EasyChair Preprint

№ 2134

Resilient IoT Infrastructures

Aicha Garci

December 10, 2019

Resilient IoT Infrastructures

Aicha Garci
Passau University
Germany
aicha.garci01@gmail.com

ABSTRACT

The Internet of Things (IoT) has become one of the world's most prominent technologies. In many essential fields of life, it provides great solutions for humanity. IoT refers to a set of sensors or objects in a certain environment with the ability to communicate without human intervention through the internet. Some of the applications in those fields are critical and don't tolerate faults which means that in certain cases the service have to be continuously delivered despite the system failure. This paper examines the concept of resilience in the context of IoT. Indeed, in order to understand the issues related to the Internet of Things, we present a state of the art of IoT resilience mechanisms. This paper describes techniques and methods used to enhance the resilience of IoT infrastructure in each layer. Finally, this state of the art allows us to identify the prerequisites and the insufficiencies of these solutions and to begin to analyze the potential improvements including the proposal of an architecture that implements the resilience mechanisms in 4 levels of the five-level IoT architecture and assures an overall resilient system.

KEYWORDS

Internet Of Things, Resilience, IoT Architecture, Fault-tolerance, Fault-masking, Fault-prevention, Fault-detection, dependability

1 INTRODUCTION

Industry 4.0 is a term that refers to the fourth world industrial revolution. Cloud, Big data, Blockchain, Artificial intelligence and IoT are part of the technologies that led to the emergence of the 4.0 industry.

The IoT being one of the components of the industry 4.0 has invaded today the daily life of the users and become more and more important and in some way indispensable.

Indeed, today we use many communicating objects in our everyday life. We are surrounded by these devices that can communicate with their environment and exchange data, which offer us more and more services facilitating our activities, and with which we interact frequently.

One of the most popular applications for a wide audience is certainly the smart home or "intelligent building", with its many communicating objects that will offer new home automation services, for better control of equipment and optimal use of the energy.

Major challenges for IoT, as mentioned in [1], is to be able to manage **technological heterogeneity** across **multiple administrative domains** and object standards coupled with a multitude of application needs and uses in terms of security services. Knowing that these needs can **evolve over time** depending on the context and preferences. Another challenge is the presence of objects with **constrained resources** like energy, that would cause the disruption of the service and threatens resilience.

IoT is used in fields where faults are not accepted, like the medical field. Indeed, doctors relay more and more on the connected health accessories that promote home hospitalization and reduce the risk of medical error. In this case availability is a fundamental need that should be guaranteed in the IoT architecture.

Previously we talk rather about dependability which is, according to [2], the ability of a system to avoid failures in critical services taking into consideration the fact that the system can fail. This definition is suitable for the IoT applications where faults may lead to system failure. However resilience is, in addition to the effect of dependability, the fact of delivering continuously the expected service in spite of the failure of the system, by changing the configuration of the system or decreasing the amount of resources used to deliver the service.

Resilience mechanisms are approaches and techniques that enable facing those challenges and adapting the configuration of the system to its current state and ensure that the service is continuously delivered no matter what happens; in addition resilience fulfill other dependability goals through delivering a reliable and trustworthy service as well as scanning the vulnerabilities and performing continuous security event monitoring. Many resilience mechanisms were designed and proposed in the literature and organised into categories such as fault prevention, elimination of faults, Prediction of faults and Fault Tolerance as discussed in [3].

This paper is motivated by the future requirements of IoT architectures to ensure resilience. In this paper, we present some methods and techniques that have already been implemented or suggested by the literature in order to improve the resilience of the IoT infrastructure in different fields such as Smart cities [4], 5G-IoT [5], Wireless Networks [6], etc.

The paper is organized as follows. In Section 2 we introduce the generic meaning of resilience in information technology as well as service degradation to ensure a continuous delivery of the service with lower performance or quality. In Section 3, we present the evolution of the IoT architecture and its different parts and components. In Section 4 we review several available remarkable methods to improve the resilience in the IoT architecture from the literature and we classify them according to certain measures based on the architectures illustrated in Section 3, then we propose a new IoT architecture for the smart home scenario that meets the key resilience requirements of IoT systems. Finally, we conclude this paper in Section 5.

2 RESILIENCE IN INFORMATION TECHNOLOGY

In this section, we define resilience in the context of information technology, its aspects, its principles and how to achieve it.

2.1 Generic Meaning

We can define the resilience as the ability of a system to continue to operate even in the case of a failure, intentional or unintentional incident, and/or extreme solicitation [7].

In information technology the resilience is the ability of the system to deliver the expected service continuously even when the regular procedure failed or was interrupted. In this case, the meaning of resilience imply also the ability to restore the regular procedure as well as the ability to change the procedure in case of risk and adapt it to the current situation of the system.

Authors Björck, Henkel, Stirna and Zdravkovic state that "The notion of continuously, means that the ability to deliver the intended outcome should be working even when regular delivery mechanisms have failed, during a crisis and after a security breach. The notion also denotes the ability to restore the regular delivery mechanisms after such events as well as the ability to continuously change or modify these delivery mechanisms if needed in the face of changing risks" [7].

2.2 Aspects of Resilience

There are five aspects of resilience, as discussed in [7]:

(1) Objective: The ability of the system to deliver the expected outcome (or "business value" as mentioned in the paper) even in presence of faults.
(2) Intention: create systems in such a way that they have the property of being fail-safe. It means that during the design of the system, the kind of failure that could happen should be taken into consideration and methods to face those failures should be predefined.
(3) Approach: The security should be part of the system and not added after the design of system.
(4) Architecture: It would be better to design an architecture with multiple security levels in order to allow for partial failure so that the system would always be able to deliver the service.
(5) Scope: consider the system as an interconnected environment because the network is a source of resilience (multipath for example).

Those aspects show how resilience can be approached.

2.3 Basic Resilience Mechanisms

Means to attain dependability were presented in the literature [8] and the aspects of resilience to offer a resilient middle-ware for the IoT architecture were implemented [3]. We reformulated both definitions to get a new definition for the four main mechanisms to ensure resilience:

(1) Fault prevention means to prevent the appearance, occurrence or introduction of faults in the system. It is usually achieved through redundancy.
(2) Fault tolerance means to avoid service failures in the presence of faults. It tries to hide the occurrence of faults and to continue to provide the requested service despite their occurrence. Handled via the deploying of the basic functionalities of the system in redundant components.
(3) Fault removal [8] or fault elimination [3] means to reduce the presence (the number and severity) of faults. This method operates both during development (verification of conditions, regression test, injection of faults, etc.) or during use (maintenance).
(4) Fault forecasting [3] or prediction of faults [8] means to estimate the present number, the future incidence, and the likely consequences of faults. It seeks to estimate (qualitatively and quantitatively) the occurrence and consequences of faults. It is realized by the modeling and evaluation of systems.

In some researches, like in [3] we find new mechanisms such as Failure management which aims to reduce the duration of the failures that cannot be avoided, or that was not anticipated. A resilient system should guarantee the delivery of the expected service as long as possible, even after failure.

2.4 Resilience and degradation of service in IoT

2.4.1 Resilience in IoT. Resilience of the system defined in Figure 1 includes the capability 1) to resist external perturbing events and internal failures 2) to recover smoothly and re-enter a stable state 3) to adapt its structure and behavior to constant changes [9]. There are also concepts [10], where a service is allowed to "degrade" e.g. deliver only a part of its functionality or displays decreased performance to to prevent the denial-of-service.

2.4.2 Degradation of service. means, according to [11], that instead of failing, the quality of the service degrades to a lower one. There are two approaches here: 1) Designing a variant of the service which is easier to compute and deliver to the user; or 2) Delivering only the important features of the service and dropping the unimportant traffic. In [12] it is called "degraded service mode", and it means that "Critical applications such as healthcare and emergency response must continue to operate meaningfully (at least in a degraded service mode) despite cloud and connectivity disruptions". Indeed, performance can range from 0% to 100%, where 100% means no degradation in service and 0% means no service is available, so if we design mechanisms that allow to decrease the performance and the amount of resources needed for the delivery of the service in case of faults occurrence or system failure but ensure the delivery of the critical features of the service, we can ensure that the service is continuously delivered to the end-user.

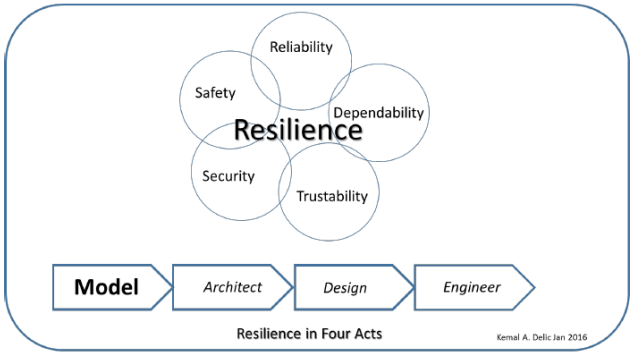

Figure 1: Resilience IoT Defined [9].

3 IOT ARCHITECTURES

In this section, the focus will be on the evolution of the IoT architecture, and how it has in one way or another contributed to the improvement of resilience. We chose to analyse three IoT architectures: three-, four- and five-level architecture.

3.1 Three-Layer Architecture

According to many researchers [13–15], the IoT primarily operates on three layers which are the Perception, Network, and the Application layer. Figure 2 shows the basic three layer architecture of IoT and demonstrates the technologies and components in each layer.

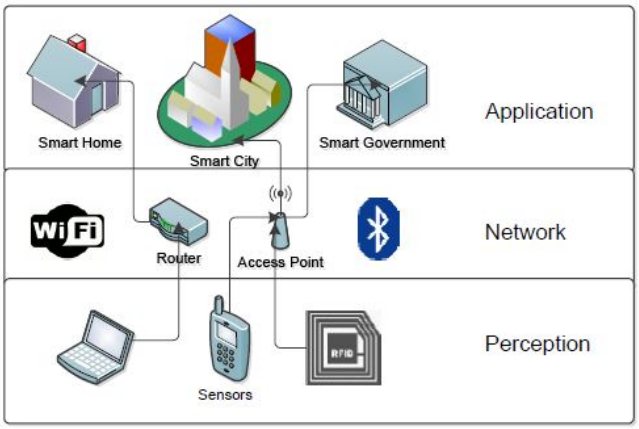

Figure 2: Three-Level Architecture [16].

3.1.1 Perception Layer. The perception layer or "Sensors" layer in IoT. The aim of this layer is to use sensors to collect data from the environment. This layer acquires, captures, treats, and transmits data from sensors to the communication layer.

3.1.2 Network Layer. This layer serves the role of data routing and transmission over the Internet to various IoT hubs and components. On this layer, Internet gateways, switching and routing components etc. operate to provide heterogeneous network services by using some of the latest technologies such as WiFi, LTE, Bluetooth, 3 G, Zigbee, etc. The network gateways act as a mediator between different IoT nodes by aggregating, filtering and transmitting data from and to various sensors.

3.1.3 Application Layer. The application layer guarantees that the data is accurate, complete and confidential. The aim of IoT, which is to build intelligent environments, is achieved in this layer. It includes protocols and interfaces used by devices to identify and communicate with each other.

3.2 Four-Layer Architecture

In this architecture one additional layer was added, which is **the support layer** between the perception layer and the network layer [17] as shown in figure 3, in order to separate the intelligent operations and the data processing from the management of the applications implemented in the IoT infrastructure (which is handled by the application layer).

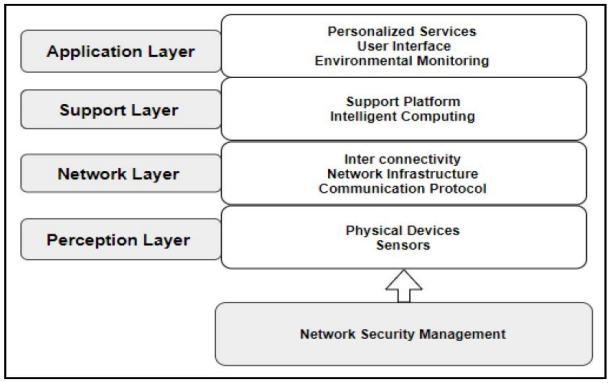

Figure 3: Four-Level Architecture [17].

3.3 Five-Layer Architecture

Figure 4 shows an improved IoT architecture with one more additional layer that is the processing layer, called by some researchers "Aggregation layer" as in [18]. It coordinates information processing, and converts the data into a standard format. Large data sets are analyzed, stored and processed. It can use servers, cloud computing and tools for large data processing.

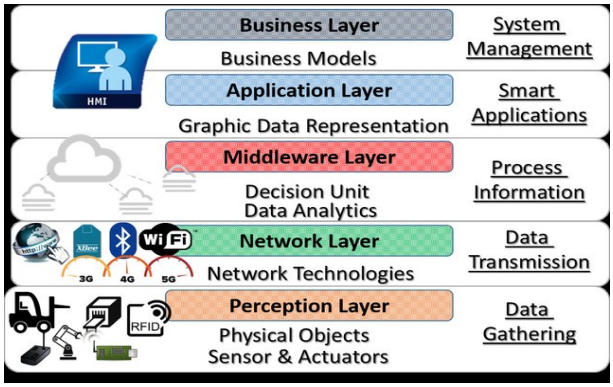

Figure 4: Five-Level Architecture [19].

4 SOLUTIONS ENABLING RESILIENCE - RELATED WORK

In this section we define criterias according to which we group the approaches of making IoT application more resilient as well as the category of each mechanism.

4.1 Specification of the Grouping Criteria

We chose to group the resilience mechanism in the IoT architecture in terms of the requirements in each layer.

4.1.1 Perception Layer (or sensing layer). The routing protocol is a key element to ensure resilience, because it allows for each object (sensor or actuator) to decide how to attach another object. Moreover, the placement of the object is as well important in order to achieve fault-tolerance.

4.1.2 Transport Layer. In this level we rather talk about the communication between the planes of the architecture (for example data plane, control plane, ...etc). There should be many possible paths between the different planes in order to achieve the availability of the service.

4.1.3 Processing Layer. This layer manages the interactions between restricted devices and cloud services that provide user applications with analytics, data storage, and support. If the communication between those two parts of the IoT architecture is lost then it will not be possible to deliver the service anymore. That's why it is important to check the mechanisms that may guarantee the resilience at this level.

4.1.4 Application Layer. In this layer the resilience concerns the Virtual Networks of the Cloud Infrastructure, the survivability of the Virtual Links with respect to the Substrate Network as mentioned in [20] as well as virtual machines with the services running on them.

We can also work on a categorization of the mechanisms using categories, indicated in [8], such as "compensation", "recovery", "diagnosis", "fault prevention" and specific security measures. In the following we explain the meaning of each category.

(1) Compensation: this is a fault masking mechanism that consists on detecting the error and handling the fault so that it wouldn't be visible to the end-user.
(2) Recovery: called also self-repair or self-healing and it means that the configuration of the system (routing paths, defected components) changes dynamically in case of failure in order to continuously deliver the service.
(3) Fault diagnosis and removal: it consists on diagnosing the fault(s) that caused the system failure, it means figuring out the kind of faults that occured, and correcting them.
(4) Fault prevention: means to predict the errors and faults that may occur in the system and design mechanisms to face them.

4.2 Related work

Table 1 shows some resilience mechanisms in the literature grouped by architecture layer for the reasons presented in the section 4.1. Even though we chose to work on the five-layer IoT architecture, we will only treat four layers in terms of resilience. Although, the technologies evolve continuously, the business logic is usually reused as it is. That's why it would be better not to implement resilience mechanisms in the business layer.

The Perception layer Group [21–24] has proposed improvements directly in the physical infrastructure which enables communication between smart objects. In general, these works suggested methods for topology control to provide fault tolerance through the placement of a smart device, or by using the communication infrastructure to provide alternate and simultaneous routing routes.

A set of mechanisms focusing on the communication process between the perception layer and the upper layers, to allow continuous exchange of messages and consequently continuous delivery of the service are proposed in [25–28]. In this context, the functionalities of the Software Defined Networking (SDN) are exploited in order to enhance the resilience in the communication layer and the whole IoT architecture.

A middleware is so important in the IoT architecture. Indeed, it acts as a bond (adaptation layer) joining the heterogeneous domains of applications communicating over heterogeneous interfaces [29] and provides consequently abstraction to applications from the things. In order to make the processing layer, or the middleware layer, more resilient some researchers like in [3, 12] have designed a middleware that integrate resilience mechanisms (fault-tolerance, fault avoidance and fault management).

The application layer group of works [20, 30] focused on the exploitation of the capabilities of the Substrate Network in order to achieve the survivability of the Virtual Links. However [31] focused more about the multi-domain aspect and the congestion phenomenon.

The works discussed focused on a single layer of the IoT architecture which leaves the other layers vulnerable. More general solutions were also proposed in the form of IoT architectures to try to handle the IoT complexity.

In the next section we present a smart home scenario and we propose a resilient four layer architecture for this scenario using the mechanisms described in this section.

4.3 Smart Home scenario

The smart home is a promise of comfort and security. Opening a lock with a smartphone or remotely viewing what happens in his house seemed utopian a short time ago. Nowadays the smart home has not only become a reality, but has also gained a growing share of the population since its first appearance. In a Smart Home, data is collected from sensors deployed in random positions most of the time. In order to process the data and make smart decisions, given the resource constraints of the Smart Objects, the gathered data has to be sent to the Cloud, where particular services perform an analysis of the activities in the home. In this context we raise the question of home resiliency when a component (from perception-, transport-, processing- or application-layer) becomes unavailable.The smart home is showed by figure 5.

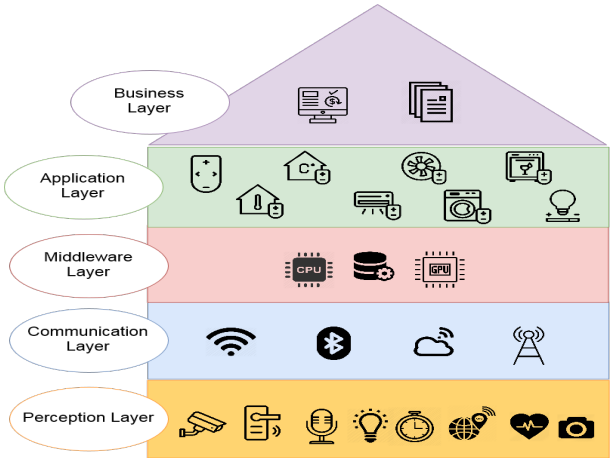

Figure 5: Smart Home.

Table 1: Mechanisms to improve Resilience in IoT Architecture

Category	Mechanism	Description
	Perception Layer	
Fault prevention, Recovery	RPL as the routing protocol for low power and lossy networks (LLNs) [21]	Review of many implementations of RPL-based routing protocols, which is considered as the best routing protocol for IoT. RPL is considered as a resilient mechanism since it ensures optimal communication between every day's objects by taking into account constraints such as low power and unstable communications. It offers efficient topology repair, and and a multi-path routing approach where nodes use multiple parents and transmit their data across all the available links, so that if one parent node is defected another one is used and the service is not disrupted.
Fault prevention	An optimized two-phase approach with the objective of maximizing network connectivity [22]	An optimized two-phase approach[1] in order to maximize the network connectivity, which is necessary to achieve network resilience.
Fault prevention	The coexistence of the two structures: RPL routing and IEEE 802.15.4 MAC [23]	Modification of the cluster-tree operation of IEEE 802.15.4 to support RPL DODAG to offer the possibility for the traffic to be equally distributed between all the possible parent nodes and not only the best one. That improves the network lifetime and avoid quick energy depletion which ensures the resilience of the architecture.
Fault prevention	Two approximation algorithms to achieve diverse levels of fault-tolerance [24]	Implementation of two approximation algorithms to optimize the Relay node placement in order to achieve fault-tolerance and resilience in Heterogeneous Wireless Sensor Networks.
	Transport Layer	
Fault prevention	FatTire (Fault Tolerating Regular Expressions) [25]	Implementation of a new programming language that specifies the routing path while ensuring resilience thanks to regular expressions.
Fault prevention	Five Nines of Southbound Reliability in Software-Defined Networks[26]	Exploit the SDN approach by designing reliable south- bound interfaces between nodes and controllers. The developed algorithm analyzes the existing network topologies in order to ensure resilience and indicates the number of controllers that should be used in the architecture, where to place them and What nodes must be connected to each controller.
Fault prevention	Plinko [27]	Building large Forwarding Tables and routing algorithm to ensure resiliency against link failure.
Fault prevention	Algorithms to improve the connection between control and forwarding planes in SDN [28]	An algorithm is proposed to improve the resilience of the connection between control and forwarding planes in SDN. The algorithm specifies where to place the controllers in the topology.
Fault Masking	Redundancy [32]	The study showed that redundancy can mask hardware, software, and network component failures by preventing it from turning visible to the end-users.
	Processing/Middleware Layer	
Fault prevention, Recovery, Compensation, Fault diagnosis and removal	CHARIOT: Goal-driven Orchestration Middleware for Resilient IoT Systems [3]	Failure avoidance is mainly achieved via redundancy or replication mechanisms. CHARIOT-ML supports functionality replication using four different redundancy patterns.

Continued on next page

[1]The first phase utilizes some geometrical structures (namely MST, DT, and Steiner tree) to construct a backbone of RNs that connect all WSN sectors and finds a finite set of candidate locations for more RNs to be deployed in the second phase. The second phase deploys the remaining RNs in some of the candidate locations with the objective of maximizing connectivity of the network; this is carried out by solving a relaxed SDP

Table 1 – *Continued from previous page*

Category	Mechanism	Description
		CHARIOT achieves failure management by minimizing downtime due to failures that cannot be avoided thanks to the sense-plan-act loop. In other terms, the system learns from the previous failures that happened and try to find a configuration that lessen it in the future.
Fault prevention, Recovery, Compensation, Fault diagnosis and removal	Ride: A Resilient IoT Data Exchange Middleware Leveraging SDN and Edge Cloud Resources [12]	Ride Data Collection (Ride-C): configures resilient IoT publisher-to-data exchange event collection flows. It tracks and adapts to local or cloud failures and determines whether further processing should occur at the cloud or edge.
		Ride Data Dissemination (Ride-D): uses an unmodified cloud data exchange when possible or resilience-enhanced edge alerting during periods of cloud connection instability [12].
Fault prevention, Recovery, Compensation, Fault diagnosis and removal	SORRIR: A Resilient Self-organizing Middleware for IoT Applications [33]	A middleware that faces the challenges of the IoT applications throughout its lifecycle. It takes into consideration the size of the IoT system, critical aspects, Latency-critical applications and the heterogeneity of the IoT lanscape components.
Application Layer		
Fault prevention	SiMPLE (Survivability in Muti-Path Link Embedding) [20]	Presenting an approach that consider node failure as a set of multiple adjacent link failures and exploit the capabilities of the Substrate Network in order to achieve the survivability of the Virtual Links using less backup bandwidth.
Fault prevention	Virtual network embedding strategies [34]	This paper compares the virtual network strategies in terms of rejection rate. Indeed, minimising the reject rate of virtual network requests increase the availability of the service and consequently improves the resilience of the service.
Fault prevention	SVNE: Survivable Virtual Network Embedding Algorithms for Network Virtualization [30]	Provides solutions for the SVNE problem which improve the resilience of the Virtual Links and minimize the impact of the failures. The solution is based on linear programming modules.
Recovery, Compensation, Fault diagnosis and removal	Reconciling the Overlay and Underlay Tussle [31]	Present a new model for the communication in multi-domain networks in order to achieve stability. It ensures resilience via congestion avoidance and even adaptation with the congestion situation by returning to a stable state.
Fault diagnosis and removal	IoT Application for Fault Diagnosis and Prediction in Elevators [35]	The fault diagnosis can be achieved with a software tool in the application level, this software is able to determine and show the fault to the user.

4.4 Proposed Architecture

To satisfy the requirements of the scenario described in Section 4.3 while guaranteeing a high resilience level in all the layers of the IoT architecture (Five-Layer Architecture presented in section 3.3) we propose the new IoT architecture described on Figure 6. The proposed architecture has five layers, but the resilience mechanisms will be implemented only in four layers (perception, communication, middleware and application layer) because, as mentioned in section 4, the technologies evolve continuously but the business logic is usually reused as it is. That's why it would be better not to implement resilience mechanisms in the business layer that represents the business model and data that's been received from the application layer [36]. Furthermore the possibility of having more than one instance per layer (redundancy) is a key feature of the architecture. In the remaining of this section, we will discuss in detail each layer's components and their interactions; in addition, we will highlight possible mechanisms that could be applied to enhance each layer's resilience.

4.4.1 Perception layer. The architecture's lower layer deals with the physical devices in the smart house. These devices, are smart objects that allow data collection and reaction to specific situations. The problem in this level is that the devices are limited from the performance point of view. Connected devices are often distributed in space and their environment context is dynamic and composite [37]. As shown by [21], RPL is the best routing protocol for IoT. It ensures optimal communication between the smart objects considering constraints such as low power and unstable communications of those devices. The coexistence of the two structures : RPL routing and IEEE 802.15.4 MAC [23] would be a good solution to equally distribute the traffic between the different parent nodes.

4.4.2 Communication layer. Composed essentially by "Gateways" which are defined in the literature [38] as an important component bridging sensing domain and network domain. Plinko [27] is a good solution to ensure resiliency against link failure. In addition software-defined networking (SDN) has many capabilities that can be exploited in order to ensure resilience [28]. Replication [32] is always a good way to avoid the disruption of service through the duplication of resources.

4.4.3 Middleware layer. This layer provides seamless integration of IoT-built devices and data. According to [39] it includes common functionalities and abstraction mechanisms that surround developers and users with the information of the IoT infrastructure to promote communication between these actors. For this purpose CHARIOT [3] was designed, it is a middleware able to fulfill those requirements and ensure resilience in this level using different redundancy patterns.

4.4.4 Application layer. The management of applications and services that support the Smart Home is achieved in the application layer. In our architecture we suggested a congestion avoidance unit that implements a new communication model that deals with multi-domain networks [31], and a virtual network manager that minimizes the reject rate of virtual network requests which increases the availability of the service and consequently improves the resilience of the system [34].

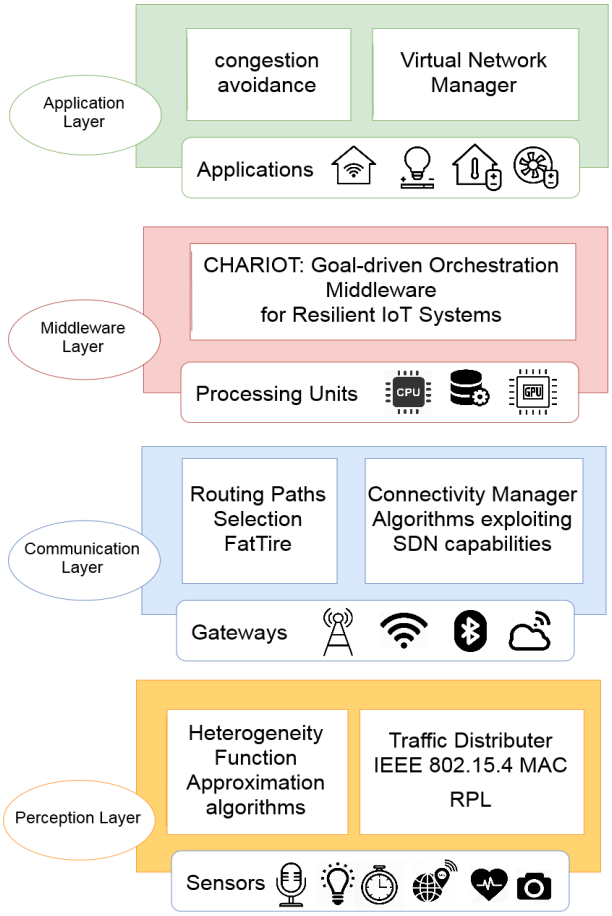

Figure 6: Proposed Resilient IoT Architecture for smart home

5 CONCLUSION

Individuals and organizations are increasingly deploying and using IoT applications in several fields, and some of those fields, such as health applications, mobility and energy, do not tolerate faults and system failures. In this paper, a survey about the resilience techniques and approaches was made. In addition, the different IoT architectures were presented and a new five-layers IoT architecture that implements the resilience mechanisms in every layer was proposed. It implements the resilience mechanisms in all the IoT layers which ensures resilient infrastructure, communications, data processing and applications. The proposed architecture takes into consideration the smart home scenario and meets the key resilience requirements of IoT systems. This topic requires further research particularly as more devices and services are continuously integrated into the IoT systems.

REFERENCES

[1] Christos Tsigkanos, Stefan Nastic, and Schahram Dustdar. Towards resilient internet of things: Vision, challenges, and research roadmap. pages 1754–1764, 07 2019.

[2] A. Avizienis, J. . Laprie, B. Randell, and C. Landwehr. Basic concepts and taxonomy of dependable and secure computing. *IEEE Transactions on Dependable and Secure*

Computing, 1(1):11–33, Jan 2004.

[3] Subhav Pradhan, Abhishek Dubey, Shweta Khare, Saideep Nannapaneni, Aniruddha Gokhale, Sankaran Mahadevan, Douglas Schmidt, and Martin Lehofer. Chariot: Goal-driven orchestration middleware for resilient iot systems. *ACM Transactions on Cyber-Physical Systems*, 2:1–37, 06 2018.

[4] David Perez Abreu, Karima Velasquez, Marilia Curado, and Edmundo Monteiro. A resilient internet of things architecture for smart cities. *Annals of Telecommunications*, 72, 06 2016.

[5] Hamed Rahimi, Ali Zibaeenejad, and Ali Safavi. A novel iot architecture based on 5g-iot and next generation technologies. pages 81–88, 11 2018.

[6] José Manuel Lozano Domínguez, Tomás Mateo Sanguino, and Manuel González. *Evaluation of a Robust Fault-Tolerant Mechanism for Resilient IoT Infrastructures: 9th International EAI Conference, Broadnets 2018, Faro, Portugal, September 19–20, 2018, Proceedings*, pages 3–12. 01 2019.

[7] Fredrik Björck, Martin Henkel, Janis Stirna, and Jelena Zdravkovic. Cyber resilience – fundamentals for a definition. *Advances in Intelligent Systems and Computing*, 353:311–316, 01 2015.

[8] Algirdas Avizienis, Jean-Claude Laprie, Brian Randell, and Carl Landwehr. Basic concepts and taxonomy of dependable and secure computing. *IEEE Trans. Dependable Secur. Comput.*, 1(1):11–33, January 2004.

[9] Kemal Delic. On resilience of iot systems. *Ubiquity*, 2016:1–7, 02 2016.

[10] Valentin Zieglmeier. Resilience metrics. *Chair for Network Architectures and Services, Department of Computer Science, Technische Universität München*, 2016.

[11] K. Benson. Enabling resilience in the internet of things. In *2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, pages 230–232, March 2015.

[12] K. E. Benson, G. Wang, N. Venkatasubramanian, and Y. Kim. Ride: A resilient iot data exchange middleware leveraging sdn and edge cloud resources. In *2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pages 72–83, April 2018.

[13] Kai Zhao and Lina Ge. A survey on the internet of things security. In *CIS*, pages 663–667, New York, NY, USA, 2013. IEEE Computer Society. 612130.

[14] Luigi Atzori, Antonio Iera, Giacomo Morabito, and Michele Nitti. The social internet of things (siot) - when social networks meet the internet of things: Concept, architecture and network characterization. *Computer Networks*, 56(16):3594–3608, 2012.

[15] Marco Leo, Federica Battisti, Marco Carli, and Alessandro Neri. A federated architecture approach for internet of things security. In *EMTC*, pages 1–5. IEEE, 2014.

[16] Tasneem Yousuf, Rwan Mahmoud, Fadi Aloul, and Imran Zualkernan. Internet of things (iot) security: Current status, challenges and countermeasures. *International Journal for Information Security Research*, 5:608–616, 12 2015.

[17] Vipindev Adat and B B Gupta. Security in internet of things: issues, challenges, taxonomy, and architecture. *Telecommunication Systems*, pages 1–19, 06 2017.

[18] Fatma Alshohoumi, Mohammed Sarrab, Abdullah Al-Hamdani, and Dawood Al-Abri. Systematic review of existing iot architectures security and privacy issues and concerns. *International Journal of Advanced Computer Science and Applications*, 10, 01 2019.

[19] Liliana Antão, Rui Pinto, João Pedro Reis, and Gil Gonçalves. Requirements for testing and validating the industrial internet of things. 04 2018.

[20] Md Khan, Nashid Shahriar, Reaz Ahmed, and R. Boutaba. Simple: Survivability in multi-path link embedding. 11 2015.

[21] Quan Le, Thu Ngo-Quynh, and Thomaz Magedanz. Rpl-based multipath routing protocols for internet of things on wireless sensor networks. pages 424–429, 10 2014.

[22] Fadi M. Al-Turjman, Hossam S. Hassanein, Waleed Alsalih, and Mohamed Ibnkahla. Optimized relay placement for wireless sensor networks federation in environmental applications. *Wireless Communications and Mobile Computing*, 11(12):1677–1688, 2011.

[23] Bogdan Pavkovic, Fabrice Theoleyre, and Andrzej Duda. Multipath opportunistic rpl routing over ieee 802.15.4. In Ahmed Helmy, Björn Landfeldt, and Luciano Bononi, editors, *MSWiM*, pages 179–186, New York, NY, USA, 2011. ACM. 617111.

[24] Xiaofeng Han, Xiang Cao, Errol L. Lloyd, and Chien-Chung Shen. Fault-tolerant relay node placement in heterogeneous wireless sensor networks. *IEEE Trans. Mob. Comput.*, 9(5):643–656, 2010.

[25] Mark Reitblatt, Marco Canini, Arjun Guha, and Nate Foster. Fattire: Declarative fault tolerance for software defined networks. In *Proceedings of the second workshop on Hot topics in software defined networks*, HotSDN '13, New York, NY, USA, 2013. ACM.

[26] Francisco Javier Ros and Pedro Miguel Ruiz. Five nines of southbound reliability in software-defined networks. In *Proceedings of the Third Workshop on Hot Topics in Software Defined Networking*, HotSDN '14, pages 31–36, New York, NY, USA, 2014. ACM.

[27] Brent Stephens, Alan L. Cox, and Scott Rixner. Plinko: Building provably resilient forwarding tables. In *Proceedings of the Twelfth ACM Workshop on Hot Topics in Networks*, HotNets-XII, pages 26:1–26:7, New York, NY, USA, 2013. ACM.

[28] Neda Beheshti and Ying Zhang. Fast failover for control traffic in software-defined networks. In *GLOBECOM*, pages 2665–2670. IEEE, Dec 2012.

[29] Soma Bandyopadhyay, Munmun Sengupta, Souvik Maiti, and Subhajit Dutta. Role of middleware for internet of things: A study. *International Journal of Computer Science and Engineering Survey*, 2, 08 2011.

[30] M. R. Rahman and R. Boutaba. Svne: Survivable virtual network embedding algorithms for network virtualization. *IEEE Transactions on Network and Service Management*, 10(2):105–118, June 2013.

[31] Jin Xiao and Raouf Boutaba. Reconciling the overlay and underlay tussle. *Networking, IEEE/ACM Transactions on*, 22:1489–1502, 10 2014.

[32] David Oppenheimer, Archana Ganapathi, and David A. Patterson. Why do internet services fail, and what can be done about it? In *Proceedings of the 4th Conference on USENIX Symposium on Internet Technologies and Systems - Volume 4*, USITS'03, pages 1–1, Berkeley, CA, USA, 2003. USENIX Association.

[33] Jörg Domaschka, Christian Berger, Hans P. Reiser, Philipp Eichhammer, Frank Griesinger, Jakob Pietron, Matthias Tichy, Franz J. Hauck, and Gerhard Habiger. Sorrir: A resilient self-organizing middleware for iot applications [position paper]. In *Proc. of the 6th International Workshop on Middleware and Applications for the Internet of Things (M4IoT'19)*, 2019.

[34] Ilhem Fajjari, Nadjib Aitsaadi, and Guy Pujolle. Cloud networking: An overview of virtual network embedding strategies. pages 1–7, 10 2013.

[35] C. Wang, H. T. Vo, and P. Ni. An iot application for fault diagnosis and prediction. In *2015 IEEE International Conference on Data Science and Data Intensive Systems*, pages 726–731, Dec 2015.

[36] Bhagya Silva, Murad Khan, and Kijun Han. Internet of things: A comprehensive review of enabling technologies, architecture, and challenges. *IETE Technical Review*, pages 1–16, 02 2017.

[37] Christos Tsigkanos, Timo Kehrer, and Carlo Ghezzi. Modeling and verification of evolving cyber-physical spaces. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*, ESEC/FSE 2017, pages 38–48, New York, NY, USA, 2017. ACM.

[38] Hao Chen, Xueqin Jia, and Heng Li. A brief introduction to iot gateway. In *IET International Conference on Communication Technology and Application (ICCTA 2011)*, pages 610–613, Oct 2011.

[39] Mohammad Abdur Razzaque, Marija Milojevic, Andrei Palade, and Siobhán Clarke. Middleware for internet of things: a survey. *IEEE Internet of Things Journal*, 3:1–1, 01 2015.