MUTUAL POSITION PLAUSIBILITY CHECKING IN FLYING AD-HOC NETWORKS USING DISTANCE MEASUREMENTS

Tobias Marks[1] & Konrad Fuger[2]

[1]German Aerospace Center, Institute of Air Transport, Hamburg, Germany
[2]Hamburg University of Technology, Institute for Communication Networks, Hamburg, Germany

Abstract

In this paper we present an approach to check the plausibility of UAV positions based on independent distance estimations between the nodes in a flying ad-hoc network (FANET) built up by the UAVs. To achieve this, we define a set of data messages that are flooded throughout the network by each node and that are used to derive a situational overview if received at a particular UAV. Based on the received data, distances between the nodes within the FANET can be estimated both from GNSS position readings as well as from time of flight information additionally provided in the data messages. The difference of both values can be used as an indicator to thereby classify edges and nodes of the FANET as plausible or implausible. In our work, we derive appropriate thresholds for the deviation of both measurements as well as a trust score for each node based on the plausibility values of adjacent edges. To showcase our approach, we apply it to an air traffic scenario based on drone package delivery in urban environments and simulate the data traffic in the resulting FANET resulting from the data messages we defined. We show, that our approach yields reasonable results for the evaluated scenario.

Keywords: flying ad-hoc networks, communication, UAV, air-to-air link, data link

1. Introduction

1.1 Motivation

In recent years more and more UAV applications (e.g. drone package delivery (1) (2), inspection or monitoring) are being realized resulting in considerable traffic volumes in the very low-level airspace especially in urban areas. The safe operation of vehicles in high traffic scenarios, however, requires reliable communication, navigation and tracking of the vehicles (3). The navigation systems of the UAVs in use, generally depend on GNSS as main navigation source. GNSS, however, is vulnerable to various attacks like spoofing or jamming leading to undesirable situations, which can cause damage to the vehicle or ground infrastructure or be harmful to people. Position verification is, therefore, an important issue that needs to be addressed. Here communication plays a major role, as despite ground-based surveillance methods exist, UAVs can actively provide their position information to other vehicles or ground entities (e.g. via remote identification systems). In addition, flying ad-hoc networks (FANETs) provide another solution to share position information and enable redundant communication (4) between the UAVs and from the UAVs to ground stations. This in turn allows for information sharing throughout the network enabling UAVs to locally construct a situational overview and to verify the plausibility of the position data provided by other UAVs.

1.2 State of the art

In general, flying ad-hoc networks are subject to various security threats either from the outside (e.g. GNSS spoofing or jamming, man in the middle attack) or from the inside of the network (e.g. wormhole attack or sybil attack), that intent to compromise a vehicle, falsify or extract data and more.

Hence, the protection against cyber-attacks is an important topic in current FANET research. Beside cryptography, certificate management or secure communication protocols, trust management is one approach to cope with these threats. (5) gives a good overview of the topic mainly addressing UAV teams to perform specific collaborative tasks such as surveillance, monitoring or data collection. Especially in the military domain security aspects are crucial. (6) presents a methodology combining authentication mechanisms with movement plausibility checks. Also, in the context of vehicular ad-hoc networks (VANETS) and mobile ad-hoc networks (MANETS), the approach to use plausibility checks is addressed (7). Such checks include among others the acceptance range threshold (ART), mobility grade threshold (MGT), the sudden appearance warning (SAW), the simple speed check (SSC) and the distance moved verifier (DMV). Of these approaches, ART and MGT have been applied to UAVs in (8), but with this approach, position verification cannot be performed for the entire network, but is limited to a UAV's direct neighbours. (9) explores the position verification between a mobile station and several ground stations. The issue of false or inaccurate positions of UAVs obtained through GNSS is also addressed in (10) and (11). Here, the authors use additional data sources, such as the inertial navigation system or received signal strength from known transmitters to verify their own position. Nevertheless, this cannot be applied to verify the position of other UAVs.

1.3 Our contribution

In the project VEREDUS funded by the German Ministry of Economic Affairs and Climate Action a flying ad-hoc network is developed that ensures communication to the vehicles in case of a failure in the primary link and additionally enables a plausibility check of the GNSS positions of all UAVs participating in the network. This check is envisioned to be obtained by distance measurements in-between the vehicles using the data link. All thereby collected independent distance measurements and GNSS positions are broadcasted through the whole FANET, such that each UAV receives all information and can construct a situational overview. This provides for checking the positions of other UAVs in terms of plausibility using both measurements by calculating a position trust value for each node (UAV) of the network graph. In this work, we present the basic assumptions taken to set up such a system followed by the description of a methodology that can be used to perform the position verification by calculating the corresponding trust values. Furthermore, we will show the suitability of the algorithm in simulated scenarios including UAV mobility and data traffic.

2. Methods

2.1 Network model

The ad-hoc network that we assess in our work is represented by a two-dimensional graph $G = (N, E)$ that is given by the nodes N (airborne stations; AS) and edges E (data links between two AS).

2.1.1 Node distance

The distance between two nodes of the network graph N_i and N_j at time t is denoted as d_{ij}^t. For each pair of two nodes N_i and N_j in the network, a distance measurement can be obtained if they are within radio range and a radio connection is established. As described in section 2.5, each node broadcasts its position to its neighbours, which then adds the time of flight information to the data message that is used to calculate the distance. The measurements for the distance between two nodes (CPR_{BA} and CPR_{AB}) are treated independent of their direction. It is $d_{ij}^t = d_{ji}^t$. A measurement $\hat{d}_{ij}$ of the distance is given by the true distance d_{ij} and a normally distributed error term $\varepsilon_d = \mathcal{N}(0, \sigma_d^2)$. With σ_d being the standard deviation for the distance measurement.

$$\hat{d}_{ij} = d_{ij} + \varepsilon_d \tag{1}$$

For each edge E_{ij} between two nodes in the network graph a distance history $\hat{\mathbf{D}}_{ij}$ is stored locally at the observing node and updated as new information is obtained. Old data can be deleted if they surpass a certain time limit. Along with the distances the history of timestamps of when the measurements were obtained $\mathbf{T}_{E,ij}$ is stored. Here t_n depicts the latest measurement. An edge having at least one valid measurement in the time interval I_R (see section 2.3) is considered an active edge. Hence, also a list of active edges $\mathbf{K}_E$ is stored at the observing node.

$$\hat{\mathbf{D}}_{ij} = (\hat{d}_{ij}^{t_n}, \hat{d}_{ij}^{t_{n-1}}, \hat{d}_{ij}^{t_{n-2}}, ...) \qquad \mathbf{T}_{E,ij} = (t_n, t_{n-1}, t_{n-2}, ...) \tag{2}$$

The distance $\tilde{d}_{ij}^{t0}$ between two nodes i and j is linearly extrapolated at the observing node for a timestamp t_0 using the applicable measurements A and B from $\hat{\mathbf{D}}_{ij}$ that are chosen by a specific selection rule (see section 2.3).

$$\tilde{d}_{ij}^{t0} = \hat{d}_{ij}^{t_A} + (\hat{d}_{ij}^{t_A} - \hat{d}_{ij}^{t_B}) \cdot \kappa_{d_{ij}}^{t0} = \underbrace{d_{ij}^{t_A} + (d_{ij}^{t_A} - d_{ij}^{t_B}) \cdot \kappa_{d_{ij}}^{t0}}_{d_{ij}^{t0}} + \varepsilon_{\tilde{d}_{ij}^{t0}} \tag{3}$$

Here $\kappa_{d_{ij}}^{t_0}$ depicts the factor derived from two selected timestamps t_A and t_B obtained from the values in $\mathbf{T}_{E,ij}$ in relation to t_0.

$$\kappa_{d_{ij}}^{t_0} = \frac{(t_0 - t_A)}{(t_A - t_B)} \tag{4}$$

2.1.2 Node position

The position of a node N_i at time t is denoted as $\mathbf{p}_i^t$. The measured positions of a node are given by the true positions of the node and a normally distributed error vector of the measurement $\varepsilon_p = \mathcal{N}(0, \sigma_p^2)$. With σ_p being the standard deviation for the position measurement.

$$\hat{\mathbf{p}}_i = \mathbf{p}_i + \boldsymbol{\varepsilon}_p \tag{5}$$

Same as for distance measurements, each node N in the network propagates its own position $\hat{\mathbf{p}}^t$ to all other network nodes. Each observing node can, therefore, store a history of positions $\hat{\mathbf{P}}_i$ for each other known node N_i locally. Along with the positions the history of timestamps of when the information was obtained $\mathbf{T}_{N,i}$ is stored. Additionally, all nodes that are known to the observing node are stored in a list $\mathbf{K}_N$.

$$\hat{\mathbf{P}}_i = (\hat{\mathbf{p}}_i^{t_n}, \hat{\mathbf{p}}_i^{t_{n-1}}, \hat{\mathbf{p}}_i^{t_{n-2}}, ...) \qquad \mathbf{T}_{N,i} = (t_n, t_{n-1}, t_{n-2}, ...) \tag{6}$$

The position $\tilde{\mathbf{p}}_i^{t0}$ of a node N_i is then linearly estimated for time t_0 based on its history $\hat{\mathbf{P}}_i$ by

$$\tilde{\mathbf{p}}_i^{t_0} = \hat{\mathbf{p}}_i^{t_A} + (\hat{\mathbf{p}}_i^{t_A} - \hat{\mathbf{p}}_i^{t_B}) \cdot \kappa_{p_i}^{t_0} = \underbrace{\mathbf{p}_i^{t_A} + (\mathbf{p}_i^{t_A} - \mathbf{p}_i^{t_B}) \cdot \kappa_{p_i}^{t_0}}_{\mathbf{p}_i^{t_0}} + \boldsymbol{\varepsilon}_{\tilde{p}_i^{t0}} \tag{7}$$

Here $\kappa_{p_i}^{t_0}$ depicts the factor derived from the two selected timestamps t_A and t_B obtained from the values in $\mathbf{T}_{N,i}$ in relation to t_0 same as for the distance estimation (see eq. 4). The distance between two nodes N_i and N_j can be obtained from the the estimated positions $\tilde{\mathbf{p}}_i^{t0}$ and $\tilde{\mathbf{p}}_j^{t0}$ of the nodes according to

$$\tilde{\delta}_{ij}^{t_0} = \mid \tilde{\mathbf{p}}_i^{t0} - \tilde{\mathbf{p}}_j^{t0} \mid = \underbrace{\mid \mathbf{p}_i^{t0} - \mathbf{p}_j^{t0} \mid}_{\delta_{ij}^{t0}} + \varepsilon_{\tilde{\delta}_{ij}^{t_0}} \tag{8}$$

2.2 Verification

2.2.1 Edge plausibility check

Edges can be verified if two estimations for the length of an edge (one from distance measurements, one from position information) exist at the observing node. The estimated distance deviation for edge E_{ij} is then given by

$$\tilde{\Delta}_{ij}^{t_0} = \tilde{d}_{ij}^{t_0} - \tilde{\delta}_{ij}^{t_0} \tag{9}$$

While $d_{ij}^{t_0} - \delta_{ij}^{t_0}$ representing the distance deviation without position and distance uncertainties denoted as $\Delta_{ij}^{t_0}$. The plausibility $\mu_{ij}^{t_0}$ of edge E_{ij} is considered to be true at time t_0 if the distance deviation is below a decider threshold Δ_{lim}.

$$\mu_{ij}^{t_0} = \begin{cases} 1 & if \quad |\tilde{\Delta}_{ij}^{t_0}| \leq \Delta_{lim} \\ 0 & else \end{cases} \tag{10}$$

Hence, the decider threshold Δ_{lim} can be used to determine the required sensor performances σ_p and σ_d. The other way around, given sensor performances determine the decider threshold and thereby the accuracy of the plausibility check.

2.2.2 Node plausibility check

The plausibility information of the edges allow for the derivation of a plausibility value for nodes. If at time t_0 all predicted edges are given by $M_E^{t_0}$ (see section 2.4), for each node N_i the adjacent predicted edges at time t_0 are given by $B_i^{t_0}$. The amount of predicted edges containing node N_i is depicted as $n_{E,i}^{t_0}$. The amount of plausible edges containing node N_i is given by

$$n_{E,pl,i}^{t_0} = \sum \mu_{ij}^{t_0} \quad \forall \quad E_{ij} \in B_i^{t_0} \tag{11}$$

Finally the trust score resulting from the edge plausibility check (SCE) at a specific node N_i and timestamp t_0 is then given by the fraction of plausible edges $n_{E,pl,i}^{t_0}$ divided by total amount of adjacent predicted edges $n_{E,i}^{t_0}$

$$SCE_i^{t_0} = \frac{n_{E,pl,i}^{t_0}}{n_{E,i}^{t_0}} \tag{12}$$

As an example, Figure 1 shows a schematic network of 9 nodes and the corresponding edges established by the network protocol in between them. If the position of one node is falsified (in this case node A moves by the distance r_f to the position of A') and the falsified position is distributed within the network, the length of the adjacent edges will yield deviations when predicted from the position information if compared to the direct distance measurements. Hence, the trust score SCE for node A as well as for the nodes neighbouring node A show SCE values < 1, while the SCE value for node A itself is zero (numbers above the nodes).

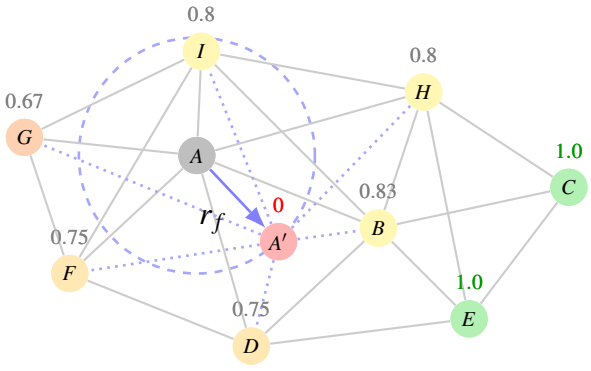


Figure 1 – Schematic of plausibility check and trust score (SCE) for falsified position of node A by r_f (note: it is assumed that all adjacent edges to node A are classified as implausible).

Same as for the edge plausibility, if the node trust score is below a threshold SCE_{lim}, the node can be considered implausible.

$$\xi_i^{t_0} = \begin{cases} 1 & if \quad SCE_i^{t_0} \leq SCE_{lim} \\ 0 & else \end{cases} \tag{13}$$

In the example above, by setting $SCE_{lim} = 0.5$ node A would be flagged as implausible $\xi_A = 0$ and mitigation measures might be put in place, while the other nodes would still be classified as plausible. This of course is an ideal case, and it is assumed, that by shifting A to A' all adjacent edges are classified as implausible. However, this might not necessarily be the case in real life networks. It depends on the constellation and number of edges whether a position shift will result in implausible edges such that the SCE value is reduced sufficiently in order to classify the node as implausible.

2.3 Timing constraints and prediction modes

As the extrapolation of the distances and positions is linear, not all values in $\hat{\mathbf{D}}_{ij}$ and $\hat{\mathbf{P}}_{ij}$ can be used for extrapolation. On the one hand, the two values that are used should not be too close to each other in order to reduce the prediction error. On the other hand, they should not exceed a certain age, as then the assumption of linear movement cannot be anticipated any more. To account for these constraints two factors t_R and t_C are introduced.

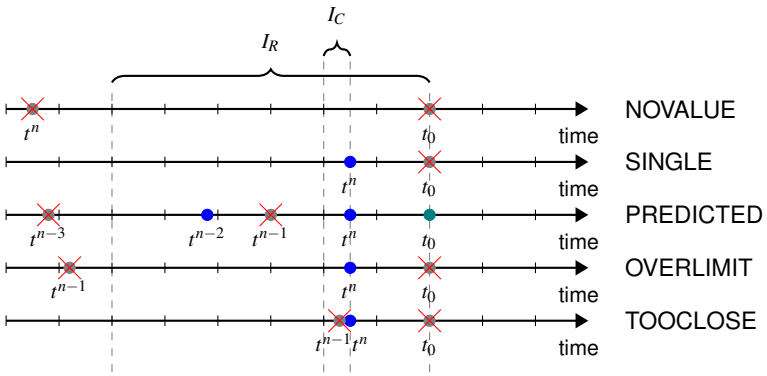


Figure 2 – Prediction modes resulting from selection of measurements based on time limitations t_R and t_C. (unused values: crossed, valid values: blue, predicted values: green).

The interval $I_R = [t_0, t_0 - t_R]$ limits the total age of a measurement to be taken into account for the prediction. Also, measurements falling in the interval $I_C = [t_n, t_n - t_C]$ will not be considered as they are to close to the latest measurement t_n and thus yield large prediction errors. If more than two values are applicable for the prediction the latest and the first value within I_R are considered. Based on these assumptions, five prediction modes for positions and distances at a time t_0 can be distinguished as presented in Table 1 and Figure 2.

mode	description
NOVALUE	no value lies within I_R
SINGLE	only one value is available and lies within I_R
OVERLIMIT	more values are available but only one lies within I_R
TOOCLOSE	more than one value lie within I_R but they are within I_C
PREDICTED	more than one value lie within I_R and not within I_C

Table 1 – Overview over prediction modes.

Additionally, for distances another mode can be distinguished that occurs if a distance measurement for an edge exists but information about the corresponding nodes is still missing.

2.4 Known nodes and edges

At a specific timestamp t_0 a distinct set of edges and nodes is known by the observing node (see sections 2.1.1and 2.1.2). Depending on the availability of data, only a fraction of these nodes and edges can be properly predicted. At time t_0 all predicted nodes and edges at the observing node N_i are given by $M_{N,i}^{t_0}$ and $M_{E,i}^{t_0}$. The known number of nodes and edges at node N_i at t_0 is depicted as $k_{N,i}^{t_0}$ and $k_{E,i}^{t_0}$. If a node is known to the observing node and cannot be predicted, it poses a potential threat and has to be treated as implausible in order to be on the safe side. Otherwise a node stopping transmission of data would not be identified as potentially malicious or corrupted. Therefore, we define the node prediction rate $\beta_{N,i}^{t_0}$ as the fraction of total predicted nodes $m_{N,i}^{t_0}$ and the known nodes from $k_{N,i}$.

$$\beta_{N,i}^{t_0} = \frac{m_{N,i}^{t_0}}{k_{N,i}^{t_0}} \tag{14}$$

Hence, $\beta_N = 1$ means all positions of nodes known by the observing node can be properly predicted. This implies, that the network needs to be designed in a way, that the prediction rate is permanently close or equal to one. Otherwise false alarms will increase drastically. As a consequence, nodes disconnecting from the network need to sign out allowing all other nodes to delete them from their list of known nodes $\mathbf{K}_N$. The edge prediction rate β_E can be defined equally. However, depending on the network topology and protocols the edges might change over time if the positions of nodes change. Therefore, known edges are limited to the interval I_R.

2.5 Data model

2.5.1 Messages

To allow for the creation of a situational overview two basic messages are defined. This is on the one hand the *initial position report* (IPR) and on the other hand the *completed position report* (CPR).

msg.	data type	variable	description	owner	size
IPR	POS-RAW	POS_s	GNSS position of sending node	self	10 Byte
	ID	ID_s	ID of sending node	self	4 Byte
	TS	TS_s	timestamp of sending message	self	8 Byte
CPR	POS-RAW	POS_s	GNSS position sending node	other	10 Byte
	ID	ID_s	ID of sending node	other	4 Byte
	TS	TS_s	timestamp of sending message	other	8 Byte
	TS	TS_r	timestamp of receiving message	self	8 Byte
	ID	ID_r	ID of receiving node	self	4 Byte

Table 2 – Message descriptions for position reports.

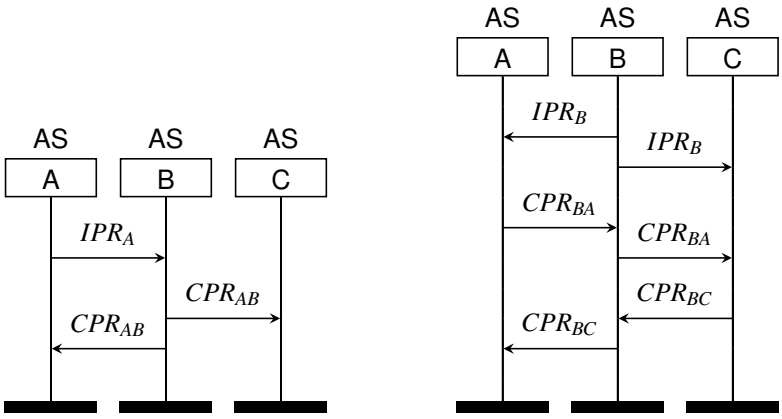
Table 2 summarizes both messages in terms of contained data and size. IPR messages contain a timestamp, ID and position of the sending node. CPR messages contain the values of the IPR message and additional information about the receiving timestamp and receiving node. The size of all data is based on the MAVLINK common message set (12). The contained timestamps represent the time when the first bit of a message is sent t_{sent} and when the first bit of a message is received t_{rec}. The distance between sender and receiver can then be calculated as

$$\hat{d} = \frac{t_{rec} - t_{sent}}{c} \tag{15}$$

where c is the speed of light. While the timestamps do not necessarily have to represent the real-world time, they have to be synchronized within the network. This requires each UAV to perform a synchronization procedure upon joining the network. To measure the distances between UAVs, a time accuracy in the range of nanoseconds is required. Every inaccuracy in timing manifests itself as an error in the distance measurement accordingly.

2.5.2 Sequences

Figure 3 shows two exemplary sequence diagrams of message transmission. It can be seen, that CPR messages are just passed through if received, whereas IPR messages are modified by the nodes upon first reception. In the left example AS_A broadcasts an initial position report to all neighbours in range (IPR). If the IPR message is received by AS_B the reception timestamp is attached to the message creating a completed message (CPR). Now AS_B starts broadcasting the CPR message to all other nodes in the network including AS_A.
Hence, a full network graph is obtained including all edges between AS within range of each other. However, due to the inherent unreliability of wireless communication information might be lost or arrive late influencing the prediction of positions and distances.



(a) AS_A as sender. (b) AS_B as sender.

Figure 3 – Sequence diagrams for situational awareness.

3. Simulation

To verify our approach, we perform the plausibility check and trust calculation on the data model described in our paper using simulated mobility data for UAV traffic movements.
In the simulation we generate data traffic for each UAV based on our data model presented above and flood all messages through the network using appropriate protocols. Messages on individual nodes are recorded and the algorithm is applied on the corresponding recorded data.

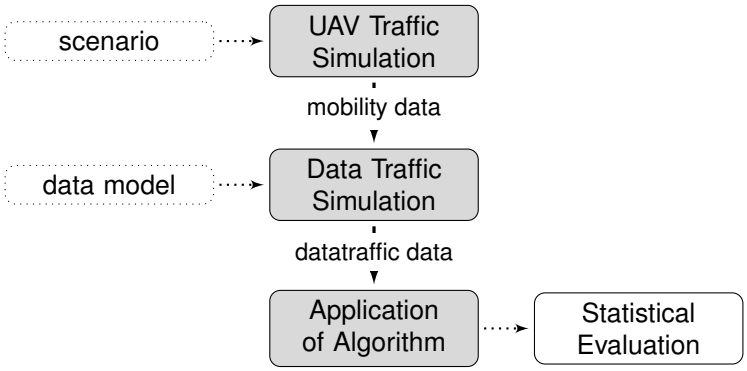


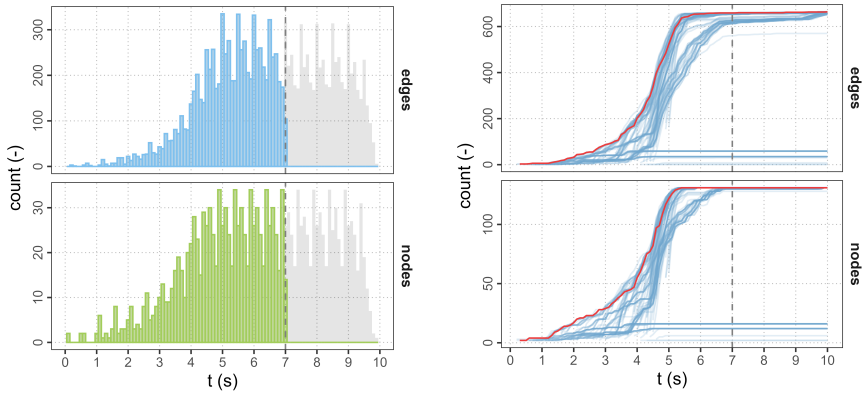
Figure 4 – Simulation workflow.

First, based on an UAV traffic demand model as described in (13) we generate UAV mobility data for package delivery service in urban areas. The movement of every UAV is simulated using an agent based fast time simulation environment and the mobility data is then handed over to a network simulation model, that simulates the network traffic using the data model as described in Section 2.5 . The network simulation is performed with a discrete-event simulator based on SimPy (14). Here every UAV is equipped with an idealized transceiver that implements a collision-free time-division multiple-access scheme. A unit disk radio is assumed with a communication range of 3km to ensure good connectivity of the network. Considering that UAVs have mostly line-of-sight communication and specialized antennas such ranges are achievable with modern communication technologies.
During the simulation, every UAV follows the trajectory described in the mobility data and generates an IPR every 0.5 s which is then converted to CPR by neighbouring UAVs. The CPRs are flooded through the entire network using Contention-Based Flooding (15). In Contention-Based Flooding, a UAV receiving a CPR for forwarding calculates a timer inversely proportional to the distance to the sender. If the UAV overhears the forwarding of the same CPR by another UAV, it cancels its timer. Otherwise, it forwards the CPR after its timer has expired. In our simulation, this ensures efficient

dissemination of CPRs throughout the network, nevertheless in congested scenarios, more efficient flooding protocols such as Rate Decay Flooding (16) can be applied at the cost of higher delays.
In a next step during the data traffic modelling the data at a specific airborne station is recorded and is then used to calculate the plausibility of the network as seen from that particular station. Here the position and distance uncertainties are applied to the data so that the total resulting errors can be estimated.
The simulation was run for 10 s and comprised 170 flights resulting in 2605 emitted IPRs and 20151 CPRs. The start time of the simulation was selected at 11:00 am of the scenario presented in (13), representing one of the busiest times.

4. Results

4.1 Time series analysis



(a) Received information at AS_1; (total: gray, until $t = 7\,s$: coloured).

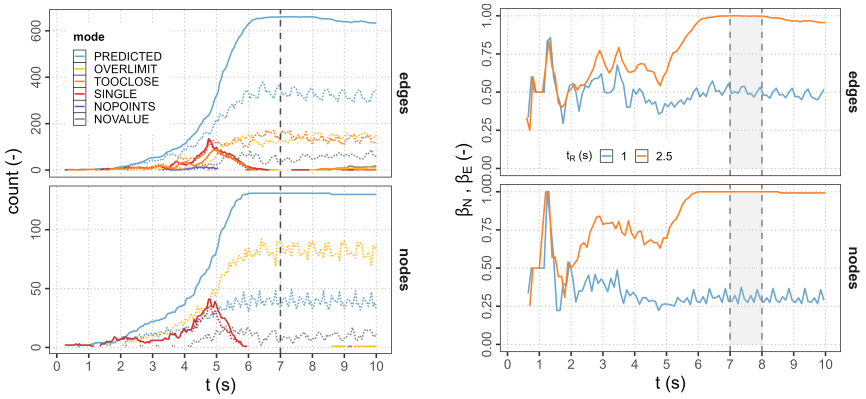
(b) Total known nodes and edges ($k_{N,i}^{t_0}$, $k_{E,i}^{t_0}$) by AS (AS_1, red).

Figure 5 – Received information and number of known nodes over simulation time t; note: y-axis scales are not equal.

Received information
The information reception protocol as acquired from the simulation for a selected AS (in the following denoted as AS_1) is presented in Figure 5a. It can be seen, that after a ramp-up phase new information for nodes and edges are continuously received, however, a wavy pattern can be observed. This results from the distance measurement mechanism. Since every node that receives an IPR completes it to a CPR and broadcasts it through the network, regions of the network with a high node density cause many CPRs to propagate through the network at the same time. In general, more information about edges is received than about nodes. This is due to the data messages defined in section 2.5that lead to many duplicates for the position information. In Figure 5a the total received information available in the simulation is shown in grey, whereas the relevant information received until a specific timestamp (here $t = 7\,s$) is coloured.

Known nodes and edges
Figure 5b shows the total known number of nodes ($k_{N,i}^{t_0}$ and edges $k_{E,i}^{t_0}$) for all AS in the scenario (AS_1 is coloured in red). It can be observed, that after a certain time, while the network is connecting itself, most of the nodes and edges are known to the majority of AS, as most of the values converge towards a common level. However, some lines converging to lower levels, indicate, that from a network perspective several clusters of AS exist that are not interconnected to each other.

Prediction modes
Based on the simulation results the algorithm can be applied to the data. In our case we considered the first 10 seconds after the selected AS connects to the network and starts to receive messages. Figure 6a shows the prediction modes for the nodes and edges known to AS_1 over time. It can be seen, that it takes more than five seconds for AS_1 in order to receive values for all nodes and edges



(a) Prediction modes at AS_1; $t_R = 1\,s$ (dotted); $t_R = 2.5\,s$ (solid).

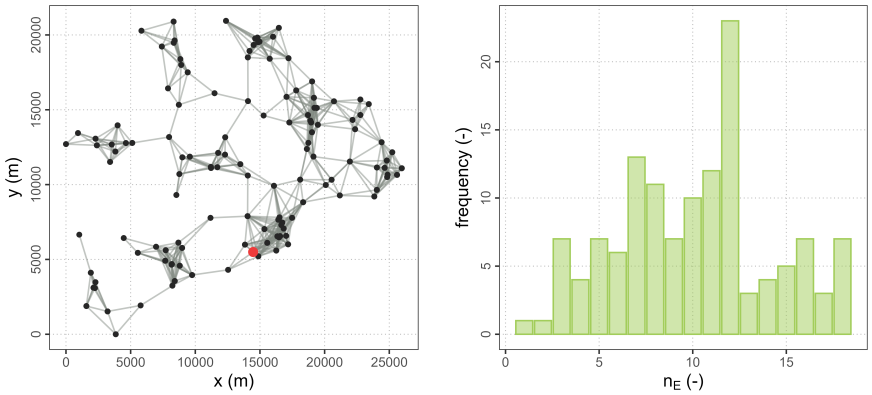
(b) Prediction rates $\beta_E^{t_0}$ and $\beta_N^{t_0}$ for different values of t_R.

Figure 6 – Prediction modes and rates over simulation time t as seen from AS_1; note: y-axis scales are not equal.

in the network. During this period for a considerable amount of nodes and edges only one value is available (SINGLE). Also, the wavy pattern can be observed as in Figure 5a. The prediction status strongly varies by t_R. Figure 6a shows the curves for $t_R = 1\,s$ (dashed) and $t_R = 2.5\,s$ (solid). For $t_R = 1\,s$ the number of OVERLIMIT nodes is very high stating, that information updates are not coming in in time. It can be seen, that only if t_R is large enough, all nodes can be properly predicted. However, as stated above, t_R should not exceed a certain limit, as then the linear extrapolation used might lose validity.

Prediction rate

Figure 6b shows the prediction rates for different settings of t_R. It can be seen, that for $t_R = 1\,s$ only a fraction of the known edges and nodes can be predicted. For $t_R = 2.5\,s$ considerably higher rates are achieved. In the interval $I = (7\,s, 8\,s)$ (shaded area) even a prediction rate of 1 is obtained for both nodes and edges. For $t \geq 8\,s$ the rate decreases again indicating that nodes are disconnecting from the network (e.g. UAVs are landing and shutting down) and edges are rearranging. Based on this analysis it seems reasonable to select $t_R = 2.5\,s$ and the timestamp $t = 7\,s$ for further detailed evaluation, as here AS_1 has knowledge of the majority of nodes and edges and all edges and nodes can be predicted properly.



(a) Snapshot of network topology at $t = 7\,s$. (AS_1: red).

(b) Number of neighbours n_E for all nodes of the network at $t = 7\,s$.

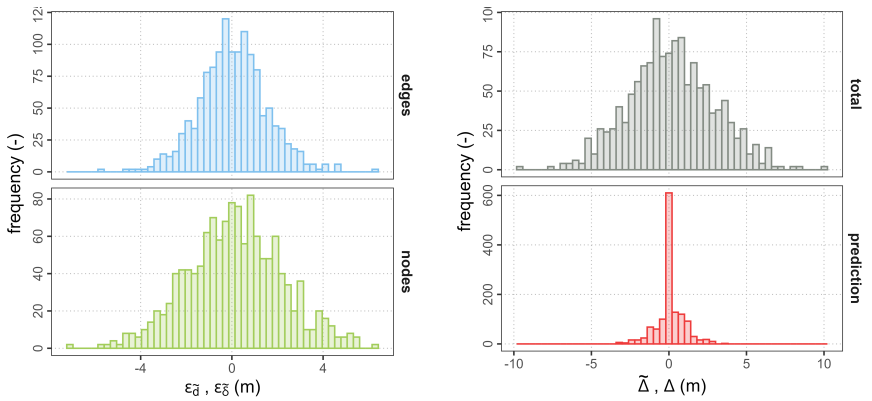
Figure 7 – Network topology and number of neighbours as seen from AS_1.

4.2 Network status

The network state as seen from AS_1 at timestamp $t = 7\,s$ is shown in Figure 7a. The graph created by distributing position and distance information using Contention-Based Flooding (CBF) as described in section 3is clearly visible. Figure 7b shows a histogram of n_E. While some nodes are connected to considerably many neighbours (up to 18), some nodes are only attached to the network by a single node. In total 131 nodes and 660 edges are known to AS_1 at this particular timestamp.

4.3 Errors and distributions

For the exemplary scenario and based on the assumptions made above, the values for the prediction errors can be calculated for all predicted values based on point or distance information.



(a) Histograms for $\varepsilon_{\tilde{d}}$ (top) and $\varepsilon_{\tilde{\delta}}$ (bottom).

(b) Histograms for $\tilde{\Delta}$ (top) and Δ (bottom).

Figure 8 – Distributions for $t = 7\,s$, $\sigma_d = 1\,m$, $\sigma_p = 1\,m$, $t_R = 2.5\,s$; as seen from AS_1.

Figure 8a shows the histograms of the resulting values for $\varepsilon_{\tilde{d}}$ and $\varepsilon_{\tilde{\delta}}$ using $\sigma_d = 1\,m$ and $\sigma_p = 1\,m$. It can be observed, that the values are normally distributed around zero. However, the width of the distribution is higher for the distance estimation by nodes as here a two-dimensional extrapolation is applied resulting in larger uncertainties accordingly. Figure 8b shows the density function of the resulting values for the distance deviations $\tilde{\Delta}$ and Δ. As the errors of both distance estimations sum up, the values for $\tilde{\Delta}$ are higher than the errors observed in Figure 8a and the distribution is shifted slightly towards positive values. The distribution of the prediction error Δ shows, that even without uncertainties in the measurements of position and distance, errors are introduced just by the linear extrapolation.

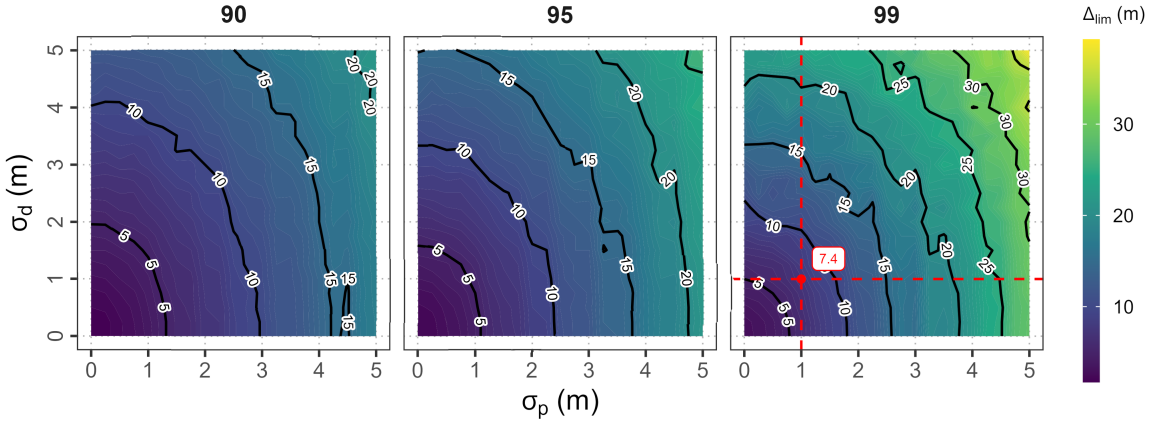


Figure 9 – Contour plots for Δ_{lim} and the 90th, 95th and 99th percentile of true negatives over the position and distance measurement uncertainties σ_p and σ_d.

4.4 Edge Classification

Based on the error distribution a classification of the edges can be obtained according to equation 10. Depending on the decision level Δ_{lim} an edge is thereby classified as plausible or not.
Figure 9 shows the contour plots for Δ_{lim} and the 90th, 95th and 99th percentile of the true negatives over the position and distance measurement uncertainties σ_p and σ_d (derived for the interval $I = (7\,s, 8\,s)$ as indicated in Figure 6b). It can can be observed, that as expected the threshold value Δ_{lim} increases if σ_d or σ_p increase. However, a stronger increase for σ_p can be observed, what seems reasonable as the position uncertainty in the two dimensions adds up. The higher the desired trust level, the larger Δ_{lim} will turn out to be.

4.5 Node Classification and Manipulation

Since one node can exhibit several neighbours (see Figure 7b), it needs to be decided, when a node is classified as implausible. Figure 1 indicates, that if a node position is altered by the distance r_f, several edges might be affected in terms that they turn implausible. This depends on the constellation of nodes. Another effect that occurs is that not only the falsified node obtains an altered SCE value, also the nodes sharing the implausible edges are affected. Hence, only from the concrete situation a falsified node position can be identified correctly. Here SCE_{lim} could be used to identify the severity of the manipulation. For $\sigma_p = 1\,m$ and $\sigma_d = 1\,m$ we applied an additional offset r_f to a randomly selected node in a random direction (see Figure 1). The CDFs for the SCE values of the thus selected manipulated nodes a variation of r_f based on 750 samples is shown in Figure 10. Here $\Delta_{lim} = 7.4\,m$ was chosen as obtained from Figure 9 for $\sigma_p = 1\,m$ and $\sigma_d = 1\,m$ and the 99th percentile.

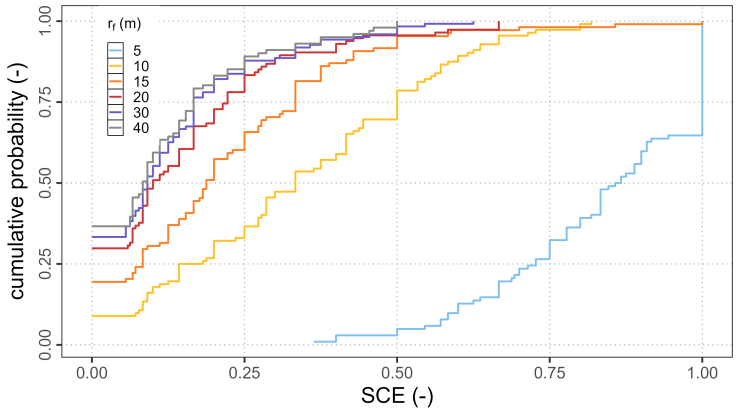


Figure 10 – CDFs of SCE score for $t = 8\,s$, $\sigma_d = 1\,m$, $\sigma_p = 1\,m$, $t_R = 2.5\,s$, $\Delta_{lim} = 7.4\,m$; for changing offsets r_f as seen from AS$_1$.

It can be seen, that while an offset $r_f = 5\,m$ yields a large amount of false positives ($SCE = 1$), offsets $r_f \geq 20\,m$ lead to all nodes being correctly classified as implausible. With other words, a falsified position reading with $r_f \geq 20\,m$ offset from the true node position will with very high probability be identified as implausible.

5. Conclusion and Outlook

In this paper we presented a method to check the plausibility of positions in a flying ad-hoc network based on separate position and distance measurements. In order to prove the applicability of this method, we adapted a simulation environment and applied it to simulated drone traffic for an urban package delivery scenario. The obtained mobility data was then used along with the data traffic generated by a set of defined data messages to enable the plausibility check. The received simulated data traffic at a specific node was then analysed to assess the method. We showed, that depending on the measurement uncertainties for position and distance an uncertainty of the distance deviation occurs that among others depends on the timing of the received messages. Based on this data we showed, that a threshold for the distance deviation can be identified at which edges can be classified as implausible without having too many false positives. This threshold seems to be in reasonable domains to enable a detection of nodes with improper position reports. Furthermore, we showed

how to derive a trust score for nodes based on the plausibility of adjoining edges. The threshold for this score, however, strongly depends on the local network topology.

Our work presents a first step toward the plausibility check based on independent distance measurements, therefore, more studies are necessary to fully cover the parameter space and identify an optimal parameter setting for the algorithm. Also, more advanced prediction models for the movement of nodes might be feasible to implement in the future along with expanding the simulation from two to three dimensions. As the method primarily is intended to address GNSS spoofing or malfunctioning equipment, the possibility of malevolent parties necessitates a detailed threat modelling. Finally, experiments with adequate communication and distance measurement technology will prove the applicability of the method in real world applications.

Funding

The work presented in this paper was part of the project VEREDUS that was funded by the German Ministry of Economic Affairs and Climate Action (BMWK) under the national aviation research programme (LuFo V-3) under the grant agreement no. 20Q1939H.

Contact Author Email Address

tobias.marks@dlr.de

Copyright Statement

References

[1] T. Benarbia and K. Kyamakya, “A Literature Review of Drone-Based Package Delivery Logistics Systems and Their Implementation Feasibility,” *Sustainability*, vol. 14, no. 1, 2022.

[2] R. Kellermann, T. Biehle, and L. Fischer, “Drones for parcel and passenger transportation: A literature review,” *Transportation Research Interdisciplinary Perspectives*, vol. 4, p. 100088, 2020.

[3] European Commission, “Commission implementing regulation (eu) 2019/947 of 24 may 2019 on the rules and procedures for the operation of unmanned aircraft,” tech. rep., 05 2019.

[4] Y. Zeng, R. Zhang, and T. J. Lim, “Wireless communications with unmanned aerial vehicles: opportunities and challenges,” *IEEE Communications Magazine*, vol. 54, no. 5, pp. 36–42, 2016.

[5] S. Benfriha, N. Labraoui, H. B. Salameh, and H. Saidi, “A survey on trust management in flying ad hoc networks: Challenges, classifications, and analysis,” in *2023 Tenth International Conference on Software Defined Systems (SDS)*, pp. 107–114, 2023.

[6] C. F. E. de Melo, T. Dapper e Silva, F. Boeira, J. M. Stocchero, A. Vinel, M. Asplund, and E. P. de Freitas, “UAVouch: A Secure Identity and Location Validation Scheme for UAV-Networks,” *IEEE Access*, vol. 9, pp. 82930–82946, 2021.

[7] T. Leinmuller, E. Schoch, and F. Kargl, “Position verification approaches for vehicular ad hoc networks,” *IEEE Wireless Communications*, vol. 13, no. 5, pp. 16–21, 2006.

[8] K. Fuger, K. Kuladinithi, M. Sood, and A. Timm-Giel, “Feasibility study on position verification in urban uav networks,” in *2023 33rd International Telecommunication Networks and Applications Conference*, pp. 38–43, IEEE, 2023.

[9] J. Naganawa, “Theoretical Analysis of Position Report Verification using Distance-based Localization,” in *2020 International Symposium on Antennas and Propagation (ISAP)*, pp. 673–674, 2021.

[10] C. Luo, S. I. McClean, G. Parr, L. Teacy, and R. De Nardi, “Uav position estimation and collision avoidance using the extended kalman filter,” *IEEE Transactions on vehicular technology*, vol. 62, no. 6, pp. 2749–2762, 2013.

[11] P. Kaniewski, R. Gil, and S. Konatowski, “Estimation of UAV position with use of smoothing algorithms,” *Metrology and Measurement Systems*, 2017.

[12] MAVLINK, “MAVLINK Common Message Set,” 2023.

[13] T. Marks and K. Fuger, “Using Generic Cities to assess Flying Ad-hoc Networks (FANETs) in Urban Environments (under review),” *CEAS Aeronautical Journal*, 2024.

[14] SimPy Team, “Simpy documentation,” 2014.

[15] H. Füßler, J. Widmer, M. Käsemann, M. Mauve, and H. Hartenstein, “Contention-based forwarding for mobile ad hoc networks,” *Ad Hoc Networks*, vol. 1, no. 4, pp. 351–369, 2003.

[16] K. Fuger and A. Timm-Giel, “On the feasibility of position-flooding in urban uav networks,” in *2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring)*, pp. 1–5, IEEE, 2023.