




RENDIMIENTO PARA LA INTEROPERABILIDAD ENTRE RASPBERRY PI, ESP8266 Y PLC CON NODE-RED PARA EL IIoT

PERFORMANCE FOR INTEROPERABILITY BETWEEN RASPBERRY PI AND ESP8266 WITH A PLC IN A NODE-RED SERVER FOR IIoT

J. Torres Ventura^{1,*} , A. H. Ruelas Puente² , J. R. Herrera García¹ 

Recibido: 21-06-2022, Recibido tras revisión: 06-12-2022, Aceptado: 13-12-2022, Publicado: 01-01-2023

Resumen

Este trabajo evalúa la viabilidad de integrar en una red descentralizada las placas de bajo costo Raspberry pi, microcontroladores ESP8266 con equipos industriales de Controladores Lógicos Programables (PLC). Estos dispositivos serán los nodos que participan en el intercambio de datos entre el piso de manufactura y los servicios empresariales de manera simple, fiable y económica. Los nodos de la red producen y consumen datos que son intercambiados por un nodo gestor de protocolos open source llamado Node-RED. El gestor de protocolos es un servidor con núcleo de Linux sobre un microprocesador RISC (Reduced instruction set computing). Los sensores en los nodos de manufactura utilizan microcontroladores SoC (System On Chip) y mediante el concepto de Edge computing adquieren, procesan y envían sus datos al gestor de protocolos. Así, mediante la herramienta Iperf3, Wireshark y MTR, mediremos el rendimiento del enlace de comunicaciones y el estado que guardan los nodos participantes en sus procesos producción/consumo de datos.

Palabras clave: IoT, interoperabilidad, producción/consumo, microservicios, enlace, distribuida

Abstract

This work evaluates the feasibility of integrating low-cost Raspberry pi boards and ESP8266 microcontrollers with industrial Programmable Logic Controllers (PLC) in a decentralized network. These devices will be the nodes that participate in the exchange of data between the manufacturing floor and the business services in a simple, reliable and economical way. The network nodes produce and consume data that is exchanged by an open-source protocol manager node called Node-RED. The protocol manager is a server with a Linux kernel on a RISC (Reduced instruction set computing) microprocessor. The sensors in the manufacturing nodes use SoC (System On Chip) microcontrollers, and through the concept of Edge computing they acquire, process and send their data to the protocol manager. Thus, the performance of the communication link and the status of the participating nodes in their data production/consumption processes will be measured using the Iperf3, Wireshark and MTR tools.

Keywords: IoT, interoperability, production/consumption, microservices, link, distributed.

^{1,*}Laboratorio de Mecatrónica, Universidad Autónoma de Baja California, México. Autor para correspondencia ✉: jose.torres.ventura@uabc.edu.mx.

²Laboratorio de Ingeniero en Computación, Universidad Autónoma de Baja California, México.

Forma sugerida de citación: Torres Ventura, J.; Ruelas Puente, A. H. y Herrera García, J. R. "Rendimiento para la interoperabilidad entre Raspberry pi, ESP8266 y PLC con Node-RED para el IIoT", *Ingenius, Revista de Ciencia y Tecnología*, N.º 29, pp. 90-97, 2023. DOI: <https://doi.org/10.17163/ings.n29.2023.08>.

1. Introducción

El concepto de IoT es el detonante de la cuarta revolución industrial [1], sin embargo, la historia comienza en 1960 con la necesidad de compartir los datos de los procesos de manufactura y producción con los servicios administrativos del negocio. A nivel industrial, el fabricante de Programmable Logic Controllers (PLC) en su proyecto Modular Digital Controller (MODICON), patenta en 1979 [2] su protocolo de intercambio de datos llamado Modbus. Las computadoras personales contemporáneas requerían también ser partícipes de esta información para la administración de operaciones del proceso industrial. En 1983, Microsoft, principal distribuidor de equipos de cómputo personal, introduce el protocolo Distributed Component Object Model (DCOM) [3], el cual ha evolucionado hasta Object Linking and Embedding for Process Control (OPC) [4].

El proceso productivo utiliza las plataformas Manufacturing Execution Systems (MES) [5], que son complejos sistemas informáticos modulares que gestionan el control de flujo de materia prima, inventario en proceso, órdenes de pedidos, lista de materiales, defectos y producto terminado entre otros. El monitoreo y control del desempeño de maquinaria y equipamiento del piso de manufactura se realiza mediante especializados sistemas de hardware y software denominados Supervisory Control And Data Acquisition (SCADA) [6]. Estos ejecutan modelos como Key Production Index (KPI) y el Overall Equipment Effectiveness (OEE). Los sistemas MES y SCADA, se agrupan en una plataforma llamada Computer Integrated Manufacturing (CIM) [7].

Por otro lado, la interoperabilidad entre los diferentes departamentos administrativos del negocio se realiza con la técnica llamada Enterprise Service Bus (ESB) [8], cuya función es controlar el proceso de flujo de datos entre los diferentes nodos de la red empresarial, apoyado en estructuras de datos con múltiples tipos de Application Programming Interface (API). De la misma manera, estos sistemas del negocio se integran en una plataforma llamada Enterprise Resource Planning (ERP) [9]. El desarrollo de este trabajo se presenta con la siguiente estructura: la sección II describe la metodología para medir el rendimiento y el estado del intercambio de datos de una red distribuida, orientada a microservicios de producción/consumo entre los nodos de la red local. La sección III presenta los resultados del análisis y diagnóstico de la red, midiendo los parámetros de transferencia y latencia del flujo de datos entre los nodos.

La sección IV expone la conclusión de la investigación y algunas áreas de oportunidad. El producto de este trabajo es evaluar de manera descriptiva la confiabilidad de la integración de las placas de desarrollo Raspberry pi y ESP8266 a las tecnologías industriales tradicionales en el rubro intercambio de

datos. Así, podremos ofrecer una alternativa de interoperabilidad confiable, de bajo costo y con una curva de aprendizaje relativamente baja. Esto permitirá a las plataformas CIM y ERP llevar a cabo sus procesos orientados a microservicios. Históricamente, la necesidad de intercambiar datos entre los diferentes departamentos administrativos del negocio y el piso de producción de la manufactura, ha sido un tema delegado a los grandes gurús de los sistemas informáticos como SAP [10], Oracle ERP [11], Microsoft Dynamics 365 [12], entre otros.

El objetivo de estos poderosos sistemas informáticos está orientado a atender servicios que llegan a incrementarse de tal manera que exigen de muchos recursos, incremento de código, pero, sobre todo, altos costos de inversión, operación y mantenimiento. La estrategia que proponemos en esta investigación se basa en la naturaleza de operaciones de los nodos industriales, orientada a los microservicios por su aplicación solo en funciones específicas, que permite una escalabilidad que depende del hardware y no del software. Por esta razón iniciamos una búsqueda de cómo se ha venido desarrollando el tema de intercambio de datos entre el piso de producción y el negocio.

1.1. Intercambio de datos entre los departamentos del negocio

En los sistemas informáticos para los servicios de negocio empresarial, se utilizan protocolos de intercambio de datos independientes de la plataforma tecnológica o tipo de servicio, así surge Simple Object Access Protocol (SOAP) que utiliza lenguaje de marcas extensible Markup Language (XML) [13]. Este protocolo genera documentos de texto Web Services Description Language (WSDL) [14] a manera de ofertar un contrato de servicio a los participantes. Posteriormente, se logran mejoras con Representational State Transfer (RESTful) basadas en obtener de manera segura una dirección virtual sin necesidad de contratos

1.2. Intercambio de datos entre los sistemas de manufactura

En 1994, a nivel internacional, la Object Linking and Embedding for Process Control Foundation (OPC foundation) [15], presenta los OPC Server y OPC Cliente como un modelo para normar la interoperabilidad en el sector industrial. Los OPC son librerías de software anidados en los equipos de automatización y en computadoras personales. Los costos varían en función de cada producto y cantidad de etiquetas, aunque también se ofrecen algunas versiones de demostración con límite de etiquetas y tiempo de servicio. En la actualidad Algunos desarrolladores de tecnologías como IBM y Eclipse han promovido ecosistemas Open Source con este modelo, así es como surge OPC Unification

Automation (OPC UA) [16] y Node-RED [17] que, además, de ser multiplataforma cuentan con soporte de comunidades Open Source.

Un gestor de protocolos industriales es un servidor que gestiona de manera centralizada el flujo de paquetes de datos en los nodos de una red, apoyándose en librerías liberadas por fabricantes de equipos de automatización industrial. El sistema informático Node-RED se incrusta perfectamente al Industrial Internet of Things (IIoT), al ser multiplataforma, multiparadigma, consumir pocos recursos, ofrecer una baja curva de aprendizaje, pero, sobre todo, ofrece un alto poder de procesamiento y gestión de comunicación. La revisión de la literatura muestra la constante búsqueda de soluciones flexibles para solventar, primero, la naturaleza no sincronizada del piso de producción con los tiempos del negocio y segundo, la no compatibilidad de los formatos de datos entre sus aplicativos, agregando que los protocolos de comunicación entre sus buses de redes distribuidas son atípicos. Esta complejidad justifica la intervención de un grupo interdisciplinario para tratar de solventar estos contratiempos, derivando en altos costos, una prolongada curva de aprendizaje y un nivel de escalabilidad relativamente bajo, retos que prevalecen hasta nuestros días.

2. Materiales y métodos

La aportación de este trabajo es proporcionar a los investigadores y desarrolladores de tecnología en el ámbito de las TIC (Tecnologías de la informática y las comunicaciones), una referencia a partir de un sistema de producción real, una alternativa para realizar el intercambio de datos producidos y consumidos entre los nodos de una red distribuida, orientada a tender sistemas CIM y ERP de manera confiable, de bajo costo y con una curva de aprendizaje relativamente baja. Para validar el intercambio de datos entre los nodos de la red, utilizaremos un método descriptivo, basado en medir el rendimiento de la red, mediante la latencia y la transferencia. Así, se procedió de la siguiente forma:

- Seleccionar la interface que permita la interoperabilidad entre los datos producidos/consumidos por todos los nodos de la red distribuida
- Construir una red distribuida con nodos de diferentes plataformas industriales y placas de desarrollo, para la automatización de procesos
- Crear un mapa de variables para ser empaquetadas y medidas.
- Transferir paquetes entre nodos participantes (carga máxima)

2.1. Node-RED como gestor de intercambio de datos

Node-RED es un proyecto desarrollado principalmente por IBM en 2013 y liberado en 2016 [18], es multiplataforma, esto significa que corre en Windows, Macintosh, núcleos de Linux y Solaris. Al ser una herramienta de programación por flujo visual, permite ser utilizado por personal no especializado, inclusive por entusiastas y creativos. Desarrollado para consumir pocos recursos de cómputo, nos permite crear un servidor de red para gestionar la conexión con dispositivos del IIoT con arquitecturas ARM (Advanced RISC Machine) y plataformas RISC.

2.2. Desplegando la red distribuida

En la Figura 1 se presenta los nodos participantes en la red distribuida de producción instalada para realizar el intercambio de datos. A nivel de capa física, el enlace entre el Node-RED server utiliza estándar IEEE.802.11, al igual que el ESP-8266 MQTT Relay. Todos los demás utilizan estándar IEEE.802.3 con cableado estructurado UTP Clase 5. El Netis wifi Router comunica todos los enlaces sobre socket TCP/IP, en una frecuencia de 2.4 GHz.

Status	Name	IP	Manufacturer	MAC address
>	Netis WiFi Router	192.168.0.1	Netcore Technology Inc.	E4BEED:70:DE36
	ESP-8266 MQTT Relay	192.168.0.5	Espressif Inc.	18:FE:34:FB:2B:D7
	PLC Siemens 1200	192.168.0.20	Siemens AG	28:63:36:84:CE:2C
>	PLC Micrologix 1100	192.168.0.30	RS Automation Co., Ltd	00:0F:73:03:7B:EC
	Node-RED Server	192.168.0.100	Raspberry Pi Foundation	B8:27:EB:87:7D:AD
	MySQL DB Server	192.168.0.150	Micro-Star INT'L CO., LTD	8C:89:A5:34:5B:E3

Figura 1. Nodos participantes en la red descentralizada

A nivel de capa de aplicación Node-RED concentra y gestiona el tráfico de flujo mediante, conectores de MySQL db María con el nodo etiquetado MySQL DB Server. Con el protocolo MQTT (Message query Telemetric Transport) interactúa con el nodo etiquetado ESP-8266 MQTT Relay, finalmente, para el nodo PLC Siemens 1200 utiliza S7 protocolo y para el nodo PLC Micrologix 1100 se soporta en el protocolo PCCC.

2.2.1. Construcción de gateway con una placa raspberry pi

El gestor de flujo etiquetado como Node-RED server, utiliza un procesador ARM-v7 single Core, 1 GHz, 512 MB RAM, Wireless 2.4 GHz, IEEE802.11. Utiliza un sistema operativo con distribución Devian en base Linux, y un aplicativo de servidor Node-RED que se despliega sobre Node JS. La plataforma tecnológica de procesadores ARM utiliza muy pocas instrucciones para su uso operativo, lo que libera recursos para utilizarlos en aplicaciones del cliente, bajo consumo de energía, confiables, económicos, con baja curva de aprendizaje, además de contar con un gran soporte de comunidades open source.

2.2.2. Funcionalidad del nodo con ESP8266

El hardware para el nodo que inicia la operación de la celda de manufactura, en el piso de producción, es un dispositivo IoT que recibe el comando etiquetado start/stop. En un microcontrolador con un integrado esp8266 manufacturado por expressif basado en un CPU RISC de 32 bits, Xtensa LX106 con un reloj de 80 MHz, RAM de 64 KB, memoria flash QSPI 512 KB. Conexión de radiofrecuencia bajo la norma IEEE 802.11 b/g/n, interface de comunicaciones SPI (Serial Peripheral Interface) y I² C (Inter-Integrated Circuit).

2.2.3. Funcionalidad de nodos industriales

Los nodos industriales los etiquetamos de la siguiente manera; un PLC Siemens 1200 el cual mide la temperatura del proceso en un rango de variable de 0 a 40 °C, también se instaló un PLC AB 1100 con un proceso que despliega un ciclo de producción una vez que recibe en etiqueta de consumo de valor “true” / “false”. Por otro lado, se instaló un dispositivo microcontrolador ESP8266 para dar el start/stop en su variable toma valores de “1” / “0”. Finalmente, en una PC con sistema operativo win10 de 32 bits, se instala un servidor de base de datos relacionales MySQL, para registrar (consumir) el historial del ciclo de proceso producido en el PLC AB 1100, en un variable que se ejecuta en un rango de enteros decimales del 0 al 5.

2.3. Mapa de variables

En la Tabla 1 se muestra los nodos físicos y su identificación a nivel de la capa 3 del modelo OSI. La columna valor indica el tipo de dato transferido como carga útil (payload), y es el que será empaquetado para posteriormente ser transferido entre los microservicios. Bajo el mismo tenor, las columnas de producción y consumo están orientadas a los microservicios que tienen como referencia su IP de origen y destino.

Tabla 1. Mapa de variables utilizadas en el flujo de transferencia de datos

N.º	Produce	Consume	Valor
1	192.168.0.100	192.168.0.5	“1” / “0”
2	192.168.0.20	192.168.0.100	0-40 °C
3	192.168.0.100	192.168.0.20	True/False
4	192.168.0.30	192.168.0.150	0-5 segundos

2.4. Medir el proceso

Los índices de medida serán la tasa de transferencia, la latencia de la red y el estado del intercambio de datos. Consideramos pertinente realizar un monitoreo de referencia inicial sin carga total entre los nodos

participantes, para ver el estado que guarda la red soportada en un enlace bajo la norma IEEE 802.11 y IEEE802.3 con protocolo de transporte TCP (Transmission Control Protocol).

2.4.1. Transferencia en la red distribuida

El cálculo analítico de la tasa de transferencia se define como la velocidad a la cual podemos transferir información en un medio de transmisión de red. Esto lo podemos modelar en la siguiente ecuación (1).

$$v_t = \frac{i_t}{t} \tag{1}$$

Donde:

- v_t = velocidad de transferencia (MB/s)
- i_t = cantidad de información (MB)
- t = tiempo de transmisión (s)

El método utilizado para medir el rendimiento de la red es de tipo descriptivo. Nos apoyamos en el *software* Iperf ver.3 para el análisis transferencia de la red. En el caso de la latencia, se realizó un diagnóstico de la red con el protocolo de paquetes ICMP (Internet Control Message Protocol) mediante los métodos Echo Request y Echo Reply. Y con la herramienta Wireshark monitoreamos el estado del dato. Finalmente, con MTR analizamos los saltos del enrutamiento.

2.4.2. Latencia de la red distribuida

La latencia considera el tiempo que le toma a un bit viajar por una red desde la fuente a su destino. Analíticamente, se modelaría como se muestra en la ecuación (2).

$$Latencia = TP + TT + TC \tag{2}$$

Donde:

- TP=tiempo de propagación (s)
- TT = tiempo de transmisión (s)
- TC = tiempo de cola (s)

El tiempo de propagación es el tiempo que le toma a un bit llegar desde la fuente hasta el destino a través de un medio de transmisión específico y se mide en milisegundos (ms). El tiempo de transmisión es el tiempo para meter todos los bits de un paquete al medio de transmisión y se mide en milisegundos (ms). Finalmente, el tiempo de cola es el tiempo que requiere un equipo de comunicaciones en la transmisión, para gestionar flujos de paquetes de datos desde la entrada a la salida al medio de transmisión y se mide en milisegundos (ms).

2.4.3. Monitoreo inicial del rendimiento de la red

Para el desarrollo del experimento se obtiene una referencia del estado que guarda el enlace de la red utilizada. En la Figura 2 se muestra el flujo de datos entre el servidor MySQL y el servidor Node-RED con protocolo TCP, vemos que la velocidad de transmisión entre ellos es aceptable considerando que el medio de transmisión es entre cable UTP Cat 5 y wifi 2.4 GHz. También se aprecia que el servidor Node-RED gestiona con un 4.4 % de utilización, esto es, con carga inicial. La herramienta de diagnóstico fue Iperf3 con un cliente y servidor.

```
C:\iperf3>iperf3 -c 192.168.0.100 -v
iperf 3.0.12
Time: Thu, 02 Jun 2022 17:49:02 GMT
Connecting to host 192.168.0.100, port 5201
Cookie: DESKTOP-V7UW090-1654192142-028223.32
TCP MSS: 537334528 (default)
[ 4] local 192.168.0.150 port 63743 connected to 192.168.0.100 port 5201
Starting Test: protocol: TCP, 1 streams, 131072 byte blocks, omitting 0 seconds, 10 second test
[ ID] Interval      Transfer     Bandwidth
[ 4] 0.00-1.00  sec  3.69 MBytes  30.9 Mbits/sec
[ 4] 1.00-2.00  sec  3.88 MBytes  32.5 Mbits/sec
[ 4] 2.00-3.00  sec  3.81 MBytes  32.0 Mbits/sec
[ 4] 3.00-4.00  sec  3.69 MBytes  31.0 Mbits/sec
[ 4] 4.00-5.00  sec  4.00 MBytes  33.5 Mbits/sec
[ 4] 5.00-6.00  sec  4.25 MBytes  35.6 Mbits/sec
[ 4] 6.00-7.00  sec  4.18 MBytes  35.1 Mbits/sec
[ 4] 7.00-8.00  sec  4.37 MBytes  36.6 Mbits/sec
[ 4] 8.00-9.00  sec  4.18 MBytes  35.1 Mbits/sec
[ 4] 9.00-10.00 sec  4.18 MBytes  35.1 Mbits/sec
-----
Test Complete. Summary Results:
[ ID] Interval      Transfer     Bandwidth          sender
[ 4] 0.00-10.00  sec  40.2 MBytes  33.7 Mbits/sec      receiver
CPU Utilization: local/sender 2.3% (0.3%/2.0%), remote/receiver 5.0% (0.6%/4.4%)
iperf Done.
```

Figura 2. Monitoreo de transmisión entre servidor MySQL y el servidor Node-RED

En la Figura 3 se muestra la variación de la latencia entre los mismos nodos. Pero esta vez utilizaremos con el protocolo de transporte UDP (User Datagram Protocol), considerando que es más transparente al no contener una ventana y tener que esperar confirmación del destino.

```
C:\iperf3>iperf3 -c 192.168.0.100 -v -u
iperf 3.0.12
Time: Thu, 02 Jun 2022 17:58:41 GMT
Connecting to host 192.168.0.100, port 5201
Cookie: DESKTOP-V7UW090-1654192721-022551.68
[ 4] local 192.168.0.150 port 62677 connected to 192.168.0.100 port 5201
Starting Test: protocol: UDP, 1 streams, 8192 byte blocks, omitting 0 seconds, 10 second test
[ ID] Interval      Transfer     Bandwidth    Total Datagrams
[ 4] 0.00-1.00  sec  120 KBytes  978 Kbits/sec  15
[ 4] 1.00-2.00  sec  128 KBytes  1.05 Mbits/sec  16
[ 4] 2.00-3.01  sec  128 KBytes  1.04 Mbits/sec  16
[ 4] 3.01-4.00  sec  128 KBytes  1.06 Mbits/sec  16
[ 4] 4.00-5.01  sec  128 KBytes  1.04 Mbits/sec  16
[ 4] 5.01-6.01  sec  128 KBytes  1.05 Mbits/sec  16
[ 4] 6.01-7.01  sec  128 KBytes  1.05 Mbits/sec  16
[ 4] 7.01-8.00  sec  128 KBytes  1.05 Mbits/sec  16
[ 4] 8.00-9.01  sec  128 KBytes  1.04 Mbits/sec  16
[ 4] 9.01-10.00 sec  128 KBytes  1.05 Mbits/sec  16
-----
Test Complete. Summary Results:
[ ID] Interval      Transfer     Bandwidth    Jitter  Lost/Total Datagrams
[ 4] 0.00-10.00  sec  1.24 MBytes  1.04 Mbits/sec  3743.423 ms  0/159 (0%)
[ 4] Sent 159 datagrams
CPU Utilization: local/sender 0.8% (0.3%/0.5%), remote/receiver 0.1% (0.0%/0.1%)
iperf Done.
```

Figura 3. Muestra la variabilidad de la latencia, también llamado jitter

Podemos apreciar que la pérdida de paquetes es de 0 %, en un paquete de 8192 bytes con un ancho de banda antes del inicio de operaciones de 1.05 Mbit/s. Podemos confirmar que es un comportamiento aceptable para la transferencia como condiciones de referencia iniciales.

3. Resultados y discusión

Primeramente, mediremos la latencia de la red, considerando a todos los nodos participantes, exigidos a carga completa. En la Figura 4 se muestra una gráfica de tendencia de la latencia de ida y vuelta resuelta por el protocolo ICMP.

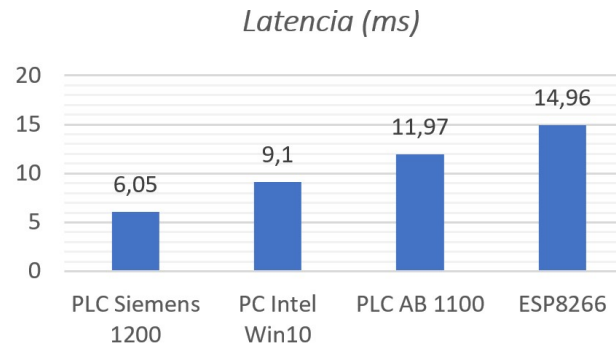


Figura 4. Diagnóstico de la latencia de ida y vuelta, utilizando paquetes ICMP

Específicamente se utilizó el comando ping desde la pantalla de comandos (Shell) del nodo servidor de Node-RED, hacia cada uno de los nodos participantes en la red. Se resolvió para un envío de diez paquetes y en todos los casos se recibieron los diez paquetes. Sin embargo, debido al incremento de la latencia, en el enlace con cada nodo, también fue prudente reportar la desviación estándar de la misma, la cual se muestra en la Figura 5.

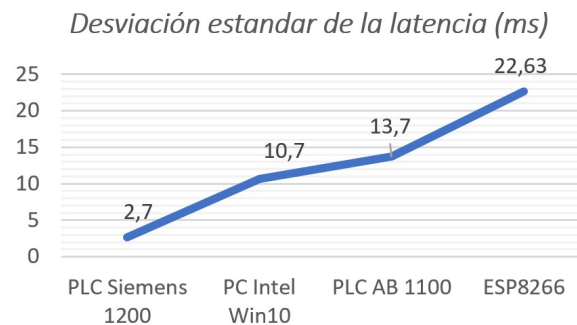


Figura 5. Diagrama de funcionamiento del horno

En el afán de encontrar alguna relación del incremento de la latencia entre el servidor de Node-RED y el nodo ESP-8266 de hasta en un 50 %, comparado con el resto de los nodos. Se realizó un análisis con la herramienta de diagnóstico MTR para descartar que fuera debido a los saltos que dan entre paquetes de la red, esto se resuelve en la Figura 6.

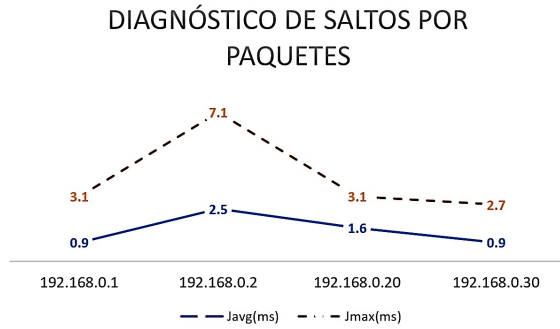


Figura 6. Diagnóstico por saltos de paquetes en cada nodo participante de la red

Donde podemos observar que desde el servidor Node-RED se enviaron diez paquetes a cada uno de los nodos y se resuelve que existe un jitter (Javg) más elevado de hasta 2.5 ms en promedio. Y nuevamente la desviación estándar (Jmax) se dispara hasta más del doble del resto de nodos participantes con 7.1 ms. Por esta razón fue también necesario verificar la transferencia del servidor Node-RED, inundando la red con diez paquetes en un protocolo UDP para exigirle al máximo al enlace wifi con un máximo de 54 Mb, lo que se resuelve en la Figura 7 siguiente.

```
iperf done.
pi@raspberrypi:~$ iperf3 -c 192.168.0.150 -u -b54m
iperf 3.6
Linux raspberrypi 4.19.57 #1294 Thu Jul 4 18:42:50 BST 2019 armv7l
Control connection MSS 1460
Setting UDP block size to 1460
Time: Fri, 03 Jun 2022 18:56:32 GMT
Connecting to host 192.168.0.150, port 5201
Cookie: b7a5c0c179d33960afad6a19447b2w
[ 5] local 192.168.0.100 port 54120 connected to 192.168.0.150 port 5201
Running Test protocol: UDP, 3 streams, 1460 byte blocks, omitting 0 seconds, 10 second test, too o
[ ID] Interval      Transfer      Bitrate      Total Datagrams
[ 5] 0.00-1.00 sec  5.29 MBytes  44.3 Mbits/sec  3796
[ 5] 1.00-2.00 sec  3.97 MBytes  32.5 Mbits/sec  2790
[ 5] 2.00-3.00 sec  4.98 MBytes  41.7 Mbits/sec  3574
[ 5] 3.00-4.00 sec  4.07 MBytes  33.2 Mbits/sec  3352
[ 5] 4.00-5.00 sec  4.83 MBytes  40.5 Mbits/sec  3470
[ 5] 5.00-6.00 sec  3.77 MBytes  31.6 Mbits/sec  2709
[ 5] 6.00-7.00 sec  4.33 MBytes  36.5 Mbits/sec  3127
[ 5] 7.00-8.00 sec  4.44 MBytes  37.2 Mbits/sec  3186
[ 5] 8.00-9.00 sec  5.17 MBytes  43.4 Mbits/sec  3716
[ 5] 9.00-10.00 sec  4.19 MBytes  35.1 Mbits/sec  3009
-----
Test Complete. Summary Results:
[ ID] Interval      Transfer      Bitrate      Jitter      Lost/Totl. Datagrams
[ 5] 0.00-10.00 sec  45.6 MBytes  38.2 Mbits/sec  0.000 ms  0/32720 (0%) sender
CPU Utilization: local/sender 41.7% (5.5%/39.3%), remote/receiver 0.4% (0.1%/0.3%)
iperf done.
```

Figura 7. Saturación del enlace wifi con 54 MB utilizando protocolo UDP

Podemos apreciar que la variabilidad de la latencia de ida y vuelta ronda los 0.396 ms, con 1.6 % de pérdidas de paquetes, considerando una utilización del procesador de lado del servidor Node-RED de 43.7 %. Finalmente, para validar el estado que muestra el intercambio de datos transmitido en la red, se ejecutó un análisis de flujo de datos con la herramienta de software Wireshark, y esto se resuelve en la Figura 8. La actualización de la etiqueta que se produce en el nodo PLC Siemens 1200, se registra dentro del servidor Node-RED con su campo de consumo «value».

Respecto a la transferencia de datos producidos por PLC AB 1100 y consumidos por el servidor de la base de datos relacionales MySQL, en la Figura 9 se observa mediante el reporte de flujo TCP realizado con Wireshark, el valor de la etiqueta en el campo «VALUES».

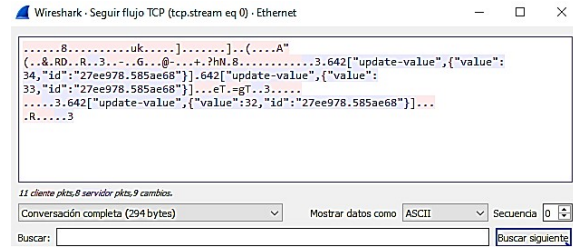


Figura 8. Transferencia del valor de la etiqueta producida en el nodo PLC Siemens1200 y consumido en el servidor Node-RED

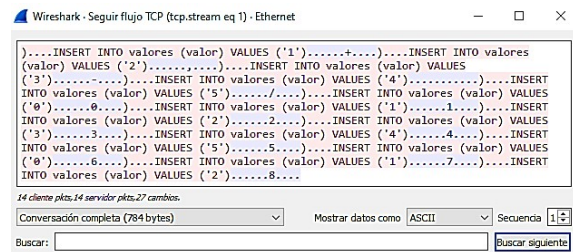


Figura 9. Transferencia del dato producido en el nodo PLC AB 1100 y consumido en el nodo Server MySQL

Resolviendo que el intercambio de datos entre los nodos y servidor gestor de protocolos Node-RED opera con un alto grado de confiabilidad. A manera de discusión podremos decir que, en función de los resultados mostrados en el diagnóstico de la transferencia, podemos afirmar que no existe evidencia que demuestre la pérdida de paquetes enviados en el enlace sugerido entre los nodos gestionados con Node-RED, en todos los casos nos arroja 0 % de pérdida de paquetes. Por otro lado, la placa de bajo costo Raspberry pi utilizada para ejecutar el gestor de flujo Node-RED se mantienen en los niveles de utilización de 4,4 %-43,7 %, nada despreciable si consideramos que está sometida a la gestión total del flujo de paquetes. Bajo el mismo tenor, es relevante mencionar que la latencia de la red se ve claramente alterada en los flujos de paquetes enrutados sobre los enlaces de radiofrecuencia 2.4 GHz como los son el ESP8266 y la Raspberry pi. Se aprecia que la jitter fluctúa hasta en un 50 % del resto de los enlaces con equipo utilizando medio de transmisión cable par trenzado UTP Cat 5.

Finalmente, se demostró que el intercambio de datos entre los diferentes microservicios para producción/consumo de los nodos, y el servidor gestor de protocolos Node-RED es 100 % confiable.

4. Conclusiones

Podemos confirmar que las placas reducidas y de bajo costo como la Raspberry pi y el microcontrolador ESP8266, son dispositivos confiables para el intercambio de datos a nivel industrial. El control de flujo de

paquetes en una red distribuida mediante un gestor de protocolos como Node-RED es totalmente confiable, cuenta con un alto grado de disponibilidad para operar con múltiples plataformas de protocolos industriales, además de ser muy intuitivo y amigable con el personal técnico y operativo. El siguiente paso sería trabajar en enlaces de radiofrecuencia orientados a la industria que nos permitan mejorar la latencia. Por el momento se sugiere utilizar redes cableadas con cable par trenzado UTP Cat 6, entre todos los nodos participantes en la red industrial.

Referencias

- [1] R. Huo, S. Zeng, Z. Wang, J. Shang, W. Chen, T. Huang, S. Wang, F. R. Yu, and Y. Liu, “A comprehensive survey on blockchain in industrial internet of things: Motivations, research progresses, and future challenges,” *IEEE Communications Surveys & Tutorials*, vol. 24, no. 1, pp. 88–122, 2022. [Online]. Available: <https://doi.org/10.1109/COMST.2022.3141490>
- [2] R. Sandusky, “Plc and pc system documentation concepts,” in *Forty-First Annual Conference of Electrical Engineering Problems in the Rubber and Plastics Industries*, 1989, pp. 38–47. [Online]. Available: <https://doi.org/10.1109/RAPCON.1989.47714>
- [3] D. Thompson and D. Watkins, “Comparisons between corba and dcom: architectures for distributed computing,” in *Proceedings. Technology of Object-Oriented Languages. TOOLS 24 (Cat. No.97TB100240)*, 1997, pp. 278–283. [Online]. Available: <https://doi.org/10.1109/TOOLS.1997.713554>
- [4] H. Bauer, S. Höppner, C. Iatrou, Z. Charania, S. Hartmann, S. U. Rehman, A. Dixius, G. Ellguth, D. Walter, J. Uhlig, F. Neumärker, M. Berthel, M. Stolba, F. Kelber, L. Urbas, and C. Mayr, “Hardware implementation of an opc ua server for industrial field devices,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 29, no. 11, pp. 1998–2002, 2021. [Online]. Available: <https://doi.org/10.1109/TVLSI.2021.3117401>
- [5] P. Helo, M. Suorsa, Y. Hao, and P. Anusornnitisarn, “Toward a cloud-based manufacturing execution system for distributed manufacturing,” *Computers in Industry*, vol. 65, no. 4, pp. 646–656, 2014. [Online]. Available: <https://doi.org/10.1016/j.compind.2014.01.015>
- [6] F. A. Osman, M. Y. M. Hashem, and M. A. R. Eltokhy, “Secured cloud SCADA system implementation for industrial applications,” *Multimedia Tools and Applications*, vol. 81, no. 7, pp. 9989–10 005, Mar. 2022. [Online]. Available: <https://doi.org/10.1007/s11042-022-12130-9>
- [7] C. Liu, Z. Su, X. Xu, and Y. Lu, “Service-oriented industrial internet of things gateway for cloud manufacturing,” *Robotics and Computer-Integrated Manufacturing*, vol. 73, p. 102217, 2022. [Online]. Available: <https://doi.org/10.1016/j.rcim.2021.102217>
- [8] L. P. Manik, “Performance factors effect on the performance metrics of the enterprise service bus,” *International Journal of Computing and Digital Systems*, no. 1, pp. 107–115, 2022. [Online]. Available: <https://dx.doi.org/10.12785/ijcds/110108>
- [9] H. ElMadany, M. Alfonse, and M. Aref, “Forecasting in enterprise resource planning (erp) systems: A survey,” in *Digital Transformation Technology*, D. A. Magdi, Y. K. Helmy, M. Mamdouh, and A. Joshi, Eds. Singapore: Springer Singapore, 2022, pp. 395–406. [Online]. Available: https://doi.org/10.1007/978-981-16-2275-5_24
- [10] J. A. Nina Chani, *El marketing B2B para optimizar el sistema interconectado de proveedores SAP y mejorar el posicionamiento industrial de la empresa METSO*. Universidad Autónoma San Francisco, Perú, 2021. [Online]. Available: <https://bit.ly/3hzSxtu>
- [11] M. Archana, D. V. Varadarajan, and S. S. Medicherla, “Study on the erp implementation methodologies on sap, oracle netsuite, and microsoft dynamics 365: A review,” 2022. [Online]. Available: <https://doi.org/10.48550/arXiv.2205.02584>
- [12] A. Luszczak, *What is Microsoft Dynamics 365/AX?* Wiesbaden: Springer Fachmedien Wiesbaden, 2019, pp. 1–4. [Online]. Available: https://doi.org/10.1007/978-3-658-24107-0_1
- [13] J. Müller, M. Lorenz, F. Geller, A. Zeier, and H. Plattner, “Assessment of communication protocols in the epc network - replacing textual soap and xml with binary google protocol buffers encoding,” in *2010 IEEE 17th International Conference on Industrial Engineering and Engineering Management*, 2010, pp. 404–409. [Online]. Available: <https://doi.org/10.1109/ICIEEM.2010.5646586>
- [14] A. Abdelli, W. Serrai, and L. Mokdad, “A novel and efficient index based web service discovery approach,” *Computer Standards & Interfaces*, vol. 80, p. 103586, 2022. [Online]. Available: <https://doi.org/10.1016/j.csi.2021.103586>

- [15] J. P. Sousa, Jand Mendonça and J. Machado, “A generic interface and a framework designed for industrial metrology integration for the internet of things,” *Computers in Industry*, vol. 138, p. 103632, 2022. [Online]. Available: <https://doi.org/10.1016/j.compind.2022.103632>
- [16] Q.-D. Nguyen, S. Dhouib, J.-P. Chanet, and P. Bellot, “Towards a web-of-things approach for opc ua field device discovery in the industrial iot,” in *2022 IEEE 18th International Conference on Factory Communication Systems (WFCS)*, 2022, pp. 1–4. [Online]. Available: <https://doi.org/10.1109/WFCS53837.2022.9779181>
- [17] A. Rajalakshmi and H. Shahnasser, “Internet of things using node-red and alexa,” in *2017 17th International Symposium on Communications and Information Technologies (ISCIT)*, 2017, pp. 1–4. [Online]. Available: <https://doi.org/10.1109/ISCIT.2017.8261194>
- [18] C. Ashhwath, V. Rohitram, and G. Sumathi, “Smart parking system using mqtt communication protocol and ibm cloud,” *Journal of Physics: Conference Series*, vol. 2115, no. 1, p. 012013, nov 2021. [Online]. Available: <https://dx.doi.org/10.1088/1742-6596/2115/1/012013>