


Disaster-FD: Federated Failure Detection in Disaster Scenarios

Abadio de Paulo Silva  [Federal University of Uberlândia | abadiops@ufu.br]


Anubis Graciela de M. Rossetto  [Sul-rio-grandense Federal Institute | anubisrossetto@ifsul.edu.br]

Pierre Sens  [Sorbonne University / CNRS | pierre.sens@lip6.fr]

Luciana Arantes  [Sorbonne University / CNRS | luciana.arantes@lip6.fr]

Rafael Pasquini  [Federal University of Uberlândia | rafael.pasquini@ufu.br]

Paulo Coelho   [Federal University of Uberlândia | paulocoelho@ufu.br]

 Faculty of Computing (FACOM), Federal University of Uberlândia (UFU), Av. João Naves de Ávila, 2121 - Santa Mônica, Uberlândia - MG, 38400-902, Brazil.

Received: 11 April 2025 • **Accepted:** 9 June 2025 • **Published:** 20 July 2025

Abstract This paper explores advanced features of Disaster-FD, a failure detector tailored for disaster-prone environments, with a specific focus on real-time monitoring of Internet of Things (IoT) networks. Leveraging federated monitoring, Disaster-FD enables the monitoring of geographically distributed regions to enhance network resilience. Inspired by Impact-FD, our proposed algorithm incorporates active monitoring and federated capabilities to ensure network reliability under adverse conditions. We conducted comprehensive experiments on the IoT-LAB platform to evaluate the robustness and resilience of Disaster-FD during potential disaster scenarios. These experiments assessed key parameters, including reliability thresholds, confidence levels, and impact factors, while ensuring efficient energy consumption and maintaining high network trust. Extensive evaluations, involving up to four geographically distinct regions in France and nearly a hundred IoT devices, demonstrate the effectiveness of Disaster-FD. Our findings highlight the potential of the algorithm to improve disaster response through enhanced IoT network monitoring, and we outline future directions for further development and optimization.

Keywords: IoT, Disaster Management, Failure Detectors, Federated Monitoring

1 Introduction

Natural disasters represent one of the greatest threats to contemporary society, causing devastating impacts ranging from disrupting essential services such as water and power supply to significant economic losses, damage to public and private properties, and most critically, the loss of human lives. These events afflict the world and, particularly in Brazil, most natural disasters are climate-related.

According to a report by the National Confederation of Municipalities (CNM) in 2022, about 3,400 cities in Brazil were directly affected by natural disasters [Janoneda, 2022]. This challenging context underscores the urgent need to develop effective strategies to monitor and manage the risks associated with these disasters. A promising approach to address this challenge is to leverage remote monitoring technologies based on the Internet of Things (IoT). The interconnectivity of various objects in a network, provided by the IoT paradigm, offers a unique opportunity to improve disaster response.

However, the effectiveness of this strategy is often compromised in large-scale disaster scenarios, where communication infrastructure, including IoT devices installed for monitoring, may fail. This failure can underestimate the severity of the disaster due to a lack of data and prevent critical alarms from triggering. Thus, it becomes essential to develop robust algorithms capable of efficiently monitoring an IoT network, ensuring device availability, and establishing a level of trust, even under adverse conditions.

In such scenarios, the importance of effective failure de-

tectors becomes evident. The work in [Chandra and Toueg, 1996], a pioneer in the study of unreliable failure detectors, highlights the importance of two fundamental properties, completeness and accuracy, provided by failure detectors that ensure that algorithms using them maintain consistency in their decisions and do not become blocked indefinitely [Chandra *et al.*, 1996].

This article is in the context of the STIC-AMSud ADMITS and ITERATION-D projects, in cooperation with universities in Brazil, France, Uruguay, and Chile, and aims to develop an algorithm for real-time monitoring of Internet of Things networks. Inspired by Impact Failure Detector [Rossetto *et al.*, 2018], or *Impact-FD*, the new proposed algorithm extends the latter by adding certain features, such as active and federated monitoring of IoT devices. The main goal is to monitor the availability of IoT devices and establish a level of reliability for the network, considering a specific set of monitored processes. The proposal also evaluates the impact of Disaster-FD on the energy consumption of IoT devices in the network.

The IoT-LAB [Adjih *et al.*, 2015], with over 1500 sensor nodes spread across various locations in France, including Grenoble, Lille, Saclay, and Strasbourg, stands out as one of the largest open testbeds available to the international scientific community. To facilitate experimentation in complex IoT networks, IoT-LAB supports a variety of communication protocols.

In addition, the IoT-LAB enables federated experiences. This means that devices can be programmed and managed

simultaneously across multiple regions, providing an ideal platform for testing algorithms and distributed applications on a network that simulates the complexity of the global Internet.

The concept of federated monitoring in IoT networks refers to the practice of integrating and managing multiple autonomous IoT device networks that are geographically distributed or belong to different administrative domains. This monitoring model is designed to improve the efficiency, security, and resilience of large IoT systems, providing more effective supervision and a coordinated response to incidents or failures. By monitoring different regions simultaneously, it is possible to quickly identify when an area begins to deteriorate due, for instance, to failures or anomalies. This detection is crucial in critical infrastructures, such as power networks, where a failure in one area can cascade to other regions.

In a federated system, each IoT network operates independently but shares information with other networks to improve global surveillance and management. This approach is emphasized by the survey in [Atzori et al., 2010], in which the authors discuss the need for collaboration between devices and distributed systems.

Disaster-FD was initially introduced in [Silva et al., 2024]. This work extends the previous publication by taking advantage of some features of Disaster-FD not explored in depth, such as effective federated monitoring with different monitoring sets per region, for a wider setup where more comprehensive and larger experiments are required. The latter are carried out in four regions of France and involve more than a hundred IoT devices. Reorganization of monitored devices into individual sets for each monitored region enhances the effectiveness of Disaster-FD. The results show that the implementation of different thresholds for local and remote regions has made Disaster-FD much more flexible, confirming the advantages of federated collaboration in identifying potential disaster situations, including the detection of partial connectivity issues that could lead to false positives.

The remainder of the paper is organized as follows. Section 2 introduces the system model and associated definitions. Section 3 describes Disaster-FD. Section 4 discusses the estimated arrival calculation. Section 5 details the implementation, with the experimental evaluation in Section 6. Section 7 briefly compares Disaster-FD with related work, and Section 8 concludes the paper.

2 System Model and Definitions

This section presents the system model and definitions related to the scope of the work.

2.1 System Model

This work considers a distributed system consisting of a finite set of processes

$$\Pi = \{q_1, \dots, q_n\}, \text{ where } |\Pi| = n, (n \geq 2)$$

and the existence of only one process per node or sensor. Each node (or process) has a unique identifier. The identifiers are ordered consecutively. Processes can fail by crash-

ing and do not recover. A process is considered correct if it does not fail throughout the entire execution.

This distributed system is **asynchronous**, defined as a system where there is no bound on the transmission time for messages or execution time for a processing step, meaning there are no assumptions related to timing [Cristian and Fetzer, 1999; Verissimo and Rodrigues, 2012]. In this type of system, no mechanism can guarantee the failure of a remote process, as it is impossible to differentiate a failed process from one that is merely slow or experiencing slow communication.

The system has **fair lossy** communication channel. According to [Aguilera et al., 2004], a lossy channel satisfies the property of integrity, meaning that a process q receives a message m from another process p at most once, and only if p previously sent m to q . Practically, this means that messages cannot be created spontaneously, and if a message m is not lost, it is eventually received at its destination. In addition, channels can intermittently drop messages. Fairness of losses requires that if a correct process p sends a message m an infinite number of times then the channel will deliver m an infinite number of times.

2.2 Unreliable Failure Detectors

Fault detection is a key element for ensuring the reliability and stability of distributed systems, especially asynchronous ones. Failure detectors can be used to circumvent the impossibility of the FLP (Fischer, Lynch, and Patterson) theorem [Fischer et al., 1985; Chandra et al., 1996], which shows that in an asynchronous setting, where only one process might crash, no distributed algorithm solves the consensus problem [Barborak et al., 1993; Leslie, 1998] deterministically.

The work in [Chandra and Toueg, 1996] introduced the concept of unreliable failure detectors, defined by two fundamental properties: completeness and accuracy. **Completeness** characterizes the failure detector's capability of suspecting faulty processes, while **accuracy** characterizes the failure detector's capability of not suspecting correct processes, i.e., restricts the mistakes that the failure detector can make. In practice, these failure detectors produce an output list of processes considered suspicious.

Disaster-FD, originally proposed in [Silva et al., 2024], extends the definition of failure detectors, using the same approach defined in [Rossetto et al., 2018]. In this context, there is a process $p \in \Pi$ that monitors subsets S_i of Π . Each process in each S_i connects to p via a communication channel. The process p can monitor many subsets S_i simultaneously. A process $q_i \in \Pi$ can belong to multiple subsets.

Thus, unlike traditional detectors defined in [Chandra and Toueg, 1996], Disaster-FD, similarly to Impact-FD, can be defined as an unreliable failure detector that provides output related to the **confidence level** in the processes in S . If the confidence level is equal to or higher than a **threshold** defined by the user, then the system is considered reliable. The work in [Rossetto et al., 2018] discusses the equivalence between such concepts and the definitions in [Chandra and Toueg, 1996].

3 Disaster-FD

By extending Impact-FD [Rossetto *et al.*, 2018], Disaster-FD is a failure detector conceived for disaster-prone environments with multiple regions that provides intra- and inter-regional real-time monitoring.

Disaster-FD introduces the use of multiple monitor processes, each deployed in a monitored region, as illustrated in **Figure 1**. In this scenario, each region has a monitor process and a set of IoT devices (sensors). The monitor of each region observes the devices in its region and a subset of the processes (IoT devices and the monitor) in another region. This arrangement enhances fault detection by preventing the complete failure of a region from going unnoticed.

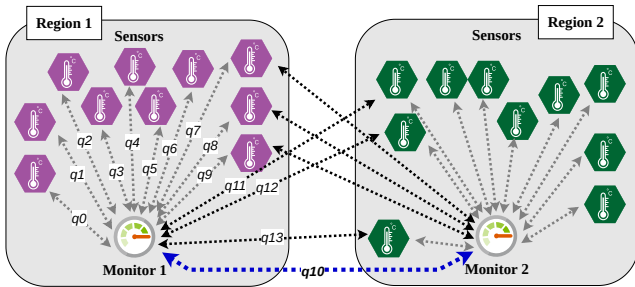


Figure 1. Monitoring scenario with 2 regions of 10 sensors each.

In this context, each process q_i within a subset $S_j \subset \Pi$ is assigned an impact factor. This factor, a positive integer value, reflects the relative importance of the process within the system. For example, in the scenario shown in **Figure 1**, a higher impact factor may be assigned to monitors compared to the impact factor of sensors, indicating that the failure of a monitor has a stronger consequence on the good operation of the system than the failure of a sensor.

Each monitor process p executes the Disaster-FD algorithm, i.e., regularly assesses the connectivity and calculates the confidence level for each monitored process in each monitored set S_j , thereby establishing the overall system reliability. This confidence level is determined by the sum of the impact factors of the processes in S_j that, at that moment, are not considered faulty.

In addition, a threshold defines the minimum level of reliability required. The system is considered trusted if the confidence level in each subset S_j is greater than or equal the defined threshold.

3.1 Formalization

This section formalizes the definitions and concepts used in this paper. Disaster-FD follows the same notation adopted in [Rossetto *et al.*, 2018], expanding the concepts to a multi-monitoring or federated monitoring scenario. This incorporation enhances the system's efficiency, accuracy, and adaptability, essential requirements for addressing the uncertainties and rapid changes in disaster scenarios.

The **Impact Factor** assigned to each process corresponds to a positive integer value that indicates its relative importance in each set S_j . The impact factor of each monitored process i (I_i) and the unique identifier of the process (id_i)

make up each set $S_j \subset \Pi, 1 \leq j \leq m$ of monitored processes. Thus, the values in each subset correspond to

$$S_j = \{\langle id_{1,j}, I_{1,j} \rangle, \langle id_{2,j}, I_{2,j} \rangle, \dots, \langle id_{k,j}, I_{k,j} \rangle\}$$

for each subset $S_j, 1 \leq j \leq m$.

For each monitored set S_j , the subset $T_p^j(t)$ represents the processes that the monitor p does not suspect at the time t . Complementarily, the set $F_p^j(t)$ represents the processes considered faulty by the monitor p at the moment t in S_j .

The **Trust Level** indicates the confidence level of the monitor process p in the set of processes in S_j at a given moment, calculated by $TL_p^j(t)$. It represents the sum of the impact factors of the non-faulty processes, that is,

$$TL_p^j(t) = \sum_i I_i, \quad \forall i \in T_p^j(t).$$

Each monitor can track various subsets of processes, with different levels of trust and individual impact factors. The set S^* encompasses the m unique subsets being monitored, indicated by

$$S^* = \{S_1, S_2, S_3, \dots, S_m\}.$$

Finally, the **Threshold** defines the minimum reliability limit for each set in S^* , mathematically represented by $Th^* = \{Th_1, Th_2, \dots, Th_m\}$, where each Th_j is related to the minimum level of trust required for a subset of processes S_j .

The values in Th^* are used by the monitor to verify the trust in the processes of the subsets in S^* . If, for each subset $S_j \in S^*, 1 \leq j \leq m$

$$TL_p^j(t) \geq Th_j,$$

then S^* is considered reliable (*trusted*) at time t by the monitor p ; otherwise, S^* is considered not reliable (*not trusted*).

This class of algorithms introduces the concept of **Flexibility Property**, which denotes the failure detector's ability to tolerate a certain margin of failures or false suspicions, that is, its ability to consider different sets of responses that lead the system to states to be considered trusted.

Table 1 presents an example with a set of monitored processes similar to the scenario depicted in Figure 1. The set S_1 of the monitor in Region 1 comprises the processes q_0 to q_9 , corresponding to local sensors of Region 1 (purple), the processes q_{11} to q_{13} corresponding to remote sensors under monitoring (green), and q_{10} as the monitor of Region 2. The maximum value of $TL_p^1(t)$ is $\sum_{i=0}^{13} I_i = 220, \forall t > 0$, with $\sum_{i=0}^9 I_i = 100$ for local sensors and $\sum_{i=10}^{13} I_i = 120$ for remote sensors and monitor. In this situation, the chosen threshold should reflect the monitoring objective. For example, to ensure that at least one process from each region always responds, it is necessary to have $120 < Th_1 \leq 220$ and $TL_p^1(t) > Th_1, \forall t > 0$, where the value 120 represents the maximum Trust Level of the remote region, and 220 corresponds to the maximum possible value for the Trust Level. The monitor of Region 2 can adopt an equivalent strategy.

Table 1. Set S_1 monitored by the monitor process of Region 1, with processes q_i and impact factors.

Set	Process and Impact Factor
S_1	$\langle q_0, 10 \rangle, \langle q_1, 10 \rangle, \langle q_2, 10 \rangle, \langle q_3, 10 \rangle, \langle q_4, 10 \rangle,$ $\langle q_5, 10 \rangle, \langle q_6, 10 \rangle, \langle q_7, 10 \rangle, \langle q_8, 10 \rangle, \langle q_9, 10 \rangle,$ $\langle q_{10}, 60 \rangle, \langle q_{11}, 20 \rangle, \langle q_{12}, 20 \rangle, \langle q_{13}, 20 \rangle$

3.2 Quality of Service (QoS) Metrics

The evaluation of Disaster-FD is based on the work by [Chen et al., 2002], where they define a set of metrics to assess the Quality of Service (QoS) of unreliable failure detectors. These metrics are centered around temporal constraints, which refer to the time required to detect a failure, to correct a mistake, and the interval between two false suspicions. Specifically, the same QoS metrics used by Impact-FD are adopted:

1. **Average Detection Time (TD):** Measures the speed and efficiency of the system in detecting failures. TD measures the period from the moment a process q fails until the failure detector in p begins to continuously suspect q . This metric is critical for understanding how quickly the system responds to incidents due to sensor failures.
2. **Average Error Rate (μR):** Represents the frequency at which the failure detector makes errors per unit of time, serving as an indicator of the detector's reliability. This metric is particularly important for assessing the system's tendency to false positives or false negatives.
3. **Accuracy Probability (PA):** Assesses the likelihood that the outputs of the failure detector are correct at a random moment, providing a measure of the overall accuracy of the system over time, derived from the total duration of the false positive period relative to the total time under analysis.

This work focuses on the Accuracy Probability property and the properties defined in Section 3.1.

4 Estimation of Heartbeat Arrival

A basic monitoring mechanism consists of waiting for periodic messages from the monitored processes, commonly referred to as **heartbeats** (HB). The non-reception of a heartbeat from a monitored process at an estimation arrival time renders the latter suspect or faulty. The method proposed by [Chen et al., 2002] to estimate the arrival of the next heartbeat (EA_{k+1}) is based on the history of the arrival times of previous heartbeats and includes a safety margin (β).

In the EA_{k+1} calculation, the process p considers a sliding window with the w most recent heartbeat messages received from process q represented by m_1, m_2, \dots, m_w . The values T_1, T_2, \dots, T_w are the respective reception times of these messages, according to the local clock of p . Thus, as defined in [Chen et al., 2002], we have **Equation 1**, where Δ_i corresponds to the interval between the sending of two consecutive heartbeats.

$$EA_{k+1} = \frac{1}{w} \sum_{i=k-w}^k (T_i - \Delta_i \times i) + (k+1) \times \Delta_i \quad (1)$$

EA are then used to compute the freshness point τ . τ_{k+1} , defined in **Equation 2**, determines the moment a process q starts to be suspected if its $k+1$ heartbeat has not been received. The value of the constant safety margin β is system-dependent. **Section 6** discusses the value chosen for the experiments in this work.

$$\tau_{k+1} = EA_{k+1} + \beta \quad (2)$$

4.1 Comparison with Impact-FD

While both Disaster-FD and Impact-FD use **Equations 1 and 2** to calculate the estimated arrival time of the next heartbeat, named EA_{k+1} , the two proposals use slightly different approaches in interpreting and implementing the **Equation 1**.

The implementation of the Disaster-FD protocol uses the sequence number identifier of the heartbeat message to calculate the difference between the actual arrival time and a theoretical arrival time, defined in **Equation 1** as the product of the identifier by the fixed interval between consecutive heartbeats (Δ_i). The Impact-FD protocol, on the other hand, does not directly use the sequence number of the heartbeat, employing an incremental index to calculate the difference between the arrival time of each heartbeat and the expected arrival time, based on Δ_i .

The methodology proposed by Chen is based on adjusting the arrival time estimate using the history of arrival times of the previous w messages, considering the difference between the actual and expected arrival times, based on the regular interval between heartbeats.

Thus, Disaster-FD is more aligned with Chen's theory, as it directly incorporates the concept of heartbeat sequentially (through the sequence number), reflecting Chen's approach of adjusting estimates based on differences between actual and expected arrival times.

Although similar in structure, by using an incremental index, the Impact-FD does not always capture sequentiality directly, and thus may not accurately have the actual sequence of heartbeats.

In practice, as observed when analyzing the logs of Impact-FD experiments, this means that Impact-FD has more difficulty handling "gaps" in a sequence of heartbeats, which occur when some messages are lost. As a result, the Impact-FD implementation requires more time to detect a false positive, i.e., to realize that it has erroneously suspected a correct process.

5 Implementation

Disaster-FD has been implemented in Java and utilizes the Californium library [Kovatsch et al., 2014]. It is distinguished by its federated and multi-protocol monitoring, using CoAP (Constrained Application Protocol) requests for

monitoring IoT devices and the ICMP (Internet Control Message Protocol) to monitor nodes of neighboring federated regions. This dual approach provides flexibility and a more comprehensive network state analysis.

The CoAP “GET” method retrieves a representation of the information currently corresponding to the resource identified by the request URI (Uniform Resource Identification), which typically corresponds to a request for sensor measurement, such as temperature, pressure, etc.

Requests use the CoAP “CON” (confirmable) type to enhance communication reliability, as each message anticipates a response from the destination device. Furthermore, the system implements a timeout mechanism for these requests, ensuring that monitoring remains efficient even when a device does not respond within the expected time. Disaster-FD uses the Californium library to handle responses from CoAP requests asynchronously, managing successful returns of heartbeat messages as well as handling errors, such as the overflow of the calculated time for the next heartbeat reception and connectivity issues, allowing continuous monitoring and uninterrupted analysis flow.

5.1 CoAP Message Tracking

Initially, each IoT device is accessible via a unique URI (Uniform Resource Identifier), based on its specific IPv6 address. Using this URI, the CoAP client in the Disaster-FD implementation derives a unique device identifier, used in generating the initial value of the message sequence number, or MID (Message ID).

The MID is generated or retrieved for each device and incremented with each new request, ensuring the uniqueness and traceability of messages. In CoAP GET and CON requests, the MID is explicitly defined in the header, allowing for a precise correlation between the sent requests and the received responses.

6 Results

This study implements and tests the Disaster-FD failure detection system in the Internet of Things (IoT) environment known as FIT-IoTLAB [Adjih et al., 2015], monitoring up to four interconnected regions with more than one hundred monitored devices.

6.1 Experiments Rationale

We conducted two sets of experiments in FIT-IoTLAB, one with fewer devices within two regions, and a second configuration with four regions and a greater number of devices.

Table 2. Approximate distance between regions.

Region 1	Region 2	Distance
Grenoble	Strasbourg	540 km
Grenoble	Lille	800 km
Grenoble	Saclay	570 km
Strasbourg	Lille	550 km
Strasbourg	Saclay	500 km
Lille	Saclay	240 km

For both sets, the choice of the safety margin β used in **Equation 2** to calculate the estimated arrival time for the next heartbeat was defined based on latency tests with IoT devices in the chosen regions. After 24 hours of monitoring, the calculated value of the standard deviation of the observed latencies was chosen, corresponding to 1,500 milliseconds.

Two-region deployment: The first configuration evaluates the behavior of Disaster-FD in only two regions, Grenoble and Strasbourg, in a configuration similar to the example in **Figure 1**. The experiments intend to evaluate the failure detector in a situation with a light workload and better understand the behavior of Disaster-FD, the IoT devices, and the monitored infrastructure. In addition, we provide a comparison with its predecessor, Impact-FD.

Four-region deployment: A second set of experiments evaluates the scalability of Disaster-FD. With four regions distant up to 800 kilometers (see **Table 2**) from each other and a total of 16, 19, 10, and 17 IoT devices in the regions of Grenoble, Lille, Saclay and Strasbourg, respectively. Besides monitoring the devices of its region, a monitor of each of the regions monitors 3 of the devices and the monitor of the other regions, thus summing a total of 110 monitored devices.

6.2 Two-region deployment

In the experiments, each region monitor is responsible for monitoring 14 processes: 10 sensors of its region plus 3 sensors and the monitor of the other region. Each monitor uses the CoAP protocol for the sensors and ICMP to track other monitors.

Requests to the sensors were made using the GET method of the CoAP protocol at an interval of 5,000 milliseconds. This rate was carefully selected aiming for efficiency in fault detection, minimization of network load, and energy consumption on devices, as shown in § 6.2.1. ICMP requests to track remote monitors were issued at the same rate.

Such experiments aim not only to evaluate the effectiveness of Disaster-FD in detecting faults in real-time in an IoT environment but also to explore the interactions between network devices in federated regions. The setting of the impact factors in each region was defined similarly to that presented in **Table 1**, namely: the 10 sensors located in the local area of each region were set with an impact factor of 10. In comparison, the 3 sensors in the neighboring region received an impact factor of 20. Additionally, the monitor in the neighboring region has an assigned impact factor of 60, reflecting its importance in that area. Consequently, the trust level for this set of processes can reach a maximum of 220 ($10 \times 10 + 3 \times 20 + 60$).

The experiments evaluate the impact of Disaster-FD on IoT device power consumption, the individual device accumulated errors, the accuracy of the estimated arrival time, as well as the overall system behavior based on the metrics defined in Section 3.2 for Disaster-FD and Impact-FD.

6.2.1 Energy consumption for IoT devices

Our goal is to investigate the impact of energy consumption on IoT devices within request sending intervals. Using the

IoT-LAB platform, experiments were conducted in three distinct devices, each configured to issue GET requests via the CoAP protocol.

Figure 2 illustrates the energy consumption in watts (W) of three IoT devices at different sending intervals, ranging from 10 to 60,000 milliseconds, during a 24-hour period. The figure shows a greater consumption discrepancy between 10 milliseconds and 1,000 milliseconds sending interval. On the other hand, it is observed that from 5,000 milliseconds onward energy consumption presents a tendency to stabilize, similar to the values observed when the devices are not receiving any CoAP request (“device stopped”).

When extending the experiments to longer intervals, such as 10,000 milliseconds, 30,000 milliseconds, and 60,000 milliseconds, we observe that the power consumption remains close to the 5,000 millisecond scenario. However, these longer intervals can be not suitable in contexts that require rapid responses and frequent data updates.

Evaluation results show that a sending interval of 5,000 milliseconds represents a good trade-off, mitigating energy consumption without compromising the frequency of updates. This sending interval represents an economy in energy consumption of 0.03 W, or 2%, compared to a 10-millisecond sending interval for each device. Although energy savings are small, they can be crucial for devices operating in environments where access to power is restricted or battery replacement is unfeasible.

6.2.2 Accumulated Errors in Strasbourg

Figure 3 shows the accumulated number of errors that occurred per monitored device during the 24 hours of monitoring in the Strasbourg region.

Devices 0 to 9 correspond to local sensors, devices 11 to 13 correspond to sensors in the Grenoble region, and device 10 represents the monitor in Grenoble.

The detector did not register any failures for devices 5 and 7, while the IoT devices 0, 1, 2, 3, 4, 6, 8, and 9 showed only a few errors, reflecting the stability of the Strasbourg region. In contrast, in the neighboring region of Grenoble, both the monitored IoT devices (11, 12, and 13) and the monitor (device 10) showed a significant number of errors, indicating instability in that area.

Notably, device 10, which serves as the monitor in the Grenoble region, registered several errors. This device also acts as the central node or edge router and is crucial for managing network traffic in Grenoble. The errors observed in device 10 suggest connectivity problems, compromising communication and the operational efficiency of the region’s IoT network. Therefore, the instability of device 10 is a critical factor that can affect the performance of the Grenoble network, causing interruptions in functionality and failures to respond to requests sent by the Disaster-FD monitor.

6.2.3 Accumulated Errors in Grenoble

In a similar manner to the previous section, **Figure 4** shows the accumulated number of errors for each monitored device over 24 hours in the Grenoble region. Devices 0 to 9 correspond to local sensors, devices 11 to 13 correspond to sensors

in the Strasbourg region, and device 10 represents the monitor in the Strasbourg region.

The results provided by the Disaster-FD failure detector reinforce the observations in the previous section and **Figure 3**, especially with respect to the network instability in the Grenoble region. While the Strasbourg monitor identified instability in Grenoble, the monitoring in Grenoble also detected a high incidence of errors in its own devices (devices 0 to 9) and few in Strasbourg (devices 10 to 13), as observed in **Figure 4**, corroborating the mutual perception of network performance between the two regions.

6.2.4 Arrival Times in Strasbourg

Figure 5 analyses the behavior of device 5, located in Strasbourg, during the 24-hour monitoring period. The choice of this device for analysis is justified by its operational robustness, as evidenced by its stable performance, which is also highlighted in **Figure 3**.

The blue curve in **Figure 5** indicates the *arrival time* and corresponds to the difference between two consecutive heartbeats. The red curve indicates *estimated arrival time*, that is, the maximum time a heartbeat is expected to arrive before the device is considered faulty.

As discussed in the introduction of **Section 6.2**, the monitor adopted a safety margin of 1500 milliseconds and intervals of 5000 milliseconds for sending requests. A comparative analysis between the estimated arrival times (in red) and the actual heartbeat arrival times (in blue) revealed the stability of the system, demonstrated by the small variation in the time intervals between the estimate and the actual reception of the heartbeats.

Furthermore, the red points above the blue points indicates that the heartbeats reached the monitor before the estimated time.

Figure 6 emphasizes the relation between actual and estimated message arrival times in Strasbourg, using a Cumulative Distribution Function (CDF) plot. The X-axis represents the time in milliseconds, while the Y-axis shows the cumulative probability. The blue curve indicates the actual arrival time of events, with 50% of events occurring before 5,000 milliseconds and 100% before 8,000 milliseconds. The red curve, representing the estimated time with a safety margin of 1,500 milliseconds, shows that the estimated times are greater than the actual times, showing the accuracy of the arrival time estimates and indicating a conservative approach in the estimates and the precision of the Disaster-FD system in predicting message arrival times.

6.2.5 Network Statistics in Strasbourg

The experiment was conducted under an established threshold value of 160 (dashed red line in **Figure 7**), which serves as a safety limit for the network in the Grenoble and Strasbourg regions, as detailed in the discussion of the values in **Table 1** in **Section 3.1**. This value of 160 corresponds, for example, to situations where all nodes in Strasbourg are not suspected, and at least the monitor in Grenoble is responding.

The analysis, depicted in **Figure 7**, provides a statistical perspective on the network behavior regarding the trust level

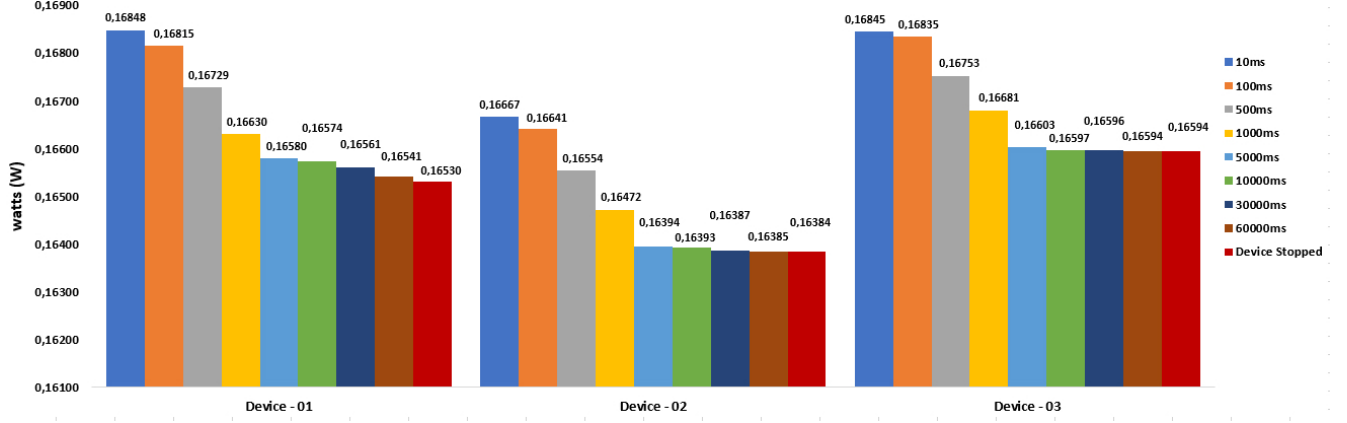


Figure 2. Energy consumption with different request intervals.

Table 3. Extract of the logs for the monitors in Strasbourg and Grenoble.

Monitor Region	Trust Level	Devices Trust Array	Probability of Accuracy	Timestamp
Strasbourg	100	T T T T T T T T T F F F F	0.8061	04-01-2024 03:17
Strasbourg	100	T T T T T T T T T F F F F	0.9756	04-01-2024 16:17
Grenoble	120	F F F F F F F F F T T T T	0.9963	04-01-2024 03:17
Grenoble	120	F F F F F F F F F T T T T	0.9676	04-01-2024 16:17

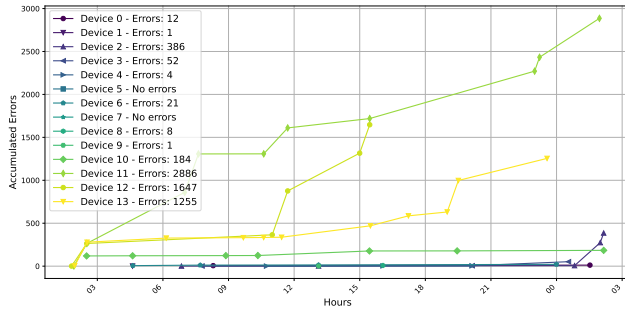


Figure 3. Accumulated Errors per device in Strasbourg.

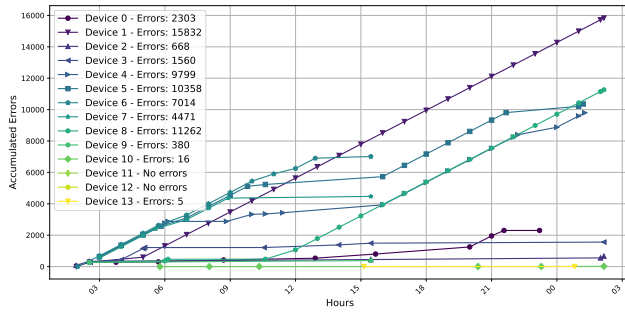


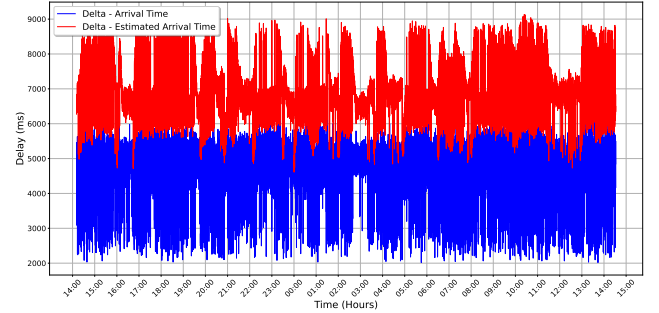
Figure 4. Accumulated errors per device in Grenoble.

($TL_p^S(t)$) (red curve).

Specifically, the trust level is influenced by the network stability parameters and, during the testing period, the accuracy of the Disaster-FD system showed a progressive improvement as the monitored devices maintained stable operations.

Figure 7 also shows that the Probability of Accuracy (PA), identified by the blue curve, remains above 95% from 8:50 onwards, highlighting the monitor's ability to accurately indicate the system state.

Finally, it is noteworthy, from the analysis of the same figure, that the regions under analysis were considered unreliable between 15:50 and 16:50 when the trust level fell below the established threshold. In a real deployment, this situation



could indicate a possible disaster, necessitating emergency actions for the affected regions.

6.2.6 Network Statistics in Grenoble

Figure 8 corroborates the results presented in **Figure 4**, indicating that the network in the Grenoble region experienced periods of instability. This conclusion can be confirmed by analyzing the trust level ($TL_p^S(t)$) (in red), which at various times approached the established safety threshold for the network (dashed red line).

Furthermore, as shown in **Table 3**, there were failures in the local devices of Grenoble, resulting in an unreliable network classification. This can be observed by the value of the complementary “Devices Trust Array” of ‘FFFFFFFFTTTT’, shown in **Table 3** for the same time in the Grenoble region, meaning that the 10 devices in Grenoble are perceived as faulty (‘F’) by the region monitor, at the same time that the 3 devices and the monitor in Strasbourg remain trusted (‘T’).

6.2.7 Comparison with Impact-FD in Strasbourg

Figure 9 illustrates the statistics for Strasbourg with results for Impact-FD. The network statistics using the Impact-FD implementation of Equation 1, showed that the Probability of Accuracy (PA) varied between approximately 0.40 and 0.90 over time. The average PA remained above the threshold of 0.40, initially lower than that of Disaster-FD, but indicating an improvement in failure detection accuracy throughout the experiment. The network safety threshold was maintained at 160. The trust level ranged between 80 and 220, showing moments of instability during the period, especially around 15:50 to 16:50, when there was a significant drop in confidence. In a real scenario, this could indicate a possible disaster, triggering emergency actions for the affected regions, a scenario also presented by the Disaster-FD failure detector.

Comparing the network statistics for Strasbourg using the Impact-FD and Disaster-FD formulas, it is observed that Disaster-FD demonstrated a probability of accuracy varying between 0.70 and 0.95, reaching 0.95 early on, as observed in **Figure 7**. In contrast, Impact-FD showed greater variability, with the PA ranging between 0.40 and 0.90, reaching 0.90 only at the end of the experiment. Thus, Disaster-FD was more stable and adapted quickly to network instabilities, while Impact-FD was less stable and adapted more slowly. **Equation 1** implementation in Disaster-FD is responsible for the observed stability. Disaster-FD considers the MID instead of a simple counter to calculate the estimated arrival time of the next message, contributing to greater accuracy and stability in failure detection.

6.2.8 Comparison with Impact-FD in Grenoble

Figure 10 illustrates the statistics for the Grenoble region using the Impact-FD implementation of **Equation 1**. For the Grenoble network, using the Impact-FD formula, the PA ranged from approximately 0.02 to 0.80. The network safety threshold was maintained at 160. The trust level varied between 120 and 200, with more pronounced instabilities throughout the experiment, which was expected due to

the region’s higher instability. The lower PA during critical moments indicated a lower reliability in failure detection, resulting in a higher occurrence of false positives.

When comparing the graphs for the Grenoble region, **Figure 8** demonstrates that Disaster-FD maintained a higher and more stable accuracy (PA), ranging from 0.60 to 0.95. At the same time, Impact-FD showed greater variability, with PA ranging from 0.02 to 0.80. Grenoble experienced more network instabilities during the experiment compared to Strasbourg. Disaster-FD quickly adapted to these instabilities, unlike Impact-FD, which had slower adaptation and inconsistencies in detection. The safety threshold was maintained at 160 for both formulas.

The trust level in the Disaster-FD formula also exhibited a clear advantage in terms of stability and less variability. Using the message ID in the Disaster-FD formula resulted in a more accurate arrival time estimate, improving accuracy (PA) and significantly reducing the probability of false positives.

6.3 Four-region deployment

This set of experiments aims to evaluate the scalability of Disaster-FD and the ability to work more effectively in a federated fashion. The assessment includes four regions, Grenoble, Lille, Saclay, and Strasbourg, with 16, 19, 10, and 17 local devices, respectively.

This configuration allows for the evaluation of important Disaster-FD features, such as federated monitoring with different sets and thresholds. A monitor process in each region watches the local devices in the corresponding region along with 3 devices and the monitor in each remaining region.

Unlike the symmetrical monitoring configuration of the two-region setting, in the current one there are different monitoring configuration sets per monitored region. In practice, this means that we can set different thresholds for each region based on the specific defined goal. The number of remote monitored devices can also be customized for each region. **Tables 5, 6, 7, and 8** show the devices and impact factor for each set of each region.

Each regional monitor watches 4 remote devices (3 IoT devices and the corresponding monitor) for each adjacent region, with a total number of 12 remote devices. This strategy enables per couple region analysis that enhances the system’s response capability even in the face of isolated failures.

The values for the safety margin β and the heartbeat interval are the same as used in the two-region deployment, namely 1,500 and 5,000 milliseconds.

Table 4 presents the quantitative distribution of the devices monitored per region. The choice of remote devices for each region tries to maximize monitoring coverage by selecting different remote devices. For instance, that the 3 remote devices monitored in Grenoble by the monitor in Lille are different from the 3 remote devices monitored by Strasbourg in Grenoble.

We adopted an impact factor of 10 per local device. Thus, if a region has 18 local devices, the total impact is 180, and the local threshold is set at 50% of that value, namely 90. This means that at least half of the local devices must be active for the network to be considered reliable.

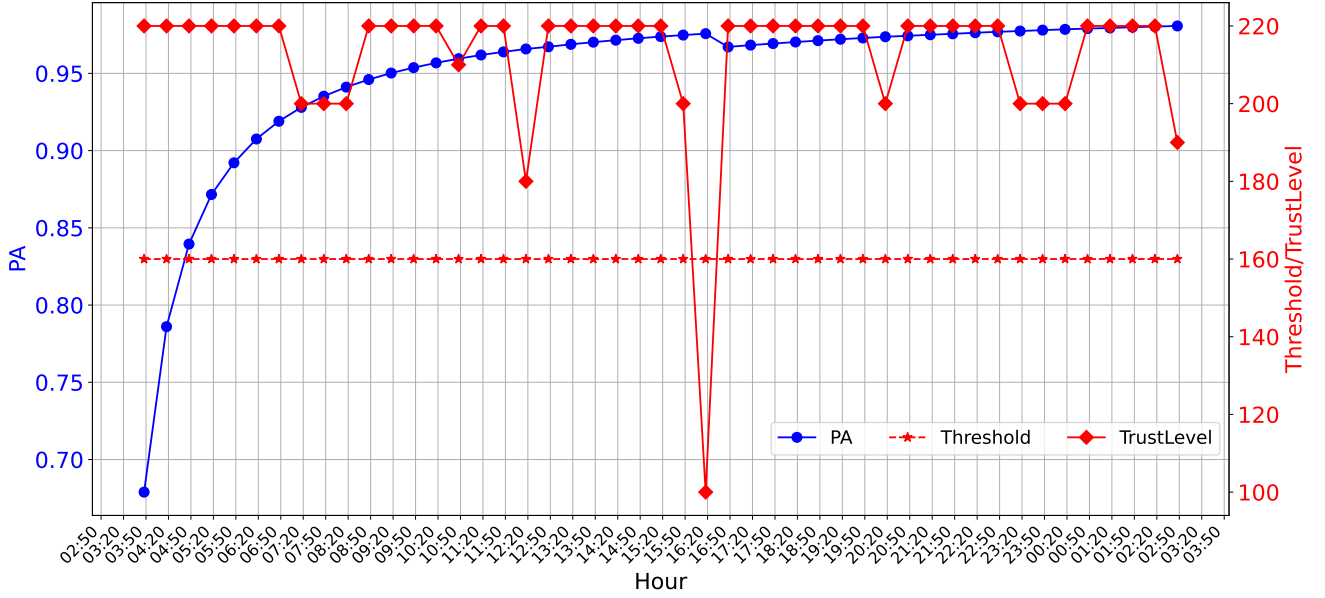


Figure 7. Network statistics in Strasbourg.

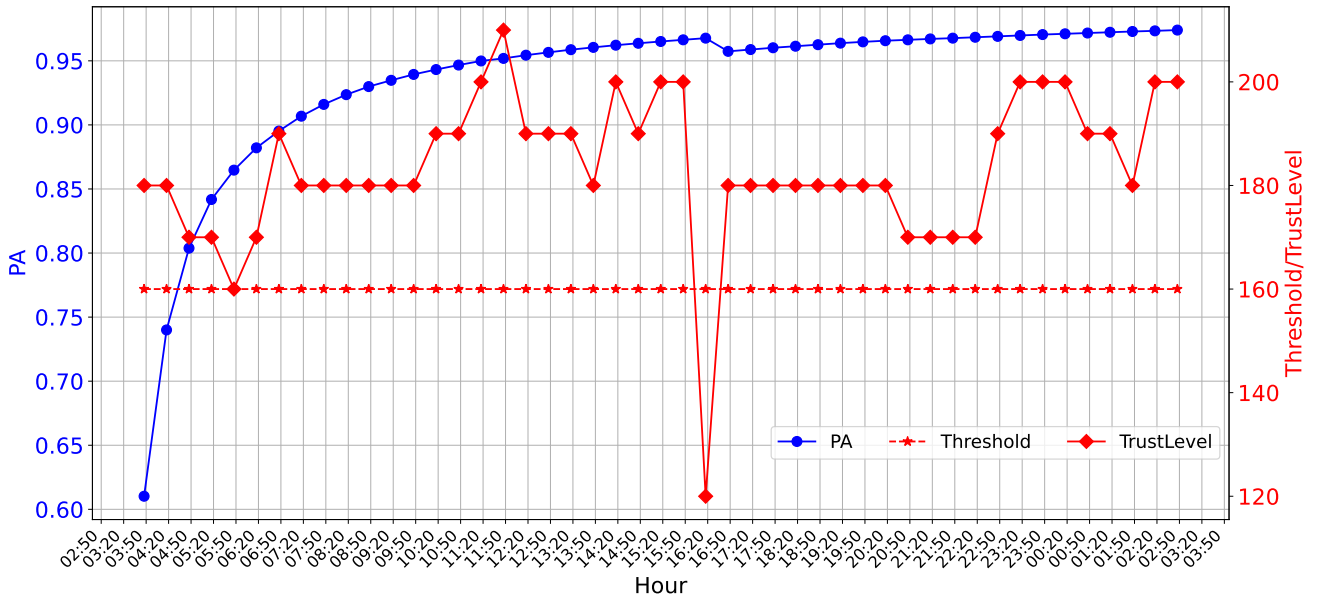


Figure 8. Network statistics in Grenoble.

Table 4. Distribution of monitored devices per region.

Region	Local Devices	Remote Devices	Total
Grenoble	16	12	28
Lille	19	12	31
Saclay	10	12	22
Strasbourg	17	12	29
Overall Total	62	48	110

For remote devices, the approach is slightly different: each of the 3 IoT devices from another region receives an impact factor of 20 (totaling 60), and the remote monitor has an impact factor of 60. In this way, the maximum trust level of the remote region reaches 120. The threshold for remote devices is then set as 60, emphasizing the need to have at least the remote monitor active to ensure federated collaboration.

The implementation of different thresholds for local and remote regions has made Disaster-FD much more flexible.

In previous scenarios involving only two regions, a single threshold was applied to both sides, without taking into account local particularities. With federated monitoring across four regions, each region can adjust its threshold according to the specific conditions and characteristics of its network. This approach not only highlights the importance of local devices which directly affect internal reliability but also ensures that interregional cooperation fundamentally depends on the operation of the remote monitor, which has a high impact.

In summary, the proposed configuration offers the following benefits:

- **Local Threshold:** Each region sets its threshold at 50% of the total impact of local devices. For example, if Lille has 19 devices (total impact of 190), the threshold is set at 95.
- **Remote Threshold:** Each region monitors four remote

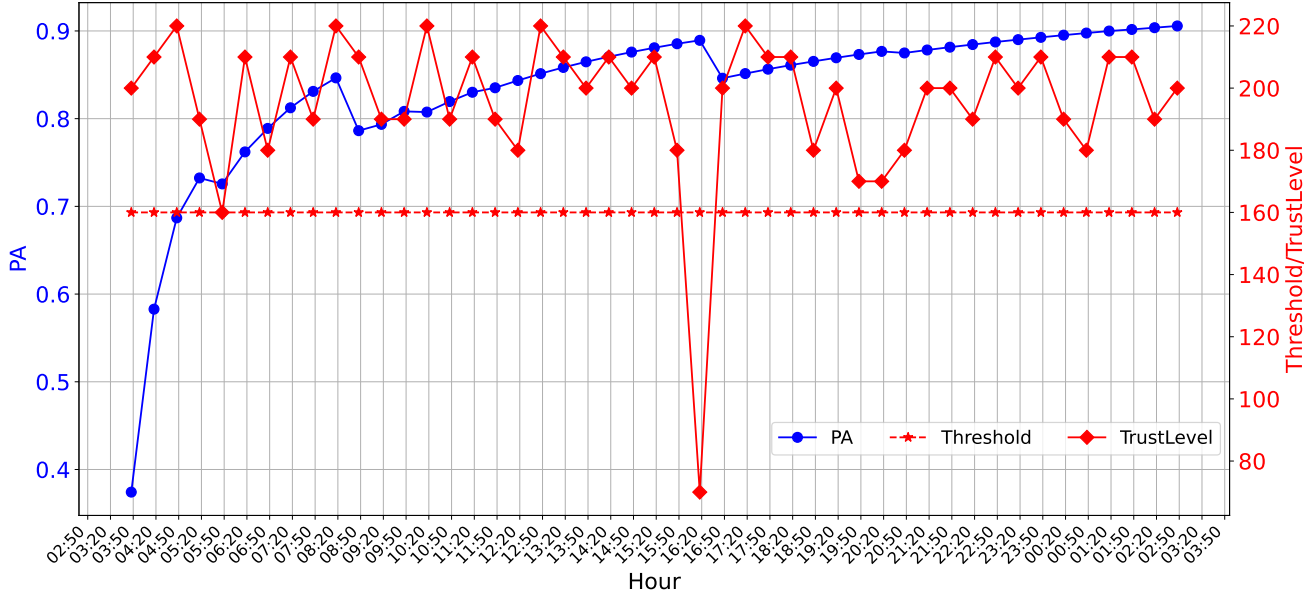


Figure 9. Network statistics in Strasbourg using Impact-FD.

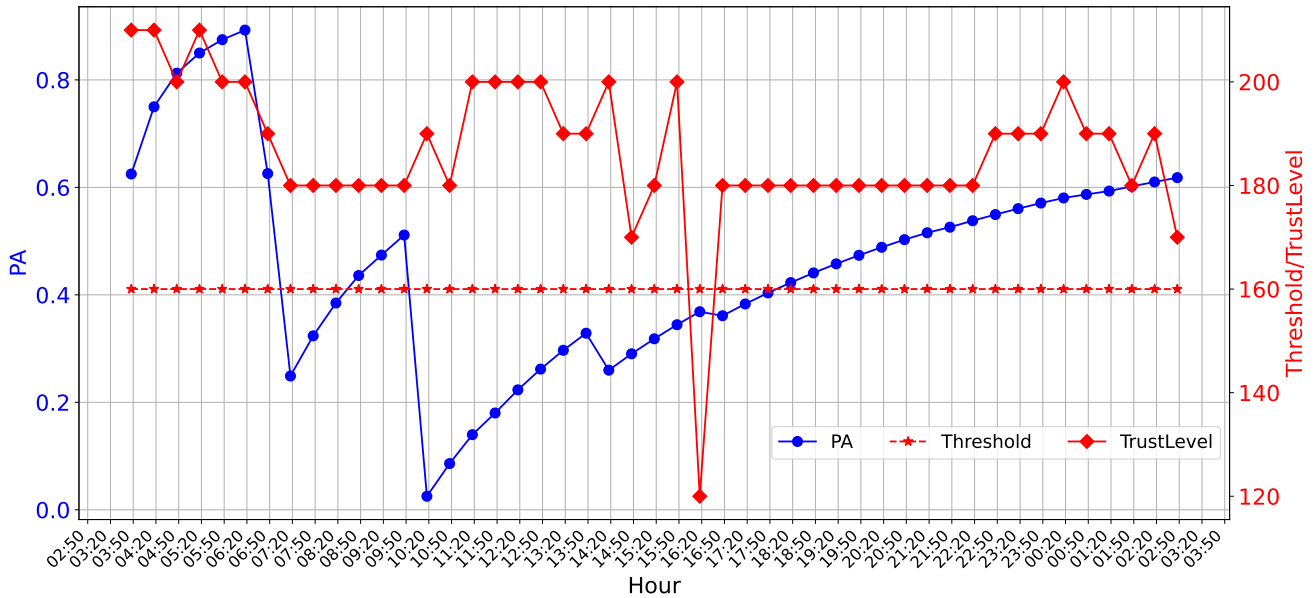


Figure 10. Network statistics in Grenoble using Impact-FD.

devices (three IoT devices and one monitor) per remote region with a total impact of 120 and defines a threshold of 60, ensuring, therefore, that at least the remote monitor (impact 60) remains operational.

- **Flexibility:** The adoption of region-specific thresholds makes the system more versatile and adaptable to changes in network conditions. This federated approach enables fine-tuned adjustments that address both local stability and interregional communication, resulting in more reliable fault detection, even in disaster scenarios.

The following sections present the results of the federated monitoring conducted by the Disaster-FD system in each region. The integrated analysis highlights the system's ability to quickly identify critical variations in network performance, as reflected by metrics such as Trust Level and Probability of Accuracy. While some regions demonstrated sta-

Table 5. Processes q_i and impact factors for Grenoble.

Region	Set	Processes
Grenoble	S_1	$\langle q_0, 10 \rangle, \dots, \langle q_{15}, 10 \rangle$
Lille	S_2	$\langle q_{16}, 20 \rangle, \langle q_{17}, 20 \rangle, \langle q_{18}, 20 \rangle, \langle q_{19}, 60 \rangle$
Saclay	S_3	$\langle q_{20}, 20 \rangle, \langle q_{21}, 20 \rangle, \langle q_{22}, 20 \rangle, \langle q_{23}, 60 \rangle$
Strasbourg	S_4	$\langle q_{24}, 20 \rangle, \langle q_{25}, 20 \rangle, \langle q_{26}, 20 \rangle, \langle q_{27}, 60 \rangle$

bility and high performance, others exhibited instabilities that were promptly detected and validated by monitors from neighboring regions.

6.3.1 Network Statistics in Grenoble

This section presents the main findings obtained from the monitoring of the Grenoble region by the Disaster-FD system, including data from the other three regions (Lille, Saclay, and Strasbourg), following the configuration illus-

Table 6. Processes q_i and impact factors for Lille.

Region	Set	Processes
Lille	S_1	$\langle q_0, 10 \rangle, \dots, \langle q_{18}, 10 \rangle$
Grenoble	S_2	$\langle q_{19}, 20 \rangle, \langle q_{20}, 20 \rangle, \langle q_{21}, 20 \rangle, \langle q_{22}, 60 \rangle$
Saclay	S_3	$\langle q_{23}, 20 \rangle, \langle q_{24}, 20 \rangle, \langle q_{25}, 20 \rangle, \langle q_{26}, 60 \rangle$
Strasbourg	S_4	$\langle q_{27}, 20 \rangle, \langle q_{28}, 20 \rangle, \langle q_{29}, 20 \rangle, \langle q_{30}, 60 \rangle$

Table 7. Processes q_i and impact factors for Saclay.

Region	Set	Processes
Saclay	S_1	$\langle q_0, 10 \rangle, \dots, \langle q_9, 10 \rangle$
Grenoble	S_2	$\langle q_{10}, 20 \rangle, \langle q_{11}, 20 \rangle, \langle q_{12}, 20 \rangle, \langle q_{13}, 60 \rangle$
Lille	S_3	$\langle q_{14}, 20 \rangle, \langle q_{15}, 20 \rangle, \langle q_{16}, 20 \rangle, \langle q_{17}, 60 \rangle$
Strasbourg	S_4	$\langle q_{18}, 20 \rangle, \langle q_{19}, 20 \rangle, \langle q_{20}, 20 \rangle, \langle q_{21}, 60 \rangle$

trated in **Table 5**.

Figure 11 exhibits the statistics obtained for Grenoble (top left) regarding Accuracy Probability (PA), Trust Level, and Threshold. Throughout the experiment, Grenoble exhibited significant stability, consistently maintaining high Trust Level values near the established maximum (160) and a PA of 1.0.

The analysis of monitoring the other regions revealed distinct behaviors. Lille (top right) displayed continuous instability, with the Trust Level frequently approaching critical values as low as 0 most of the time with some occurrences of Trust Level above the Threshold. This behavior caused notable variations in PA, indicating generally low reliability with constant false positives.

Conversely, Saclay (bottom left) demonstrated high stability, consistently maintaining Trust Level values near the maximum (between 100 and 120) and a PA at 1.0 for most of the observation period. Strasbourg (bottom right) also showed satisfactory performance, maintaining intermediate to high Trust Level values and stable PA at 1.0 throughout almost the entire monitored period.

However, towards the end of the experiment, Grenoble experienced a significant drop in Trust Level, becoming unreliable at 01:41. This critical event was promptly confirmed by monitors in neighboring regions, validating Grenoble's alert and highlighting the effectiveness of Disaster-FD in quickly identifying critical situations through its federated approach.

Power Consumption in Grenoble

Figure 12 presents the analysis of the power consumption of the IoT devices in Grenoble, revealing a particular case with progressive degradation in energy consumption over time for device 2. This behavior suggests a possible operational failure or interruption in its functioning.

The scenario is reinforced by the registration of 10,058 error events for the same device observed by the monitor in Grenoble (**Figure 13**, left, device 2), indicating that cessation of operations led to a reduction in energy consumption.

These findings demonstrate that the Disaster-FD system was effective in detecting this failure using the high incidence of errors and the significant reduction in power consumption as reliable indicators, allowing early identification of the device's unavailability issue.

In Lille, the same equipment was classified as device 1 and recorded 10,064 errors (**Figure 13**, right), a value very simi-

Table 8. Processes q_i and impact factors for Strasbourg.

Region	Set	Processes
Strasbourg	S_1	$\langle q_0, 10 \rangle, \dots, \langle q_{16}, 10 \rangle$
Grenoble	S_2	$\langle q_{17}, 20 \rangle, \langle q_{18}, 20 \rangle, \langle q_{19}, 20 \rangle, \langle q_{20}, 60 \rangle$
Lille	S_3	$\langle q_{21}, 20 \rangle, \langle q_{22}, 20 \rangle, \langle q_{23}, 20 \rangle, \langle q_{24}, 60 \rangle$
Saclay	S_4	$\langle q_{25}, 20 \rangle, \langle q_{26}, 20 \rangle, \langle q_{27}, 20 \rangle, \langle q_{28}, 60 \rangle$

lar to that observed in Grenoble. Thus, despite the minimal variation in error counts, the data collected in these regions confirm the critical condition of the device.

6.3.2 Network Statistics in Lille

This section presents the main results for the monitor in the Lille region using the Disaster-FD system, which simultaneously monitors its own network and neighboring regions (Grenoble, Saclay, and Strasbourg), as shown in **Table 6**.

Figure 14 shows the statistics obtained in Lille, highlighting the values of Probability of Accuracy (PA), Trust Level and Threshold. Throughout the experiment, Lille (top right) exhibited persistent instability, with Trust Level values frequently below the established threshold (100), reaching critical values close to 0 at certain times. These conditions indicate a frequent occurrence of failures or unavailability of monitored IoT devices.

This instability was also reflected in the significant PA fluctuation during the experiment. Since the beginning of the experiment, high values close to 1.0 were observed, indicating that the local monitor correctly interpreted the region as not trusted. A significant event occurred at 16:41, when some local devices in Lille began to operate, resulting in an abrupt drop in the PA value to 0 and a corresponding increase in the Trust Level to about 60. This scenario highlighted a long period of false positives. After this event, as the devices continue to malfunction, the PA value begins to increase again.

In the federated monitoring performed by Lille on Strasbourg (bottom right), a critical event occurred at 16:41, when Lille classified Strasbourg as an untrustworthy network due to the significant drop in the Trust Level of that region, which reached a minimum value of 20. This drop began around 16:11, with a progressive decline until the critical point.

Interestingly, this instability was not detected by the monitors located in Grenoble, Saclay, or even locally in Strasbourg, to the extent of considering the Strasbourg network as untrustworthy. Section 6.3.5 presents a comprehensive discussion of this particular situation.

The remote monitoring of Grenoble (top left) identified some instability, but it was not enough to cause the Trust Level to go below the threshold, except in the last hour of the experiment, when every region correctly observed an abrupt reduction of connectivity related to Grenoble.

Concerning the Saclay region (bottom left), we can observe that the network remained stable during the experiment, which was similarly noted by the other regions. These observations confirm the effectiveness of Disaster-FD in the rapid identification and validation of failures through the federated approach, simultaneously monitoring multiple regions and performing cross-analysis of identified critical events.

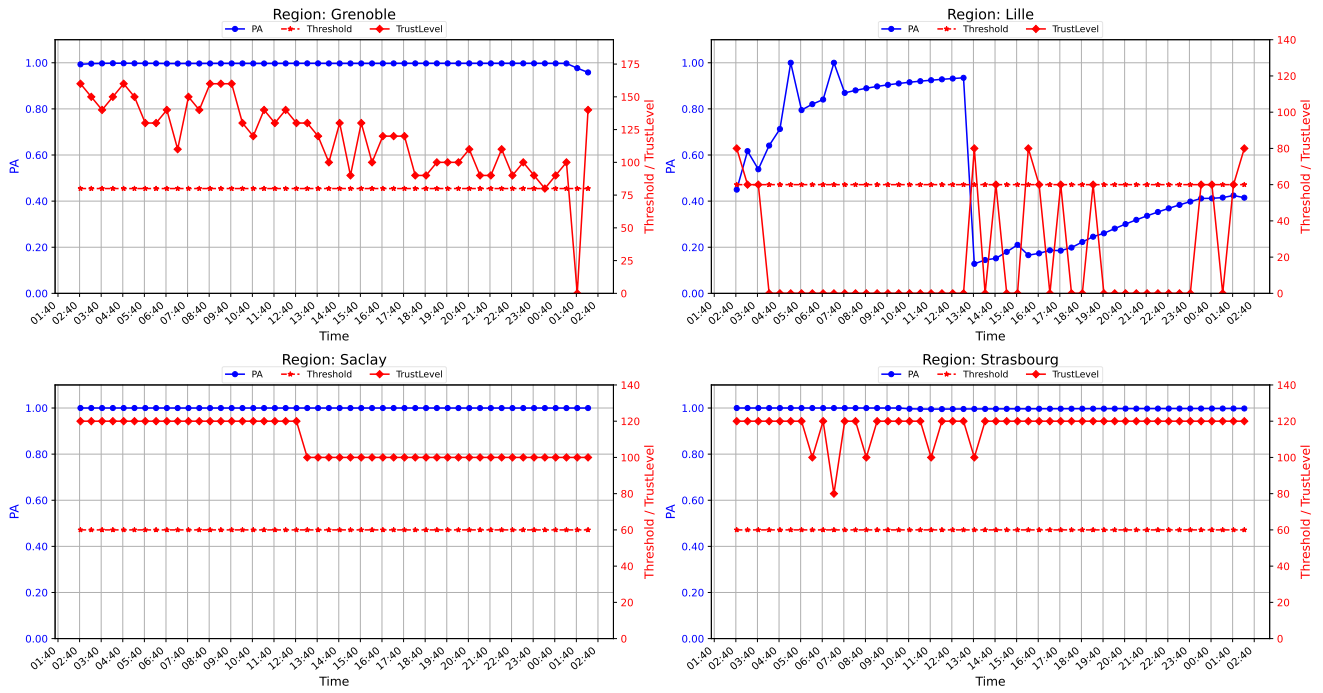


Figure 11. Trust level and PA for each monitored region in Grenoble.

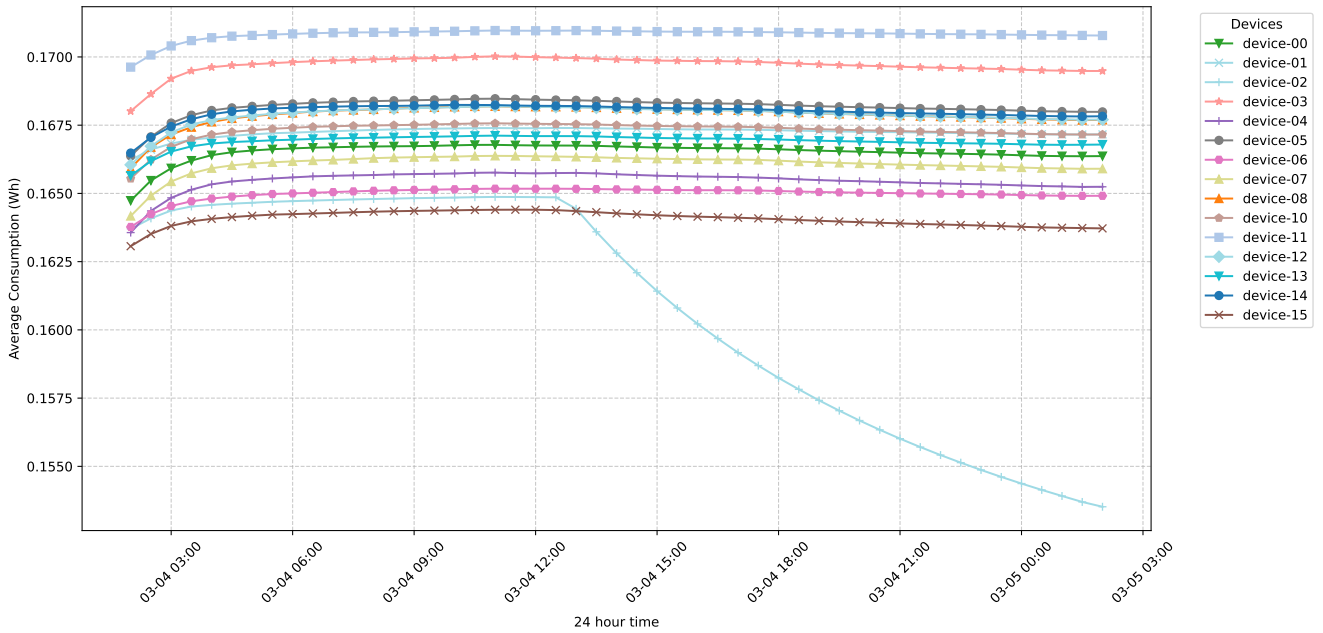


Figure 12. Power consumption for devices in Grenoble.

6.3.3 Network Statistics in Saclay

This section presents the main results of the monitoring in the Saclay region, carried out using the Disaster-FD system, which simultaneously monitors its own network and neighboring regions (Grenoble, Lille, and Strasbourg), as shown in Table 7.

Figure 15 displays the statistics obtained during the monitoring in Saclay through a graph that consolidates the results of the monitored regions. In the graph, the dashed lines represent the network threshold, while the solid lines indicate the trust level of the monitored regions. It is observed that

the Saclay region (bottom left) maintained network stability throughout the experiment, remaining above the safety threshold defined at 50, and was thus considered a reliable and stable network, a fact corroborated by the neighboring regions that also monitored Saclay.

It is further noted that the Grenoble (top left) region became unreliable at 01:41, a fact similar to that identified by the other regions in their respective graphs, demonstrating the efficiency of the Disaster-FD system in its federated monitoring.

In turn, the Lille region (top right) exhibited instability

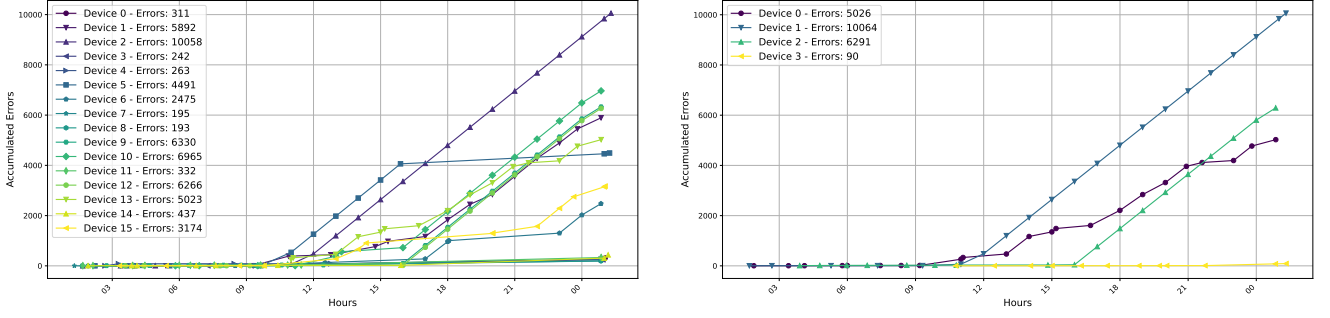


Figure 13. Accumulated errors per device in Grenoble (left) and Grenoble neighboring Lille (right).

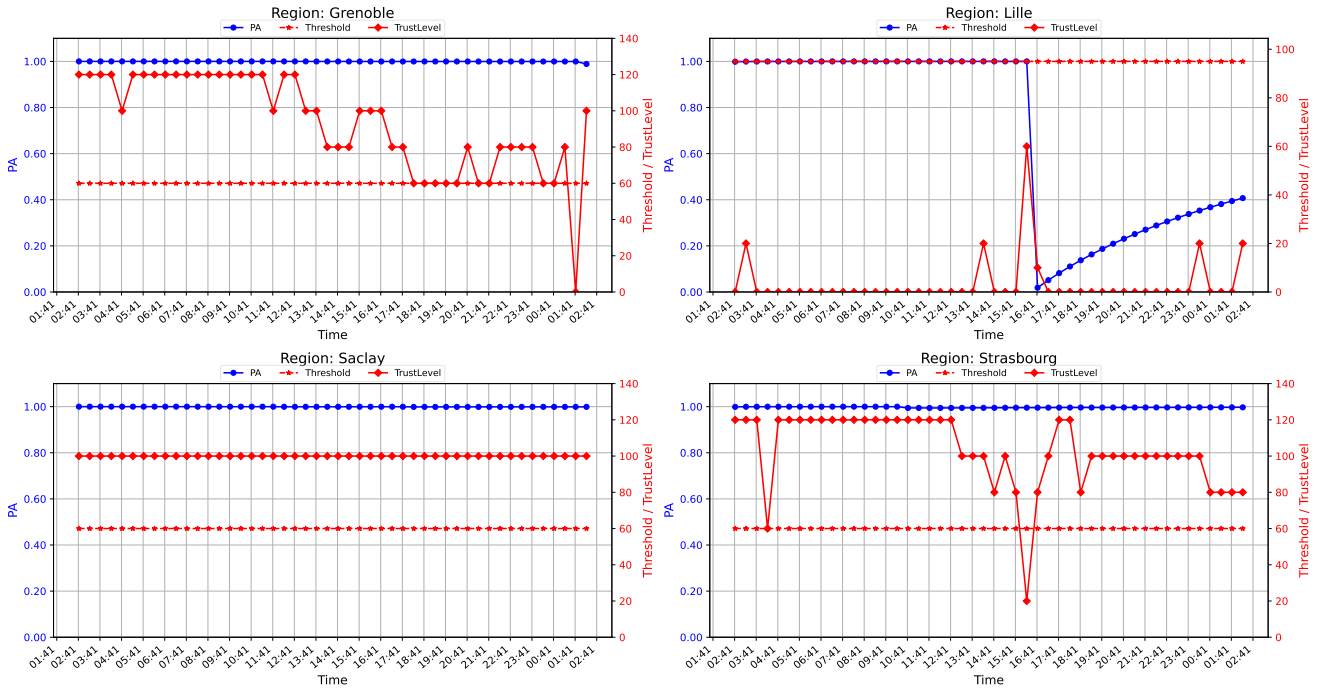


Figure 14. Trust level and PA for each monitored region in Lille.

during most of the 24-hour monitoring period. This behavior validates the graph of the Lille region itself, which also showed instability. The PA curve for Lille showed low accuracy for that region, while the other regions registered a PA of 1.0, corroborating the results obtained in the local monitoring conducted by the Disaster-FD system.

We also observed that errors detected in local devices in the Saclay region, as well as in devices monitored in the neighboring regions, occurred simultaneously and reached values similar to those recorded in Saclay.

The Strasbourg region (bottom right), monitored in Saclay, was also considered stable, with the trust level varying between 60 and 120, since the safety threshold for the Strasbourg network to be considered stable in Saclay is 60, as indicated by the corresponding graph in Figure 15.

6.3.4 Network Statistics in Strasbourg

This section describes the results for the Strasbourg region, conducted using the Disaster-FD system, which simultaneously monitors its network and neighboring regions (Grenoble, Lille, and Saclay), as shown in Table 8.

ble, Lille, and Saclay), as shown in Table 8.

Figure 16 displays the statistics obtained in Strasbourg, compiled into graphs that consolidate data from the monitored regions. In this graph, the dashed red lines represent the network threshold, while the continuous red lines indicate the trust level.

The Grenoble region (top left) became unreliable at 01:41, as observed in the graphs of all regions, further evidencing the efficiency of the Disaster-FD system in federated monitoring.

Meanwhile, the Lille region (top right) showed instability during most of the 24 hours, although it registered moments of improvement in trust level, such as between 13:11 and 14:41 and between 00:41 and 01:41 when the experiment ended. All regions observed this instability in the Lille network in their monitoring.

The Saclay region, monitored from Strasbourg, proved to be stable, corroborating the results of the other monitoring efforts and reinforcing its reliability.

We observe that Strasbourg (bottom right) maintained network stability for most of the experiment, remaining above

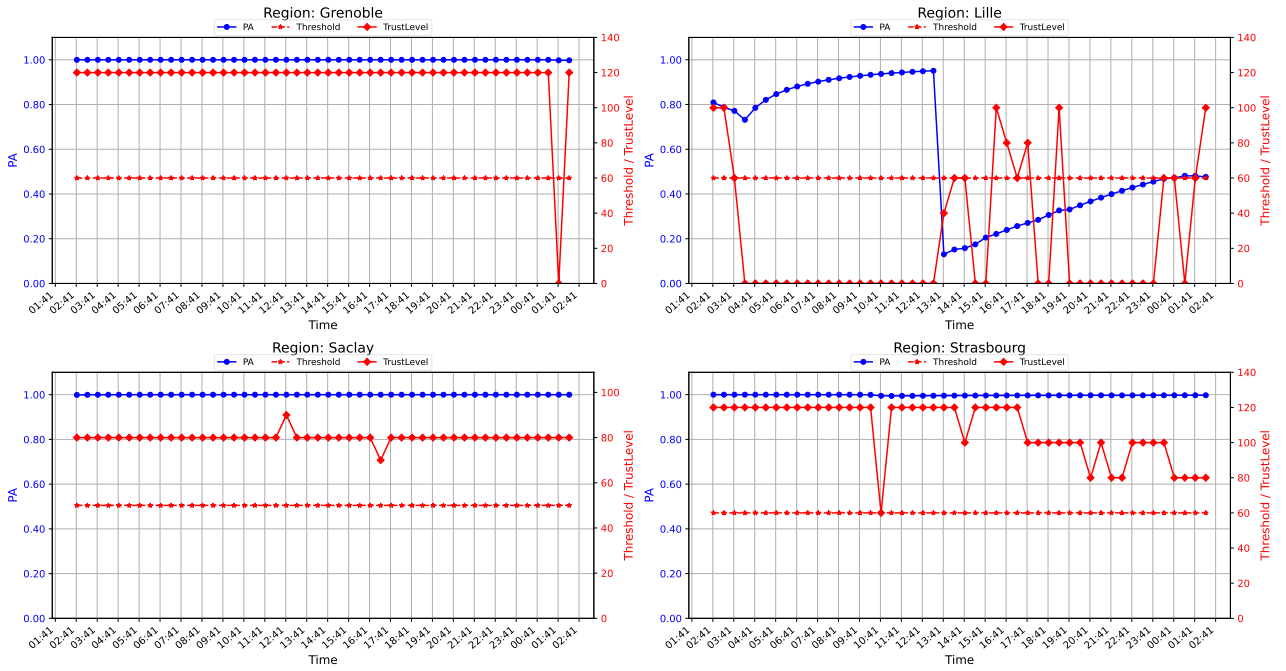


Figure 15. Trust level and PA for each monitored region in Saclay.

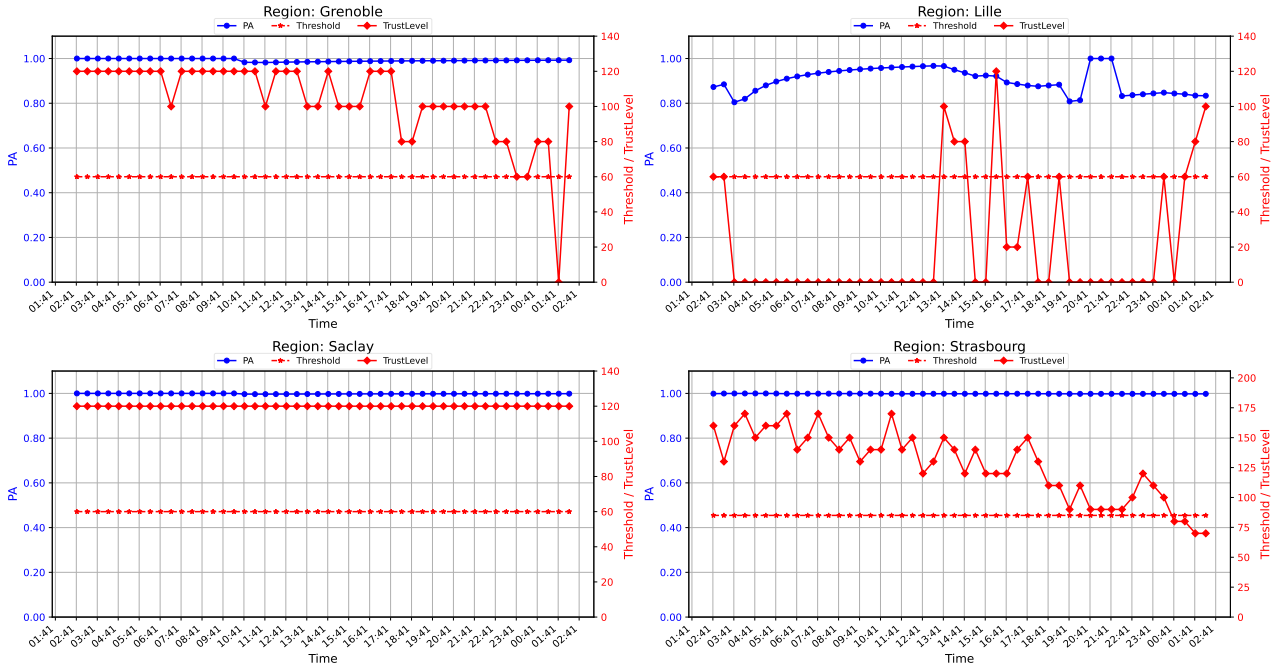


Figure 16. Trust level and PA for each monitored region in Strasbourg.

the defined safety threshold of 85. However, near the end of the experiment, the trust level dropped to 70, signaling a condition of unreliability, a situation also identified by the Saclay region, which monitored Strasbourg and detected failures in Strasbourg's remotely monitored IoT devices.

Thus, Disaster-FD demonstrated that some IoT sensors in Strasbourg exhibited failures, which would likely result in an unreliable condition, as shown in the local statistics of the Strasbourg region.

6.3.5 Importance of federated monitoring

Figure 17 shows the Trust Level and Threshold for every region for the Strasbourg region. The dashed black line represents the threshold for the remote regions of Grenoble, Saclay, and Lille. The red dashed and solid lines represent the threshold and Trust Level, respectively, for the local monitoring in Strasbourg. The remaining curves show the Trust Level as measured in Grenoble (blue), Lille (green), and Saclay (yellow).

In this scenario, we emphasize an interesting situation that

corroborates the importance of federated monitoring. We observe that between 15:42 and 16:42 Strasbourg locally presented a real reduction in its Trust Level (120), indicating a reduction in the number of working devices. During this interval, specific devices monitored by Lille in Strasbourg exhibited intermittent behavior similar to that observed locally in Strasbourg. The remaining regions were not affected by this “hiccup”.

Considering that the Trust Level for 3 out of 4 regions remained above the threshold, this region should be considered trusted, despite the false suspicion of the monitor in Lille.

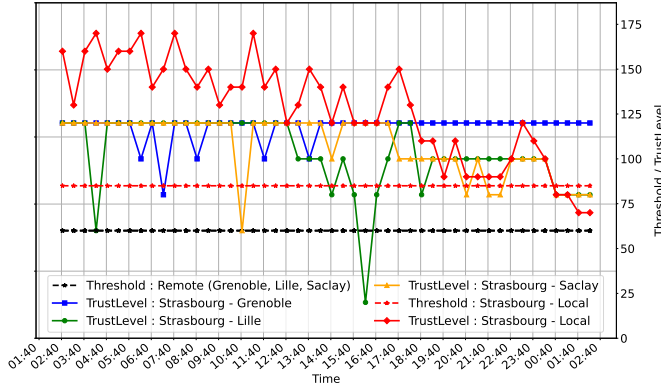


Figure 17. Trust levels for Strasbourg.

7 Related Work

The field of failure detection in Internet of Things (IoT) networks has seen significant advances, with several notable contributions aimed at improving the accuracy and efficiency of such systems. This section reviews the literature on failure detection technologies, focusing on their methodologies, strengths, and limitations in the context of disaster-prone environments.

Impact-FD [Rossetto *et al.*, 2018] is a solution that proposes a new failure detector called Impact FD, which provides an output expressing the confidence of the failure detector in the system (or set of processes) as a whole. The confidence is configured by the impact factor, allowing the user to define the importance of each node within an acceptable margin of failure. Additionally, some flexibility properties are defined, which characterize the ability of Impact FD to tolerate a certain margin of failures or suspicions. Disaster-FD extends Impact-FD, focusing on failure detection and network reliability assessment in IoT environments, with an emphasis on real-time monitoring. Additionally, Disaster-FD performs federated monitoring across different regions and considers additional aspects such as device energy consumption.

Medley [Yang *et al.*, 2019] is a decentralized solution for failure detection in IoT operating in ad-hoc networks, employing spatial selection to send ping messages and prioritizing nearby nodes. It uses multiple votes from various devices to assess the condition of a node to reduce false positives. The SWIM protocol, integrated with Medley, ensures scalable and tolerant failure detection. However, Disaster-FD focuses on real-time monitoring and analysis of IoT net-

works, especially in disaster situations, prioritizing system reliability assessment.

Stab-FD [Sens *et al.*, 2024] is a solution that proposes a failure detector for distributed systems, especially suitable for wide area networks (WAN), which dynamically adjusts within a safety margin to adapt to variations in the quality of communication links. Additionally, Stab-FD has a cooperative version, in which nodes exchange information about link stability and the list of suspected nodes [Sens *et al.*, 2024]. Disaster-FD stands out in IoT environments, especially in disaster scenarios, using an algorithm to assess network reliability. The techniques proposed in Stab-FD for the adaptation and reduction of false positives are orthogonal and could be applied to Disaster-FD.

SWIM, the Scalable Weakly-consistent Infection-style Membership protocol, introduced by [Das *et al.*, 2002], is designed for failure detection in large-scale distributed systems. SWIM employs a combination of pinging and indirect probing to detect node failures. Since SWIM presents very good scalability and robustness in large networks, Disaster-FD integrates SWIM’s scalable concepts within the context of IoT networks, ensuring that the system remains efficient and reliable even in the face of large-scale disasters.

8 Conclusion

The presented study details the development and evaluation of Disaster-FD, a failure detector designed for disaster-prone environments, with a focus on real-time monitoring of IoT networks. Inspired by Impact-FD [Rossetto *et al.*, 2018], Disaster-FD introduces features such as federated monitoring and continuous assessment of network reliability, demonstrating robustness in both simpler scenarios and large-scale deployments.

In the initial scenario, involving two regions, each monitor supervised local devices as well as devices and monitor from the neighboring region. In this configuration, the error patterns of the local devices were similar to those of the remotely monitored ones, highlighting the effectiveness of cross-monitoring for early fault detection. Comparison with Impact-FD showed that Disaster-FD adapts more efficiently to IoT network instabilities, resulting in less variability in confidence levels and a reduction in false positives, largely due to the use of a message identifier to estimate the arrival time of heartbeat messages.

Extending the experiments to a federated deployment across four regions – Grenoble, Lille, Saclay, and Strasbourg – the scalability of Disaster-FD was confirmed by a total of 110 devices distributed supervisions. In this scenario, each monitor not only observed local devices, but also monitored three other devices and the monitor of the other regions. A crucial finding was that, regardless of scale, all monitors detected persistent instability in Grenoble’s network.

Statistical data reinforced this observation. Monitors reported that around 01:41, the Grenoble Trust Level fell below the reliability threshold evidenced by a low Probability of Accuracy (PA) value while Saclay and Strasbourg maintained consistently high PA values (close to 1.0), and Lille showed greater variability, reflecting its instability.

The analysis of energy consumption also revealed characteristic behaviors. For example, a device in Grenoble exhibited reduced energy consumption and an increased number of errors, serving as an early indicator of operational failure. This evidence demonstrates that using metrics such as PA and Trust Level assessment enables more precise fault detection, regardless of the number of monitored devices.

In summary, Disaster-FD presents itself as a promising tool for monitoring IoT networks in critical environments, leveraging federated monitoring to offer greater resilience and consistency, essential elements for early fault detection and rapid response to natural disasters.

Future work includes analysis of fault frequency to identify trends and determine the need for preventive maintenance; the implementation of fault-based notifications to anticipate potential natural disasters; the application of the *ComputeMargin* function, inspired by the Stab-FD [Sens et al., 2024], to dynamically adjust monitoring timers based on communication link stability; and a more systematic combination of observations from multiple monitors to improve the reliability of the results.

Declarations

Acknowledgements

The authors thank the anonymous reviewers for their valuable feedback.

Funding

This work was partially supported by the Brazilian Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) under grants 88881.368742/2019-01 and 88881.985518/2024-01. Luciana Arantes and Pierre Sens acknowledge the support of the French Agence Nationale de la Recherche (ANR), under grant ANR-22-CE25-0008-01 (SkyData project).

Authors' Contributions

Abadio implemented Disaster-FD, performed the experiments, and, along with Paulo, wrote most of the manuscript. All authors read, reviewed, and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

The datasets generated and/or analyzed during the current study, as well as Disaster-FD source code, are publicly available¹.

References

Adjih, C., Baccelli, E., Fleury, E., Harter, G., Mitton, N., Noel, T., Pissard-Gibollet, R., Saint-Marcel, F., Schreiner, G., Vandaele, J., et al. (2015). FIT IoT-LAB: A large scale

open experimental IoT testbed. In *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, pages 459–464. IEEE. DOI: 10.1109/wf-iot.2015.7389098.

Aguilera, M. K., Delporte-Gallet, C., Fauconnier, H., and Toueg, S. (2004). Communication-efficient leader election and consensus with limited link synchrony. In *Proceedings of the twenty-third annual ACM symposium on Principles of distributed computing*, pages 328–337. DOI: 10.1145/1011767.1011816.

Atzori, L., Iera, A., and Morabito, G. (2010). The internet of things: A survey. *Computer networks*, 54(15):2787–2805. DOI: 10.1016/j.comnet.2010.05.010.

Barborak, M., Dahbura, A., and Malek, M. (1993). The consensus problem in fault-tolerant computing. *ACM Computing Surveys (CSur)*, 25(2):171–220. DOI: 10.1145/152610.152612.

Chandra, T. D., Hadzilacos, V., and Toueg, S. (1996). The weakest failure detector for solving consensus. *Journal of the ACM (JACM)*, 43(4):685–722. DOI: 10.1145/234533.234549.

Chandra, T. D. and Toueg, S. (1996). Unreliable failure detectors for reliable distributed systems. *Journal of the ACM (JACM)*, 43(2):225–267. DOI: 10.1145/226643.226647.

Chen, W., Toueg, S., and Aguilera, M. K. (2002). On the quality of service of failure detectors. *IEEE Transactions on computers*, 51(5):561–580. DOI: 10.1109/icdsn.2000.857535.

Cristian, F. and Fetzer, C. (1999). The timed asynchronous distributed system model. *IEEE Transactions on Parallel and Distributed Systems*, 10(6):642–657. DOI: 10.1109/71.774912.

Das, A., Gupta, I., and Motivala, A. (2002). Swim: Scalable weakly-consistent infection-style process group membership protocol. In *Proceedings International Conference on Dependable Systems and Networks*, pages 303–312. IEEE. DOI: 10.1109/DSN.2002.1028914.

Fischer, M. J., Lynch, N. A., and Paterson, M. S. (1985). Impossibility of distributed consensus with one faulty process. *Journal of the ACM (JACM)*, 32(2):374–382. DOI: 10.1145/3149.214121.

Janoneda, L. (2022). A cada desastre natural no brasil, em média, 3,4 mil pessoas são afetadas. Available at: <https://www.cnnbrasil.com.br/nacional/a-cada-desastre-natural-no-brasil-em-media-34-mil-pessoas-sao-afetadas>. Accessed in 2022-01-09.

Kovatsch, M., Lanter, M., and Shelby, Z. (2014). Californium: Scalable cloud services for the internet of things with coap. In *2014 International Conference on the Internet of Things (IOT)*, pages 1–6. IEEE. DOI: 10.1109/iot.2014.7030106.

Leslie, L. (1998). The part-time parliament. *ACM Trans. on Computer Systems*, 16:133–169. DOI: 10.1145/3335772.3335939.

Rossetto, A. G. d. M., Geyer, C. F., Arantes, L., and Sens, P. (2018). Impact fd: An unreliable failure detector based on process relevance and confidence in the system. *The Computer Journal*, 61(10):1557–1576. DOI:

¹<https://github.com/abadiopaulo/disasterFd.git>

- 10.1093/comjnl/bxy041.
- Sens, P., Arantes, L., Rosseto, A. G. D. M., and Marin, O. (2024). Stab-fd: A cooperative and adaptive failure detector for wide area networks. *Journal of Parallel and Distributed Computing*, 186. DOI: 10.1016/j.jpdc.2023.104803.
- Silva, A. d. P., Rosseto, A. G. d. M., Sens, P., Arantes, L., Pasquini, R., and Coelho, P. (2024). Disaster-fd: A failure detector for disaster-prone environments. In *Proceedings of the 13th Latin-American Symposium on Dependable and Secure Computing*, pages 200–209. DOI: 10.1145/3697090.3697096.
- Verissimo, P. and Rodrigues, L. (2012). *Distributed Systems for System Architects*, volume 1. Springer Science & Business Media. DOI: <https://doi.org/10.1007/978-1-4615-1663-7>.
- Yang, R., Zhu, S., Li, Y., and Gupta, I. (2019). Medley: A novel distributed failure detector for IoT networks. In *Proceedings of the 20th International Middleware Conference*, pages 319–331. DOI: 10.1145/3361525.33615567.