Scientific Paper / Artículo Científico

UNIVERSIDAD POLITÉCNICA SALESIANA ECUADOR
INGENIUS
Revista de Ciencia y Tecnología
https://doi.org/10.17163/ings.n29.2023.08
pISSN: 1390-650X / eISSN: 1390-860X

Performance for interoperability between Rasperry pi and ESP8266 with a PLC in a Node-RED server for IIoT

Rendimiento para la interoperabilidad entre Rasperry pi, ESP8266 y PLC con Node-RED para el IIoT

J. Torres Ventura[1,*], A. H. Ruelas Puente[2], J. R. Herrera García[1]

Received: 21-06-2022, Received after review: 06-12-2022, Accepted: 13-12-2022, Published: 01-01-2023

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.

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

[1,*]Laboratorio de Mecatrónica, Universidad Autónoma de Baja California, México. Corresponding author ✉: jose.torres.ventura@uabc.edu.mx.

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

Suggested citation: Torres Ventura, J.; Ruelas Puente, A. H. and Herrera García, J. R. "Performance for interoperability between Rasperry pi and ESP8266 with a PLC in a Node-RED server for IIoT," *Ingenius, Revista de Ciencia y Tecnología*, N.° 29, pp. 90-97, 2023, doi: https://doi.org/10.17163/ings.n29.2023.08.

1. Introduction

The concept of IoT has triggered the fourth industrial revolution [1]; however, the history started at 1960 with the need to share data of manufacturing and production processes with the business administrative services. At an industrial level, Modular Digital Controller (MODICON), a manufacturer of Programmable Logic Controllers (PLC), patented in 1979 [2] the protocol for data exchange known as Modbus. Contemporary personal computers also required to be part of this information for managing the operation of the industrial process. In 1983, Microsoft, the main distributor of personal computers, introduced the Distributed Component Object Model (DCOM) protocol [3], which has evolved to become the Object Linking and Embedding Process Control (OPC) [4].

The production process uses Manufacturing Execution Systems (MES) platforms [5], which are complex modular computing systems that manage the flow control of raw material, inventory in the process, purchase orders, list of materials, defects and finished products among others. The monitoring and control of the performance of the machinery and equipment in the manufacturing floor is carried out through specialized hardware and software systems known as Supervisory Control and Data Acquisition (SCADA) [6]. They execute models such as Key Production Index (KPI) and Overall Equipment Effectiveness (OEE). The MES and SCADA systems are grouped in a platform known as Computer Integrated Manufacturing (CIM) [7].

On the other hand, the interoperability between the different administrative departments of the business is carried out with the technique known as Enterprise Service Bus (ESB) [8], whose function is to control the process of data flow between the different nodes of the corporate network, supported on data structures with multiple types of Application Programming Interface (API). Similarly, these business systems are integrated in a platform known as Enterprise Resource Planning (ERP) [9]. This work has the following structure: section II describes the methodology to measure the performance and status of the data exchange in a distributed network, oriented to production/consumption microservices between the nodes of the local network. Section III presents the results of network analysis and diagnosis, measuring parameters associated to the transfer and latency of the data flow between the nodes.

Section IV presents the conclusion of the research work and some areas of opportunity. The objective of this work is to evaluate, in a descriptive manner, the reliability of the integration of the Raspberry Pi and ESP8266 development boards to traditional industrial technologies regarding data exchange. Thus, it will be possible to provide an interoperability alternative which is reliable, low-cost and a with a relatively shallow learning curve. This will enable the CIM and ERP platforms to carry out their processes oriented to microservices. Historically, the need of data exchange between the different administrative departments of the business and the manufacturing floor, has been a topic delegated to the great gurus of computer systems such as SAP [10], Oracle ERP [11], Microsoft Dynamics 365 [12], among others.

The objective of these powerful computer systems is oriented to address services that grow in such a way that they demand many resources, code increase, and, above all high investment, operation and maintenance costs. The strategy proposed in this research is based on the nature of the operations of industrial nodes, oriented to microservices due to its application only for specific functionalities, which enables a scalability that depends on the hardware and not on the software. For this reason, a search is started about how the issue of data exchange between the production floor and the business has been addressed.

1.1. Data exchange between business departments

Data exchange protocols independent of the technological platform or type of service are used in computer systems for corporate business services. In this manner, the Simple Object Access Protocol (SOAP) was developed, which uses the eXtensible Markup Language (XML) [13]. This protocol generates Web Services Description Language (WSDL) text documents [14], to offer a service contract to the participants. Afterwards, improvements are achieved with the Representational State Transfer (RESTful), which seek to safely obtain a virtual address without the need of contracts.

1.2. Data exchange between manufacturing systems

In 1994, the Object Linking and Embedding for Process Control Foundation (OPC foundation) [15], presented internationally the OPC Server and OPC Client as a model to regulate the interoperability in the industrial sector. OPCs are software libraries nested in automation equipment and personal computers. Their cost varies according to each product and the number of labels, although some demonstration versions are available with limited labels and service time. At present, some technology developers, such as IBM and Eclipse, have promoted Open-Source ecosystems based on this model; this is how OPC Unification Automation (OPC UA) [16] and Node-RED [17] emerged; the latter is multiplatform, and also has support from Open-Source communities.

An administrator of industrial protocols is a server that manages in a centralized manner the flow of data packets through the nodes of the network, taking advantage of libraries released by manufacturers of industrial automation equipment. The Node-RED computer system is perfectly embedded in the Industrial Internet of Things (IIoT), because it is multiplatform and multiparadigm, consumes few resources, provides a shallow learning curve, and above all, provides a high processing power and administration of communications. The literature reviewed shows the continuous search for flexible solutions to address, on one hand, the unsynchronized nature of the production floor with the business times and, on the other hand, the no compatibility of the data formats between their applications; moreover, the communication protocols between their distributed network buses are atypical. This complexity justifies the involvement of a multidisciplinary group to attempt to address these setbacks, which result in high costs, a steep learning curve and a relatively low scalability level, challenges that prevail until today.

2. Materials and methods

The contribution of this work is to become a reference, based on real production system, for researchers and technology developers in the area of Information and Communication Technologies (ICT), as an alternative to carry out the exchange of the data produced and consumed between the nodes of a distributed network, oriented to deploy CIM and ERP systems in a reliable and low-cost manner, and also with a relatively shallow learning curve. The descriptive method will be used to validate the data exchange between the nodes of the network, based on the measurement of the network performance using the latency and the transfer. Thus, it was proceeded as follows:

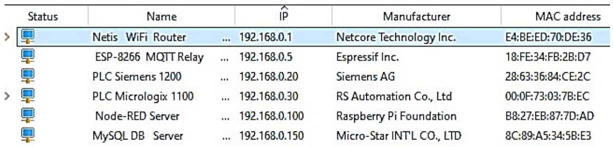
- Select the interface that enables the interoperability between the data produced/consumed by all nodes of the distributed network
- Build a distributed network with nodes from different industrial platforms and development boards, for process automation
- Create a map of variables to be packaged and measured
- Transfer packages between participating nodes (maximum load)

2.1. Node-RED as an administrator of data exchange

Node-RED is a project developed mainly by IBM in 2013 and released in 2016 [18]. It is multiplatform, which implies that it runs on Windows, Macintosh, and Linux and Solaris cores. Since it is a visual flow programming tool, it may be used by non-specialized staff, even enthusiastic and creative people. It was developed to consume few computer resources, and enables to create a network server to manage the connection with IIoT devices with Advanced RISC Machine (ARM) architectures and RISC platforms.

2.2. Deployment of the distributed network

Figure 1 shows the nodes participating in the distributed network that was installed to carry out the data exchange. At the physical layer level, the link between the Node-RED server uses the IEEE.802.11 standard, as well as the ESP-8266 MQTT Relay. All other use the IEEE.802.3 standard with UTP Cat 5 structured cabling. The Netis Wi-Fi router communicates all link on a TCP/IP socket, at a frequency of 2.4 GHz.



Status	Name		IP	Manufacturer	MAC address
›	Netis WiFi Router	...	192.168.0.1	Netcore Technology Inc.	E4:BE:ED:70:DE:36
	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

Figure 1. Nodes participating in the decentralized network

At the application layer level, Node-RED concentrates and manages flow traffic by means of MySQL DB María connectors, with the node labeled MySQL DB Server. The labeled node ESP-8266 MQTT Relay interacts through the MQTT (Message Query Telemetric Transport) protocol, whereas it uses the S7 protocol for the PLC Siemens 1200 node, and relies on the PCCC protocol for the PLC Micrologix 1100 node.

2.2.1. Construction of a gateway using a Raspberry Pi board

The flow administrator labeled as Node-RED server uses an ARM-v7 single core processor of 1 GHz, with 512 MB of RAM, Wireless 2.4 GHz and IEEE802.11. It uses the Debian Linux distribution as operating system, and a Node-RED server application that is deployed on Node JS. The technological platform of ARM processors uses very few instructions for its operation, which releases resources for client applications; they also have a low energy consumption, are reliable, low-cost, with a shallow learning curve, and also have a large support from open-source communities.

2.2.2. Functionality of the node with ESP8266

The hardware for the node that starts the operation of the manufacturing cell in the production floor is an IoT device, which receives the command labeled as start/stop. It consists of a microcontroller with an ESP8266 integrated circuit manufactured by Espressif, based on a 32-bit Xtensa LX106 RISC CPU, with an 80 MHz clock, 64 KB of RAM, and a QSPI 512 KB flash memory. It has a radiofrequency connection under the IEEE 802.11 b/g/n standard, and communication interface SPI (Serial Peripherical Interface) and I^2 C (Inter-Integrated Circuit).

2.2.3. Functionality of industrial nodes

The industrial nodes have been labeled as explained below. A Siemens 1200 PLC that measures the temperature of the process in a variable range from 0 to 40 °C. It was also installed an AB 1100 PLC with a process that deploys a production cycle, once it receives in a consumption label the value “true” / “false”. On the other hand, an ESP8266 microcontroller device was installed to give the start/stop, whose variable takes values of “1” / “0”. Finally, a server of relational MySQL databases is installed in a PC with the Win10 operating system, to record (consume) the history of the process cycle produced in the AB 1100 PLC, in a variable executed in a range of decimal integers from 0 to 5.

2.3. Map of variables

Table 1 shows the physical nodes and their identification at the level of layer 3 of the OSI model. The value column indicates the type of data transferred as useful load (payload), which will be packaged to be further transferred between the microservices. Similarly, the production and consumption columns are oriented to the microservices that have its source and destination IP address as reference.

Table 1. Map of the variables used in the data transfer flow

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. Measure the process

The performance indices will be the transfer rate, the network latency and the data exchange status. It is considered relevant to monitor the initial reference without a total load between the participating nodes, to see the status stored by the network that relies on a link under the IEEE 802.11 and IEEE 802.3 standards, with TCP (Transmission Control Protocol) transport protocol.

2.4.1. Transfer in the distributed network

The analytical calculation of the transfer rate is defined as the speed at which information may be transferred in a network transmission medium. This may be modeled by equation (1).

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

Where:

v_t = transfer rate (MB/s)
i_t = amount of information (MB)
t = transmission time (s)

The method used to measure the performance of the network is descriptive. The Iperf ver. 3 software is used to analyze the network transfer. In the case of the latency, a diagnosis of the network was performed using the Internet Control Message Protocol (ICMP), through the Echo Requests and Echo Replay methods. In addition, the Wireshark tool is used to monitor the status of the data. Finally, MTR is used to analyze the router jumps.

2.4.2. Latency of the distributed network

The latency considers the time that a bit takes to travel through a network, from source to destination. Analytically, it would be modelled as shown by equation (2).

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

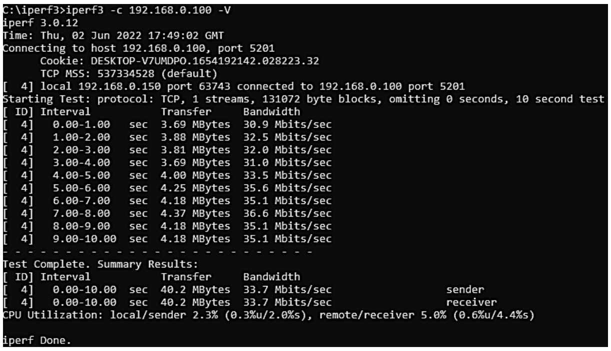
Where:

TP = propagation time (s)
TT = transmission time (s)
TC = queue time (s)

The propagation time is the time taken by a bit to travel from source to destination through a specific transmission medium, measured in milliseconds (ms). The transmission time is the time necessary to put all the bits of a package in the transmission medium, measured in milliseconds (ms). Finally, the queue time is the time required by a communication equipment during the transmission, to manage the flow of data packets from the input to the output of the transmission medium, measured in milliseconds (ms).

2.4.3. Initial monitoring of network performance

For developing experiment, it is obtained a reference of the status stored by the link of the network used. Figure 2 shows the data flow between the MySQL server and the Node-RED server with TCP protocol. It is seen that the transmission velocity between them is acceptable, considering that the transmission medium is between UTP Cat 5 cable and Wi-Fi at 2.4 GHz. It is also observed that the Node-RED server has a utilization of 4.4%, i.e., with initial load. The diagnosis tool was Iperf3, with a client and a server.

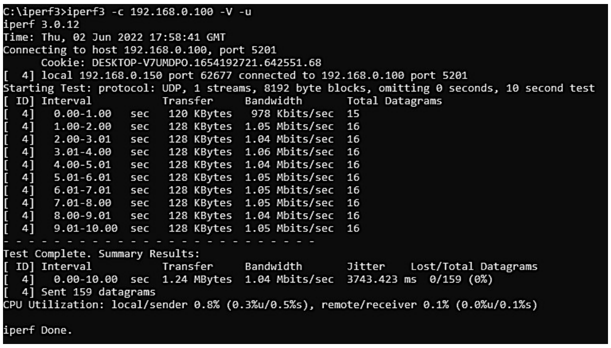


```
C:\iperf3>iperf3 -c 192.168.0.100 -V
iperf 3.0.12
Time: Thu, 02 Jun 2022 17:49:02 GMT
Connecting to host 192.168.0.100, port 5201
      Cookie: DESKTOP-V7UMDPO.1654192142.028223.32
      TCP MSS: 537334528 (default)
[  4] local 192.168.0.150 port 63743 connected to 192.168.0.100 port 5201
Starting 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
[  4]   0.00-10.00  sec  40.2 MBytes  33.7 Mbits/sec                  sender
[  4]   0.00-10.00  sec  40.2 MBytes  33.7 Mbits/sec                  receiver
CPU Utilization: local/sender 2.3% (0.3%u/2.0%s), remote/receiver 5.0% (0.6%u/4.4%s)

iperf Done.
```

Figure 2. Monitoring the transmission between the MySQL server and the Node-RED server

Figure 3 shows the variation of the latency between the same nodes. In this case the User Datagram Protocol (UDP) will be used, considering that it is most transparent since it does not have a window, and thus it is necessary to wait for the confirmation from the destination.



```
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-V7UMDPO.1654192721.642551.68
[  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%u/0.5%s), remote/receiver 0.1% (0.0%u/0.1%s)

iperf Done.
```

Figure 3. Variability of the latency, also known as jitter

It can be seen that the loss of packages is 0%, for a packet of 8192 bytes with a bandwidth of 1.05 Mbits/s before the beginning of operations. It may be confirmed that it is an acceptable behavior for the transfer as initial reference conditions.

3. Results and discussion

First, the latency of the network will be measured considering all participating nodes, demanded at full load. Figure 4 shows a plot of the trend of the latency round-trip solved by the ICMP protocol.

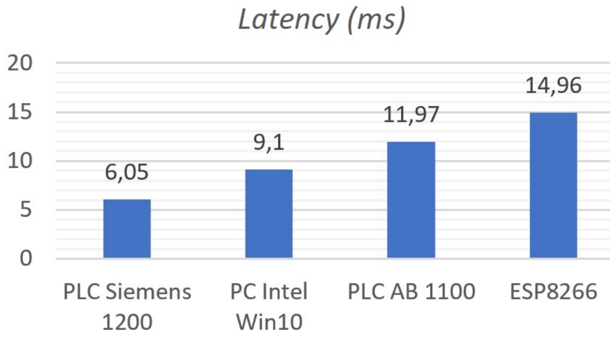


Figure 4. Diagnosis of the round-trip latency using ICMP packages

Specifically, the ping command was used from the command screen (Shell) of the Node-RED server node, to each of the nodes participating in the network. It was resolved for ten packets that were sent, and entirely received in all cases. Nevertheless, due to the latency increase at the link of each node, it was also prudent to report its standard deviation, which is shown in Figure 5.

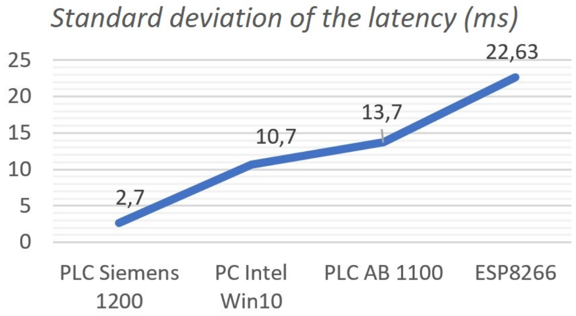


Figure 5. Standard deviation as a function of the network routing

It is sought to find a relationship of the increase of up to 50% in the latency between the Node-RED server and the ESP-8266 node, compared to the rest of the nodes. An analysis was performed with the MTR diagnosis tool to discard that this is due to the jumps between packets of the network; this is shown in Figure 6.

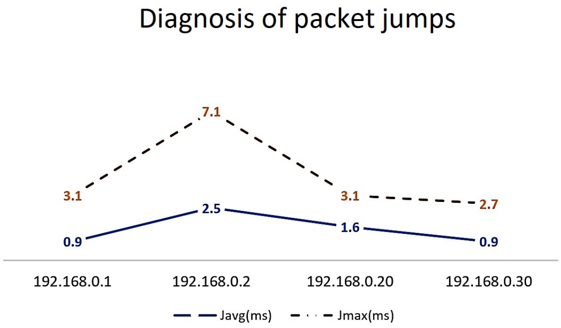
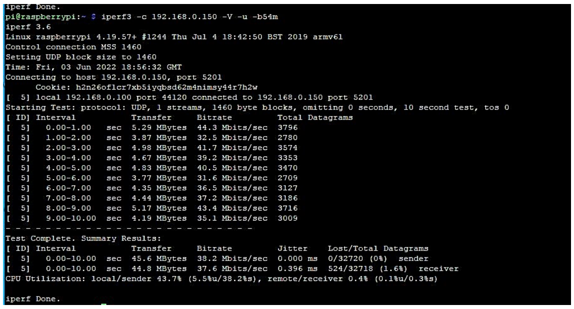


Figure 6. Diagnosis of packet jumps in each node participating in the network

It can be observed that ten packets were sent from the Node-RED server, and it is found that there is a jitter (Javg) up to 2.5 ms larger in average. In addition, the standard deviation once again increases significantly to more than twice of the remaining participating nodes with 7.1 ms. For this reason, it was also necessary to verify the transfer of the Node-RED server, flooding the network with ten packages in a UDP protocol, for a maximum demand of 54 MB on the Wi-Fi link, which is shown in Figure 7.



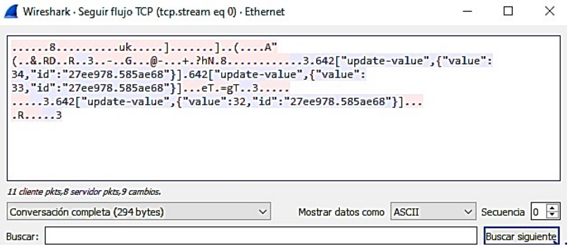
```
iperf Done.
pi@raspberrypi:~ $ iperf3 -c 192.168.0.150 -V -u -b54m
iperf 3.6
Linux raspberrypi 4.19.57+ #1244 Thu Jul 4 18:42:50 BST 2019 armv6l
Control connection MSS 1460
Setting UDP block size to 1460
Time: Fri, 03 Jun 2022 18:56:32 GMT
Connecting to host 192.168.0.150, port 5201
      Cookie: h2n26oflcr7xb5iyqbsd62m4nimsy44r7h2w
[  5] local 192.168.0.100 port 44120 connected to 192.168.0.150 port 5201
Starting Test: protocol: UDP, 1 streams, 1460 byte blocks, omitting 0 seconds, 10 second test, tos 0
[ 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  3.87 MBytes  32.5 Mbits/sec  2780
[  5]   2.00-3.00   sec  4.98 MBytes  41.7 Mbits/sec  3574
[  5]   3.00-4.00   sec  4.67 MBytes  39.2 Mbits/sec  3353
[  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  4.35 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/Total Datagrams
[  5]   0.00-10.00  sec  45.6 MBytes  38.2 Mbits/sec  0.000 ms  0/32720 (0%)  sender
[  5]   0.00-10.00  sec  44.8 MBytes  37.6 Mbits/sec  0.396 ms  524/32718 (1.6%)  receiver
CPU Utilization: local/sender 43.7% (5.5%u/38.2%s), remote/receiver 0.4% (0.1%u/0.3%s)

iperf Done.
```

Figure 7. Saturation of the Wi-Fi link with 54 MB using the UDP protocol

It can be seen that the variability of the round-trip latency is around 0.396 ms, with 1.6% of packet loss, considering a utilization of 43.7% of the processor at the side of the Node-RED server. Finally, an analysis of the data flow was performed with the Wireshark software tool to validate the status of the data exchange transmitted through the network, and this is shown in Figure 8. The update of the label produced at the Siemens 1200 PLC is recorded in the Node-RED server with the consumption field «value».

Regarding the transfer of the data produced by the AB 1100 PLC and consumed by the server of the MySQL relational database, Figure 9 shows the label value in the field «VALUES» by means of the report of TCP flow made with Wireshark.



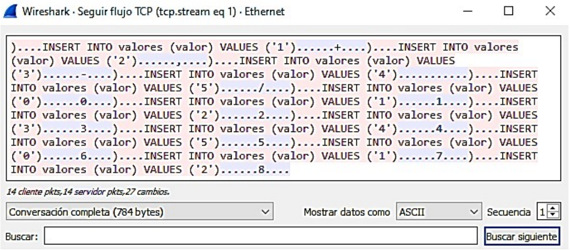
```
......8..........uk.....].......]..(....A"
(..&.RD..R..3..-..G...@-...+.?hN.8............3.642["update-value",{"value":
34,"id":"27ee978.585ae68"}].642["update-value",{"value":
33,"id":"27ee978.585ae68"}]...eT.=gT..3.....
.....3.642["update-value",{"value":32,"id":"27ee978.585ae68"}]...
.R.....3
```

11 cliente pkts,8 servidor pkts,9 cambios.

Conversación completa (294 bytes) Mostrar datos como ASCII Secuencia 0

Buscar:

Figure 8. Transfer of the label value produced at the Siemens 1200 PLC node and consumed in the Node-RED server



```
)....INSERT INTO valores (valor) VALUES ('1')......+....)....INSERT INTO valores
(valor) VALUES ('2')......,....)....INSERT INTO valores (valor) VALUES
('3')......-....)....INSERT INTO valores (valor) VALUES ('4')...........)....INSERT
INTO valores (valor) VALUES ('5')....../....)....INSERT INTO valores (valor) VALUES
('0')......0....)....INSERT INTO valores (valor) VALUES ('1')......1....)....INSERT
INTO valores (valor) VALUES ('2')......2....)....INSERT INTO valores (valor) VALUES
('3')......3....)....INSERT INTO valores (valor) VALUES ('4')......4....)....INSERT
INTO valores (valor) VALUES ('5')......5....)....INSERT INTO valores (valor) VALUES
('0')......6....)....INSERT INTO valores (valor) VALUES ('1')......7....)....INSERT
INTO valores (valor) VALUES ('2')......8....
```

14 cliente pkts,14 servidor pkts,27 cambios.

Conversación completa (784 bytes) Mostrar datos como ASCII Secuencia 1

Buscar:

Figure 9. Transfer of the data produced at the AB 1100 PLC node and consumed at the MySQL Server node

It was found that the exchange of data between the nodes and the Node-RED server, which manages the protocols, operates with a high reliability. Based on the results shown in the transfer diagnosis, it can be stated that there is no evidence that demonstrates loss of the packets sent through the link suggested between the nodes managed with Node-RED; a 0% of packet loss is achieved in all cases. On the other hand, the low-cost Raspberry Pi board used to execute the Node-RED flow administrator maintains utilization levels between 4.4% and 43.7%, which is not negligible considering that it is in charge of the total administration of the flow of packets. Similarly, it is relevant to mention that the latency of the network is clearly altered in the flow of packets routed on the 2.4 GHz radiofrequency links, such as the ESP8266 and the Raspberry Pi. It is observed that the jitter fluctuates in up to 50% of the remaining links with equipment, using a UTP Cat 5 twisted pair cable as transmission medium.

Finally, it was demonstrated that the data exchange between the different microservices for production/-consumption of the nodes, and the Node-RED protocol administrator server is 100% reliable.

4. Conclusions

It may be confirmed that the reduced and low-cost boards such as the Raspberry Pi and the ESP8266 microcontroller, are reliable devices for data exchange at an industrial level. The flow control of packets in a distributed network by means of a protocol administrator such as Node-RED is totally reliable, and it has a high degree of availability to operate with multiple platforms of industrial protocols; in addition, it is very intuitive and friendly with the technical and operational staff. The following step would be to work in radiofrequency links oriented to industry which enable to improve the latency. At this moment, it is suggested to employ networks wired using UTP Cat 6 twisted pair, among all the nodes participating in the industrial network.

References

[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. Anussornnitisarn, "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 Mendoną 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