Introdución a OPC UA TNS para Sistemas de Comunicación Industrial

Jorge Portilla[1]

***Abstract*— Con la introducción de *Internet of things* (IoT) en diferentes escenarios como la electrónica de consumo, automóviles, computación y recientemente en la industria. La automatización industrial está experimentando cambios abruptos, esto debido al crecimiento exponencial en parte por los recientes avances que permiten la interconexión en una escala más amplia y más fina (interconexión inteligente de redes de sensores y maquinaria).**

Entorno a los procesos industriales, en los últimos años se han realizado investigaciones sobre tecnologías y protocolos que contribuyan a la eficiencia de los procesos industriales, actualmente, el mercado es dominado por los sistemas fieldbus basados en Ethernet. Estos sistemas, aunque comparten requisitos, sus implementaciones y frameworks difieren notablemente unos de otros. Como resultado, los usuarios finales y los fabricantes de dispositivos se enfrentan a una multitud de tecnologías que deben ser ejecutadas, diagnosticadas y reproducidas para ser empleadas.

En el contexto de las tecnologías emergentes, este trabajo aborda una nueva tecnología llamada Open Platform Communication ed Architecture (OPC UA TNS). Como trabajo preliminar se realiza un estudio detallado de la tecnología y la implementación de un servidor OPC UA en un sistema embebido.

I. Introducción

Entorno a los procesos industriales, ha sido tendencia creciente el termino *Industria 4.0*, con el fin de aumentar la conectividad de los dispositivos y la construcción de sistemas industriales inteligentes, la computación en la nube se ha empleado en la automatización industrial para proporcionar una infraestructura flexible que cumpla con los requerimientos de la industria.

Actualmente, el mercado es dominado por los sistemas *fieldbus* basados en Ethernet. La alta disponibilidad de productos y servicios es satisfactoria, pero emplear múltiples soluciones generan altos costos y limitan la capacidad de IoT. Los sistemas de automatización actuales aplican tradicionalmente una arquitectura jerárquica conocida como pirámide de automatización, un punto principal en la arquitectura es la comunicación, y la estructura jerárquica induce a la comunicación entre las diferentes capas.

Estas capas están organizadas en tecnología operativa (OT) y tecnología de información (IT). OT hace referencia a las capas inferiores de la jerarquía y se encarga de intercambiar datos con grandes restricciones de tiempo, en su mayor parte en real-time. Por su parte IT, basa su funcionamiento en el protocolo de internet (IP). El gran desafió es la interconexión de estos dos mundos, por lo que a lo largo de los años se han realizado una gran cantidad de investigaciones siendo la forma más común de interconexión OPC (*Open Platform Communication*), y la estrategia de interconexión por medio de gateways. OPC permite la comunicación entre capas OT e IT.

Por otra parte, ~~entre los avances en las recientes tecnologías~~, existen conceptos de servicios basados en la nube para aumentar la eficiencia de los procesos industriales, para procesamiento y almacenamiento de datos. Además, existe IoT que sugiere una red de dispositivos interconectados en lugar de una jerarquía ostentosa. La aplicación de estas nuevas tecnologías conduce a una nueva arquitectura de automatización llamada Industrial IoT (IIoT). El IIot reduce el numero de gateways y propone una comunicación basada en IP para todas las capas funcionales, cumpliendo con requisitos de baja latencia, comunicación sensor-to-cloud y alta integrabilidad de dispositivos.

Como estrategia para abordar estos desafíos de los requisitos asociados a IIoT, parece sensata la evolución de la comunicación industrial. En estudios recientes realizados en [1] plantean una nueva arquitectura basada en: (i) Las redes TSN (*Time-Sensitive Networking*),actividad de estandarización reciente dentro de IEEE 802.1 que incluye capacidades en tiempo real en el estándar de Ethernet; OPC UA, la combinación de ambos promete cumplir con todos los requisitos de IIoT e Industry 4.0, que finalmente se espera ofrecer el potencial de reemplazar los sistemas de comunicación industrial de hoy en día.

II. Background

La comunicación industrial es principalmente organizada en base a la arquitectura jerárquica conocida como pirámide de automatización, mostrada en la figura 1. Esta arquitectura se diseñó con el objetivo de estructurar la complejidad de los sistemas de automatización [1]. Las capas funcionales típicas y las herramientas asociadas en esta pirámide son shop floor, controlador lógico programable (PLC), control de supervisión y control de adquisición de datos (SCADA), sistema de ejecución de fabricación (MES) y planificación de recursos empresariales (ERP) [2], aunque esto de ninguna manera es la única estructura posible y muchas variantes de la pirámide de automatización se han definido a lo largo de los años [3].

Un aspecto fundamental en la pirámide de automatización es la comunicación. Dentro de la jerárquica es importante el intercambio de información entre las diferentes funciones de automatización, los sistemas de comunicación como un medio para la transferencia de datos dentro y entre las capas funcionales. Esto estimuló el desarrollo de una gran

[1] Jorge Portilla is with the Department of Electronic Engineering, Universidad Tecnica Federico Santa Maria, Ph.D. student, Chile. jorge.portilla@sansano.usm.cl. This project is presented to the Computer Networks subject.

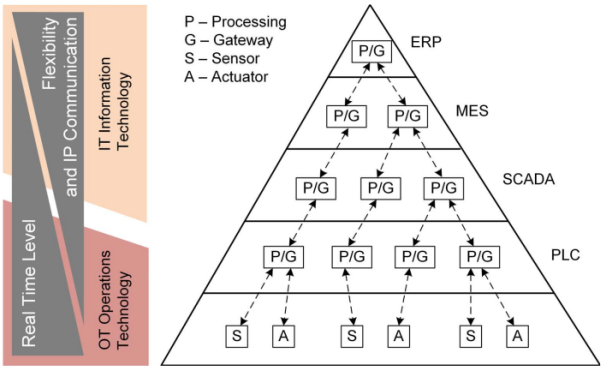

Fig. 1: Pirámide de automatización [4]

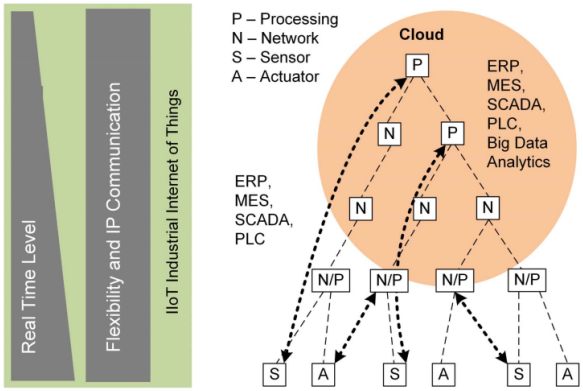

Fig. 2: Arquitectura de Industria 4.0 : IIoT [4]

cantidad de sistemas de comunicación industrial diseñados específicamente para la tarea de intercambiar datos críticos en su mayoría en tiempo real en las capas inferiores de la jerarquía. Estas capas inferiores se denominan *tecnología operativa*(OT).

En lo relacionado a comunicación de máquina a máquina (M2M), el papel de OPC UA (IEC 62541 2) está aumentando rápidamente en importancia junto con los sistemas de bus de campo M2M basados en Ethernet tradicionales (PROFINET, EtherNet / IP , etc).

Dentro de la máquina (niveles de dispositivo y sensor), los protocolos con capacidades en tiempo real (real-time Ethernet) dominan el área. Se utilizan sistemas de comunicación industrial como el *POWERLINK* basado en Ethernet, PROFINET, o EtherCAT, así como los llamados sistemas fieldbus más antiguos [1] para cumplir los requisitos relativos a la baja latencia determinista, y sincronización. Aunque estas tecnologías comparten requisitos comunes, sus implementaciones difieren significativamente. Por lo tanto, compararlos es un asunto complicado y depende en gran medida de la aplicación a desarrollar, la relación de E / S, entre otros).

Las capas superiores de la pirámide aplican la tecnología de información (IT), que actualmente se basa exclusivamente en tecnología de Internet, es decir, comunicación de protocolo de Internet (IP). La interconexión de estas dos áreas de comunicación diferentes con requisitos significativamente diferentes siempre ha sido un desafío. Una estrategia común para conectar los dos mundos es usar gateways [4].

La transferencia de datos pura es solo un aspecto del intercambio de información en los sistemas de automatización. Otro aspecto igualmente importante es el modelado de datos y el acceso a los datos a través de plataformas. Dada la gran variedad de sistemas de comunicación, también se requiere una solución independiente de la red. Con los años, OPC evolucionó como un enfoque de modelado neutral que fue respaldado por un número creciente de proveedores de dispositivos. y gradualmente se convirtió en un estándar de manejo de información comúnmente aceptado. OPC define muchas funciones no abordadas por los sistemas de comunicación industrial tradicionales, por lo tanto, también es un link adecuado entre las capas OT e IT [3].

El reciente movimiento de la Industria 4.0 está generando cambios en los sistemas de automatización. Primero, define nuevos requisitos y pone los antiguos requisitos en una nueva perspectiva [4]: la personalización masiva y la producción automatizada eficiente de lotes pequeños han sido una visión durante décadas. Los nuevos conceptos son servicios basados en datos basados en la nube que ayudan a aumentar la eficiencia general (optimización de energía, monitoreo de condiciones y mantenimiento predictivo) [5]. En segundo lugar, se basa en el paradigma de Internet de las cosas (IoT) que sugiere una nube plana de dispositivos interconectados en lugar de una jerarquía sofisticada. La adopción de estas nuevas tecnologías nos lleva a una nueva arquitectura de automatización llamada Industrial IoT [4], mostrada en la figura 2.

El IIoT utiliza menos gateways y tiene una comunicación uniforme basada en IP en todas las capas funcionales. Los requisitos típicos de comunicación relacionados con la automatización, como baja latencia, alta disponibilidad, sincronización de tiempo, gran cantidad de datos y dispositivos, así como la reconfiguración sin interrupciones (Hot Plug), plug and play, comunicación sensor-to-cloud y convergencia están presentes y se abordan en todas las capas funcionales [6].

Sin embargo, una jerarquía más plana también exige sistemas de comunicación adecuados. Requiere la coexistencia de los aspectos de OT y TI, en particular, con respecto a la comunicación determinista.

Además, el concepto de IoT requiere acceso IP a los dispositivos de campo, lo cual no es posible con los sistemas de bus de campo heredados y aún es en parte difícil con las soluciones de Ethernet en tiempo real (RTE) actuales.

III. OPC UA TNS

A continuación, una descripción de los dos elementos principales de la nueva arquitectura: TSN y OPC OPC UA.

III-A. ***OPC UA***

El protocolo OPC UA es M2M diseñado para permitir la interoperabilidad y comunicación entre maquinas conectadas bajo la *industria 4.0*.

Algunos de los fabricantes de sistemas embebidos, han desarrollado herramientas para la implementar *OPC UA* bajo *server development kit (SDK)* para permitir la comunicación en proyectos o diseños específicos. En el contexto de los sistemas embebidos, OPC UA controla datos en tiempo real y es adecuado para aplicaciones de automatización industrial donde las restricciones de tiempo son un aspecto importante de los datos.

En el contexto de los buses de campo tradicionales, se emplea una versión del modelo ISO/OSI para ubicar la funcionalidad. El estándar IEC 61784-2 distingue ~~solo~~ la *capa de enlace* de datos del fieldbus y la *capa de aplicación* del fieldbus, por lo que la convergencia de las redes OT y IT, requiere un protocolo más sofisticado como el mostrado en la figura 5.

III-A.1. ***Arquitectura OPC UA****:* OPC-UA es un estándar industrial independiente de la plataforma. Donde los dispositivos en diferentes tipos de redes pueden establecer una comunicación intercambiando mensajes entre clientes y servidores. Es un estandar fundamentalmente para el modelado y transporte de datos.

Las especificaciones básicas de OPCUA proporcionan solo la infraestructura para modelar una información, y de forma adicional, reporta las especificaciones de la industria del modelo de información definido por los proveedores y organizaciones de estándares.

OPC-UA se puede mapear en una variedad de protocolos de comunicación y datos pueden ser codificado de varias maneras para intercambiar portabilidad y eficiencia [7]. Dentro de la arquitectura de los sistemas OPC-UA se modelan los clientes y servidores, cada sistema puede contener múltiples clientes y servidores.

Cada cliente puede interactuar simultáneamente con uno o más servidores, y cada servidor puede interactuar simultáneamente con uno o más clientes [8].

Aplicaciones mas especificas pueden combinar servidores y clientes para permitir el intercambio de comunicación en varios niveles de la jerarquía de automatización.

III-A.2. ***Arquitectura Client/Server OPC UA****:* Una aplicación Client/Server emplea una API para el intercambio de solicitudes y respuestas del servicio OPC UA (mostrado en figura 3). La implementación de un cliente OPC UA accede al stack de comunicación utilizado por la API OPC UA.

Un cliente OPC UA ~~consiste en una implementación de cliente que~~ utiliza una pila de comunicación OPC UA. Algunos trabajos reportan que la API no esta estandarizada, por lo cual varia para diferentes lenguajes de programación [9]. Por ejemplo, puede haber un stack de comunicación para Java y un stack de comunicación para la nueva Windows Communication Foundation (WCF) de Microsoft.

OPC dentro de sus estándares permite que los desarrolladores pueden crear su propia API para comunicación.

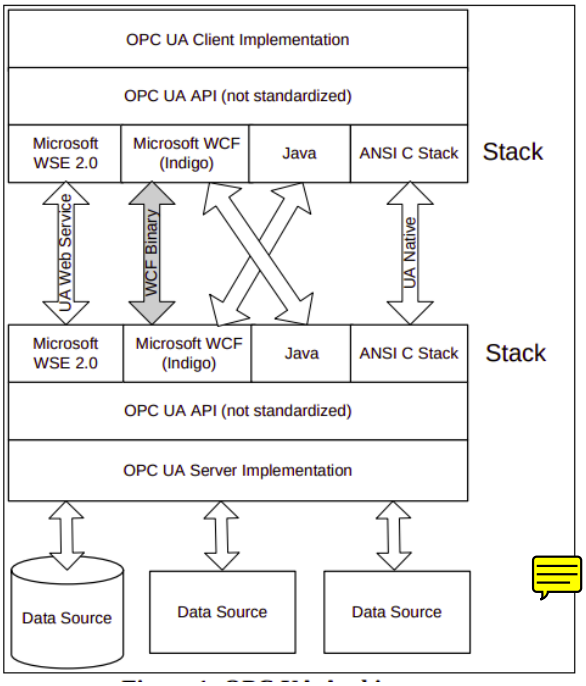

Fig. 3: Arquitectura OPC UA [9]

Ademas, el estandar describe la funcionalidad minima que debe realizar un servidor:

- Tiene un componente *Address Space component* que maneja una dirección especifica de la aplicación que esta expuesta al mundo entero.
- *Security component*, no proporciona características de seguridad estrictas pero se debe cumplir con un perfil de seguridad establecido por la fundación OPC.
- *Session component*, establece una instancia para la comunicación entre el cliente y el servidor, y se puede establecer cesión en cualquier momento dado.
- *Transport component* maneja un mecanismo de transporte basado en TCP/IP.
- *Discovery component*,según lo descrito por el estándar OPC, no se admite el registro en un servidor de descubrimiento externo, sin embargo, responde consultas de autodescubrimiento de clientes externos.
- *Attribute component*, que permite leer y escribir atributos específicos a un objeto en el espacio de direcciones.

III-A.3. ***OPC UA en sistemas embebidos****:* La principal ventaja es la facilidad, eficiencia y rentabilidad de acceder a sus datos, todo sin agregar un computador extra. Esta diseñado para ser una plataforma y sistema operativo independientes que permite una comunicación fluida entre todos componentes de un sistema de automatización.

Las aplicaciones de OPC UA se pueden desarrollar para cualquier sistema operativo y para sistemas que ejecutan un RTOS (Sistema operativo en tiempo real). En la actualidad existen muchas opciones para implementar un servidor OPC UA, basados en: microcontroladores ARM Cortex-A15s, desarrollos (IP Core) en FPGA (Field Programmable Gate Array) y algunos desarrollos por terceros para raspberry pi.

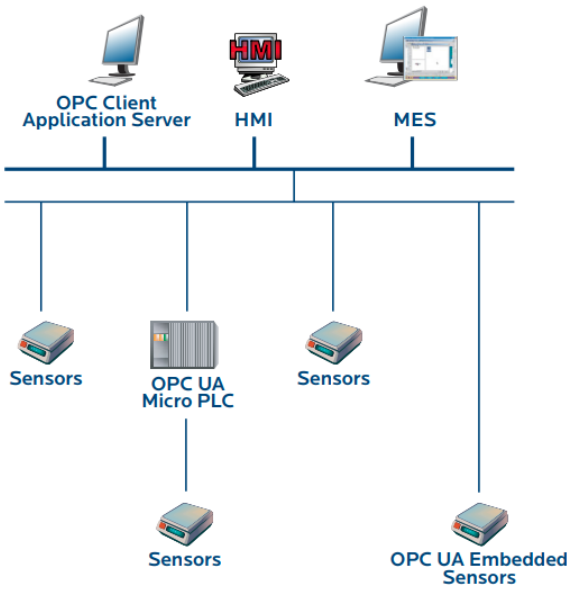

Fig. 4: Conectividad para un servidor OPC UA embebido [10].

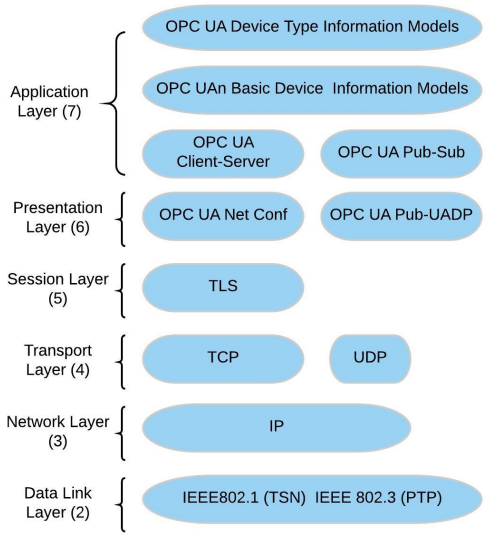

Fig. 5: Model OSI.

Las principales ventajas de este enfoque es la posibilidad de acceder a los pines GPIO, y el uso de RS-485, CAN y industrial ethernet (EtherCAT) usados para PRU-ICSS (Programmable Real-time Unit Industrial Communication Subsystems) [1]. En la figura 4 se muestra la conectividad de un servidor OPC UA embebido.

III-B. Time-Sensitive Networking (TNS)

IoT depende de Internet y hasta el momento no ejerce un rol importante en las comunicaciones industriales. Además, requieren únicamente comunicación basada en Internet, lo que tampoco ha sido posible en la automatización industrial.

La tecnología de la información (TI) y las redes de telecomunicaciones no pueden hacer frente a las necesidades específicas de la automatización para lograr una comunicación determinista, de alta confiabilidad y eficiente.

En este contexto, Ethernet TSN promete capacidades hard real-time para la redes industriales. Además, el desarrollo de las redes 5G beneficia al crecimiento de la industria de las telecomunicaciones en los procesos industriales. Estos desarrollos, pueden cambiar la estructura de las redes industriales, y podrían ser el requisito para implementar realmente IoT industrial (IIoT).

La característica principal de TSN es la posibilidad de coexistencia de diferentes tipos de tráfico, mientras se mantienen las propiedades de temporización del tráfico en tiempo real. Algunas tecnologías existentes en tiempo real (EtherNet / IP, Profinet) utilizan la planificación del tráfico y la QoS para garantizar un comportamiento en tiempo real bajo la condición de dispositivos que se comporten bien. Con TSN como capa de enlace de datos, esas tecnologías pueden aprovechar mejor la eficiencia del ancho de banda, ya que TSN protege el tráfico de alta prioridad.

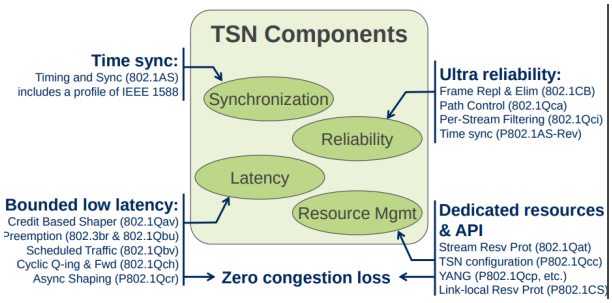

Fig. 6: Estandar TNS [1].

El estándar IEEE 802.1 (figura 6), es el encargado de desarrollar estándares para garantizar el transporte de datos con baja latencia, baja variación de delay, y extremadamente baja pérdida.

IV. METODOLOGÍA

Un concepto clave de la Industria 4.0 y IIoT es diseñar sistemas inteligentes distribuidos que requieren la intercomunicación de información a través de diferentes niveles de una red. Esta solución permite que estos dispositivos comuniquen datos tales como información del sensor y notificaciones de eventos, que se utilizan para el análisis, el mantenimiento y la adquisición de datos.

OPC UA es una buena opción para los sistemas de automatización industrial, ya que proporciona la comunicación estandarizada,con la seguridad y la abstracción necesarias para permitir las comunicaciones de máquina a máquina, además de servir como puerta de entrada a la nube.

OPC UA es un estándar de comunicación liviano destinado a ser utilizado en plataformas integradas como PLC, módulos de E / S en fábricas. Aborda los desafíos de comunicación que existen entre los equipos en diferentes redes en una

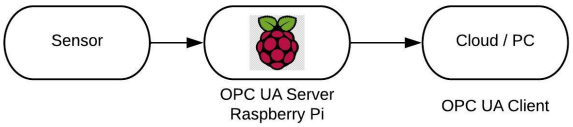

Fig. 7: flujo de transporte de datos desde el sensor hasta el dispositivo receptor

fábrica, que pueden estar utilizando diferentes estándares de comunicación.

En esta sección como trabajo preliminar para abordar la tecnología, se construye una aplicación IoT para enviar datos desde el sensor hasta un dispositivo receptor (PC o en la nube), usando raspberry Pi como hub y el protocolo OPC UA.

IV-A. Setup

En la figura 7 se muestra el flujo de diseño del proyecto, el fin de la implementación es adquirir datos provenientes de sensores, procesarlos y enviarlos a la nube o un dispositivo remoto. El esquema de trabajo en bloques es el siguiente:

- **Adquisición de Datos:** Sensores o datos sintéticos.
- **Procesamiento:** Raspberry Pi, Servidor OPC UA embebido.
- **Visualización y almacenamiento:** El Cliente OPC UA, accede a los datos del servidor, almacena en la base de datos InfluxDB y realiza la visualización en la interfaz Grafana.

En la actualidad existen muchas opciones para implementar un servidor OPC UA, como se mostró en la sección III-A.3.

IV-B. Adquisición de Datos

En este nivel se encuentra una Raspberry Pi que aloja una aplicación de servidor OPC UA (que se ejecuta con Prosys OPC UA Java SDK). La estación realiza mediciones de diferentes sensores, podrían ser que temperatura, humedad, distancia, entre otros. Con el fin de proporcionar estos datos al servidor OPC UA para que los clientes OPC UA puedan acceder a los da ~~En este trabajo, se realiza la implementación en una raspberry pi.~~ tos y realizar un monitoreo en línea. En este caso, el desarrollo de este proyecto sera para enviar valores desde los sensores a la nube, por lo cual no es fundamental las variables que se monitorean en el nivel de sensores y se pueden utilizar datos sintéticos.

En la siguiente sección realizamos una aplicación, un caso de IoT en dominio de la automatización industrial.

V. Resultados-Caso de estudio

En este trabajo, se realiza la implementación de un servidor OPC UA en una raspberry pi. los resultados obtenidos en el este trabajo son similares al caso de estudio presentado en [7].

Se realizan pruebas en la arquitectura para el envió y recepción de datos. Una conexión OPC UA no basado en tiempo real es usado para comunicar el dispositivo con la infraestructura de alto nivel donde se encuentra MES, ERP, y Internet. Con el fin de realizar un monitoreo y hacer diagnostico sobre la aplicación.

Se realiza la implementación de un servidor OPC UA en la raspberry pi 3 haciendo uso del código abierto *opcuafree*, que tiene soporte para python y C++. En este caso, en la aplicación solo tendremos un nodo por el cual enviaremos los datos de temperatura y humedad generados y adquiridos por el sistema de adquisición de datos. La configuración del servidor incluye la asignación de una dirección IP, sobre la cual el cliente puede solicitar información al servidor. El esquema básico de comunicación es presentado en la figura:

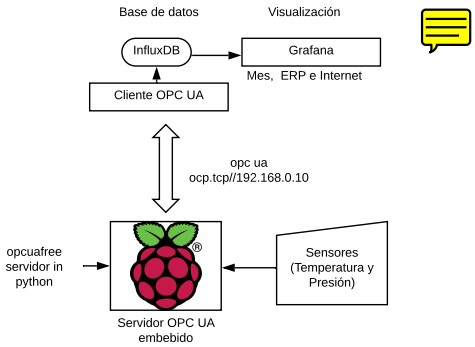

Fig. 8: Caso de estudio esquema básico.

Una de las opciones más comunes para implementar un cliente OPC UA es usando el software UaExpert V1.4.1, se muestra en la figura la configuración 11, y en la figura 12 se muestran la interfaz del programa, donde se asocian las parámetros definidos por el servidor, para el caso especifico de la aplicación son: Presión, temperatura y tiempo.

Los datos que se reciben del servidor son almacenados en una base datos por el cliente OPC UA, para este propósito se empleo la base de datos InfluxDB para series de tiempo. El único procesamiento asociado para almacenar los datos en la aplicación es almacenar los datos en formato *json*.

Las figuras 9 y 10la ejecución de los codigos y se muestran los datos adquiridos en el servidor OPC UA y los datos recibidos por el client.

La figura 13, muestra los valores de temperatura y humedad almacenados en la base de datos. Independientemente de la frecuencia de adquisición de datos del hardware, los valores de sensores son actualizados cada 2 segundos. Para la visualización de los datos, se construyo un dashboad de Grafana, la cual contiene múltiples gráficos y paneles que logran simplificar las métricas en una vista simple.

```
pi@raspberrypi:~/Desktop $ nano server.py
pi@raspberrypi:~/Desktop $ python3 server.py
Endpoints other than open requested but private key and certificate are not set.
Listening on 172.16.24.144:4840
Server started at opc.tcp://172.16.24.144:4840
46 245 2020-02-20 13:58:02.090247
26 629 2020-02-20 13:58:03.093202
44 746 2020-02-20 13:58:04.095897
26 821 2020-02-20 13:58:05.098521
41 952 2020-02-20 13:58:06.101131
48 743 2020-02-20 13:58:07.103760
35 460 2020-02-20 13:58:08.106476
15 552 2020-02-20 13:58:09.109068
31 871 2020-02-20 13:58:10.111821
```

Fig. 9: Datos adquiridos de los sensores en el servidor OPC UA embebido en la raspberry pi

```
jorge@jorge:~/Desktop$ python3 client.py
Cliente Conectado al Servidor
Temperatura = 38 Presión = 321  Tiempo 2020-02-20 14:05:38.343113
Temperatura = 39 Presión = 311  Tiempo 2020-02-20 14:05:39.345883
Temperatura = 21 Presión = 915  Tiempo 2020-02-20 14:05:41.351312
Temperatura = 24 Presión = 222  Tiempo 2020-02-20 14:05:42.354030
Temperatura = 15 Presión = 590  Tiempo 2020-02-20 14:05:43.356830
Temperatura = 25 Presión = 230  Tiempo 2020-02-20 14:05:44.359561
Temperatura = 25 Presión = 529  Tiempo 2020-02-20 14:05:45.362292
Temperatura = 39 Presión = 954  Tiempo 2020-02-20 14:05:46.365014
Temperatura = 18 Presión = 926  Tiempo 2020-02-20 14:05:47.367743
Temperatura = 28 Presión = 396  Tiempo 2020-02-20 14:05:49.378385
Temperatura = 23 Presión = 569  Tiempo 2020-02-20 14:05:50.381112
Temperatura = 15 Presión = 609  Tiempo 2020-02-20 14:05:51.383867
Temperatura = 12 Presión = 273  Tiempo 2020-02-20 14:05:52.386696
Temperatura = 23 Presión = 783  Tiempo 2020-02-20 14:05:53.389417
Temperatura = 40 Presión = 960  Tiempo 2020-02-20 14:05:54.392110
```

Fig. 10: Datos recibidos por el Client (Client en un PC)

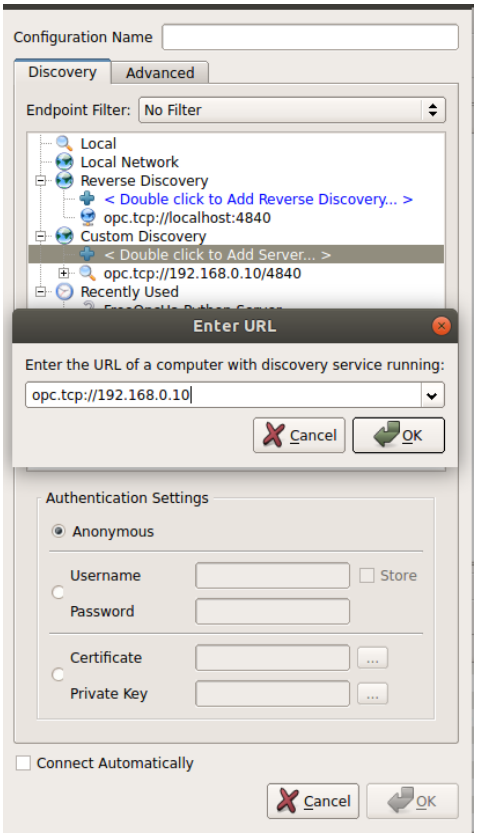

Fig. 11: Configuración del Cliente, ingreso de la url del Server OPC UA (UA Expert).

VI. Trabajo futuro

Como trabajo futuro, implementar TNS, el estándar IEEE 802.1 encargado de desarrollar estándares para garantizar el transporte de datos con baja latencia, baja variación de delay, y extremadamente baja pérdida.

Además, caracterizar latencia y thoughput, tomando como referencias el periodo de tiempo desde que se adquieren los datos y hasta su llegada al destino en la aplicación de visualización Grafana.

VII. Conclusiones

El IIoT reduce el numero de gateways, propone comunicación basada en IP para todas las capas funcionales, cumpliendo con requisitos de baja latencia, comunicación sensor-to-cloud y alta integrabilidad de dispositivos.

La aplicación desarrollada cumple con las exigencias de las recientes tecnologías de servicios basados en la nube para aumentar la eficiencia de los procesos industriales, para procesamiento y almacenamiento de datos.

Se puede encontrar aplicaciones especificas en sistemas de adquisición de datos, construcción de dataset y aplicaciones de control.

La principal ventaja de implementar un servidor OPC UA es la facilidad, eficiencia y rentabilidad de acc a sus datos, todo sin agregar un PC extra.

References

[1] D. Bruckner, M.-P. Stănică, R. Blair, S. Schriegel, S. Kehrer, M. Seewald, and T. Sauter, "An introduction to opc ua tsn for industrial communication systems," *Proceedings of the IEEE*, vol. 107, no. 6, pp. 1121–1131, 2019.

[2] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0," *IEEE industrial electronics magazine*, vol. 11, no. 1, pp. 17–27, 2017.

[3] T. Sauter and M. Lobashov, "How to access factory floor information using internet technologies and gateways," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 4, pp. 699–712, 2011.

[4] S. Schriegel, T. Kobzan, and J. Jasperneite, "Investigation on a distributed sdn control plane architecture for heterogeneous time sensitive networks," in *2018 14th IEEE International Workshop on Factory Communication Systems (WFCS)*, pp. 1–10, IEEE, 2018.

[5] K. Al-Gumaei, K. Schuba, A. Friesen, S. Heymann, C. Pieper, F. Pethig, and S. Schriegel, "A survey of internet of things and big data integrated solutions for industrie 4.0," in *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, pp. 1417–1424, IEEE, 2018.

[6] S. Heymann, L. Stojanovci, K. Watson, S. Nam, B. Song, H. Gschossmann, S. Schriegel, and J. Jasperneite, "Cloud-based plug and work architecture of the iic testbed smart factory web," in *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, pp. 187–194, IEEE, 2018.

[7] J. Imtiaz and J. Jasperneite, "Scalability of opc-ua down to the chip level enables "internet of things"," in *2013 11th IEEE International Conference on Industrial Informatics (INDIN)*, pp. 500–505, IEEE, 2013.

[8] G. Cândido, F. Jammes, J. B. de Oliveira, and A. W. Colombo, "Soa at device level in the industrial domain: Assessment of opc ua and dpws specifications," in *2010 8th IEEE International Conference on Industrial Informatics*, pp. 598–603, IEEE, 2010.

[9] S.-H. Leitner and W. Mahnke, "Opc ua–service-oriented architecture for industrial applications," *ABB Corporate Research Center*, vol. 48, pp. 61–66, 2006.

[10] D. Kominek, "Keys to developing an embedded ua server, liam power, embedded opc ua," 2013.

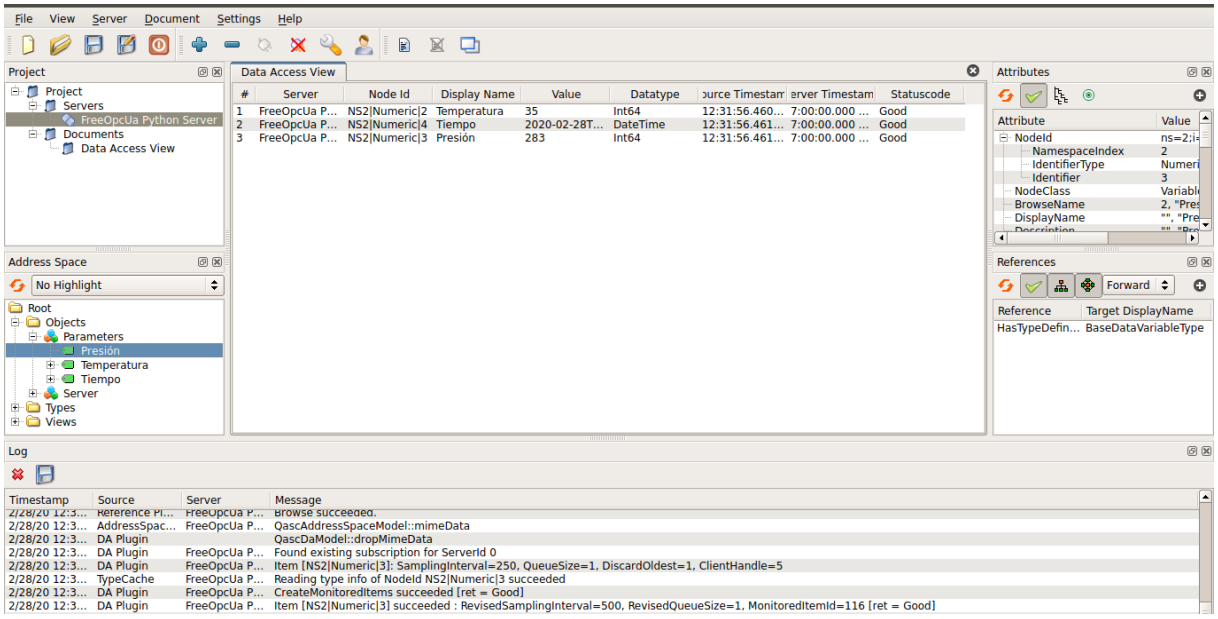

Fig. 12: Datos recibidos por el Client (Client en un PC)

Fig. 13: Grafana : Open source análisis y monitoreo para la base de datos influxDB