

Análisis del rendimiento de OpenBTS como solución de red viable para dispositivos IoT

Analyzing OpenBTS Performance as a viable network solution for IoT devices

Edward Paul Guillen Pinto^{1*}  <https://orcid.org/0000-0002-2002-4021>
José Daniel Aragón Monroy¹  <https://orcid.org/0009-0004-1155-1964>
Juan Carlos Martínez Quintero¹  <https://orcid.org/0000-0001-9893-6592>

Recibido 04 de Mayo de 2023, aceptado 28 de diciembre de 2023

Received: May 04, 2023 Accepted: December 28, 2023

RESUMEN

La tecnología GSM ha sido esencial para extender las redes de comunicaciones móviles y sigue siendo ampliamente desplegada en zonas con menor densidad de población. Al mismo tiempo, el desarrollo del IoT a nivel mundial se extiende a las áreas rurales para mejorar los procesos agrícolas y aumentar la productividad. Sin embargo, muchas comunidades rurales carecen de conectividad debido a los modelos de negocio restrictivos de los operadores de servicios de telecomunicaciones. En este sentido, la tecnología GSM y específicamente tecnologías como OPENBTS se presentan como una alternativa viable para brindar soporte de red. Casos exitosos como Rhizomática en México y sistemas OpenBTS en Zambia han demostrado que es posible implementar soluciones de comunicación que benefician a las comunidades. El presente artículo busca mostrar cómo es posible aprovechar esta infraestructura tecnológica existente para avanzar en el desarrollo de soluciones IoT empleando sistemas como OpenBTS, incluso con limitaciones en las tasas de transmisión de datos. Se incluye una evaluación del desempeño del sistema OpenBTS en un enlace GPRS, considerando el nivel de throughput y latencia, así como el análisis de latencia para los niveles QoS 0, 1 y 2 en MQTT. Dentro de los resultados se obtuvo una latencia entre 0,4s y 0,75s usando un esquema cliente servidor FTP y entre 0,05s y 0,15s usando un modelo cliente servidor con MQTT con los diferentes niveles de calidad de servicio. En cuanto al throughput se obtuvo un valor máximo de 15616,17 b/s en el downlink y 10853,58 b/s en el uplink.

Palabras clave: Internet de las cosas, servicio general de paquetes vía radio, OpenBTS.

ABSTRACT

The GSM technology has been essential for extending mobile communication networks and continues to be widely deployed in areas with lower population density. At the same time, the development of IoT worldwide is extending to rural areas to improve agricultural processes and increase productivity. However, many rural communities need more connectivity due to the restrictive business models of telecommunications service operators. In this regard, GSM technology, and specifically technologies like OPENBTS, presents a viable alternative to providing network support. Successful cases like Rhizomática in Mexico and OpenBTS systems in Zambia have shown that it is possible to implement communication solutions that benefit communities. This article shows the potential of taking advantage of existing

¹ Universidad Militar Nueva Granada. Programa de Ingeniería en Telecomunicaciones. Bogotá, Colombia.
E-mail: Edward.guillen@unimilitar.edu.co; u1401134@unimilitar.edu.co; juan.martinezq@unimilitar.edu.co

* Autor de correspondencia: edward.guillen@unimilitar.edu.co

technological infrastructures to promote the development of IoT solutions employing systems such as OpenBTS, even with data transmission rate limitations. An evaluation of the performance of the OpenBTS system was included in a GPRS link, considering the level of throughput and latency, as well as latency analysis for QoS levels 0, 1, and 2 in MQTT. The results obtained a latency between 0.4s and 0.75s using an FTP client-server scheme and between 0.05s and 0.15s using a client-server model with MQTT with different levels of quality of service. As for throughput, a maximum value of 15616.17 b/s was obtained in the downlink and 10853.58 b/s in the uplink.

Keywords: Internet of Things (IoT), general packet radio service (GPRS), OpenBTS

INTRODUCCIÓN

El Internet de las cosas (IoT) ha sido una tecnología que ha experimentado un crecimiento vertiginoso en los últimos años, con la proliferación de dispositivos conectados a la red en hogares, ciudades y empresas. Sin embargo, a pesar de la rápida expansión del IoT, su despliegue en zonas rurales o muy apartadas ha sido lento. Esto se debe en gran medida a la falta de infraestructura adecuada en estas zonas, lo que hace que el despliegue de sistemas IoT sea más costoso y complejo. Además, la baja densidad poblacional en estas zonas hace que no sea rentable para los operadores desplegar redes de comunicación de alta velocidad y capacidad, lo que restringe la posibilidad de implementar sistemas IoT en estas áreas. OpenBTS puede ser una herramienta valiosa para el despliegue de tecnologías de IoT en zonas remotas con bajas velocidades de comunicación. Una de las ventajas de OpenBTS es su flexibilidad y capacidad para adaptarse a diferentes necesidades y condiciones, lo que lo convierte en una opción atractiva para las aplicaciones de IoT. Por ejemplo, puede ser utilizado para proporcionar conectividad celular a dispositivos de IoT a través de la red GSM existente, lo que permite a los dispositivos comunicarse de forma remota en zonas donde la cobertura celular es limitada o inexistente. Además, OpenBTS puede ser utilizado para crear redes privadas de IoT, lo que es especialmente útil en entornos donde se requiere una mayor seguridad de los datos. En resumen, OpenBTS puede ayudar a superar los desafíos de conectividad y permitir el despliegue de tecnologías de IoT incluso en zonas muy apartadas con baja velocidad de despliegue. A continuación, se presentan algunos y desarrollos realizados con OPENBTS en los últimos años.

OpenBTS tiene un gran potencial para su implementación en diferentes aplicaciones, pero

algunos estudios han demostrado que también puede ser utilizado de manera malintencionada. Por ejemplo, en [1] se presenta un análisis de vulnerabilidades en redes móviles GSM a través de la ejecución de un ataque man-in-the-middle activo/pasivo. En este estudio se estableció un sistema GSM malicioso utilizando USRP y OpenBTS, con el objetivo de enviar SMS modificados a múltiples usuarios y obtener sus IMSI. El análisis fue llevado a cabo en un ambiente controlado y tuvo éxito en la ejecución, evidenciando vulnerabilidades en la autenticación de los usuarios móviles. En [2], se implementó una red móvil utilizando OpenBTS y USRP Ettus N210, con el fin de generar tráfico VoIP mediante Asterisk. Se capturó el tráfico generado entre los móviles durante las llamadas, utilizando Wireshark para identificar vulnerabilidades en los procesos de autenticación de los usuarios móviles, como la autenticación bilateral y el registro de ubicación del móvil. Estos análisis pueden ser útiles para evitar ataques man-in-the-middle. Finalmente, en [3], se presenta la implementación de un IMSI catcher utilizando USRP y OpenBTS, que envía SMS maliciosos a múltiples usuarios. Durante el ataque, se capturó el tráfico no encriptado por la BTS y se determinó que una de las principales vulnerabilidades de las redes GSM es que el atacante puede aislar completamente a un usuario de la red móvil sin su conocimiento, obligándolo a conectarse a una estación base malintencionada. Sin embargo, es importante destacar que este tipo de ataque tiene restricciones de distancia debido a la potencia de las USRP y debe ser ejecutado en las proximidades de los usuarios.

OpenBTS tiene múltiples aplicaciones, entre ellas, la comunicación en situaciones de desastre. En estos casos, la infraestructura de comunicaciones puede verse afectada por daños estructurales o cortes de energía. El uso de OpenBTS puede ser efectivo

tanto para la comunicación entre rescatistas como para localizar o contactar personas atrapadas. Por ejemplo, en [4] se presenta un prototipo de red GSM de rápido despliegue que brinda servicios de voz en situaciones de emergencia. Se evaluó la calidad del servicio mediante un número variable de llamadas usando dos nodos OpenBTS conectados por un enlace wifi. En [5], se propone el uso de vehículos aéreos no tripulados para transportar una celda de telefonía GSM implementada con OpenBTS que brinda cobertura a los organismos de socorro en caso de un desastre. En [6], se ofrece al equipo de rescate una celda OpenBTS móvil que permite predecir la distancia a la víctima, además de los servicios de voz y mensajes de texto. El diseño del sistema aprovecha que la mayoría de las personas tienen un equipo de telefonía móvil y que, en caso de estar atrapadas, el dispositivo se conecta automáticamente a OpenBTS en ausencia de otra infraestructura de telefonía móvil. OpenBTS también se muestra como una solución para brindar cobertura de servicios de voz y mensajería de texto en zonas apartadas donde no es rentable para los operadores desplegar infraestructura de telefonía móvil debido a la baja densidad poblacional o a los reducidos ingresos de las comunidades que allí habitan. En [7] y [8], se describe la implementación de una red de telefonía móvil usando OpenBTS en una zona rural de Macha en Zambia. Cada estación base contó con un amplificador de potencia de 1W, lo que permitió obtener una cobertura de aproximadamente 3 km de radio. Para establecer la comunicación con otras redes, se utilizó un enlace wifi de larga distancia que ya estaba implementado en la zona. En [9] se propone una infraestructura de telecomunicaciones GSM portátil, simple y económica utilizando OpenBTS con USRP N210, que puede ser aplicada en áreas remotas de Indonesia que no cuentan con redes de telecomunicaciones. Se describe la implementación experimental de la versión 5.0 del software GSM OpenBTS en USRP N210, que ha sido adaptada para evitar fallos. Como resultado de la experimentación se obtuvo una infraestructura de OpenBTS económica y portátil que permite conectar 4 teléfonos celulares a la red GSM con una cobertura de 25 metros para llamadas de audio y 75 metros para mensajes de texto. En [10] se describe la implementación de aplicaciones de Internet de las cosas (IoT) en Perú y destaca que, aunque existen varias tecnologías de acceso a radio para que los dispositivos finales lleguen a la nube IoT, GPRS sigue siendo la opción más disponible en zonas remotas y es adecuada para aplicaciones de IoT

como la agricultura inteligente o el monitoreo de la contaminación del agua. Para probar los prototipos de nodos de sensores basados en GPRS, se construyó una pequeña red GPRS basada en la plataforma de software OpenBTS. Se describe la implementación del prototipo y los resultados de dos experimentos realizados en la red de prueba.

En la revisión del estado del arte presentada no se evidencia ningún estudio que muestre resultados en la medición de parámetros de calidad del servicio entre OpenBTS y nodos IoT. El presente artículo muestra la viabilidad de uso de OpenBTS para conectar dispositivos IoT en regiones apartadas o con baja cobertura por parte de operadores comerciales mediante escenarios de prueba que validan la suficiencia de este tipo de redes. En la primera parte del artículo se presenta una contextualización teórica de las tecnologías empleadas en la investigación seguido de la metodología empleada en la que se incluyen los escenarios de prueba utilizados y la forma en que se realizaron las mediciones. Por último, se presentan los resultados y conclusiones.

CONTEXTO TEÓRICO

OpenBTS

El proyecto OpenBTS es una iniciativa que busca revolucionar las redes móviles, sustituyendo los protocolos de telecomunicaciones tradicionales y los sistemas de hardware complejos con el Protocolo de Internet y una arquitectura de software flexible [11]. OpenBTS es una colección de componentes de software de código abierto que se pueden usar para construir una red más moderna y liviana, permitiendo que la interfaz de radio “Um” de la red móvil tradicional se interconecte directamente con los protocolos de telefonía de Internet [12]. Para implementar una red con OpenBTS, solo se necesita un computador y un radio SDR (Radio Definida por Software), que se conectan a través de un cable giga Ethernet o USB, según el hardware que se utilice. OpenBTS consta de tres componentes principales, interconectados mediante una API llamada NodeManager, tal como se muestra en la Figura 1.

MQTT

El protocolo Message Queue Telemetry Transport, también conocido como Transporte de Telemetría de la Cola de Mensajes, es ampliamente utilizado para la transmisión de mensajes de baja carga

Fuente: [10].

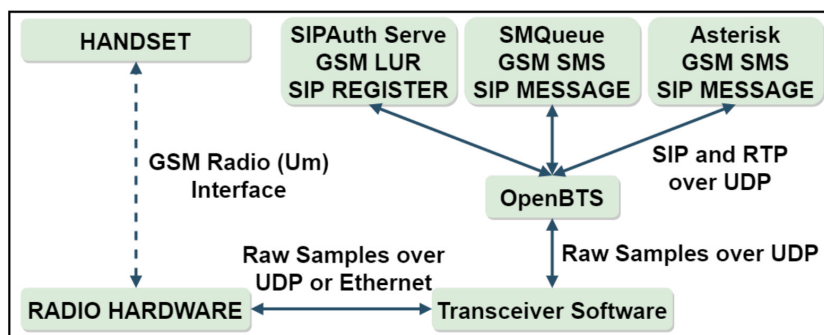


Figura 1. Arquitectura de OpenBTS.

útil, especialmente en sensores de baja potencia. Aunque se ha diseñado específicamente para este tipo de dispositivos, su aplicación se ha extendido a diferentes escenarios. Su funcionamiento se basa en el modelo de publicación y suscripción, donde el cliente debe suscribirse a un tópico compartido por los sensores que publican los mensajes o datos recolectados. El tópico, que es una etiqueta común, permite la comunicación entre los dispositivos [13].

Calidad del Servicio en MQTT

MQTT ofrece tres niveles de calidad de servicio (QoS) que permiten determinar si un mensaje ha sido recibido correctamente en el extremo receptor. Al suscribirse a un tópico, el cliente puede especificar el nivel de QoS deseado para la recepción de los mensajes. El nivel de QoS también se aplica al envío de los mensajes, lo que permite al cliente elegir la máxima garantía de entrega del mensaje. Sin embargo, el nivel de QoS más alto implica un mayor uso de recursos, como la latencia y el ancho de banda [13]. Los tres niveles de QoS son los siguientes (Figura 2):

- QoS 0: se envía el mensaje, pero no se recibe ninguna confirmación de llegada.
- QoS 1: se envía el mensaje y se recibe una confirmación de llegada una vez que llega al otro extremo.
- QoS 2: el mensaje se entrega después de que ambas partes hayan realizado un handshake de cuatro pasos.

Asignación de multislot de GPRS

GPRS agrupa timeslots para incrementar tasas de transferencia, esta característica es llamada “multislotting”. El máximo multislot GPRS de carga y

descarga es un parámetro que controla la cantidad de slots que OpenBTS entregará a cada dispositivo o estación móvil para sus respectivos enlaces de carga y descarga y que finalmente serán utilizados por cada usuario [14].

Las combinaciones de Multislots que OpenBTS acepta se pueden observar en la Tabla 1.

METODOLOGÍA

La metodología propuesta contempla la implementación de la red GSM, el desarrollo de los escenarios de prueba para realizar las medidas de throughput y latencia y la realización de las

Fuente: [11].

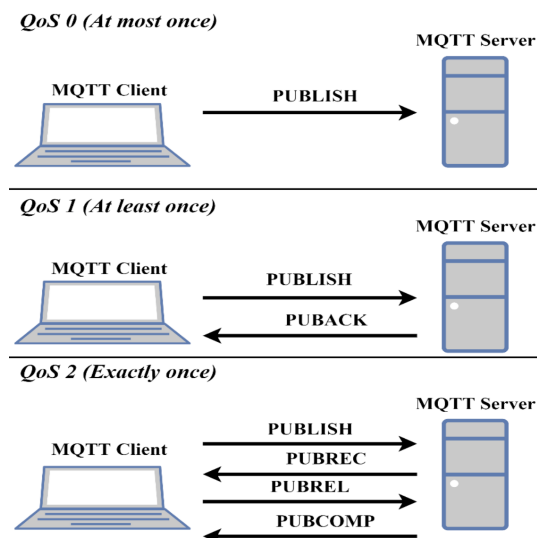


Figura 2. Niveles de QoS en MQTT.

Tabla 1. Clases comunes de multislot en GPRS que entregan las máximas capacidades de carga y descarga [14].

MutislotClass	MaxRx	MaxTx	MaxSumRx+Tx
1	1	1	2
2	2	1	3
3	2	2	3
4	3	1	4
5	2	2	4
6	3	2	4
7	3	3	4
8	4	1	5
9	3	2	5
10	4	2	5
11	4	3	5
12	4	4	5

mediciones. A continuación, se detallan estos procesos.

Características de Implementación de la Red

Se ha propuesto una metodología para probar y analizar los parámetros fundamentales de la comunicación entre nodos IoT y un servidor mediante el uso del servicio GPRS en una red celular GSM implementada con OpenBTS. Los parámetros básicos de identificación de la red GSM se definieron como se muestra en la Tabla 2.

La banda y el canal, se seleccionaron mediante la verificación con un analizador de espectro para determinar cuáles canales no se estaban utilizando en la zona de prueba y así no recibir ni realizar ninguna interferencia. El MCC = 732 y MNC = 88 se seleccionaron porque se encuentran disponibles según la CRC (Comisión de Regulación de Comunicaciones) [15].

Tabla 2. Parámetros de identificación de la red GSM.

Parámetro de Celda	Valor	Parámetro OPENBTS
Band	900Mhz	GSM.Radio.Band
Channel	50	GSM.Radio.C0
MCC	732	GSM.Identity.MCC
MNC	88	GSM.Identity.MNC
Network	OpenUMNG	GSM.Identity.ShortName

En la puesta en marcha del servicio GPRS en la red GSM se tomó la configuración más veloz y con más cobertura de acuerdo a lo planteado por el desarrollador de OpenBTS y el libro “GPRS in Practice” [14], [16]. La configuración de la codificación y la cantidad de timeslots usados en los enlaces de subida y bajada se puede observar en la Tabla 3.

La celda GSM implementada utiliza OpenBTS con una USRP N210 como interfaz de radio. Esta celda permite en principio la comunicación entre dispositivos móviles compatibles con GSM, por medio de mensajes (SMS) y llamadas. En la Figura 3 se presenta la topología básica implementada. No se usó amplificador de potencia de RF.

Luego de tener en funcionamiento la red básica, se adicionó el servicio de GPRS para poder brindar internet a los dispositivos móviles conectados. Para este servicio se debe de tener en cuenta los slots que se usaran y el tipo de codificación definidos previamente en la Tabla 3. La topología de la red GSM con el servicio GPRS se muestra en la Figura 4.

Tabla 3. Configuración para GPRS.

Parámetro	Configuración
CodecUplink	cs-1
CodecDownlink	cs-1
TimeslotsUplink	2
TimeslotsDownlink	3

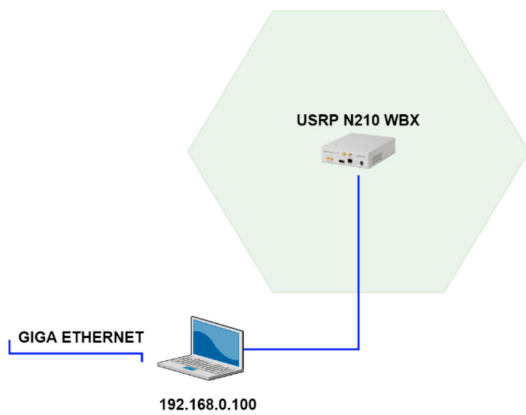


Figura 3. Topología de GSM con OpenBTS.

Escenario de Prueba 1

Se implementó una arquitectura cliente-servidor FTP para realizar medidas de throughput y latencia. La Figura 5 muestra la topología implementada en la que se observa un servidor FTP, OpenBTS con el hardware de radio (USRP N210), un celular y un cliente FTP.

Se realizaron pruebas de throughput y latencia a dos distancias diferentes (20 m y 40 m) entre la celda y el celular para experimentar una variación de la potencia de recepción como se muestra en la Figura 6. La potencia de recepción en la distancia 1 es de $-70,20 \pm 5$ dBm y la potencia de recepción en la distancia 2 es de $-85,65 \pm 5$ dBm.

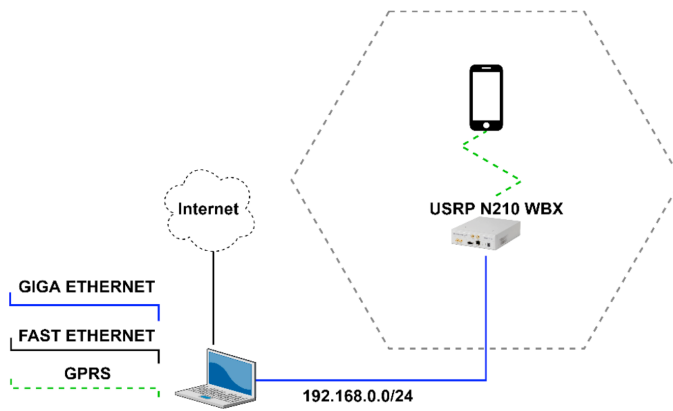


Figura 4. Topología de GPRS con OpenBTS.

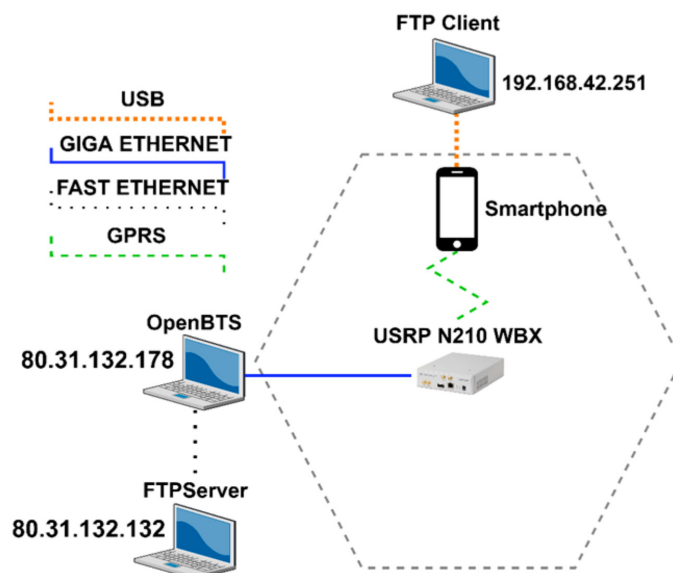


Figura 5. Topología para la prueba 1.

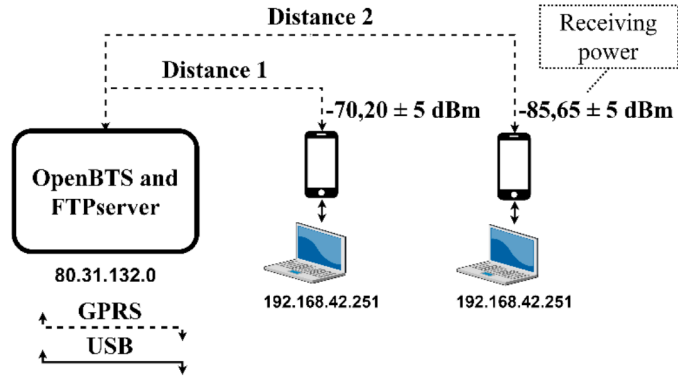


Figura 6. Cambio de distancia en el escenario de prueba 1.

Escenario de Prueba 2

Se realizaron medidas de latencia mientras el servicio MQTT estaba en funcionamiento. La Figura 7 muestra el escenario implementado, en el que se encuentra un servidor Mosquitto (MQTT Server), OpenBTS con hardware de radio USRP N210, un celular y un cliente Mosquitto.

En las pruebas de latencia se realizó una variación de la calidad del servicio de MQTT cambiando el nivel QoS entre 0, 1 y 2, como se muestra en la Figura 8.

Proceso de medición

La captura de paquetes se lleva a cabo usando el software Wireshark. Este software se instala en el

cliente y en el servidor para realizar las medidas requeridas y obtener información relacionada con throughput y latencia.

Para determinar el throughput se requiere medir los paquetes en el cliente, mientras se envían y reciben archivos con un tamaño conocido, midiendo también los tiempos involucrados.

$$\text{Throughput}(bps) = \frac{S(b)}{t_2(s) - t_1(s)} \quad (1)$$

En la captura de paquetes se debe establecer desde que trama se empieza a enviar información y hasta que trama finaliza el envío como se observa en la Figura 9 para el escenario de prueba 1. Después

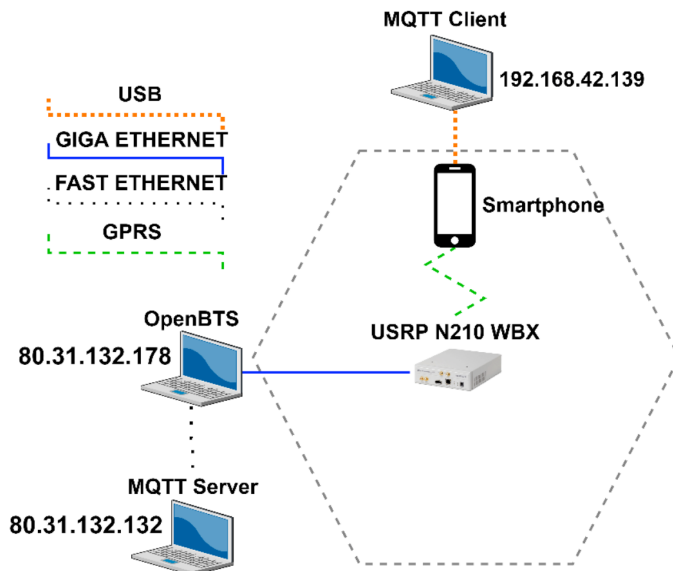


Figura 7. Topología para el escenario de prueba 2.

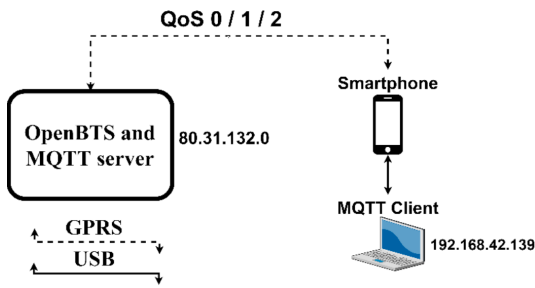


Figura 8. Cambio de QoS de MQTT en el escenario de prueba 2.

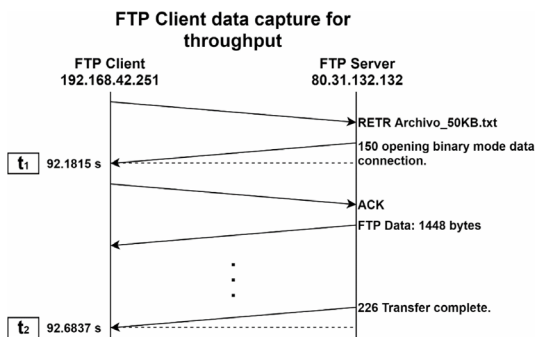


Figura 9. Medición de Throughput utilizando un servidor FTP.

de identificar las tramas y sus tiempos, se resta el tiempo de inicio t_1 con el tiempo de fin t_2 obteniendo el tiempo de envío. El tamaño del archivo S se divide entre el tiempo de envío y el resultado es el throughput como se ve en la ecuación (1).

Para medir la latencia se capturan los paquetes en el cliente y en el servidor mientras se transmite algún archivo. En el software después de medir

la transmisión se filtran las tramas, dejando solo las importantes para esta prueba, es decir, las que pertenecen a la transmisión entre cliente y servidor. Con la herramienta “flow graph” de Wireshark se identifican mejor los elementos de la captura. Con los identificadores, las tramas se asocian al cliente o al servidor. En este caso como las tramas son de tipo TCP se puede identificar con el número de ACK, SEQ y puerto utilizado como se muestra en la Figura 10.

De la información obtenida se extrae el tiempo de envío de trama t_{1c} y cuando se recibe la respuesta en el cliente t_{2c} , también se obtiene el tiempo cuando se recibe la trama en OpenBTS t_{1s} y cuando envía la respuesta el servidor t_{2s} .

$$Latency = (t_{2c} - t_{1c}) - (t_{2s} - t_{1s}) \quad (2)$$

El tiempo del cliente se resta con el del servidor y se obtiene el tiempo de latencia como se puede observar en la ecuación (2).

En el escenario de prueba 2 también se realizó medición de latencia. En este caso, se filtran las tramas de MQTT y se buscan los tiempos en los cuales inicia la conexión (t_{1c} y t_{1s}) y los tiempos en el que se desconecta (t_{2c} y t_{2s}). Como en el caso anterior se toman los tiempos de cliente y servidor y se realiza el cálculo de la latencia usando la ecuación 2.

RESULTADOS

Para el escenario 1 se realizaron las pruebas antes mencionadas mediante los enlaces “uplink” y “downlink” de la celda y la comunicación establecida entre el servidor FTP local y un cliente. El cliente

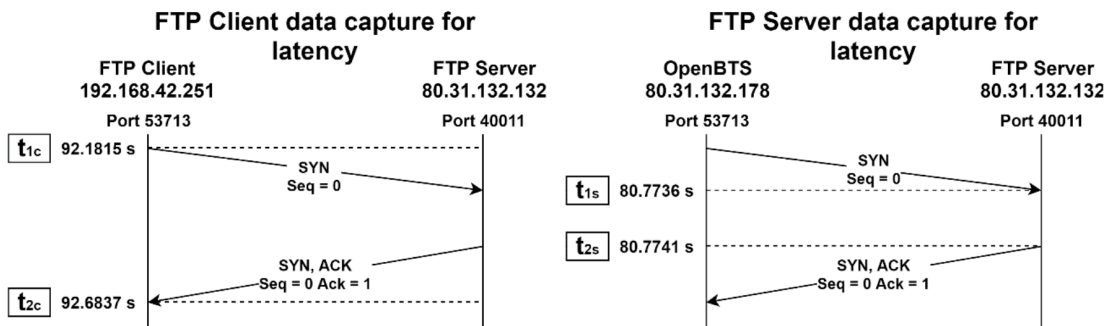


Figura 10. Medición de latencia utilizando un servidor FTP.

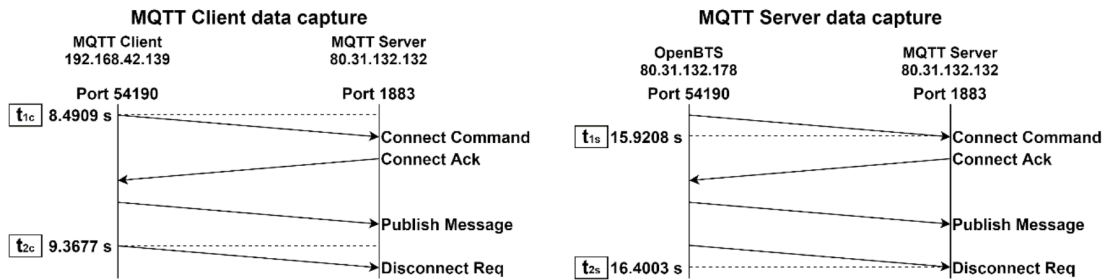


Figura 11. Medición de latencia utilizando un servidor MQTT.

FTP obtiene los datos mediante el celular conectado al servicio GPRS de la celda implementada. Se obtuvieron un total de 120 mediciones con los que se generó la Figura 12.

El comportamiento del throughput con respecto a la carga y descarga se midió usando 3 archivos de distintos tamaños como se observa en la Figura 12, en este proceso se analiza también el efecto provocado por el cambio de potencia recibida al variar las distancias entre la celda (OpenBTS) y el celular.

El throughput es mayor en el downlink para los tres casos debido a que en ese enlace se dedican 3 slots mientras que el uplink tiene solo 2. En la Figura 12 se puede observar también que las variaciones de potencia en el receptor y el tamaño del archivo tienen poca afectación en la medida del throughput.

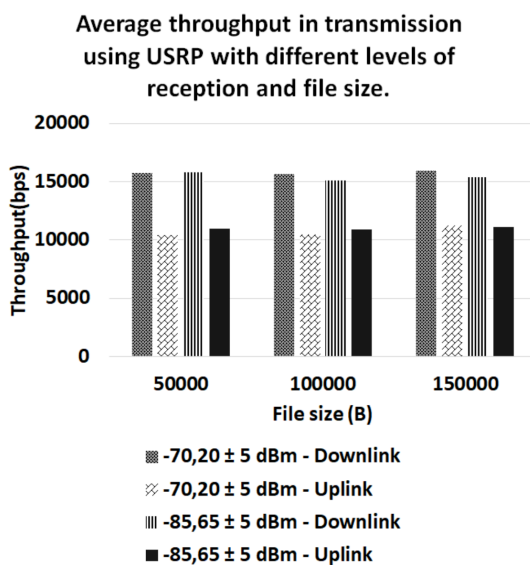


Figura 12. Resultado de la prueba de throughput usando un servidor FTP.

En la Tabla 4 se compara la velocidad teórica que entrega OpenBTS con la velocidad obtenida en las pruebas.

Por otro lado, se realizaron pruebas de latencia en los dos escenarios. Con el servicio de FTP (escenario de prueba 1) se obtuvieron 120 medidas y con el servicio MQTT (escenario de prueba 2) fueron 60. Con estos datos se generaron las Figura 13 y Figura 14.

Tabla 4. Comparación del rendimiento obtenido con el rendimiento teórico.

Link	Theoretical throughput b/s	Obtained throughput b/s
Downlink	27150	15616,17
Uplink	18100	10853,58

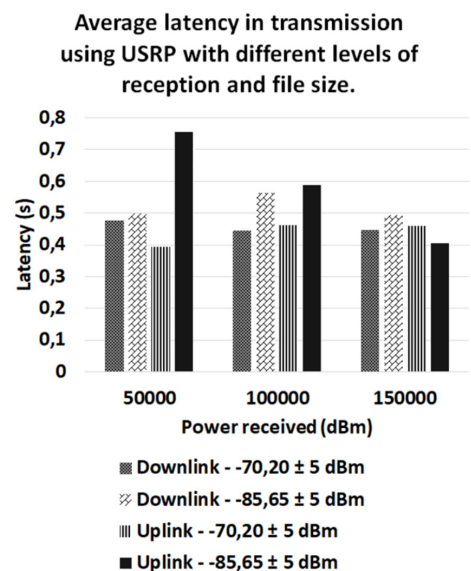


Figura 13. Resultado de la prueba de latencia usando un servidor FTP.

En la Figura 13, se presenta una gráfica donde se evidencia que el rango de latencia obtenido está entre 0,4s y 0,75s y se afectaron levemente por el tamaño del archivo y por la potencia de recepción para este escenario de prueba.

En la Figura 14 se observa una gráfica que relaciona la latencia con los niveles de calidad de servicio (QoS) en MQTT, se puede observar una tendencia que entre mayor es el nivel de calidad del servicio mayor va a ser la latencia, también se puede determinar que el rango de la latencia de MQTT será entre 0,05s y 0,15s para este escenario.

CONCLUSIONES

Basándonos en los valores de throughput y latencia alcanzados, podemos concluir que OpenBTS es capaz de proporcionar una conectividad suficiente para modelos de comunicación como el sistema de publicación y suscripción de MQTT. En muchos casos, las redes de sensores IoT no requieren la transmisión de grandes cantidades de datos ni el envío periódico de los mismos, por lo tanto, se puede considerar que OpenBTS es una alternativa viable para regiones remotas donde la infraestructura móvil es limitada o no existe.

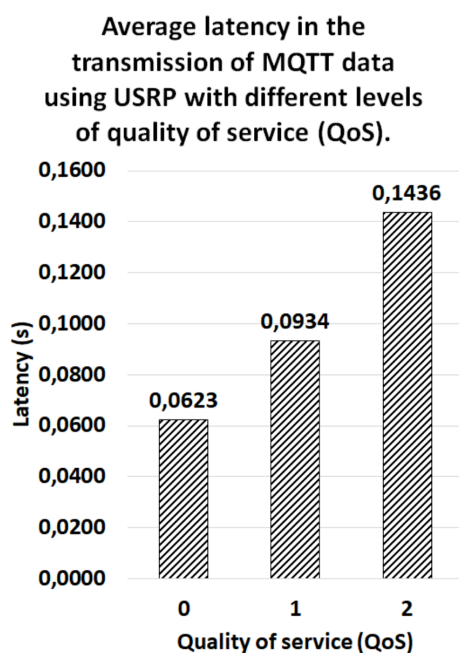


Figura 14. Resultado de la prueba de throughput usando un servidor MQTT.

OpenBTS puede ser una alternativa atractiva para implementar soluciones de comunicación en sistemas IoT debido a su flexibilidad y adaptabilidad. Esta tecnología puede ser utilizada para crear redes privadas de comunicación de bajo costo y alta disponibilidad en áreas remotas, lo que es especialmente útil en aplicaciones agrícolas, monitoreo ambiental, monitoreo de infraestructuras y otras aplicaciones IoT. Además, la implementación de soluciones de comunicación basadas en OpenBTS puede contribuir a la reducción de costos operativos y a una mayor eficiencia en la transmisión de datos en sistemas IoT. En resumen, OpenBTS puede ser una herramienta valiosa para mejorar la conectividad y aumentar la eficiencia en la implementación de soluciones IoT en regiones con poca infraestructura móvil.

La topología planteada en la Figura 7 puede ser usada para crear una red IoT con más nodos al reemplazar el cliente MQTT por un Gateway MQTT que haga uso de otras tecnologías inalámbricas como LORA, WIFI o Bluetooth para dar alcance a otros nodos. Como se muestra en las Figuras 12, 13 y 14 es posible realizar transmisión de datos de nodos IoT con velocidades y latencias razonables teniendo en cuenta los alcances del diseño de red IoT que se requiera.

Por otro lado, el uso de tecnologías basadas en radio definida por software se consolida como una herramienta que puede emplearse para lograr conectividad y proporcionar servicio a comunidades que carecen de ella. Se propone como trabajo futuro un estudio para determinar el throughput y la latencia en función de cualquier distancia o de la potencia.

AGRADECIMIENTOS

Este Trabajo fue financiado por la Vicerrectoría de Investigaciones de la Universidad Militar Nueva Granada y desarrollado por el grupo de investigación GISSIC asociado como resultado del proyecto de investigación ING-INV 2646 “GSM-DETECT: Sistema de diagnóstico de vulnerabilidades tipo suplantación en redes móviles GSM mediante Software Defined Radio”.

REFERENCIAS

- [1] A. Dubey, D. Vohra, K. Vachhani, and A. Rao, “Demonstration of vulnerabilities in GSM security with USRP B200 and open-

- source penetration tools,” in *Asia-Pacific Conference on Communications*, APCC 2016, 2016, pp. 496-501.
- [2] B. Harmat et al., “The Security Implications of IMSI Catchers,” in *International Conference on Security and Management (SAM’15)*, 2015, pp. 57-62.
- [3] M. Hadžialić, M. Škrbić, K. Huseinović, I. Kočan, and J. Mušović, “An Approach to Analyze Security of GSM Network,” *22nd Telecommunications Forum Telfor (TELFOR)*, Belgrade, Serbia, 2014, pp. 99-102, doi: 10.1109/TELFOR.2014.7034366.
- [4] S. Ghafoor, K.N. Brown, and C.J. Sreenan, “Experimental evaluation of a software defined radio-based prototype for a disaster response cellular network,” in *Proceedings of the 2015 2nd International Conference on Information and Communication Technologies for Disaster Management, ICT-DM 2015*, 2016, pp. 57-63, doi: 10.1109/ict-dm.2015.7402034.
- [5] K. Guevara, M. Rodriguez, N. Gallo, G. Velasco, K. Vasudeva and I. Guvenc, “UAV-based GSM network for public safety communications,” *SoutheastCon 2015*, Fort Lauderdale, FL, USA, 2015, pp. 1-2, doi: 10.1109/SECON.2015.7132971.
- [6] T.D. Putri and T. Juhana, “Mobile-openbts implementation of natural disaster victims search,” *2017 3rd International Conference on Wireless and Telematics (ICWT)*, Palembang, Indonesia, 2017, pp. 149-154, doi: 10.1109/ICWT.2017.8284157.
- [7] J. Mpala and G. van Stam, “Open BTS, a GSM experiment in rural Zambia,” *AFRICOMM 2012. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, vol. 119, K. Jonas, I.A. Rai, and M. Tchunte, Eds., Yaounde, Cameroon, 2012, pp. 1-9.
- [8] M. Zheleva, A. Paul, D.L. Johnson, and E. Belding, “Kwiizya: Local Cellular Network Services in Remote Areas,” in *MobiSys’13: The 11th Annual International Conference on Mobile Systems, Applications, and Services*, p. 417, 2013.
- [9] F.H. Partiansyah, S. Kusmaryanto, R. Ambarwati, and S.H. Pramono, “Experimental Study of USRP N210 as Simple GSM OpenBTS 5.0 for Remote Areas,” *2022 11th Electrical Power, Electronics, Communications, Controls and Informatics Seminar (EECCIS)*, Malang, Indonesia, 2022, pp. 185-190, doi: 10.1109/EECCIS54468.2022.9902951.
- [10] O.E.L. Castro, “Network GPRS Prototype based on SDR and OpenBTS, as an IoT-lab Testbed,” *2020 Seventh International Conference on Software Defined Systems (SDS)*, Paris, France, 2020, pp. 14-19, doi: 10.1109/SDS49854.2020.9143873.
- [11] OpenBTS, openbts.org. [Online]. Available: www.openbts.org (Accessed: Apr. 15, 2019).
- [12] M. Ledema, *Getting Started with OpenBTS: Build Open Source Mobile Networks*, Sebastopol, CA, USA: O’Reilly Media, 2014.
- [13] R. Light, “MQTT man page,” mosquitto.org. <https://mosquitto.org/man/mqtt-7.html> (Accessed: Nov. 22, 2019).
- [14] OpenBTS, *OpenBTS Application Suite*, openbts.org. [Online]. Available: <http://openbts.org/site/wp-content/uploads/2014/07/OpenBTS-4.0-Manual.pdf> (Accessed: Nov. 22, 2019).
- [15] CRC, “SIGRID”, pnn.gov.co. [Online]. Available: www.pnn.gov.co/mapa/codigosMNC.xhtml (Accedido: 22 Nov. de 2019).
- [16] P. McGuiggan, *GPRS in Practice: A Companion to the Specifications*; Sussex, UK: John Wiley & Sons, 2005.