SAPIENT: enabling real-time monitoring and control in the Future Communication Infrastructure of Air Traffic Management

Antonio Virdis, Giovanni Stea, Gianluca Dini

Abstract—This paper describes the SAPIENT system, a realtime monitoring and control infrastructure for Air Traffic Management. Within the latter, aircrafts constantly measure the state and quality of their datalinks, and *report* these measurements to a ground entity, tagging them with a time/space reference. The ground entity, then, builds a map of the monitored portion of the sky, and can feed back information to the aircrafts themselves regarding conditions that they would not be able to measure otherwise. This allows optimal *vertical handover* decisions to be made, increasing service continuity and improving communication performance. We show that the SAPIENT system can be implemented using existing technologies, without the need for expensive hardware. Moreover, we show via simulation that a small, negligible increase in the communication overhead due to SAPIENT reporting brings about considerable benefits.

Index Terms—Future Communication Infrastructure, Aeronautical Communication, Air Traffic Management, Single European Sky.

I. INTRODUCTION

A IR-Traffic-Management (ATM) operations include heterogeneous services, such as Communication, Navigation and Surveillance (CNS), Air Traffic Services (ATS), Airline Operational Communications (AOC), etc., having different data rates and performance requirements for the communication between aircrafts (ACs) and ground nodes. Moreover, the above services involve different stakeholders, e.g. airport authorities, Air Navigation Service Providers (ANSP), air-traffic-flow managers, etc., which traditionally employ dedicated communication systems.

In this context, the Single European Sky Air Traffic Management Research (SESAR) program in Europe, and the NextGen project in the US, have focused on facilitating the interoperability of existing communication systems for the next generation of aviation communication systems. One of SESAR's objectives is to define a novel communication architecture to improve performance, security and safety in ATM. Such system is called Future Communication Infrastructure (FCI) and is designed to support ATS and AOC communications in an end-to-end fashion, including both air-to-ground and air-to-air segments [1]. The FCI will mainly deal with *data* communication, initially leaving out digital voice. Services will have to meet tight operational requirements as for safety, communication capacity, flight and cost efficiency [11].

The FCI will integrate multiple datalinks (DLs), both legacy, such as satellite communication (SATCOM) and VHF Data Link (VDL) [3], and future ones, such as L-Band Digital Aeronautical Communications System (LDACS) [4], for performance and reliability reasons, thus leveraging multilink capabilities. As each DL will have its own network, the FCI will be responsible for connecting them all together, thus realizing a system of systems. Multilink capabilities, i.e. the ability to use two or more DLs, either concurrently or serially, are envisaged. This requires addressing both vertical (i.e., cross-technology, or inter-DL) and horizontal (intra-DL) handovers. In a multilink context, there are several issues to be addressed. The first one is how to ensure service continuity, i.e. to limit the interruptions in the execution of a service. Current practices allow for manual Vertical Handover (VH), often *reacting* to a degradation of the DL quality (e.g., loss of coverage from the serving DL). This poses safety challenges and hinders the deployment of Remotely-piloted Aircrafts (RPAs) for civil transportation. A proactive approach, whereby a new DL is sought and connected to while the communication (on the old DL) is still ongoing, would instead guarantee service continuity. A related issue is how to ensure optimal performance in the presence of multiple DLs. The availability of multiple DLs calls for algorithms for efficient and effective VH. We argue here that an AC may not possess enough information to make optimal VH decisions. For instance, an AC may well assess the quality of two or more available DLs, for instance by measuring their Signal to Noise Ratios (SNRs), and select the best one. However, it would be hard for it - if possible at all - to assess the level of congestion of a prospective DL, which will have a direct impact on its Quality of Service. Similarly, an AC may not know that the DL perceived as optimal at a given time may become suboptimal soon after, due e.g. to adverse weather conditions or a jamming attack further along the AC trajectory.

The above issues can be solved if - on one hand - AC are endowed with the capability to measure the state of *all their DLs*, and to make VH decisions based on their measurements, and - on the other - ACs are allowed to *report* the results of these measurements to a *central entity*, which can then collate them into a global picture of the portion of the sky under its control, and use that picture to suggest optimal VH decisions, this time with a *global* perspective.

Manuscript submitted on 02/04/2019 This work was supported by the SESAR Joint Undertaking under grant agreement No 699328 under European Union's Horizon 2020 research and innovation programme.

Antonio Virdis, Giovanni Stea and Gianluca Dini are with the Dipartimento di Ingegneria dell'Informazione, University of Pisa, Largo L. Lazzarino 1, Pisa, Italy (name.surname@unipi.it).

In this paper, we describe SAPIENT (*Satellite and terrestrial architectures improving performance, security and safety in ATM*), a monitoring and control system for aeronautical communication in ATM aimed at solving the above problem, realized within the framework of a SESAR EU-funded project¹. SAPIENT exploits ACs as *crowdsensing agents*, having them measure a set of Key Performance Indicators (KPIs) related to the DLs that they are hearing, and report them to a central *SAPIENT Server*. The latter, in turn, builds a map by collating all the reports, associating each KPI with the position and time where it was measured. Such a *4D Map* details the state of the sky over time. Based on the latter, the SAPIENT Server can then extract and send relevant views, called *summaries*, of a portion of sky to the AC themselves, in order to provide them with the necessary information to make globally optimal VH decisions.

The potential benefits brought along by such a system are manifold. First, and foremost, it allows each AC to leverage global information, i.e., information that it could not measure itself locally, but has instead been measured by other ACs and processed at the SAPIENT server. For instance, the state of a DL ahead in its trajectory, or the load of a DL to which that AC is not connected yet. This allows ACs to maintain service continuity and optimize performance. Moreover, the database of measurements of a portion of sky residing at the SAPIENT server can be used for several purposes: mining anomalies to discover potential security threats or underperforming DLs in real time, e.g., by comparing an AC's measurements with those of other ACs flying over the same trajectory in the past; factoring in communication reliability when optimizing trajectories; optimizing network operation (e.g., for DL providers) based on certified data. The same database can also be used to store related data, such as AC digital records for post-accident investigations. This would meet the needs of several stakeholders (e.g., air companies, network operators, security agencies, regulatory bodies, etc.).

In this work, we discuss the architecture of the SAPIENT system, highlighting the roles of its actors and functions and the implications of its design choices. Furthermore, we discuss practical issues related to the deployment and scalability of the SAPIENT system, therein including security aspects, showing that it can be deployed at little cost, using existing technology. Finally, we present case studies, obtained in a detailed simulated environment, highlighting some of the benefits introduced by SAPIENT for the ACs.

The rest of the paper is organized as follows: Section II reviews the related work. Section III describes the SAPIENT system. Section IV discusses technical issues related to the SAPIENT deployment. In Section V, we evaluate the system performance. Section VI concludes the paper.

II. RELATED WORK

Over the last few years, the goal of improving the performance and increasing the efficiency of ATM has been pursued using various methodologies and focusing on different aspects. We provide here a description of the main approaches, detailing how they relate to our work.

Work [2] considers the problem of spectrum management for aeronautical communications, providing an extensive taxonomy of cognitive-radio approaches. Most of the analyzed works share the idea of sensing the transmission medium at each AC before performing the actual transmission. ACs will then regulate the transmission power consequently, or possibly refrain from transmitting at all. A cognitive radio approach, however, relies on taking measurements at a single point, i.e. at the AC, and makes use of performance indicators related to radio aspects only. On the other hand, SAPIENT decisions can leverage global information on the state of the DLs which cannot be obtained by a single AC.

A different approach towards ATM improvement relies on AC-trajectory prediction and optimization. Work [5] presents a method to predict the behavior of ACs during flights. It uses a search tree to evaluate the possible maneuvers of the AC and generates both altitude and coordinates for the trajectory profiles. Such information can then be used to efficiently predict the position of ACs during the flight. In [6], instead, authors propose a method that uses air-traffic data to compute traffic routes and identify the risk of collisions in certain hot spots. All the methods falling in this category are complementary to the SAPIENT system, as information coming from the latter can be used together with knowledge of the trajectory to select the collision-free route having the best communication performance.

Finally, [7] describes a method for optimal DL selection, based on a multi-attribute decision making (MADM) approach. The authors identify a set of DL parameters, which are used, together with a set of user preferences, to select the best DL over time. The importance of including congestion-related parameters in the decisions is recognized by the authors; however, [7] only considers metrics evaluated locally at each AC. Moreover, it does not address how DL information is collected and maintained at the AC. This work can thus be seen as a use-case for SAPIENT, as it might exploit information collected through SAPIENT, both local and global, to support decision making.

The SAPIENT system is, to the best of our knowledge, the first proposal of an information system tailored to aeronautical communications, that collects information measured at ACs, and provides the same ACs with processed relevant information which they could not otherwise acquire, working in real time, and storing all the acquired data for possible future use.

Finally, two previous conference papers have presented an overview of the SAPIENT system [22] and of the simulator used to assess it [23]. This work presents a comprehensive view of the architecture of the SAPIENT system, discusses issues related to its deployment, and presents two comprehensive case studies.

III. THE SAPIENT SYSTEM

As discussed in the Introduction, the SAPIENT system defines an infrastructure to provide real-time monitoring of the FCI. Its linchpin are the KPIs, measured by various entities and shared through the SAPIENT infrastructure. Each KPI is measured and



Fig. 1. High-level representation of an ATM system.

associated to a time/space tag, representing the local view possessed by a single entity, e.g. an AC or a ground station (GS). Tagged KPIs are periodically transmitted towards a central SAPIENT server, which merges them together to create KPI summaries and stores them permanently for future use. KPI summaries contain information on the status of a given, wider-scope area over time (e.g. the congestion status of an LDACS sector, or the signal quality of a link over a trajectory). KPI summaries are shared within the SAPIENT system (notably, sent back to the ACs) to optimize the (re)configuration of the communication network. When assuming the point of view of an entity of the SAPIENT system capable of taking measurements (e.g., an AC or a GS), it pays to distinguish the information into local and global. We will call local any information that can be measured by the entity itself. For instance, the SNR measured by an AC is local information. Conversely, we will call global any information that an entity can only obtain through the SAPIENT system. For instance, an estimate of the SNR ahead in an AC's trajectory is global information, which the AC receives in the form of a KPI summary from the SAPIENT server, following the elaboration of KPIs measured and reported by other ACs in the vicinity.

In the rest of this section, we will provide a detailed description of the SAPIENT system. First, we will list all the elements composing the SAPIENT architecture and describe how they are connected. Then, we will discuss KPIs and their characteristics. Finally, we will describe the SAPIENT functions and present two use cases. In doing so, we will often make reference to a logical representation of an FCI-based ATM network, shown in Fig. 1. The overall network can be divided into *domains* as follows:

- AC domain: it is the network within the ACs, comprising airborne applications and routers, i.e. connection interfaces towards ground;
- DL domain: it provides the connection between the AC domain and the core network on ground, comprising all the available DLs, each one managed by a Service Provider (SP). For each DL, connectivity is provided by means of several Ground Stations (GSs);



Fig. 2. Main elements of the SAPIENT system and their operations, namely measurement and reporting [M], collection [C], processing [P], dissemination [D] and decision making [E].



Fig. 3. Exemplary KPI reporting and summary distribution within a SAPIENT Architecture composed of 2 ACs, two DLSPs and one Sapient Server.

- Core Network Domain: it is an IP-based network that connects the DL domains with the ground applications;
- *Application Domain*: it includes entities that provide service to the airborne applications, such as Air Traffic Control Centers, ANSP, etc.

A. System Architecture

The architecture of the SAPIENT system incorporates two types of nodes, namely *subscribers* and a SAPIENT Server (SS). A subscriber is a node that subscribes to SAPIENT and behaves cooperatively according to it, e.g. an AC or a DL service provider (DLSP). The subscriber, on one side, measures KPIs and reports them to the SS, and - on the other - it *may* receive and use *KPI summaries* from the SS. The latter is a centralized entity that collects all the KPIs from the subscribers, stores them and processes them to compose summaries, which are then shared with a subset of subscribers, based on contractual agreements or public sharing of information. It is worth mentioning that the SAPIENT system does not require existing business relationships between FCI entities to be modified. Specifically, it complies with the decentralized nature of the infrastructure, where multiple DLSPs and network operators have preexisting bi- or multi-lateral agreements.

The applications that run the SAPIENT logic are called SAPIENT Application Nodes (SANs) and are located within the subscribers and the SS. A high-level view of the SAPIENT architecture and of the performed operations is shown in Fig. 2. We describe in detail these operations in Section III.C. In Fig. 3 we show an exemplary deployment of the SAPIENT system composed of two ACs and two DLSPs (acting as subscribers) and one SS. AC₁ communicates through SATCOM, whereas AC₂ uses LDACS. Each subscriber measures KPIs (step 1) and sends them towards the SS, namely AC₁ generates KPI_{AC1}, SATCOM DLSP generates KPI_{SAT}, etc. The SS collects the KPIs from all subscribers, processes them and generates a KPI summary (step 2). The



Fig. 4. High-level view of the operations of the SAPIENT protocol and the corresponding interactions.

summary is finally sent back towards the ACs using their main DL, first through the ground network (step 3_a), then through the air segment (step 3_b).

B. SAPIENT KPIs

KPIs are binary or multi-level parameters that are measured and exchanged within the SAPIENT system. SAPIENT defines an extensible KPI syntax, which allows users to define new KPIs. However, the project also identified and classified a group of fundamental KPIs for DL monitoring, among those that can be measured without requiring massive architectural modifications of the LDACS and SATCOM segments, or major changes to the existing AC equipment. KPIs are divided into the following three categories, depending on the pace at which they are measured, collected and processed:

- *Real-time KPIs* (RTKs) are collected with high frequency (e.g., seconds or less) and describe the status of DL connections during the flight. These KPIs are originated at and disseminated by the AC. An example is the SNR or the communication bitrate.
- *Periodic KPIs* (PKs) are collected with periods in the order of minutes or hours. These KPIs are originated at and disseminated by the DLSPs. An example is the communication latency or network congestion level.
- Statistical KPIs (SKs) are calculated over long periods of time (days to months), by processing RTKs and PKs. These KPIs are originated at and disseminated by the SS or the DLSPs. Examples are the link availability or continuity.

It is worth mentioning that the KPI reporting is independent of their semantics, which allows new ones to be defined depending on the considered use case.

We exemplify here two KPIs that will be used in the next sections in the context of use cases. First, the SNR, that measures the received power relative to the strength of the receiver noise. Increasing SNRs indicate better quality of reception, and generally higher communications accuracy and reliability. The SNR KPI is measured on the AC side at high frequency, thus being a RTK, and can be used to monitor the physical-level quality of the communications over a DL. Monitoring of the SNR KPI can be exploited to prevent a connection loss of a DL.

Second, the Load Indicator (LI) KPI measures the amount of data buffered for transmission at each DL transmission interface, on either the ACs or the GSs. This KPI can be used to evaluate the congestion-level of the GSs, and to identify over- and underloaded portions of the sky. The LI KPI can be either measured in



Fig. 5. Format of a SAPIENT KPI report.

real time, to capture fast-paced load variations, or at longer periods, to provide a rough estimation of the overall number of active connections over a portion of the DL network.

C. SAPIENT Functions

SAPIENT functions are the operations required to measure and disseminate KPIs, create and maintain a global sky map, and optimize DL communications. The main SAPIENT functions include the KPI measurements, the tagging and KPI reporting, the KPI collection, the creation of KPI summaries and the link optimization, as shown in Fig. 4. The above functions run with independent periods, in a fully decoupled way, and they interact via information exchanges. The KPI measurement is performed by the DL hardware located within each subscriber node, either an AC or a GS, by periodically reading the levels of the monitored KPIs. The Tagging associates a time and 3D position tag, called a 4D tag, to a measured KPI. Time and position references can be obtained via the available Global Navigation Satellite System (GNSS) infrastructure. Tagged KPIs are then arranged into SAPIENT KPI Reports (SKRs) for reporting to the SS. The SKR message format is shown in Fig. 5. It includes the ID of the subscriber and the DL being measured, a set of tagged KPIs identified with a KPI ID, and security information (of which more in Section IV.B). SKRs can be transmitted either upon request from the SS, or periodically. In this last case, the period may differ from the one of KPI measurement, which allows the reported value to be configured as a specific *aggregator* of a moving window of measurements (i.e., the mean, median, maximum, minimum, etc.). The KPI-Collection function receives the SKRs and stores them for fast and efficient retrieval. Section IV.B discusses the time/space overhead required for KPI collection and storage.

The *KPI-summaries* creation and dissemination retrieves the stored KPIs, processes them to produce a summary of the status of a DL over a certain 4D span, i.e. over time and covering a given space, and forwards them to the subscribers. The means by which summaries are created and their actual format do depend on the considered use-case, e.g. summarizing the load of a DL in a certain region in the last 30 minutes. In Section III.D we will discuss two use-cases using different KPI summaries.

Finally, the *link optimization* function uses the information on the system status, obtained via either or both tagged KPIs and summaries, to improve the performance of the communications, e.g. by triggering a vertical/horizontal handover, modifying the DL parameters (modulation, transmission power), etc. As shown in Fig. 4, link optimization can run at both subscribers (e.g. ACs) or the SS. In the first case, it will leverage both *local* information (i.e., locally measured KPIs) and *global* information, i.e., the KPI summaries received from the SS. In the second case instead, the SS will leverage visibility of possibly all the *reported* KPIs in its area of influence.

As already discussed, each SAPIENT function runs independently, at a distinct and configurable pace. This allows one to optimize operations separately by tuning its parameters. The optimal tuning of these parameters depends on many factors, e.g. availability of bandwidth, AC speed, size of the monitored area, etc., and strongly depends on the considered scenario and use case. In Section V we will evaluate how the system performance varies with summary dissemination periods. An exhaustive evaluation of the system parameters is outside the scope of this paper and is left for future study.

D. Use Cases

We describe here two typical scenarios involving, respectively, a horizontal and a vertical handover, showing how employing SAPIENT brings significant benefits.

1) Horizontal Handover

Horizontal handover (HH) is the procedure through which an AC changes the serving satellite or GS, without changing the DL. The SAPIENT system can provide benefits in the following scenarios:

- legacy ACs equipped with limited monitoring capabilities, that can measure KPIs for one frequency at a time on their DL (say, a satellite-based DL). In this case, the AC SAN could not infer through measurement that another satellite, operating at a different frequency, is available and guarantees a stronger signal. However, SAPIENT can convey to the AC SAN an association map, i.e. a 4D map associating points in space with the best satellite, which the latter can exploit to trigger a HH. Such association map can be created by the SS using the received KPIs, i.e., leveraging the sensing capabilities of other ACs that flew in the same area, and be transmitted to the AC via KPI summaries, or even disseminated before take-off. During the flight, the AC SAN will identify the best serving satellite based on its GNSS position, and its link optimization function will trigger a HH when needed.
- Load-aware Horizontal Handover: DL SANs can report per-cell load information through the LI KPI to the SS. The latter, in turn, generates and reports to each AC a KPI summary, containing the most up-to-date load information. Finally, the AC SAN collects KPI summaries, and uses them in conjunction with the locally measured ones to select the best available GS or satellite. For instance, an AC may not want to perform a HH to the LDACS GS with the strongest SNR (as measured locally), because the latter is currently overloaded. There would be no way to know this proactively, unless the SAPIENT system is used: the builtin HH mechanism of LDACS would rely on physical-layer, *local* information only (e.g., the SNR), neglecting higherlayer, *global* information (i.e., the cell load), which is instead highly relevant.
- 2) Vertical Handover

As for SAPIENT-assisted VH, we consider here the case



Fig. 6 - Example of usage of the performance-drop map.

where each AC can use either SATCOM or LDACS. Two scenarios are envisioned:

- Each DL works in nominal conditions. In this case the AC SAN will compare the locally measured KPIs of both DLs, and its link optimization function will trigger VHs to stay on the best DL at any time.
- One or more DLs are suffering from heavy localized performance drops due to external factors (e.g., weather conditions, jamming attacks, etc.), in an area intersecting a future portion of the AC's trajectory. These conditions could only be detected by that AC after the fact, i.e. following loss or degradation of connectivity, but they can be known earlier at the SS, which observes KPIs reported by other ACs traversing the affected area. The SS can then create and disseminate, via KPI summary, a performancedrop map, with information on performance drops, to the interested ACs (i.e., those about to enter the affected area). The AC SAN will use the locally measured KPIs for the available DLs, as well as the (global) performance-drop map, and trigger a preemptive VH to prevent a connection loss or degradation. In Fig. 6 we show an exemplary scenario where the performance-drop map is used. An AC flying north-west is connected to satellite A. Assuming it can detect the best DL dynamically, at time t_1 , it will select SATCOM, and B as its serving antenna. However, at time t_3 the AC is bound to enter an area where it may experience poor connectivity, e.g. due to weather conditions, possibly leading to a connection loss. Using a performance-drop map, instead, the AC will be able to foresee the performance drop at time t_2 and switch to the LDACS DL. Doing so might result in an overall lower SNR, due to the greater distance from the antenna, but it will avoid a potential connection loss.

Note that HH and VH are not necessarily alternatives, since a VH cannot occur when only one DL is available (e.g., on oceanic airspace), and HH cannot occur unless two alternatives for the same DL are available (e.g., near the cell border of an LDACS deployment). When both HH and VH are possible, SAPIENT will be able to support a multiple choice among all the possible configurations. If the SAPIENT decision involves a VH, then the routing in the core network will need to be changed, as we discuss in the next section. Otherwise, if a HH decision is made, routing will only change in the DLSP's own domain. Different usage costs for the various DLs, as well as

business relationships between actors, may also play a role in a VH decisions. However, such non-technical aspects are outside the scope of this paper.

IV. DEPLOYMENT ISSUES

SAPIENT is meant to be integrated into the FCI, ensuring efficient, scalable and secure operations. In this section, we analyze three deployment issues. First, we discuss the implementation of multilink functions in an IP-based network; then we explore the options to provide secure operations of the SAPIENT protocol, also taking into account possible confidentiality issues; finally, we analyze the storage, computation and communication requirements of a large-scale SS. Our goal is to show that the deployment of the SAPIENT system does not require new underlying technology or expensive hardware.

A. Network solutions for multilink routing

As discussed in the Introduction, the FCI is expected to provide multilink capabilities to ACs, enabling the use of two or more DLs. SAPIENT allows one to design multilink decision policies, i.e. to select a new DL and initiate a VH, to optimize the performance with a global outlook. However, a VH decision has to be enforced in the network, i.e., the FCI network must possess the necessary functions to enforce the multilink decisions. The network functions to support VH depend on the network architecture and are therefore outside the scope of SAPIENT. However, for the sake of concreteness, we discuss here the available options for ML routing in an IP-based network. Several approaches have been proposed in the last years [18]. It is first worth mentioning that exploiting the standard IP interdomain routing protocol, i.e. the Border Gateway Protocol (BGP), is not a feasible option. In fact, BGP is too slow for the time constants involved in AC mobility. BGP's convergence time is in the order of tens of minutes, and too frequent announcements (at a pace faster than four hours) are ignored by BGP neighbors. Therefore, if an AC acts as Autonomous System, and its network prefix is announced to the network by the serving GS and withdrawn after a VH, most of the time the rest of the network will just ignore the updates.

A better option would be to use the Network Mobility (NEMO) protocol [21], a tunneling-based extension to Mobile IPv6 (MIPv6), which extends the concept of mobile node to that of a Mobile Router (MR). The MR has at least one mobile network prefix and provides connectivity to several mobile nodes. As with MIPv6, the MR is registered to a Home Agent (HA) within its home network and, as soon as it attaches to a new network, it registers the new address, called Care of Address (CoA) to the HA. The latter will receive all the packets directed to the old address, and use a tunnel to forward packet to the MR. In the context of ATM, NEMO regards the AC as a mobile network, whose MR is the airborne router, which is given an IP address by the HA. The main problem with NEMO, however, is that every communication must pass through the HA, making routing suboptimal and increasing latencies.

A third option is the Locator/Identifier Separation Protocol (LISP) [12], which has been originally proposed for aeronautical



Fig. 7 - Exemplary configuration of an aeronautical network using LISP.

system in the framework of the SESAR project P15.02.04. The main goal of LISP is to provide an efficient and scalable multilink solution, minimizing the complexity in the AC and overhead cost on the DLs. Consider as an example the scenario of Fig. 7, where an AC can be reached through two DLs, SATCOM and LDACS. The AC is given a position-independent and link-independent address space, called End-system Identifier (EID), which uniquely identify the AC's network but that cannot be used to route traffic on the ground network. Border routers between the DL and core network domains, and between the latter and the application domains, need to be LISP-capable routers (shown in a darker color in Fig. 7), i.e. capable of translating EIDs to routable addresses, called Routing LOCators (RLOCs). When an AC connects to a DL, its EID is notified to the DL network, which registers to a LISP Registry the address of one of its border routers, as the RLOC for the given EID. Whenever a node attempts to communicate with an EID, the border router of its network will first query the LISP Registry and obtain its RLOC, and then it will tunnel the original packet towards that RLOC. SAPIENT multilink selection policies can be enforced in a LISP-enabled system, by changing the EID-to-RLOC mapping dynamically, e.g. when the AC changes its DL following a VH.

This field of research is active at the time of writing. There is space for developing ad hoc solutions, possibly enhancing existing protocols to better cope with the critical nature of aeronautic communications. A possible improvement would be to pre-provision interdomain paths, at least in part, so that VH operations occur faster.

B. Security

The communication between the SANs and the SS has strong safety implications, hence must be made secure. This subsection presents the security mechanisms that have been chosen to that effect, discusses hacking risks and mitigation strategies, as well as information confidentiality issues.

SAPIENT security design is mainly concerned with application-layer security, and it builds upon the security mechanisms and services already available at the lower layers. In fact, DL layers already provide countermeasures against jamming and eavesdropping [33]. Furthermore, international organisms such as ICAO, CANSO, EUROCONTROL and ENISA provide recommendations and guidelines for cybersecurity in ATNs including IP networks. The communication between the SANs and the SS can be made secure using standard mechanisms. We guarantee confidentiality, integrity, and data origin authentica-

tion of SAPIENT communications by means of an authenticated-encryption, following ICAO cybersecurity recommendations for IP ATN [34]. This is a cryptographic scheme that provides a unique interface such that encryption simultaneously conveys confidentiality, integrity and authenticity assurance on data, whereas decryption is combined in a single step with integrity verification. Authenticated-encryption also provides plaintext awareness and security against chosen-ciphertext attacks [24]. Authenticated-encryption can be implemented at the network layer, by canalizing traffic through an IPSec tunnel in the IP Encapsulated Security Protocol (ESP) mode, for example [25]. Alternatively, at the transport layer, traffic could flow through a Transport Layer Security (TLS) channel [26]. A potential problem with TLS is that SATCOM links have large delays, which may affect the establishment of sessions. However, TLS provides compression and ways to reuse old sessions, thus mitigating this issue [27]. Finally, protection could be implemented directly at the application layer [26]. Although this places a greater burden on the solution development, it has the advantage that protection mechanisms would have a minimum conflict, if any at all, with mechanisms for performance enhancement which generally require to manipulate network and/or transport headers [27].

The originator and recipient of data protected by the authenticated-encryption mechanism must share a secret symmetric key. There are many ways in which a key can be established [29],[30],[31]. For both the Ground-Ground Network and the Air-Ground Network, ICAO recommends the Internet Key Exchange (IKEv2), provided security is addressed at the network or the application layer [26]. At the transport layer, TLS instead encapsulates its own authentication and key establishment protocol.

ISO/IEC 19772:2009 has standardized six different authenticated-encryption schemes, namely Offset Codebook (OCB), Key Wrap, Counter with CBC-MAC (CCM), EAX, Encryptthen-MAC (EtM), and Galois Counter Mode (GCM) [28]. While in principle any of these schemes could be used, CCM, EtM and GCM are better suited for SAPIENT, because they can reuse cryptographic transforms already available on the platform for air-ground and ground-ground security (e.g., in IKEv2, TLS) [26].

Much like what happened in the early 1990s when desktop PCs started being massively connected, the fact that ACs will have broadband connectivity, and that systems like SAPIENT will exploit it for machine-to-machine communications, will increase the likelihood of hacking attacks. An infamous such attack is documented in [35], where the attacker gained access to the AC's Flight Management System through the Aircraft Communications Addressing and Reporting System (ACARS), when the AC was in autopilot, under controlled laboratory conditions. The attacker was then able to control the airliner's course, speed, altitude and other internal systems. Although SAPIENT is not going to use ACARS, the above work proves that a hacking attack exploiting software vulnerabilities may have severe consequences.

However, many of the lessons learned in hardening server and desktop computing systems can be quickly re-purposed for this context, as done e.g. for automotive and SCADA systems [36]. In order to mitigate exploit attack, it is advisable to employ well-known software engineering methodologies and tools for finding and preventing software vulnerabilities including [37]-[38]. As for development, managed programming languages should be used, e.g., strongly typed languages such as Java or Python, together with safe libraries, (e.g., Libsafe), that implement countermeasures (e.g.) against buffer overflow. Source code analysis tools capable of detecting well-known software vulnerabilities (e.g., Its4 or Rational Purify) should be employed, as well as compiler-level countermeasures, i.e., compiler extensions that perform additional control at compile-time (e.g., StackShield and StackGuard for the gcc compiler). As for software operation, the use of operating-system-level countermeasures should be required, such as executable space protection mechanisms that can be supported either natively (e.g., BSD and OSX) or as an extension (e.g., PaX, Exec Shield, and OpenWall).

Finally, another key issue is that of information confidentiality. The SAPIENT system is meant to be used by different actors, including national and international authorities and private companies, whose interests and information sharing policies may not always be aligned. Some information may be restricted: e.g., a KPI may be obfuscated or omitted in a given region of space due to reasons of national interest, or some KPI summaries should not reach the ACs of a given airline company. We argue that it is relatively easy to implement information restriction rules in the SANs and in the SS (e.g., using predicates in database queries). A possible deployment option is also to deploy SSs on a per-country basis, the better to safeguard national interest.

C. SAPIENT server requirements

The SS collects, stores, retrieves and disseminates KPI information, hence may constitute a potential performance bottleneck. Hereafter, we show that the communication, storage and computation requirements of the SS are easily met by today's technology, even considering future scenarios with increased air traffic. The SS is in fact a Geographic Information System (GIS), i.e., a database for geographically referenced data [14].

The SS performs both real-time and non-real-time operations. Real-time operations are KPI reception and storage, and creation of the KPI summaries, and these pose the most challenging requirements. Non-real-time operations, such as anomaly detection, network performance evaluation, etc., will also be performed at the SS, but they do not pose significant time constraints. For these we refer the interested reader to works on GIS in the context of Big Data analysis [15].

In order to infer requirements, we start from the scenario analyzed in [19]. In the latter, the peak number of ACs flying simultaneously over the ECAC area in 2014 was ~2800, and a growth to ~3500 by 2020 is foreseen. In Table I we provide a list of deployment-cost metrics, assuming (to err on the safe side) that each airborne AC generates a KPI report of 250 bytes (corresponding to the reporting of 15 KPIs per update) every 15 seconds. As far as communication capabilities are concerned, the *peak data rate* is the maximum instantaneous data rate that is expected to reach the SS, thus affecting its network interfaces. As the table shows, the peak data rate is certainly not an issue for today's technology, not by a long shot. Regarding storage, instead, the *daily data volume* is the total amount of new SAPIENT-related data that must be stored by the SS per day, and is in the order of 4-5 GB, i.e. less than 2 TB per year. We do not foresee that the SS will need to store historical data reaching too far back in the past (not for real-time operations, in any case). The storage capabilities of current general-purpose computers are already enough to store months of data, and the monthly cost of online storage providers is around 0.0045 USD per GB [20], i.e., few dollars a year overall.

Finally, for what concerns computational requirements, we assess the cost of querying the database and retrieving the necessary information to compute a KPI summary. Assuming a KPI summary reporting period of 30 seconds, the query size is the total amount of information that must be retrieved per minute from the disk to compute the summary, assuming no query optimization (again, to err on the safe side). The resulting *disk read time* is the time required to complete the read operation, and it depends on the disk throughput. In Table I we consider two examples of disk performance for general purpose computers, having 25 MB/s (low-end) and 200 MB/s (high-end) throughput, respectively. The figures are in the order of tens or hundreds of milliseconds per query, which is again not challenging. The above considerations show that a SS can be built out of inexpensive technology, and it will not constitute a performance bottleneck given the amount of foreseen air traffic in the future.

 TABLE I

 SS DEPLOYMENT COST FOR 2014 AND 2020 TRAFFIC SCENARIOS

	2014	2020
Peak data rate	45.57 KB/s	56.62 KB/s
Daily data volume	4.03 GB	5.01 GB
Query size	1.4 MB	1.74 MB
Disk read time		
- @25 MB/s	56 ms	69.57 ms
- @200 MB/s	7 ms	8.7 ms

V. PERFORMANCE EVALUATION

In this section we analyze the impact of the SAPIENT system on the ATM performance. We first introduce the simulator used for the evaluation; then we analyze in detail the performance of the use-cases discussed in section III.D.

A. Overview of the SAPIENT simulator

The SAPIENT simulator has been designed, coded and validated as part of the project effort. It is based on the OMNET++ framework [13], an extensible, modular, component-based C++ simulation library and framework, and is implemented following the *discrete-event dynamic simulation* (DEDS) paradigm, i.e. one where simulated time is advanced in discrete steps, as a consequence of *events* occurring (e.g., the arrival of a packet at a node). The SAPIENT simulator includes over 19k lines of code (not including imported models), and models an ATM communication network, including several components: the ACs, the DLs infrastructure, the core network [23]. Each component of the architecture is itself composed of a number of nodes, which are then modeled and implemented with the required level of detail.

With reference to Fig. 1, each domain of the ATM network is modeled as follows: the Application domain includes models of



Fig. 8 - Graphical representation of the simulated scenario.

TABLE II	
MAIN SIMULATION PARAMETERS	

Parameter Name	Value	
Simulation duration	24 hours	
Warm-up duration	100 seconds	
# of replicas	6	
# ACs	80 to 200	
- Flying speed	900 km/h	
Low-Priority App		
 Packet size 	100 Bytes	
 Inter-packet time 	1 second	
High-Priority App		
 Packet size 	40 Bytes	
 Inter-packet time 	40 milliseconds	
Transport layer	User Datagram Protocol (UDP)	
TERRESTRIAL DL		
 Slot duration 	60 milliseconds	
 Slot size FW 	2236.416 Bytes (291.2 kbit/s)	
 Slot size RT 	1691.904 Bytes (220 kbit/s)	
SATCOM DL		
 Slot duration 	224 milliseconds	
 Slot size FW 	12000 Bytes (420 kbit/s)	
- Slot size RT	1800 Bytes (60 kbit/s)	

applications that are used for ATM operations, e.g. ATS and AOC. Applications can be deployed at either an AC, or a groundbased server. Each application can generate traffic having various priority levels, and each network element schedules packets according to that priority. The Core Network (CN) comprises all the relevant elements of the ATN/IPS, e.g. IP routers, which are used to provide connectivity for the ground network. The model of the CN domain includes information on the position of nodes and their connection, link bandwidth and delay, routing etc. The DL Domain includes models of the SATCOM and LDACS DLs, which provide connectivity between air and ground domains. For each DL we model both the communication elements of the air segment, i.e. GSs and interfaces between the latter and the ACs, and the ground segment of the DL. Finally, the AC Domain models the ACs as communication endpoints, and includes applications, mobility models, etc. Each AC moves over a floorplan following a trajectory simulated as a waypoint model, i.e. a sequence of {3D coordinates, speed}, i.e. keeping a constant speed until the next waypoint. ACs communicate using either or both the available DL networks. The SAPIENT simulator can easily be extended to incorporate other DLs.

We simulate a communication network covering an area of $2400 \times 2400 \text{ km}^2$, and composed of 16 LDACS antennas and 4

satellites, as shown in Fig. 8. A configurable number of ACs moves across the floorplan at a speed of 900 km/h and at an altitude varying between 0 and 10000 meters to simulate different flying phases. ACs generate data traffic periodically from two applications, having low and high priority respectively, and the traffic is sent towards a ground-located server. A summary of the main simulation parameters is given in Table II. In the graphs that follow, confidence intervals are omitted when negligible.

B. Scenario 1: horizontal handover

In this scenario, we analyze the performance of the load-aware HH described in Section III.D.1). We compare it against a scenario where HH are initiated only based on locally measurable SNR information, neglecting a cell's load information (which is not directly measurable by an AC, at least not until after the AC performs the handover itself).

We simulate an increasing number of ACs moving across the floorplan, each using the LDACS DL. In a baseline scenario, ACs will only change their serving GS because of a better SNR, regardless of the perceived load (i.e., they will stay on a congested GS if its SNR is high enough). On the other hand, in a system running SAPIENT, DL SANs report the LI KPI to the SS every second (note that this does not generate traffic on the air segments, as it only travels through the core network). The SS, in turn, will generate and report to each AC a KPI summary containing the most recent load information at the LDACS GSs in the area traversed by the AC, at a variable period. The AC SAN will then test the ratio of the LI to the SNR for each GS, and it will connect to the GS having the lowest ratio, possibly performing a HH.

First, we measure the frame loss for high-priority traffic on the air-to-ground link. Frames are lost if they are dropped (e.g., due to decoding errors) or they reach their destination later than a predefined (end-to-end) delay threshold. Fig. 9 reports the loss rate, assuming a threshold equal to 485ms, as per ATS-voice performance requirements [32], and with an increasing number of ACs (up to 200). The figure clearly shows that the baseline solution exhibits high loss rates with more than 80 ACs. On the other hand, SAPIENT reduces the loss rate. The SAPIENT solution is tested with several reporting periods for KPI summaries, from a very short one (1s) to a very large one (60s). While activating SAPIENT provides benefits, whichever the period, a trend is clearly visible: too long a period implies that reactions to overload conditions are delayed, and frames may get lost in the meantime. A shorter period, instead, guarantees faster reaction, but increases the load on the air-to-ground link as well (due to the added load of the reporting traffic). This last effect starts showing up when the period is particularly short (e.g., 1s in the simulation). Period of 15-30 seconds appear to be the optimal choice in the above scenario. In a real deployment, the optimum period will depend on the actual load of the DL in the production network. Fig. 10 translates the frame-loss reduction, which is a networklevel metric, to a user-level metric, namely the percentage of AC that can be considered in ideal communication conditions, i.e., having a frame loss rate below 10%. The figure shows the percentage of ACs in ideal conditions against the number of ACs being simulated. For instance, it shows that, with 120 ACs,



Fig. 9. Average Frame Loss for a delay threshold of 485ms, with an increasing number of ACs and for various KPI summary reporting periods in SAPIENT.



Fig. 10. Percentage of satisfied users having a packet loss below 10%, in case of a delay threshold equal to 485ms, for an increasing number of ACs and for various KPI summary reporting periods in SAPIENT.



Fig. 11. Distribution of the per AC application delay, for an increasing number of ACs and for various KPI summary reporting periods in SAPIENT.



Fig. 12 - Lorenz curve of the average number of served ACs per GS in a scenario having 200 ACs.

SAPIENT will keep 87% of ACs in ideal conditions (using a 30s reporting period), against a 67% of the baseline, resulting in a 30% increase, or 24 more ACs. Therefore, SAPIENT reduces the outage, minimizing it when the reporting period is 30s.

In Fig. 11 we show the distribution of the end-to-end delay (measured at the application level) for four values of the SAPIENT reporting period, when the number of ACs ranges from 80 to 200. Each bar represents the interval between the 90th (upper edge) and 10th (lower edge) percentiles. The solid line marks the median (50th percentile), and the two dashed lines mark the lower and upper quartiles (i.e., 25th and 75th percentiles). This graph allows one to estimate the expected frame loss rate (due to missed deadlines) when a given delay threshold (to be read on the y axis) is imposed on the system. For instance, a delay threshold of 1s would generate more than 50% frame loss rate in a baseline system with 200 ACs, and 10% loss rate in a SAPIENT-enabled system with the same number of ACs, configured with a 15s or 30s reporting period. The perfomance improvement we described so far is achieved by spreading traffc load as evenly as possible over the network. To demonstrate this fairness, in Fig. 12 we show the Lorenz curve of the number of served ACs per LDACS antenna, in both the baseline scenario and the SAPIENT one configured with a 15s recording period. The dashed line is the line of maximum fairness, i.e. wherein every antenna serves the same number of ACs, and the closer the curve is to this line, the fairer the system is. To further confirm this, we computed Jain's index of fairness for the above two scenarios, which is 0.79 for the baseline and 0.91 for SAPIENT, over a scale of 1.

C. Scenario 2: vertical handover

In this scenario, we analyze the performance of a VH mechanism that leverages performance-drop maps. We remove from the deployment of Fig. 8 the rightmost 8 LDACS antennas, thus modeling the rightmost half of the figure as oceanic airspace, which is only covered by SATCOM. We simulate 100 ACs moving in the same way as in Scenario 1 and running the same applications. Two performance drop zones of 300×500 km, are configured within the floorplan, which affect the SATCOM DL only, by reducing its SNR by 30dB. LDACS SNR is instead unaffected. We set three baselines: in the first two, which we call SATCOM only and LDACS only, the AC uses the same DL, respectively SATCOM and LDACS, for the whole simulation, only performing HHs. In the third one, called local SAPIENT, the AC performs VHs based on a best-SNR policy (i.e., using only local information). The global SAPIENT solution instead leverages the performance-drop map, i.e. *global* information made available by the SAPIENT server. We assume that the performance-drop map is sent every 60 seconds to ACs that are within 100 km of the borders of the drop zone, as a KPI summary occupying 1Kbyte, which adds a negligible traffic. In both the localand global-SAPIENT scenarios ACs generate a KPI report of 15 KPIs every 15 seconds.

Fig. 13 shows the connection availability, i.e. the fraction of time during which a connection is available, in the three baselines and with the *global* SAPIENT option. In a real system, this figure must be as close to 100% as possible, any small deviation representing a possible safety risk (e.g., loss of connectivity with

RPAS). The LDACS only figure is unacceptably small, since part of the floorplan has no LDACS coverage. The SATCOM only is closer to 100%, the difference being given by the fraction of time when SATCOM connectivity is lost due to a performance drop. Using SAPIENT, instead, increases the availability over both baselines. The local SAPIENT solution senses degradation of LDACS signal (by measuring SNRs at the AC), compares it to the SATCOM signal and requests a proactive VH, in a make-before-break approach, whenever the AC is about to enter oceanic airspace. The global SAPIENT solution adds a make-beforebreak VH before the AC enters an unanticipated performancedrop zone, due to the information obtained via global context information, i.e. performance-drop maps. To appreciate the difference, Fig. 14 zooms in the previous figure, showing that this brings the availability to a full 100%, which is not achieved instead with the local SAPIENT version. The perfect availability exhibited by the global SAPIENT solution is to be taken with a grain of salt, because it is obtained in a scenario simulated for 24 hours (a relatively short time for such measures), with no other potential sources of external interference that could affect the it. Thus, rather than expectable absolute figures, the numbers of Fig. 13 and 14 are meant to reflect the relative increase in availability due to the make-before-break SAPIENT approach, obtained by













leveraging global information, all else being equal.

Fig. 15 shows the average frame delay. LDACS has smaller latency than SATCOM. The average frame delay with both versions of SAPIENT is a weighted average of LDACS's and SATCOM's, the weights being given by the fraction of simulated time the AC spends using either DL. This fraction differs negligibly in the two SAPIENT versions, since the only difference is that the global SAPIENT solution performs a *proactive* VH to LDACS before entering the performance drop zone, bringing forward by a few seconds the same decision that the local SAPIENT solution would make *reactively* anyway. A few seconds' lead cannot make any visible difference over a simulated time of 24h.

VI. CONCLUSION AND FUTURE WORKS

This paper has described the SAPIENT system, which – to the best of our knowledge – is the first proposal for a distributed monitoring and control system for the FCI. We have presented the system and its actors, showing that it can be supported using existing network technologies, acting mostly at the application level and using standard software technologies, without requiring a costly hardware infrastructure. We have shown that SAPIENT brings considerable benefits to ACs, reducing their frame loss, balancing load across multiple cells, and increasing the connection availability by enabling a proactive, make-before-break approach to vertical handover.

The benefits of SAPIENT are linked to its role as an enabler of further services. While this paper explores benefits for the ACs, other entities may leverage the SAPIENT infrastructure: DLSPs, which may use it for real-time monitoring of their own infrastructure; ANSPs, which may use it for trajectory optimization (e.g., using DL QoS as a constraint in trajectory building); security agencies, which may mine the database on the SAPIENT server for anomalies and possibly discover jamming attacks or other security threats. Researchers and practitioners in the field of air traffic management could use the SAPIENT server database as an authoritative source of data to validate and test their models. Moreover, the SAPIENT system can be extended to serve other, related purposes: for instance, messages could be added to allow the storage of digital records, or other certified data from ACs, in the SAPIENT server. This would be useful when investigating accidents. Research on these aspects is part of the ongoing work.

Parallel to the above research directions, the implementation logic itself of the SAPIENT system requires further investigation. We are currently investigating several alternatives for the deployment of SAPIENT functions. These alternatives imply different performance at different communication and implementation costs, and may require a varying degree of change in the peering relationships between entities in the aeronautical communications world (from none to severe). The choice of which entity performs which functions (e.g., if link optimization is made at the SAPIENT server, at the AC, or at both) also impacts the kind of algorithms that can be run to make these decisions. For instance, the SAPIENT server may make a handover decision for *several ACs* simultaneously, possibly using mathematical programming techniques to enable near-perfect load balancing, whereas decisions.

sions made at an AC may not achieve the same objective or require the same logic.

ACKNOWLEDGMENT

This work was partially supported by the SESAR Joint Undertaking under grant agreement No 699328 under European Union's Horizon 2020 research and innovation programme. The authors would also like to thank the other project partners.

This work was partially supported by the Italian Ministry of Education and Research (MIUR) in the framework of the CrossLab project (Departments of Excellence).

REFERENCES

- Future Communications Infrastructure (FCI), Step 2 Technology Assessment Results, v1.0P, 29/10/2007.
- [2] Jacob, R. P. Sirigina, A. S. Madhukumar, V. A. Prasad, "Cognitive radio for aeronautical communications: A survey", IEEE Access, vol. 4, pp. 3417-3443, 2016.
- [3] VDL mode 2 Capacity and Performance Analysis, PDF available at http://tinyurl.com/y97rtd6k.
- [4] M. Sajatovic, et al., "Updated LDACS1 System Specification", SESAR JU, Brussels, Belgium.
- [5] G. Frontera, J. A. Besada, J. Lopez-Leones. "Generation of Aircraft Intent Based on a Microstrategy Search Tree". IEEE Transactions on Intelligent Transportation Systems, (2016), pp. 1-17
- [6] Minh-Ha Nguyen, Sameer Alam, "Airspace Collision Risk Hot-Spot Identification using Clustering Models", IEEE Transactions on Intelligent Transportation Systems (Volume: 19, Issue: 1)
- [7] Atm S. Alam, *et al.* "Optimal Datalink Selection for Future Aeronautical Telecommunication Networks", IEEE Transactions on Aerospace and Electronic Systems (Volume: 53, Issue: 5, Oct. 2017).
- [8] "Manual on Required Communication Performance (RCP)", International Civil Aviation Organization (ICAO), 2016.
- [9] A. S. Alam, Y.-F. Hu, P. Pillai, K. Xu, and J. Baddoo, "Optimal Datalink Selection for Future Aeronautical Telecommunication Networks," IEEE Trans. on Aerospace and Electronic Systems, Vol. 53, Issue 5, Oct. 2017.
- [10] K. Ongkasae, H. Nakazato, T. Koga, X. Lu, "QoS Implementation in System Wide Information Management (SWIM) Network Model", in proceedings of IEEE Twelfth International Symposium on Autonomous Decentralized Systems (ISADS), 25-27 March 2015.
- [11] B. Haindl, M. Lindner, "Ground based lisp for multilink operation in ATN/IPS communication infrastructure", in proc. 2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC), 25-29 Sept. 2016.
- [12] W. Kampichler, M. Lindner, B. Haindl, D. Eier and B. Gronau, "LISP: A novel approach towards a future communication infrastructure multilink service," DASC 2013, East Syracuse, NY, 2013.
- [13] OMNeT++ website, https://omnetpp.org, accessed Nov. 2017.
- [14] S. Fotheringham, and P. Rogerson, Spatial analysis and GIS: CRC Press, 2013.
- [15] P. Praveen, Ch. Jayanth Babu, B. Rama, "Big data environment for geospatial data analysis", in proc. of ICCES, Coimbatore, India, 21-22 Oct. 2016.
- [16] M. S. Ben Mahmoud, A. Pirovano, N. Larrieu, "Aeronautical communication transition from analog to digital data: A network security survey", Comput. Sci. Rev., vol. 11, pp. 1-29, May 2014.
- [17] M. Strohmeier, et al., "On Perception and Reality in Wireless Air Traffic Communications Security", 2016.
- [18] C. Bauer, M. Zitterbart, "A Survey of Protocols to Support IP Mobility in Aeronautical Communications", IEEE Communications Surveys & Tutorials, vol. 13, no. 4, pp. 642-657, 2011.
- [19] VDL mode 2 Capacity and Performance Analysis, pdf available at <u>https://tinyurl.com/y97rtd6k</u>
- [20] Amazon storage services, <u>https://aws.amazon.com/s3/pricing/</u>, accessed April 2019.
- [21] V. Devarapalli, R. Wakikawa, A. Petrescu, P. Thubert, "Network Mobility (NEMO) Basic Support Protocol", RFC 3963, Jan. 2005.
- [22] S. La Barbera, et al. "SESAR Exploratory Research SAPIENT Project Overview", in Proc. 23rd Ka and Broadband Communications Conference, Trieste, Italy, Oct. 4-6 2017, ISBN:ISSN-2573-6124

- [23] A. Virdis, G. Stea, S. La Barbera, R. Winkler, "SAPIENT-Simulator Modelling and Architecture", in Proc. 23rd Ka and Broadband Communications Conference, Trieste, Italy, Oct. 4-6 2017, ISBN:ISSN-2573-6124.
- [24] M. Bellare, C. Namprempre, "Authenticated encryption: Relations among notions and analysis of the generic composition paradigm", In Proc. of ASIACRYPT '00, Kyoto (Japan), 03–07 December 2000.
- [25] M. Ehammer, T. Graupl, C.H. Rokitansky, T. Brikey, T, "Security consideration for IP based aeronautical networks," In *Proc.* 27th *IEEE/AIAA Digital Avionics Systems Conference*, Saint Paul (MN), 26–30 Oct. 2008.
- [26] International Civil Aviation Organization, Manual for the ATN using IPS Standards and Protocols. ICAO Document 9896, September 2011.
- [27] J.M.R. Bejarano, A. Yun, B. De La Cuesta, "Security in IP satellite networks: COMSEC and TRANSEC integration aspects," in Proceedings of the 6th ASMS and 12th SPSC workshop Baiona (Spain), 5-7 Sept. 2012.
- [28] International Organization for Standard. Information technology Security techniques Authenticated encryption. ISO/IEC 19722:2009.
- [29] International Organization for Standard. Information technology Security techniques – Key management – Part 1: Framework. In ISO/IEC 11770-1:2010
- [30] International Organization for Standard. Information technology Security techniques – Key management – Part 2: Mechanisms using symmetric techniques. In ISO/IEC 11770-2:2018.
- [31] International Organization for Standard. Information technology Security techniques – Key management – Part 3: Mechanisms using asymmetric techniques. In ISO/IEC 11770-3:2015.
- [32] Communications Operating Concept and Requirements for the Future Radio System, EUROCONTROL/FAA Future Communications Study Operational Concepts and Requirements Team, May 01 2007.
- [33] J. M. R. Bejarano, A. Yun, B. De La Cuesta "Security in IP satellite networks: COMSEC and TRANSEC integration aspects". In 2012 6th Advanced Satellite Multimedia Systems Conference (ASMS) and 12th Signal Processing for Space Communications Workshop (SPSC) (pp. 281-288), September 2012.
- [34] International Civil Aviation Organization. Manual for the ATN using IPS Standards and Protocols. In ICAO Document 9896, September 2011.
- [35] H. Teso "Aircraft hacking: Practical aero series." The Fourth Annual Hack In The Box Security Conference in Europe, 10-11 April 2013. <u>https://www.youtube.com/watch?v=wk1jIKQvMx8</u>.
- [36] S. Checkoway, et al. "Comprehensive Experimental Analyses of Automotive Attack Surfaces." USENIX Security Symposium, 2011.
- [37] P.T. Devanbu, S. Stubblebine "Software engineering for security: a roadmap". In Proceedings of the Conference on the Future of Software Engineering (pp. 227-239), ACM, May 2000.
- [38] European Network and Information Security Agency. Secure Software Engineering Initiatives — Listing SSE Initiatives across Europe and Abroad. 2011 <u>https://www.enisa.europa.eu/publications/secure-softwareengineering-initiatives</u>

Antonio Virdis is Assistant Professor at the University of Pisa, where he obtained his MSc degree in Computer System Engineering in 2011, and his PhD in Information Engineering in 2015. His research interests include Quality of Service, scheduling and resource allocation in wireless networks, network simulation and performance evaluation. He has been involved in national, EU-funded and industry-funded research projects. He coauthored seven patents and more than 40 peer-reviewed papers.

Giovanni Stea is Associate Professor at the Department of Information Engineering of the University of Pisa, Italy, where he also got his PhD in 2003. His current research interests include Quality of Service and resource allocation in wireline and wireless networks, performance evaluation through simulation and analytical techniques, traffic engineering. In these fields he has coauthored more than 100 peer-reviewed papers and 16 patents. He has been involved in national and European research projects, and he has led joint research projects with industrial partners. He has served as a member of the technical and/or organization committees for several international conferences, including SIGCOMM, WoWMoM, and VALUETOOLS, and he is serving on the editorial board of the Wireless Networks journal. He is a fellow of the European Alliance for Innovation.

Gianluca Dini is Full Professor at the Department of Information Engineering of the University of Pisa. His principal research interests are cybersecurity, with particular reference to information systems, industrial control systems and Internet of Things. He coauthored more than 100 papers on peer-reviewed journals and conferences. He has been responsible for many national, international, and privately-funded research projects.