international journal of satellite communications and networking

Reliable M2M/IoT data delivery from FANETs via satellite

Journal:	*International Journal of Satellite Communications and Networking*
Manuscript ID	SAT-17-0073.R1
Wiley - Manuscript type:	Special Issue Paper
Date Submitted by the Author:	17-May-2018
Complete List of Authors:	Bacco, Manlio; CNR, ISTI Colucci, Marco; CNR, Italy, Institute of Information Science and Technologies (ISTI) Gotta, Alberto; ISTI-CNR, ICT; Kourogiorgas, Charilaos; National Technical University of Athens, Electrical and Computer Engineering Panagopoulos, Athanasios; National Technical University of Athens, Electrical and Computer Engineering
Keywords:	FANET, satellite, channel modeling, CoAP, MQTT, M2M, IoT, random access
Abstract:	The use of Unmanned Aerial Vehicles (UAVs) is rising in several application fields. This work deals with the communication challenges in UAV swarms, or Flying Ad-Hoc Networks (FANET), when taking into account non-line-of-sight scenarios. The use of satellites is a necessity in such operating conditions, thus this work provides architectural considerations and performance assessments when several FANETs share an uplink Random Access (RA) satellite channel, fed with M2M/IoT traffic generated from on-board sensors, to be reliably delivered to a remote ground destination.

For Peer Review Only

SCHOLARONE™
Manuscripts

Received ; Revised ; Accepted

DOI: xxx/xxxx

SPECIAL ISSUE PAPER

Reliable M2M/IoT data delivery from FANETs via satellite

Manlio Bacco*[1] | Marco Colucci[1] | Alberto Gotta[1] | Charilaos Kourogiorgas[2] | Athanasios D. Panagopoulos[2]

[1]Institute of Information Science and Technologies (ISTI), CNR Italy
[2]School of Electrical and Computer Engineering, National Technical University of Athens, Greece

Correspondence
*Corresponding author name: Manlio Bacco.
Email: manlio.bacco@isti.cnr.it

Summary

The use of Unmanned Aerial Vehicles (UAVs) is rising in several application fields. This work deals with the communication challenges in UAV swarms, or Flying Ad-Hoc Networks (FANETs), when taking into account non-line-of-sight scenarios. The use of satellites is a necessity in such operating conditions, thus this work provides architectural considerations and performance assessments when several FANETs share an uplink Random Access (RA) satellite channel, fed with M2M/IoT traffic generated from on-board sensors, to be reliably delivered to a remote ground destination.

KEYWORDS:
FANET, satellite, channel modeling, CoAP, MQTT, M2M, IoT, random access

For Review Only

1 | INTRODUCTION

Nowadays, the use of UAVs in civilian applications is rising[1], attracting an ever increasing attention from both research and industrial fields. While the use of a single UAV can be considered a consolidate use case, the use of multiple units in swarm configurations still requires an active investigation, in order to identify and to solve the open issues. As pointed out in[2], communications, collision-free, and seamless operations are the most challenging issues to be dealt with when thinking of a deployment of a massive number of UAVs. Currently, part of those issues have been limited by the implementation of strict national regulations all across Europe, in order to maintain the so-called *safety of flight levels*[3]. Several application fields would largely benefit of the use of UAVs in a coordinated manner, instead of using a single unit at a time: precision agriculture[4], search-and-rescue, surveillance and monitoring, and goods delivery[5], to cite a few examples. The use of UAVs in a coordinated manner can be referred to as a FANET, in a similar way to the concepts of Mobile Ad-Hoc Network (MANET) and Vehicular Ad-hoc Network (VANET)[6]: anyway, the main differences with those are related to the way UAVs move and to the distances under consideration. With respect to MANETs and VANETs, FANETs exhibit greater moving speeds and 3D movements, which leads to larger distances between the units composing the swarm, spread over a certain area. Such a classification focuses the attention on the communication issues. Moving from communication challenges to the advantages brought by using FANETs, the following ones can be enumerated: (*i*) low-cost operations, because smaller and Commercial Off The Shelf (COTS) UAVs can be used to achieve similar features as larger and more expensive drones; (*ii*) larger coverage because of the possibility of a flight formation that can broadly cover the target area at the same time; (*iii*) the possibility to carry multiple sensor payloads on different vehicles, which reduces the limitations posed by Size, Weight and Power (SWaP) requirements on single units (of paramount importance in UAV deployments); (*iv*) redundancy, because multiple units can carry the same payload in order to protect against the failure of a unit or of a subsystem; (*v*) fault-tolerance, which is inherently provided by the redundancy of the systems. Along with the advantages, peculiar challenges are to be considered, which pose different constraints than those under consideration in MANETs and VANETs, as pointed out in[6]. It is worth noting that the definition of realistic mobility models for UAV swarms[7] is still an open issue in the literature.

In this work, we focus our attention on scenarios in which the data is generated by on-board sensor payloads that sense the environment, then to be delivered to a fixed ground station. A cluster of UAVs can carry multiple different sensors generating and collecting a certain amount of data, similarly to an Internet of Things (IoT) platform[8]. The amount of data depends on the FANET size, which nowadays varies between a few units and tens of them in real deployments (apart from shows, like the one on late 2017 in LA, when Intel flew 300 UAVs[1] in a controlled aerial environment). Those data can be generated according to a time-driven pattern or to a event-driven pattern, according to Machine to Machine (M2M)/IoT traffic patterns: for instance, periodic updates on the swarm status (time-driven traffic), and notifications coming from the on-board sensors (event-driven traffic). In our scenario, a master UAV in the swarm receives fresh data from other UAVs, according to a Publish / Subscribe (PUB/SUB) paradigm, to be forwarded to a remote ground station.

Our interest lies in the scenarios in which the swarm operates in Non-Radio-Line-of-Sight (NRLoS) conditions: therefore, our reference scenario foresees a back-haul via satellite to deliver data in a successful way to the intended destination. We assume DVB-RCS2 compliant satellite user terminals operating in Ku/Ka band. In order to properly take into account the effects of both speed and path loss, an UAV-to-satellite channel model is presented in this work. The channel model for the UAV-to-satellite link takes into account two factors: the mobility of the UAV and its effect on first (cumulative statistics) and second (autocorrelation function) order statistics of rain attenuation, and the height of the UAV for the calculation of rain attenuation, since the length of the slant path affected by rain is different than the length of the slant path from the ground station. The main assumptions for the channel modeling are in what follows: if the height of the UAV is higher than the rainfall height (0^{o}C isotherm), then the rain attenuation is equal to 0 dB; when rain attenuation is induced into the link, it is further assumed that it follows a lognormal distribution with statistical parameters depending on the UAV speed, the UAV height, and the local climatic characteristics; furthermore, the underlined Gaussian process of rain attenuation is a Markov diffusion process which is calculated by the Langevin stochastic differential equation.

One of the most significant metrics for M2M/IoT scenario under consideration in this work is the *completion time*, i.e., the time required to successfully deliver data to the destination. In order to achieve that, we compare the performance achievable when using Message Queue Telemetry Transport (MQTT) and Constrained Application Protocol (CoAP) as application protocols in charge of the reliable data delivery, providing considerations on their applicability to scenarios such as the one under consideration, and evaluating the provided performance level in different operating conditions.

The rest of this work is organized as follows: Section 2 discusses the most relevant works in the literature on FANETs, highlighting those that explicitly consider the use of satellites. Section 3 accurately describes the scenario and the application protocols under consideration for reliable data delivery, taking into account a M2M/IoT traffic pattern. Section 4 describes the UAV-to-satellite channel model in use in this work, accounting for mobility and fading phenomena. Section 5 provides insights on different performance metrics and on the effect of different protocol stacks on the achievable performance level. Section 6 draws the conclusions.

2 | RELATED WORK

A valuable survey on FANETs is provided in[2], with a clear focus on the communications issues inside a swarm. As already anticipated, a FANET shows peculiar differences with respect to MANETs and VANETs: in addition to those already cited, the frequent topology changes are worth to be mentioned, due to high mobility of UAVs. Furthermore, given the plethora of different sensing systems that a swarm can carry on board, different data delivery strategies should be considered, each with its own specific features. Because of those motivations, FANETs are to be considered a separate network family. In addition to the motivations provided by[1,2] to this matter, SWaP requirements are a further element that reinforces the differentiation among the aforementioned ad-hoc network configurations. SWaP requirements are crucial when dealing with UAVs, which are small-medium[2] objects that typically rely on batteries for their functioning. Because of the reduced size, also the payloads must satisfy stringent requirements, in order to minimize the impact on energy and flight dynamics. Despite those limitations, UAVs are more and more at the center of scenarios in which several sensing systems are available on-board, such as IoT scenarios[8], earthquakes, crowd surveillance[8], road traffic monitoring, goods delivery[5] and so on. The aggregated traffic profile generated by the on-board systems responds to M2M patterns, exhibiting low data-rates[3]. This traffic is composed by a *time-driven* fraction, such as telemetry data used to constantly monitor the correct functioning of all on-board systems and the position, and an *event-driven* fraction, due to the sensing systems that send out fresh data in the case of events of interest. The transmission of collected data can be done via IEEE 802.11-based systems, as for instance tested in[7] exploiting ad-hoc connectivity, or by taking into consideration 3G/4G, or even short-range connectivity[1]. UAVs are also used for data muling in impervious regions[10], which usually see the use of satellites as principal solution.

The aforementioned communication technologies are applicable in the case of Radio-Line-of-Sight (RLoS) or in delay-tolerant scenarios, while NRLoS ones depends on the use of satellite-based communications. Scenarios based on the use of satellites are analyzed in[1]. Those works focus on the applications provided by the use of FANETs, as for instance also done in[11] that considers micro aerial vehicles in a search and rescue scenario,

[1] Details can be found at: newsroom.intel.com/editorials/intel-drones-dazzle-los-angeles-sky-wonder-woman-light-show/
[2] The use of larger UAVs is mainly relegated to military uses at today.
[3] Of course, UAVs can be also used in multimedia scenarios[9], but they are not under consideration in this work.

For Peer Review Only

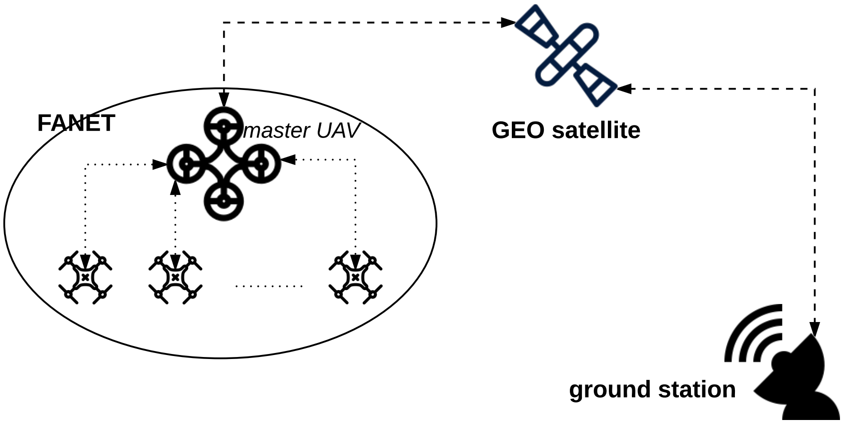


FIGURE 1 Scenario under consideration: a FANET generates data to be reliably delivered to a remote ground station via a geostationary satellite.

For Peer Review Only

relying on both terrestrial and satellite communications. The main focus in [11] is on the achievable Quality of Service (QoS) with different communication technologies in the case of small swarms. Reference [12] describes scenarios involving both terrestrial and satellite communications, where the latter is taken into account as a backup command and control link in the scenarios of UAV-aided wireless communications for the purposes of ubiquitous coverage, relaying, and information dissemination. This work sheds some light on possible application protocols to be used for reliable data exchange in mobility conditions, also taking into account fading phenomena. In the aforementioned scientific works, large attention is payed to the lower layers of UAV-based networks, rarely taking into account upper layer protocols and full protocol stacks. But, in order to be able to effectively guarantee interoperability in M2M/IoT scenarios involving the use of UAVs, such architectural choices cannot be neglected, thus motivating the present work. We presented some preliminary considerations in [5], where nanosatellites were considered, instead of a GEO satellite as in this work. Furthermore, with respect to [13,14], this work considers a more realistic scenario, taking into account an UAV-to-satellite channel model (described in Section 4) that explicitly considers the impact of mobility. When comparing the achievable performance level when using MQTT w.r.t. to the use of the CoAP protocol, a Transmission Control Protocol (TCP)-satellite flavor is used, in order to take into account the necessary countermeasures on long delay links, while [13,14] relies on the use of TCP NewReno.

3 | SCENARIO DESCRIPTION

This section describes the scenario under consideration in this work. As already anticipated, we aim at characterizing the achievable performance level when different M2M/IoT application protocols for reliable data exchanges are in use in a scenario involving a fast-moving FANET, also taking into account fading phenomena. This scenario is depicted in Figure 1 in which a FANET collects and/or generates data to be transferred to a remote ground station via a GEO satellite. Fresh data are sent to the master UAV according to a PUB/SUB paradigm, then transferred to the intended destination. We assume that those data, collected by the FANET from terrestrial Wireless Sensor Networks (WSNs) or generated from on-board sensors, are always correctly received from the master UAV: several possible solutions to achieve good results on this are presented in Section 2, so that we do not further investigate on it. Instead, we focus on the communication issues in the satellite part. A hierarchical FANET is considered, as described in [5], where different classes of UAVs compose the swarm. Only the master UAV, or cluster head, can access the satellite channel. Several UAVs act as routers within the FANET, relaying traffic to the master; furthermore, in the case of a failure of the master UAV, a router can replace it, providing fault-tolerance and redundancy. The return link of the DVB-RCS2 standard [15] is in use to deliver data from the FANET to the ground station(s), and the RA scheme in use is Contention Resolution Diversity Slotted Aloha (CRDSA) [16] with 3 replicas. We assume an ideal forward link, so that Acknowledgments (ACKs) are always correctly received. Because of the peculiar characteristic of M2M/IoT traffic that typically precludes the possibility to schedule sending operations, the use of dedicated access mechanisms is to be avoided in favor of RA ones, in order to support larger populations and to remove the need for time-consuming bandwidth on-demand mechanisms in the presence of small traffic bursts. On the other hand, the price of collisions must be paid, since data packets can be erased due to unresolved collisions after the Successive Interference Cancellation (SIC) process, or due to fading. In fact, we investigate on this by relying on the UAV-to-satellite channel model we present in this work, which is analytically described in Section 4.

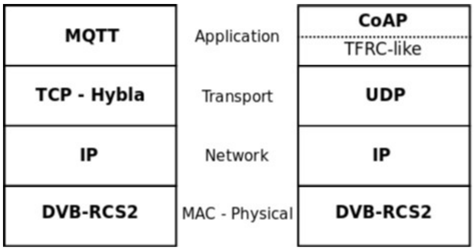


FIGURE 2 Protocol stacks under consideration. The CoAP-based protocol stack highlights the TCP Friendly Rate Control (TFRC)-like congestion control mechanism running at the application layer.

The considered traffic pattern is detailed in Section 3.1. Two application protocols are considered and compared for the reliable data delivery: CoAP, described in Section 3.2, and MQTT, described in Section 3.3. Both protocol stacks are visible in Figure 2 . In order to reduce the impact of the overhead in the presence of small packets, as in the case of M2M/IoT scenarios, RObust Header Compression (ROHC) is adopted, and is described in Section 3.4.

For Peer Review Only

3.1 | M2M traffic profile

The traffic sent from the FANET to the ground station(s) exhibits a M2M traffic profile, i.e., low data-rate, short packets and a bursty nature. In order to model such a behavior, each UAV in the swarm generates both time-driven (i.e., telemetry) and event-driven (i.e., event notifications) traffic. The time-driven fraction is here modelled as a Poisson distribution with an arrival rate λ_{td} [s^{-1}]. Each packet has size s_{td} [bytes], such that the resulting average sending rate is $\text{sr}_{\text{td}} = \lambda\,\text{s}_{\text{td}}$ [Bps] per UAV. On the other hand, the event-driven fraction is here modelled as a Beta distribution of parameters (α, β), according to [17]. Each packet has size s_{ed} [bytes] and the average source rate is of $\alpha/(\alpha+\beta)$ [arrivals/second], such that the resulting average sending rate is $\text{sr}_{\text{ed}} = \text{s}_{\text{ed}}\,\alpha/(\alpha+\beta)$ [Bps] per UAV. Both contributions generate the aggregated amount of traffic to be delivered per UAV in the swarm. The number of UAVs in a FANET is N, so that the whole amount of traffic per FANET is $\text{N}(\text{sr}_{\text{td}} + \text{sr}_{\text{ed}})$ [Bps].

3.2 | Constrained Application Protocol (CoAP)

CoAP is a User Datagram Protocol (UDP)-based protocol that can provide optional reliability to be implemented at the application layer. Because of this, CoAP can be used in scenarios in which a *transmit-only* mode is desired: in more words, when just a traffic flow from the FANET to the remote ground station is needed via the satellite return link. As expected, reliability cannot be guaranteed in this case because no feedback is available, but several scenarios may not require reliability while instead benefiting of the absence of traffic on the forward link.

In this work, we limit our analysis to the use of CoAP *Confirmable* messages, in order to provide a fair comparison with MQTT, later described. When only *Confirmable* messages are sent, the maximum number of in-flight packets is NSTART. According to RFC 7252, NSTART is defined as the maximum number of *simultaneous outstanding interactions*, which can be roughly translated into the use of a fixed-size transmission window [packets]. In the default configuration, NSTART is equal to 1, thus CoAP sends out a single packet, then waits for its ACK. Therefore, when using the default configuration, a simple Stop-and-Wait Automatic Repeat reQuest (ARQ) mechanism is employed, but the specifications open to the use of a larger NSTART value if proper congestion control mechanisms are employed: we do so in this work, and the congestion control strategy we implemented is described in Section 3.2.2. The choice of having NSTART $= 1$ as a default setting is motivated by the fact that the protocol is intended for low-power resource-constrained devices. Even if energy is a very precious resource on a battery-powered device, such as an UAV, we use NSTART > 1, in order to fully exploit the available bandwidth on the satellite return link. The latter choice is further motivated by the fact that the largest battery drain on an UAV is due to engines, under typical operating conditions.

CoAP employs a request-response pattern. In our scenario, such a configuration would mean that the ground station must periodically query all the swarm members (through the master UAV) to collect fresh data, if any. Aiming at reducing the delivery delay of the request-response pattern and at limiting the complexity of the architectural design, CoAP default behavior can be modified, as proposed in RFC 7641 and RFC 7252: those modifications, which we implemented in our simulator, are described in Section 3.2.1.

3.2.1 | Observer pattern and proxying functionalities

CoAP specifications open to the implementation of the so-called *observer* pattern, a data exchange model close to the PUB/SUB paradigm. Similarly to the latter, a client must perform a registration to the server(s) encapsulating the resource(s), indicating the Uniform Resource Identifiers (URIs) it is interested in. In this way, the server must keep a list of the subscribers and notify any new data. While this mechanism moves from a request-response to a PUB/SUB-like interaction, it requires that each CoAP server keeps track of the subscribers; in order to also remove this and to guarantee the decoupling between data producers and data subscribers, as in the PUB/SUB paradigm, the *proxying* functionality can be employed. A proxy is defined as a CoAP endpoint that can be delegated by clients to perform requests on their behalf. Thus, a CoAP proxy is an intermediate entity, which can actually decouple the clients from the servers. By implementing both the proxying functionality and the observer pattern, as we described in [13], a compliant PUB/SUB implementation of the CoAP protocol can be obtained. In our scenario, the UAVs are the CoAP servers encapsulating the resources (data generated or collected from sensors), the master UAV is the CoAP proxy acting as a decoupling entity, and the ground station is the only CoAP client subscribed to the resources (data) in the FANET.

In addition to the CoAP header, which is 6 bytes long, the use of the the observer pattern requires two additional bytes because of the so-called *observe option*. Thus, the application header is 8 bytes long (see Table 1).

For Peer Review Only

3.2.2 | CoAP congestion control

This section describes the congestion control in use in this work we proposed in [18] and here briefly recalled. It is inspired to the well-known TFRC mechanism (RFC 5348). TFRC is a congestion control mechanism designed for unicast flows, ensuring reasonable fairness with competing TCP flows. It exploits a feedback mechanism: the measured loss rate p at the receiver is sent back at the sender, in order to adjust the sending rate and match the perceived channel statistics. In this work, we exploit a sender-based variant of the TFRC protocol, which is used as CoAP congestion control mechanism. TFRC estimates p at the receiver side, then sends it back to the sender. In our implementation, p is estimated at the sender by exploiting CoAP ACKs, similarly to the way TCP works. The receiver is unmodified, and only acknowledges the received CoAP packets. Further than this, the sending rate is updated more frequently in this implementation than by TFRC, because it exploits each received ACK. The rationale behind this is in tracking the instantaneous value of the loss rate on the satellite channel as precisely as possible, and in avoiding any additional processing delay on an already long-delay link. In our scenario, we assume an in-order packet reception, thus the reception of an out-of-order ACK is interpreted as the symptom of a packet loss. If so, the lost packet is retransmitted at the expiration of the current timeout, whose value is doubled (up to 64 seconds) after each retransmission attempt.

3.3 | Message Queue Telemetry Transport (MQTT) protocol

MQTT is an IoT application protocol designed by IBM in 1999 for use in satellite networks. It represents a well-established solution in lots of scenarios exploiting a PUB/SUB data exchange model. A typical MQTT data packet is composed of a 2 bytes long fixed header part, a variable header part whose size depends on the packet type, and a variable length payload. Each data packet is sent to the broker (decoupling entity), labelled with a *topic*, which then forwards it to the MQTT nodes that subscribed the topic. MQTT is TCP-based, thus reliability is left out as a transport level issue. In addition to that, MQTT provides three QoS levels to address End to End (E2E) reliability[4]. We adopt QoS 1 as MQTT setting, which means that each packet must be delivered *at least once*: because of this, the application header is 10 bytes long (see Table 1).

3.4 | Robust Header Compression (ROHC)

ROHC is a set of profiles that can be used to compress packet headers, in order to reduce the overhead introduced by the protocols in the stack. In this work, MQTT relies on the use of the TCP-IP stack, and CoAP on the use of the UDP-IP stack. Given the small amount of traffic produced by M2M/IoT traffic sources, the use of ROHC provides great benefits, effectively reducing the overall overhead. In the case of TCP-IP, we rely on the mechanisms described in RFC 6846, which can, on average, reduce the overhead to 7 bytes per datagram. In the case of UDP-IP, we rely on the mechanisms described in RFC 5225, which can, on average, reduce the overhead to 5 bytes per datagram.

The whole overhead, from the application layer to the physical layer, can be read in Table 1 : the use of a CoAP-based protocol stack (in a PUB/SUB-like pattern) has an overall overhead of 21 bytes. On the other hand, the use of a MQTT-based protocol stack has an overall overhead of 25 bytes because of the aforementioned settings.

[4] It is worth recalling that the broker decouples publishers and subscribers, thus TCP can only provide reliability from/to the broker.

Layer	Length [bytes]	Layer	Length [bytes]
CoAP	8	MQTT	10
compressed UDP/IP	5	compressed TCP/IP	7
DVB-RCS2	8	DVB-RCS2	8

TABLE 1 Headers size for CoAP and MQTT-based protocol stacks.

4 | UAV-TO-SATELLITE CHANNEL MODEL

At the operating frequencies under investigation, i.e. Ku- Ka- bands, rain is the dominant fading mechanism[19]. Most models for rain attenuation time series synthesis and modeling of first order statistics refer to fixed ground stations[20–24]. However, a characteristic of FANETs is that the user terminals are mobile and therefore the aforementioned models are not directly applicable. In[25], the exceedance probability of rain attenuation for a mobile user ($\mathrm{P_{MOB}}$) is related to this for a fixed user ($\mathrm{P_{FIX}}$) through:

$$P_{MOB} = \xi \cdot P_{FIX} \tag{1}$$

where ξ is defined as:

$$\xi = \frac{u_R}{|u_M - u_R \cos\varphi|} \tag{2}$$

where $\mathrm{u_M}$ [km/h] is the amplitude of the velocity vector of the mobile terminal, $\mathrm{u_R}$ [km/h] is the amplitude of the velocity vector of the raincells (advection or front speed), and ϕ is the angle between these two vectors. Similarly to[21,22], in[26] the following stochastic differential equation has been proposed for the generation of rain attenuation time series $\mathrm{a_{MOB}}(\mathrm{t})$ for mobile terminals:

$$\frac{da_{MOB}(t)}{dt} = H(a_{MOB}, t) + W(a_{MOB}, t) \cdot n(t) \tag{3}$$

where $\mathrm{H}(\mathrm{a_{MOB}}, \mathrm{t})$ and $\mathrm{W}(\mathrm{a_{MOB}}, \mathrm{t})$ are the drift and diffusion coefficients, respectively, while $\mathrm{n}(\mathrm{t})$ is an additive white Gaussian noise (AWGN) stochastic process. The coefficients are given by:

$$H(a_{MOB}, t) = a_{MOB} \cdot d_{MOB} \cdot \left[\sigma^2_{\mathrm{MOB}} - (\ln(a_{MOB}) - \ln(m_{MOB}))\right] \tag{4}$$

$$W^2(a_{MOB}, t) = 2d_{MOB} \cdot a^2_{MOB}(t) \cdot \sigma_{MOB} \tag{5}$$

where $\mathrm{d_{MOB}}$ is the dynamic parameter that describes the transition rate of $\mathrm{a_{MOB}}(\mathrm{t})$, while $\mathrm{m_{MOB}}$ and σ_{MOB} are the mean value and the standard deviation of the lognormal distribution of rain attenuation, respectively. The parameters $\mathrm{m_{MOB}}$ and σ_{MOB} are calculated through the fitting of a lognormal distribution to the exceedance probability of rain attenuation. Moreover, as also stated in[25] and found in[26], the rate of change of attenuation for a moving terminal would equal that of the fixed terminal scaled down by ξ. Therefore, for the dynamic parameters of rain attenuation for mobile and fixed terminal, it holds that:

$$d_{MOB} = d_{FIX}/\xi \tag{6}$$

where $\mathrm{d_{FIX}}$ can be assumed equal to $2 \cdot 10^{-4}\mathrm{sec}^{-1}$, as also recommended in[27] and found for excess attenuation in[28].

In order to solve (3), the rain attenuation is transformed into a Gaussian process $\mathrm{x_a}(\mathrm{t})$, in a similar way to the approach in[21] and[26]:

$$x_a(t) = \frac{\ln(a_{MOB}(t)) - \ln(m_{MOB})}{\sigma_{MOB}} \tag{7}$$

The equation that describes the diffusion process $\mathrm{x_a}(\mathrm{t})$ is the Langevin equation:

$$\frac{dx_a(t)}{dt} = -d_{MOB} \cdot x_a(t) + \sqrt{2d_{MOB}} \cdot n(t)\,. \tag{8}$$

The solution of (8) is:

$$x_a(t) = e^{-td_{MOB}} \cdot x_a(0) + e^{-td_{MOB}} \cdot \sqrt{2d_{MOB}} \int_0^t e^{s \cdot d_{MOB}} dW_s \tag{9}$$

where $\mathrm{dW_s}$ is the Wiener process.

There is another significant difference between the case of mobile terminals on the ground and the case of a FANET, which is the height of the terminals. The master UAV is flying above ground, therefore the path UAVto-satellite through rain depends on its height. A methodology is presented in[29] for the calculation of rain attenuation for aeronautical scenarios. According to it, in the case of a link between an airborne platform and the satellite, the recommendation in[20] should be used, but in the case of an Earth station at a height equal to the height of the airborne platform. Of course, when the platform is higher than the rain height, then the rain attenuation is assumed equal to 0 dB. However, it does not take into account the fact that the airborne platform is a mobile terminal. To address that, in this proposed model, the exceedance probability of rain attenuation

For Peer Review Only

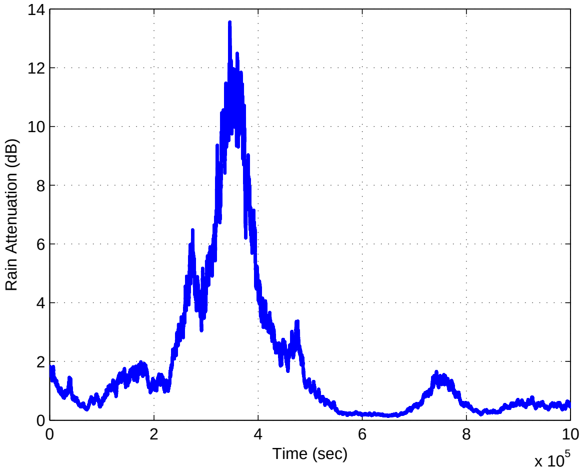


FIGURE 3 Rain attenuation time-series for a Ku-band downlink with an UAV.

between an airborne platform and the satellite is calculated using (1) with P_{FIX} derived from[20] assuming the station height equal to the height of the master UAV. The step-by-step algorithm for the generation of rain attenuation time series is:

1. calculate the parameter ξ of (2) for a given storm velocity and mobile terminal velocity;
2. calculate P_{FIX} based on the recommendation in[20] using the height of the master UAV. If the height of the master UAV is higher than the rain height, then the rain attenuation is 0 dB and the following steps are omitted;
3. using (1), calculate the exceedance probability for the mobile terminal P_{MOB};
4. using the methodology in[30], calculate the lognormal parameters (m_{MOB} and σ_{MOB}) of the rain attenuation induced on mobile links;
5. calculate the dynamic parameter of rain attenuation for mobile terminals using (6);
6. calculate the time series of $x_a(t)$ using (9) with $x_a(0)$ equal to 0.5 dB;
7. calculate rain attenuation using the inverse of (7).

As an example, a snapshot of rain attenuation for an UAV at a height of 100 [m] with a speed of 50 [km/h] located above Pisa, Italy, is shown for a Ku-band downlink in Figure 3 .

5 | PERFORMANCE EVALUATION

In this section, the scenario under configuration is evaluated by means of simulations. The simulator we built is based on the use of S-NS3[31], which provides a compliant DVB-RCS2 implementation to the NS3 environment. The channel model described in Section 4 and the traffic model in Section 3.1 have been implemented in Matlab, and their outputs imported as trace-files in the simulator. More details can be found in Appendix A. For the simulation purposes, a clear sky Signal to Noise Ratio (SNR) reference value of 13 dB has been considered, taking into account the antenna gains, the pointing losses, and other deterministic path losses. This SNR value is then deteriorated by taking into account the rain attenuation effects for the UAV-to-satellite link according to the model in Section 4. The construction of effective antennas with suitable characteristics for these links remains a significant technical challenge in the case of FANETs[32–34].

We recall that the satellite return link is used to send data from the FANET to the ground station. We used waveform ID 4, as detailed in Table 2 : this is a recommended choice for M2M traffic profiles in DVB-RCS2 RA specifications. The RA block has a duration of 43 [ms]. In our configuration, an RA block spans the whole superframe and is composed of 64 time-slots, or *transmission opportunities*. Each master UAV can use a single transmission opportunity per RA block, so that the whole setup provides an overall bandwidth B of approximately 10.72 [Kbit/s] per FANET. Regarding the traffic model, the Beta distribution used for the event-driven fraction of the traffic has parameters $(\alpha, \beta) = (3, 4)$ with a period $T = 10$ [s][17]. The Poisson distribution used for the time-driven fraction of the traffic has rate $\lambda = 0.1$ [s^{-1}]. In both cases, the packet size is chosen such as to exactly fit the

For Peer Review Only

Waveform ID	Burst length [symbols]	Payload length (bytes, symbols)	Modulation scheme	Code rate
4	536	59, 472	QPSK	1/2

TABLE 2 Details on the DVB-RCS2 waveform in use.

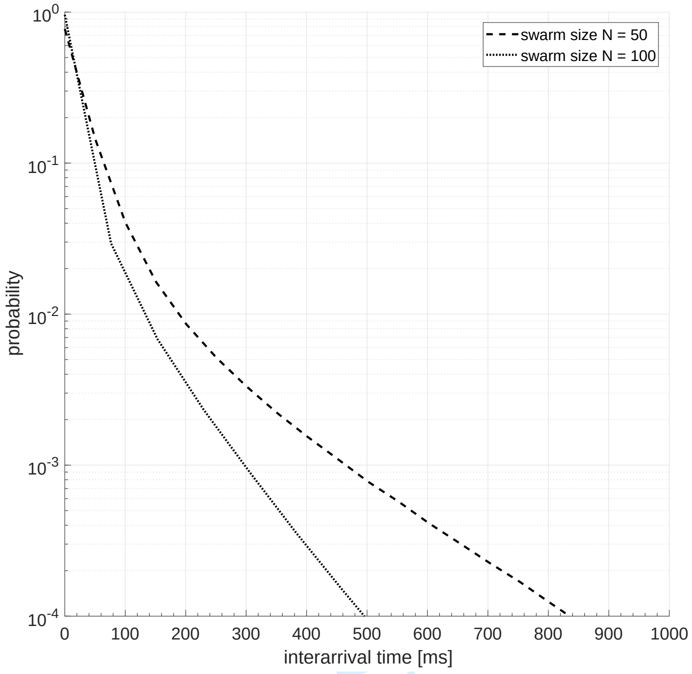


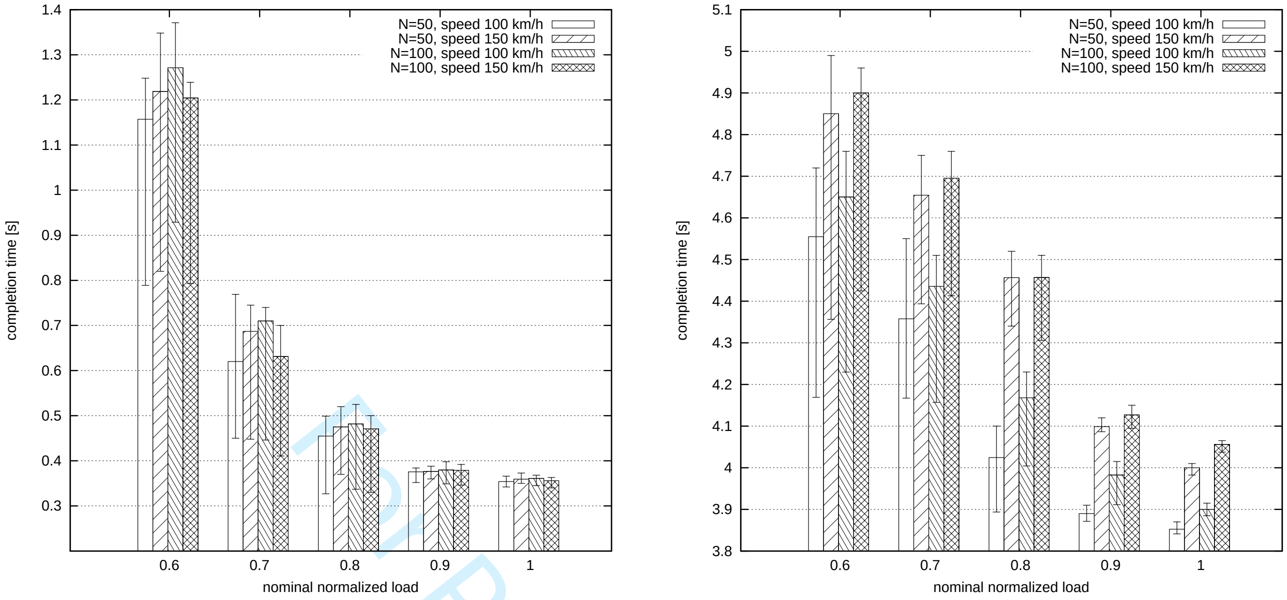
FIGURE 4 Distributions of the inter-arrival time between two consecutive data packets for two different swarm sizes: $\mathrm{N} = 50$ and $\mathrm{N} = 100$.

view Only

time-slot in the RA block, so that a CoAP payload is 38 bytes long, and an MQTT payload is 34 bytes long. Two different swarm sizes have been tested, aiming at evaluating the performance level provided by the two application protocols under consideration in the proposed setup, with the intent of assessing the achievable average completion time in the cases of two different swarm sizes, i.e. $\mathrm{N} = 50$ and $\mathrm{N} = 100$. In fact, the completion time provides a more significant assessment than the throughput metric in the presence of short M2M/IoT data bursts. The resulting distributions of the inter-arrival time between two consecutive data packets in those two cases can be seen in Figure 4 , with mean values of 40 [ms] and of 20 [ms] for $\mathrm{N} = 50$ and $\mathrm{N} = 100$, respectively.

The first metric under consideration is the completion time, as anticipated. The simulation results in Figure 5 show that CoAP achieves a lower average completion time (in Figure 5 a) when coupled with the congestion control mechanism described in Section 3.2.2, than MQTT (in Figure 5 b). The completion time, in both cases, decreases as the MAC normalized load increases: this is due to the fact that the increased contention in the RA channel forces the congestion control mechanism to lower the sending rate, in order to counteract the increasing collision rate. The lower the sending rate in a high load case, the lower the average queue length of packets pending for transmission as well, since less packets are enqueued per unit of time w.r.t. a lower load case. In addition to this, a TFRC-based congestion control mechanism is less aggressive on the bottleneck than the one of a TCP-based one. The latter relies on the use of TCP Hybla[35] at transport layer, a TCP variant designed for long delay links.

The aforementioned behavior can also be found in Figure 6 , which shows the normalized MAC load and throughput level in both cases. The throughput is the aggregated normalized value, calculated as the number of correctly decoded MAC frames at the destination divided by the number of available time-slots. The difference between the load and the throughput levels is due to unrecoverable collisions. The less aggressive behavior of the CoAP-based scenario can be recognized because of the smaller oscillations around the average value, visible in Figure 6 a, when compared to the oscillations visible in Figure 6 b, highlighting the different MAC load per congestion control mechanism. Further than this, Figure 6 also shows the benefits of the *capture effect* at destination. In fact, because of both the attenuation and the mobility impact (described in Section 4) on the uplink seen by each master UAV in a FANET, the SIC process behaves in a more efficient way. The overall performance level greatly benefits of the latter.



(a) CoAP-based protocol stack

(b) MQTT-based protocol stack

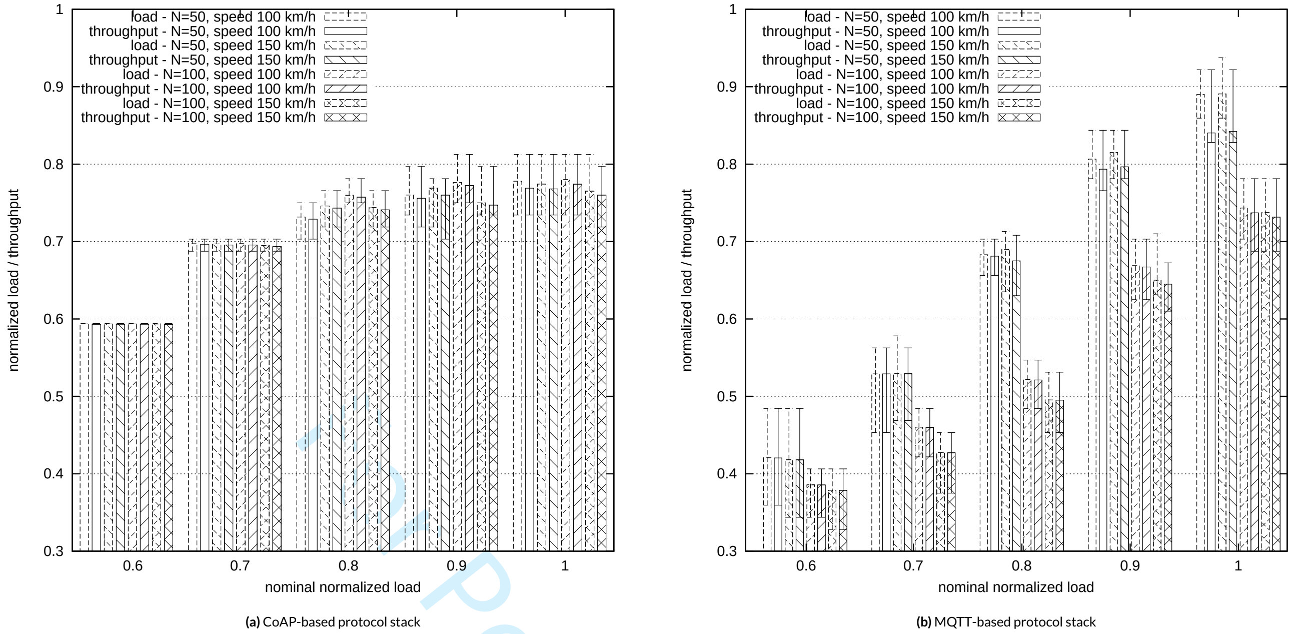
For Peer Review Only

FIGURE 5 The completion time [s] for both protocol stacks under consideration. The nominal normalized MAC load level can be read on the x-axis, which represents the number of swarms in our simulations. The height of each column represents the average value of the completion time per scenario, while the 0.25 and 0.75 quantiles are visible as errorbars.

In order to fully characterize the system, Table 3 shows the absolute value of the application layer goodput g_{app} per FANET per scenario. The goodput is the information rate [Kbit/s], excluding the overhead due to the two different protocol stacks in use in this work. In the first column, the FANET size can be read, while the moving speed can be read in the second column. We tested the system for an increasing number of FANETs operating at the same time (III column), in order to provide an extensive assessment of the achievable performance level for both the protocol stacks under consideration. Because of the lower overhead (see Table 1), CoAP outperforms MQTT at every load under consideration: such a behaviour is visible by comparing columns IV and V. The value outside parentheses in both columns is the absolute goodput value [Kbit/s], confirming that the amount of useful data is larger when using CoAP than when using MQTT. The value among parentheses in both columns show the normalized value of the goodput w.r.t. the available bandwidth B, calculated as $\hat{g}_{app} = g_{app}/B$. In order to further ease the reader, column VI in Table 3 shows the normalized goodput gain provided by the use of CoAP, calculated as $\hat{g}_{app}^{CoAP} - \hat{g}_{app}^{MQTT}$. The different congestion control mechanism should be also taken into account here, because the congestion control in the CoAP-based stack (described in Section 3.2.2) shows less abrupt variations in the sending rate when compared to the MQTT-based stack (we recall that TCP Hybla is in use at the transport layer). Eventually, Table 3 proves that CoAP, coupled with the congestion control mechanism fully described in [18], is able to better utilize the available network resources, and such a gain can be larger than 20% at medium loads. Because of this, a lower completion time is provided by the use of CoAP (approximatively four times less), as we shown in Figure 5 .

6 | CONCLUSIONS

In this work, we discussed reliable data delivery from FANETs to remote ground stations via satellite. FANETs will likely represent the next step in the use of UAVs in several application scenarios, as we reported in the literature survey. We explicitly consider NRLoS scenarios via satellite back-haul, taking into account fast-moving swarms. In order to provide credible simulation results, we also present an UAV-to-satellite channel model. The most established protocol stacks for M2M/IoT traffic have been considered and compared in order to assess which between CoAP and MQTT outperforms the other in terms of completion time and goodput. The attention to complete protocol stacks is motivated by the need for interoperability in the M2M/IoT field, and we stress how such a choice can be helpful in better integrating satellite and terrestrial networks. The considered traffic patterns are built upon 3GPP models, accounting for both time-driven and event-driven traffic. For a fair comparison with MQTT in terms of reliability, we designed and implemented a TFRC-based congestion control mechanism for CoAP, as far as it only implements a simple Stop&Wait ARQ mechanism at today, so that no rate control would be needed. In addition, the use of the observer pattern and the proxying



(a) CoAP-based protocol stack

(b) MQTT-based protocol stack

For Peer Review Only

FIGURE 6 Average aggregated load and throughput load levels for both protocols stacks. The nominal normalized MAC load level can be read on the x-axis, which represents the number of swarms in the simulations. The height of each column represents the average value of the actual load / throughput per scenario, while the 0.25 and 0.75 quantiles are visible as errorbars.

FANET size N	Speed [km/h]	Nominal normalized load level	Average goodput per FANET g_{app} [Kbit/s] ($\hat{g}_{\text{app}}$)		CoAP gain
			CoAP	MQTT	
50	100	0.6	6.90 (0.64)	4.37 (0.41)	0.23
		0.7	6.84 (0.64)	4.65 (0.43)	0.21
		0.8	6.32 (0.59)	5.28 (0.49)	0.10
		0.9	5.76 (0.54)	5.41 (0.50)	0.04
		1	5.31 (0.50)	5.19 (0.48)	0.02
	150	0.6	6.90 (0.64)	4.35 (0.41)	0.23
		0.7	6.83 (0.64)	4.65 (0.43)	0.21
		0.8	6.44 (0.60)	5.23 (0.49)	0.11
		0.9	5.79 (0.54)	5.43 (0.51)	0.03
		1	5.30 (0.49)	5.20 (0.49)	0.00
100	100	0.6	6.90 (0.64)	4.01 (0.37)	0.27
		0.7	6.83 (0.64)	4.04 (0.38)	0.26
		0.8	6.56 (0.61)	4.04 (0.38)	0.23
		0.9	5.88 (0.55)	4.55 (0.42)	0.13
		1	5.34 (0.50)	4.55 (0.42)	0.08
	150	0.6	6.90 (0.64)	3.94 (0.37)	0.27
		0.7	6.81 (0.64)	4.12 (0.38)	0.26
		0.8	6.42 (0.60)	4.25 (0.40)	0.20
		0.9	5.69 (0.53)	4.40 (0.41)	0.12
		1	5.25 (0.49)	4.52 (0.42)	0.07

TABLE 3 Average goodput in both absolute and normalized values for the protocol stacks under consideration. The last column reports the gain provided by the use of CoAP w.r.t. the use of MQTT in our scenario.

functionality for CoAP is described and implemented in our simulator, so that data is delivered in a PUB/SUB-like fashion, similarly to MQTT. Then, by exploiting the proposed channel model, we evaluated the completion time and the normalized goodput achieved by using the two protocols stacks, pointing out how CoAP outperforms MQTT in the operating conditions under consideration.

ACKNOWLEDGMENTS

This work contains the outcomes of a study carried out within the *Multi-homed network architectures for flying ad-hoc networks (FANETs) and nano-satellite swarms* working group, part of the Satellite Network of Excellence (SatNEx) IV, phase I, funded by the European Space Agency (ESA). The view expressed herein can in no way be taken to reflect the official opinion of ESA.

□

For Peer Review Only

APPENDIX

A - IMPORTING TRACE-FILES INTO THE S-NS3 SIMULATOR

External traffic traces have been generated offline through an external tool and consist in a sequence of inter-arrivals distributed according to the details reported in Section 3.1. The entities accessing the satellite channel are the master UAVs, thus a traffic traces is needed for each one of them. Each traffic trace is assigned to the sender application (logically encapsulated in the master UAV) that corresponds to either the CoAP proxy or to the MQTT publisher. Both these applications have been implemented in the simulator from scratch as NS3 classes. The corresponding instances receives as input parameter the array *IAT*, composed by inter-arrival times read from the trace-file in the form of a CSV. When the simulation is in progress, the *i*-th packet sending event is scheduled by the *generatePkt* method according to the value stored in the *i*-th entry of the *IAT* array.

A similar approach has been used with the channel model time series. They have been generated offline and imported into the simulator. For each satellite user terminal, two time series have been generated, corresponding to the rain attenuation affecting *(i)* the return link and *(ii)* the forward link. Each row of the time series is composed of two fields: *timestamp* [sec] and *fading* [dB]. In order to import external fading files, the attribute *SatChannel::EnableExternalFadingInputTrace* must be set to *true* in the simulator settings, whereas the attributes *SatFadingExternalInputTraceContainer::UtFwdDownIndexFileName* and *SatFadingExternalInputTraceContainer::UtRtnUpIndexFileName* expect a string parameter indicating the path of the binary files used to store the fading time-series. In the latter, values are encoded as single-precision floating-point numbers (binary format). The external fading files must be stored within the folder `/contrib/satellite/data/ext-fadingtraces/input/` found under the NS3 installation folder. The time series are imported during the initialization phase of the simulation and then used to compute the received signal power at the gateway.

References

1. Hayat, Samira and Yanmaz, Evşen and Muzaffar, Raheeb. Survey on unmanned aerial vehicle networks for civil applications: a communications viewpoint. IEEE Communications Surveys & Tutorials, 2016;18(4):2624–2661

2. Bekmezci, Ilker and Sen, Ismail and Erkalkan, Ercan. Flying Ad Hoc Networks (FANET) test bed implementation. IEEE Recent Advances in Space Technologies (RAST), Istanbul, Turkey, 2015;665-668.

3. ITU-R M.2171. Characteristics of unmanned aircraft systems and spectrum requirements to support their safe operation in non-segregated airspace. December 2009.

4. Bacco, Manlio and Ferro, Erina and Gotta, Alberto. UAVs in WSNs for agricultural applications: An analysis of the two-ray radio propagation model. IEEE SENSORS conference, 2014, Valencia Spain;130-133.

5. Bacco, Manlio and Cassará, Pietro and Colucci, Marco and Gotta, Alberto and Marchese, Mario and Patrone, Fabio. A Survey on Network Architectures and Applications for Nanosat and UAV Swarms. In: Pillai P., Sithamparanathan K., Giambene G., Vázquez M., Mitchell P. (eds) Wireless and Satellite Systems. WiSATS 2017. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 231. Springer, Cham

6. Bekmezci, Ilker and Sahingoz, Ozgur Koray and Temel, Şamil. Flying ad-hoc networks (FANETs): A survey. Elsevier Ad Hoc Networks. 2013;11(3):1254–1270.

7. Yanmaz, Evşen and Yahyanejad, Saeed and Rinner, Bernhard and Hellwagner, Hermann and Bettstetter, Christian. Drone networks: Communications, coordination, and sensing. Elsevier Ad Hoc Networks, 2018. 68:1-15.

8. Motlagh, Naser Hossein and Bagaa, Miloud and Taleb, Tarik. UAV-based IoT platform: A crowd surveillance use case. IEEE Communications Magazine, 2017. 55(2):128–134.

9. Bacco, Manlio and Chessa, Stefano and Di Benedetto, Marco and Fabbri, Davide and Girolami, Michele and Gotta, Alberto and Moroni, Davide and Pascali, Maria Antonietta and Pellegrini, Vincenzo. UAVs and UAV Swarms for Civilian Applications: Communications and Image Processing in the SCIADRO Project. In: Pillai P., Sithamparanathan K., Giambene G., Vázquez M., Mitchell P. (eds) Wireless and Satellite Systems. WiSATS 2017. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 231. Springer, Cham.

10. Palma, David and Zolich, Artur and Jiang, Yuming and Johansen, Tor Arne. Unmanned Aerial Vehicles as Data Mules: An Experimental Assessment. IEEE Access 2017.

11. Andre, Torsten and Hummel, Karin Anna and Schoellig, Angela P and Yanmaz, Evsen and Asadpour, Mahdi and Bettstetter, Christian and Grippa, Pasquale and Hellwagner, Hermann and Sand, Stephan and Zhang, Siwei. Application-driven design of aerial communication networks. IEEE Communications Magazine, 2014. 52(5);129-137.

12. Zeng, Yong and Zhang, Rui and Lim, Teng Joon. Wireless communications with unmanned aerial vehicles: opportunities and challenges. IEEE Communications Magazine, 2016. 54(4);36–42.

13. Bacco, Manlio and Colucci, Marco and Gotta, Alberto. Application protocols enabling Internet of Remote Things via random access satellite channels. IEEE International Conference on Communications (ICC), 2017, Paris, France:1–6.

14. Bacco, Manlio and De Cola, Tomaso and Giambene, Giovanni and Gotta, Alberto. Advances on elastic traffic via M2M satellite user terminals. IEEE Wireless Communication Systems (ISWCS), 2015, Brussels, Belgium:226–230.

15. ETSI EN 301 545-2. Second Generation, DVB-RCS2 Part2: Lower Layers for Satellite Standard, 2012.

16. E. Casini and R. De Gaudenzi and O.D.R. Herrero. Contention Resolution Diversity Slotted ALOHA (CRDSA): An enhanced random access scheme for satellite access packet networks. IEEE Transactions on Wireless Communications, 2007. 6(4):1408–1419.

17. Laner, Markus and Svoboda, Philipp and Nikaein, Navid and Rupp, Markus. Traffic models for machine type communications. Wireless Communication Systems (ISWCS), 2013, Ilmenau, Germany:1–5.

18. Bacco, Manlio and Cassarà, Pietro and Colucci, Marco and Gotta, Alberto. Modeling Reliable M2M/IoT Traffic over Random Access Satellite Links in Non-saturated Conditions. IEEE Journal on Selected Areas in Communications, 2018:1–10.

19. A. D. Panagopoulos and P. D. M. Arapoglou and P. G. Cottis. Satellite communications at KU, KA, and V bands: Propagation impairments and mitigation techniques. IEEE Communications Surveys & Tutorials, 2004. 6(3):2–14.

20. ITU-R. P. 618-12. Propagation data and prediction methods required for the design of Earth-space telecommunication systems, 2015.

21. T. Maseng and P. Bakken. A Stochastic Dynamic Model of Rain Attenuation. IEEE Transactions on Communications, 1981. 29(5):660–669.

22. S. A. Kanellopoulos and A. D. Panagopoulos and J. D. Kanellopoulos. Calculation of the Dynamic Input Parameter for a Stochastic Model Simulating Rain Attenuation: A Novel Mathematical Approach. IEEE Transactions on Antennas and Propagation, 2007. 55(11):3257-3264.

23. X. Boulanger and L. Feral and L. Castanet and N. Jeannin and G. Carrie and F. Lacoste. A Rain Attenuation Time-Series Synthesizer Based on a Dirac and Lognormal Distribution. IEEE Transactions on Antennas and Propagation, 2013. 61(3):1396–1406.

24. C. Kourogiorgas and A. Kelmendi and A. D. Panagopoulos and A. Vilhar. On Rain Attenuation Time Series Generation: A New Simple Copula-based Channel Model for Satellite Slant Paths. IEEE Transactions on Antennas and Propagation, 2016. 64(7):3206–3211.

25. E. Matricciani. Transformation of rain attenuation statistics from fixed to mobile satellite communication systems. IEEE Transactions on Vehicular Technology, 1995. 44(3):565–569.

For Peer Review Only

26. Arapoglou, Pantelis-Daniel and Liolis, Konstantinos P. and Panagopoulos, Athanasios D.. Railway satellite channel at Ku band and above: Composite dynamic modeling for the design of fade mitigation techniques. International Journal of Satellite Communications and Networking, 2012. 30(1):1–17.

27. ITU-R. P.1853-1. Tropospheric Attenuation Time Series Synthesis, 2012.

28. Papafragkakis, A. and Kourogiorgas, C. and Panagopoulos, A.D. and Ventouras, S.. Large- and Short-Scale Diversity in Greece and United Kingdom for High Throughput Satellite Systems. 23rd Ka and Broadband Communications Conference, 2017, Trieste, Italy.

29. ITU-R. P.2041-1. Prediction of path attenuation on links between an airborne platform and Space and between an airborne platform and the surface of the Earth, 2013.

30. ITU-R. P.1057-4. Probability distributions relevant to radiowave propagation modelling, 2015.

31. J. Puttonen and S. Rantanen and F. Laakso and J. Kurjenniemi and K. Aho and G. Acar. Satellite model for Network Simulator 3. Proceedings of the 7th International ICST Conference on Simulation Tools and Techniques, 2014, Lisboa, Portugal:86–91.

32. J. Zhao and F. Gao and Q. Wu and S. Jin and Y. Wu and W. Jia. Beam Tracking for UAV Mounted SatCom on-the-Move With Massive Antenna Array. IEEE Journal on Selected Areas in Communications, 2018. 36(2):363–375.

33. R. Baggen and S. Holzwarth and M. Bottcher and S. Otto. Innovative Antenna Front Ends from L-Band to Ka-Band [Antenna Applications Corner]. IEEE Antennas and Propagation Magazine, 2017. 59(5):116–129.

34. T. Lambard and O. Lafond and M. Himdi and H. Jeuland and S. Bolioli and L. Le Coq. Ka-Band Phased Array Antenna for High-Data-Rate SATCOM. IEEE Antennas and Wireless Propagation Letters, 2012:256–259.

35. Caini, Carlo and Firrincieli, Rosario. TCP Hybla: a TCP enhancement for heterogeneous networks. International journal of satellite communications and networking, 2004. 22(5):547–566.

For Peer Review Only