A Review on Fault Tolerance Techniques towards Robust and Secure IoT Platform

[1]ARSHI QUERESI, [2] DEEPTI JAIN
SCHOOL OF RESEARCH AND TECHNOLOY, PEOPLE'S UNIVERSITY, INDIA

Abstract- Internet of things (IoT) is that the assortment of the many totally different technologies and networks. Because of increase the quality of the web of Things (IoT) applications, fault prediction becomes a vital challenge in interactions between human and good devices. Fault prediction is one amongst the key factors to attain higher arrangement the IoT applications. Failures will happen due to malfunctioning of hardware and software system put in into the devices. Fault tolerance will increase system accessibility and responsibility by creating systems sturdy to failures and proactive enough to tackle failures. Fault tolerance is introduced at totally different bailiwick layers of the web of Things (IoT) as fault will occur at any of the layers. IoT based mostly systems area unit fallible and fragile, resulting in the creation of faults within the entire network, inflicting wrongful conduct again and again. Several varieties of faults do happen among IoT networks due to the node, link, protocol conversion, and communication failures .So it is critical to search out and determine numerous ways that one will adopt to confirm a awfully high level of fault tolerance of the IoT based mostly system. The paper aims at characteristic and classifying the present Fault tolerance mechanisms which will tolerate the IoT systems failure.

Keywords- Internet of things, Fault Tolerance, Sensors, Wireless Sensor Networks, Reliability and Availability.

1. INTRODUCTION

IoT technologies measure following generation of technologies when net technologies. IoT technologies square measure dynamic and take issue from typical networks. several applications square measure being engineered these days mistreatment the IoT primarily based networks, as well as the home automation, defense, and police investigation systems. Each device in associate degree IoT network fails and so must be created fail-free. Failure as such will happen due to breakdown, amiss, or security outflow. Failures in IoT networks will happen at the device level, native network level, controller level, entree level, net level, and remote

storage and computing level. where the failure happens, the IoT network shall become in-operational and serves no purpose. IoT systems should be fault-tolerant. Fault tolerance should be in-built as half in IoT system. Every device in associate degree IoT network fail and so must be created fail free. Failure intrinsically will happen thanks to breakdown, amiss, or security outflow. Failures in IoT will happen at associate degree level of an IoT network where the failure, the IoT shall become in-operational and serves no purpose. In order to deliver sensible services, IoT is the inner and external communication of intelligent parts through the web. Reliable and fault-free facilities ought to be offered by a dependable IoT theme. A fault is a flaw that impacts the correct practicality at intervals the hardware or software package systems [1]. As IoT devices square measure heterogeneous, extremely distributed, powered, and dependent on wireless communication and affected by measurability, it is particularly troublesome to form a pattern for Fault Tolerance in IoT. The IoT devices that square measure distributed [2] in nature might cause the system to suffer from server crashes, server omissions, incorrect responses, and whimsical errors. The reliance on wireless and battery makes the IoT devices hardly redeemable [3]. In Addition, being exposed to new instrumentation and facilities influences the performance of the system.

2. RELATED WORK

Alipour, M., Dupuy-Chessa, et al. [10] Proposed that the advances in information technologies allows disaster management stakeholder to design and develop platforms and tools that can raise endangered people's preparedness level. One primordial element which impacts human behavior and decision making during an incident is perceived emotion. It is

repeatedly argued that such emotions influence the preparedness level and risk perception towards an adaptation in behavior. Thus, improving emergency preparedness systems by capturing real human behavior and emotions is an advantageous consideration. Hang, J., Wang, et al [11] proposed that the Customers expect integral service at the same time higher speed. They showed that the main purpose of using this techno logy is not required for special software installation and not need for any maintenance. This cloud computing is mainly used for data storage e apart from this they provide platform and infrastructure service. El Kafhali et. al.[12] proposed an analytic model for a fog/cloud-based Medical IoT system showing how to reduce the cost of computing resources while guaranteeing performance constraints. They used the QN concept to predict the system response time and estimate the minimum required number of P&S resources to meet the service level agreement. Huang et al.[13] propose a theoretical approach of performance evaluation for IoT services, which provides a mathematical prediction on performance metrics during design before system implementation. The authors formulate an atomic service by a queuing system in order to model IoT systems by a queuing network and obtain performance metrics. Whilst using QN, this paper does not address any modeling based on software architecture to be assessed by performance indices. Whilst few related works have been found on IoT systems modeling with QNs, we did not find any previous work on modeling emergency evacuation systems. Asad Javeda et al.[14] used for implementing a variety of IoT based applications. In the architecture, they have considered the placement of software stacks at different locations for making deployment decisions at run time. They have also considered many other issues such as long-distance network connectivity, faults happening within edge devices, harsh operating environment, etc. In the architecture that included the issue of processing that should take place at both the edges of devices and the cloud. Alexander Power et al. [15], they built a framework using Micro-services. In the framework, they have included the support required for the IoT system to tolerate the faults when they happen through the inclusion of machine learning processes. The machine learns when the faults happen and then take tolerant actions immediately so that the network will fail free. Achene Bounceur1 et al.[16] They have expressed that the leader must be elected dynamically considering the paths that must have failed. They have presented an algorithm for electing a leader through the use of a local minimum as a root and the concept of flooding is used to determine a spanning tree for routing the communication over the spanning tree. The two spanning trees coincide, the better one is selected, and the other ignored. The root of the spanning tree will be the leader through which the communication is affected. Jitender Grover et al. [17] they proposed a cloud-based IoT network architecture. The architecture built with the components required for making the network survive even in the presence of failure of the edge servers. The network recognized as different hierarchies, and the communication is re-directed to different hierarchy when a fault noticed in a different hierarchy. They have included mobile agents on the servers that share the system states, data, and other agents if the system fails at fog, edge, mist, or cloud. Inclusion of these components will help re-direction in the case of any server failure. I, H. et al.[18]They proposed a framework that has become the heart of the modern science. This develops a main difference between the human decisions and smart machines. AI has shown its significance in machine learning and deep learning in modern systems. This can automate the responses and processes of the systems. As per the analysis, companies spend around 70% on AI workers and com-pounded annual will reach $57.6 billion by 2021. The companies that fail to adapt AI and ML are fated to be left. There are some traditional problems in AI which includes reasoning, planning, and learning and d percept ion. Al can generate automated responses for the action given by any de vice or network without any human interaction by using ML and DL.

Regarding IoT architectural styles and patterns, Cavalcante et al [19] introduce two reference architectures for IoT and analyze their characteristics. They realized that, both architectures need to fulfill the essential IoT non-functional requirements such as interoperability, scalability, and security, whilst considering the mandatory requirement of dynamic

adaptation for IoT systems. Khan et al [20] present a cloud-based architecture for context-aware services for IoT and smart cities and walk through it using a hypothetical case study. Their architecture is based on cloud and they argue that cloud computing can provide a suitable computing infrastructure for data storage and processing needs of IoT and smart cities applications. Butzin et al [21] investigates on patterns and best practices that are used in the micro services approach and how they can be used in the internet of things. Azimi et al [22] propose a hierarchical computing architecture for IoT-based health monitoring systems. The model benefits from the features of fog and cloud with an adaptive architecture based on MAPE loop that is discussed into a 3-tier IoT-based system. Lee et al [23] propose a self-adaptive software framework for performing runtime verification using the finite state machine-based model checking. For the run-time verification, a self-adaptation process based on a MAPE loop is implemented. Shekhar et al [24] identify the key challenges that inhibit the universal adoption of cloud, especially in the context of IoT applications. They propose a dynamic data driven cloud and edge system that uses measurement data collected from adaptively incrementing the cloud and edge resources.

3. OVERVIEW OF IOT NETWORK ARCHITECTURE

IoT systems unfolded over an oversized physical space and involve multiple completely different technologies ranging from sensors and actors within the field over network parts accustomed transfer collected knowledge and management signals from one participant to a different victimization numerous protocols like MQTT and CoAP. Knowledge square measure processed and hold on, mixed and mass on their manner from the perimeters of the network to the core.

A. Layers design

Actuator: Actuators rework associate degree electrical signal into a physical amount that correlates, such as motion, force, sound, etc. Fault tolerance is introduced at actuate layers by creating use of multiple devices achieving a common task. In order to notice presence of person devices like CCTV, Bluetooth, WI-Fi, noise detector square measure being employed.

Sensor: A detector is a tool capable of detection modifications in associate degree setting. A detector will live and convert a physical development (such as temperature, pressure, associate degreed thus on) into an electrical signal. Fault tolerance is introduced at the sense layer [4].

Storage and Processing: The output level depends on however typically the parts of process and storage square measure localized forced to the sting. Process is that the execution step for a specific system which will be judged supported the time. Storage is another side which will play a crucial role in effectively storing massive volumes of knowledge. Fault tolerance is achieved in edge computing systems by introducing manual laborer, Apache writer and Kubernetes [5]

Network Capabilities: In IoT the reliable ness of networks is additionally a crucial side to check, network topologies ought to be easy and pliable to the changes. Fault tolerance at network layer, in which, a routing algorithmic program is projected that searches disjoint routes for message exchanges in the system, creating it sturdy to failure

B. SPEC

Distributed: The pattern of the design will be distributed that successively divides the network and the knowledge into completely different sites.

Centralized: A centralized design suggests that a single or a few organizations square measure out there that have management over the entire network. A centralized typically implies one-hop communication for all members of the network, however is often accomplished by a multi-hop network in the context of short-range embedded systems.

Hybrid: This kind of design combines each the techniques i.e. centralized and localized or distributed. This will result in additional improve in the overall performance of the system.

C. SERVICE ARCHITECTURE:

Micro services: In IoT systems, micro services and SOA have the same purpose, that is to produce one or many applications from a group of various services. A micro service may be a light-weight, single-responsibility program that will be severally deployed, scaled and evaluated.

Service familiarized design (SOA): Service familiarized design (SOA) place the service at the core of the planning of their IoT application. The core application element, in reality, makes the service accessible over a network for different IoT elements. A Platform-as-a-service (PaaS) to permits developers to develop IoT based mostly applications victimization API, modules, frameworks etc.

Publish-Subscribe: Publish-Subscribe is a pattern of electronic messaging aimed at decoupling the causation (publisher) and receiving (subscriber) teams. Fault tolerance is handled by victimization Apache Kafka, publish/subscribe vogue for achieving knowledge replication, showing high performance [5]

4. FAULT-TOLERANCE

Within the IOT architectures the most vital issue in IoT analysis is its design. And the main side to think about is the elements used in the completely different layers. IoT elements embrace RFID, WSN, Addressing Share, knowledge Storage and Virtualization. Every the IoT elements should be mutual e.g. WSN. These networks square measure freelance of every different and have their own distinctive design. Network nodes will communicate with every different based mostly on outlined standards. however once the wireless sensing element networks square measure interconnected as half of associate IoT network, they ought to be viewed from a completely different position. Standards, architectures, and protocols can vary. Naturally, these changes can additionally influence ways and Fault tolerance sweetening techniques.

Fault Tolerance feature of a wireless network system is fault prediction and a sensible methodology of that is represented in [6]. This explicit technique brings regarding constant and time period system observation. A 4-layer design style is employed to predict fault by observation conditions within the knowledge transmission with comprehensive and reliable processing. These styles of architectures square measure typically impractical in several systems as well as Real Time systems as a result of the square measure pricey and increase system reaction time. Additionally, if fault happens for any reason despite previous predictions, therefore me measures ought to be taken so that fault is discovered and renovated. The design is strictly centered on the fault prediction section.

5. TOLERANCE TECHNIQUES

Replication: Replication is primarily used in the distributed systems analysis field to give fault tolerance. In active replication every shopper request is processed by all the servers. In passive replication there is solely one server that processes shopper requests. Fault tolerance is achieved by dividing finish nodes into completely different teams, among every cluster all different nodes act as a backup node for every node [7]

Network Control: The IoT network is commonly split into separate clusters among the network management theme. a selected cluster head (CH) makes roll decision requests to the different nodes often and the failure can be verified if it will not receive a response message. The CH itself will but, establishes a single purpose of failure.

Distributed Recovery Block: during this method, one program is dead at the same time on a tray of nodes, one of that is active and the different is inactive. the most active node performs the task during a no-fault scenario and the different node performs the same task in the shadow. Finally all results can be checked and the results related to the most node are transmitted because the output if the check is passed properly. The shadow node becomes active and generates the outputs if the primary node check fails. If the first node check fails, the shadow node becomes active and produces the outputs [8].

Time Redundancy: At all instruction and task stages, time replication could be done. The code is duplicated at the instruction level and also the results square measure afterward compared to discover attainable error. A program is run double or a lot of at the task level to minimize complicated faults. Whereas this technique will not introduce extra hardware prices, it will increase the time taken to confirm redundancy. The tactic reduces the potency of computation and therefore absorbs a lot of resources.

6. QUALITY OF IOT SERVICE

Performance: How a system goes to perform in several situations by considering some set of measures like time constraints, surroundings, etc. Distributed edge procedure

network introduces preprocessing at finish nodes, a lot of the computation is handled at the tip nodes itself, thence reducing network traffic in between finish nodes and also the central server. This additionally helped in reducing network utilization and decreasing network latency.

Availableness & Security: availableness is that the ability of the system, to be fully or partly operating whenever needed. Fault Tolerance and availableness square measure not equivalent, as a fault-tolerant system is predicted to keep the system running while not interruption, but service interruptions will occur in a extremely accessible system. A fault-tolerant theme, however, ought to additionally preserve a high degree of device availableness and performance. Security may be a major concern in IoT systems that link numerous elements and entities through a network to every different.

Scalability: As IoT systems ought to be ready to work properly considering an outsized variety of heterogeneous devices, quantifiability is associate vital attribute. It is troublesome to comment on IoT quantifiability as a whole system, however it depends on however to include new resources on demand.

Interoperability: Interoperability permits heterogeneous IoT elements to figure expeditiously along. The paper [4] performs a comprehensive survey on the progressive solutions for facilitating ability between completely different IoT platforms. Also, the key challenges during this topic square measure given.

Energy Consumption: Most IoT devices square measure powered, and it's vital to own energy potency connected to several different quality attributes, such as performance.

7. SECURITY IN THE IOT

Due to the growing size, raising complexity and implementation in critical infrastructures of IoT systems the security aspect is gaining more and more importance. Vashi et al. [9] introduce security challenges of IoT on three different layers (perception layer, network layer and application layer) and describe countermeasures. The layers correspond to IoT devices at the perception layer, IoT gate-ways and execution sites at the network layer, and the application itself at the application layer. Each layer faces different security problems starting from physical attacks at the perception layer over Denial-of-Service (DoS) attacks at the network layer to malicious code injection at the application layer. According to the authors the countermeasures to those problems comprise encryption, authentication, authorization, confidentiality, certification and access control.

8. CONCLUSION

The Internet of Things (IoT) provides an infrastructure for a vast network of real-world “things” to be interconnected in order to achieve greater value and services in domains such as logistics, healthcare, and agriculture etc. Most IoT systems operate in dynamic contexts, where new services, devices, and features may be added, removed and changed over time. The energetic and developing environment of IoT systems makes it complicated to identify sufficient error detection and improvement mechanisms a priori. Also IoT networks are fragile that connect numerous devices using many protocols and interfaces. The reliability of IoT networking is quite dependent on the kind of topology and networking. An imperative challenge to understand IoT is how to provide a responsible infrastructure for billions of devices and distribute their intended services not including failing in unexpected and disastrous ways. In this paper, we present a systematic learning of mapping with the objective of classifying and defining the state-of-the-art methods and techniques as well as security concern for Fault Tolerance in IoT. The results of this study are both research and industry oriented and are intended to make a framework for future research in Fault Tolerance IoT related fields. In future work, we will expand our taxonomy of error-detection NFAs to establish a generic framework for the easy and adaptive Fault Tolerance implementation in IoT systems.

REFERENCES

[1] N. Mohamed, J. Al-Jaroodi and I. Jawhar, "Towards Fault Tolerant Fog Computing for IoT-Based Smart City Applications," 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, pp. 0752-0757, 2019.

[2]Tusher Chakraborty, Akshay Uttama Nambi, Ranveer Chandra, Rahul Sharma, Manohar Swaminathan, Zerina Kapetanovic, and Jonathan Appavoo. 2018. Fall-curve: A novel primitive for IoT Fault Detection and Isolation. In Proceedings of the 16th ACM Conference on Embedded

Networked Sensor Systems (SenSys '18). Association for Computing Machinery, New York, NY, USA, 95–107.

[3] "What Is Fault Tolerance?: Creating a Fault Tolerant System: Imperva." Learning Center, Imperva, 30 Dec. 2019, www.imperva.com/learn/availability/fault-tolerance/.

[4] N. Mohamed, J. Al-Jaroodi and I. Jawhar, "Towards Fault Tolerant Fog Computing for IoT-Based Smart City Applications," 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, pp. 0752-0757, 2019.

[5] A. Javed, K. Heljanko, A. Buda and K. Främling, "CEFIoT: A fault-tolerant IoT architecture for edge and cloud," 2018 IEEE 4th World Forum on Internet of Things (WF-IoT), Singapore, pp. 813-818, 2018.

[6] Xu, X., Chen, T. and Minami, M., "Intelligent fault prediction system based on internet of things", Computers and Mathematics with Applications, Elsevier 64, PP.833-839, 2012.

[7] M. Mudassar, Y. Zhai, L. Liao and J. Shen, "A Decentralized Latency-Aware Task Allocation and Group Formation Approach With Fault Tolerance for IoT Applications," in IEEE Access, vol. 8, pp. 4912-4923,2020.

[8] Tusher Chakraborty, Akshay Uttama Nambi, Ranveer Chandra, Rahul Sharma, Manohar Swaminathan, Zerina Kapetanovic, and Jonathan Appavoo. 2018. Fall-curve: A novel primitive for IoT Fault Detection and Isolation. In Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems (SenSys '18). Association for Computing Machinery, New York, NY, USA, 95–107

[9] S. Vashi, J. Ram, J. Modi, S. Verma, and C. Prakash. 2017. Internet of Things (IoT): A vision, architectural elements, and security issues. In 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC).492–496.https://doi.org/10.1109/I-SMAC.2017.8058399

[10] Alipour, M., Dupuy-Chessa, S., and Jongmans, E-.Disaster mitigation using interface adaptation to emotions: a targeted literature review. In 10th International Conference on the Internet of Things Companion, pp. 1-15, 2020.

[11].Zhang, J., Wang, B., He, D., Wang, X.A.: Improved secure fuzzy auditing protocol for cloud data storage. Soft.Comput. 23(10), 3411– 3422 (2019)

[12].El Kafhali, S. and Salah, K. (2018). "Performance Modeling and Analysis of IoT-enabled Healthcare Monitoring Systems". In: IET Networks.

[13].Huang, J., Li, S., Chen, Y., and Chen, J. (2018). "Performance modelling and analysis for IoT services". International Journal of Web and Grid Services 14, p. 146.

[14]. AsadJ aved, Keijo Heljanko, Andrea Buda, and Kary Främling, (2018). A Fault-Tolerant IoT Architecture for Edge and Cloud. IEEE, 978-1-4673-9944-9/18, DOI:10.1109/wf-IoT.2018.8355149, pp: 813-818.

[15] Alexander Power and Gerald Kotonya (2018). A Micro services Architecture for Reactive and Proactive Fault Tolerance in IoT Systems. IEEE, 978-1-5386-4725-7/18/$31.00,D OI:10.1109/wowmom. 2018.8449789, pp: 1-6

[16]Ahcene Bounceur, MadaniBezoui, Massinissa Lounis, Reinhardt Euler, Ciprian Teodorov, (2018). A New Dominating Tree Routing Algorithm for Efficient Leader Election in IoT Networks. IEEE 978-1-5386-4790-5/18, DOI:10.1109/ccnc.2018.8319292, pp:1-2

[17] Jitender Grover and Rama Murthy Garimella, (2018). Reliable and Fault-Tolerant IoT-Edge Architecture. DOI: 10.1109/ICSENS.2018.8589624, pp:1-4

[18]. Li, H., Ota, K., Dong, M.: Learning IoT in edge: deep learning for the Internet of Things with edge computing. IEEE Netw. 32 (1), 96–101 (2018)

[19] Everton Cavalcante, Marcelo Pitanga Alves, Thais Batista, Flavia Coimbra Delicato, and Paulo F Pires. 2015. An analysis of reference architectures for the internet of things. In Proceedings of the 1st International Workshop on Exploring Component-based Techniques for Constructing Reference Architectures. ACM, 13–16.

[20] Zaheer Khan and Saad Liaquat Kiani. 2012. A cloud-based architecture for citizen services in smart cities. In Utility and Cloud Computing (UCC), 2012 IEEE Fifth International Conference on. IEEE, 315–320.

[21] Björn Butzin, Frank Golatowski, and Dirk Timmermann. 2016. Microservices approach for the internet of things. In

Emerging Technologies and Factory Automation (ETFA), 2016 IEEE 21st International Conference on. IEEE, 1–6.

[22] Iman Azimi, Arman Anzanpour, Amir M Rahmani, Tapio Pahikkala, Marco Levorato, Pasi Liljeberg, and Nikil Dutt. 2017. HiCH: Hierarchical Fog-Assisted Computing Architecture for Healthcare IoT. ACM Transactions on Embedded Computing Systems (TECS) 16, 5s (2017), 174.

[23] Euijong Lee, Young-Gab Kim, Young-Duk Seo, Kwangsoo Seol, and Doo-Kwon Baik. 2018. RINGA: Design and verification of finite state machine for selfadaptive software at runtime. Information and Software Technology 93 (2018), 200–222.

[24] Shashank Shekhar and Aniruddha Gokhale. 2017. Dynamic resource management across cloud-edge resources for performance-sensitive applications. In Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing. IEEE Press, 707–710.