AIRBORNE FOG COMPUTING FRAMEWORK: INSTANT PRIMERS AND DESIGN CHOICES

Antonio Santos da Silva[1], Paulo Mendes[2], Denis Rosário[3], Eduardo Cerqueira[3], Edison Pignaton de Freitas[1]
[1]Institute of Informatics, Federal University of Rio Grande do Sul, Porto Alegre - Brazil, 91.501-970, [2]Airbus, Willy-Messerschmitt-Strasse 1 Taufkirchen - Germany, 82024, [3]Federal University of Pará (UFPA), Belém - Brazil, 66.075-110

Corresponding author: Antonio Santos da Silva, (aassilva@inf.ufrgs.br)

Abstract – *This article focuses on the challenge of providing services and applications by a swarm of mobile flying devices, which may belong to the same tenant or not. Starting by introducing the concepts of mobile edge computing and fog computing, this work shows how the latter can be leveraged to create a novel framework of airborne fog computing. This article defines the applicability, design principles, and challenges inherent to such a framework, starting by showing how the technical challenges can be tackled based on the combination of three different networking paradigms: opportunistic networking, information-centric networking, and software-defined networking. This article shows how these three networking paradigms can be combined to define an airborne fog computing framework based on the notion of in-network computing dynamically programming the data plane to support the execution of services and the integration of the paradigms, in which services are defined as chains of computational functions that are dynamically installed and executed in different flying devices.*

Keywords – Fog computing, information-centric networking, mobile edge computing, opportunistic networking, software-defined networking

1. INTRODUCTION

The growth of Internet applications and services, especially mobile data traffic, is having a significant impact on the Internet's architecture. In addition, the development of latency-critical services, such as video stream analysis, augmented reality, and autonomous driving, requires extensive real-time computing [1]. Therefore, Mobile Edge Computing (MEC) [2] has been introduced to define a new network paradigm in which computing services are deployed at the network edge (e.g. mobile operator base station) to improve the computing performance required by mobile users. Based on MEC, the mobile user loads his processing task to available servers, which will allocate the resources needed to perform that task. Although many approaches are proposed to solve some technical limitations, the main drawback of MEC is that the locations of MEC servers are generally fixed and cannot be changed according to the needs of mobile users.

There are several approaches for bringing computing to the edge and fixed infrastructures [3, 4, 5]. On the other hand, there are some approaches to providing connectivity using drones [6, 7, 8]. However, there are no approaches to providing cloud computing on drones. These limitations have given rise to several works aiming at reducing the delay. In the work by Zarandi and Tabassum [3], the problem is divided into two sub-problems: (i) download decision-making and (ii) pooled computing resources. Following the same line, Feng *et. al.* [4] investigate the problem of task partitioning and user association in a MEC system, aiming to minimize the average latency of all users. Given the limitations of existing MEC techniques, which struggle in wireless network scenarios with constrained infrastructure, such as in disaster response, emergency relief, military operations, or rural environments, the concept of MEC has been expanded to include fog computing. This extension serves to enhance MEC's applicability and effectiveness in managing diverse and resource-limited settings by leveraging decentralized computing resources [5], in which the computational functions can run not only at the edge of an infra-structured network (e.g. mobile provider) but also using computing units co-located with the data-generating devices.

To enhance the performance of routers, switches, and gateways in running services, it is essential to seek solutions that optimize the utilization of computational resources. One avenue of exploration is using λ functions, as referenced by Kaur and Mittal [9] and Mekbungwan *et. al.* [10], which offer a lightweight approach. However, it's worth noting that λ functions have certain operational limitations and present increased developer complexity. Notably, a single λ function cannot access multiple services. However, they are all based on a server architecture, meaning that developers need to be aware of the application to be developed and the resources needed for their execution. Finally, the deployment of server-based solutions is subject to cost and power issues [11], and CPU power and memory limitations often characterize the increased latencies of edge devices.

Rani *et. al.'s* [12] analysis and assimilation of existing approaches to storage techniques in fog computing are performed. The study suggests the classification of temporary storage for end devices at the network edge and fog computing processing. Haghi *et. al.* [13] provides an analysis focused on the Quality of Service (QoS) approach in cloud computing. Finally, Zahmatkesh and Al-Turjman [14] use drones in conjunction with various Artificial Intelligence (AI) and Machine Learning (ML) techniques to enable fog computing technologies for sustainable smart cities in IoT environments. While the concept of fog computing is best suited to support services in dynamic wireless networks [15], such as airborne networks, several challenges need to be solved, such as offloading decisions, resource allocation, and mobility management.

MEC and fog computing introduce overhead related to routing requests to proxies or controllers and routing the computation results back to the requested entities. This means that the links connecting these entities become bottlenecks, mainly because the network only forwards packets. At the same time, decisions are made at the application layer (e.g. proxies). In this context, Information-Centric Networks (ICNs) provide an alternative to the traditional host-centric IP architecture, making it a perfect candidate for distributed provider-independent computing, such as fog and edge computing [16, 17, 18, 19]. In the Named Data Networking (NDN), a specific type of ICN paradigm, computing can be performed remotely by any node that supports the required computing resources, making the client and the network agnostic to the list of providers and their corresponding IP mapping [17, 20].

Although the ICN paradigm is best suited for working in dynamic wireless networks, it was defined with networks with fixed infrastructures in mind, so it would be necessary to leverage the ICN paradigm with the notion of opportunistic networking, which is able to exploit the mobility of flying devices as an opportunity, rather than a challenge, to improve routing performance. This article aims to discuss the feasibility of developing an airborne fog computing platform based on a paradigm that leverages NDN and opportunistic networks to offload computing tasks transparently and efficiently.

The contributions of the work described are as follows:

- **Introduction of airborne fog computing:** This article introduces the novel concept of airborne fog computing, a framework that considers swarms of mobile flying devices for providing services and applications, potentially revolutionizing how computing resources are deployed and managed in dynamic environments.

- **Combination of networking paradigms:** It highlights how integrating three distinct networking paradigms, opportunistic networking, information-centric networking, and software-defined networking, can address technical challenges within the framework. This combination promotes a more efficient, dynamic method of in-network computing where services are defined as chains of computational functions dynamically installed and executed on data planes across various devices.

- **Focus on Mobility and dynamic networks:** The work focuses on enhancing the performance of mobile and dynamic networks by leveraging the mobility of devices as an opportunity to improve routing performance and resource utilization rather than as a challenge.

- **Exploration of NDN and opportunistic networks:** The article discusses the feasibility of leveraging NDN and opportunistic networks to efficiently offload computing tasks within the airborne fog computing platform, offering a detailed analysis of how these technologies can transcend traditional networking limitations.

- **Resource efficiency and management:** It addresses how airborne fog computing can maintain essential services even amid connectivity failures and efficiently scales on constrained

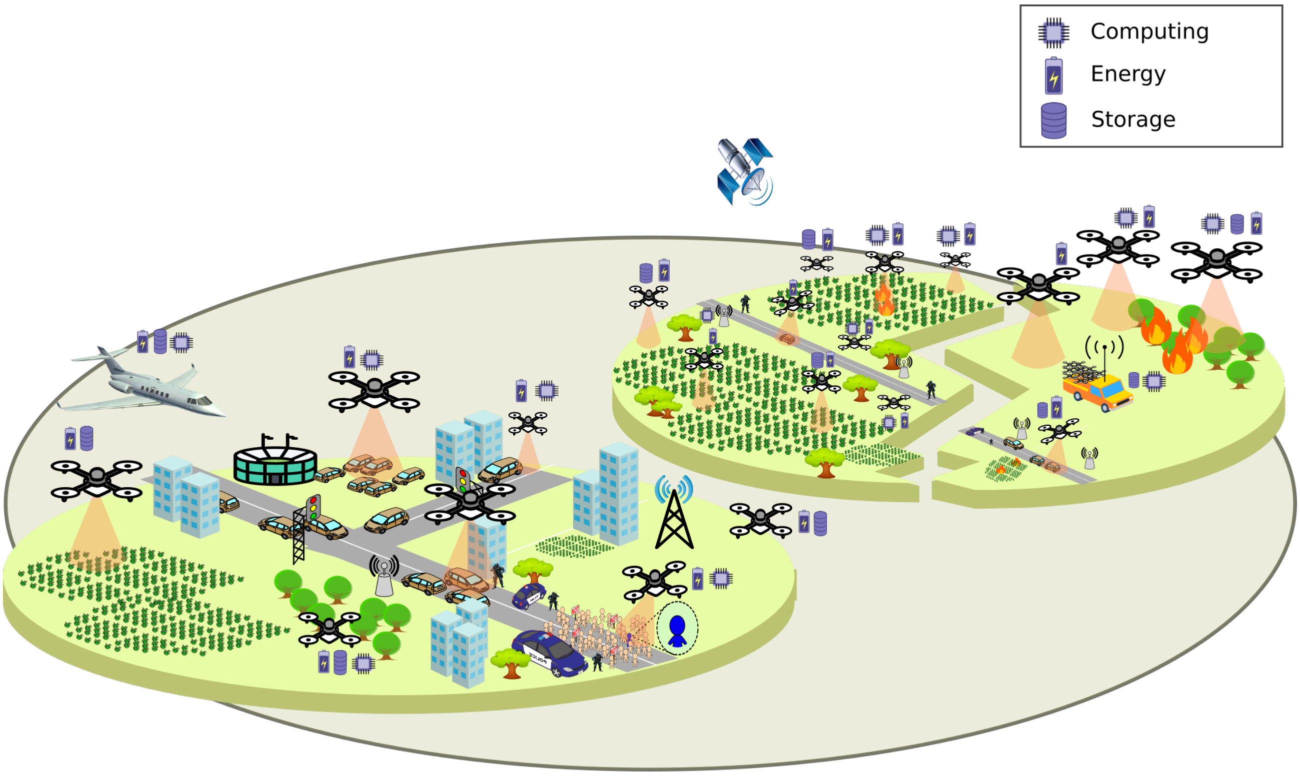

Figure 1 – Overview of airborne fog computing framework.

computing devices through cooperative service provision among all devices.

The remainder of this article is organized as follows. First, the design options are identified, and the open issues to enable the secure, efficient, and continuous operation of airborne fog computing are highlighted. Next, the design options for the airborne fog computing framework are presented. The last two sections end with a description of a use case and final considerations.

2. AIRBORNE FOG COMPUTING

This article presents the concept of airborne fog computing, which aims to transform a network of flying devices as illustrated in Fig. 1 (e.g. drones, aircraft, satellites) into a flying ad hoc cloud. Based on the airborne fog computing framework, flying devices collect resources from untrusted and sporadically available infrastructures. These flying devices can be considered as unreliable, as the primary function of the devices involved does not include providing computational and storage resources, and they are typically connected through intermittently available links. While the goal of providing cloud services by collecting resources from a set of untrusted devices has similarities with grid computing, the airborne fog computing paradigm has several differences to resource management, trust, continuous operation, and interference. Unlike cloud or grid computing, an airborne fog computing framework must maintain service availability in the presence of device unavailability and intermittent wireless link behavior to ensure continuity of computing tasks.

With an IP-based airborne fog computing framework, users and applications would likely rely on centralized entities such as software-defined network controllers or would need to rely on direct communication between all devices in the fog framework to make any computational assignment decisions. Regardless of their centralized or distributed nature, these solutions would introduce additional overhead to route all the necessary data to the controllers or discover the fog computing devices involved in the described service chain. These characteristics result in IP protocols functioning poorly. ICN, particularly NDN, are well suited for dynamic networks due to their distributed nature, providing an alternative to the host-centric IP architecture, which may make it a suitable candidate for the development of a host-agnostic airborne fog computing framework.

Various types of devices, dispersed throughout the environment, will constitute the physical components of an airborne fog computing framework featuring distributed resources. These devices could potentially be aggregated and collectively utilized to offer services surpassing individual end-user devices' capabilities. Leveraging this assortment of intermittently available resources, encompassing computational power, networking, and storage, necessitates innovative, opportunistic network solutions designed to orchestrate, compose, and manage various resources across numerous devices. In this context, the design of an airborne fog computing framework can be effectively achieved by integrating two innovative paradigms: information-centric network and opportunistic network.

2.1 NDN in a nutshell

Unlike IP networks that use IP addresses to control how packets are forwarded, the fundamental idea of NDN is to retrieve named blocks of data from any device that can provide them [21]. In the NDN, routers are equipped with a Content Store (CS), a Pending Interest Table (PIT), and a Forwarding Information Base (FIB). The FIB is populated using a routing protocol, which needs to be named Hosted in the case of NDN. Any device that receives a packet of interest for a data object performs a CS lookup on the data name. If the data is not available in the CS, the device performs a query on its PIT. If the PIT search is successful, the router adds the interface of the incoming interest packet to the PIT entry (interest aggregation) and discards the interest packet. If no PIT match is found, the router creates a new PIT entry for the new interest packet and forwards the interest using FIB to an upstream router in the data source(s) direction. The data objects are forwarded in the reverse path of the interest packets back to the consumer.

Within the Named Data Networking (NDN) framework, computational tasks can be conducted remotely on any device that is part of the airborne fog computing platform, where the airborne fog computing platform has the necessary computational resources to support the services offered, making the users and network agnostic to the list of data providers, as well as the providers of computational and storage resources. Specifically, this approach considers Named Function Networking (NFN) and Named-Function as a Service (NFaaS) [20]. NFN uses NDN function nomenclature to locate remote computing resources by supporting name resolution for functions and data. On the other hand, NFaaS is a framework proposed to extend NFN to support dynamic code execution using lightweight virtual machines. Although these two solutions can use function naming to locate remote computational resources and perform network computation, they face some technical challenges, such as name aggregation, efficient resource discovery, computation reuse, and mobility management.

Resource discovery is one of the main challenges of edge computing in general and, in particular, dynamic networks that target the airborne fog computing framework. In particular, the devices in an airborne fog computing framework often change these capabilities, namely the available computing resources. The availability of computational and storage resources must be discovered by any device to make efficient task-routing decisions. In NDN, consumers are responsible for requesting data again after the timeout. In the context of the airborne fog computing framework, computational tasks can be resource-intensive in terms of resources used, and execution times can be challenging to estimate, resulting in longer timeouts.

One solution to this mobility management problem may involve having a mobile device monitor intermittent wireless link conditions relative to its nearest neighbor and having the mobile device resend packets of interest once wireless connectivity resumes. While this approach can work in a mobile network scenario in which mobile devices are connected to a fixed wireless infrastructure, it faces severe challenges in ad hoc wireless scenarios in which all devices are mobile. To help solve this problem, this work proposes combining the NDN routing plan, more specifically, NFN, with an opportunistic network approach.

2.2 Opportunistic networking

NDN faces several challenges when operating in wireless ad hoc networks, such as those that target the airborne fog computing framework. In this type of network, events such as long disconnects and network partitions are pretty standard; at best, network stability still allows the use of ad hoc routing protocols to find end-to-end paths. However, in more challenging scenarios, no simultaneous multi-hop path can be guaranteed between any pair of devices at any time. This scenario brings several challenges for coordinating computational functions in an airborne fog computing framework. However, by using opportunistic networking schemes, the mobility of NDN devices in a dynamic network can be seen as a communication opportunity rather than a challenge, allowing mobile devices to communicate even if an end-to-end route connecting them never exists [22]. This is possible because the opportunistic network allows routes to be built dynamically, based on protocols such as dLife [23] or Dabber [24], the latter of which is name-based and tailored to be used in conjunction with an NDN data plane.

However, with opportunistic networking, interest packets can be forwarded using any possible neighboring contacts, if there is a probability of bringing such packets closer to their destination: devices containing a copy of the requested data object or the data producer. However, the use of opportunistic networks to support the airborne fog computing framework brings the need to make fundamental changes to the NDN, since the usual breadcrumb approach to data forwarding, using PIT information, may not be feasible due to the probability that wireless links towards the data consumer may not be available. A simple adaptation of the NDN with an opportunistic forwarding mechanism may involve creating opportunistic queues to soon store the interest and data packets while an opportunity to transmit is not present and while the lifetime (validity) of the data does not expire.

Although human social structures are at the core of opportunistic network solutions, the mobility patterns of flying devices, such as missions and monitoring contact times with each other, can be used to adapt opportunistic network solutions to the airborne fog computing framework. Exploiting wireless contact opportunities for communication can be further leveraged in an airborne fog computing framework. When two devices come into contact, it is an opportunity to share and exploit each other's resources and perform computational tasks remotely. This creates the possibility of developing the airborne fog computing framework in challenging scenarios where wireless connectivity is intermittent.

By using opportunistic networking, the airborne fog computing framework exploits a multitude of heterogeneous resources that can be used to implement the required computational function. The main point is how the proposed framework allows these resources to be accessed and combined securely and optimally. Specifically, the goal of an airborne fog computing framework is to enable users to use the resources available on their device and opportunistically exploit a composition of other resources in the wireless environment reliably and securely. This ability to operate in very dynamic networks with unstable topologies makes the proposed framework different from other concepts, such as service-oriented computing.

2.3 Applicability scenarios

Future smart networks envision the use of the airborne fog computing framework in various applicability cases, spanning different aircraft types, including low and high-altitude drones and spacecraft. For example, drone swarms can be deployed in disaster scenarios or even for short-term, large-scale outdoor events (e.g. music events and sporting). Another example is the deployment of computationally capable High Altitude Platform Stations (HAPS), temporarily flying over an area to provide video-on-demand services to a group of people for a few days or collecting information from remote sensors.

The airborne fog computing framework enables drones to function as a distributed mobile fog, expanding their utility across various sectors where real-time data processing and decentralized operations are crucial. Other examples include, precision agriculture, drones equipped with sensors collect and process data on crop health and soil moisture levels directly in the field, allowing for immediate adjustments to irrigation and pesticide applications. In disaster management, drones can form a mobile fog network to map disaster-stricken areas, process the data on-site, and coordinate search and rescue operations without relying on damaged infrastructure. In infrastructure maintenance, drones gather data on the condition of bridges and railways and process the data in real time to identify and prioritize repair needs, streamlining maintenance workflows. Additionally, in large-scale events, such as concerts or sports events, drones can provide enhanced communication networks and real-time surveillance, processing and distributing data directly where needed to manage crowds and ensure safety. These applications showcase how drones, as part of an airborne fog computing framework, can transform their roles from mere data collectors to powerful nodes of a distributed computing layer, driving efficiency and effectiveness in operations across various domains.

In any of the examples mentioned, there is a need to consider optimizing the parameters of an airborne fog computing platform that would minimize costs and power usage while providing an adequate quality of experience in terms of storage, processing, and network services. An advantage of an airborne fog computing platform is its adaptability and potential scalability to provide more users and content services than expected. The successful applicability of such a platform also depends on its ability to add new computing elements in real time without further configuration.

2.4 Design principles

An airborne fog computing framework is able to allow devices to harvest resources from other non-exclusive and sporadically available devices and expose these resources to computational tasks submitted by users. Computational tasks are submitted to the network itself, which then should be able to schedule different jobs (a computation task may encompass different jobs) to other devices within the flying network. Each device has the capability of monitoring resources and providing state information to other elements of the flying network. In the airborne fog computing framework, it is crucial to monitor networks, processes, and storage resources to facilitate the creation of service chaining: a set of connected devices that offer a particular service.

Many service chains may exist simultaneously over multiple devices to provide a unique service. For instance, users may define an earth observation service that requires video analysis and sound detection in a certain geographic area. To support this service the network may need to have two different service chains holding the necessary resources to execute video analysis and sound detection functions. Each device in the network may either be dedicated to a single service chain or be part of many if multiple jobs are scheduled onto the same device. Grouping devices into service chains makes it easier to schedule computational tasks and monitor the infrastructure. For example, only devices within a specific service chain need to be taken into account when scheduling a computational job associated with that service chain. The resources of service chains may also need to be adjusted based on current computational requirements.

Service chains and devices can be managed centrally or in a distributed fashion. The former approach assumes that all devices can communicate with a central server, similar to the model used in software-defined networking strategies. In this scenario, the server will accept computational jobs, schedule jobs for devices, and maintain the system state. Alternatively, the airborne fog computing framework can be implemented using a distributed architecture, in which the network operates in a self-organized way, utilizing a set of networking protocols that support resource discovery and conflict-free job assignment, for example. In both approaches, the processes and resources supporting the airborne fog computing framework must be concealed from the users.

2.5 Challenges

Several challenges need to be tackled to ensure a successful deployment of airborne fog computing platforms: handling large amounts of data, contextual awareness, reliability, multi-variable optimization, scalability, security, and flexibility. In what concerns data volume, it is expected that drones that provide civilian services will generate large volumes of data (e.g. streaming). Hence, there is a need to deal with such data volumes, including classification, processing, and analysis. There is also the challenge of fusing different data types to serve diverse applications while using airborne and ground infrastructure wisely.

There is also the need for the airborne fog computing framework to be aware of the surrounding context to support adaptations to dynamic situations. This ranges from weather changes to changes in the position of mobile users being serviced. For instance, based on awareness about groups of users, drones can move to positions where servicing them may require minimal energy. Context data may be collected from various sensors (in any of the specific devices) and fed to appropriate devices in the airborne fog computing infrastructure to inform run-time adaptations. A key challenge is the reliability of the airborne fog computing framework, primarily when critical services are handled. Data and computational redundancy is a standard solution. For instance, to meet reliability requirements in airborne devices where instability is always present, data replication is required, which increases networking, computational, and energy costs. Where and how data replication should happen and at what rate should be a design consideration for an airborne fog computing platform.

A fundamental challenge in the mentioned applicability scenarios is optimizing the framework in what concerns several different optimization variables, such as energy consumption, networking costs, and service efficiency. For example, it aims to reach the suitable configuration of the airborne fog computing infrastructure and its interaction with ground servers to provide the best means of handling video data for an earth observation application. Scalability and flexibility are fundamental challenges for the success of a platform, namely to allow service composition to process data arising from different applications, including services for data storage, caching, aggregation, and processing. With such flexibility, different devices might be involved in various stages of a service chain.

Ensuring data security within the airborne fog computing framework poses several critical challenges, particularly due to the high volumes and sensitive nature of the data handled by drones. As drones are increasingly used for civilian services, they generate and stream large amounts of data, necessitating robust mechanisms for data classification, processing, and analysis. Integrating various data types and the extensive use of airborne and ground infrastructure amplifies the risk of data breaches and unauthorized access, making data security a paramount concern.

To address these security challenges, implementing comprehensive data encryption protocols is essential. Encryption should be applied not only during data transmission but also for data at rest. This ensures that even if data interception occurs, the information remains protected against unauthorized access. Furthermore, the framework must incorporate advanced intrusion detection systems to identify and mitigate threats from rogue drones that may attempt to infiltrate the network. Such systems can analyze patterns of data traffic to detect anomalies that may indicate a security breach.

Another critical aspect is the strategic implementation of data and computational redundancy. This enhances the system's reliability in the face of airborne device instability and secures data integrity and availability. However, redundancy increases networking, computational, and energy costs, hence the framework must smartly determine where and how data replication should occur. Decisions regarding the rate and extent of replication must be optimized to balance reliability with resource conservation.

2.6 Technology integration for airborne fog computing

Integrating ICN, opportunistic networking, and SDN into an airborne fog computing framework offers a cohesive approach to creating a robust architecture. Each paradigm uniquely contributes to increasing network services' efficiency, reliability, and scalability, especially in dynamic and challenging environments like those in airborne systems.

ICN plays a crucial role in shifting the focus from location to content. In an airborne fog computing scenario, data can be accessed based on content names, not physical addresses, which is ideal in highly mobile topologies. In addition, NDN-based extensions allow services to be viewed as content, emphasizing the consumer/user, reducing latency, and improving efficiency in environments where the network topology changes frequently.

On the other hand, opportunistic networking takes advantage of the transient contacts and intermittent connectivity standards in mobile networks to transfer data. In fog computing, this paradigm manages data packets during periods of disconnection or when traditional paths are unavailable. The opportunistic network adapts dynamically, using all accessible links and nodes to maintain service continuity, making node mobility a feature, not a limitation.

SDN contributes to this integration by providing a centralized control mechanism to manage network resources and adjust flow policies dynamically in real time. In an airborne environment, SDN can efficiently orchestrate network resources, facilitating the flexible management of data flows between nodes, an essential factor in maintaining robustness and quality of service. SDN's ability to reconfigure strategies in real time is compatible with both the ICN's content-centric approaches and the adaptive strategies required by the opportunistic network.

With this integration, the airborne fog computing framework can be conceptualized as a content-centric network that adaptively manages data transfers. SDN provides the comprehensive framework to orchestrate these operations, while the opportunistic network allows the use of transient contacts to maintain communication. The synergy between ICN, SDN, and opportunistic networking creates a resilient, adaptable, and scalable architecture capable of handling complex tasks and variable connectivity challenges.

This integration of technologies enables the development of advanced network strategies that can handle mobility, large-scale data management, and unstable connectivity. The decentralized but coordinated approach that results from this integration improves operational efficiency, ensuring that the network remains flexible and adaptable to meet a wide range of applications, from surveillance and communication to real-time analysis in remote or temporary environments.

3. DESIGN CHOICES

To implement an airborne fog computing platform, a suitable approach may use the service concept as the key abstraction. Each resource available in the network (from hardware resources like sensors, cameras, and CPU to software resources like databases, particular computational functions, and chunks of data) can be seen as a service that local and remote applications can invoke. The airborne fog computing platform aims to allow users to use and create more complex services by concatenating such basic services. Those more complex services are designed *service chains* [25]. A service chain is a sequence of computational functions applied to a data object or a group of different data objects. The conceptual architecture may be defined based on a set of design choices. For instance, as a basic component of such a platform, it is possible to name the local resources available on a device and abstracted as a set of basic services. An NFN-based network is itself seen as a service that can access other services available on remote devices. Specifically, in a dynamic environment, each network interface is a special resource managed through opportunistic networking technologies. A device can build a local abstraction (proxy) of each remote service by exploiting the network service.

In this way, an application running on a device cannot only use and compose services available locally, but it can also use and compose basic services or service chains based on remote services. The simplest case is when a service can be directly implemented by exploiting locally available resources. In this case, the service platform running on a particular device can work in isolation from the other devices of the network. However, in general, not all required resources are available locally, and hence the service platform will opportunistically look in the network for the missing services. The platform will discover the required resources, identify the optimal way of invoking them, gather and combine the results, and present them to the application in a suitable format. Besides building up the platform based on the concept of service (basic and chains), another useful design choice is task farming, a concept central to many distributed computing systems, such as Condor [26, 27] and MapReduce [28]. In all of these systems, a single master process manages a queue of tasks and distributes these among an ensemble of worker processes. Task farming algorithms also handle worker failure and load balancing of computational tasks.

In the airborne fog computing framework, the concept of task farming can be used as follows. Each created service (basic or chain) can be decomposed into a large number of tasks that are atomic, meaning that they have a constant execution duration. In this context, a fully sustainable architecture is proposed. This architecture is designed to uphold the required services even in the face of connectivity link failures and to scale more effectively on constrained computing devices, as all devices collaborate to provide the necessary support. In the subsections below, an example of conceptual modeling of network service designs is presented as functions to handle different constraints for the various applications in operation.

3.1 ICN-chained functions

The ICN-Chained Functions (ICNCF) presented is similar to Service Function Chaining (SFC) [25]. However, instead of the IP architecture in SFC, the ICNCF architecture uses NDN-based functions [29]. The consumer expresses an interest in content, generating an express interest. Fig. 2 demonstrates this request indicated by the green arrow. The express interest generates a process and the main node acts as the master, maintaining a queue of tasks to be executed. All other devices in the ad hoc network are workers, maintaining a local queue of a certain length, and being able to process a task in a certain time frame.

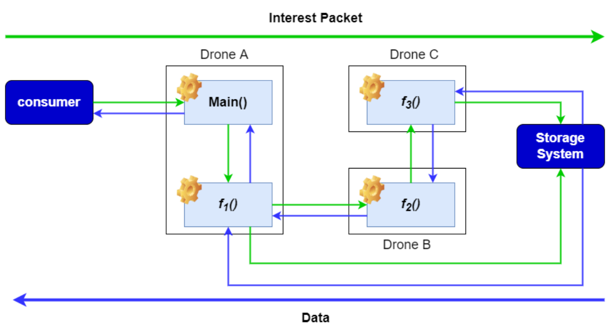

Figure 2 – Overview of ICN-Chained Functions (ICNCF).

When the consumer wants to execute a service it sends an interest packet with the set of tasks to be executed to the network. When the interest packet reaches a device (worker) that is able to execute all or some of the tasks, the worker's queue is filled with a defined number of new tasks. If such tasks depend on the result from other tasks that cannot be executed in the reached worker, the interest packet is further forwarded in the direction of a set of nodes that are able to execute the needed functions. After the functions are executed, the results are forwarded back to the previous workers, which will use them to complete the execution of the tasks that are locally queued. A successful service is one that has been processed by a set of workers and the result has been communicated to the consumer/master. In addition to this context, tasks can be cumbersome and sensitive to delays, in which case the task sets can be executed in parallel.

NDN can be advantageous for completing the service, due to the intermittent nature of the drones' connection, which can be in motion and temporarily lose a connection link, opportunistic networking strategy, and CS functionality. In these cases, if the connection is interrupted briefly, the drones store the context and service data locally in the CS. When the connection is re-established, this data can be quickly retrieved without the need to reprocess the information previously computed. In prolonged connection loss, an opportunistic networking strategy is implemented to establish a new communication link. In more complex scenarios, a data mule strategy can be used, in which a specific drone is assigned to manage data traffic, creating a temporary connection between the device and the network.

3.2 ICN-forked functions

In the context of the airborne fog computing framework, computational tasks can be resource-intensive in terms of resources used, and execution times can be difficult to estimate, resulting in longer timeouts. Thus, unlike ICNCF, ICN-Forked Functions (ICNFF) use a bifurcation or branching function structure to support parallelism. This is an option for delay-sensitive applications. The ICNFF operates in a more distributed nature, where functions operate in a self-organized fashion based on a set of network protocols capable of supporting conflict-free resource discovery and task assignment, and these tasks can be executed concurrently [30, 31, 32]. The bifurcation of functions enables the use of a segment-based approach, so the function does not need to wait for all following segments to become available to start executing its computational functions. Instead, the function can continue running its tasks on the current feature while other features are being fetched, as is shown by Fig. 3.

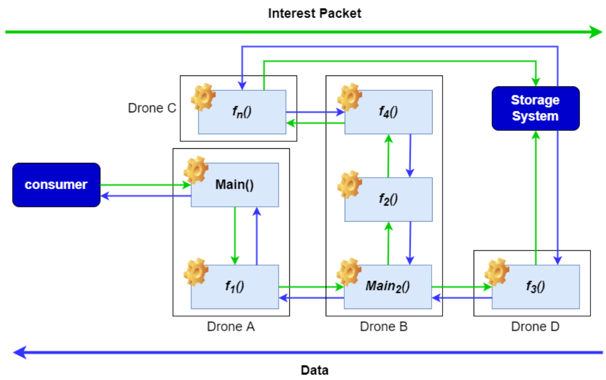

Figure 3 – Overview of ICN-Forked Functions (ICNFF).

The basic idea is that you don't have schedules. That's why resources are reserved at the time you start the first function. Equipment characteristics and data availability significantly affect this kind of design. The adoption of multiple Main() functions in distributed and forked chaining services has significant advantages, especially in terms of saving computing resources and reducing management complexity. This strategy allows services to be treated independently, optimizing management and resource allocation. In a scenario with forked services, the functions that precede a specific Main() function can remain in a sleep state until they need to be re-engaged. This occurs when the Main() closest to the data source has finished processing and is ready to merge and relay the results to the previous node in the chain. At this critical point, computing resources are then effectively allocated only when needed, avoiding waste and maximizing operational efficiency.

The functions are started and executed in the express interest direction with some computational effort. In the opposite direction, Send Data is processed (modified or some treatment performed, like a merge, for example), the Main() functions act as a manager and are present at the initial moment or when there is a branching (express the need for flow control in order not to compromise the final result of the service). Besides the concept of service and task farming, a third design choice is the concept of data storage virtualization able to combine multiple physical storage services into one or more logical services for the purposes of data redundancy. This will allow the airborne fog computing framework to keep operational even in the event of the collapse of some of its devices (e.g. drone failure due to battery drain out).

3.3 Reducing storage

With a data storage virtualization approach, distributed arrays of storage services also contain rebuild areas that are used to maintain redundancy after the failure of a device. The storage system uses the publish/subscribe paradigm to obtain the content made available by the producers [33, 34]. Fig. 4 presents an overview of the publish/subscribe paradigm acting on the storage system.

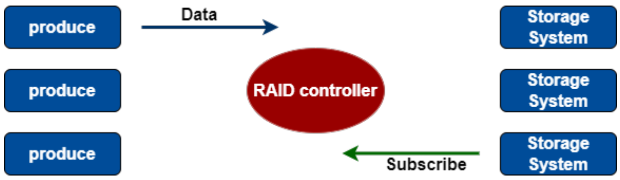

Figure 4 – Publish/subscribe paradigm to the storage system.

Distributed arrays of storage services are similar to RAID arrays in what concerns rebuilding bottlenecks in non-distributed array configurations because rebuild areas are distributed across all devices in the airborne network [35]. The workload for rebuilding operations is distributed across all devices, resulting in quicker restoration of service storage across all systems. Regarding managing idle systems, instead of assigning one or more devices as spares, the spare capacity is distributed over specific rebuild areas across all the member devices. Data can be copied faster to the rebuild area and redundancy is restored much more rapidly. After a failed device is replaced, data is copied back to that device from the distributed spare capacity. The number of rebuilt areas is based on the size of the network.

The size of the rebuild area determines how many times the distributed array can recover failed devices without risking becoming degraded. For example, a system similar to a distributed array that uses RAID 6 drives can handle two concurrent failures. After the failed devices are rebuilt, the array can tolerate another two failures. If all the rebuild areas are used to recover data, the array degrades on the next drive failure. Data replication is a process coordinated by the RAID controller, and some rules are applied. The goal is that the replicated data is more than a backup and works as a kind of Content Delivery Network (CDN). Thus, data is replicated across multiple storage systems using strategies that facilitate data consumption, improving the efficiency of content delivery to end users, and consequently decreasing data delivery time (thus serving delay-sensitive applications) while reducing data traffic on the network.

4. USE CASES

The airborne fog computing framework enables devices to harness resources from other non-exclusive and sporadically available devices. These resources support complex applications and operational requirements within a dynamic domain infrastructure. Such environments, which might include drone swarms, are relatively isolated, power-limited, and only intermittently connected to external networks, experiencing significant latency in these connections, as shown in Fig. 1. This section introduces two use cases that can be implemented in any settings outlined in Section 2.3. Initially, each use case is outlined, followed by an explanation of how the airborne fog computing framework facilitates their execution.

USE CASE #1: AREA OBSERVATION (VIDEO ANALYSIS AND SOUND DETECTION)

The first use case considers the airborne fog computing framework operating in a scenario post-disaster, performing search and rescue operations. The infrastructure comprises a swarm of drones, with wireless network transmission and packet capabilities, along with computational power for calculations and storage capacity. This first use case scenario describes the observation of an area using various producers for data generation (e.g. low range camera-equipped drones, small sound, and presence sensors dropped by airplanes over the area). For example, this task can be helpful for teams searching for survivors or monitoring teams to check the stability of a specific location, where real-time observation is hugely critical. In this scenario, receiving video with long delays and low video and audio quality cannot be tolerated. It is also considered that the consumers of this content are heterogeneous equipment, and the end user should not be expected to have computing power, so the content should be delivered appropriately (i.e., the content should be available ready to be consumed in the equipment without the need for any computational process by the end user).

Thus, considering the airborne fog computing framework performing forked or branched chained functions, packets are forwarded over the wireless network in a multi-hop fashion. The express interest are sent by consumers and generating calculation functions to fulfill the request. An optimization model is applied to create chaining of functions in the direction of consumer-producers. This is possible because the NFN approach allows functions to be transported as NDOS and started in any node with computational capacity for execution.

The data is preprocessed independently (for example, audio, and video). The fusion of this data is performed to decrease the network traffic as close as possible to the preprocessing functions. This happens while the data is being transmitted, i.e., in the opposite direction to express interest. Finally, near the consumers, the content is decompressed and delivered to the end user. Note that all the computation and data processing are generated at the same time that the data is being retransmitted. Furthermore, the computation and compression process happens close to the producers, decreasing the traffic and consequently the delay. Again, preprocessing can use more significant sources of producers, increasing the quality of the content. Furthermore, functions that guarantee data quality can be part of the chaining of functions depending on the policies and requirements of each application.

USE CASE #2: 3D MAP GENERATION

The second use case considers the execution of a 3D map generation task. For example, a rescue team wants to plan a search mission, or it is necessary to decide where the team can act to be more effective. Generating a map is, at first, a less complex task. However, it is necessary to be very accurate to achieve success in the mission, i.e., map failures are not tolerated.

It is expected to use several pieces of equipment (e.g. low-range drones equipped with cameras and video equipment installed on the team members' helmets) for the fusion of data to ensure more accuracy. This is one case where the data generated can be bound and made available to the requester directly from the database. However, a Pub/Sub approach to ICN may be appropriate in this context.

Thus, the calculation functions would behave differently from the previous use case, as a Pub/Sub between the producers and the database and another Pub/Sub between the database and the consumers would be needed to ensure an up-to-date map with the actual context since, in this scenario this is extremely important. The application can use functions to define the fusion of small maps (local maps of each piece of equipment) and send them to the storage system that would hold a global map, still depending on the application requirements. More complex calculation functions or not would be acting between the storage system and the consumer to deliver the appropriate content.

5. TRENDS AND FINAL REMARKS

Airborne fog computing resources can provide significant benefits over ad hoc networks, in particular limited available infrastructures. Depending on the tasks and heterogeneity of the network devices, multiple applications can perform complex tasks in the network core, with resource allocation and flow priorities being defined. Airborne fog computing can maintain necessary services even with connectivity link failures and will scale better even on constrained computing devices since all devices cooperate in providing necessary services.

Several legacy, non-ICN devices are expected to be part of the Airborne fog computing infrastructure, but it is not easy to completely replace communication technologies with ICN. Thus, hybrid Information-Centric Networking (hICN) can be considered in scenarios where interoperability with IP networks is essential. In this context, hICN allows reusing IP protocols to deploy hybrid IP-ICN solutions. Furthermore, hICN can exploit IP-based hardware and software implementations, making it much simpler to insert airborne fog computing infrastructure into current networks.

Future work directions include enhancing the airborne fog computing infrastructure and its operational scenarios. For example, reusing running compute intelligent discovery protocols for fetching content, such as BitSwap, which is inspired by BitTorrent to decrease response time and decrease thread inconsistencies.

ACKNOWLEDGMENTS

This study is partially supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) and Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES).

REFERENCES

[1] Mengting Liu, Yinglei Teng, F. Richard Yu, Victor C. M. Leung, and Mei Song. "A Mobile Edge Computing (MEC)-Enabled Transcoding Framework for Blockchain-Based Video Streaming". In: *IEEE Wireless Communications* 27.2 (2020), pp. 81–87. DOI: 10.1109/MWC.001.1800332.

[2] Hongxing Li, Guochu Shou, Yihong Hu, and Zhigang Guo. "Mobile Edge Computing: Progress and Challenges". In: *2016 4th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud)*. 2016, pp. 83–84. DOI: 10.1109/MobileCloud.2016.16.

[3] Sheyda Zarandi and Hina Tabassum. "Delay Minimization in Sliced Multi-Cell Mobile Edge Computing (MEC) Systems". In: *IEEE Communications Letters* 25.6 (2021), pp. 1964–1968. DOI: 10.1109/LCOMM.2021.3051558.

[4] Mingjie Feng, Marwan Krunz, and Wenhan Zhang. "Task Partitioning and User Association for Latency Minimization in Mobile Edge Computing Networks". In: *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. 2021, pp. 1–6. DOI: 10.1109/INFOCOMWKSHPS51825.2021.9484605.

[5] Asif Ali Laghari, Awais Khan Jumani, and Rashid Ali Laghari. "Review and State of Art of Fog Computing". In: *Archives of Computational Methods in Engineering* (2021), pp. 1–13.

[6] H. J. Na and S. Yoo. "PSO-Based Dynamic UAV Positioning Algorithm for Sensing Information Acquisition in Wireless Sensor Networks". In: *IEEE Access* 7 (2019), pp. 77499–77513. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2922203.

[7] Tulio Dapper e Silva, Carlos F. Emygdio de Melo, Pedro Cumino, Denis Rosário, Eduardo Cerqueira, and Edison Pignaton de Freitas. "STFANET: SDN-Based Topology Management for Flying Ad Hoc Network". In: *IEEE Access* 7 (2019), pp. 173499–173514. DOI: 10.1109/ACCESS.2019.2956724.

[8] Lucas Pacheco, Helder Oliveira, Denis Rosário, Zhongliang Zhao, Eduardo Cerqueira, Torsten Braun, and Paulo Mendes. "Towards the Future of Edge Computing in the Sky: Outlook and Future Directions". In: *2021 17th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. 2021, pp. 220–227. DOI: 10.1109/DCOSS52077.2021.00045.

[9] Navjeet Kaur and Ayush Mittal. "Fog Computing Serverless Architecture for Real Time Unpredictable Traffic". In: vol. 1022. 1. IOP Publishing, Jan. 2021, p. 012026. DOI: 10.1088/1757-899X/1022/1/012026. URL: https://dx.doi.org/10.1088/1757-899X/1022/1/012026.

[10] Preechai Mekbungwan, Giovanni Pau, and Kanchana Kanchanasut. "In-network Computation for IoT Data Processing with ActiveNDN in Wireless Sensor Networks". In: *2022 5th Conference on Cloud and Internet of Things (CIoT)*. 2022, pp. 197–204. DOI: 10.1109/CIoT53061.2022.9766613.

[11] Panos Patros, Josef Spillner, Alessandro V. Papadopoulos, Blesson Varghese, Omer Rana, and Schahram Dustdar. "Toward Sustainable Serverless Computing". In: *IEEE Internet Computing* 25.6 (2021), pp. 42–50. DOI: 10.1109/MIC.2021.3093105.

[12] Ridhima Rani, Neeraj Kumar, Meenu Khurana, Ashok Kumar, and Ahmed Barnawi. "Storage as a service in Fog computing : A systematic review". In: *Journal of Systems Architecture* 116 (2021), p. 102033. ISSN: 1383-7621. DOI: https://doi.org/10.1016/j.sysarc.2021.102033. URL: https://www.sciencedirect.com/science/article/pii/S1383762121000369.

[13] Mostafa Haghi Kashani, Amir Masoud Rahmani, and Nima Jafari Navimipour. "Quality of service-aware approaches in fog computing". In: *International Journal of Communication Systems* 33.8 (2020), e4340. DOI: https://doi.org/10.1002/dac.4340.

[14] Hadi Zahmatkesh and Fadi Al-Turjman. "Fog computing for sustainable smart cities in the IoT era: Caching techniques and enabling technologies - an overview". In: *Sustainable Cities and Society* 59 (2020), p. 102139. ISSN: 2210-6707. DOI: https://doi.org/10.1016/j.scs.2020.102139. URL: https://www.sciencedirect.com/science/article/pii/S2210670720301268.

[15] Brian Carpenter and B Liu. "Limited Domains and Internet Protocols". In: (2020).

[16] Dong Doan Van and Qingsong Ai. "In-network caching in information-centric networks for different applications: A survey". In: *Cogent Engineering* 10.1 (2023), p. 2210000. DOI: https://doi.org/10.1080/23311916.2023.2210000.

[17] Sana Fayyaz, Muhammad Atif Ur Rehman, Muhammad Salah Ud Din, Md. Israfil Biswas, Ali Kashif Bashir, and Byung-Seo Kim. "Information-Centric Mobile Networks: A Survey, Discussion, and Future Research Directions". In: *IEEE Access* 11 (2023), pp. 40328–40372. DOI: 10.1109/ACCESS.2023.3268775.

[18] Antonio Cortés Castillo. "An Overview of Integration of the Virtualization of Network Functions in the Context of Information Centric Networks". In: *2023 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*. 2023, pp. 1–7. DOI: 10.1109/ICECCME57830.2023.10253441.

[19] Shahzad Rizwan, Ghassan Husnain, Farhan Aadil, Fayaz Ali, and Sangsoon Lim. "Mobile Edge-Based Information-Centric Network for Emergency Messages Dissemination in Internet of Vehicles: A Deep Learning Approach". In: *IEEE Access* 11 (2023), pp. 62574–62590. DOI: 10.1109/ACCESS.2023.3288420.

[20] Michał Król and Ioannis Psaras. "NFaaS: named function as a service". In: *Proceedings of the 4th ACM Conference on Information-Centric Networking*. ICN '17. Berlin, Germany: Association for Computing Machinery, 2017, pp. 134–144. ISBN: 9781450351225. DOI: 10.1145/3125719.3125727. URL: https://doi.org/10.1145/3125719.3125727.

[21] Hussein Al-Omaisi, Elankovan A. Sundararajan, Raed Alsaqour, Nor Fadzilah Abdullah, and Maha Abdelhaq. "A survey of data dissemination schemes in vehicular named data networking". In: *Vehicular Communications* 30 (2021), p. 100353. ISSN: 2214-2096. DOI: https://doi.org/10.1016/j.vehcom.2021.100353. URL: https://www.sciencedirect.com/science/article/pii/S221420962100022X.

[22] Seweryn Dynerowicz and Paulo Mendes. "Named-data networking in opportunistic networks". In: *in Proc. of the ACM Conference on Information-Centric Networking*. Sept. 2017. DOI: https://doi.org/10.1145/3125719.3132107.

[23] W. Moreira, P. Mendes, and E. Cerqueira. "Opportunistic routing based on users daily life routine". In: *Internet Draft (draft-moreira-dlife-04)*. IEEE. May 2014.

[24] Paulo Mendes, Rute C. Sofia, Jose Soares, Vassilis Tsaoussidis, Sotiris Diamantopoulos, and Christos-Alexandros Sarros. "Information-centric routing for opportunistic wireless networks". In: *Proceedings of the 5th ACM Conference on Information-Centric Networking*. ICN '18. Boston, Massachusetts: Association for Computing Machinery, 2018, pp. 194–195. ISBN: 9781450359597. DOI: 10.1145/3267955.3269011. URL: https://doi.org/10.1145/3267955.3269011.

[25] Junning Zhang, Yicen Liu, Zhiwei Li, and Yu Lu. "Forecast-Assisted Service Function Chain Dynamic Deployment for SDN/NFV-Enabled Cloud Management Systems". In: *IEEE Systems Journal* 17.3 (2023), pp. 4371–4382. DOI: 10.1109/JSYST.2023.3263865.

[26] S. Prakash, Jeevanandham S, Abhinand S. L, and Adith S. "Enhanced Charge Scheduling of Electric Vehicle Using Optimal Andean Condor Algorithm". In: *2023 International Conference on Energy, Materials and Communication Engineering (ICEMCE)*. 2023, pp. 1–6. DOI: 10.1109/ICEMCE57940.2023.10433981.

[27] Marco Garofalo, Gabriele Morabito, Maria Fazio, Antonio Celesti, and Massimo Villari. "Workflow Engines in the Compute Continuum: a Comparative Analysis". In: *Proceedings of the IEEE/ACM 16th International Conference on Utility and Cloud Computing*. UCC '23. Taormina (Messina), Italy: Association for Computing Machinery, 2024. ISBN: 9798400702341. DOI: 10.1145/3603166.3632148. URL: https://doi.org/10.1145/3603166.3632148.

[28] Abdiaziz Omar Hassan and Abdulkadir Abdulahi Hasan. "Simplified Data Processing for Large Cluster: A MapReduce and Hadoop Based Study". In: *Advances in Applied Sciences* 6.3 (2021), p. 43.

[29] Hideyuki Saito, Kenji Kanai, and Hidenori Nakazato. "Development and Evaluation of Information-Centric Service Mesh Using NDN Function Chaining". In: *2024 IEEE 21st Consumer Communications Networking Conference (CCNC)*. 2024, pp. 222–227. DOI: 10.1109/CCNC51664.2024.10454719.

[30] Sadaf Javed, Ali Hassan, Rizwan Ahmad, Waqas Ahmed, Rehan Ahmed, Ahsan Saadat, and Mohsen Guizani. "State-of-the-Art and Future Research Challenges in UAV Swarms". In: *IEEE Internet of Things Journal* 11.11 (2024), pp. 19023–19045. DOI: 10.1109/JIOT.2024.3364230.

[31] Sabitri Poudel and Sangman Moh. "Task assignment algorithms for unmanned aerial vehicle networks: A comprehensive survey". In: *Vehicular Communications* 35 (2022), p. 100469. DOI: https://doi.org/10.1016/j.vehcom.2022.100469.

[32] Jie Li and Runfeng Chen. "A distributed task scheduling method based on conflict prediction for ad hoc uav swarms". In: *Drones* 6.11 (2022), p. 356. DOI: https://doi.org/10.3390/drones6110356.

[33] Cenk Gündoğan, Peter Kietzmann, Thomas C Schmidt, and Matthias Wählisch. "A mobility-compliant publish–subscribe system for an information-centric Internet of Things". In: *Computer Networks* 203 (2022), p. 108656. DOI: https://doi.org/10.1016/j.comnet.2021.108656.

[34] Diego Hernandez, Luís Gameiro, Carlos Senna, Miguel Luís, and Susana Sargento. "Handling Producer and Consumer Mobility in IoT Publish–Subscribe Named Data Networks". In: *IEEE Internet of Things Journal* 9.2 (2022), pp. 868–884. DOI: 10.1109/JIOT.2021.3069054.

[35] Alexander Thomasian and Yujie Tang. "Performance, reliability, and performability of a hybrid RAID array and a comparison with traditional RAID1 arrays". In: *Cluster Computing* 15 (2012), pp. 239–253. DOI: https://doi.org/10.1007/s10586-012-0216-9.

AUTHORS

Antonio S. da Silva is a Ph.D. candidate at the Federal University of Rio Grande do Sul (UFRGS), Brazil. He holds an MSc. in computer science from UFRGS (2021) and a BS in software engineering from the Federal University of Goiás (2019). His interests include ICN, SDN, fog computing, 5G, Beyond 5G and V2X communication.

Paulo Mendes is an expert in network architectures and protocols at Airbus in Munich, Germany, and associate researcher at the Technical University of Munich. In 2004, he got his Ph.D. degree (summa cum laude) in informatics engineering from the University of Coimbra, while being a visiting scholar at Columbia University, New York.

Denis Rosário received a PhD degree in electrical engineering at the Federal University of Pará (UFPA) with joint supervision undertaken by the Institute of Computer Science and Applied Mathematics of the University of Bern, Switzerland. He is now a professor at UFPA. His current research interests include the following topics: edge computing, FANET, and VANET.

Eduardo Cerqueira received his PhD in informatics engineering from the University of Coimbra, Portugal (2008). He is now an associate professor at the Federal University of Pará (UFPA). His research interests include edge/fog computing, smart cities, and mobility.

Edison Pignaton de Freitas has a Ph.D. in computer science and engineering from Halmstad University, Sweden (2011), an M.Sc. in computer science from UFRGS (2007), and a computer engineering degree from the Military Institute of Engineering (2003). Currently, he is an associate professor at UFRGS. His interests include computer networks, real-time systems, UAVs, and IoT.