computing Conference (IACC) (pp. 1055–1059). IEEE. <https://doi.org/10.1109/IAdCC.2013.6514372>

Mouradian, C., Naboulsi, D., Yangui, S., Glitho, R. H., Morrow, M. J., & Polakos, P. A. (2018). A Comprehensive Survey on Fog Computing: State-of-the-Art and Research Challenges. *IEEE Communications Surveys and Tutorials*, 20(1), 416–464.
<https://doi.org/10.1109/COMST.2017.2771153>

- Mukherjee, M., Shu, L., & Wang, D. (2018). Survey of Fog Computing: Fundamental, Network Applications, and Research Challenges. *IEEE Communications Surveys & Tutorials*, 20(3), 1826–1857. <https://doi.org/10.1109/COMST.2018.2814571>
- New York City DOT. (2018a). Connected Vehicle technology is coming to the streets of New York City!. Recuperado el 14 de noviembre de 2018 de <https://www.cvp.nyc/>
- New York City DOT. (2018b). Project Scope. Recuperado el 13 de noviembre de 2018 de <https://www.cvp.nyc/project-scope>
- Nordrum, A. (2016). The internet of fewer things [News]. *IEEE Spectrum*, 53(10), 12–13. <https://doi.org/10.1109/MSPEC.2016.7572524>
- OpenFog Consortium Architecture Working Group. (2017). OpenFog Reference Architecture for Fog Computing. OpenFogConsortium, (February), 1–162. <https://doi.org/OPFRA001.020817>
- Orozco, A. M., Michoud, R., & Llano, G. (2013). Routing Protocols simulation for Efficiency Applications in Vehicular Environments. *Sistemas y Telemática*, 11(27), 27. <https://doi.org/10.18046/syt.v11i27.1693>
- Ossorio Miguel. (2018). Llegan los coches que hablan (contigo, entre ellos y con el entorno). Recuperado el 6 de noviembre de 2018 de <https://www.media-tics.com/noticia/8105/tecnologias-emergentes/llegan-los-coches-que-hablan-contigo-entre-ellos-y-con-el-entorno.html>
- Pastor, J. (2018). “La solución es la luz” Así explican LiFi quienes están detrás de la prometedora alternativa a WiFi. Recuperado el 9 de noviembre de 2018 de <https://www.xataka.com/investigacion/la-solucion-es-la-luz-asi-explican-lifi-quienes-estan-detras-de-la-prometedora-alternativa-a-wifi>

- Rajagopal, S., Roberts, R., & Lim, S.-K. (2012). IEEE 802.15.7 visible light communication: modulation schemes and dimming support. *IEEE Communications Magazine*, 50(3), 72–82. <https://doi.org/10.1109/MCOM.2012.6163585>
- Rapid Flow. (2016). Surtrac Adaptive Traffic Signal Control System. Recuperado el 11 de noviembre de 2018 de <https://www.rapidflowtech.com/surtrac/what-is-surtrac>
- Raspberry Org. (2018). What is a Raspberry Pi? Recuperado el 4 de diciembre de 2018 de <https://www.raspberrypi.org/help/what-is-a-raspberry-pi/>
- Saad, A. A., Zouka, H. A. El, & Al-Soufi, S. A. (2016). Secure and Intelligent Road Traffic Management System Based on RFID Technology. En 2016 World Symposium on Computer Applications & Research (WSCAR) (pp. 41–46). IEEE. <https://doi.org/10.1109/WSCAR.2016.9>
- Sabih ur, R. hman, Khan, A., Zia, T., & Zheng, L. (2013). Vehicular Ad-Hoc Networks (VANETs) - An Overview and Challenges. *Wireless Networking and Communications*, 3(2), 29-38. <https://doi.org/10.5923/j.jwnc.20130303.02>
- Sairam, K. V. S. S. S. S., Gunasekaran, N., & Redd, S. R. (2002). Bluetooth in wireless communication. *IEEE Communications Magazine*, 40(6), 90–96. <https://doi.org/10.1109/MCOM.2002.1007414>
- Santos, J., Leroux, P., Wauters, T., Volckaert, B., & De Turck, F. (2018). Anomaly detection for Smart City applications over 5G low power wide area networks. En NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium (pp. 1–9). IEEE. <https://doi.org/10.1109/NOMS.2018.8406257>

- Santos Sara. (2018). Raspberry Pi 3 Model B+ Review - What's New? Recuperado el 4 de diciembre de 2018 de <https://makeradvisor.com/raspberry-pi-3-model-b-plus-review/>
- Shen, W.-H., & Tsai, H.-M. (2017). Testing vehicle-to-vehicle visible light communications in real-world driving scenarios. En 2017 IEEE Vehicular Networking Conference (VNC) (pp. 187–194). IEEE. <https://doi.org/10.1109/VNC.2017.8275596>
- Shiff, L. (2018). Gartner Magic Quadrant for Cloud Infrastructure as a Service 2018. Recuperado el 17 de diciembre de 2018 de <https://www.bmc.com/blogs/gartner-magic-quadrant-cloud-iaas/>
- Snow, J. (2017). This AI traffic system in Pittsburgh has reduced travel time by 25%. Recuperado el 12 de noviembre de 2018 de <https://www.smartcitiesdive.com/news/this-ai-traffic-system-in-pittsburgh-has-reduced-travel-time-by-25/447494/>
- The OpenFog Consortium. (2018). OpenFog Consortium. Recuperado el 4 de enero de 2019 de <https://www.openfogconsortium.org/>
- Unión Internacional de las Telecomunicaciones. (2011). Intelligent transport systems – Guidelines and objectives (Rec. ITU-R M.1890). Recuperado el 10 de noviembre de 2018 de https://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.1890-0-201104-!!!PDF-E.pdf
- United States Department of Transportation. (2017). Intelligent Transportation Systems - Connected Vehicle Basics. Recuperado el 11 de noviembre de 2018 de https://www.its.dot.gov/cv_basics/cv_basics_20qs.htm
- Universidad Militar Nueva Granada, O., & Llano Ramírez, G. (2014). Ciencia e ingeniería neogranadina. Ciencia e Ingeniería Neogranadina (Vol. 24). Universidad Militar Nueva Granada. Recuperado de

http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0124-81702014000200007

Verge. (2018). Z83-F MINI PC Review. Recuperado el 4 de enero de 2019 de <https://pcverge.com/z83-f-fanless-mini-pc-review/>

Yaqoob, I., Ahmed, E., Hashem, I. A. T., Ahmed, A. I. A., Gani, A., Imran, M., & Guizani, M. (2017). Internet of Things Architecture: Recent Advances, Taxonomy, Requirements, and Open Challenges. *IEEE Wireless Communications*, 24(3), 10–16. <https://doi.org/10.1109/MWC.2017.1600421>

Universidad de Navarra, (2019). *Juicio de Valor*. Recuperado el 19 de octubre de 2018 de <https://www.cun.es/diccionario-medico/terminos/juicio-valor>

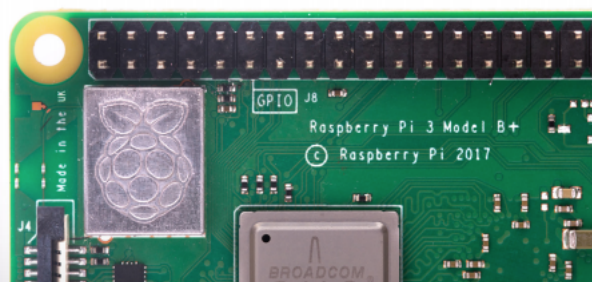
Villaverde, I., (2008), *La resolución de conflictos entre derechos fundamentales. El principio de proporcionalidad*, Quito, Ecuador: Ministerio de Justicia y Derechos Humanos

ANEXOS

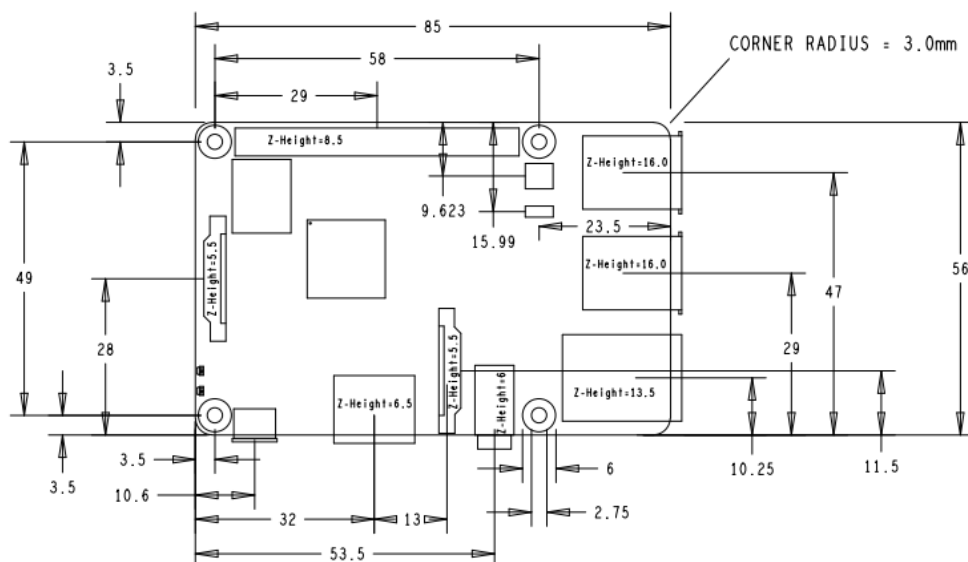
Anexo 1. Raspberry Pi 3B+ especificaciones.

Specifications

Processor:	Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.4GHz
Memory:	1GB LPDDR2 SDRAM
Connectivity:	<ul style="list-style-type: none"> ■ 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE ■ Gigabit Ethernet over USB 2.0 (maximum throughput 300Mbps) ■ 4 × USB 2.0 ports
Access:	Extended 40-pin GPIO header
Video & sound:	<ul style="list-style-type: none"> ■ 1 × full size HDMI ■ MIPI DSI display port ■ MIPI CSI camera port ■ 4 pole stereo output and composite video port
Multimedia:	H.264, MPEG-4 decode (1080p30); H.264 encode (1080p30); OpenGL ES 1.1, 2.0 graphics
SD card support:	Micro SD format for loading operating system and data storage
Input power:	<ul style="list-style-type: none"> ■ 5V/2.5A DC via micro USB connector ■ 5V DC via GPIO header ■ Power over Ethernet (PoE)-enabled (requires separate PoE HAT)
Environment:	Operating temperature, 0–50 °C
Compliance:	For a full list of local and regional product approvals, please visit www.raspberrypi.org/products/raspberry-pi-3-model-b+
Production lifetime:	The Raspberry Pi 3 Model B+ will remain in production until at least January 2023.



Physical specifications



Warnings

- This product should only be connected to an external power supply rated at 5V/2.5 A DC. Any external power supply used with the Raspberry Pi 3 Model B+ shall comply with relevant regulations and standards applicable in the country of intended use.
- This product should be operated in a well-ventilated environment and, if used inside a case, the case should not be covered.
- Whilst in use, this product should be placed on a stable, flat, non-conductive surface and should not be contacted by conductive items.
- The connection of incompatible devices to the GPIO connection may affect compliance, result in damage to the unit, and invalidate the warranty.
- All peripherals used with this product should comply with relevant standards for the country of use and be marked accordingly to ensure that safety and performance requirements are met. These articles include but are not limited to keyboards, monitors, and mice when used in conjunction with the Raspberry Pi.
- The cables and connectors of all peripherals used with this product must have adequate insulation so that relevant safety requirements are met.

Safety instructions

To avoid malfunction of or damage to this product, please observe the following:

- Do not expose to water or moisture, or place on a conductive surface whilst in operation.
- Do not expose to heat from any source; the Raspberry Pi 3 Model B+ is designed for reliable operation at normal ambient temperatures.
- Take care whilst handling to avoid mechanical or electrical damage to the printed circuit board and connectors.
- Whilst it is powered, avoid handling the printed circuit board, or only handle it by the edges to minimise the risk of electrostatic discharge damage.



raspberrypi.org

Anexo 2. Semtech sx1278 datasheet.

1. Introduction

The purpose of this application note is to assist the engineer with the selection of a suitable crystal oscillator for the LoRa® modulation family of long-range wireless ISM transceiver devices.

It is recommended to read this application note in conjunction with Semtech Application Note AN1200.07, "Improving the Accuracy of a Crystal Oscillator" [1], for further information on crystal oscillator specifications and parameters.

2. LoRa® Modulation

The LoRa® modulation is a spread-spectrum technique that uses wideband linear frequency modulated pulses to encode information, whose frequency increases or decreases over a certain amount of time.

As with other spread-spectrum modulation techniques, LoRa® uses the entire channel bandwidth to broadcast a signal, making it robust to channel noise. In addition, because LoRa® modulation uses a broad band of the spectrum, it is also resistant to long term relative frequency error, multi-path fading and Doppler effects.

3. LoRa® Modulation Crystal Specification

The crystal specification for LoRa® modulation is given in Table 1. It can be observed that the crystal specification for Semtech's family of LoRa® modulation transceivers is similar to those of existing Semtech FSK ISM transceivers. Since the internal oscillator drive circuitry of Semtech transceivers are similar, no special IP is required to condition the oscillator.

Table 1: Typical Crystal Specification

Symbol	Description	Conditions	Min	Typ	Max	Unit
F _{XOSC}	Crystal Frequency		26	32	32	MHz
R _S	Crystal Series Resistance	For SX1272 ¹ and SX1276	-	30	40	Ω
R _S	Crystal Series Resistance	For SX1276 ² only	-	30	100	Ω
C ₀	Crystal Shunt Capacitance		-	2.8	7	pF
C _{FOOT}	External Foot Capacitance	On each pin XTA and XTB	8	15	22	pF
C _{LOAD}	Crystal Load Capacitance		6	-	12	pF
F _{TOL}	Initial Frequency Tolerance		-	±10	±30	ppm
F _{TEMP}	Temp. Characteristics	Application specific	-	-	-	ppm
F _{AGING}	Aging Characteristics	Application specific	-	-	-	ppm
G _S	Acceleration Sensitivity	Application specific ³	-	2	-	ppb/G

There are two crystal specifications that warrant further analysis:

- Frequency Tolerance
- G or Acceleration Sensitivity

3.1. Frequency Tolerance

The frequency or calibration tolerance, expressed in ppm, is typically an application-specific parameter.

Usually modulation techniques offering sensitivity performances similar to LoRa® such as narrow-band FHSS or high spreading factor DSSS typically require a crystal oscillator tolerance of only a few ppm to ensure both frequency and symbol rate accuracy.

As previously highlighted, the LoRa® modulation technique is impervious to the relative initial frequency error (and subsequently symbol rate tolerance) between the transmitter and the receiver.

This immunity to both frequency and symbol tolerance is illustrated in Figure 1.

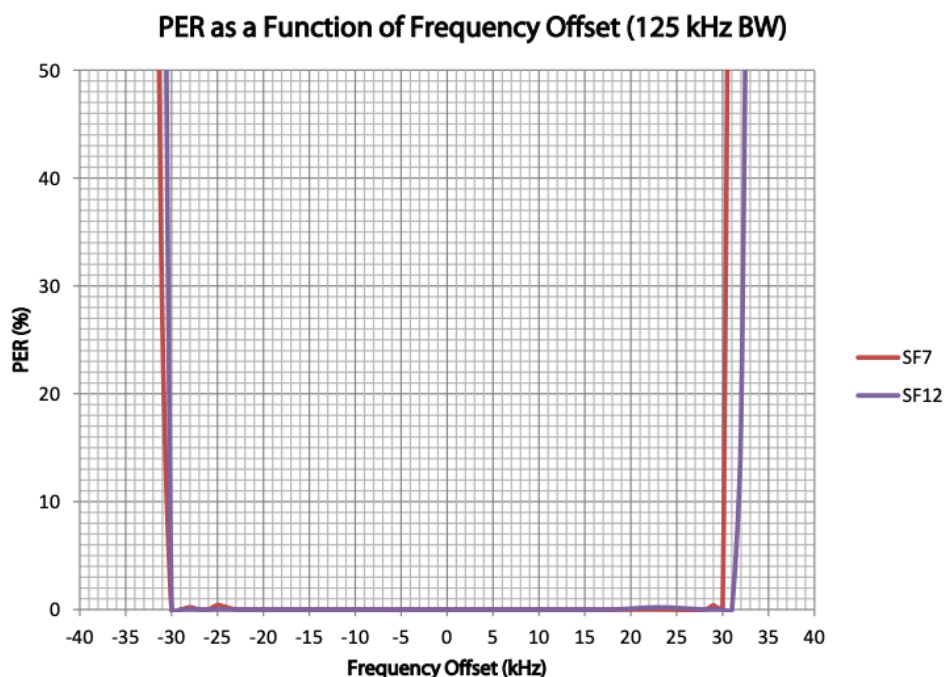


Figure 1: LoRa® PER versus Relative Crystal Oscillator Frequency Offset

The figure indicates that frequency tolerances of typically $\pm 25\%$ of the LoRa® modulation bandwidth can be withstood and still maintain a 10% PER link. This compares favorably with current high-link budget systems.

3.2. G – Acceleration Sensitivity

G or acceleration sensitivity is a measurement of the sensitivity of a crystal oscillator to acceleration, and describes a short-term or instantaneous frequency error.

A crystal oscillator subject to acceleration or mechanical shock has a slightly different series resonant frequency than the same oscillator experiencing no acceleration. It has been observed that the magnitude of the frequency shift is proportional to both the magnitude and the direction of the acceleration relative to a coordinate system applied to the crystal. [2]

A representation of this effect can be considered in the case of a crystal subject to a cycle of acceleration at a rate of f_v :

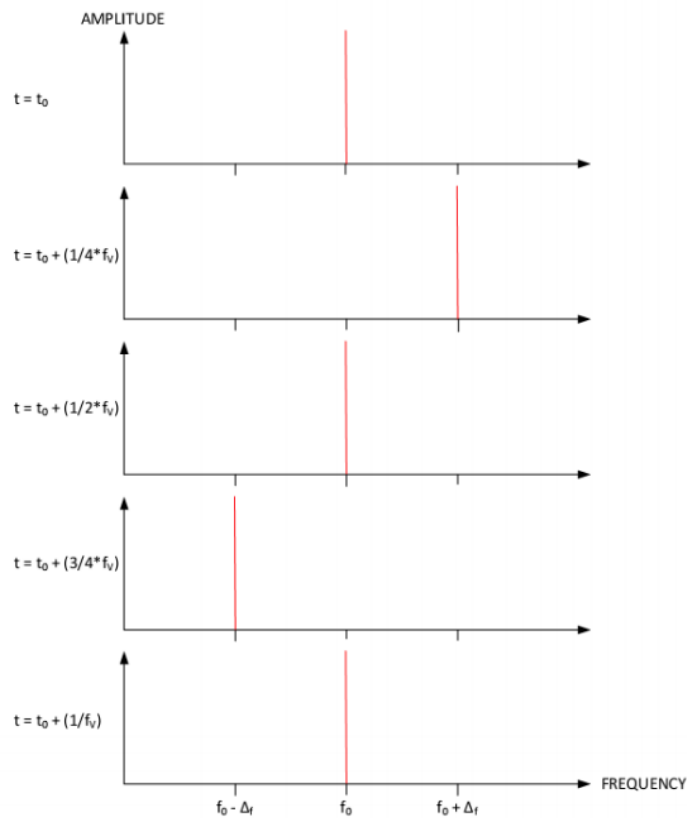


Figure 2: Instantaneous Frequency during one Vibration Cycle

Each plot shows the instantaneous output frequency sampled at a time period $n/4 * f_c$. If we consider the rate of acceleration to be sinusoidal, it can be observed that the instantaneous frequency deviation occurs at the crest and trough of the applied sinusoidal force.

It can be shown that the magnitude of the instantaneous frequency deviation is proportional to the instantaneous amplitude of the acceleration. [2]

This can be viewed practically by observing the effect of acceleration or shock on the spectrum of a phase-locked loop carrier implemented with a crystal oscillator that does not have a low-G sensitivity crystal specified, as illustrated in Figure 3.

Applying an acceleration force to the crystal causes an instantaneous change in frequency. The apparent amplitude of the instantaneous frequency is limited only by the measurement instrumentation.

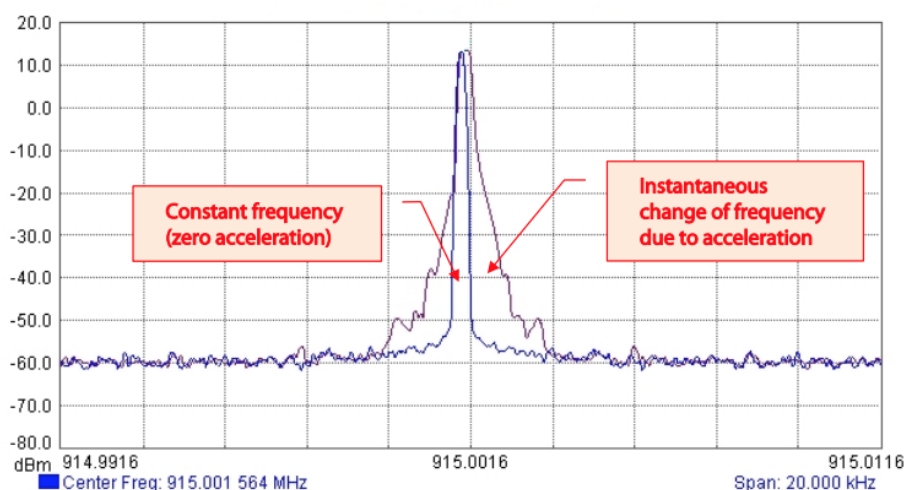
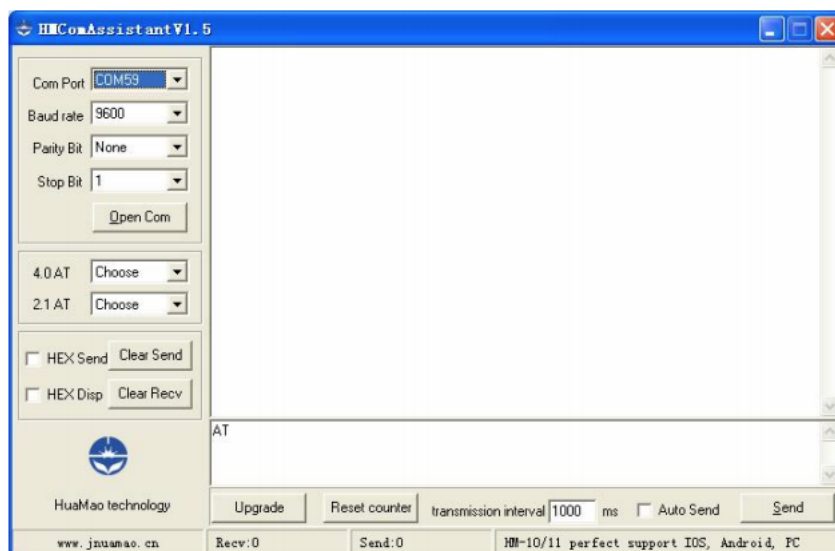


Figure 3: Effect of Acceleration due to Shock on a Crystal-Referenced PLL Transmitted Carrier

Anexo 3. Comandos HM-10 Beacon



On Transmit version: Only accept AT Command from UART interface when Bluetooth device is not connected with remote device.

On Remote version: Can accept AT Command from UART interface when Bluetooth Device is not connected with remote device, Also can accept AT Command from remote Bluetooth device when connected that.

On PIO collection version: Only accept AT Command from UART interface when Bluetooth device is not connected with remote device.

Bluetooth V2.1 version Command is not here, please download datasheet from http://www.jhuamao.cn/Bluetooth_en.rar

1. Test Command

Send	Receive	Parameter
AT	OK OK+LOST	None

If Module is not connected to remote device will receive: "OK"

If Module has connected, module will disconnected from remote device, if "AT + NOTI" is setup to 1, will receive: "OK+LOST"

2. Query module address

Send	Receive	Parameter
AT+ADDR?	OK+ADDR:MAC Address	None

3. Query/Set Advertising interval

Send	Receive	Parameter
AT+ADVI?	OK+ Get:[Para]	None
AT+ADVI[Para]	OK+ Set:[Para]	Para: 0 ~ F 0: 100ms 1: 152.5 ms 2: 211.25 ms 3: 318.75 ms 4: 417.5 ms 5: 546.25 ms 6: 760 ms 7: 852.5 ms 8: 1022.5 ms 9: 1285 ms A: 2000ms B: 3000ms C: 4000ms D: 5000ms E: 6000ms F: 7000ms Default: 9 HMSoft Default: 0 HMSensor Default: 9

The maximum 1285ms recommendations from the IOS system. That is to say, 1285ms is apple allowed, but in response to scan and connected all the time will be long.

This command is added since V517 version.

V522 allow max value F.

4. Query/Set Advertising Type

Send	Receive	Parameter
AT+ADTY?	OK+ Get:[Para]	None
AT+ADTY[Para]	OK+ Set:[Para]	Para: 0 ~ 3 0: Advertising ScanResponse, Connectable 1: Only allow last device connect in 1.28 seconds 2: Only allow Advertising and ScanResponse. 3: Only allow Advertising Default: 0

Added since V519

5. Query/Set ANCS switch

Send	Receive	Parameter
AT+ANCS?	OK+ Get:[Para]	None
AT+ANCS[Para]	OK+ Set:[Para]	Para: 0 ~ 1 0: Off 1: On Default: 0

Note1: This command added in V524.

Note2: Please send AT+RESET to restart module if you set value 1.

Note3: Must execute AT+TYPE3 first.

6. Query/Set whitelist switch

Send	Receive	Parameter
AT+ALLO?	OK+ Get:[Para]	None

AT+ALLO[Para]	OK+ Set:[Para]	Para: 0 ~ 1 0: Off 1: On Default: 0
---------------	----------------	--

Note1: This command added in V523.

Note2: Whitelist allow three mac address link to module. Please use AT+AD command set whitelist mac address.

7. Query/Set whitelist mac address

Send	Receive	Parameter
AT+AD[para1]??	OK+ Get:[Para2]	None
AT+ALLO[Para1][Para2]	OK+ Set:[Para2]	Para1: 1, 2, 3 Para2: MAC address Para1 default: 0

Note1: This command added in V523.

E.g.

Send AT+ID1001122334455

Recv OK+Set:001122334455

8. Query/Set battery monitor switch

Send	Receive	Parameter
AT+BATC?	OK+ Get:[Para]	None
AT+BATC[Para]	OK+ Set:[Para]	Para: 0 ~ 1 0: Off 1: On Default: 0

This command added in V520

9. Query battery information

Send	Receive	Parameter
AT+BATT?	OK+BATT:[Para]	Para: 000~100

There has three ways to get battery information:

10. Set iBeacon into service mode(*)

Send	Receive	Parameter
AT+BUSHU	OK+BUSHU	

This command is added in V520, Removed in V521, Please use AT+DELO

This command set iBeacon into service mode until next power on.

In service mode, module not allow any link request.

BUSHU is Chinese spelling, meaning the deployment.

Note: Should to open iBeacon switch first (AT+IBEA).

11. Query/Set Bit format

Send	Receive	Parameter
AT+BIT7?	OK+Get:[para1]	Para1: bit7 switch.
AT+BIT7[para1]	OK+Set:[para1]	0----Not compatible 1----Compatible Default: 0

This command is used only for compatible uses 7 data bits, 2 stop bit device.

12. Query/Set baud rate

HM Bluetooth module datasheet

Send	Receive	Parameter
AT+BAUD?	OK+Get:[para1]	Para1: Baud rate No.
AT+BAUD[para1]	OK+Set:[para1]	0-----9600 1-----19200 2-----38400 3-----57600 4-----115200 5-----4800 6-----2400 7-----1200 8-----230400 Default: 0(9600)

e.g.

Query baud:

Send: AT+BAUD?

Receive: OK+Get:0

Setup baud:

Send: AT+BAUD1

Receive: OK+Set:1

Note: If setup to Value 7, After next power on, module will not support any AT Commands, until PIO0 is pressed, Module will change Baud to 9600.

13. Query/Set Characteristic

Send	Receive	Parameter
AT+CHAR?	OK+Get:[para1]	Para1: 0x0001~0xFFFFE
AT+CHAR[para1]	OK+Set:[para1]	Default: 0xFFE1

e.g. change characteristic value to 0xAAA0

Send: AT+CHAR0xAAA0

Recv: OK+Set:0xAAA0

14. Clear Last Connected device address

Send	Receive	Parameter
AT+CLEAR	OK+CLEAR	None

Notice: Only Central role is used.

15. Try connect to last succeeded device

Send	Receive	Parameter
AT+CONN	OK+CONN[Para1]	Para1: L, E, F, N L: Connecting E: Connect error F: Connect Fail N: No Address

Notice: Only Central role is used.

If remote device has already connected to other device or shut down, "OK+CONN" will received after about 10 seconds.

16. Try connect an address

Send	Receive	Parameter
AT+CON[Para1]	OK+CONN[Para2]	Para1: Address Like: 0017EA090909 Para2: A, E, F A: Connecting E: Connect error F: Connect Fail

Notice: Only central role is used.

If remote device has already connected to other device or shut down, "OK+CONN" will received after about 10 Seconds.

e.g.

Try to connect an device which MAC address is 00:17:EA:09:09:09

Send: AT+CON0017EA090909

May receive a reply:

OK+CONNA ===== Accept request, connecting

