From Conventional to Microservices and NVF: A survey of the paradigm shift in IoT Network System Architecture

Subhransu Das[1], Suman Bhunia[2], Subhabrata Barman[3], Santanu Koley[4], Pinaki Pratim Acharjya[5]

[1, 3, 4, 5] Department of Computer Science and Engineering, Haldia Institute of Technology, Haldia, India

[2] Miami University, 205 H Benton Hall, 501 E High St, Oxford, OH 45056

ABSTRACT

The contents of this paper can be segregated into two major parts. The first part presents a summarized survey of the conventional Wireless Sensor Network (WSN) architectures along with several issues and problems of it. while the second part deals with architectural modifications that is observed modern Internet of Things (IoT) networks. The description of modern IoT network contains the description distributed computing paradigms (such as Edge and Fog computing) along with summarized survey of latest techniques such as Microservices and computational off loadings. The discussion and simulation results led us to enlist some open issues in the concluding section.

Keywords - Wireless Sensor Network (WSN), Internet of Things (IoT), Docker, Data Inconsistency, Network Virtual Function (NVF)

I. INTRODUCTION

From conventional Wireless sensor network (WSN) to modern Internet of Things (IoT), networking solu- tions have always provided the ease of availing necessary data to the end users. With continuous research and availability of newer technologies, sensor network systems have become more promising, scalable and se- cured. The conventional WSN architecture which used to have a handful of sensor nodes to collect and forward data, has now turned into an IoT space, where several hundred and thousand sensors are connected with fog or edge computing servers. According to some recent economic survey, researchers have estimated that the IoT could will have an economic impact of $11 tril- lion per year by 2025, contributing almost 11% of the world's overall economy. It is also forecasted that ap- proximately a trillion of IoT devices would be deployed worldwide by 2025 [13]. To handle this increasing vol- ume of data, several changes have been made to the conventional WSN architecture. In conventional architecture, the sensors used to forwarded the collected data to the clouds via gateways. Although the cloud was the main computational and storage unit of the data, yet the physical distance between gateway and cloud was still not a matter of consideration, since the data was delay tolerant itself. This trend was broken when medical data sensors, Body area network sensors, video surveillance equipment demanded real time processing of the data along with su cient QoS from the network. The physical distance between gate- way and cloud became a matter of concern, since it introduces considerable latency and significant disruption in real time processing of the data [8]. The most promising solution to this problem would have been to bring the computation unit nearer to the sensors [4]. In edge and fog computing we move edge or fog servers nearer to the sensors to mitigate the issue of real time data processing. Additionally deployment of Edge servers along with microservice or Network Virtual Function (NVF) will allow it handle heterogeneous traffic (such as video data, Data from standard IoT sensors like smart lock, smart phone or even body sensor data) in real time. Finally the collaboration of newer technology like Microservices, NVF along with upcoming 5G network infrastructure has shifted the paradigm from conventional WSN age to modern Edge computing era. Moreover, we comprehensively investigate fault tolerance, resource management, or microservices in as a measure of technological shift in the architecture. Along with this survey, this paper provides the research directions in for further improvement of resource allocation and scheduling, fault tolerance, simulation tools, and Fog-based microservices.

II. CONTRIBUTIONS

The primary goal of this paper is to present a clear depiction of the paradigm shift from conventional WSN to modern IoT network architecture. To instantiate our discussion we introduce several new terms like Service reliability, Data-Inconsistency etc. We outline the key contributions of this work for easy and quick reference of the readers.

i. In this paper, we define, formulate and describe the "Data Inconsistency" issue which is quite common in traditional WSN Systems.

ii. We adopt separate simulation experiments (per formed in different platforms) to instantiate the paradigm shift along with performance measures.

III. CONVENTIONAL WIRELESS SENSOR NETWORK

In conventional sensor network, a number of application specific sensor devices used to be deployed over a terrain to collect the data (shown in figure 1). These tiny battery-operated sensor devices sense the external environment and incorporate the sensing results into transferable data units. A typical sensor device used to have TCP-IP protocol stack along with three service layer (planes) i.e, i. Power management plane ii. Mobility Management plane iii.Task Management Plane as shown in figure 1 [1].

The Power management plane supplies the re- quired power to all it's hardware equipment. Mobility management plane takes care of the discovery, connection establishment with the neighbors.

The task manager, on the other hand, manipulates all tasks such as sleep, Tx/Rx etc. The efficiency of task manager allows power management plane to sustain the battery life of the sensor device.

Along with data acquisition, sensor devices perform routing as well. Tiny battery-operated sensors forward the data of other sensors towards the repository using either a reactive or proactive or hybrid routing proto- col. In proactive routing protocol, the routing table is updated on any change on the network. In contrast, in reactive routing, the routing table is created and maintained on demand (whenever a Tx/Rx operation is to be performed). In hybrid routing protocol both reactive and protocol operations are performed together. Normally, Hybrid Routing protocol divides the overall deployment zone in several regions to deploy reactive and proactive routing protocols in di erent zones. Since Tx/Rx operation has the maximum energy depreciation, the task manager introduces additional enhancements such as, i. Sleep scheduling and ii. Energy aware Clustering for energy optimization.

i Sleep Scheduling: Sleep is a state of wire- less sensor device (like other state such as, Idle, Tx/Rx

etc.) where the energy depreciation remains minimum and the radio remains Idle. A sensor node needs to listen the medium continuously, in order to maintain a synchronized communication with the network. There- fore, we need to schedule the sleep properly to maintain the communication with the network along with minimizing the energy depreciation. Numerous approaches such as TDMA based, SDN based, AI based sleep scheduling can be found in [12] [23] [17].

ii. Clustering: Clustering is a technique in which, a group of sensors chose a node to be the cluster head node amongst them [21]. All the nodes which are present in the cluster forward their data to the cluster head, accordingly, cluster head forwards the data to the repository. The primary goal of clustering is to optimize energy consumption and to minimize routing overhead of the network.

III.i. ISSUES WITH CONVENTIONAL SENSOR NETWORKS

Two major issues of conventional sensor network are i. Service reliability ii. Application Specific deployment.

III.ii. SERVICE RELIABILITY

Tiny battery-operated sensor nodes have low computational power along with limited energy budget. Wire- less sensor nodes often su er internal and external hazard which deprecate their normal work ability and Tx/Rx capability as well. Internal issues like battery failure, circuit issues may cause the sensor to be dead. Exter- nal hazard such as high temperature, fog, rainfall often deprecates the normal Tx/Rx capability of a sensor node. Under adverse environmental condition when the communication radius of a single or a group of sensor node shrinks, and thus it originates dynamic communication loop holes in the network. To better visualize the such service reliability issues, we use the notion of "Data inconsistency" in this section. The performance modeling and simulation results of "Data inconsistency" depicts almost a loss of throughput under adverse environmental conditions.

Remark 1. Data inconsistency (Di): Data Inconsistency is the difference between expected amount of data to be acquired in a unit time slot, and the amount of data actually received in that time slot. From the definition, we can write that

$$\frac{E_{dv} - R_{dv}}{\Delta t_i}$$

(1)

We assume that, there exists N senor nodes in the network and all of them are working properly. If they are scheduled to send P packets (with ρ packet length each) in Δti time slot, than we can expect NPρ. data volume to be received. The internal problems such as End of Battery or circuit failure have persistence effect on the sensor device. However, the effects of external hazard is mostly transient and dynamic in nature. To model the randomness of such system we use probabilistic approach in the subsequent discussion.

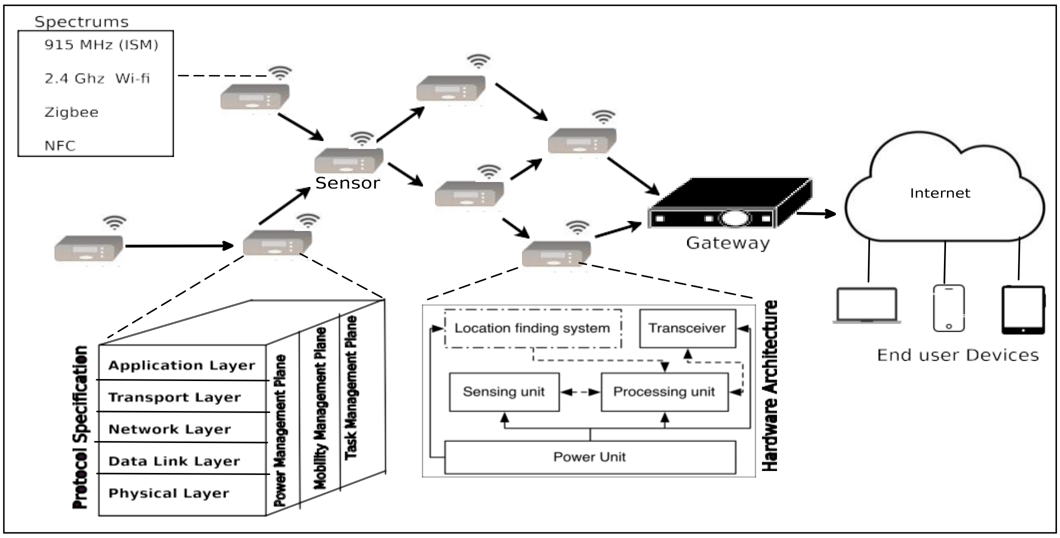

Fig.1: Protocol, hardware specification and architectural overview of the Conventional WSN Network.

IV. DISTRIBUTED COMPUTING SCENARIO

The conventional architecture primarily focused on the deployment of the sensor node. However, the communication between the gateway and cloud also plays a key role in the performance. Every time the gate- way receives a packet it forwards the packet to the cloud for computation, analysis and storage purpose. As the number of sensors in the network increases the frequency of the message also increases. In such a scenario, a significant delay and overhead can be observed in transmitting every message from the gateway to the far distant cloud. The physical distance between the cloud and the gateways invokes the performance issue in following cases as well.

In this era of IoT our daily life is being eased with numerous sensor devices, This trend not only introduces different types of sensors but also generates huge volume of data. The physical distance between gateway and cloud makes it cumbersome to handle such huge data volume.

Along with the Physical distance, the link quality between the gateway and cloud also plays a key role in monitoring and handling real-time multimedia data.

The obvious solution for this problem is to bring the computational unit nearer to the gateway or the sensors itself. This is exactly where edge and cloud computing come into the picture.

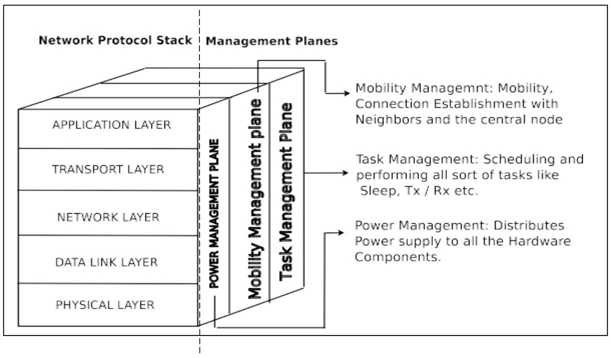

Fig.2: Protocol Stack of a Conventional Sensor device and Gateway

Symbol	Expression
D_i	Indicator of Data Inconsistency
E_{dv}	Estimated data volume
R_{dv}	Received Data volume
Δt_i	Observation time slot
P	No of Packets expected to receive in Δt_i time span
	Packet length

Table 1: Table of Symbols

IV.i. MICROSERVICE BASED NETWORK

Cloud based IoT architecture was further enhanced with several virtualization technologies like Virtual Ma- chines (VM), LXC, Docker etc. A comprehensive performance Analysis on these technologies shows the sig- ni cant improvement in CPU utilization in Docker and LXC environment amongst others[16]. In this study the performance of di erent virtualization technologies were tested on slightly different hardware and software platforms. In [6] authors tested the performance of KVM and Docker in same Hardware platform on IBM System x3650- M4 server. However, in both of these studies Docker outperformed other competitive tech- nologies such as LXC or Hypervisor. Authors in [3] presented a comprehensive study on KVM and Docker to present some interesting results. Out of several experimental results of [3] we list a few here. i. In a standard application platform docker reports 28.27% less CPU utilization along with almost 66% less mem- ory consumption than KVM ii. When the number of users increased from 10 to 100 the CPU utilization of Docker increased nominally compared to 3.2% increase in KVM. IN the same test case docker uses only 126 Mb extra memory compared to 200 Mb extra mem- ory to serve the increased number of users. Authors in [22] presented a several valuable results with numerous experiments. The outcomes of these experiments include the followings i. In this paper, We observe a significant difference between booting times of VM and Docker applications. The booting time of docker and VM is depicted as 11.095 seconds and 159.3 seconds respectively. ii. This paper clearly depicts the difference between service hosting of VMs and containers. In [19] authors presented a case study on the performance on real time multimedia data transmission on Software Defined Radio (SDN)-NVF paradigm. The results also corroborated the suitability of Virtualization tested under Amazon EC-2 cloud platforms. There are several other studies which clearly shows the superiority of Docker container-based application among other competitive solutions. The architecture of Docker based container is simple yet powerful. Dockers come up with "docker managers", a tool that allows you to replicate docker instances. Along with replication docker manger orchestrates resource allocation, service management and storage managements as well. Thus, using multiple managers, a network architecture can be formed easily. A typical Docker-IoT architecture is pro- posed in [18] where three interconnected mangers per- form the overall task as shown in Figure 3. There are three Docker mangers presented in [18] namely i. Node Manager ii. Swarm Manager iii. Cloud Manger. The Swarm Manager uses the Raft Consensus algorithm to maintain a consistent swarm state. In [20] authors presented a multi-layer network architecture in the perspective of Software defined Radio(SDN). The proposed architecture has three layers namely i. Sensing Layer ii. Mediation layer and iii. Enterprise Layer, with the following functionalities

i. Sensing Layer: This layer consists of cyber-physical systems which are engaged in collecting data. ii. Mediation Layer: This layer can be compared to the edge layer (fog) which is having data storage and Machine Learning Unit (MLU). The MLU provides basic intelligence to the gateway to meet real-time requirements. On the other hand the storage unit works as a transient or persistent storage of unit of the data. iii. Enterprise Layer: This layer is similar to the cloud architecture. It is deployed to serve three basic needs, namely data warehousing, data treatment and business logistics.

Authors in [11] used Occopus cloud orchestrating tool [9] to present extended architecture on IoT docker environment. The architecture consists of two main units i. Occopus Cloud unit and ii. multiple Swarm manager units. A standard Docker container based IoT architecture can be extended for several services based solutions as well. Video or real time data has always been an attention seeker for performance analyzers. [14] came up with two basic agendas i. Setting up deep learning framework on Raspberry Pi to analysis real-time video feed of surveillance camera. ii. Calculation of overhead of the overall process.

The video feed used in this paper was 1080p (1920 × 1280) at 10 frames per second(fps). The output depicted a nominal overhead along with minuscule resource consumption on a simple hardware like Pi3 Model B (used as Gateway). There are several other Docker based layered architecture layered IoT system architectures and cloud (fog/Edge) computing paradigms are present in [7] [10] [10].

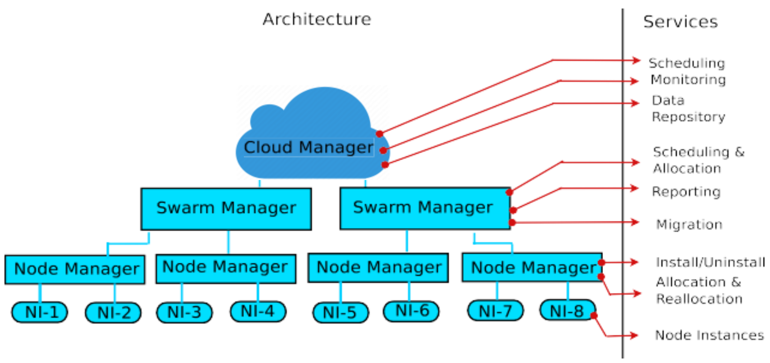

Fig.3: Architecture and services of IoT-Docker

V. CONCLUSIONS

In recent years microservice based applications have gained enormous popularity due to their simplicity, portability and easy deployment across different hardware platforms [2]. However, the development started with simple Wireless Network architectures where a group of nodes forward there collected data to the repository. We use Network Simulator-3 (ns3) to design a conventional Wireless Sensor Network system.. Our first set of experiments shows a general depiction of throughput with node density. The packet length and the was 512 bytes for this experiment. The results are quite intuitive, as we see the increment of throughput with increasing number of nodes in the network. When the number of node increased from 10 to 30, we observe a sharp inclination (upto 2.5 times) in the throughput. The experiment was performed with AODV and OLSR routing protocols where OLSR routing protocol outperformed AODV in all test cases. OLSR protocol outperformed AODV with providing 25% extra throughput in the final test case. To address the Data inconsistency, we deprecate the normal Send() method of different nodes for a pseudo- random time interval. We deploy 10 nodes in a 300m × 300m terrain and deprecate the sending capacity of different nodes in pseudo-random time interval. We observe the fluctuation of throughput in Figure 4b. As the existence of misbehaving nodes greatly hampers the packet delivery ratio, we observe a great inconstancy of throughput in this experiment.

REFERENCES

[1]] Akyildiz, Ian F. ; Su, Weilian ; Sankarasubramaniam, Yogesh ; Cayirci, Erdal: A survey on sensor networks. In: IEEE Communications magazine 40 (2002), Nr. 8, S. 102 114

[2] Babaria, Urvashi: Why IoT Development Needs Microservices and Containerization. (2019). URL: https://www.einfochips.com/blog/

[3] Chae, MinSu ; Lee, HwaMin ; Lee, Kiyeol: A performance comparison of linux containers and virtual machines using Docker and KVM. In: Cluster Computing 22 (2019), Nr. 1, S. 1765 1775

[4] Dastjerdi, Amir V. ; Buyya, Rajkumar: Fog comput- ing: Helping the Internet of Things realize its potential. In:Computer 49 (2016), Nr. 8, S. 112 116

[5] Dehury, Chinmaya K. ; Sahoo, Prasan K.: Design and im- plementation of a novel service management framework for IoT devices in cloud. In: Journal of Systems and Software 119 (2016), S. 149 161

[6] Felter, Wes ; Ferreira, Alexandre ; Rajamony, Ram ; Rubio, Juan: An updated performance comparison of vir- tual machines and linux containers. In: 2015 IEEE interna- tional symposium on performance analysis of systems and software (ISPASS) IEEE (Veranst.), 2015, S. 171 172

[7] Khazaei, Hamzeh ; Bannazadeh, Hadi ; Leon-Garcia, Alberto: Savi-iot: A self-managing containerized iot plat- form. In: 2017 IEEE 5th international conference on future Internet of Things and Cloud (FiCloud) IEEE (Veranst.), 2017, S. 227 234

[8] Kim, Beom-Su ; Park, HoSung ; Kim, Kyong H. ; God- frey, Daniel ; Kim, Ki-Il: A survey on real-time communi- cations in wireless sensor networks. In: Wireless communi- cations and mobile computing 2017 (2017)

[9] Laghrissi, Abdelquoddouss ; Taleb, Tarik: A survey on the placement of virtual resources and virtual network func- tions. In: IEEE Communications Surveys & Tutorials 21 (2018), Nr. 2, S. 1409 1434

[10] Letondeur, Loic; Ottogalli, Francois-Gael ; Coupaye, Thierry: A demo of application lifecycle management for IoT collaborative neighborhood in the Fog: Practical exper- iments and lessons learned around docker. In: 2017 IEEE Fog World Congress (FWC) IEEE (Veranst.), 2017, S. 1 6

[11] Lovas, Robert ; Farkas, Attila ; Marosi, Attila C. ; cs, Sandor ; Kovacs, J zsef ; Szaloki, Adam ; Kadar, Botond: Orchestrated platform for cyber-physical systems. In: Complexity 2018 (2018)

[12] Ma, Junchao ; Lou, Wei ; Wu, Yanwei ; Li, X-Y ; Chen, Guihai: Energy effcient TDMA sleep scheduling in wire- less sensor networks. In: IEEE INFOCOM 2009 IEEE (Veranst.), 2009, S. 630 638

[13] Manyika, James ; Chui, Michael ; Bisson, Peter ; Woet- zel, Jonathan ; Dobbs, Richard ; Bughin, Jacques ; Aharon, Dan: Unlocking the Potential of the Internet of Things. In: McKinsey Global Institute(2015)

[14] Mendki, Pankaj: Docker container based analytics at iot edge video analytics usecase. In: 2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU) IEEE (Veranst.), 2018, S. 1-4

[15] Morabito, Roberto ; Farris, Ivan ; Iera, Antonio ; Taleb, Tarik: Evaluating performance of containerized IoT services for clustered devices at the network edge. In: IEEE Internet of Things Journal 4 (2017), Nr. 4, S. 1019 1030

[16] Morabito, Roberto ; Kj llman, Jimmy ; Komu, Miika: Hypervisors vs. lightweight virtualization: a performance comparison. In: 2015 IEEE International Conference on Cloud Engineering IEEE (Veranst.), 2015, S. 386 393

[17] Mostafaei, Habib ; Montieri, Antonio ; Persico, Vale- rio ; Pescap , Antonio: A sleep scheduling approach based on learning automata for WSN partialcoverage. In: Journal of Network and Computer Applications 80 (2017), S. 67 78

[18] Noor, Shahid ; Koehler, Bridget ; Steenson, Abby ; Ca- ballero, Jesus ; Ellenberger, David ; Heilman, Lucas: IoTDoc: A Docker-Container based Architecture of IoT- enabled cloud system. In: 3rd IEEE/ACIS International

Conference on Big Data, Cloud Computing, and Data Sci- ence Engineering Springer (Veranst.), 2019, S. 51 68

[19] Premsankar, Gopika ; Di Francesco, Mario ; Taleb, Tarik: Edge computing for the Internet of Things: A case study. In: IEEE Internet of Things Journal 5 (2018), Nr. 2, S. 1275 1284

[20] Rufino, Joao ; Alam, Muhammad ; Ferreira, Joaquim ; Rehman, Abdur ; Tsang, Kim F.: Orchestration of con- tainerized microservices for IIoT using Docker. In: 2017 IEEE International Conference on Industrial Technology (ICIT) IEEE (Veranst.), 2017, S. 1532 1536

[21] Singh, Sunil K. ; Kumar, Prabhat ; Singh, Jyoti P.: A survey on successors of LEACH protocol. In: Ieee Access 5 (2017), S. 4298 4328

[22] Tseng, Chia-Wei ; Tseng, Fan-Hsun ; Yang, Yao-Tsung ; Liu, Chien-Chang ; Chou, Li-Der: Task scheduling for edge computing with agile VNFs on-demand service model toward 5G and beyond. In: Wireless Communications and Mobile Computing 2018 (2018)

[23] Wang, Yanwen ; Chen, Hainan ; Wu, Xiaoling ; Shu, Lei: An energy-e cient SDN based sleep scheduling algorithm for WSNs. In: Journal of Network and Computer Applica- tions 59 (2016), S. 39 45