White Paper

Modernizing Electronic Flow Measurement with MQTT

Cirrus Link SOLUTIONS
844.924.7787
www.cirrus-link.com

Modernizing Electronic Flow Measurement with MQTT

By Arlen Nipper, President and CTO of Cirrus Link

A short history of EFM

In preparation for a discussion on modernizing Electronic Flow Measurement (EFM), let's start with a quick look at its history. Back in 1980 I was at Amoco Oil, and EFM, or measuring how much oil and gas flows through a pipeline, was calculated with round chart recorders.

A pin would chart the pressure and temperature over the course of a month and then someone would physically collect the round charts if the ink hadn't run out or it hadn't been eaten by mice. Then, engineers would integrate the temperature and pressure with a calculator to determine volume.

You can imagine an engineer's office with 200 of these paper charts lying around trying to make sense of the data. Sometime in the early 80s someone realized they could measure the data electronically and integrate it on a computer. Eventually the American Petroleum Institute (API) released a standard that said if you are going to measure oil and gas and then sell it, the measurement has to be very accurate.

There was a rush to automate EFM for better measurement so in the early days millions of flow computers were installed at companies all over the world. Over the next 30 years there were something like a dozen different manufacturers and a dozen protocols for EFM.

As a result of these dozens of solutions, today we see a heterogeneous, legacy infrastructure for EFM with many customer-specific, proprietary solutions. Some oil and gas companies have thousands of flow computers; therefore, upgrading to new technology is not easy. A tech has to go out to unwire/uninstall the flow computers, then reinstall new equipment. When you are looking at tens of thousands of dollars to modernize each flow computer, the cost is often prohibitive.

Not to mention, the flow computers in the field today are very accurate. They work. We all know if it isn't broken, don't fix it. The problem is not accuracy, but consistency.

The challenges with varied EFM data types

Today's flow computers are doing double duty with multiple hosts and multiple data types, which is not conducive to typical SCADA processes. From an operations standpoint the real-time operational data includes configuration, alarm, event, and history data. Per the API standard these data points are atomic, meaning they need to stay together, and you need to prove they were all gathered at the same time, which is incredibly challenging. These data points are accessed by protocol.

On the other side of the flow computer is the cash register which says – in the last hour this pipeline delivered x barrels of oil or x cubic feet of natural gas. Also, two hours ago someone changed the calibration of the flow computer. This accounting type of data is accessed differently but from the same instrument.

There are several challenges with EFM today, but the primary concern is that most of the existing networks do not have the bandwidth to meet all of these demands for data**.** Companies are making decisions they shouldn't have to – they have to decide what data to leave stranded since they cannot get to it all. They are sacrificing data availability due to bandwidth.

The reason networks do not have enough bandwidth is because EFM protocols are poll/response. On the network, the user sends out a poll, waits for the flow computer to assemble a response, the response comes back, and then they move to the next flow computer. Poll/response protocols have limits because you can't poll fast enough on the network and a lot of very valuable data is left stranded in the field.

The takeaway is that because of the state of EFM, customers are having to choose which data they want to leave stranded in the field – and that is not a good thing.

New Requirements for EFM

Even if customers were keeping the status quo it is clear there are challenges with EFM. Now – what about new requirements? The industry is abuzz with artificial intelligence, process optimization, increasing production, and more. All of these activities mean multiple data consumers, more strain on bandwidth, and more siloed data sets.

The good news is that in order to enable these new technologies, companies don't need to re-instrument their entire system. All of the equipment out there is doing the job well, we just need to get better access to the data.

Now, a plant can improve their network a little and get a bit more data. But these new requirements ask for a magnitude more data and faster than ever. Perhaps a manufacturer was asking for a set of 5 process variables, but now they want 100. Instead of every hour, they want 100 every minute. These types of improvements are possible, but changes have to be made.

Once a customer achieves this increase in data points by order of magnitude, the advanced functions become a biproduct of the data. If you see data every minute, you can enable artificial intelligence and predictive maintenance and other new technologies by using that data for real intelligence.

I worked with a customer who rolled out the changes we are about to discuss and went from getting data once per hour to once per minute. Achieving that kind of visibility required them to retrain their operators because they are seeing live data happening and they've never seen that before. They can see how the pipeline is actually operating in near real-time. Forget about optimization or artificial intelligence, just having more visibility allows them to do a better job operating the assets they have today.

MQTT solves EFM challenges

If we forgot about the past 35 years of EFM technology and eliminated any preconceived notions and invented something new today – how would we want it to work? We would want a flow computer plugged into a modern TCP/IP network. We would want to connect, authenticate, then find everything we want to know. It should be that simple.

The good news is there is a way to do that efficiently, using all of the advantages of the underlying TCP/IP networks we have today. It turns out we don't have to use poll/response, there is a better way.

I co-invented MQ Telemetry Transport (MQTT) in 1999 with Dr. Andy Stanford-Clark of IBM as an open standard for running a pipeline. The project was for Phillips 66, and they wanted to use VSAT communications more efficiently for their real-time, mission critical SCADA system. Multiple data consumers wanted access to the real time information (Figure 2).

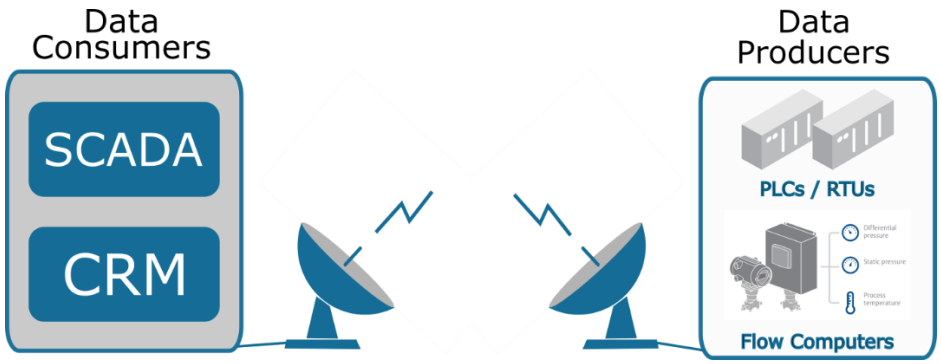

Figure 2: Phillips 66 had multiple data consumers and multiple data producers. MQTT was born to solve the problem.

MQTT is a publish/subscribe, extremely simple and lightweight messaging protocol. It is designed for constrained devices and low-bandwidth, high-latency or unreliable networks. MQTT minimizes network bandwidth and device resource requirements while attempting to ensure reliability and some degree of assurance of delivery. MQTT is based squarely on top of TCP/IP so we use those standards for best-in-class security.

To illustrate the value of MQTT's publish/subscribe methodology over poll/response, consider how much bandwidth we waste with the old method. For instance, previously companies would ask, every 30 or 60 minutes, "What is the flow rate?" and the answer would be "10." Thirty minutes later, "What is the flow rate?" "10". The question is the same, and the answer is the same – again and again. With MQTT you can find out the flow rate is 10 and then not hear the answer again unless it changes. Computers can just talk to each other whenever they want to send data – they report by exception.

Replacing a poll/response EFM network with an MQTT-based network saves 80 to 95% of bandwidth. That means stranded data can be rescued.

MQTT also allows for multiple data consumers (Figure 3). You can publish the data from an EFM device and multiple applications can consume it, all at the same time. MQTT allows for a single source of truth for data and that data is standard and open source, so anyone can use it.

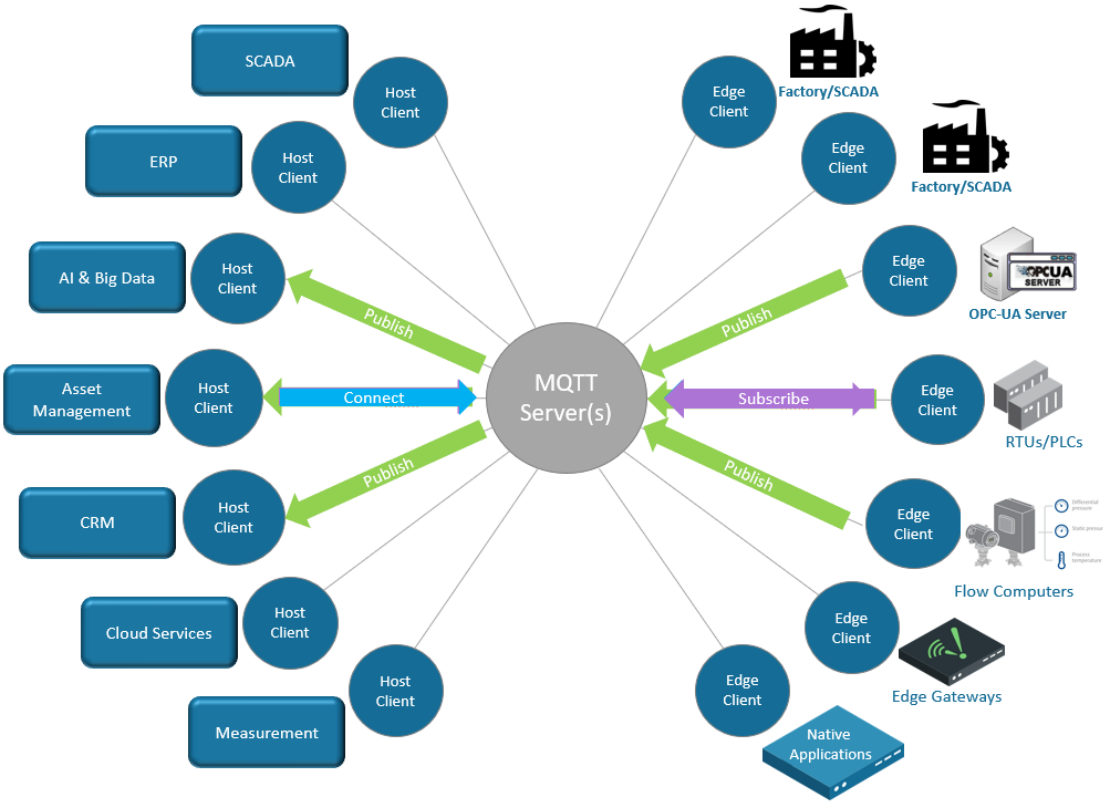

Figure 3: The basic MQTT architecture allows for unlimited clients over a publish/subscribe protocol.

MQTT is the most-used messaging protocol for an IoT solution, but it's time for the oil and gas industry to catch up for EFM. We don't have to prove that MQTT has become a dominant IoT transport – everyone is using it because it is simple, efficient, and runs on a small footprint. You can publish any data that you want on any topic.

We recently created a specification within the Eclipse Tahu project called Sparkplug that defines how to use MQTT in a mission-critical, real-time environment. Sparkplug defines a standard MQTT topic namespace, payload, and session state management for industrial applications while meeting the requirements of real-time SCADA implementations. Sparkplug is a great starting point for how to use MQTT in EFM.

One of the challenges with old EFM methods is dealing with atomic data. SCADA systems can handle single process variables, but multiple put together as a record gives them trouble. MQTT can publish this type of data as records. MQTT and Sparkplug can handle both single process and records as a whole object with data standardization, data time stamped at the source, data sent as an immutable object and the ability to store and forward.

EFM and MQTT in practice

EFM creates vast amounts of data at the source. With MQTT there is a standard way to send that data across the enterprise so no matter what piece of equipment or what end application speaks what language, it's just a piece of data, such as a temperature. All of those backend systems now can use the temperature independently; however they like. EFM data can be connected to the cloud, to big data applications – the possibilities are endless.

Let's take one last look at a customer who implemented MQTT on their EFM system. Their number one goal was to get more production from their wells. With MQTT they can adjust their well once per minute instead of once per hour, which led to an increase in field efficiency of five percent, or hundreds of millions of dollars. They were able to eliminate three different host systems each representing hundreds of rack servers and moved to a single source of truth for data with one large MQTT broker, simplifying the overall infrastructure significantly.

The misnomer in the industry is that modernizing EFM requires a large financial investment along with a lot of time and effort. MQTT is open-source, and it can be implemented on existing legacy equipment.

About the Author:

Arlen Nipper brings over 42 years of experience in the SCADA industry to Cirrus Link as President and CTO. He was one of the early architects of pervasive computing and the Internet of Things and co-invented MQTT, a publish-subscribe network protocol that has become the dominant messaging standard in IoT. Arlen holds a bachelor's degree in Electrical and Electronics Engineering (BSEE) from Oklahoma State University.