Performance Analysis of Virtual Communication Between Thermal Cameras and Contact-tracing Applications for COVID-19

Shariq Haseeb, Aisha Hassan A. Hashim, Othman O. Khalifa, Ahmad Faris Ismail

Abstract: *Late 2019 and a significant part of early 2020 have witnessed the outbreak of Coronavirus disease-19 (COVID-19) across the world. As a desperate attempt to control the virus spread, many countries are enforcing measures to restrict large concentration of people at one place and are implementing some form of contact tracing mobile application to quickly track close interactions between people. Furthermore, as businesses come out of lockdowns, they are required to record the temperature of all visitors and staffs that move in and out of their premises. Some businesses are employing medical grade contactless thermal imaging cameras for scanning temperature. Communication and sharing of information between the contact tracing application and thermal cameras would make an effective tool against COVID-19. However, there is a disconnect between the contact tracing applications and contactless thermal imaging solutions because they employ different communication stacks, platforms, data formats, and protocols. Furthermore, any kind of middleware to mediate between the cameras and the mobile applications would render the solution useless because of the induced latencies. In this paper, we are proposing to virtualize the communication between the cameras and mobile applications so that they could communicate and interoperate over a common protocol stack. We further model and simulate the proposed virtualized communication algorithm, under various topologies and configurations to comprehensively evaluate the performance, scalability, and deployment feasibility. The simulation aptly and efficiently evaluates the results for latency, energy, and bandwidth consumption parameters.*

Keywords **:** *Cloud, COVID-19, Fog computing, Virtual devices and protocols.*

I. INTRODUCTION

Emerging technologies are being deployed across the world to fight the COVID-19 pandemic [1]. From Malaysia to Korea to China and to most of Europe, contact tracing applications are being deployed to exchange Bluetooth beacons between mobile phones in order to measure close proximity between the people. Towards this, Apple and Google have also decided to collaborate and develop a common Software Development Kit (SDK) to enable iOS and Android devices to communicate and exchange Bluetooth information with each other [2]. Singapore has launched an app known as “TraceTogether” [3]. The app is designed to use Bluetooth technology to detect close proximity between individuals so that they could stay informed and quarantine themselves if someone in close proximity were to be tested positive of COVID-19. Malaysia has implemented a similar technology that combines GPS location along with the Bluetooth beacons in their “GerakMalaysia” applications [4]. Many European countries are also exploring similar options for their citizens. Hong Kong has adopted a slightly different approach with a RFID wristband that is linked to a smartphone app, where the app has the ability to inform authorities if a geofence has been breached [5]. South Korea has employed the data correlation approach between Credit Card transactions, smartphone GPS information, CCTV footage and manual data from physical interaction with people [1]. On top of contact tracing, most countries have taken heed of World Health Organization (WHO) guidelines [6] for post lockdown business continuity and have mandated regular body temperature measurements of visitors for up to 3 months of records. The manual contact intensive process of log keeping and temperature measurements with a thermometer is not a very practical approach. Hence, a lot of businesses have adopted the use of thermal imaging cameras with facial recognition capabilities to identify people, measure and record temperature information that can conveniently be retrieved on as and when needed. An integrated solution, which combines facial recognition, thermal imaging and contact tracing would be a formidable solution to combat the uncontrolled spread of COVID-19. However, contact tracing applications operate in the application layer domain, while the thermal imaging cameras from different vendors employ their own platforms. It is virtually impossible for the two domains to interoperate unless some form of standardization is done.

Manuscript received on May 25, 2020.
Revised Manuscript received on June 29, 2020.
Manuscript published on July 30, 2020.

***** Correspondence Author

Shariq Haseeb*, Faculty of Engineering, International Islamic University Malaysia, Gombak, Malaysia. Email: **shariqkhan1**@yahoo.com

Aisha Hassan A. Hashim, Faculty of **Engineering**, International Islamic University Malaysia, Gombak, Malaysia. Email: aisha@iium.edu.my

Othman O. Khalifa, Faculty of Engineering, **International** Islamic University Malaysia, Gombak, Malaysia. Email: khalifa@iium.edu.my

Ahmad Faris Ismail, Faculty of Engineering, International Islamic University Malaysia, Gombak, Malaysia. Email: faris@iium.edu.my

Retrieval Number: B3929079220/2020©BEIESP
DOI:10.35940/ijrte.B3929.079220
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

The use of mediators such as middleware or a gateway would induce too much latencies and would not be effective enough for the authorities to respond to the crisis [7][8][9] in real-time. This problem is not unique to the current technology landscape for COVID-19. In fact, these interoperability issues have been incommodious for most Internet of Things (IoT) deployments.

Just like the thermal camera, which can be classified as a constrained device [10], IoT deployments have been employing Sensors and Actuators as constrained devices within their architecture shown in Fig. 1. The bottom-most layer of the architecture is where the constrained devices reside, and they have the capability to either directly communicate with their cloud-based IoT backend or they may communicate through an intermediary gateway. Gateways are typically less constrained devices [10]. Application communicates through the IoT backend if they need any information from the constrained devices. The IoT backend, brokers the data on behalf of the application if the constrained device is active or responds with an error i.e. the device is unreachable. In the context of contact tracing applications, the constrained devices are Bluetooth sensors, GPS sensors and wristbands, while the gateway is the individual's smartphone device. In the context of thermal imaging system, the camera and their PTZ motors are the constrained devices, while the embedded computer that they connect to is their gateway. The typical IoT architecture of Fig. 1 has been widely employed across all types of IoT deployments. Even though most IoT deployment follows the same architecture, their execution of the IoT backend couldn't be more heterogeneous. As a matter of fact, to-date there have been more than 300 variations of the IoT backend from different vendors and product types [11]. The variations of the IoT backend renders one IoT deployment incompatible with the other.

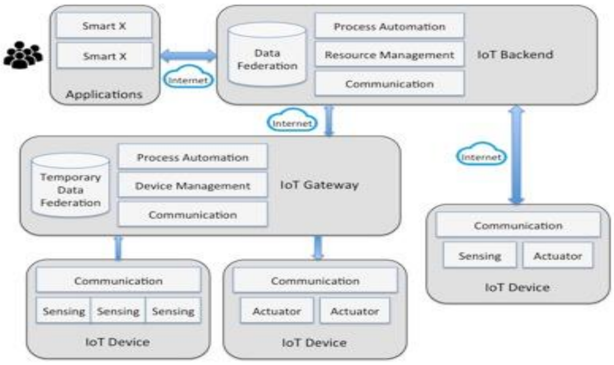

Fig. 1. Reference IoT Architecture

In order to manage the constrained devices and heterogeneous IoT architecture, we are proposing to virtualize the devices and their communication stacks. Virtual devices would no longer be physically constrained and would not need to depend on the IoT backend for communication. This will enable exchange of information between systems, resulting in a more useful tool for combating the COVID-19 crisis.

II. DEFINITION OF A CONSTRAINED DEVICE

Constrained devices, as the name suggests, are limited capability devices. They are usually deployed as leaf nodes of an IoT network to perform the highly specialized task of either sensing or actuating [12]. Due to their specialized nature, they are usually constrained for power so that they can be deployed remotely. They also have limited processing, memory and storage abilities to reduce unnecessary costs. Most constrained devices operate in vulnerable radio conditions with minimal human intervention and highly asymmetrical link characteristics. The Internet Engineering Task Force (IETF), in RFC 7228 [13] has classified the constrained devices into 4 main classes. Class 0 represents devices that are very constrained on memory and CPU capabilities, hence they can't directly communicate over the Internet without the help of a gateway. Class 1 devices are quite constrained in code-space; hence they are unable to employ a TCP/IP based communication stack. Class 2 devices are mostly energy-constrained devices that can't operate for long periods without recharging. They also need energy-efficient operating systems and protocol stacks. Finally, Class 3 and beyond devices may have energy constraints but are not limited by processing ability or any specific type of protocol [10]. Thermal imaging cameras and wristbands would typically belong to Class 0 because they don't have much computation capabilities and communicate to the attached servers over proprietary protocols. Due to the capabilities of the smartphones, Bluetooth sensors and GPS sensors would normally be classified as Class 2 constrained devices.

III. EXISTING RESEARCH FOR MANAGING INTEROPERABILITY ISSUES

Interoperability related research has been the focus of many researchers. There have been many attempts and proposals both by the industry and academia. In several cases, there have also been joint efforts with the device vendors to unify communications between constrained devices. Interoperability related research can be largely classified into the following major area:

- Architecture and Platform Standardizations: IoT backend standardization has been the focus of many researchers [14][15][16][17]. They have proposed Application Program Interfaces (APIs) between IoT platforms and backends to interoperate. This is a great step towards interoperability but has its own limitations. Firstly, APIs syntaxes are predefined and do not carry and semantics information, resulting in interoperability between platforms from within the same vertical industry. Secondly, such integrations would enable the IoT backend to security vulnerabilities and hence, deter many from making such partnerships. Finally, even after successful integration, there is no actual device to device communication. Platform to platform integration only enables data exchange.
- MAC Layer Standardizations: research in this area focuses mainly on enabling interoperability at the MAC layer by either standardizing the MAC protocols or by proposing a handful of low-power optimized MAC alternatives for constrained devices [7][18][19][20].

There is a fundamental flaw in these proposals, as any form of adoption would require proper certifications and large vendor alliances. Most vendors would prefer to stay out because of the involved costs and furthermore, they can offer differentiated services if they don't conform to alliances.

- IP Layer Standardizations: research at the IP layer is largely focused on low-power, small-footprint TCP/IP protocol adaptations for constrained devices [21][22][23][24]. This is a great idea for achieving interoperability through IP. However, most Class 0 and Class 1 devices would not be able to adapt these protocols due to constrained code-space and processing capabilities. Hence, this will offer interoperability solutions for only a handful of Class 2 devices.
- Infrastructure Assisted Standardizations: research in this area have focused mainly on emerging technology such as Software Defined Networks (SDN) and Network Function Virtualization (NFV) [25][26][27]. SDN approach to interoperability is by splitting the data plane from the control plane. SDN proposes data abstraction from the device and hence it enables interoperability at the data layer. This solves the problem for data exchange between sensors, but it does not solve the problems faced by actuators. Furthermore, if the device itself is abstracted, it becomes impossible to manage the devices and this is very critical for remote IoT deployment scenarios.
- Sensor Data Virtualization: the latest research trend is focused on the virtualization of sensing nodes so that their data is always available on the cloud [28][29]. This approach again suffers from the same limitations of the SDN approach.

Our approach to solving interoperability issues builds on top of the sensor virtualization approach but it extends the concept of virtualization to actuators and their network. The approach, termed as Constrained Device Virtualization Algorithm (CDVA) is detailed in the following sections.

IV. CDVA OPERATIONS

The first step in virtualizing a constrained device is to identify its physical characteristics that are critical to be implemented as a software instance. The critical characteristics that define a constrained device are as follows:

- Device ID: this is a unique address assigned to a device like the camera or the wristband or any other sensor deployed in the contact tracing apps. The ID does not need to be standardized as long as the device gateway is able to identify the constrained device with the help of this ID. It will be used to eventually maintain map between protocols.
- Device IP: this is the IP address of the constrained device if it supports TCP/IP stack.
- Gateway ID: this is a unique ID of the constrained device gateway. In the case of the contact tracing apps, the gateway would be smartphone while, in the case of thermal cameras, the gateway would be its embedded PC for image processing.
- Gateway IP: this is the IP address assigned to the gateway for two-way communication.
- Device Physical Architecture: these characteristics define the processing capabilities of the constrained device. It is not a must to know this but will be useful for protocols like virtual SNMP.
- Device Resource Utilization: these parameters could include CPU utilization, memory utilization, battery utilization. They are helpful parameters to determine the status of the constrained devices.
- Data/Status/Service/Network Streams: these parameters are continuously synchronized between the constrained device and the virtual device. They help to define the behavior of the virtual device. The status stream will also be used to perform cleanup if the device leaves the network or is down. In the case of contact tracing apps, the data streams would comprise of GPS and Bluetooth beacon data, the status stream would determine if the device is reachable, service streams would instruct the virtual device to behave like a GPS and Bluetooth. While, the network stream would carry the latency and throughput information. In case of the case of thermal imaging cameras, the data streams would comprise of high-fever triggers, motion triggers and facial recognition data, status streams would represent if the cameras are up, service streams would dictate the capabilities of the virtual camera and the network stream would carry latency and throughput information to the virtual device.

Once the constrained device characteristics have been identified, the second step is to virtualize the physical devices based on Fig. 2, where the physical attributes layer represents the physical characteristics of the constrained device like GPS, Bluetooth, RFID, etc. The Software Process layer contains software codes that can be executed on-demand as they are executed on the actual physical device. The runtime environment mimics the communication system of the physical device in the virtual domain. The API layer is essentially the interface for interacting with the virtual devices.

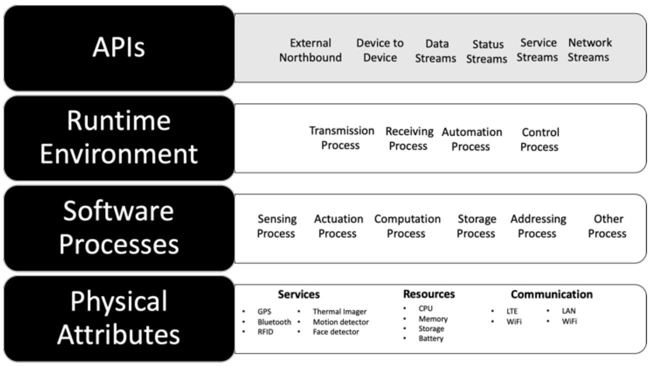

Fig 2. System Architecture of Virtual Constrained contact tracing and thermal imaging devices

The third step, after representing a physical device as a virtual device involves the assignment of IP addresses to the virtual device so that it can communicate over the Internet.

As soon as the virtual device is created, the host virtual machine creates a virtual communication interface to broker an IP address from addressing servers within the network. Once the address has been assigned, the device becomes reachable over the network with its IP address. The last step is more of a bookkeeping step where the host virtual machine maintains the IP address to device ID binding for management purposes.

The binding solves two purposes, firstly, it allows the IP traffic to be routed back to the physical device if required and secondly, it is used for clean-up process once the physical device leaves the network or becomes unreachable. The flowchart incorporating all the steps is illustrated in Fig. 3.

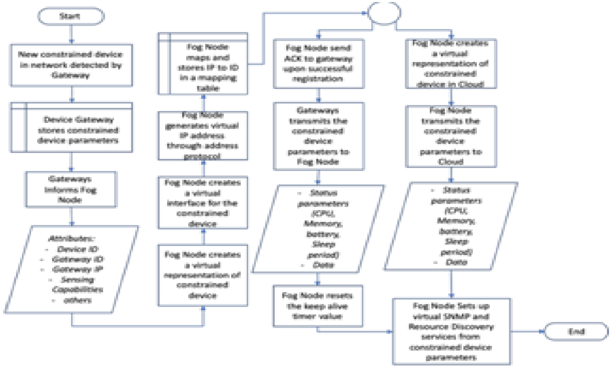
Start
New constrained device in network detected by Gateway
Device Gateway stores constrained device parameters
Gateways informs Fog Node
Attributes:
- Device ID
- Gateway ID
- Gateway IP
- Sensing Capabilities
- others
Fog Node creates a virtual representation of constrained device
Fog Node creates a virtual interface for the constrained device
Fog Node generates virtual IP address through address protocol
Fog Node maps and stores IP to ID in a mapping table
Fog Node send ACK to gateway upon successful registration
Gateways transmits the constrained device parameters to Fog Node
- Status parameters (CPU, Memory, battery, Sleep period)
- Data
Fog Node resets the keep alive timer value
Fog Node creates a virtual representation of constrained device in Cloud
Fog Node transmits the constrained device parameters to Cloud
- Status parameters (CPU, Memory, battery, Sleep period)
- Data
Fog Node Sets up virtual SNMP and Resource Discovery services from constrained device parameters
End

Fig 3. CDVA Flowchart Illustrating the Four Steps for Virtualizing Constrained Devices

The virtualized constrained device can be easily adopted in the reference IoT architecture with minor changes to the IoT Backend. Additional capabilities on the IoT backend would require it to support the CDVA algorithm and host some new APIs for the virtual devices to be accessible by the applications. It wound also need to support virtual device control and management functions. The constrained device gateways would need to collaborate with the IoT backend through APIs to transmit data, status, service and network streams on behalf of the constrained devices. Fig. 4 shows how the constrained device from the contact tracing app and thermal imaging cameras will exist within the reference IoT architecture. The constrained devices form the bottom-most layer of the architecture, their gateways act as an intermediate layer to the IoT backend, while the IoT backend hosts the virtual instances of the constrained devices.

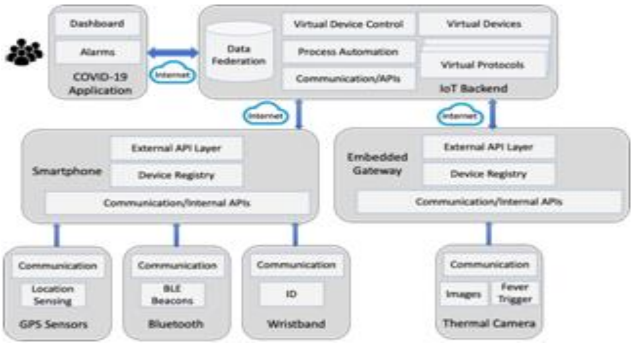

Fig 4. Virtual constrained devices within the IoT architecture

V. APPLICATION MODEL OF THE INTELLIGENT COVID-19 APPLICATION

The Intelligent COVID-19 application model for simulation is presented in Fig. 5. Camera, embedded processors, smartphones and wrist bands are the physical devices within the model. There are five main types of data streams or edges in this model. They are described as follows:

- Image streams: these are raw images captured by the thermal sensors for processing in the embedded processor.
- Fever trigger streams: these are processed thermal images of individuals with a high fever at that moment.
- GPS location stream: these are continuous location and point of interest information gathered by the GPS sensors.
- BLE beacon streams: there are signals emitted by Bluetooth to uniquely identify a smartphone. These beacons are picked up by other devices to determine the close proximity of one mobile phone to another.
- ID stream: this is a unique ID of a person stored in a database somewhere on the cloud. This ID can be read by readers to determine the identity of an individual.

The requirements for the intelligent COVID-19 application is to geotag every fever trigger image and combine it with a unique ID from the wrist band or BLE beacon so that the individual with high fever can be identified immediately and can be quarantined. Furthermore, every other person they have been in contact with can also be notified through the contact tracing app. In order for this to happen, all physical devices must be able to communicate with each other in real-time.

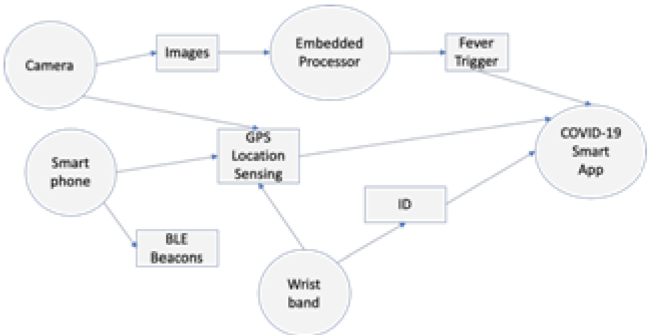

Fig 5. Application Model of the Intelligent COVID-19 Application.

VI. SIMULATOR CONFIGURATIONS

The application model is simulated over 3 hours, for 100 cameras, 100 smartphones and 100 RFID wristbands. Link characteristics between physical devices follow the WiFi model and the constrained devices have been virtualized in a Fog/edge computing environment to minimize latency between physical and virtual devices. The intelligent COVID-19 application resides in the Cloud datacenter connected over the Internet. Table 1 describes the edges of the intermediate modules for simulation. Table 2 describes the deterministic transmission frequency configurations of all the sensors within the smartphone and the RFID wristband.

TABLE- I: Edges of Intermediate Modules

Type	CPU Length	NW Length
IMAGES_STREAM	1000	20000
FEVER_TRIGGER	2000	2000
GPS_STREAM	1000	100
BLE_STREAM	500	100
RFID_STREAM	500	100

TABLE- II: Transmission Frequency

Type	NW Length	Interval
IMAGES_STREAM	20000	25/s
GPS_STREAM	100	5ms
BLE_STREAM	100	5ms
RFID_STREAM	100	5ms

The simulation has been conducted over a varying probability of data availability in the virtual devices and the results are discussed in the next section.

VII. PERFORMANCE EVALUATION

The average end-to-end latency of IP communication and data retrieval from the constrained devices, by the Cloud-based intelligent COVID-19 application, is shown in Fig. 6. It can be observed from the figure, that the average latency of the intelligent COVID-19 application reduces as an increasing number of data streams are found at the virtual devices. Average end-to-end latency approaches the requirements for real-time communication of 50ms around 50% probability of data hit at virtual devices. This is because, for hit probability lower than 50%, the virtual devices have to probe the physical devices and that introduces a much larger end-to-end latency. It is therefore recommended to allow the virtual network to converge before communicating with the virtual devices. Typical convergence time for the network would depend on the data stream with the lowest transmission frequency.

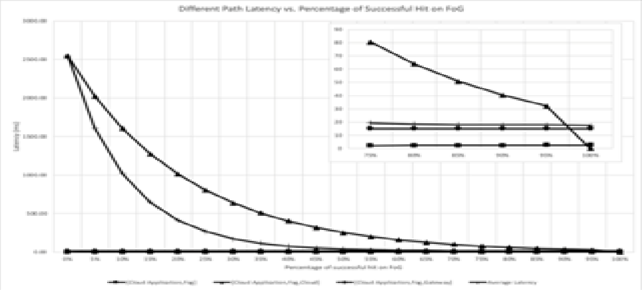

Fig 6. End-to-end Latency Experienced by Intelligent COVID-19 Application

Computation latencies of application model edges are shown in Fig. 7 where the longest computation time is required to analyze the raw thermal images for fever. The next most computation-intensive virtual processes are the GPS coordinates to location computation and processing of images for further analytics. The last two edges are not that computation heavy. An interesting observation here is that average computation latencies are not really affected by the hit probability.

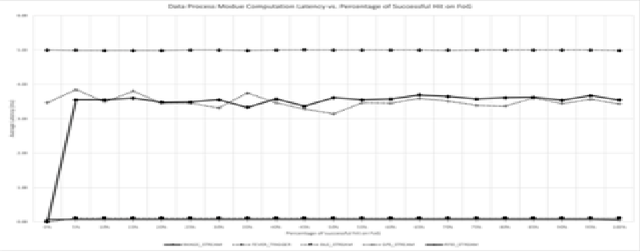

Fig 7. Computation Latency Experienced by Different Application Edges

Deployment of virtual devices on the Fog/Edge or the Cloud computation nodes would incur extra energy requirements. Fig. 8 shows the graph to energy consumption on intermediate nodes and also explores the cost of execution on the Cloud in the form of millions of instructions, which will eventually translate to cost of using Cloud computing resources in actual deployments. It can be observed from the figure that changes in the hit probably does not have any significant effect on the energy consumption in intermediate network nodes. However, in order to run the CDVA, the Fog/Edge nodes have to invest in about 14 Megajoules of energy over the period of simulation. It can also be concluded from the figure that cost of execution on the Cloud reduces as the hit probability increase because the intelligent COVID-19 application can get the response much faster and has to execute for a shorter period of time.

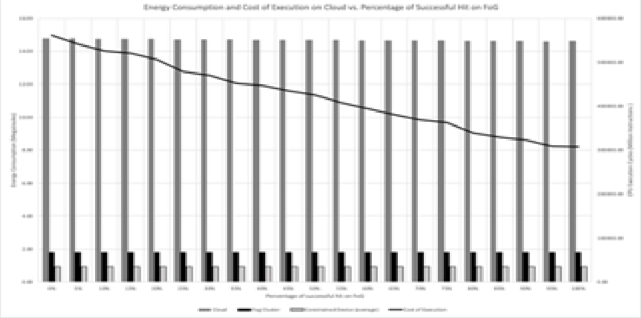

Fig 8. Energy Consumption in Network Nodes and Cost of Execution in Cloud

Fig. 9 shows the total network bandwidth utilization over increasing hit probability. It can be observed from the figure that the overall bandwidth requirements are lower when the hit probability is higher. This is predominantly due to the fact that there is a lesser need to communicate with the physical devices when the virtual devices are able to supply the required data instantly.

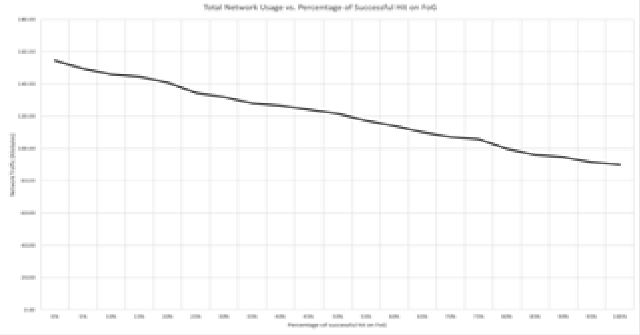

Fig 9. Network Bandwidth Utilization

Retrieval Number: B3929079220/2020©BEIESP
DOI:10.35940/ijrte.B3929.079220
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

VIII. CONCLUSION

Lack of interoperability between physical devices from different manufacturers is limiting the ability of governments to effectively combat the COVID-19 pandemic. In this paper, we, therefore, propose the CDVA algorithm that has enough potential to successfully resolve these interoperability issues by virtualizing the constrained physical devices and their communication protocols.

A virtual device is no longer constrained by physical limitations and can manifest itself within a Fog/Edge-based architecture or even on the Cloud datacenters. The virtual device would leverage on the available computation to support virtual versions of the standard communication protocols. The simulation results show that, with sufficient convergence of the virtual device network, the much needed, real-time communication can be achieved between devices present in the intelligent COVID-19 application. CDVA does introduce higher energy consumption on Fog/Edge devices and that could be considered for further research aimed to design application scheduling algorithms for Cloud and Fog/Edge-based architectures.

REFERENCES

1. J. Louis, “Using tech to fight COVID-19,” pp. 1–5, 2020.
2. F. S. (Apple), “Apple and Google partner on COVID-19 contact tracing technology,” 2020. [Online]. Available: https://www.apple.com/newsroom/2020/04/apple-and-google-partner-on-covid-19-contact-tracing-technology/. [Accessed: 03-May-2020].
3. S. Gov, “TraceTogether.” [Online]. Available: https://www.tracetogether.gov.sg/. [Accessed: 03-May-2020].
4. S. Kwan, “App-ly via Gerak Malaysia | The Star Online,” *The Star*, 2020. [Online]. Available: https://www.thestar.com.my/news/nation/2020/04/26/app-ly-via-gerak-malaysia. [Accessed: 03-May-2020].
5. S. Kwan, “Covid-19: Hong Kong’s wristbands allow quarantined to wander free | The Star Online,” *The Star*, 2020.
6. WHO, “Interim guidance,” 2020.
7. A. Chio, G. Bouloukakis, C. H. Hsu, S. Mehrotra, and N. Venkatasubramanian, “Adaptive mediation for data exchange in IoT systems,” in *ARM 2019 - Proceedings of the 2019 18th Workshop on Adaptive and Reflexive Middleware, Part of Middleware 2019*, 2019, pp. 1–6.
8. A. M. Mohd, A. Suhaimi, Q. S. M. Faisal, and S. Haseeb, “Evaluating QoS performance of Streaming Video On both IPv4 and IPv6 Protocols,” *Proc. Spring Simulait. Multiconference*, vol. 1, pp. 109–116, 2007.
9. G. Bouloukakis, N. Georgantas, P. Ntumba, and V. Issarny, “Automated synthesis of mediators for middleware-layer protocol interoperability in the IoT,” *Futur. Gener. Comput. Syst.*, vol. 101, pp. 1271–1294, Dec. 2019.
10. Nagasai, “Classification of IoT Devices - CISO Platform.” CISO Platfom, Bangalore, 2017.
11. M. Noura, M. Atiquzzaman, and M. Gaedke, “Interoperability in Internet of Things: Taxonomies and Open Challenges,” *Mob. Networks Appl.*, pp. 1–14, Jul. 2018.
12. R. Sutaria and R. Govindachari, “Making sense of interoperability: Protocols and Standardization initiatives in IOT,” in *2nd International Workshop on Computing and Networking for Internet of Things (CoMNet-IoT) held in conjunction with 14th International Conference on Distributed Computing and Networking (ICDCN 2013)*, 2013, pp. 2–5.
13. RFC, “Terminology for Constrained-Node Networks [RFC 7228],” *Ietf Lwig*. pp. 1–17, 2014.
14. T. Pflanzner and A. Kertesz, “A survey of IoT cloud providers,” in *2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2016 - Proceedings*, 2016, pp. 730–735.
15. *Interoperability and Open-Source Solutions for the Internet of Things*, vol. 10218. 2017.
16. R. Morabito, I. Farris, A. Iera, and T. Taleb, “Evaluating Performance of Containerized IoT Services for Clustered Devices at the Network Edge,” *IEEE Internet Things J.*, vol. 4, no. 4, pp. 1019–1030, 2017.
17. S. Haseeb, A. H. A. Hashim, O. O. Khalifa, and A. F. Ismail, “Connectivity, interoperability and manageability challenges in internet of things,” in *AIP Conference Proceedings*, 2017, vol. 1883.
18. R. Sanchez-Iborra and M. D. Cano, “State of the art in LP-WAN solutions for industrial IoT services,” *Sensors (Switzerland)*, vol. 16, no. 5. 2016.
19. M. A. S. Mosleh, G. Radhamani, M. A. G. Hazber, and S. H. Hasan, “Adaptive Cost-Based Task Scheduling in Cloud Environment,” *Sci. Program.*, vol. 2016, 2016.
20. L. Davoli, M. Antonini, and G. Ferrari, “DiRPL: A RPL-based resource and service discovery algorithm for 6LoWPANs,” *Appl. Sci.*, vol. 9, no. 1, p. 33, Dec. 2018.
21. S. Raza and T. Voigt, “Interconnecting WirelessHART and legacy HART networks,” in *DCOSS ’10 - International Conference on Distributed Computing in Sensor Systems, Adjunct Workshop Proceedings: IWSN, MobiSensors, Poster and Demo Sessions*, 2010, pp. 1–8.
22. M. M. Feroz and A. K. Kiani, “SHIM6 Assisted Mobility Scheme, an intelligent approach,” *2013 IEEE 10th Consum. Commun. Netw. Conf. CCNC 2013*, pp. 725–728, 2013.
23. B. Tank, H. Upadhyay, and H. Patel, “A survey on iot privacy issues and mitigation techniques,” in *ACM International Conference Proceeding Series*, 2016, vol. 04-05-Marc.
24. O. Gaddour *et al.*, “Demo Abstract: Z-Monitor: A Monitoring Software for IEEE 802.15.4 Wireless Sensor Networks,” 2018.
25. Y. Jararweh, M. Al-Ayyoub, A. Darabseh, E. Benkhelifa, M. Vouk, and A. Rindos, “SDIoT: a software defined based internet of things framework,” *J. Ambient Intell. Humaniz. Comput.*, vol. 6, no. 4, pp. 453–461, Aug. 2015.
26. P. Martinez-julia and A. F. Skarmeta, “Empowering the Internet of Things with Software Defined Networking,” *White Pap. IoT6-FP7 Eur. Res. Proj.*, 2014.
27. S. Bera, S. Misra, S. K. Roy, and M. S. Obaidat, “Soft-WSN: Software-defined WSN management system for IoT applications,” *IEEE Syst. J.*, vol. 12, no. 3, pp. 2074–2081, 2018.
28. A. Gupta and N. Mukherjee, “Rationale behind the virtual sensors and their applications,” *2016 Int. Conf. Adv. Comput. Commun. Informatics, ICACCI 2016*, pp. 1608–1614, 2016.
29. A. Gupta and N. Mukherjee, “A Cloudlet Platform with Virtual Sensors for Smart Edge Computing,” *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8455–8462, Oct. 2019.

AUTHORS PROFILE

Shariq Haseeb is an IoT Specialist at TMR&D and REDtone IoT, Shariq Haseeb has architected several IoT solutions under the Smart City concept. He is the key architect being the TMR&D Smart Helmet, Smart Streetlight, Smart Vehicle and REDtone IoT’s citizen engagement solution (CitiAct). Prior to joining REDtone IoT, Shariq was Head of Project in MIMOS Berhad. He has over 12 years research and prototyping experience in the field of computer communication and networks. He has worked cross functionally within protocol, network, infrastructure and device development to invent bleeding edge technology for global market space. He has filed more than 50 patents within Malaysia and Internationally and has more than 35 publications in journals and conferences.

Aisha Hassan Abdalla Hashim received her Ph.D in Computer Engineering (2007), M.Sc. in Computer Science (1996) and B.Sc. in Electronics Engineering (1990). She won the Best Graduating Ph.D Student Award during the IIUM Convocation ceremony in 2007. She joined IIUM in 1997 and is currently a Professor at the Department of Electrical and Computer Engineering. Professor Aisha has taught several courses related to Communication and Computer Engineering and is actively involved in curriculum development and programme accreditation. She has been a member of the Department Board of Studies for several years. She received the Best Teacher Award during IIUM Quality Day in 2007. Prof. Aisha has been appointed as external examiner/visiting professor/adjunct professor at different universities.

Professor Aisha who is actively involved in research and postgraduate programmes, has published more than 200 journal/conference papers, and supervised/co-supervised more than 60 Ph.D/Master students. She received the Promising Researcher Award in 2009 during IIUM Quality Day. She has also received many medals/awards in different national/international research exhibitions. One of her research exhibitions won the Promising Commercial Value Award (Second Runner Up) in IRIIE 2014. As a researcher, she has secured research grants from IIUM, Ministry of Higher Education (MOHE) and Ministry of Science, Technology and Innovation (MOSTI). She has actively contributed as a reviewer /technical committee member in many journals/conferences.

Professor Aisha has established several teaching/ research networks between IIUM and overseas universities. She has participated in initiating several MoUs as well as encouraging the PhD Student Mobility programme between IIUM and overseas Universities.

Othman Omran Khalifa received his Bachelor's degree in Electronic Engineering from Garyounis University, Libya in 1986. He obtained his Master degree in Electronics Science Engineering and PhD from Newcastle University, UK in 1996 and 2000 respectively. He worked in industry for eight years and he is currently a Professor and at the department of Electrical and Computer Engineering, International Islamic University Malaysia. His area of research interest is Communication Systems, Digital image / video processing, coding and Compression, Wavelets, Fractal and Pattern Recognition. Prof. Khalifa is a Charter Engineer (CEng) and Senior member of IEEE, USA and a member IET, UK. and a member of the Council of Professors of Malaysia. Prof. khalifa was the chairman of the International Conference on Computer and Communication Engineering (ICCCE), 2006, 2010, 2012, 2014. Prof. Khalifa has extensively contributed through his writings in international journals, conferences and book. He published more than 450 publications including 10 books. He is a member of many international advisory boards for many international conferences a member of many editorial boards of many international.

Ahmad Faris Ismail is a Professor and the Dean of Engineering at the International Islamic University Malaysia (IIUM). He was the IIUM Deputy Rector (Research & Innovation) from July 2009 until June 2013. He served as the Dean of Engineering from 1997 until 2009. He obtained his B.Sc. in Chemical Engineering, in 1988, from the University of Houston, USA, and Ph.D in Engineering from Rice University, USA, in 1993. He is the Chief Editor for the IIUM Engineering Journal. Prof. Ismail was a Visiting Academic at the University of Southern Queensland in 2014 and a Visiting Scientist at Kyoto University in 2004. He has been invited as keynote speakers at various international conferences and congresses. He is also a co-inventor for at least eight filed patents of research products and has published more than 200 papers in refereed journals and conference proceedings. His research topics include energy and environment, simulation and modelling, computational fluid dynamics, combustion, and nanofluids.

Retrieval Number: B3929079220/2020©BEIESP
DOI:10.35940/ijrte.B3929.079220
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication