Scenario-Based Analysis of IoT Protocols at Device and Application Layers

Clay Monoceros and Yan Shi

October 11, 2018

Scenario-Based Analysis of IoT Protocols at Device and Application Layers

C A Monoceros and Yan Shi
Department of Computer Science, University of Wisconsin-Platteville

Abstract

Internet of Things (IoT) is the next big thing. Communication protocols play a critical rule in IoT solutions. Based on different needs, there is a variety of communication protocols to choose from. An important question is: which is the best protocol for a specific IoT solution? This paper provides an overview of available protocols at both device and application layers. Common usage scenarios are discussed, based on which a thorough comparison and analysis of various protocols are performed and recommendations are made to different IoT solutions.

1 Introduction

The Internet of Things (IoT) is the network of embedded devices, sensors, vehicles, and other items that are able to work with the internet, either directly or indirectly. This includes a wide range of devices, from internet-connected refrigerators, to water leakage sensors, to vehicles communicating with each other on the highway, to a variety of other devices. IoT has become an increasingly important concept in the world of technology. There are currently billions of IoT devices [1]. While this number depends on exactly which devices you count, it is evidence that IoT is becoming an important part of people's daily life.

IoT devices communicate amongst themselves, as well as through the internet to servers. Figure 1 demonstrates what a generic IoT solution might look like. Communication protocols are used to facilitate different communication needs in IoT solutions. There are protocols that define the way IoT devices are set up, and protocols defining how servers and devices communicate with each other.

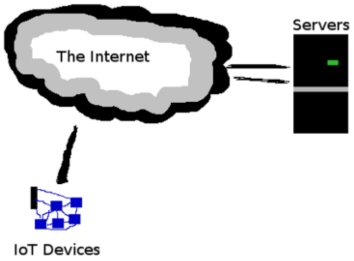


Figure 1. Generic IoT Solution

Turning toward working on IoT solutions, when people are trying to utilize an internet-connected device, they often have different needs. E.g., they might need to put the device in a small space, or some place that is not going to be touched for months or years. They might have very limited power. They might have plenty of available space and power, but need to connect to something remotely. Or they might just need to communicate with a few other IoT devices. This paper discusses some of those usage scenarios, and more specifically, looks at the available protocols that are best suited for those scenarios. We will compare available protocols for a variety of different tasks that we want to perform, and explore whether or not the protocols are up to the task.

There have been other papers on general surveying on many of the protocols covered in this document. Yassein et al. take a short look at just the application layer protocols [17]. Asim takes a similar look at a slightly different set of application protocols [3]. This paper covers similar ground concerning application layer protocols, but also approaches them from a different perspective by comparing and analyzing them in the context of different usage scenarios. This paper also covers device-layer protocols, which were missing in other papers.

2 Overview of Available Technologies

There are large amounts of communication protocols in use with loT solutions. The following is a short description of many of the protocols used in IoT today [3][17]. These protocols are broken up into two major parts: Device-layer and application layer, as can be seen in Figure 2. Device-layer is what defines the device itself. It is what you find directly in the hardware. Application layer is about communication between devices in software.

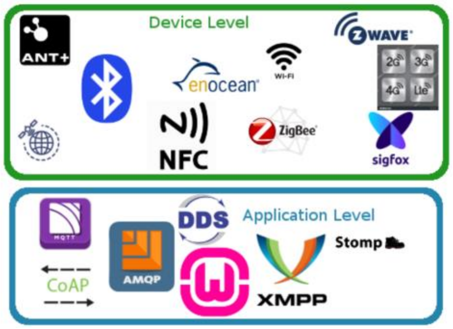
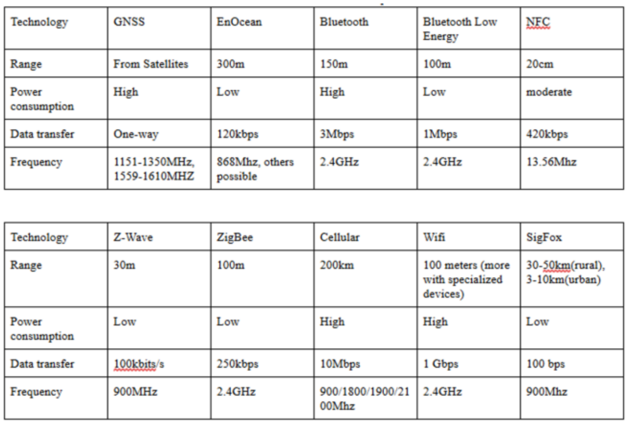


Figure 2. Protocols at Device and Application Layers

2.1 Device-Layer Protocols

Table 1 provides a basic comparison of popular device-layer protocols, each of which will be introduced in the following subsections.

Table 1. Overview of Device Layer Protocols



Technology	GNSS	EnOcean	Bluetooth	Bluetooth Low Energy	NFC
Range	From Satellites	300m	150m	100m	20cm
Power consumption	High	Low	High	Low	moderate
Data transfer	One-way	120kbps	3Mbps	1Mbps	420kbps
Frequency	1151-1350MHz, 1559-1610MHZ	868Mhz, others possible	2.4GHz	2.4GHz	13.56Mhz

Technology	Z-Wave	ZigBee	Cellular	Wifi	SigFox
Range	30m	100m	200km	100 meters (more with specialized devices)	30-50km(rural), 3-10km(urban)
Power consumption	Low	Low	High	High	Low
Data transfer	100kbits/s	250kbps	10Mbps	1 Gbps	100 bps
Frequency	900MHz	2.4GHz	900/1800/1900/2100Mhz	2.4GHz	900Mhz

2.1.1 Bluetooth

Bluetooth is designed for exchanging data over a short distance. Nils Rydbeck, and Johan Ullman first initiated it in 1989. The intent was to create wireless headsets. This grew into the IEEE 802.15.1 standard. Currently, Bluetooth SIG handles the protocol, and it is now on version 5. According to Bluetooth protocol, Class 1 devices can have a range of about 100 meters, but require higher power consumption. Bluetooth Low Energy (BLE), also known as Bluetooth Smart, is a different type of Bluetooth, which is designed for short bits of data. The standard now has a mesh-network version, with version 1.0.0 released in July 2017. Today, Bluetooth is still used for wireless headsets, but is also used for device tethering, playing multiplayer between two game systems, connect different devices, and to control home security devices.

2.1.2 IEEE 802.15.4 / ZigBee

ZigBee, which builds on top of IEEE 802.15.4, facilitates low-cost communication between nearby devices with little or no underlying infrastructure. IEEE 802.15.4 defines low-rate wireless personal area networks (LR-WPANs), and underlies ZigBee, as well as ISA 100.11a, WirelessHART, MiWi, SNAP, and Thread specifications. The standard has low-powered embedded devices in mind, and has a data transfer rate between 20 and 100 kbps, depending on the needs of the device. ZigBee transfers data faster at 250 kbps in a 10-100m range. A mesh network can extend that distance. Zigbee is often used in wireless light switches, home energy monitors, and traffic management systems.

2.1.3 Near Field Communication (NFC)

NFC is designed for extremely local wireless communication, with a range of only about 20 centimeters. It is a fairly fast protocol with transfer speeds in the hundreds of kilobytes per second. NFC is standardized under the ISO/IEC 18000-3. Contactless payment options often uses NFC, where the payment point and NFC device are very close. Other applications would be anything where you might want to put a couple devices together to transfer something, but not have the communication field extend particularly far from the device.

2.1.4 Wifi

Wifi is a familiar standard for local wireless communication. The standard is specified in IEEE 802.11b, g, and n, among others. Wifi can deliver high throughput, but because of that will also consume more power, which is problematic for a lot of IoT applications.

2.1.5 Cellular

Cellular protocols are designed for longer-range wireless communication [5]. This covers many different standards. Any technology cell phones use for their networks would fall into this category. This is a higher-power standard, and likely is going to only be used in cases where there’s enough power for a fairly significant operation. Generally this is going to be more expensive than other options, but there are some moderately constrained options like SparqEE. The next generation of cellular standards is emerging providing better performances [11].

2.1.6 SigFox LPWA / 802.15.4

SigFox is designed for communication with a longer, but not quite cellular, distance on public bands that avoid the need of paying providers. SigFox is a Low Power Wide Area network (LPWA) built off of IEEE 802.15.4.

2.1.7 EnOcean / ASK

EnOcean using ASK is a self-powered wireless technology suitable for home automation solutions. It has a range of up to 300 meters outside, and 30 meters inside a building. It is standardized with ISO/IEC 14543-3-10. Although the EnOcean company owns the standard, there are open-source versions available.

2.1.8 ANT+

ANT+ is designed to connect local devices together to collect and transfer data. Sensor devices use it a lot. ANT+ grew out of ANT. Garmin owns the standards but keeps them open access. ANT+ uses the 2.4Ghz band, as to many of the other protocols. The protocol is good at going into low-power sleep and staying in sleep mode for long periods. It can also wake up to send a bit of data, then immediately go back into that sleep mode. Devices using ANT+ can run on a coin cell for years. The ANT group is trying to broaden usage into home automation, health, and industrial applications. Garmin uses ANT+ in Garmin fitness devices, as well as their geocaching devices.

2.1.9 Global navigation satellite system (GNSS)

GNSS is designed for navigation. GPS would be an example of GNSS. GNSS is not strictly an IoT protocol, although many IoT devices use it.

2.1.10 Z-Wave

Z-Wave, controlled by Sima Designs, is intended for home automation, on the 900Mhz part of the wireless spectrum. It avoids the 2.4Ghz band used by Wifi, ZigBee, and other standards. The protocol is standardized under Z-Wave Alliance ZAD12837 / ITU-T G.9959. It supports full mesh networks of up to 232 devices, is designed for reliable, low-latency communication, and has a simpler protocol, making it easier to implement.

2.2 Application-Layer Protocols

2.2.1 Message Queuing Telemetry Transport (MQTT or MQ Telemetry Transport)

MQTT's purpose is to collect data from a device and communicate it to servers. It is a publish/subscribe, extremely simple and lightweight messaging protocol, designed for constrained devices and low-bandwidth, high-latency or unreliable networks. These principles also turn out to make the protocol ideal of the emerging "machine-to-machine" (M2M) or "Internet of Things" world of connected devices [16]. Its current version, v3.1.1, is standardized under OASIS and ISO/IEC PRF 20922.

2.2.2 Advanced Message Queueing Protocol (AMQP)

AMQP is a queuing system for connecting servers to each other. This is most appropriate for control plane or server-based analysis functions. It is standardized under ISO/IEC 19464, and its current OASIS standard is v1.0 (finalized 2012). It can do publish-subscribe messaging, but also does point-to-point routing. It was developed by John O'Hara at JPMorgan Chase in London, UK in 2003 arising from the banking industry with a focus on not losing messages.

2.2.3 Extensible Messaging and Presence Protocol (XMPP)

XMPP is designed to help in connecting devices to people. It was originally developed for instant messaging. The pace for this protocol is on the human scale, and so has things happen in seconds, rather than milliseconds or microseconds. It offers an easy way to address a device, and is good on security and scalability.

2.2.4 Data Distribution Server (DDS)

DDS can serve as a fast bus to integrate devices without contacting external servers. It has high speeds in microseconds. It offers QoS (Quality of Service) control, multicast, configurable reliability, and pervasive redundancy. Everything in the protocol is designed around speed, and it has lightweight versions available for more constrained devices.

2.2.5 Streaming Text Oriented Messaging Protocol (STOMP)

STOMP is a text-based, human readable, simple and lightweight protocol. It was formerly known as TTMP, which is similar to HTTP, and works over TCP. While it is lightweight, because of being text-based it can send big amount of data through a connection that would not be there in a different format.

2.2.6 Constrained Application Protocol (CoAP)

CoAP is designed to easily translate to HTTP, while having multicast support, low overhead, and be simple. It excels at running on UDP, which has lower overhead than TCP. While there are drawbacks to this, there are some IoT use cases where it is clearly the best. CoAP is standardized under RFC 7228.

2.2.7 Web Application Messaging Protocol (WAMP)

WAMP is a WebSocket subprotocol, though it is technically possible to use it elsewhere. Its serves as an open standard for exchanging messages between components. It is designed with WebSockets in mind, so is used in IoT devices where one would use raw websockets.

3 Usage Scenarios for Device-Layer Protocols

Having covered a variety of IoT protocols, this section covers various usage scenarios and discusses what technologies would be appropriate to use in those scenarios. Each scenario needs an IoT solution. We analyze the design needs of each IoT solution, as well as what available protocols to meet those needs.

3.1 High Power Scenario

In this scenario, the devices have ready access to power. Therefore, a high data rate is possible, and likely desired. The solution may want a short or long range, depending on the situation. It may be desirable to have a solution that easily connects to networks already deployed in the area. The solution may call for the device to work even in motion. A protocol to be used in such an IoT solution should have the following features:

- ready access to power
- high data rate

- easy connection to already existing networks
- may not want a direct connection to the internet
- may be close or fairly far away
- may need to work while in motion

The device layer protocols to consider include wifi, cellular, and Bluetooth. If needing to work while in motion, or if range beyond a few hundred meters is required, likely cellular technologies are the best choice. If there is already a wifi network available, and/or there is a need to send the largest amounts of data, wifi is a good choice. If the devices are likely to stay close together, but you want an easy way to connect to a device without also connecting the device directly to the internet, Bluetooth is a winning choice.

A good example of this scenario is watching videos online on a smartphone. Modern smartphone usually have high-capacity battery. Streaming videos requires a high bandwidth, which both cellular and wifi can provide. If a person is watching videos at home, she is very likely to choose wifi connection. If she is watching videos on a bus, cellular is a more practical option. She can also wear a wireless headset using Bluetooth to connect to her phone while watching the videos.

3.2 Low Power Scenario

Oftentimes it would be ideal to have a sensor that just sits in a spot, and only sends a signal when something bad happens. The sensor only has to communicate when something goes wrong, and likely only has to send a small amount of data. Moreover, the device may be in a difficult to access location, as its purpose is to detect some sort of problem. The device may also be far away from any base station. These likely mean that the device will need to have minimal power usage, so that it’s possible to leave a device in place, without power, for a long period of time. A protocol that is suitable for an IoT solution in such a scenario should have the following features:

- low power usage
- able to independently operate for long periods of time
- long connection range

Protocols to consider are Sigfox, Bluetooth Low Energy, and EnOcean. If the device need to be self-powered, EnOcean using ASK is a good solution. If powered by a battery and the devices are fairly close to where they have to communicate, Bluetooth Low Energy is a good solution. Sigfox is also a good solution in that case, as well as in cases where a device is further away from any base station.

A good example of this scenario is a water leakage sensor using EnOcean technologies. It uses a low-power, 868 MHz (ASK) signal to inform someone that it has detected a leak [6]. The way it detects the leak is through a fiber disk expanding, and that fiber disk expanding also powers the device. This is a case where you absolutely need an ultra-low-power communication standard, because that disk growing and shrinking is only going to provide a little bit of power, and then not be able to change in size any more until it dries out or gets wet again.

3.3 Multiple Interconnected Devices / Mesh network

This scenario is where there are multiple sensors or other IoT devices that need to communicate with each other or with a base station. In this scenario, devices may need to connect with each other without phoning home with each connection. There may or may not be readily available power, but the devices are likely to be close together. There may be an additional concern about all these devices connecting in the same wireless spectrum, especially if there are other unrelated devices also using that spectrum. An IoT solution for such a scenario should use protocols with following features:

- may or may not need higher power usage
- can handle multiple devices connected simultaneously
- can handle interference from other devices

Solutions to consider are ZigBee, Wifi, Bluetooth Low Energy, and ZWave. ZigBee is a good choice because it is designed on top of IEEE 802.15.4 with mesh networks in mind. It is also designed for having devices send information occasionally, rather than constantly, so as to reduce power usage. If interference from other devices already on the spectrum that Wifi and Zigbee use is a concern, ZWave is a good option as it functions in a different band. Wifi is a good option if there is readily available power and the network to handle it. Unlike Zigbee and Zwave, Wifi also allows the devices directly communicate with users/applications. Bluetooth Low Energy is a good option if you are looking to communicate directly with a mobile user, but have lower energy usage than you would with Wifi [8].

One of the example is an IoT solution that includes sensors and internet connections for street lights so that they can be monitored from afar and adjusted based off ambient light [4]. In this case, the device has ready access to power through the streetlights, but still has to keep the device size minimal. Since streetlights are not close enough to each other and there is usually no ready Wifi network on the streets, ZigBee or ZWave are too suitable protocols.

3.4 Low-Cost Scenarios

Many places have significant infrastructure already available, or have a budget that can handle higher-power, higher-monthly-cost solutions. This scenario assumes that one or both of these aspects are not available. In such a case, an IoT solution may consider Sigfox / IEEE 802.15.4, as it allows for low-power, and thus low-cost communications over a longer distance.

There are various areas, such as sub-Saharan Africa, where it is important to have low-cost, low-power, wide area networks. C. Pham, A. Rahim, and P. Cousin talk about how Sigfox and similar technologies “provide a better connectivity answer for IoT as several kilometers can be achieved without relay nodes to reach a central gateway or base station.”[13]

3.5 Short Range Scenario

When a very limited range is desired, but a power connection is not an issue, NFC would be used. It is designed for contactless payment, but you do not want people to be able to eavesdrop on the communication from across the room.

4 Usage Scenarios for Application-Layer Protocols

In the case of application-layer protocols, the protocols discussed in this paper have clear dividing lines, and thus this section will look at various possible scenarios and which protocol is designed for that scenario.

4.1 Device to Server Communication

A common IoT consideration is taking data from an internet-connected device and getting it to a server, and thus to wherever the internet connects to. MQTT is designed for this type of scenario [10]. It is lightweight, works on TCP, and assures the delivery of messages from device to server. Hantrakul et al. propose to use MQTT as part of parking lot guidance software, using MQTT to communicate from devices to internet servers [7].

4.2 No Loss Server Communication

While it is less directly an IoT situation, a possible important consideration when designing an IoT solution is how servers communicate with each other. In those sorts of situations, it may be important to make sure that there are no lost information in a message. Since AMQP was designed by the banking industry to have communications where no packets are lost, it is the choice for this sort of application [15]. One instance of using AMQP is StormMQ, a cloud-hosted messaging service based on AMQP [2].

4.3 Connecting Devices to People

Another important aspect with IoT solutions is that oftentimes it is important to connect devices to humans, so that they can see the information coming from the IoT devices around them. XMPP is a protocol with that human connection in mind. Since humans work at human speeds, XMPP is an appropriate technology when humans are directly in the loop. One example is for medical devices to provide ubiquitous environments to their users [12]. Since the intention is to communicate directly with the user, XMPP is a solid choice.

4.4 High-Speed Connection between Devices

When an instant response is necessary, or when devices focus on communicating with each other rather than the internet at large, it is important to have a fast, low-overhead protocol. DDS suits such needs.

DDS is fast, and excludes anything extraneous -- including constantly sending data to the internet. One example of utilizing DDS is managing generation of power [18]. Since it is desirable to be able to manage the power even if some central connection point has gone offline, a decentralized protocol like DDS works well. Moreover, high speed is essential for power routing considering the damage of having too much energy go into an object. The fast speed of DDS works well

4.5 Simple, Text-Based Implementation

Sometimes people uses an IoT device to understand the status of the backend system. This would be significantly easier if the protocols were in a format that humans can read. STOMP is designed for this purpose. It is not fast for any particular application, and thus is not the best choice for any situation other than when someone wants the messaging to be human-readable [20]. Regardless, STOMP is a good choice if someone is looking for an easy to implement and web friendly message oriented middleware.

4.6 Low Overhead or Translation to HTTP

The internet is part of the name of “Internet of Things”, and it makes a project easier if there is a simple way of connecting things directly without involving a lot of additional work. It’s also good to be able to use underlying protocols that have less overhead, like using UDP over TCP. UDP sends data without bothering with sending lots of confirmation packets back and forth. Either the data will arrive, or it will not, but there will not be data used to figure out whether or not it did. UDP is frequently used with sending streaming videos or audios, where arriving 10 seconds late is the same as not arriving at all. CoAP is oftentimes used with UDP. Since UDP offers no guarantee that the packets will arrive, the projects where it is useful for are on the opposite side of the spectrum from AQMP.

One IoT example of this would be turning on your light switch [9], as it allows a quick packet to be sent out. Confirmation is not important since in the worst case, the user will have to press a button again because the packet did not arrive the first time.

4.7 Using WebSockets

WebSockets are designed to bring low-latency communication to web applications, as connections are made and kept alive, rather than sending back and forth lots of extra overhead that is inherent in setting up connections over the internet [14]. WAMP is the protocol that brings WebSockets to IoT. For example, if a person wants to connect their Arduino to the internet, and needs to have the connection open at all times rather than letting the device repeatedly re-establish the connection, WAMP a suitable protocol to use.

5 Conclusions and Future Work

In this paper, we explored ten different device layer IoT protocols and seven different application layer communication protocols. Unique features of these protocols were discussed and compared. Based on that, we introduced multiple common usage scenarios requiring IoT solutions, analyzed the communication needs of these solutions and made suggestions on the appropriate IoT protocols to choose.

This survey paper gives a general overview of IoT protocols, which can be beneficial to audiences new to the IoT world. In the future, we plan to perform more detailed case studies comparing performances of specific protocols from different perspectives, such as communication range, data transfer speed, security, etc. Moreover, IoT is a fast-evolving field and new protocols are emerging all the time. In addition, protocols introduced in this paper are continually changing to better address their targeted issues. We will periodically update this survey to include new and updated protocols.

References

[1]M. Amadeo et al., "Information-centric networking for the internet of things: challenges and opportunities," IEEE Netw., vol. 30, no. 2, pp. 92–100, 2016.

[2]Amqp.org, "AMQP," https://www.amqp.org/about/examples.

[3]M. Asim, "A Survey on Application Layer Protocols for Internet of Things (Iot)," International Journal of Advanced Research in Computer Science, vol. 8, no. 3, pp. 996–1000, Mar. 2017.

[4]R. L. Baggam, "Smart City with Internet of Things," International Journal of Advanced Research in Computer Science, vol. 8, no. 5, pp. 1242–1245, May 2017.

[5]F. Chiti, D. Di Giacomo, R. Fantacci, L. Pierucci, and C. Carlini, "Optimized Narrow-Band M2M Systems for Massive Cellular IoT Communications," in 2016 IEEE Global Communications Conference (GLOBECOM), 2016.

[6]EnOcean, "WaterSensor eco," https://www.enocean-alliance.org/.

[7]K. Hantrakul, S. Sitti, and N. Tantitharanukul, "Parking lot guidance software based on MQTT Protocol," in 2017 International Conference on Digital Arts, Media and Technology (ICDAMT), 2017.

[8]D. Hortelano, T. Olivares, M. C. Ruiz, C. Garrido-Hidalgo, and V. López, "From Sensor Networks to Internet of Things. Bluetooth Low Energy, a Standard for This Evolution," Sensors , vol. 17, no. 2, Feb. 2017.

[9]Y. Kang and K. Kang, "Software Architecture for Building DDS Application in IoT Environment," 2016.

[10]R. K. Kodali and S. Soratkal, "MQTT based home automation system using ESP8266," in 2016 IEEE Region 10 Humanitarian Technology Conference (R10-HTC), 2016.

[11]S. Okasaka et al., "Proof-of-Concept of a Millimeter-Wave Integrated Heterogeneous Network for 5G Cellular," Sensors , vol. 16, no. 9, Aug. 2016.

[12]Y.-J. Park and K.-H. Lee, "Construction of IoT Environment for XMPP Protocol Based Medical Devices Using Powershell," Journal of The Korea Internet of Things Society, vol. 2, no. 2, pp. 15–20, 2016.

[13]C. Pham, A. Rahim, and P. Cousin, "Low-cost, Long-range open IoT for smarter rural African villages," in 2016 IEEE International Smart Cities Conference (ISC2), 2016.

[14]D. G. Puranik, D. C. Feiock, and J. H. Hill, "Real-Time Monitoring using AJAX and WebSockets," in 2013 20th IEEE International Conference and Workshops on Engineering of Computer Based Systems (ECBS), 2013.

[15]S. Schneider, "Understanding The Protocols Behind The Internet Of Things," http://www.electronicdesign.com/iot/understanding-protocols-behind-internet-things, 2013.

[16]P. Thota and Y. Kim, "Implementation and Comparison of M2M Protocols for Internet of Things," in 2016 4th Intl Conf on Applied Computing and Information Technology/3rd Intl Conf on Computational Science/Intelligence and Applied Informatics/1st Intl Conf on Big Data, Cloud Computing, Data Science & Engineering (ACIT-CSII-BCD), 2016.

[17]M. B. Yassein, M. Q. Shatnawi, and D. Al-zoubi, "Application layer protocols for the Internet of Things: A survey," in 2016 International Conference on Engineering & MIS (ICEMIS), 2016.

[18]Xiangrong Zu, X. Zu, Y. Bai, and X. Yao, "Data-centric publish-subscribe approach for Distributed Complex Event Processing deployment in smart grid Internet of Things," in 2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS), 2016.