

# Evaluación del rendimiento de un sistema de ficheros para sistemas IoT

Elías Del-Pozo-Puñal<sup>1</sup>, Félix García-Carballeira<sup>1</sup>, Diego Camarmas-Alonso<sup>1</sup> y Alejandro Calderón-Mateos<sup>1</sup>

*Resumen*— Los IoT son dispositivos interconectados a través de una red, que pueden interactuar sin la intervención humana. Estos dispositivos podrían ser desde sensores hasta objetos cotidianos, como un frigorífico o coches, etc. Estos dispositivos han ido evolucionando a lo largo del tiempo para tener una mayor capacidad de cómputo y almacenamiento. Esta evolución permite que estos dispositivos puedan realizar una mayor cantidad de tareas.

En este trabajo se quiere aprovechar dichas capacidades para dotar a estos dispositivos de almacenar los datos recogidos en un sistema de ficheros paralelo y distribuido adaptado a IoT, aprovechando el protocolo de comunicación MQTT.

La evaluación se realiza en tres fases. Primero, se analiza en un entorno real con una Raspberry Pi para medir consumo de energía y latencia, determinando la viabilidad práctica del uso del sistema de ficheros. En segundo lugar, se han realizado pruebas en un entorno emulado para evaluar la escalabilidad del sistema. Y, tercero, se han llevado a cabo pruebas de simulación para validar los resultados de la primera y, además, a gran escala para comprobar la efectividad de la solución en entornos de mayor escala.

Tras la evaluación realizada el sistema de ficheros distribuido y paralelo Expand basado en el uso de MQTT se presenta como alternativa para proporcionar una base para el procesamiento eficiente de datos en entornos IoT.

*Palabras clave*— Almacenamiento IoT, MQTT, Sistema de Ficheros Paralelo y Distribuido, Expand

## I. INTRODUCCIÓN

LOS sistemas de ficheros paralelos y distribuidos han experimentado un continuo proceso de diseño y desarrollo durante las últimas décadas [1], respondiendo a demandas emergentes. Algunos de estos ejemplos son GPFS [2], HDFS [3], PVFS [4] o Lustre [5]. Sin embargo, el panorama actual presenta diferentes retos debido a la aparición de aplicaciones y dispositivos de almacenamiento innovadores. Como consecuencia, estos sistemas necesitan adaptarse a estas nuevas demandas y escenarios tecnológicos de forma eficaz. Por lo tanto, es importante comprender estos cambios con enfoques que garanticen la mejora del rendimiento y la funcionalidad en un entorno en constante evolución [6].

Como ejemplo, es evidente que las aplicaciones en las áreas de Inteligencia Artificial [7], [8] o *big data* [9] están experimentando un avance vertiginoso, generando una demanda continua de procesamiento masivo de datos mediante la ejecución de flujos de trabajo [10], [11]. Esta cantidad de datos puede provenir de una gran variedad de sensores que residen en un sistema IoT (*Internet of Things*) [12], que requieren de un procesamiento, envío y almacenamiento

eficientes. En esta constante evolución de las aplicaciones y de los dispositivos, surge la oportunidad de descentralizar el cómputo e, incluso, del almacenar parcialmente los datos en los propios dispositivos.

En consonancia con este panorama, las áreas de trabajo asociadas a la Inteligencia Artificial o *big data* requieren un entorno informático de alto rendimiento adaptado a sus características específicas. A diferencia de las aplicaciones convencionales de cómputo de alto rendimiento (HPC), estas nuevas aplicaciones exhiben patrones de acceso a los datos totalmente distintos.

En los últimos años, el uso de dispositivos conectados ha experimentado un crecimiento exponencial. Sin embargo, este aumento masivo del número de dispositivos conlleva importantes retos, como la congestión del tráfico y la pérdida de datos que, en determinados casos de uso, pueden ser críticos [13], [14].

Para hacer frente a estos problemas, se han desarrollado diversos mecanismos y protocolos de comunicación. Aunque se puede elegir la opción de crear una solución ad-hoc sobre el protocolo TCP/IP[15], que escale y no pierda mensajes también se puede utilizar un estándar como MQTT[16], que ofrezca un protocolo de red ligero, de publicación–suscripción y con un servicio de colas de mensajes.

El objetivo de este trabajo consiste en dotar a los dispositivos IoT la capacidad de almacenar directamente los datos recogidos en los propios sistemas de ficheros, aprovechando la existencia de un protocolo de comunicación preparado para estos dispositivos. Para poder lograrlo, se ha usado el sistema de ficheros Expand [17], [18] para el almacenamiento de dichos datos y la implementación de Mosquitto, que usa el protocolo MQTT, para la transmisión de la información.

Con el propósito de alcanzar dicho objetivo y proporcionar una alternativa a estos sistemas, se llevaron a cabo pruebas en tres niveles distintos. En primer lugar, se realizó un análisis de rendimiento y consumo energético en un entorno real utilizando un cluster reducido de nodos con Raspberry Pi, a las cuales se les incorporó la implementación del sistema de ficheros Expand junto con Mosquitto.

Para estudiar los límites de dicha propuesta, se estudió el rendimiento y escalabilidad de Expand con Mosquitto usando un supercomputador. De esta forma, se busca emular un cluster de Raspberry Pi con un número de nodos y una mayor capacidad de computación (en un caso ideal). Finalmente, para verificar la viabilidad del sistema en entornos a gran escala, se empleó la simulación para entornos

<sup>1</sup>Dpto. de Informática, Universidad Carlos III de Madrid, e-mail: {edelpozo, fgcarbal, dcamarma, acaldero}@inf.uc3m.es

HPC, como es el caso de ENIGMA [19][20], basado en Simgrid [21]. Con esto se consigue validar el uso del protocolo de MQTT y estudiar todo tipo de configuraciones (no solo el caso ideal del supercomputador) para IoT.

El resto del documento se divide en las siguientes secciones: En la Sección II se muestra el Estado del Arte; en la Sección III se presenta el sistema de ficheros Expand adaptado a IoT; en la Sección IV se evalúa el sistema de ficheros. Para terminar, en la Sección V se muestran las conclusiones.

## II. ESTADO DEL ARTE

El Internet de las Cosas (IoT) [12], ha experimentado una evolución significativa, generando un impacto considerable en la vida cotidiana de las personas en los últimos años. Este paradigma no solo ha impulsado la productividad, sino que también ha encontrado aplicaciones destacadas en campos tan diversos como la educación, la ciencia y la tecnología.

La interconexión de dispositivos a través de Internet ha desempeñado un papel fundamental al facilitar y agilizar el acceso, análisis y procesamiento de información tanto a pequeña como a gran escala. Este enfoque ha allanado el camino para una gestión más eficiente de los datos, abriendo oportunidades para la innovación en diversos sectores de la sociedad [22].

A pesar de su cambio constante, el Internet de las Cosas se presenta como una arquitectura que fomenta la comunicación eficiente entre múltiples puntos de la red. Esta característica no solo posibilita la transmisión fluida de diversos tipos de datos, sino que también impulsa la capacidad y velocidad de acceso a la información. En consecuencia, se mejora la productividad en una amplia gama de servicios [23].

En la última década, el crecimiento exponencial de los dispositivos conectados se ha visto impulsado por la aparición de nuevos dispositivos como *smartphones*, *tablets*, *smartwatches* y otros dispositivos inteligentes. Esta proliferación de dispositivos conectados ha influido significativamente en el aumento del número de dispositivos interconectados a través de Internet, proporcionando nuevas oportunidades para el intercambio de datos y la generación de información en tiempo real.

Además, el rápido crecimiento en la cantidad de dispositivos conectados ha resultado en un sustancial aumento en el tráfico de información entre distintos puntos de la red. Como consecuencia directa, se ha vuelto esencial contar con un aumento de infraestructuras capaces de gestionar este constante flujo de datos en tiempo real.

La demanda de infraestructuras de red más sólidas y eficientes se ha vuelto imperativa para asegurar un procesamiento ágil y confiable de la información generada por los dispositivos IoT. Estas infraestructuras deben estar preparadas para manejar grandes volúmenes de datos, garantizar una comunicación fiable y minimizar la latencia, sobre todo en aplicaciones críticas que requieren respuestas en tiempo real.

El reto reside en desarrollar y mantener una infraestructura escalable y flexible que pueda adaptarse a la continua expansión del Internet de las Cosas y a la diversidad de dispositivos conectados. La gestión eficiente de este creciente tráfico de información es esencial para aprovechar plenamente el potencial de IoT y permitir el desarrollo de aplicaciones y servicios innovadores en todos los ámbitos de la sociedad [24].

Otra aproximación a los dispositivos IoT es tratarlos como un superconjunto de dispositivos conectados que solo pueden ser identificados mediante técnicas de *Near Field Communication* (NFC).

Cuando se trata de dispositivos IoT, es importante tener en cuenta las diversas cuestiones que plantea una infraestructura a gran escala. A continuación se enumeran algunas de las principales cuestiones que deben tenerse en cuenta al diseñar un sistema formado por estos dispositivos [25]:

- **Interoperabilidad:** Un aspecto esencial de IoT es la gran variedad de dispositivos existentes que deben funcionar con protocolos y estándares diferentes. Aunque en las redes de sensores los dispositivos comparten similitudes en términos de consumo de energía o capacidad de cálculo, la presencia de numerosos protocolos diferentes dificulta la interoperabilidad entre ellos, generando obstáculos significativos para su utilización.
- **Escalabilidad:** Factor importante en la viabilidad de un sistema IoT, ya que el número de dispositivos en una infraestructura puede determinar su eficacia. Conforme aumenta el número de elementos involucrados, la escalabilidad se convierte en un desafío significativo que impacta la gestión de redes y protocolos. Los autores del trabajo [26] han evaluado la semántica de los mecanismos de servicio IoT existentes para analizar su capacidad para cumplir con los requisitos de escalabilidad en estos sistemas. En [27], los autores han mejorado la escalabilidad de la integración IoT mediante el uso de los protocolos MQTT y CoAP, adaptándolos tanto para dispositivos con restricciones como para aquellos sin limitaciones.
- **La gestión de datos:** vinculada al desafío previamente mencionado, se intensifica a medida que crece el número de dispositivos involucrados en una infraestructura, demandando mayores recursos para la transmisión, almacenamiento y administración de datos. En [28], los autores proponen una arquitectura de borde distribuida destinada a procesar y almacenar datos generados por dispositivos IoT cercanos a los productores. Esta estrategia reduce el tiempo de respuesta para adecuarse a los períodos de demanda y, al mismo tiempo, disminuye el consumo de energía.

## III. EXPAND IOT

Este trabajo ha consistido en modificar un sistema de ficheros que se usa en entornos HPC como

es Expand<sup>1</sup> [17], [18] para que sea capaz de comunicarse con dispositivos IoT y así poder almacenar la información generada por dichos dispositivos, cuyo diseño se explica en la Subsección III-A y el resultado se muestra en la Figura 1.

Cuando los dispositivos de IoT necesiten enviar datos de un extremo a otro, podrán escribir directamente en los nodos sin intermediarios, evitando sobrecargar la red. Esto no solo reduce los tiempos de escritura, sino que también aumenta la capacidad de manejo de datos de los dispositivos sin comprometer la fiabilidad ni sobrecargar el sistema.

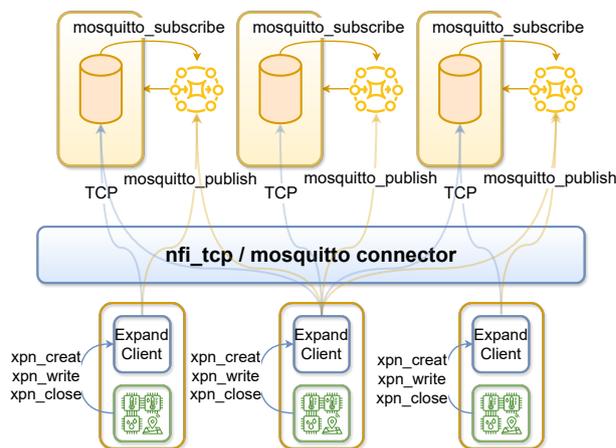


Fig. 1: Arquitectura de Expand para IoT.

Para poder desarrollar esta extensión al propio sistema de ficheros se estudió previamente las diferentes alternativas que existían en el ámbito de las comunicaciones para los IoT, con diferentes protocolos como puede ser MQTT, COaP o REST.

Entre las diferentes opciones existentes, se decidió implementar un conector con el protocolo de comunicación MQTT [29] [30], protocolo de mensajería ligero diseñado para la comunicación entre dispositivos en redes con ancho de banda limitado o en condiciones de alta latencia. Usar este protocolo nos permite minimizar la sobrecarga de la red y a mantener un bajo consumo de energía, entre otros aspectos.

Existen varias implementaciones de MQTT como puede ser HiveMQ[31], diseñado para ser escalable y altamente disponible, AWS IoT Core [32] y ActiveMQ [33] entre otras. HiveMQ destaca por su escalabilidad y su gestión de sesiones y control de calidad de servicio, lo que lo hace una opción viable para escenarios de alta carga. Por último, ActiveMQ es una solución que permite el uso de diferentes protocolos y ofrece unas características que le ofrecen mayor flexibilidad y control de entrega de mensajes.

Sin embargo, entre las posibles elecciones existentes [31] se ha decidido usar Mosquitto [34] ya que destaca por su simplicidad, ligereza y facilidad de uso, lo que lo hace adecuado para implementaciones sencillas y entornos con recursos limitados.

## A. Diseño

La Figura 1 muestra la arquitectura resultante del cambio del diseño de Expand para transformarlo en un sistema de ficheros paralelo para IoT.

Los distintos nodos IoT ejecutan aplicaciones que usan una biblioteca de interceptación con el cliente de Expand. No obstante, las aplicaciones existentes pueden beneficiarse también de usar Expand a través de una biblioteca de interceptación, la cual permite utilizar las llamadas al sistema de POSIX en las aplicaciones sin tener que modificar el código.

La biblioteca cliente de Expand se comunica con los distintos servidores de Expand mediante el uso de una implementación basada en sockets TCP/IP o bien una implementación basada en MQTT. Los servidores de Expand utilizan el espacio de almacenamiento local del nodo donde se ejecutan (ya sea SSD, disco duro, etc.) para guardar datos. Los datos que envía el cliente de Expand se distribuyen entre un conjunto de servidores que forman una partición paralela, de forma que los accesos para guardar (o para leer) los datos se hace en paralelo.

## B. Infraestructuras a evaluar

Uno de los objetivos de la propuesta presentada en este trabajo es, como se ha dicho previamente, la de dotar a los dispositivos IoT la capacidad de gestionar los ficheros almacenando directamente en los nodos de cómputo la información.

Por eso, se desea plantear un nuevo tipo de infraestructura, donde los diferentes IoT forman parte del sistema de almacenamiento, de tal forma que se ahorra un nivel en la infraestructura, reduciendo la carga y la latencia en el envío de información entre extremos y, además, el tiempo de escritura, tal y como se muestra en la Figura 2.

La idea principal consiste en que los sensores se comportan como publicadores de MQTT que envían la información a los propios *broker*, los propios nodos de cómputo actúan como suscriptores, recibiendo directamente la información de los *broker* y almacenándola directamente en sus discos.

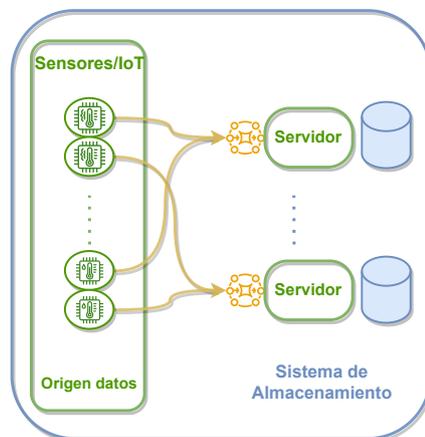


Fig. 2: Nuevo tipo de infraestructura *Edge/Cloud* en la que los dispositivos IoT escriben en el sistema de almacenamiento.

Por otra parte, la Figura 3 ilustra la infraestructura convencional de sistemas IoT/*Edge*, en la cual

<sup>1</sup><https://github.com/xpn-arcos/xpn>

los dispositivos operan fuera del ámbito del sistema de almacenamiento designado. En este escenario, los dispositivos envían o publican información utilizando el protocolo MQTT a los correspondientes *brokers*. Simultáneamente, existe una capa intermedia que recibe esta información de los servidores MQTT y la retransmite o almacena en los nodos de cómputo. Este proceso añade una mayor complejidad y latencia a la tarea de almacenar los datos.

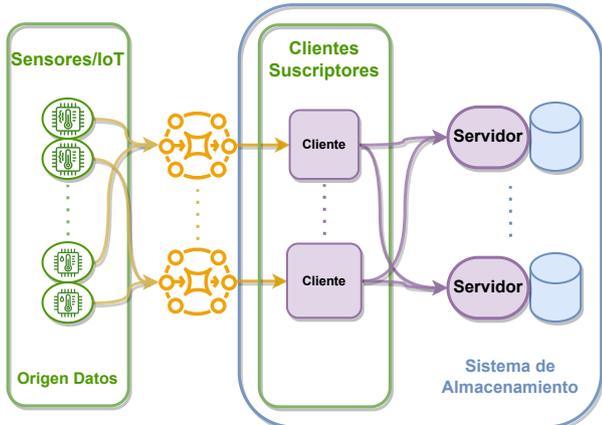


Fig. 3: Infraestructura tradicional *Edge/Cloud* donde los dispositivos IoT envían los paquetes de datos que posteriormente serán procesados y almacenados.

#### IV. EVALUACIÓN

En esta sección se detallarán las diversas evaluaciones realizadas para demostrar las capacidades de extensión del sistema de ficheros Expand. Estas capacidades se orientan a permitir que los dispositivos IoT almacenen directamente la información recopilada o generada en tiempo real.

Para ello, las pruebas que se han realizado para este trabajo se han dividido en tres partes diferenciadas en el entorno que se ha querido probar: real, emulado y simulado. Entorno real para determinar si la implementación desarrollada funciona correctamente y muestra un consumo de energía aceptable. Entorno emulado para comprobar si dicho sistema de ficheros con la extensión de MQTT es capaz de gestionar un aumento del número de dispositivos IoT capaces de enviar datos al mismo tiempo y analizar la estabilidad. Y, por último, un entorno simulado, para estudiar la escalabilidad de dicha implementación y protocolos de comunicación usados.

##### A. Entorno real

En primer lugar se evaluó el rendimiento y funcionamiento de Expand con la implementación proporcionada por la librería Mosquitto en una Raspberry Pi 4. De esta forma, el dispositivo enviará una determinada cantidad de datos a una máquina de propósito general mientras se mide el consumo energético, en Julios, de dicho dispositivo.

El objetivo de utilizar una máquina de propósito general se debe para que se comporte como un servidor que recibirá y almacenará dicha información. Además, para minimizar la latencia entre am-

bos puntos, ambos extremos están conectados bajo el mismo *switch*.



Fig. 4: Dispositivo de medición de resistencia, voltaje y potencia UM34C.

Para poder estudiar el consumo de energía de la propia Raspberry Pi 4 se ha usado el dispositivo UM34C (véase Figura 4). Este dispositivo USB se conecta entre la propia Raspberry y la toma de corriente para determinar cuánta energía consume mientras se está usando durante el envío de datos. Además, ofrece la posibilidad de conectarse a un dispositivo móvil o *smartphone* para enviar datos en todo momento del voltaje y la intensidad consumidos por cada segundo que esté en funcionamiento. De esta forma, podemos obtener el consumo de energía durante la ejecución de las pruebas.

Con respecto a las pruebas realizadas, estas se dividen en dos tipos diferentes. En primer lugar se ha comparado el protocolo de comunicación MQTT con servicios REST, también ampliamente usado para el envío y recepción de datos.

El segundo tipo de pruebas realizadas es determinar el uso y la sobrecarga que conllevaría usar MQTT dentro del sistema de ficheros Expand. De esta forma se compara la implementación que posee ya Expand con conexión vía sockets TCP/IP o la nueva implementación que usa Mosquitto.

##### MQTT vs. REST

En este estudio, se llevaron a cabo diversas pruebas para evaluar el rendimiento de la comunicación entre una Raspberry Pi 4 y una máquina de propósito general (que hace de servidor) utilizando los protocolos MQTT y REST.

El objetivo principal fue analizar el comportamiento de ambos protocolos al enviar un volumen determinado de datos. Se realizaron un total de 1.000 envíos de paquetes, variando su tamaño desde 128B hasta 2KB, con el propósito de simular escenarios de uso diversificados y representativos de situaciones del mundo real. Además, se ajustaron otros parámetros relevantes, como el tiempo de espera y los límites de reintentos, para optimizar la comunicación en ambos casos.

El estudio se realizó con una Raspberry Pi 4 como cliente y una máquina de propósito general como servidor. Para el protocolo MQTT, se configuraron

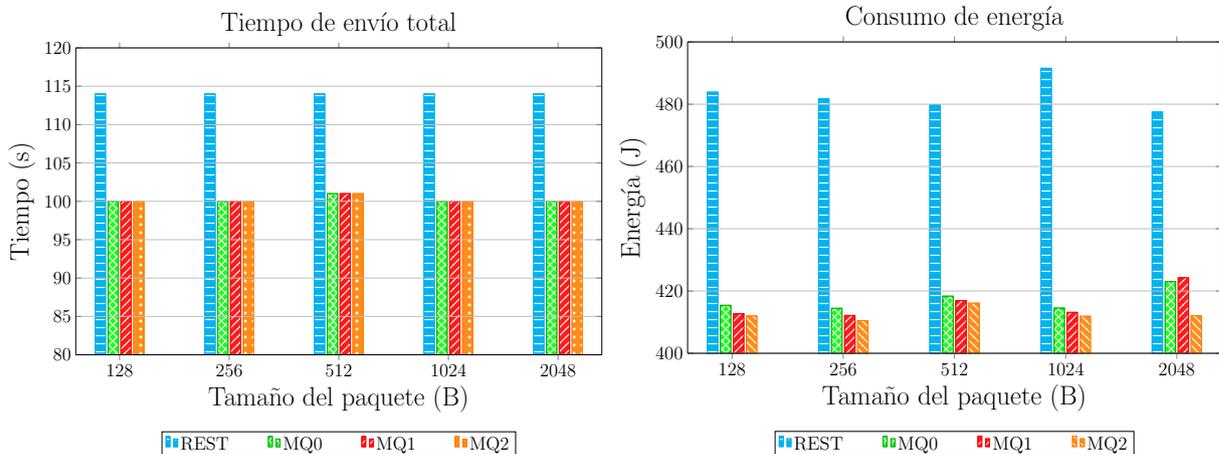


Fig. 5: Tiempo y energía consumida de MQTT y REST en la Raspberry Pi 4.

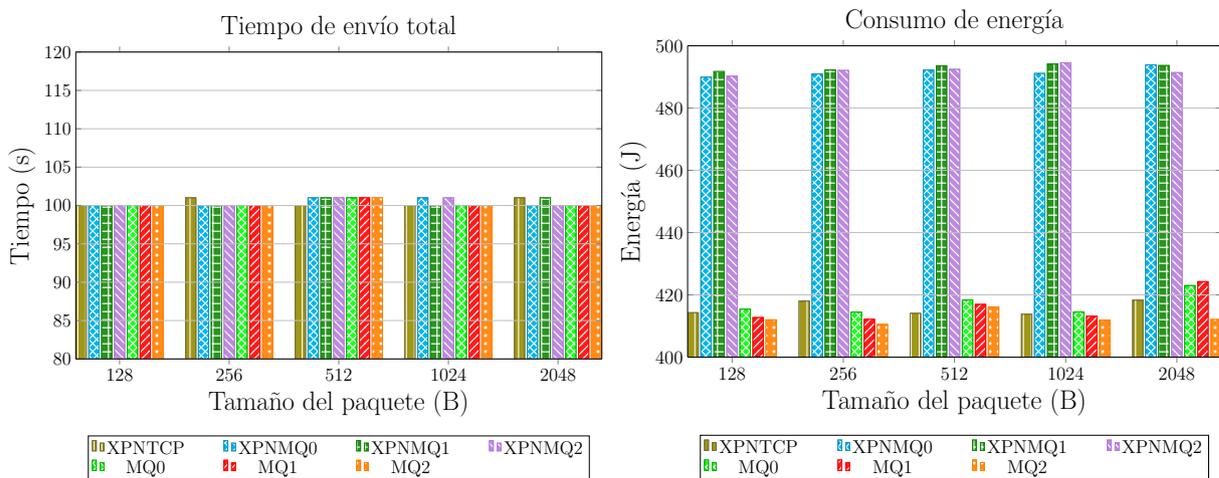


Fig. 6: Tiempo y energía consumida por Expand (TCP y MQTT) y MQTT en una Raspberry Pi 4.

los parámetros de calidad de servicio (QoS 0, 1 y 2) y se establecieron los temas o *topics* correspondientes para la transmisión de datos. En el caso de REST, se definieron las rutas y los métodos HTTP pertinentes para gestionar las solicitudes entrantes.

Estas pruebas se llevaron a cabo en un entorno controlado para minimizar las interferencias externas y garantizar la coherencia de los resultados.

Como se puede comprobar en la Figura 5 se hace una comparativa entre ambas implementaciones usadas entre la Raspberry Pi y la máquina servidora. El objetivo de esta prueba ha sido determinar la sobrecarga de la transmisión de datos y analizar cuánto se tarda en enviar la totalidad de los mensajes, además de, cuánto consume usar un protocolo u otro. En dicha Figura se puede observar tanto el tiempo de ejecución y consumo de energía de REST, como de la implementación de Mosquitto con los diferentes niveles de Calidad de Servicio (MQ0, MQ1 y MQ2).

Usar REST como método de envío es más lento y, por ende, conlleva a que el dispositivo consuma más energía a lo largo del tiempo en comparación a lo que tarda MQTT, puesto que el lado cliente envía los datos sin esperar respuesta del servidor o *broker* para enviar el siguiente bloque de información. Además, la información enviada mediante REST es

habitualmente texto (se ha de convertir la información a texto) mientras que MQTT admite el uso de una representación binaria (lo que podría permitir compactar la cantidad de datos mandados, así como el tiempo de ordenación de los datos).

#### Expand (MQTT y TCP) vs. MQTT

De la misma forma que se ha analizado en el apartado anterior, se ha querido estudiar el comportamiento del protocolo MQTT dentro del sistema de ficheros Expand explicado en anteriores líneas. El objetivo de estas pruebas es el de comprobar la sobrecarga que supondría (en tiempo y energía) utilizar la implementación de MQTT dentro de un sistema de almacenamiento como es Expand.

Para ello, se ha querido estudiar qué sobrecarga tendría utilizar un protocolo de comunicación como es MQTT dentro del sistema de ficheros Expand (XPNMQ0, XPNMQ1 y XPNMQ2 en la Figura 6) junto con una implementación que ya posee dicho sistema basado en sockets de TCP/IP (XPNTCP). Además, para saber si existe un incremento con respecto a usar solamente una implementación de MQTT usando Mosquitto, también se han añadido dichos resultados (MQ0, MQ1 y MQ2).

De la misma forma que se ha realizado anteriormente, se han utilizado paquetes de datos desde los

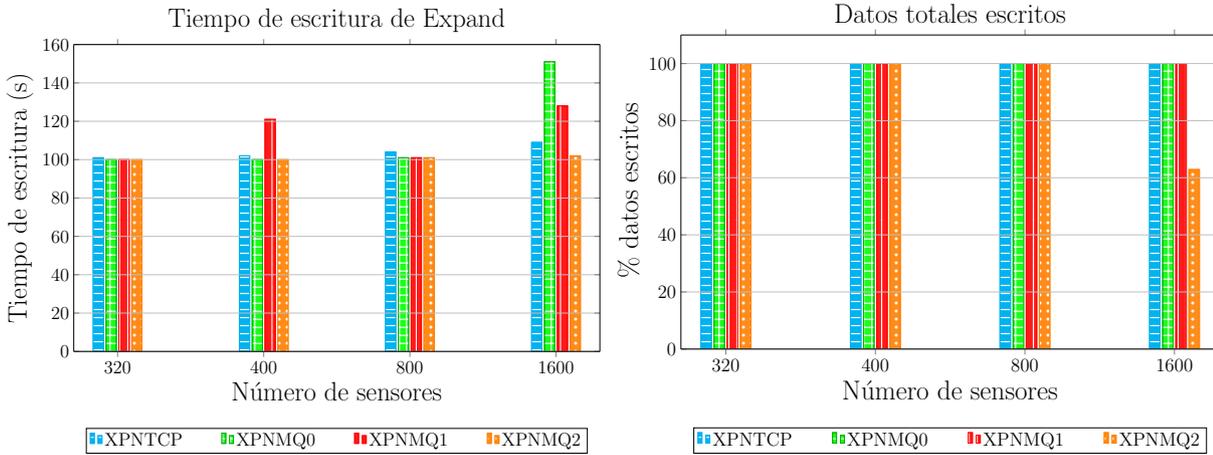


Fig. 7: Datos escritos y tiempo de escritura de Expand en un entorno emulado.

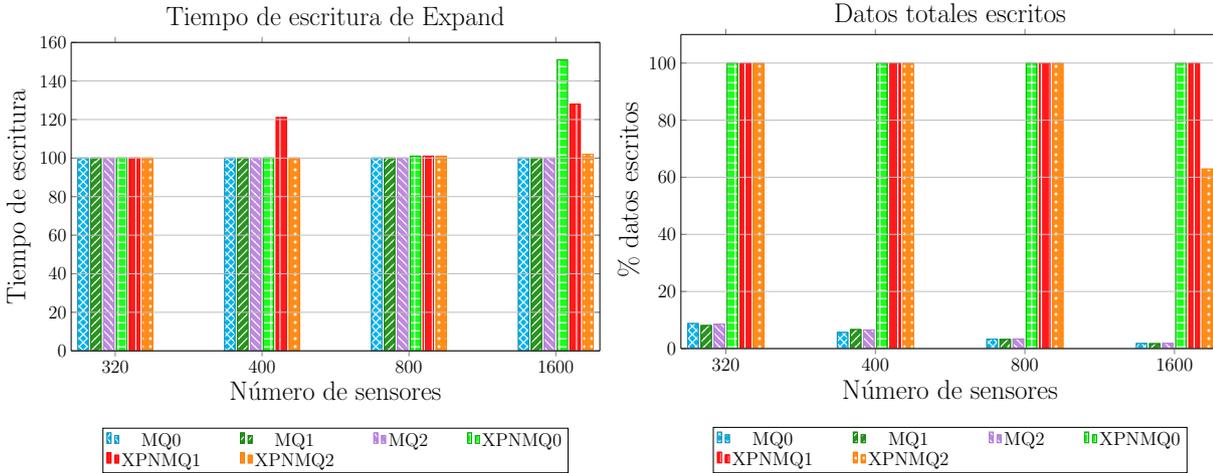


Fig. 8: Datos escritos y tiempo de escritura de una aplicación que usa solo MQTT vs una aplicación que usa Expand-MQTT, en un entorno emulado.

128B hasta los 2KB, con un intervalo predefinido de envío, con el objetivo de analizar el consumo de energía de esta infraestructura en un periodo más largo de tiempo.

Como se puede comprobar en la Figura 6, tras realizar las pruebas de envío de datos usando el sistema de ficheros, se aprecia un incremento en el consumo de energía si se usa la implementación Mosquitto de MQTT en Expand.

No obstante, aunque sí que es cierto que aumenta aproximadamente en 60 Julios el uso de un sistema de ficheros, la cantidad de datos escritos en dicho sistema es mucho mayor en comparación a si se usase una implementación tradicional de *publisher/subscriber* donde se pierden más de la mitad de la información transmitida. Además, aunque usar MQTT dentro de Expand suponga un consumo mayor que usar sockets de TCP/IP, el tiempo de envío de envío es el mismo, por lo que no supone una sobrecarga ni ralentiza el envío de datos, permitiendo que sea una alternativa a usar.

### B. Entorno emulado

Tras haber estudiado la posibilidad de llevar a un dispositivo de menor capacidad como es una Raspberry Pi un sistema de ficheros dedicado a los en-

tornos HPC, se quiso determinar la escalabilidad del sistema. Se buscaba determinar el comportamiento del sistema de ficheros en condiciones en las que la cantidad de elementos en el IoT y su capacidad fuera elevada. Para ello, dicho estudio se ha realizado en un supercomputador, aumentando así las capacidades del sistema para poder comprobar su escalabilidad a medida que se aumentan los dispositivos conectados.

Para esta subsección, hemos querido comparar el tiempo y la tasa de transferencia de datos con dos tipos de infraestructuras. La primera de las infraestructuras a comparar es la nueva arquitectura propuesta en este trabajo, en la que los sistemas IoT hacen uso de las capacidades que ofrece Expand para almacenar directamente la información en la capa *Edge*, sin intermediarios, utilizando una interfaz POSIX (véase Figura 2).

Mientras que el segundo tipo de infraestructuras a evaluar es el tipo clásico de entornos y arquitecturas donde los sistemas o dispositivos IoT no almacenan directamente la información sino que ésta pasa por un nivel intermedio en la infraestructura *Edge-Cloud* para ser almacenada en los nodos correspondientes (véase Figura 3).

Como se ha comentado, el objetivo de esta evalua-

ción es determinar cuál de estos dos tipos de infraestructuras es más beneficiosa para un sistema IoT como el que se propone en este trabajo.

Se ha evaluado estas infraestructuras emulando el comportamiento de los sensores en el cluster *HPC4AI Laboratory* en Torino. Las especificaciones de este clúster se describen en la Tabla I.

Tabla I: Resumen de las especificaciones del superordenador.

Atributo	HPC4AI Laboratory [35]
Número de nodos	68 nodos
Proveedor	Lenovo NeXtScale nx360 M5
Red	100 Gb Omni-Path
CPU por nodo	2 x Intel Xeon CPU E5-2697 v4 36 procesadores, a 2.3 GHz
SSD por nodo	110 GiB
RAM por nodo	125 GiB
Sistema Operativo	Ubuntu 20.04.5 LTS
Distribución MPI	MPICH 4.1.1
Versión GCC	9.4.0
Versión SLURM	22.05.5

Las pruebas que se han realizado en este entorno emulado son las siguientes:

1. Envío de 1.000 paquetes de 128B entre clientes y servidores con intervalos de envío de 100ms entre ellos usando Expand TCP y Expand MQTT con las diferentes calidades de servicio (Figura 2).
2. Envío de 1.000 paquetes de 128B entre clientes y servidores con intervalos de envío de 100ms entre ellos usando una implementación a 3 niveles con MQTT y Expand TCP como aparece en la Figura 3.
3. Envío de 1.000 paquetes de 128B entre clientes y servidores con intervalos de envío de 100ms entre ellos usando una implementación a 3 niveles con REST y Expand TCP.

Cada sensor se ha modelado como un proceso independiente que ejecuta sobre uno de los nodos de este *cluster*. Para el sistema de ficheros, se han empleado servidores reales que ejecutan también en este *cluster*. El objetivo de estas pruebas es comprobar si, con un aumento del número de sensores/dispositivos que envían información al sistema de ficheros, el sistema de almacenamiento es capaz de soportar la carga y almacenar los datos usando la implementación de MQTT dentro de Expand, o usando una infraestructura tradicional con MQTT o REST siendo usado fuera del sistema de almacenamiento.

En la Figura 7 se muestra la cantidad, en porcentaje, de datos escrito por Expand en los nodos de almacenamiento tras recibir todos los datos que se supone que deberían enviar los sensores, usando tanto la implementación de Expand de TCP (XPNTCP), como la nueva implementación de MQTT (XPNMQ0, XPNMQ1 y XPNMQ2). Si se tienen un total de 320 sensores que deberían enviar 1.000 paquetes de 128 B cada uno, se tiene que se deberán escribir un total de 40'96 MB de datos, y así sucesivamente conforme se aumentan los sensores de la infraestructura.

Como se puede ver en la Figura 7, usar MQTT

dentro del sistema de ficheros no supone un aumento de carga o de tiempo significativo y, además, se escribe en casi la totalidad de casos, a excepción de XPNMQ2 con 1600 sensores donde se escribe un 60 % aproximadamente.

Por otro lado, si se desea estudiar si MQTT es mejor o peor si se usa a dos o tres niveles, en la Figura 8 se puede ver que usando MQTT fuera de Expand (MQ0, MQ1 y MQ2) el tiempo de escritura es menor en muchos casos, pero se pierden más del 80 % de los datos en todos los casos.

Como se puede ver, no aparece en las pruebas realizadas los resultados de REST. El motivo es que esta implementación no es capaz de soportar el envío de datos por parte de estas cantidades de sensores, perdiendo prácticamente la totalidad de los datos y obteniendo unos tiempos de escritura muy altos en comparación con el resto de resultados mostrados.

### C. Entorno simulado en ENIGMA

Para finalizar este apartado de evaluación, se ha estudiado la escalabilidad del protocolo MQTT en entornos HPC utilizando el simulador ENIGMA [19][20].

ENIGMA<sup>2</sup> (*gENeric Iot edGe siMulAtor*) es un simulador de Edge y Fog Computing que ha sido desarrollado por los autores de este trabajo con el objetivo de realizar simulaciones de este tipo de infraestructuras para la obtención de datos, como el consumo de CPU o el consumo energético. El uso de este simulador ayuda a obtener información esencial para seleccionar un entorno que satisfaga las necesidades del usuario.

ENIGMA utiliza como base la herramienta SimGrid [21], que ofrece la posibilidad de enviar paquetes de datos entre los componentes como tareas a procesar durante la simulación. Se ha utilizado SimGrid para ENIGMA porque añade dos capas de complejidad, una para aquellos que no conozcan o no quieran conocer los detalles internos del simulador, facilitando así su uso y reduciendo su complejidad, y otra capa para aquellos que quieran profundizar en las simulaciones e infraestructuras, permitiendo modificar y añadir más elementos a la simulación.

Por estos motivos, en las siguientes líneas, se mostrará el proceso de validación del protocolo MQTT en ENIGMA y posteriormente se diseña un caso a mayor escala para determinar si es posible utilizar este protocolo en entornos mucho mayores.

### Validación de Expand MQTT en ENIGMA

Como se ha dicho en anteriores líneas, para la validación de los resultados obtenidos en Expand con MQTT en este trabajo se ha decidido utilizar ENIGMA, un simulador para plataformas Edge y Fog Computing que utiliza SimGrid [21] para la simulación de aplicaciones distribuidas en entornos heterogéneos.

Se medirá el uso de la red así como la tendencia a utilizar una aplicación que envía mensajes a través

<sup>2</sup><https://github.com/ENIGMA-Sim/ENIGMA>

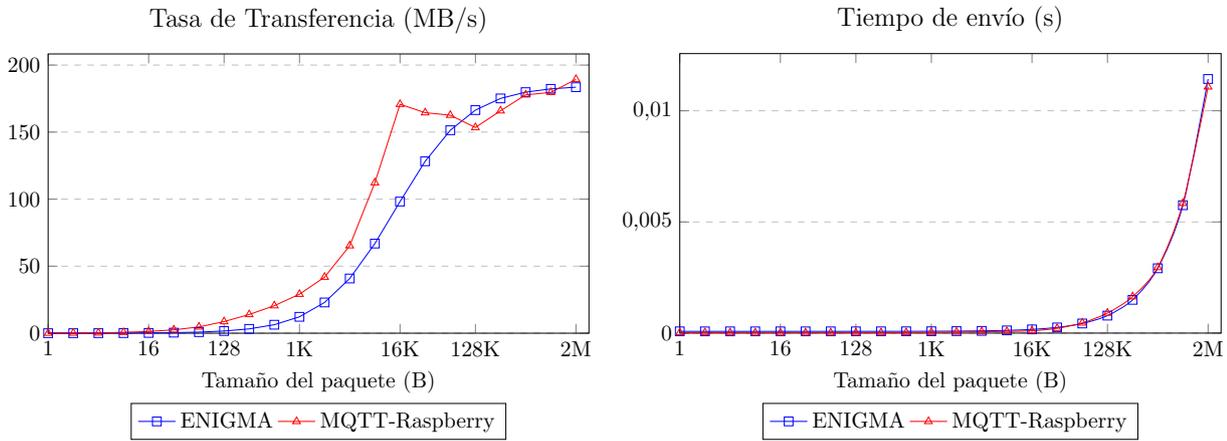


Fig. 9: Ancho de banda utilizando el protocolo MQTT en un entorno real y simulado en ENIGMA.

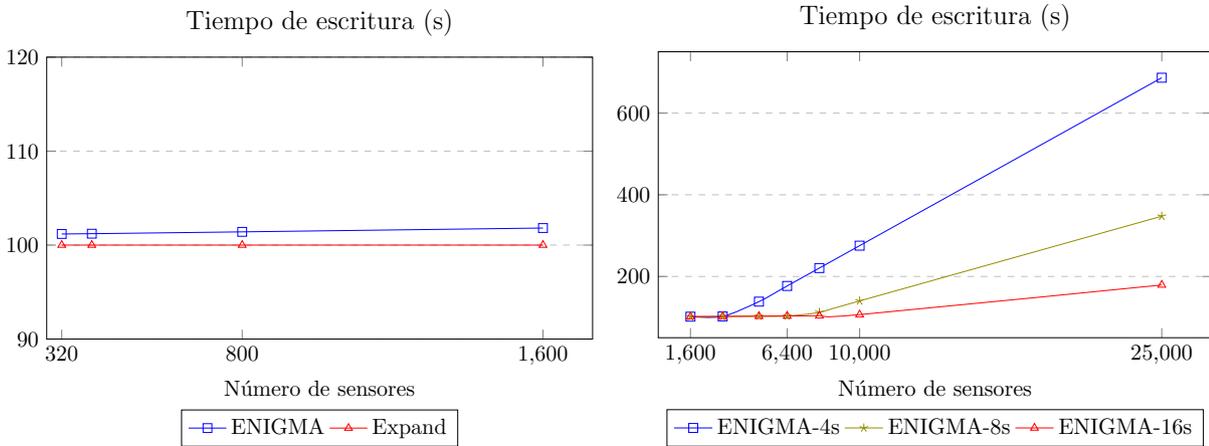


Fig. 10: Tiempo de escritura de los datos usando Expand frente a la simulación mostrada por ENIGMA (izq.) y tiempo de escritura simulado por ENIGMA conforme aumenta el número de sensores en la infraestructura (der.).

de MQTT con Mosquitto y compararlo con la simulación de una plataforma de dos actores en ENIGMA que actuará como IoT y *broker*.

Para las pruebas se ha usado una Raspberry Pi 4 que está conectada por cable a una máquina de propósito general que tendrá un proceso *broker* escuchando en todo momento las publicaciones de la Raspberry Pi, de la misma forma que se ha usado en la Subsección IV-A, explicada previamente.

La Raspberry Pi, por su parte, publicará mensajes de entre 1B y 2 MB de información. Para garantizar la fiabilidad del sistema, cada paquete se enviará un total de 100 veces y se obtendrá el tiempo medio que se tarda en enviar los datos.

Del mismo modo, se ha diseñado la misma aplicación en ENIGMA con una plataforma formada por un nodo IoT que publicará o enviará la misma cantidad de información que la aplicación Raspberry. En la Figura 9 se muestran los resultados obtenidos tras realizar estas pruebas.

Como se puede observar en la Figura 9, a la izquierda se muestra la tasa de transferencia en MB/s que la Raspberry Pi tarda en enviar a la máquina y, a su vez, la tasa simulada por ENIGMA con los diferentes tamaños de paquetes a publicar. En la parte derecha de la Figura, se puede ver el tiempo que tardan tanto la Raspberry Pi como el IoT ENIGMA en

publicar el paquete.

#### Simulación a gran escala

Para finalizar la evaluación de este trabajo, se ha realizado un último conjunto de pruebas en ENIGMA. Estas pruebas consisten en determinar y estudiar cuál sería el comportamiento del sistema de ficheros Expand, más concretamente, cuánto se tardaría en escribir la información generada conforme aumenta en la infraestructura el número de sensores que actuarían en el envío de datos.

Para dichas pruebas se ha partido de los resultados de tiempo mostrados en la Figura 8, empezando por un total de 320 sensores que actúan en la infraestructura, hasta un total de 25.000 sensores participantes. Además, se ha aumentado el número de servidores presentes en la infraestructura, desde los 4 iniciales (usados también en el supercomputador) hasta los 16 nodos.

Como se puede ver en la Figura 10, a la izquierda se muestra la comparación de tiempos de escritura real frente a lo simulado por ENIGMA, mostrando que los tiempos son lo más similares posibles entre ambos. Por otro lado, en la parte derecha de la Figura se muestra cómo iría aumentando los tiempos de escritura en los nodos servidores de Expand a medida que participan más sensores en el envío de datos,

aumentando de forma lineal dicha escritura. Otro aspecto importante a destacar es que, conforme aumenta el número de nodos servidores en la simulación, el tiempo de escritura de los datos se reduce en gran medida.

El empleo de este simulador permite hacer una estimación de la infraestructura de almacenamiento que sería necesaria para poder atender a un número determinado de sensores. Como se puede ver en las figuras anteriores, 4 servidores son suficientes para dar servicio hasta 2000 sensores. Cuando se incrementa el número de sensores es necesario incrementar el número de servidores para evitar que estos se conviertan en un cuello de botella.

## V. CONCLUSIONES Y TRABAJOS FUTUROS

En este trabajo se ha presentado la evaluación del rendimiento de un sistema de ficheros paralelo y distribuido llamado Expand para que pueda ser usado en sistemas e infraestructuras del Internet de las Cosas o IoT.

El artículo ha descrito la incorporación en Expand del protocolo MQTT, que permite que las aplicaciones que ejecutan en los sensores puedan escribir datos en ficheros remotos utilizando este protocolo, especialmente pensado para este tipo de infraestructuras. En la evaluación se ha trabajado con tres entornos: entorno real sobre un *cluster* reducido de Raspberry Pi 4, un entorno emulado sobre un *cluster* real y un entorno simulado mediante ENIGMA. Este nuevo sistema de ficheros permite simplificar el desarrollo de aplicaciones IoT, ya que se pueden escribir como cualquier tipo de aplicación que escribe en ficheros, sin necesidad de recurrir a tecnologías específicas.

El trabajo futuro se va a centrar en conectar Expand con servicios de almacenamiento *Cloud*, de forma que se puedan tener niveles jerárquicos de almacenamiento, cubriendo, de esta forma, el almacenamiento *Fog* en Expand y el almacenamiento posterior en la nube, todo ello de forma transparente a las aplicaciones que ejecutan en los dispositivos IoT.

## AGRADECIMIENTOS

Este trabajo ha sido parcialmente apoyado por el proyecto de la Agencia Estatal de Investigación “Expand - Sistema de almacenamiento de altas prestaciones para entornos HPC y *Big Data*” (Ref. TED2021-131798B-I00) y el proyecto “Nuevas técnicas escalables de E/S para cargas de trabajo híbridas de HPC e intensivas en datos (SCIOT)” con referencia PID2022-138050NB-I00.

## REFERENCIAS

- [1] Viacheslav Dubeyko, Vladimir Kuznetsov, and Natalia Kuznetsova, “Comparative analysis of distributed and parallel file systems’ internal techniques,” *arXiv preprint arXiv:1904.03997*, 2019.
- [2] Frank Schmuck and Robert Haskin, “Gpfs: A shared-disk file system for large computing clusters,” in *Proceedings of the 1st symposium on Operating systems design and implementation*, 2002, pp. 231–244.
- [3] Konstantin Shvachko, Hairong Kuang, Sanjay Radia, and Robert Chansler, “The hadoop distributed file system,” *IEEE Symposium on Mass Storage Systems and Technologies*, pp. 1–10, 2010.
- [4] Philip Carns, Walter B Ligon III, Robert B Ross, and Rajeev Thakur, “Pvfs: A parallel file system for linux clusters,” *Computing in science & engineering*, vol. 2, no. 2, pp. 60–65, 2000.
- [5] Peter Braam, “The lustre storage architecture,” *CoRR*, vol. abs/1903.01955, 2019.
- [6] Anthony JG Hey, Stewart Tansley, Kristin Michele Tolle, et al., *The fourth paradigm: data-intensive scientific discovery*, vol. 1, Microsoft research Redmond, WA, 2009.
- [7] Ashish Singh, Suresh Chandra Satapathy, Arnab Roy, and Adnan Gutub, “Ai-based mobile edge computing for iot: Applications, challenges, and future scope,” *Arabian Journal for Science and Engineering*, pp. 1–31, 2022.
- [8] Joey Li, Munur Sacit Herdem, Jatin Nathwani, and John Z Wen, “Methods and applications for artificial intelligence, big data, internet-of-things, and blockchain in smart energy management,” *Energy and AI*, p. 100208, 2022.
- [9] Mumtaz Karatas, Levent Eriskin, Muhammet Deveci, Dragan Pamucar, and Harish Garg, “Big data for healthcare industry 4.0: Applications, challenges and future perspectives,” *Expert Systems with Applications*, vol. 200, pp. 116912, 2022.
- [10] Ladjel Bellatreche, Carlos Ordóñez, Dominique Méry, Matteo Golfarelli, et al., “The central role of data repositories and data models in data science and advanced analytics,” 2022.
- [11] Gideon Juve, Ann Chervenak, Ewa Deelman, Shishir Bharathi, Gaurang Mehta, and Karan Vahi, “Characterizing and profiling scientific workflows,” *Future generation computer systems*, vol. 29, no. 3, pp. 682–692, 2013.
- [12] Bam Bahadur Sinha and R Dhanalakshmi, “Recent advancements and challenges of internet of things in smart agriculture: A survey,” *Future Generation Computer Systems*, vol. 126, pp. 169–184, 2022.
- [13] Karen Rose, Scott Eldridge, and Lyman Chapin, “The internet of things: An overview,” *The internet society (ISOC)*, vol. 80, pp. 1–50, 2015.
- [14] Dinh C Nguyen, Ming Ding, Pubudu N Pathirana, Aruna Seneviratne, Jun Li, Dusit Niyato, Octavia Dobre, and H Vincent Poor, “6G internet of things: A comprehensive survey,” *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 359–383, 2021.
- [15] Alexander Afanasyev, Neil Tilley, Peter Reiher, and Leonard Kleinrock, “Host-to-host congestion control for TCP,” *IEEE Communications surveys & tutorials*, vol. 12, no. 3, pp. 304–342, 2010.
- [16] Dan Dinculeană and Xiaochun Cheng, “Vulnerabilities and limitations of MQTT protocol used between IoT devices,” *Applied Sciences*, vol. 9, no. 5, pp. 848, 2019.
- [17] F. Garcia-Carballeira, A. Calderon, J. Carretero, J. Fernandez, and J. M. Perez, “The design of the expand parallel file system,” *The International Journal of High Performance Computing Applications*, vol. 17, no. 1, pp. 21–37, 2003.
- [18] Félix García-Carballeira, Jesús Carretero, Alejandro Calderón, J. Daniel García, and Luis. M. Sanchez, “A global and parallel file system for grids,” *Future Generation Computer Systems*, vol. 23, no. 1, pp. 116–122, 2007.
- [19] Elías Del-Pozo-Puñal, Félix García-Carballeira, and Diego Camarmas-Alonso, “A scalable simulator for cloud, fog and edge computing platforms with mobility support,” *Future Generation Computer Systems*, vol. 144, pp. 117–130, 2023.
- [20] Elías Del-Pozo-Puñal and Félix García Carballeira, “A generic simulator for edge computing platforms,” 2020.
- [21] Henri Casanova, Arnaud Giersch, Arnaud Legrand, Martin Quinson, and Frédéric Suter, “Versatile, scalable, and accurate simulation of distributed applications and platforms,” *Journal of Parallel and Distributed Computing*, vol. 74, no. 10, pp. 2899–2917, June 2014.
- [22] Dave Evans, “Internet de las cosas,” *Cómo la próxima evolución de Internet lo cambia todo. Cisco Internet Business Solutions Group-IBSG*, vol. 11, no. 1, pp. 4–11, 2011.
- [23] Rolf H Weber, “Internet of things—new security and privacy challenges,” *Computer law & security review*, vol. 26, no. 1, pp. 23–30, 2010.
- [24] Jordi Salazar and Santiago Silvestre, “Internet de las

- cosas,” *Techpedia. České vysoké učení technické v Praze Fakulta elektrotechnická*, 2016.
- [25] Laith Farhan, Rupak Kharel, Omprakash Kaiwartya, Marcela Quiroz-Castellanos, Ali Alissa, and Mohamed Abdulsalam, “A concise review on internet of things (iot)-problems, challenges and opportunities,” in *2018 11th International Symposium on Communication Systems, Networks & Digital Signal Processing (CSNDSP)*, 2018, pp. 1–6.
- [26] Damian Arellanes and Kung-Kiu Lau, “Evaluating iot service composition mechanisms for the scalability of iot systems,” *Future Generation Computer Systems*, vol. 108, pp. 827–848, 2020.
- [27] Paolo Bellavista and Alessandro Zanni, “Towards better scalability for iot-cloud interactions via combined exploitation of mqtt and coap,” in *2016 IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI)*, 2016, pp. 1–6.
- [28] Olivier Debauche, Saïd Mahmoudi, and Adriano Gutta-dauria, “A new edge computing architecture for iot and multimedia data management,” *Information*, vol. 13, no. 2, pp. 89, Feb 2022.
- [29] Dinesh Thangavel, Xiaoping Ma, Alvin Valera, Hwee-Xian Tan, and Colin Keng-Yan Tan, “Performance evaluation of mqtt and coap via a common middleware,” in *2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, 2014, pp. 1–6.
- [30] Tim Pulver, *Hands-on internet of things with MQTT : build connected IoT devices with Arduino and MQ telemetry transport (MQTT)*, Packt Publishing, 1st edition, 2019.
- [31] Filippo Antonielli, “Development and comparison of mqtt distributed algorithms for hivemq,” 2021.
- [32] Souvik Pal, Vicente Garcia Diaz, and Dac-Nhuong Le, “Iot: Security and privacy paradigm,” 2022.
- [33] Binildas Christudas, “Activemq,” 2019.
- [34] Eclipse Mosquitto, “An open source mqtt broker,” *Eclipse Mosquitto™[cit. 2018-04-23]. Dostupné z: Mosquitto.org*, 2018.
- [35] HPC4AI, “HPC4AI Laboratory specification,” 2023, Accessed Jul. 3, 2023. [Online].