



Towards a Novel UAV Position Tracking and Reporting System for Very Low Level Airspace

Bruno Chianca Ferreira, Guillaume Dufour, Guthemberg Silvestre

► To cite this version:

Bruno Chianca Ferreira, Guillaume Dufour, Guthemberg Silvestre. Towards a Novel UAV Position Tracking and Reporting System for Very Low Level Airspace. ERTS2022, Jun 2022, Toulouse, France. hal-03450168v2

HAL Id: hal-03450168

<https://hal.science/hal-03450168v2>

Submitted on 2 Jun 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Towards a Novel UAV Position Tracking and Reporting System for Very Low Level Airspace

Bruno Chianca Ferreira¹, Guillaume Dufour², and Guthemberg Silvestre³

¹ ENAC/ReSCo, Université de Toulouse, France
`bruno.chianca@enac.fr`

² ONERA/DTIS, Université de Toulouse, France
`guillaume.dufour@onera.fr`

³ ENAC/ReSCo, Université de Toulouse, France
`silvestre@enac.fr`

Short paper about ongoing research.

Abstract. The integration of Unmanned Aircraft Systems into the airspace is a challenging, complex task for both operators and regulators. To ensure a safe and secure UAS integration, they should rely on wide variety of dependable sub-systems, including a tracking and position reporting system. This paper proposes a system architecture for tracking and position reporting of UAS in very-low level airspace. The system leverages low-latency communications and distributed computing to offer highly available, consistent tracking information. Preliminary results suggest that our approach is especially useful for the traffic management and operations in densely populated areas.

Keywords: utm, uas, edge, emulation, ads-b, mobile, ad-hoc

1 Introduction

Recently, there has been increasing interest in Unmanned Aircraft Systems (UAS) for many real-life, civilian applications [1], including real-time monitoring, remote sensing, search and rescue, agriculture services and delivery of goods. In addition, a rapid growth in the number of drones in activity is expected, especially at very low-altitude of airspace with high demand in densely populated areas [2]. For instance, a recent study [3] forecasts that 98% of the aircraft operating in the airspace of Paris by 2035 will be autonomous. The integration of UAS will certainly result in fundamental changes of the air traffic management (ATM). Consequently, there is a need for the design of an Unmanned Traffic Management (UTM) system that will be able to manage a large number of autonomous vehicles in a safe and efficient manner.

One of the key components of the emerging UTM system is the the capability to track the vehicles' positions. Currently, tracking and position reporting of aircraft relies on long-range radio communications. Unfortunately, a large

number of vehicles communicating through conventional radio frequencies (RF) in densely populated areas is likely to incur mutual interference and, consequently, poor performance. One way of reducing interference is to limit the range of the wireless radio but, which in conjunction with the limitation introduced by the buildings, it can render systems such as Automatic Dependent Surveillance–Broadcast (ADS-B) unfeasible. Furthermore, it remains unclear how the current ATM systems could be redesigned to enforce highly available, consistent position reporting to meet safety and performance requirements of novel UTM systems.

This study aims to investigate the feasibility of a novel tracking and position reporting system that could replace ADS-B in the context of UTM and UAS. The rationale behind it is two-fold. First, the new system should allow small autonomous vehicles to communicate with low-cost, low-footprint, short range wireless interfaces and broadcast real-time data with local stations in the vicinity. Second, the system should rely on distributed computing techniques to share positioning data of vehicles with high availability and strong consistency, allowing even aircraft which are beyond line of sight to have updated information, ensuring proper functionality of their own applications.

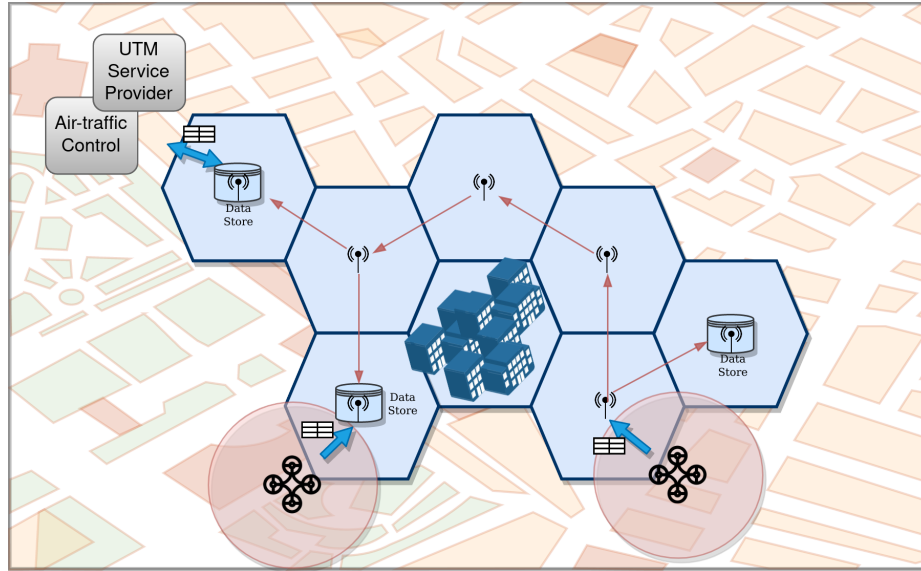


Fig. 1: Distributed UTM Data Service

In this paper, we introduce a system architecture to reach the aforementioned design goals by deploying a distributed tracking and position reporting system throughout stations of Radio Access Network cloudlets, as depicted in Figure 1. The architecture provides performance requirements, such as low-latency, high-

throughput data exchange thanks to 5G's communication capability. To meet consistent, highly available positioning data sharing, our architecture leverages the functionalities of a strongly consistent distributed key-value store. Preliminary results based on mobile systems' emulation suggest that this architecture meets both performance and safety requirements of emerging UTM's tracking and position reporting system.

2 State of the Art

2.1 Unmanned Traffic Management

The rapid expansion of the market of small UAS has created new risks for civil aviation [4][3]. That increases the need for regulation and deployment of a fully functional UTM [5]. Some governmental agencies have been more prominent in UTM proposals, such as USA with NASA/FAA and European Union's SESAR/U-space. In [6], the author described NASA's research initiative related to the concept of operations (ConOps). NASA established a research platform which is used together with their partners to test and evaluate the challenges and solutions proposed for each of the technical capabilities associated with UTM. For NASA/FAA the UTM is set to evolve incrementally according to technical capability levels (TCL), and some research works have been conducted in order to verify the full development and verification of TCL levels. In [7] flight tests demonstrating the most basic TCL levels 1 and 2 were performed even with beyond visual-line-of-sight (BVLOS) for unpopulated areas. BVLOS in populated areas is introduced in [8], where the authors performed a flight demonstration of TCL3. In [9] the author demonstrates TCL4 capabilities using simulation tools, and gave a high emphasis in safety by describing in details the SAFE50 architecture.

The framework on how a UTM system should work and the interaction between the stakeholders is not yet finalized [10] and many aspects are not clear in order to achieve the highest levels of autonomy. However, the main proposed architectures are composed of distinctive applications [11] such as sense and avoid [12–15], path planning [16–19], which rely on the fact that trustable data can be shared among stakeholders with low latency.

2.2 Aircraft Surveillance

A position tracking and reporting system is a crucial system for aircraft surveillance. The Automatic Dependent Surveillance–Broadcast (ADS-B) is a service present in aircraft⁴ that broadcasts relevant unencrypted data in real-time. It aids radars in keeping updated and correct information about all the aircraft currently flying. With such information about all aircraft, air traffic management (ATM) authorities can develop applications to improve security, and efficiency of air transportation system and airports. Albeit not mandatory like *ADS-B Out*,

⁴ Mandatory in many countries depending on aircraft weight or airspace class.

other aircraft can also receive broadcasts with *ADS-B In* and use the information to build knowledge about its surroundings in order to improve conflicts avoidance. Aircraft equipped with ADS-B equipment, periodically broadcast their call sign, latitude and longitude, ground speed and flight number with signals modulated in 1090 MHz. Considering the importance systems like ADS-B have for ATM, a similar technology is also expected to play an important role in UTM. Unlike large aircraft, small UAVs are harder to be detected with radars, increasing the importance of a *ADS-B like* system. Previous works have proposed ADS-B alternatives in the context of UTM as explored in [20]. The authors in [21] and [22] proposed an architecture based in low-cost radio transferring data to a central cloud, which does not solve the problem of fast replication across the urban area for low latency and availability to all stakeholders. In [22] the author proposes an ADS-B like system for UTM based on APRS (Automatic Packet Reporting System), which is an approach different to what is proposed in this work, since it is based on relatively long range radio (40Km) with restricted throughput. In [13], the author proposed a solution to inject the position data in the SSID of WiFi and the results has shown that the proposed setup yielded reasonably low latency (under 125 ms) and acceptable throughput (4 messages/s) under an experimental setup. By having a secure ADS-B like data layer, applications such as presented in [15] where the authors proposes a sense and avoid system that used ADS-B data as input, can be built.

3 A Novel System Architecture for Tracking and Position Reporting

Our system architecture is designed to enable fault-tolerant and efficient data sharing for UTM stakeholders. The main blocks of this architecture are shown in Figure 2. It is built on top of two key components: a distributed, strongly consistent key-value store and the next-generation cellular communication technology. The distributed key-value store is composed of distributed UTM Data Service (UDS) units spread across the urban area as shown in Figure 1. We assume that UDSs are deployed in federated cloudlets interconnected through low-latency mobile network, such as 5G or 6G[23]. In this section, we describe the design of our architecture by focusing on the SMR protocol and the architecture’s API endpoints. We skip further details about the mobile network due to space constraint.

3.1 State Machine Replication

To enable fault-tolerant, highly available key-value store, UDSs run state machine replication (SMR) protocol[24]. SMR is a technique to implement fault-tolerant service by replicating servers and coordinating client interactions with server replicas. In particular, stakeholders (e.g., service providers, operators and vehicles) are clients and servers/replicas are UDSs. Our architecture leverages

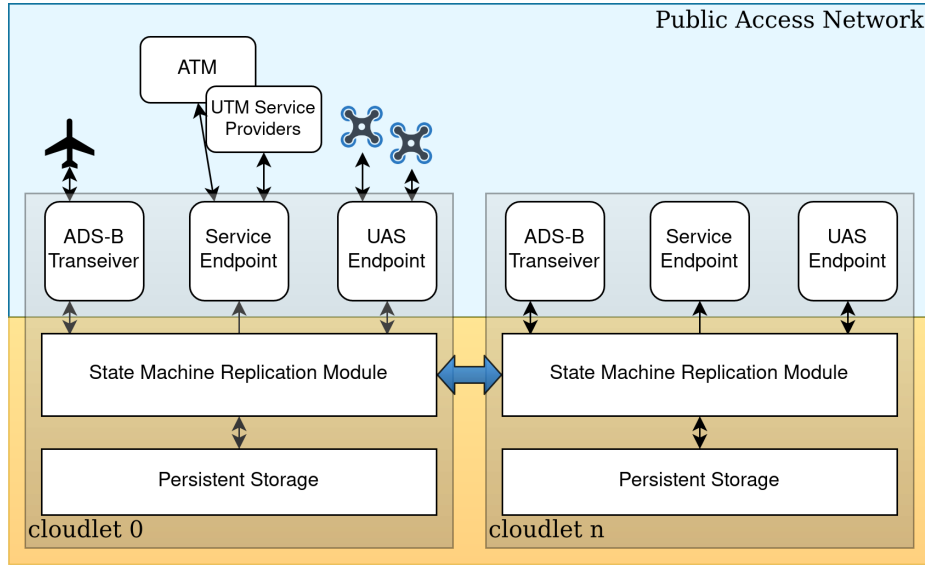


Fig. 2: Our System Architecture

*etcd*⁵, a distributed, strongly-consistent key-value store, to run SMR. This distributed approach solves important problems of ADS-B [26], such as lack of fault tolerance, data inconsistency, and unpredictable availability of shared data.

3.2 API Endpoints

The API endpoints are used to put or get data from the system.

- ADS-B Transponder - A ADS-B receiver connected to the system can put data from aircraft.
- Service Endpoint - The end-point that can be used by the UTM service providers or operators to get the current and past state of the airspace via restful API, time series or stream.
- UAS Endpoint - Used by UASs to insert broadcast data into the system.

4 Experimental Evaluation

This section presents the performance evaluation of our system architecture using MACE [27], an emulation framework to study, design and evaluate mobile Ad-hoc applications. Figure 3 shows the architecture of our emulation with MACE detailing how the main blocks communicate with each other.

⁵ <https://etcd.io/>

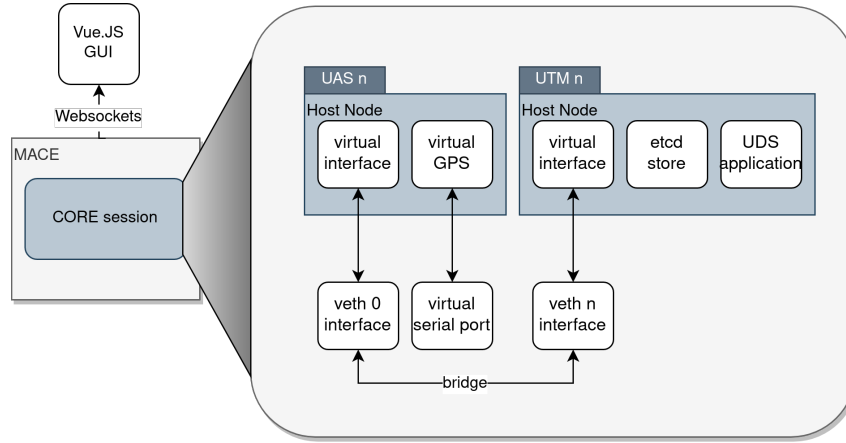


Fig. 3: Emulation Architecture

4.1 Cloudlets and Aircraft

MACE uses CORE as network emulator, and it emulates each network instance as Linux namespaces serving as minimal containers. Each aircraft client application runs inside these namespaces, and they communicate via *veth* network interfaces with connectivity controlled by the distance between the nodes. The cloudlets are also emulated in such namespaces with a server running the UAS endpoint and interfacing instances of etcd running in the same namespaces. The emulated scenario is run over an area of one square kilometre.

4.2 Mobility

The *Random Waypoint* mobility model was adopted for the experiment. In this model each aircraft receives a random waypoint and a random velocity to simulate a mission's objective. The movement of the aircraft is emulated in MACE, and the real-time position is injected directly in the network emulator so that it is reflected in the network connectivity. The position is made available to the applications running in the virtual aircraft via UNIX sockets.

4.3 UAS Broadcasts

For the payload reporting, a client running in each aircraft broadcasts a JSON object containing the position and additional data via IPV4 UDP sockets using the emulated Ad-hoc wireless links. The payload includes also an unique message ID, timestamp, an aircraft ID, velocity and status. The unique message ID is used to verify that the message was received and the timestamp used to calculate the latency between the broadcast and full replication.

4.4 UDS Cloudlet

To simulate the UDS cloudlets, a server representing them is deployed in each emulated cloudlet. The servers which are in range of a broadcasted message, will receive the new data in the UAS endpoint and replicate it to the other towers which are not in range by using the state machine replication layer. Data is also written in persistent storage for posterior analysis. The consistent replication ensures that the data is reliable across all participant cloudlets, so that distributed third part actors can see this reporting system as a trusted oracle for UTM applications such as asset managers, schedulers and collision avoidance path planners. When a message is fully replicated among all replicas, a callback function records the time stamp so that the total latency can be calculated.

All the experiments were performed on a Linux server whose configuration is described in Table 1. Our empirical evaluation is split in the scenarios detailed in Table 2.

4.5 Preliminary Results

In the first scenario, the main goal was to measure the latency between the data broadcasting and full replication when there are no network losses between cloudlets or in the UAS communication interfaces. One of the side effects of the strong consistency is the communication complexity consequence of the multistep consensus algorithm. So, in addition of the broadcast latency, it is expected also a component for the replication latency.

We assume that ADS-B broadcasts 2 messages per second since detect and avoid / path planning algorithms usually run with frequencies ranging from 1Hz to 2Hz. A previous experiment with only one fixed UAS was run to measure the replication overhead. Running such scenario yielded an replication overhead of 10ms, which is acceptable but is closely related to the number of replicas and, more considerably, to the total coverage area. Larger areas would require more communication hops even if the number of replicas is left unchanged, hence increasing the total latency.

Table 1: Test Platform

Processor Manufacturer / Model:	Intel i7-8550U @ 4GHz
Number of cores	8
Memory	16GB DDR4
Emulator	MACE
Operating System	Linux Mint 20 x86 64bits
Routing Protocol	B.A.T.M.A.N IV

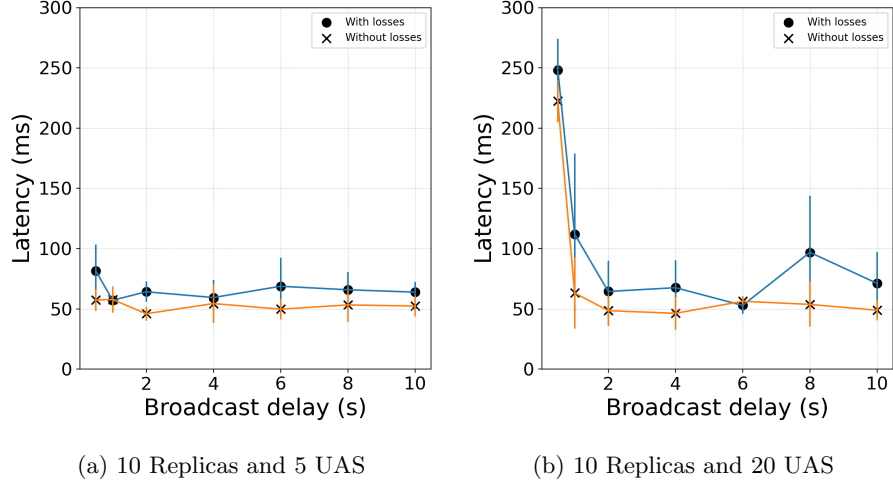


Fig. 4: Total replication latency with increasing broadcast delay

In the second scenario, losses are added to emulate a more realistic scenario, but the values of losses, latency and bandwidth are closely related to communication technology adopted. Due to the random nature of the mobility model, each emulation session yields different results, so the median and standard deviation of 5 sessions are shown. Figure 4(a) illustrate the results for 5 UAS, while Figure 4(b) the results for 20 UAS for both scenarios.

The impact of the introduction of losses is perceived differently in the UAS links and intra-cloudlets links. Since in this implementation the UAS broadcasts data via UDP links (more suitable for mobile and lossy links), lost messages do not impact the broadcast latency because there are no timeouts and retransmissions. Since *etcd* is implemented to communicate via TCP, the losses increase the overall replication latency as shown in the Figure 4. It is important to notice in Figure 4(b) that increasing the number of aircraft while leaving the delay

Table 2: Emulation Scenarios

Setting	Without Losses	With Losses
Cloudlet Range	250m	250m
Number of cloudlets	4	10
Cloudlet Bitrate / Delay	433.3Mbps / 1000us	433.3Mbps / 1000us
UAS Range	90m	250m
Number of UASs	5 and 20	5 and 20
UAS Bitrate / Delay	54Mbps / 3000us	54Mbps / 3000us
Jitter	0 us	2 us
Error rate	0%	1%

between broadcasts low increase significantly the end-to-end latency. This could mean that in order for such platform scale to more representative scenarios of a crowded airspace, the number of replicas cannot be too high. It is also important to highlight that since these were performed with an emulator, all software components related to all cloudlets and UAS, such as the clients, servers and etcd, are running in the same computer and competing for resources, and that the its state can affect the results. It is therefore unfeasible to emulate scalability tests on an emulation platform, since they consume too much computer resources, and more tests with a simplified replication model running on a scalable simulator are required.

5 Discussion, Limitation and Future Work

The emulator helped us to implement and evaluate a very first prototype of our proposal, providing quick, yet preliminary results. For instance, latency measurements with many applications running on a single computer might skew the results. The server implementation (i.e., an UDS unit) used a callback function from a `etcd` library that was not designed to have low-latency capability, specially when running multiple requests simultaneously. In this preliminary study, we also overlooked the impact of the number of replicas in the overall latency while keeping a large number of access points. We also ignored the impact of different failure models, including a Byzantine one. One important aspect of such deployments is how they will act in presence of malicious actors, which is a reasonable assumption for a open air platform with competing actors. We believe that the target system could tolerate attacks such as Sybil attacks [28]. Alternatively, we could consider a distributed data store based on blockchain technique in order to cope with Byzantine faults.

For future work, scenarios with larger areas, different number of replicas are required. This would ensure a more clear view of the latency impact of the replication quorum. Furthermore, scalability studies are also required to simulate scenarios with hundreds of UAS. A fast simulator would also allow the study of optimal placement of such replicas in different realistic maps.

6 Conclusions

This work presents a novel tracking and position reporting system to enable emerging UTM services for autonomous vehicles operating at very low level airspace. Our preliminary performance evaluation suggests that the latency measurements yielded results that are suitable for building a system architecture for UTM in densely populated areas. The highly available, distributed data store allows even vehicles that are hidden behind constructions to be spotted by all stakeholders in a consistent way. In contrast to the current ADS-B, our architecture enable fast, fault-tolerant data sharing by design.

References

1. H. Shakhathreh, A. H. Sawalmeh, A. Al-Fuqaha, Z. Dou, E. Almaita, I. Khalil, N. S. Othman, A. Khreishah, and M. Guizani, "Unmanned aerial vehicles (uavs): A survey on civil applications and key research challenges," *Ieee Access*, vol. 7, pp. 48 572–48 634, 2019.
2. S. J. Undertaking, "European drones outlook study," *SESAR Joint Undertaking: Brussels, Belgium*, 2016.
3. R. Rumba and A. Nikitenko, "The wild west of drones: A review on autonomous-UAV traffic-management," *2020 International Conference on Unmanned Aircraft Systems, ICUAS 2020*, pp. 1317–1322, 2020.
4. J. Lundberg, K. L. Palmerius, and B. Josefsson, "Urban Air Traffic Management (UTM) Implementation in cities - Sampled side-effects," *AIAA/IEEE Digital Avionics Systems Conference - Proceedings*, vol. 2018-Septe, no. September, 2018.
5. M. Elsayed and M. Mohamed, "The impact of airspace regulations on unmanned aerial vehicles in last-mile operation," *Transportation Research Part D: Transport and Environment*, vol. 87, no. August, p. 102480, 2020. [Online]. Available: <https://doi.org/10.1016/j.trd.2020.102480>
6. P. Kopardekar, J. Rios, T. Prevot, M. Johnson, J. Jung, and J. E. Robinson, "Unmanned aircraft system traffic management (UTM) concept of operations," *16th AIAA Aviation Technology, Integration, and Operations Conference*, pp. 1–16, 2016.
7. J. Rios, D. Mulfinger, J. Homola, and P. Venkatesan, "NASA UAS traffic management national campaign: Operations across Six UAS Test Sites," *AIAA/IEEE Digital Avionics Systems Conference - Proceedings*, vol. 2016-December, pp. 1–6, 2016.
8. J. Rios, A. Aweiss, J. Jung, J. Homola, M. Johnson, and R. Johnson, "Flight Demonstration of Unmanned Aircraft System (Uas) Traffic Management (utm) at Technical Capability Level 3," *Aiaa Aviation 2020 Forum*, vol. 1 PartF, 2020.
9. A. Chakrabarty and C. Ippolito, "Autonomous flight for multi-copters flying in UTM -TCL4+ sharing common airspace," *AIAA Scitech 2020 Forum*, vol. 1 PartF, no. January, 2020.
10. T. McCarthy, L. Pforte, and R. Burke, "Fundamental elements of an urban UTM," *Aerospace*, vol. 7, no. 7, 2020.
11. N. S. Labib, G. Danoy, J. Musial, M. R. Brust, and P. Bouvry, "A multilayer low-altitude airspace model for UAV traffic management," *DIVANet 2019 - Proceedings of the 9th ACM Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications*, pp. 57–63, 2019.
12. D. Scott, M. Radmanesh, M. Sarim, A. Deshpande, M. Kumar, and R. Pragada, "Distributed bidding-based detect-and-avoid for multiple unmanned aerial vehicles in national airspace," in *2019 International Conference on Unmanned Aircraft Systems, ICUAS 2019*, 2019, pp. 930–936.
13. F. Minucci, E. Vinogradov, and S. Pollin, "Avoiding Collisions at Any (Low) Cost: ADS-B like Position Broadcast for UAVs," *IEEE Access*, vol. 8, pp. 121 843–121 857, 2020.
14. J. H. Park, S. C. Choi, J. Kim, and K. H. Won, "Unmanned Aerial System Traffic Management with WAVE Protocol for Collision Avoidance," *International Conference on Ubiquitous and Future Networks, ICUFN*, vol. 2018-July, pp. 8–10, 2018.
15. Yucong Lin and S. Saripalli, "Sense and avoid for unmanned aerial vehicles using ads-b," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 6402–6407.

16. J. Besada, I. Campaña, L. Bergesio, A. Bernardos, and G. de Miguel, "Drone flight planning for safe urban operations: UTM requirements and tools," *Personal and Ubiquitous Computing*, pp. 924–930, 2020.
17. Q. Tan, Z. Wang, Y. S. Ong, and K. H. Low, "Evolutionary optimization-based mission planning for UAS traffic management (UTM)," *2019 International Conference on Unmanned Aircraft Systems, ICUAS 2019*, pp. 952–958, 2019.
18. P. B. Sujit and R. Beard, "Multiple uav path planning using anytime algorithms," in *2009 American Control Conference*, 2009, pp. 2978–2983.
19. A. Chakrabarty, V. Stepanyan, K. Krishnakumar, and C. Ippolito, "Real-time path planning for multi-copters flying in UTM-TCL4," *AIAA Scitech 2019 Forum*, no. January, 2019.
20. G. L. Orrell, A. Chen, and C. J. Reynolds, "Small unmanned aircraft system (SUAS) automatic dependent surveillance-broadcast (ADS-B) like surveillance concept of operations: A path forward for small UAS surveillance," *AIAA/IEEE Digital Avionics Systems Conference - Proceedings*, vol. 2017-Septe, 2017.
21. C. E. Lin, "An ADS-B like communication for UTM," *Integrated Communications, Navigation and Surveillance Conference, ICNS*, vol. 2019-April, pp. 1–12, 2019.
22. Y. H. Lin, C. E. Lin, and H. C. Chen, "ADS-B Like UTM surveillance using APRS infrastructure," *Aerospace*, vol. 7, no. 7, pp. 1–14, 2020.
23. M. M. Azari, S. Solanki, S. Chatzinotas, O. Kordheli, H. Sallouha, A. Colpaert, J. Fabian, M. Montoya, S. Pollin, A. Haqiqatnejad, A. Mostaani, E. Lagunas, and B. Ottersten, "Evolution of Non-Terrestrial Networks From 5G to 6G : A Survey," pp. 1–35.
24. F. B. Schneider, "Implementing Fault-Tolerant Services Using the State Machine Approach: A Tutorial," *ACM Computing Surveys (CSUR)*, vol. 22, no. 4, pp. 299–319, 1990.
25. L. Larsson, W. Tärneberg, C. Klein, E. Elmroth, and M. Kihl, "Impact of etcd deployment on kubernetes, istio, and application performance," *Software: Practice and Experience*, vol. 50, no. 10, pp. 1986–2007, 2020. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/spe.2885>
26. Y. Kim, J. Y. Jo, and S. Lee, "ADS-B vulnerabilities and a security solution with a timestamp," *IEEE Aerospace and Electronic Systems Magazine*, vol. 32, no. 11, pp. 52–61, 2017.
27. B. C. Ferreira, G. Dufour, and G. Silvestre, "Mace: A mobile ad-hoc computing emulation framework," in *2021 International Conference on Computer Communications and Networks (ICCCN)*, 2021, pp. 1–6.
28. B. Yu, C. Z. Xu, and B. Xiao, "Detecting Sybil attacks in VANETs," *Journal of Parallel and Distributed Computing*, vol. 73, no. 6, pp. 746–756, 2013.