A Survey On Connectivity And Cloud Automation Technologies For The Internet Of Things

[1]Younis Al-Kharusi [3]Mahdi Amiri-Kordestani
[2]Ahmed Al-Farsi [4]Mohammed Sarrab [5]Hadj Bourdoucen
{[1]younis-m, [2]ahmed.alfarsi, [3]amiri, [4]sarrab [5]hadj}@squ.edu.om
{[1-5]Communication and Information Research Center, [5]College of Engineering, ECE Dept.}
Sultan Qaboos University, Muscat, Oman

***Abstract*— Internet of Things (IoT) is a paradigm of networked devices (Things) that are interconnected and have the ability to exchange data without requiring any human involvement. The IoT paradigm, mostly based on FOSS systems, has contributed to shape many new directions in industries mainly hardware, communication, software, cloud and big data. Internet of Things and its related smart systems are a combination of three main parts: sensors and actuators, connectivity technologies and processes on the cloud. An essential requirement to build successful IoT projects is to identify suitable technologies for each part to be used. For instance, on the connectivity side, Bluetooth lower energy (BLE), ZigBee, Near Field Communication (NFC), Wi-Fi (IEEE 802.11ah), and LoRaWAN are some leading techniques that can be used.**
The first part of this paper reviews some major connectivity techniques used for IoT and provides a comparison of their key characteristics and the second part studies a number of open source cloud management and automation tools and reviews Docker as a suitable light-weight application container and virtualization technology. This paper also presents some fundamental differences between Docker and other competitor software such as Rocket and LXC.

Keywords—Internet of Things; Connectivity; Ducker, Open Source

I. INTRODUCTION

Many IoT projects and frameworks are built on top of a typical ecosystem that lets embedded IoT devices communicate to the Internet, and usually to a cloud system for storing and processing the data and generating advanced reports and controlling signals [1-2]. An IoT ecosystem as represented in Figure 1 can be divided into three main parts as Things, Communications, and Cloud. Each part is a composite of many technologies and can be studied from different aspects. For example, Things can be embedded computing devices attached to objects with various hardware architectures, power consumptions, system software (firmware or operating system), middle-wares, network stacks, sensors, actuators, and other components that can be studied individually.

Communication technologies can be divided into two groups; the first group is the communications over the short range and the second group is communications over the longer distances. Based on the project situations, the IoT vendor can choose the suitable technology from each group. Some leading technologies for short-range communications are Bluetooth and Bluetooth Lower Energy (BLE) also known as Bluetooth Smart, Wi-Fi (IEEE 802.11), ZigBee (IEEE 802.15.4-based), Thread (IEEE 802.15.4-based), Z-Wave, Near Field Communication (NFC), and ANT. For longer distance communications, cellular networks, proprietary Sub-GHz solutions, LoRaWAN, and WiMAX, are some more popular methods. The first part of this paper studies a number of short-range connectivity technologies such as Bluetooth, ZigBee, NFC, IEEE 802.11ah, and the emerging LoRaWAN as a suitable long distance technique for IoT.

The final stage of IoT is usually the cloud that includes many individual topics such as automation and management software, virtualization techniques, operating systems, storage models, big data paradigms, software defined networks, platform stacks and middleware, process control systems, business and artificial intelligent, reports, dashboards, etc. Every technology such as virtualization can be realized in many different types such as full virtualization, para-virtualization, and containers to address different workloads. A light-weight form of virtualization is containers that are usually divided into two groups that are application containers and the system containers. The second part of this paper studies some container-based virtualization technologies such as Docker, Rocket, and LXD.

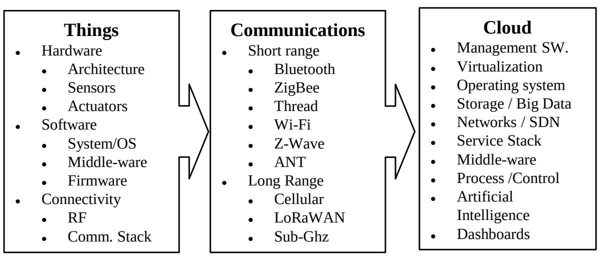

Figure 1: Example IoT ecosystem

II. IoT Communications

A. Bluetooth

Bluetooth is a wireless technology standard which is used to exchange data between devices in short-range using the frequency band of 2.4 GHz. It is predicted that over 10 billion Bluetooth enabled devices will be on the market by 2018 and Bluetooth will be the most common technique for many applications such as WPAN, M2M, and IoT [15].

Bluetooth Special Interest Group (SIG) introduced the Smart Bluetooth also know as Bluetooth Low Energy (BLE) [3] to address some IoT needs. BLE was initially available on the market since 2011 as Bluetooth 4.0 standard. BLE devices similar to Bluetooth operate in 2.4 GHz ISM band but the protocol is not backward compatible and a device can support each or both technologies concurrently.

The significant difference between Bluetooth and BLE is the concern of power consumption for communications. Many IoT applications need to exchange a small amount of data periodically and still run on the battery for many years. BLE can remains in sleep mode until a connection is initiated and therefore the power consumption is much lower than Bluetooth. The power consumption for Bluetooth is one watt, which is a lot for IoT applications, while in BLE; it is between 10 to 100 milliwatt.

Another difference between BLE and Bluetooth is connection period that is reduced to few milliseconds in BLE while it is about 100 milliseconds in Bluetooth. As a result the data rate of BLE is limited to 1 MB/s compared to 1 to 3 MB/s of Bluetooth.

BLE has an extensive support by many devices and systems and can be used to create personal area networks (PAN) to connect a few numbers of nodes. For instance iOS, Android, and Windows devices all support Bluetooth and BLE and can be connected to each other.

BLE is also an ideal solution for mobile users in short distances, while having many connected devices surround. For instance, BLE can be used for smart cars to connect all in vehicle devices such as audio system, dialing system, and speakers to get connected to the smart phone and to each other.

B. ZigBee

ZigBee [4-5] is a mesh network protocol which means that each node in this network is interconnected with another node. It uses the 2.4 GHz ISM frequency band, and is based on 802.15.4 IEEE standard. It can be used to implement a mesh network topology so that the information or data of each node can travel through other nodes until it reaches the gateway.

ZigBee is designed to carry a small amount of data over short to medium distances which may reach up to 250 meters (approximately) and can be used in local area network. In contrast to BLE that is mostly used for personal area network (PAN) and was originally designed to exchange data with big size at closer ranges.

While, the greater range coverage of ZigBee is an advantage over BLE, mesh topology has some limitations. For instance, network latency may easily happen if multiple nodes try to pass their data through a single node and to reach to the gateway or endpoint. Due to this latency of mesh networks, ZigBee and similar technologies are not suitable for many applications especially the ones with a high density of nodes. Another problem is facing the ZigBee user which is related to the mobility such as mobile nodes or parking sensors when the nodes are moving and not fixed at one point.

C. NFC

NFC [6] stands for Near Field Communication technology. NFC has some handy specifications that lead this technology to be a great solution for IoT connectivity such as the ease of use, direct or explicit interaction, low cost, and low energy. One of the most fields that NFC is used widely in smart home applications. NFC makes it easy connect, commission, and control IoT devices in smart homes [7]. For connecting, NFC enables the user to pair with devices that use other communication technologies, so NFC is helpful for devices that do not have interfaces such as sensors, light bulbs, and small appliances. For commissioning, NFC makes it easier to install the setup, logging data and maintaining the network. Regarding controlling, NFC, gives the user higher degree of flexibility and security and enable the user to control over settings and access privilege. NFC operates on 13.56 MHz frequency band with 424 Kbit/s bit rate and less than 20 cm coverage range.

D. Wi-Fi

Wi-Fi IEEE 802.11 protocol with many variants is one of the leading wireless communication technologies for IoT. Wi-Fi was originally designed to serve a limited number of devices, located indoor and with a short distance from the access point (and also from each other) with some high throughput data rates and is mostly used for local area networks (WLAN). The wide range adoption of Wi-Fi makes it the first choice for some IoT applications such as smartphones and mobile devices. However, the requirement of many embedded IoT applications is to have lower power consumption or data rates that grow the need for other technologies.

Because of the low power requirements for many IoT projects, many variants of Wi-Fi 802.11 such as a/b/g/n/ac are not suitable standards. Therefore, 802.11ah [5] standard was proposed to meet the new IoT lower-power applications. The aim of 802.11ah is to be an energy efficient standard and allow thousands of devices to work in a same physical area. The proposed coverage can also support much wider areas comparing to the older IEEE 802.11 standards and can theoretically reach to 1 Km in good conditions. Meanwhile, it can connect up to 8191 devices with by a single modern access point. IEEE 802.11ah is designed to operate on the unlicensed sub-1GHz band. One of the other interesting features of IEEE 802.11ah standards is that it can also coexistence with 802.15.4 standard

E. Thread

Thread [16] is a new wireless networking protocol which has been developed specifically to support IoT applications. It was designed to introduce many features that are not available in current standards regarding power consumption, cost-effectiveness, and security models. Thread aims to allow hundreds of devices to communicate with each other quickly and in a reliable design. Thread can directly communicate to the cloud using Internet Protocol.

Thread group released the first version of Thread at 2015. Thread protocol is based on IEEE 802.15.4, and it is designed to enable the connectivity over IPv6. Thread is capable of connecting up to 250 devices securely using wireless mesh network. Thread operates in 2.4 GHz band with a maximum 250 Kbps data rate and provides many advantages for IoT applications such as direct connection to the Internet, security, simplicity, lower power consumptions, and reliability.

F. LoRaWAN

Bluetooth, ZigBee, NFC and IEEE 802.11ah are primary standards for short range communications. The coverage of these standards will make them incompatible for wider IoT applications such as smart cities and smart grids. Some IoT applications such as industrial IoT may also require embedded devices such as sensors and actuators to be transmit over longer distances while running on batteries.

LPWAN (low-power wide area network) is an emerging standard to overcome coverage and power consumption limitations and better address wider IoT applications. LPWAN such as LoRa [8] standard drops the overhead associated using mesh techniques that ZigBee and IEEE 802.11ah use to extend their coverage such as forwarding and routing features. LoRa presents much wider coverage to short-range wireless standards such as Zigbee. LoRa (sometimes referred as LoRaWAN) can cover hundreds of square kilometers and with a single gateway can address the whole city similar to cellular networks. Cellular techniques compare to LoRa have much higher data rates and power consumptions. LoRa is suitable for IoT and M2M applications that require a tiny amount of data to be transmitted over a very long distance.

G. Comparision

A comparison of above studied wireless technologies regarding frequency, range, application, and data rate is represented at

TABLE **1**. As represented in the table, each technology has its characteristics and highlights. For instance, LoRaWAN has a comprehensive coverage among all others while 802.11ah can achieve the highest data rate.

Other than the studied standards in this paper, many proprietary wireless solutions such as ANT are also very common among IoT devices.

TABLE 1. IOT WIRELESS TECHNOLOGIES SUMMARY

	BLE	ZigBee	NFC	802.11ah	LoRaWAN
Frequency	2.4	2.4	13.56	Sub-1 GHz	EU868, EU433, US915, AS430
Range	50-100m	50-100, 300m	20cm	1 Km	2-22+ Km
Application	WPAN	WPAN, LAN	P2P	WLAN	WWAN
Rate	1Mbps	20 - 250 Kbps	106-424 Kbps	150 Kbps - 346 Mbps	0.3 - 50 Kbps

III. CONTAINERS

Instead of running a full operating system on cloud infrastructure in the form of a virtual machine, Linux and some other operating systems can provide a lightweight virtualization mechanism known as containers. For instance Linux containers can create isolated environments [9-13] to execute Linux software without having to simulating a full system. The container environment executes on the top of the Linux kernel and can be limited to a fraction of hardware resources such as processing power, network, and main memory. The rich resource management features and the strong isolated environment of Linux containers are close to full feature virtualization technologies and without their overhead costs such as separate operating system kernel for guest systems. However containers compared to full virtualization technologies have their limitations as well. For instance it is only possible to repeat the same environment of host system for guest systems. This means by using Linux as host system, we can only have Linux as guests systems as well.

Containers do not have the overall delay of the boot process and other startup costs and hence can be started and deployed much faster than any other virtualization techniques [9-13]. Meanwhile, in a large system, using many virtual machines, there will be many redundant instances of the same software such as operating system kernel and many duplicated boot volumes.

There are two common methods to utilize the containers: system containers and application level containers. System containers also known as OS containers create a full user space such as a complete Linux distribution in the container. It is also possible to install, configure and run new applications on the container just like any virtual machine instance but it's impossible to upgrade, patch or customize the kernel due the kernel is shared among the container and the host system. System containers are useful when there is a need to have a complete sort of applications. Some common technologies for creating system containers are LXC, Linux VServer, OpenVZ, Solaris Zones and BSD Jails.

As described, a system container may contain a full set of softwares including some Internet services such as mail server, or it can address a single service such as web server. For better manageability and greater security, it is possible create a minimized runtime for each application and keep the containers small and efficient. The small system containers are the basis of application level containers.

Application level containers are mostly about providing an isolated execution environment for a single software. Having each process in a separate container makes it secure, simple to maintain, update, backup, and migrate the container. Application containers are also widely accepted by system programmer and administrators as a unified run-time environment to deploy and test new software. Application containers also allow system wide configurations and the depended software to be narrowed to the container level. It's also convenience to use containers in environments with rapid software changes such as DevOps model.

A. Docker

Docker [14] is an open-source project that enables the users to automate the process of software deployments in the form of application containers. Docker utilizes containers as an additional layer for software installation, configurations, and automation. Docker can create full isolated execution environments on Linux systems by using the Linux kernel namespaces, cgroups, and other features as operating system-level virtualization for applications.

Docker is designed to be platform independent. The Docker Engine can be installed on most commercial and open source platforms such as Windows, Linux, and Mac OS and applications in the form of Docker images can be imported, moved and executed on all supported platforms. Docker essentially brings the cloud computing paradigm and flexibility to individual applications and any software suite that is capable of running within containers.

Another advantage is application containers such as Docker is designed to be very small and independent from machine resources. This will bring a greater flexibility for system administrations to migrate a Docker container from host to another. The Docker specifications usually guarantee that the runtime environment will always remain identical. A stable runtime environment as application containers is also more convenience for developers to develop and distribute software.

Docker containers provide read-only runtime to the software and all the changes such as logs and created data are kept out of the container image very similar to "diff" files.

B. Rocket

As an alternative to Docker, CoreOS has introduced Rocket project (rkt) [10] as new container runtime environment which is also based on Linux containers technology. Both Rocket and Docker projects are targeting software automation tasks and to have a greater control over software installation and deployments using application containers. While Docker has a better focus on complex environments that can support a variety of situations such as microservers, Rocket is more structured to address essential tasks in a secure and fast model for better administration and automation of software even on desktop systems.

Rocket is the first implementation of the App Container Specification (appc), which was introduced by CoreOS to better describe and standardize portable application containers.

C. LXC

LXC [9-12] is another mature and light-weight system container technology. LXC very similar to Docker and Rocket utilizes Linux containers based on Linux kernel cgroups and namespaces. As mentioned, application containers such as Docker are designed to have a single application within each container. In contrast, a system container very similar to a full Linux distribution will have a classic initial service such as "init" or "systemd" that will manage and start other services. As a result, a single LXC image can easily create sophisticated software suites such as an integrated web server with php and database services. It's also possible to create the same suites by application container, but it needs many containers for each service such as one for a web server, another for database and so forth.

D. CoreOS

Container Linux (CoreOS) is a new Linux, light-weight distribution with a focus on running typical applications inside containers. The aim of CoreOS is to create a better security, reliability, and automation model by isolating each process within a container.

The CoreOS has introduced Application Container specifications (appc) and Application Container Image (ACI) format. The Rocket is the first implementation of Application Containers that also uses ACI images. In contrast, the competitor technology, Docker does not support this specification yet, and hence Docker images are not compatible with ACI standard. Docker image format is accessible but in a proprietary form and it is possible to convert them to ACI, and other types of images with third party tools. Rocket also supports Docker and other forms of container image formats.

IV. CONCLUSIONS

A number of major connectivity techniques and cloud automation tools have been studied in this paper. Each technology has some advantages and drawbacks. In terms of short-range communication technologies, BLE has some advantage over ZigBee and NFC on simple deployments mainly because of the data rate (1 Mbit/s) and the easy connectivity option to connect to smartphones and many other mobile devices. However BLE currently does not have native support for mesh network architecture, and ZigBee and the upcoming open protocol Thread can better address those needs.

LoRaWAN has currently an extensive coverage, very low power consumption, and readymade solutions. Other technologies such as LTE-M and WiFi IEEE 802.11ah can be the future candidates for higher data rates.

Regarding the containers, there are two main options namely the system containers such as LXC and the application containers such as Docker and Rocket. Docker offers many features that can support sophisticated situations and great tools for further integration with hosting providers and public image repositories. Docker is portable and possible to be executed directly from the command line on the host machine whether it is Linux, iOS or Windows. Docker also offers an extensive documentation with many images already available.

Rocket has its own advantages such as the support for the open specifications and is considered to be more secure as Docker needs a service to be running as root to manage containers and allocate resources but Rocket doesn't need that extra service. Rocket also uses a trusted key model for signing and verifying downloaded images that make them more secure.

ACKNOWLEDGMENT

This work is part of the research that is conducted at the "*Free and Open Source Software Laboratory*" at

Communication and Information Research Center (CIRC), Sultan Qaboos University, Muscat, Oman.

References

[1] H. Derhamy, J. Eliasson, J. Delsing, and P. Priller, "A survey of commercial frameworks for the Internet of Things," in 2015 IEEE 20th Conference on Emerging Technologies Factory Automation (ETFA), 2015, pp. 1–8.

[2] M. Amiri-Kordestani, H. Bourdoucen, "A survey on embedded open source system software for Internet of Things," in FOSSC'17, 2017

[3] C. Gomez, J. Oller, and J. Paradells, "Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology," Sensors, vol. 12, no. 9, pp. 11734–11753, 2012.

[4] S. Movassaghi, M. Abolhasan, J. Lipman, D. Smith, and A. Jamalipour, "Wireless Body Area Networks: A Survey," IEEE Commun. Surv. Tutorials, vol. 16, no. 3, pp. 1658–1686, 2014.

[5] E. Georgakakis, S. A. Nikolidakis, D. D. Vergados, and C. Douligeris, "An Analysis of Bluetooth, Zigbee and Bluetooth Low Energy and Their Use in WBANs," in Wireless Mobile Communication and Healthcare: Second International ICST Conference, MobiHealth 2010, Ayia Napa, Cyprus, October 18-20, 2010. Revised Selected Papers, J. C. Lin and K. S. Nikita, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 168–175.

[6] V. Coskun, B. Ozdenizci, and K. Ok, "A Survey on Near Field Communication (NFC) Technology," *Wirel. Pers. Commun.*, vol. 71, no. 3, pp. 2259–2294, 2013.

[7] P. Goarin, "The IoT at home: NFC makes it easier to create smart environments," NXP SEMICONDUCTORS, 2016. [Online]. Available: http://blog.nxp.com/iot/the-iot-at-home-nfc-makes-it-easier-to-create-smart-environments. [Accessed: 28-Dec-2016].

[8] "White Paper: A Technical Overview of Lora and Lorawan," The LoRa Alliance: San Ramon, CA, USA, 2015. [Online].

[9] W. Felter, A. Ferreira, R. Rajamony, and J. Rubio, "An updated performance comparison of virtual machines and Linux containers," in 2015 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), 2015, pp. 171–172..

[10] Z. Kozhirbayev and R. O. Sinnott, "A performance comparison of container-based technologies for the Cloud," Futur. Gener. Comput. Syst., vol. 68, pp. 175–182, 2017.

[11] R. Morabito, J. Kjällman, and M. Komu, "Hypervisors vs. Lightweight Virtualization: A Performance Comparison," in 2015 IEEE International Conference on Cloud Engineering, 2015, pp. 386–393.

[12] K. T. Seo, H. S. Hwang, I.-Y. Moon, O.-Y. Kwon, and B.-J. Kim, "Performance comparison analysis of linux container and virtual machine for building cloud," Adv. Sci. Technol. Lett., vol. 66, no. 105–111, p. 2, 2014.

[13] P. Padala, X. Zhu, Z. Wang, S. Singhal, K. G. Shin, and others, "Performance evaluation of virtualization technologies for server consolidation," HP Labs Tec. Rep., 2007.

[14] D. Merkel, "Docker: Lightweight Linux Containers for Consistent Development and Deployment," Linux J., vol. 2014, no. 239, Mar. 2014.

[15] "With an Installed Base of 10 Billion Devices Expected in 2018, Bluetooth will be an Essential Tool for Building the Internet of Everything," Allied Business Intelligence (abiresearch), 2013. [Online]. Available: https://www.abiresearch.com/press/with-an-installed-base-of-10-billion-devices-expec/. [Accessed: 25-Jan-2017].

[16] S. Lawson, "Samsung, ARM, And Nest Launch Thread, A Low-Power Network For The Smart Home," PCWorld, 2014. [Online]. Available: http://www.pcworld.com/article/2454120/samsung-arm-and-nest-launch-thread-group-to-tie-home-iot-devices-together.html.