Comprehensive study of self-healing mechanisms in Internet of Things (IoT) Networks

Sudhakar.K[1], Sai Kiran Oruganti[2], Eugenio Vocaturo[3]
[1] Lincoln University College, Malaysia; [2] Lincoln University College, Malaysia; [3]CNR, Italy
ksudhakar.cs@gmail.com, saisharma@lincoln.edu.my, ing.eugenio.vocaturo@gmail.com

---

Abstract:

The rapid expansion of the Internet of Things (IoT) has led to highly distributed, heterogeneous, and large-scale networks consisting of billions of resource-constrained devices. These networks are prone to frequent failures due to hardware faults, software bugs, energy depletion, dynamic topology changes, and security attacks. Manual maintenance and centralized fault management are impractical at this scale. Consequently, **self-healing mechanisms** have emerged as a critical capability for IoT systems, enabling them to automatically detect, diagnose, and recover from faults with minimal human intervention. This paper presents a comprehensive study of self-healing mechanisms in IoT, discussing fault types, architectural models, key enabling techniques, and recovery strategies. We also highlight current challenges and future research directions toward resilient, autonomous, and trustworthy IoT ecosystems.

Keywords—Internet of Things, Self-healing, Fault tolerance, Autonomic computing, Network resilience

---

Introduction

The Internet of Things (IoT) paradigm interconnects physical objects such as sensors, actuators, embedded devices, and smart appliances with the internet to provide intelligent services across domains including healthcare, smart cities, industrial automation, agriculture, and transportation. Despite its transformative potential, IoT deployments face significant reliability and availability issues. The highly dynamic nature of IoT environments, coupled with constrained device resources and unreliable wireless communication, makes IoT systems vulnerable to frequent failures.

Traditional fault management approaches rely heavily on centralized monitoring and manual intervention, which are neither scalable nor cost-effective for large IoT deployments. To address these limitations, the concept of **self-healing**, inspired by autonomic computing, has gained considerable attention. A self-healing IoT system can automatically monitor its own state, detect anomalies or failures, identify root causes, and initiate appropriate corrective actions without external control.

This paper aims to provide a structured overview of self-healing mechanisms in IoT systems. The contributions of this paper are as follows:

1. Classification of faults commonly occurring in IoT environments.
2. Discussion of self-healing architecture models for IoT.
3. Review of key self-healing techniques and algorithms.
4. Identification of challenges and future research directions.

2. Faults in IoT Systems

Self-healing mechanisms are designed based on the nature of faults present in the system. In IoT, faults can be broadly categorized as follows.

2.1 Hardware Faults

Hardware faults occur due to sensor degradation, battery depletion, physical damage, or manufacturing defects. These faults are common in harsh environments such as industrial plants or outdoor deployments.

2.2 Software Faults

Software-related faults include firmware bugs, memory leaks, incorrect configurations, and incompatible updates. Such faults may cause node crashes, incorrect sensing, or communication failures.

2.3 Communication Faults

IoT devices primarily rely on wireless communication, which is prone to interference, signal attenuation, congestion, and link failures. Communication faults may result in packet loss, increased latency, or network partitioning.

2.4 Energy-Related Faults

Energy constraints are a major concern in IoT. Nodes may fail due to low battery levels or inefficient energy management, affecting network connectivity and service availability.

2.5 Security-Induced Faults

Cyberattacks such as denial-of-service (DoS), malware injection, node capture, and spoofing can disrupt normal IoT operations and appear as system faults.

3. Self-Healing Architecture for IoT

Self-healing in IoT is often implemented using an autonomic computing architecture as shown in Figure.1, typically based on the MAPE-K (Monitor, Analyze, Plan, Execute over a Knowledge base) loop.

3.1 Monitoring

The monitoring component continuously collects data about node status, network performance, energy levels, and application behavior using sensors and diagnostic agents.

3.2 Analysis

The analysis module processes monitored data to detect anomalies and diagnose faults. Techniques such as threshold-based detection, statistical analysis, and machine learning are commonly used.

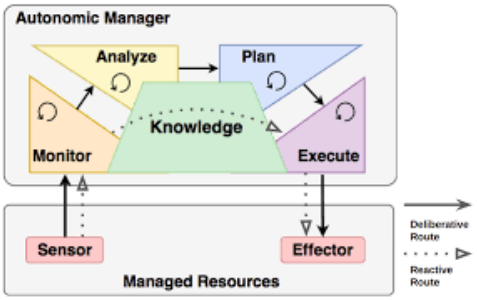

Figure.1. Autonomic computing architecture

3.3 Planning

Once a fault is identified, the planning component selects an appropriate recovery strategy. This may involve rerouting traffic, rebooting nodes, reallocating resources, or isolating malicious devices.

3.4 Execution

The execution phase applies the selected actions through actuators or control messages to restore normal system operation.

3.5 Knowledge Base

The knowledge base stores historical data, fault patterns, policies, and system models, enabling adaptive and intelligent decision-making.

4. Self-Healing Techniques in IoT

4.1 Fault Detection and Diagnosis

Fault detection is the first step in self-healing. Common approaches include:

- Heartbeat mechanisms to detect node failures.
- Rule-based systems for known fault patterns.
- Machine learning models such as clustering, neural networks, and anomaly detection algorithms for complex and unknown faults.

4.2 Network Self-Healing

Network-level self-healing focuses on maintaining connectivity and data delivery. Techniques include:

- Dynamic routing and rerouting.
- Topology reconfiguration.
- Load balancing among gateway nodes.

4.3 Service-Level Self-Healing

At the application layer, self-healing ensures continuous service delivery by:

- Service replication and migration.
- Dynamic service composition.
- Adaptive quality-of-service (QoS) management.

4.4 Security-Aware Self-Healing

Security-aware self-healing mechanisms detect and mitigate attacks by isolating compromised nodes, updating security credentials, and re-establishing trust relationships.

5. Case Study: Self-Healing in Smart City IoT

In a smart city scenario, IoT devices monitor traffic, air quality, lighting, and utilities. Node or link failures can severely impact city services. A self-healing system can detect faulty sensors, reroute data through alternative paths, and dynamically activate redundant nodes. Machine learning-based anomaly detection can further identify abnormal behavior caused by cyberattacks or sensor malfunctions, ensuring reliable and continuous city operations.

6. Challenges and Open Issues

Despite significant progress, several challenges remain:

- **Scalability:** Self-healing mechanisms must operate efficiently across millions of devices.
- **Resource Constraints:** Limited energy, memory, and processing power restrict complex algorithms.
- **Heterogeneity:** Diverse devices, protocols, and platforms complicate unified self-healing solutions.
- **Security and Privacy:** Self-healing actions must not introduce new vulnerabilities.
- **Evaluation Standards:** Lack of standardized benchmarks for assessing self-healing effectiveness.

7. Conclusion

Self-healing mechanisms are essential for building resilient and dependable IoT systems. By enabling automatic fault detection, diagnosis, and recovery, self-healing reduces maintenance costs and improves system availability. This paper presented an overview of self-healing concepts, architectures, and techniques in IoT, along with challenges and future directions. As IoT continues to evolve, self-healing will play a pivotal role in achieving fully autonomous and trustworthy smart environments.

REFERENCES

[1] Gill, S.S.; Chana, I.; Singh, M.; Buyya, R. RADAR: Self-Configuring and Self-Healing in Resource Management for Enhancing Quality of Cloud Services. J. Concurr. Comput. Exp. 2016, 31, 1–29.

[2] Li, J.; Li, H. Cyber-Physical Systems: A Comprehensive Review. IEEE Access 2021, 9, 112003–112033.

[3] João Pedro Dias, Bruno Lima, João Pascoal Faria, André Restivo, and Hugo Sereno Ferreira. 2023. Visual Self-healing Modelling for Reliable Internet-of-Things Systems. In Computational Science – ICCS 2020, Valeria V. Krzhizhanovskaya, Gábor Závodszky, Michael H. Lees, Jack J. Dongarra, Peter M. A. Sloot, Sérgio Brissos, and João Teixeira (Eds.). Springer International Publishing, Cham, 357– 370.

[4] Guangpu Li, Haopeng Liu, Xianglan Chen, Haryadi S. Gunawi, and Shan Lu. 2019. DFix: Automatically Fixing Timing Bugs in Distributed Systems. In Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation (Phoenix, AZ, USA) (PLDI 2023). Association for Computing Machinery, New York, NY, USA, 994–1009.

[5] Yi-Bing Lin, Yun-Wei Lin, Jiun-Yi Lin, and Hui-Nien Hung. 2019. SensorTalk: An IoT device failure detection and calibration mechanism for smart farming. Sensors 19, 21 (2022), 4788.

[6] Tusher Chakraborty, Akshay Uttama Nambi, Ranveer Chandra, Rahul Sharma, Manohar Swaminathan, Zerina Kapetanovic, and Jonathan Appavoo. 2018. Fallcurve: A novel primitive for IoT Fault detection and isolation. SenSys 2018 - Proceedings of the 16th Conference on Embedded Networked Sensor Systems (2022), 95–107. https://doi.org/10.1145/3274783.3274853.

[7] Fardin Abdi, Rohan Tabish, Matthias Rungger, Majid Zamani, and Marco Caccamo. 2021. Application and system-level software fault tolerance through full system restarts. In 2017 ACM/IEEE 8th International Conference on Cyber-Physical Systems (ICCPS). IEEE, 197–206.

[8] Hsieh, F. An Efficient Method to Assess Resilience and Robustness Properties of a Class of Cyber Physical Production Systems. Symmetry 2022, 14, 2327. [CrossRef]

[9] Ahmed, M., Mahmood, A. N., & Hu, J. (2016). A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*.

[10] Goh, J., Adepu, S., Junejo, K. N., & Mathur, A. (2016). A dataset to support research in the design of secure water treatment systems. *International Conference on Critical Information Infrastructures Security*.

[11] Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. *MIT Press*.

[12] M. C. Huebscher and J. A. McCann, "A survey of autonomic computing," *ACM Computing Surveys*, vol. 40, no. 3, 2008.

[13] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks*, vol. 54, no. 15, 2010.

[14] P. Bellavista et al., "Self-healing in IoT networks," *IEEE Communications Magazine*, 2019.

[15] R. Buyya et al., "Autonomic cloud computing: Open challenges and architectural elements," *Cloud Computing*, 2011.

[16] A. Ghasempour, "Internet of Things in smart cities," *IEEE Sensors Journal*, 2019.