Design of an Intelligent Sensor Network for Dam Monitoring Based on IoT Technology

Luiz C. Magrini,
Paula S. D. Kayano,
Ferdinando Crispino
FDTE - Fundação para o Desenvolvimento Tecnológico da Engenharia
São Paulo, Brazil
magrini@fdte.org.br
pkayano@fdte.org.br
fcrispino@fdte.org.br

Edvaldo F. Carneiro,
Tatiana P. A. Cappi,
Antonio L. C. Santos
CESP - Companhia Energética de São Paulo
São Paulo, Brazil
edvaldo.carneiro@cesp.com.br
tatiana.araripe@cesp.com.br
antonio.carmo@cesp.com.br

Abstract— **The evolution of IoT (Internet of Things) technology has provided low cost tools that helps solve the integration difficulty of heterogeneous hardware and software platforms found in the Smart Grid, especially when hardware platforms presents low computational power, low power consumption energy and are geographically dispersed. This article introduces the implementation of a structural dam safety monitoring system for hydroelectric power plants that uses the standard protocol XMPP (eXtensible Messaging and Presence Protocol) to provide communication services between the different sensors installed at auscultation instruments. The XMPP protocol provides almost real-time and reliable communication between Intelligent Electronic Device (IEDs) as well as the communication between IEDS and SCADA systems. The data flow is transmitted encapsulated in a XML (Extensible Markup Language) format, using the standardization defined by the IEEE Sensei-IoT group. The system is developed and installed at CESP's hydroelectric power plant in Porto Primavera, São Paulo State, Brazil.**

Keywords— IoT, Internet of Things, Structural dam monitoring, Structural dam safety systems, XMPP protocol, IEEE Sensei-IoT.

I. INTRODUCTION

In hydroelectric plants the concrete and earth dam water way lock structures is usually monitored at the construction and operation phases of the water reservoir.

The auscultation civil instruments are equipments that are installed to help the engineers responsible for the structural health to analyze the behavior of the concrete and earth dam as well the waterway lock during the construction and reservoir filling phases. These auscultation instruments are inserted in the structure in places defined by the structural projecting company at construction, reservoir filling and operation phases to allow the tracking of mechanical forces that result in structural effects like displacements, overturning, elevations, etc.

These equipments are usually referred as auscultation civil instruments, and their periodic monitoring makes possible to detect problems in the initial stage, where repair is usually simpler and less costly. It also allows the structural stability monitoring of elements with many years of operation and showing signs of deterioration. The location and type of the auscultation civil instruments are conceived during the dam´s construction phase and have different functionalities, such as strain gages that evaluate the structure deformation due to the mechanical stresses variations caused by climatic and hydrological variations. Another very important instrument is the stand pipe that allows the sub pressure´s mapping at the installation site.

The flow meter is another auscultation instrument often encountered in dams, and it provides measurements of percolated water flows through the structures and foundations of concrete and earth dams.

By adapting transducers to the different types of civil instruments in the plant, it becomes possible to use microprocessors or Intelligent Electronic Device (IEDs) to digitize the collected measurements and make them available in the data communication network, creating a distributed system. This computational distributed system consists of a large number of IEDs geographically dispersed throughout the region comprised by the earth filled dam and the concrete dam, forming a distributed heterogeneous architecture, loosely coupled, and composed of several layers of software (multitier).

II. THE EMERGENCE OF THE INTERNET OF THINGS (IOT)

The emergence of the Internet of Things (IoT) concept cheapened and enabled interoperability between sensors, IEDs and SCADAs marketed by different manufacturers away from proprietary factory floor technologies that hindered and enhanced the connectivity between products from different manufacturers and technologies. The adoption of open web-based architectures democratizes access to the information through tools that are already known to most users, thus enabling access to information across all levels of business. Current approaches on IoT focus mainly on communication protocols to integrate

978-1-5386-7842-8/18/$31.00 ©2018 IEEE

things with Internet protocol standards, considering limited computing and memory resources as well as restricted bandwidth and power availability. In the particular case of the ongoing project, a small scale prototype of a dam safety monitoring system is in operation, where sensors and transducers are adapted to the existing auscultation instruments and that in its final stage will be a distributed system with more than two thousand civil instruments being monitored, whose information must be periodically collected, processed, and saved in a historical database. The use of equipment adhering to IoT technology, through its industrial versions, also called IIoT (Industrial Internet of Things) has cheapened the cost of local data acquisition and processing equipment, as well as providing data communication protocols that take advantage of the (TCP /IP) network infrastructure, both cable (fiber optic or twisted pair) and wireless communication (WiFi) powered by solar panel and battery set in remote locations. This architecture identifies software modules installed in the IEDs that play the role of producers and are responsible for the measurements production, while the SCADA software will have processes that will consume this information, process it, update screens and store it in database. The software modules development that perform the data communication between the IEDs equipped with producing processes responsible for acquire the sensor´s data, perform data´s preprocessing and transmitting it to the consuming processes of a SCADA system that will consume this information, can be facilitated through the use of middlewares which abstract the details of data communication made more evident on heterogeneous platforms, thus minimizing development time.

III. Middleware

In a distributed computing system, such as a network of intelligent sensors, where processing is divided into logical units residing in distributed IEDs, the smooth execution of the various software modules will produce the desired functionality. To fulfill this interaction, there is a great need for communication between IEDs that can hosts producer and consumer processes, and that can be supplied by a layer of software intermediate between the operating system and the application program, which is called Middleware. Standardized middleware facilitates interoperability between software from different manufacturers, as well as facilitates the development of new applications and simplifies integration with legacy systems. Most of the recently developed ICT (Information and Communications Technology) middlewares follow a service-oriented architecture based on SOA (Service Oriented Architecture) model. This architecture proposes the integration of software modules created with different computer languages and running at distinct operating system through the adoption of Web-based computational services (or web services) that exchange information and commands embedded in XML(Extensible Markup Language) format. This architecture is already endorsed by the NIST (NIST Framework and Smart Grid Interoperability Standards) as well as by the meta-model called Smart Grid Architecture Models (SGAM) created in a partnership between CEN (European Committee for Standardization), CENELEC (European Committee for Electrotechnical Standardization) and ETSI (European Telecommunications Standards Institute).

The communication needs in a Smart Grid are complex, due to the equipment, software and functions heterogeneity. It is up to the middleware to provide an abstraction layer allowing the user to focus on the application logic rather than taking care of the numerous exceptions and compatibility issues encountered. The middlewares can be classified into Remote Procedure Call (RPC) oriented Middleware, Transaction-Oriented Middleware (TOM), Object-Oriented/Component middleware (OOCM), and Message-Oriented Middleware (MOM).

The RPC middleware provides remote procedure invocation capabilities without the need to worry about communication details. Originally it was developed for synchronous procedure calls, that is, the consumer process is waiting for the call to resume in order continuing its execution, making the application difficult to be scalable and also having a low fault tolerance.

The Transaction-Oriented (TOM) type middlewares were initially designed for database access and have the capability for synchronous or asynchronous communication in a heterogeneous distributed system.

The Object-Oriented/Component Middleware (OOCM) was developed as a RPC extension to the object-oriented programming model.

Meanwhile, MOM-type middleware (Message-Oriented Middleware) provides message passing capabilities for distributed systems incorporating features like: synchronous or asynchronous communication, a common format for data transport. These characteristics make it suitable for applications that are poorly coupled and require messages with individual treatment according to their level of priority. This type of tool still offers resources only for message passing, where the processes involved must be defined explicitly without masking the participants' identity. This type of middleware can also be used with message queuing software that responds to the message persistence until delivery. In addition, asynchronous communication occurs through logical names that are resolved by the message queue manager software. All messages in the queue can be recovered in any order and only when needed.

For heterogeneous distributed systems requiring high availability, low power consumption (when powered by solar panel and battery) and high performance, such as Smart Grid and IoT applications, the messaging oriented middleware is the most appropriate since it decouples the network nodes making it easier for equipments with different hardware and software architectures to exchange messages with each other. It enables the exchange of synchronous and asynchronous messages, data format transformation to suit the needs of different remote applications running in heterogeneous IEDs, support for different levels of priority and parallel processing of many messages. In addition to point-to-point communication, there are situations in which exists several IEDs producing measurements. In this case, some implementations adds the publish-subscribe mechanism that distributes messages between many producers to many consumers.

Among the standards developed for IoT we can highlight: Message Queuing Telemetry Transport (MQTT) protocol, Data Distribution Service (DDS), eXtensible Messaging and Presence Protocol (XMPP) and Advanced Message Queuing Protocol (AMQP). Some protocols such as CoAP (The Constrained Application Protocol), Message Queuing Telemetry Transport (MQTT), and Web Application Messaging Protocol (WAMP) have interoperability problems between heterogeneous networks for large-scale IoT deployments.

The XMPP protocol can be used to solve interoperability problems between heterogeneous networks, making it possible to communicate within the sensor network through a simple exchange of text messages that can be done anywhere and with any device that has access to the Internet. The working group IEEE P21451 is dedicated to define an unified communication architecture for smart sensor networks and has adopted XMPP as the international standard for semantic Web and M2M/IoT (Machine-to-Machine communications/Internet of Things).

IV. The XMPP Protocol

The eXtensible Messaging and Presence Protocol (XMPP), or Jabber as it was originally known, is an open, extensible, XML-based protocol for exchanging data in instant messaging systems. It was originally developed at the request of the Internet Engineering Task Force (IETF), an international community of technicians, agencies, manufacturers, vendors, and researchers concerned with the evolution of the Internet architecture and its perfect functioning. The XMPP protocol is described by RFC 6121. This protocol follows the client-server architecture, but can be configured for other message-based communication models, such as publish/subscribe, presence and status updates, alerts, feature negotiation and service discovery.

The client authentication at a XMPP server is implemented with the help of the Simple Authentication and Security Layer (SASL) protocol, which is based on the exchange of data in the client/server application aiming authenticate the client on the server and establish a high communication level between them. The XMPP protocol also uses the Transport Layer Security (TLS) protocol, which works in conjunction with the TCP transport protocol for sending messages. After the communication being established the TLS protocol is responsible for the data communication security, providing privacy, authenticity and integrity to them.

Several XMPP open source implementations are available, making it extremely viable, as well as considering others factors such as:

- Security - Any XMPP server can be isolated from the public network (running, for example, on an enterprise intranet). The core of the XMPP specifications includes robust security mechanisms making use of Simple Authentication Security Layer (SASL) and Transport Layer Security (TLS).
- Low cost – There are several Open Source implementations that can be tailored to a particular project.
- Decentralized – The XMPP network architecture is similar to an email system, allowing anyone to run their own XMPP server, making it possible for individuals or organizations to take complete control of their communications.
- Extensible - Using XML capabilities, anyone can build functionalities creating another layer over the protocol core. In order to certify interoperability common extensions has its functionalities documented by the XEPs publications. However such publication is not required and companies can maintain their own extensions if they desire.
- Flexible - XMPP applications, in addition to IM (Instant Messaging), includes services such as network management, content organization, collaboration tools, file transfer, games, remote system monitoring and more.
- Diversity - There are a large number of companies and open-source projects that uses XMPP to build and deploy services and applications that implement real-time communication.

V. Structural Dam Monitoring System Architecture

A structural dam monitoring system is installed at the Hydroelectric power plant Eng. Sérgio Motta (also known as Porto Primavera) that is a large size power plant belonging to the CESP - Companhia Energética de São Paulo, Brazil and contains 14 generating units, each one equipped with 110 MW (1,540 MW) Kaplan turbine generators.

The earth dam is located at the Paraná river´s right bank and is 10.4 km long, while the riverbed concentrates the concrete structures (water intake, powerhouse, central wall and right lateral wall), the dam in the riverbed, the SF6 substation and the fish elevator. The left bank is the location of the left lateral wall, the left bank dam itself, water way lock and fish ladder.

To fulfill the monitoring necessities of about 2,500 civil auscultation instruments, a digital system was tailored for this power plant and has as an innovation factor the use of IoT technology contributions, like the XMPP communication protocol IoT extensions for the message exchange between IEDs and their respective transducers with the SCADA system. For this purpose, applications have been developed both on the side of the software producer (IEDs) and on the consumer side (SCADA system). This client software was developed based on the Python language for the IEDs that process a simplified version of an open source software in order to transfer the collected measurements after converting them into engineering units. It then formats the data to the particular

XML format standardized by the IEEE Senseio-IoT group, and sends them to the XMPP server installed on the SCADA system computer. In this case, extensions of the XMPP protocol for IoT were used, such as XEP-0323 (IoT Sensor Data), XEP-0325 (IoT Control) to standardize the XML content and tags of the file produced. To make the system more flexible, a communication driver for the ModBus/TCP protocol was incorporated in the client to allow integration with legacy sensors, since the information received by this driver is also formatted according to the standard XML and made available for the SCADA.

In this architecture, on the side of the SCADA software, a XMPP Server software was used to control and authenticate which XMPP clients can exchange messages. Another software module was developed and added to this server to parse the XML file content and insert it into the database after having been passed through a pre-processing that validates it through a reasonability analysis. As an XMPP server we used Openfire that is licensed under the Open Source Apache License and developed by the Open Source community called Ignite Realtime, made up of end users, developers and service providers around the world.

This XMPP server was configured with a unique configuration that can be described by using the following components:

- XMPP-IoT Component - Responsible for sending and receiving XML messages between XMPP clients;
- Database Component - Responsible for recording the data received from the XML format in the database.

The XMPP client installed on the IED can be described by using the following components:

- XMPP-IoT Component - Responsible for sending and receiving XML messages between XMPP clients;
- Modbus/TCP component - Responsible for integration with legacy sensors.

The XMPP client hosted at IEDs is installed as a service in order to provide a greater control of the program execution, especially in case of system reboot. In the same IED will also run a second service that will periodically check if the resident XMPP client is running and otherwise it will take the necessary steps to reactivate this service.

A. *Message Format*

Messages exchanged between the devices scattered at dam´s reservoir collects the readings of the civil monitoring instruments and converts to the format established by the Senseio/IoT group which is the first joint effort between ISO, IEC and IEEE, known as ISO/IEC/IEEE 21451-1-4 - Smart transducer interface for sensors, actuators and devices as a first Semantic Web 3.0 Sensor Standard. This format is defined with the help of the XML language, making it self-descriptive and easy to interpret by other network nodes.

The IED identification on the XMPP network is made by the ISO/IEC/IEEE 21451-1-4 standard. This standard adopts a 64-bit address called UUID (Universal Unique IDentifier), defined according to ISO 29161 - Automatic Identification for the Internet of Things.

The communication between XMPP network clients follows the request-response model, and the message is defined according the standard XEP-0323 - Internet of Things - Sensor Data. This standard already defines a list tags that can be used to describe the measured quantities and their units, but not all the types of physical quantities measured by the civil instruments of auscultation could be found, forcing an extension of the original standard.

In the implementation developed, all XMPP clients are identified through JIDs (Ex.: Ae558n222PPR@sicesp.com/km428) in a unique way that allows their location and resources to be made available on the network (XEP-0030). Each XMPP client is responsible for starting a persistent TCP/IP connection with the XMPP server to which this particular XMPP client belongs, thereby making the server aware of which transducers are online. The following is a typical respond message to a data request to a magnetic repression measuring instrument KM0428:

```
<message to="cesp_1@pc-i7/7516br8pdq" from="
Ae558n222PPR @sicesp.com/km428" xml:lang="en">
<fields xmlns="urn:xmpp:iot:sensordata" seqnr="1"
done="true">
<node nodeId="km428">
<timestamp value="2018-07-18T16:53:07">
<numeric unit="C" automaticReadout="true"
name="Temperatur" value="20.10376" momentary="true" />
<numeric unit="mm" automaticReadout="true"
name="Measur_Placa1" value="0.76000" momentary="true"
/>
<numeric unit="mm" automaticReadout="true" name="
Measur _Placa2" value="1.00061" momentary="true" />
<numeric unit="mm" automaticReadout="true" name="
Measur _Placa3" value="2.50038" momentary="true" />
</timestamp></node></fields></message>
```

In this IED four measures are acquired, one of which is a temperature while the others indicate the relative position of a metal plate buried in the earth filled dam of the hydroelectric plant. Thus the field identified by "numeric unit" defines the unit of measure and the value tag corresponds to its value at the time specified by the tag "timestamp value". In this particular application each IED can have from one to six measurements channels, so that the payload will not have an excessive number of bytes. In applications where the payload has a size that causes an expressive impact and in the data communication network traffic, it can be compressed using an XML format represented in binary, as defined in XEP-0322- Efficient XML Interchange (EXI) Format.

VI. Development and Testing

The XMPP technology uses distributed software architecture with client / server model, such as that used in the World Wide Web and the e-mail services. For the CESP´s system installed, an software architecture was defined consisting of a XMPP server and several XMPP clients installed in the IEDs next to the sensors. The XMPP server process is responsible for protocol management, client identification and data transport, while the XMPP client processes data values collected from the sensors. The XMPP protocol enables horizontal communication, meaning that clients can interrogate and receive data from other clients. Exploring this feature a special client was created to provide interrogation of the other clients installed in the IEDs and to store the data in a historical relational database. The figure 1 shows the structural monitoring system architecture.

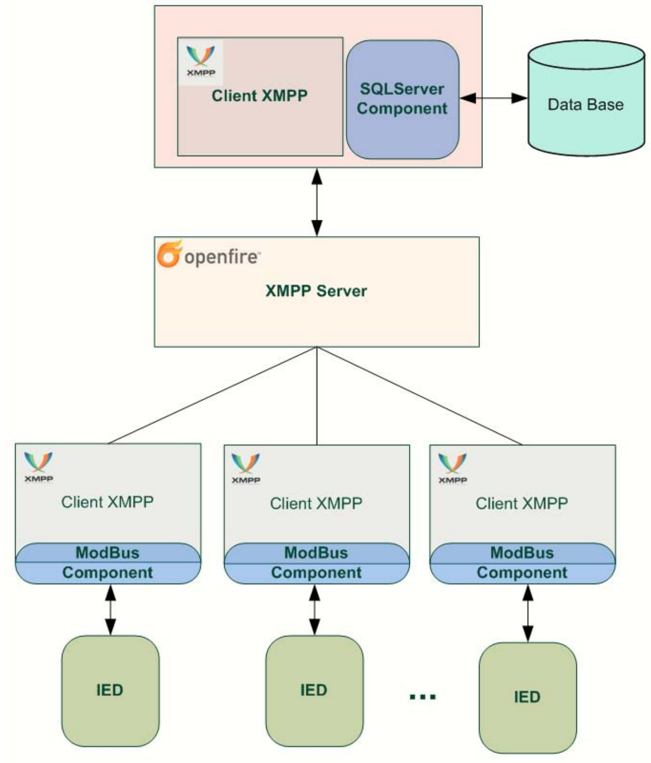

Fig. 1. Monitoring system architecture

The XMPP client processes installed in IEDs can also act as gateways interrogating other legacy IEDs using different communication protocols, and in this structural dam monitoring system, the XMPP clients collect data from ModBus/TCP protocol of old installed devices. Data received through ModBus/TCP is converted to standard Sensei-IoT XML stream and transmitted when interrogated.

The data collected by the XMPP client that interrogates the other client processes in the IEDs, stores the data on a historical relational database implemented using the corporate MS SQL Server. Other RDBMS (Relational Database Management System) can be used for data storing, by simply choosing the appropriate connection component for each different type of database supplier.

For an initial test, the LVDT (Linear Variable Differential Transformer) sensors installed in the auscultation instruments called rod extensometers (EH) and also in the triortogonal (MT) meters were chosen at different points at the dam structure of the plant. These sensors are used to monitor the structural dam blocks displacement and are connected to IEDs where an XMPP client responds to periodic queries. When interrogated, the XMPP client process converts the electrical signals into engineering values, adds the timestamp, also measurement units and transforms the payload into XML format to be forwarded to the XMPP Client that that are acquiring the data.

The XML stream received by the XMPP client process that performed interrogation is parsed and the retrieved information is saved at the database. The data stored in the DB can be viewed through a CESP´s intranet Web page. The figure 2 shows some graphs generated by the Web Server using the database information for the EH and MT instruments.

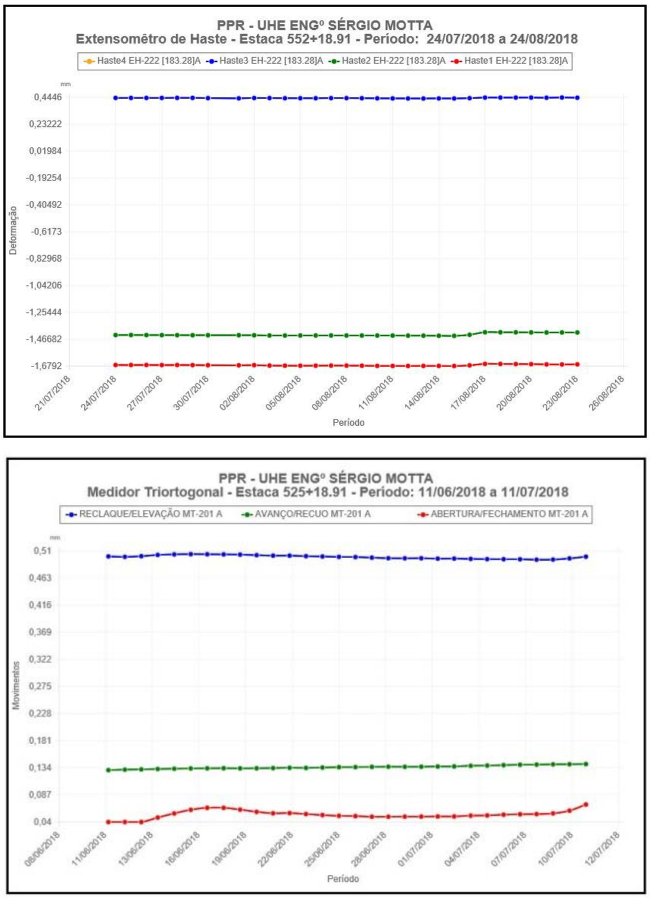

Fig. 2. Graph plotted using data collected from EH and MT sensors.

VII. Future Developments

It was also verified the benefits of a sensor localization feature, that could be implemented using the extension "XEP-0080 - User location", which allows the exchange of messages containing sensor location information. This information is very useful for system maintenance considering that Porto Primavera´s structural monitoring system is a distributed system consisting of a large number of IEDs spread between the different levels of the concrete dam and in a wide geographical area.

In addition, another service that could be helpful is the sending of commands to the IEDs, making it possible to carry out scale and measurement unit configurations, or the remote calibration of sensors, among other commands.

VIII. Conclusion

The application of the IoT concept was implemented in a system to monitor the dam structure safety of a hydroelectric power plant. A pilot is installed in the hydroelectric plant operated by CESP (Companhia Energética de São Paulo, Brazil) in Porto Primavera and uses the XMPP communication protocol for the exchange of messages between the IEDs and their respective transducers and the SCADA system. This protocol is an open, highly scalable and extensible international standard that can be configured to use different models of data communication, as well as client/server model. It also enables interoperability with legacy protocols, meets security requirements, and supports real-time communication. It is supported by a working group called Sensei/IoT, formed by ISO, IEC and IEEE through the XMPP Working Group formed by the IEEE Instrumentation and Measurement Society.

Acknowledgment

This work was supported by CESP (Companhia Energética de São Paulo, Brazil), under the R&D ANEEL (Agencia Nacional de Energia Eletrica, Brazil) program.

References

[1] NIST Framework and Roadmap for Smart Grid Interoperability, Interoperability Standards, Release 3.0 (2014), Office of the National Coordinator for Smart Grid Interoperability, National Institute of Standards and Technology, U.S. Department of Commerce. Online: https:// www.nist.gov/sites/default/files/documents/smartgrid/Draft-NIST-SG-Framework-3.pdf.

[2] Smart Grid Reference Architecture (SGAM), CEN/Cenelec/ETSI Smart Grid Coordination Group Std., Nov. 2012.

[3] M. Albano, L. L. Ferreira, L. M. Pinho, and A. R. Alkhawaja, “Message-oriented middleware for smart grids,” *Comput. Stand. Interfaces*, vol. 38, pp. 133–143, 2015.

[4] P. Saint-Andre, K. Smith, and R. Troncon, “XMPP - The Definitive Guide: Building Real-Time Applications with Jabber Technologies,” 2009.

[5] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence", RFC 6121, March 2011.

[6] P. Waher, “XEP-0323: Internet of Things - Sensor Data,” 2017. [Online]. Available: https://xmpp.org/extensions/xep-0323.html.

[7] P. Waher, “XEP-0325: Internet of Things - Control,” 2017. [Online]. Available: https://xmpp.org/extensions/xep-0325.html.

* . *