doi:10.1109/TMC.2016.2561291

Route or Carry: Motion-Driven Packet Forwarding in Micro Aerial Vehicle Networks

Mahdi Asadpour, Karin Anna Hummel, Domenico Giustiniano, and Stefan Draskovic

Abstract—Micro aerial vehicles (MAVs) provide data such as images and videos from an aerial perspective, with data typically transferred to the ground. To establish connectivity in larger areas, a fleet of MAVs may set up an ad-hoc wireless network. Packet forwarding in aerial networks is challenged by unstable link quality and intermittent connectivity caused by MAV movement. We show that signal obstruction by the MAV frame can be alleviated by adapting the MAV platform, even for low-priced MAVs, and the aerial link can be properly characterized by its geographical distance. Based on this link characterization and making use of GPS and inertial sensors on-board of MAVs, we design and implement a motion-driven packet forwarding algorithm. The algorithm unites location-aware end-to-end routing and delay-tolerant forwarding, extended by two predictive heuristics. Given the current location, speed, and orientation of the MAVs, future locations are estimated and used to refine packet forwarding decisions. We study the forwarding algorithm in a field measurement campaign with quadcopters connected over Wi-Fi IEEE 802.11n, complemented by simulation. Our analysis confirms that the proposed algorithm masters intermittent connectivity well, but also discloses inefficiencies of location-aware forwarding. By anticipating motion, such inefficiencies can be counteracted and the forwarding performance can be improved.

Index Terms—Micro aerial vehicle networks, motion-driven packet forwarding, location-aware delay tolerant networking

1 Introduction

MICRO aerial vehicles (MAVs) are small unmanned aerial vehicles of a weight up to a few kilograms that feature embedded computing, wireless communication, sensors, and small cameras, ready to gather information and transmit often large-sized data to a ground station [1], [2]. MAVs are increasingly adopted in a variety of civilian domains, such as surveillance, farmland monitoring, search and rescue missions, and entertainment. In time-critical missions, the delay of data transmission is a key factor to consider [3].

Whereas the communication of a single MAV is already well understood, MAV fleets necessary to cover larger areas pose new research questions for networking. In principle, multiple MAVs may provide connectivity and high-throughput transmission in an area by creating an ad-hoc, multi-hop *flying wireless network*. Yet, a unique characteristic of aerial communication is the continuous movement of the MAVs either towards a waypoint or due to flight dynamics, which impairs the quality of the wireless links [1]. Frequently changing MAV link quality and disconnections impact end-to-end transmission more than in traditional mobile ad-hoc networks. It is currently unclear how many of the vast approaches published on multi-hop networking in non-aerial communications can be re-used here.

- *M. Asadpour and S. Draskovic are with the Department of Information Technology and Electrical Engineering, ETH Zurich, Switzerland. E-mail: mahdi.asadpour@alumni.ethz.ch, stefan.draskovic@tik.ee.ethz.ch.*
- *K.A. Hummel is with the Institute of Telecooperation, JKU Linz, Austria. E-mail: karin_anna.hummel@jku.at.*
- *D. Giustiniano is with the IMDEA Networks Institute, Spain. E-mail: domenico.giustiniano@imdea.org.*

Manuscript received 24 July 2015; revised 16 Mar. 2016; accepted 21 Mar. 2016. Date of publication 3 May 2016; date of current version 2 Feb. 2017. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TMC.2016.2561291

Packet forwarding in aerial networks can also take advantage of the fact that MAVs are flying robots which report their location frequently and allow control of their movement. MAVs may act as a communication relay and also as a communication ferry [4]. While communication relaying is the classical approach to extend a network by proper placement of relay nodes (e.g., employed in sensor networks [5]), communication ferries move data physically to the destination or next relay node [3]. The ferry concept shows similarities to the concept of a collector or throwbox [6], which is a stationary, often battery powered system deployed in specific places of disconnected regions to increase connectivity by intermittently storing data. Yet, ferries are mobile and not operating long-term as traditional throwboxes do.

This particular setting of MAV networks requires a rethinking of routing protocols. Among the classes of existing routing schemes, traditional source, distance-vector, or link-state ad-hoc routing protocols fail in the highly dynamic aerial environment as they require an end-to-end path and a certain degree of link stability to converge [7], [8]. As location information of MAVs is available such as provided by the Global Positioning System (GPS), geographic routing based on forwarding packets to nodes that are spatially closer to the destination is a feasible approach [8], [9]. Pure geographic routing is, however, not adequate for networks that face intermittent connectivity. A known approach to target intermittent connectivity is delay-tolerant networking (DTN), which is in principle well suited for MAV networks [8], but pure DTN concepts often use a form of limited flooding based on stochastic knowledge about the moving nodes and usually feature long periods of disconnections. Traditional DTN schemes are optimized for this use case. Instead in the MAV case, node trajectories are to a high degree deterministic, the concept of

ferries can be leveraged, and disconnection times are considerably shorter than in traditional DTNs. Thus, probabilistic DTN multi-copy schemes do not fit well. In contrast, more sophisticated, movement-aware forwarding schemes that borrow basic disconnected operation from DTNs are promising. So far, extensions to incorporate delay-tolerant approaches in aerial networks have been proposed in [10], [11]. Yet, the difficulties to build an experimental MAV testbed have limited the realism of the implementation, and evaluation to simulation.

Our contributions target the design and development of a packet forwarding scheme and are detailed as follows:

- We model MAV networks as a graph of mobile aerial vehicles. We introduce a mobility model for MAVs that allows for linear short-time prediction of MAV trajectories. The model is based on the physical limitations of an MAV's in-flight behavior. The links in the MAV network model are either classical wireless links characterized by the transmission delay or DTN store-carry-forward links including the time the data is physically carried by the MAVs. We describe the transmission delay with an empirically derived throughput function that varies with geographical distance (Section 2).
- Using the network and mobility model, we design a location-aware DTN/geo-routing algorithm that basically routes a packet along the spatially shortest path to the destination, if it exists. Otherwise the packet is physically carried closer to the destination. The DTN/geo-routing algorithm is extended by two anticipatory heuristics that make use of predicted MAV movement (Section 3).
- We implement our forwarding algorithm in a testbed of quadcopters connected over a Wi-Fi IEEE 802.11n network, optimized for reduced signal obstruction (Section 4). We present the results of our measurement campaign with a small fleet of up to three copters and one ground station (Section 5). To the best of our knowledge, we are the first to practically investigate a location-aware packet forwarding algorithm with DTN support in outdoor fields. We complement our analysis with a simulation study of the algorithm compared to epidemic routing in a larger MAV fleet and discuss factors impacting the performance of the algorithm and its heuristics (Section 6).

2 Modeling MAV Networks

A network of MAVs is a wireless multi-hop network of aeronautical mobile nodes. The network ensures data communication among MAVs and to the ground. Different to nodes in other mobile ad-hoc networks, the MAVs are flying robots and their movement and actions are to a high degree mission-driven. Further, geographic positions of MAVs are usually known. In the following, we motivate the network model with a scenario. Then, we state general assumptions of our approach, introduce the mobility model, and describe the MAV network and link throughput model.

The mobility model and link throughput model are designed based on experimental 2D in-flight measurements.

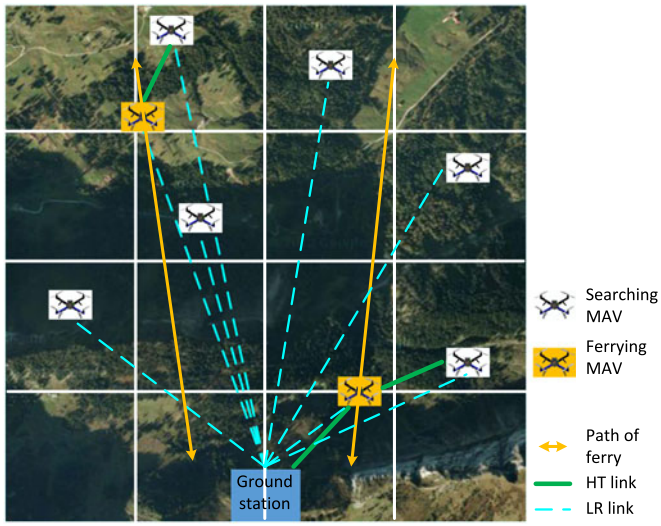


Fig. 1. Example search and rescue scenario with MAVs: Six MAVs are searching for an object and transfer images to the ground station using high-throughput links (HT links) with limited ranges. Further, long-range links (LR links) are leveraged for telemetry and control information. Two ferry MAVs establish HT connectivity by flying back and forth between the searching MAVs and the ground station.

An extension to 3D is possible, yet it requires additional experimental investigation of effects appearing in practice [12]. The network model is general and can integrate both 2D and 3D-variants of the models.

2.1 MAV Network Use Case Scenario

As an illustrative example, consider the scenario case where MAVs are employed in a search and rescue mission to screen an area and to provide geo-tagged camera images in order to spot a missing person or an object [13]. As depicted in Fig. 1, multiple MAVs are sent to different areas to take images. For further processing, the potentially large-sized images are sent via high-throughput short-range links to a ground station, which is stationary or only moderately moving. Each MAV (as well as the ground station) is always aware of its own position. Further, each MAV may change its flight behavior when receiving a command through an additional long-range low-throughput network.

In case an MAV moves out of communication range of the ground station (high-throughput network), connectivity has to be re-established. Flying back to the ground station is a simple, but power-hungry solution that does not scale well. Observing MAVs should use their scarce battery-power on mission-driven tasks instead of consuming it by moving closer to the transmission peer (in a testbed of quadcopters, we measured a power consumption of about $200 - 250$ Watt for autonomous flight, the dominating factor is the mechanical part of the copter [1]). Similarly, dense placement of relay MAVs for maintaining connectivity is a possible solution, however, this solution comes at high deployment and operational costs. By introducing ferry nodes to establish connectivity, most of the MAVs may search while the ferries move data physically before transmitting. Note that every MAV in the network may additionally relay messages when possible, including the searching MAVs.

2.2 MAV System and Network Assumptions

Our network model relies on the following assumptions:

- *Line-of-sight link characteristics:* Wireless links are assumed to have line-of-sight characteristics. Indeed, for safety reasons, the vast majority of outdoor MAVs can be considered to operate in flat areas when monitoring, e.g., farmland, or above buildings and trees, e.g., during search and rescue operations, thus having basically line-of-sight properties. Yet, as we have shown in past work, the quality of the Wi-Fi signal is also largely affected by signal obstructions of the MAV's frame and sub-optimal antenna properties [1]. To mitigate these effects, a customized MAV with arms transparent to signal propagation and light-weight external antennas is introduced (cf. Section 4.1). Now distance can be exploited as the main criterion for modeling aerial link throughput.
- *Out-of-band channel:* In addition to the high-throughput radio technology used for data traffic, we use another radio technology for control traffic. Such a channel should feature long range, but requires only low throughput. As many MAVs need a reliable communication channel for control and telemetry data for safe flight operation anyway, assuming the existence of an out-of-band-channel is reasonable [2], [14].
- *Availability of location and motion information*: The availability of a positioning and motion sensing technology is assumed such as provided by GPS and inertial measurement units (IMUs). To distribute geographic position and motion information, the out-of-band channel is leveraged. We exploit location information to determine the distance of a network link based on the Haversine formula which calculates the shortest geographical distance between two points on the earth surface. The individual MAVs may take distributed decisions based on the disseminated mobility information.
- *Mobility can be leveraged:* Two aspects of MAV mobility are assumed in our model. First, MAVs do not move at random allowing to predict near future MAV positions based on motion information. Then, controlled mobility is leveraged by the employment of ferry MAVs. Ferry MAVs may be sent to waypoints to enforce that messages can eventually reach the destination by store-carry-forward mechanisms. The control of MAV mobility is centralized mainly to assure safe operation and prevent collisions.
- *Sparse node deployment:* Sparse node deployment is assumed in MAV networks. Thus, relaying only is not sufficient to establish connectivity. Also overload and interferences are not a major problem and, thus, not included in the models.

Although not a principle assumption, network use cases of MAV fleets typically show an asymmetric data traffic flow towards the ground (one ground device). This makes it in particular feasible to define an appropriate path for ferries in a way that messages can eventually reach the ground device. Packet forwarding itself is neither limited nor optimized to an asymmetric data flow but is generally applicable to any type of MAV communication.

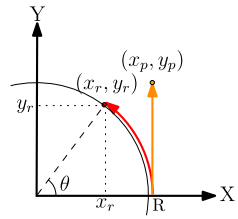


Fig. 2. Computation of the maximum horizontal prediction error.

2.3 MAV Mobility Model

MAV movement can be described by a linear, deterministic mobility model with memory that derives a position based on the MAV's current position, orientation, and speed. The model may be classified as a Gauss-Markov mobility model with strong memory for orientation and speed, where the future orientation and speed are exactly the same as the ones measured [15]. We aim to predict the trajectory of an MAV until a future point in time, which we term the *prediction time* $\mathcal{F}$. The predicted position at time $\mathcal{F}$ is found by linear extrapolation of the current orientation and speed of the MAV at the current geographic position.

We now compute the error bound for a horizontal change of direction, which is a likely change of orientation during operation when a safe altitude is reached. The MAV cannot change its direction "instantaneously" as it needs some time to adjust its sensors and propellers to perform a smooth directional change along a turning radius R, which can also not be arbitrary small in practice.

Prediction error bound of the mobility model (change of horizontal orientation). The worst-case scenario happens when the MAV is turning with the largest angle. To calculate the worst case error, we therefore consider that the MAV has a given turning radius R within the prediction time $\mathcal{F}$, resulting in a deviation from the expected pathway. The predicted position is denoted by (x_p, y_p) and the real position of the MAV is denoted by (x_r, y_r). As depicted in Fig. 2, without lack of generality prediction takes place when the MAV reaches position $(R, 0)$ and is heading north. The position predicted by linear extrapolation with prediction time $\mathcal{F}$ is $(x_p, y_p) = (R, v \times \mathcal{F})$, v is the MAV's speed. At worst, the MAV starts changing its direction right at $(R, 0)$. Given the MAV's turning radius R and assuming a constant speed v during prediction time $\mathcal{F}$, we calculate:

$$\begin{aligned}
\text{Predicted pos.:}(x_p, y_p) &= (R, v \times \mathcal{F}) \\
\text{Resultant angle:}\theta &= \frac{v \times \mathcal{F}}{R}, \text{with } 0 \leq \theta < 2\pi \\
\text{Real pos.:}(x_r, y_r) &= (\cos(\theta) \times R, \sin(\theta) \times R) \\
\text{Prediction error:}P_{error} &= \sqrt{(x_r - x_p)^2 + (y_r - y_p)^2}.
\end{aligned}$$

The *maximum* prediction error for MAVs in our testbed is exemplified for sample prediction times in Table 1. With increasing prediction time, the error also increases. We can conclude that even when looking four seconds into the future, $P_{error} = 7.92$ m is an acceptable error when considering also the accuracy of on-board sensors such as GPS as well as the impact of wind. When the MAV changes to hovering, the speed is gradually reduced, which decreases the prediction error gradually as well. When the speed is

TABLE 1
Sample Prediction Error Calculation for Different Prediction Times $\mathcal{F}$

Speed	$v = 4.5$ m/s
Turning radius	$R = 20$ m
$\mathcal{F} = 1$ s	$P_{error} \simeq 0.51$ m
$\mathcal{F} = 2$ s	$P_{error} \simeq 2.01$ m
$\mathcal{F} = 4$ s	$P_{error} \simeq 7.92$ m

decreased to half the default speed, i.e., 2.25 m/s in our testbed, $P_{error} \simeq 2.01$ m ($\mathcal{F} = 4$ s). The reduction happens over a time frame of about 3 to 5 seconds. When reaching the hovering position, the prediction error is zero.

The presented model is a micro-mobility model, which differs from other mobility models defined by a mission. Related to our motivating use case search and rescue, models describing mobility behavior of multiple agents in an emergency response mission are presented in [16], [17]. While these models are representative only in the respective use case mission, our mobility model is generally applicable.

2.4 MAV Network Model

We present a network model for MAVs as a weighted graph $\mathcal{G}$ of $\mathcal{N}$ mobile nodes. Every node possesses a geographic position (latitude, longitude, and altitude), orientation, and speed, making it possible to predict future motion. Aerial wireless links are set up between pairs of MAVs, which can be in different transmission phases while either moving or hovering. For each MAV with available data, we distinguish among the following different states of operation:

- *Transmit*: the MAV is in range of another MAV or the ground station. In this state, the MAV transmits the data according to the forwarding algorithm. The transmission time is denoted by T_{tx}.
- *Carry*: no other MAV or the ground station is in communication range and thus the MAV stores the data; the time the MAV physically carries the data is denoted by T_c.

The transmission delay of a packet on a single hop is calculated as the sum of the time the packet is carried and the time it takes to transmit the packet:

$$C_{delay} = T_c + T_{tx}. \tag{1}$$

Considering the mentioned states, we introduce "real links" that represent classical wireless links used to transmit data, and hypothetical "virtual links" that correspond to links that require carrying before data are transmitted.

Real links. Two nodes n_i and n_j with geographical distance d are assumed to be connected via a *real link*, if $d \leq \mathcal{D}$, with $\mathcal{D}$ being the transmission range of the nodes. The weight of the real link is represented by w_{ij} and expressed by the transmission delay that is expected on that link between nodes n_i and n_j:

$$w_{ij} = T_{tx} = \frac{M_{data}}{s(n_i, n_j)}. \tag{2}$$

Here, M_{data} is the amount of data that has to be transmitted, and $s(n_i, n_j)$ denotes the throughput of the single link between the two nodes n_i, n_j. An empirical link throughput function is derived in Section 2.5. In the case of routing in a connected network, the weighted graph is used to find the shortest path to deliver the message with minimal delay.

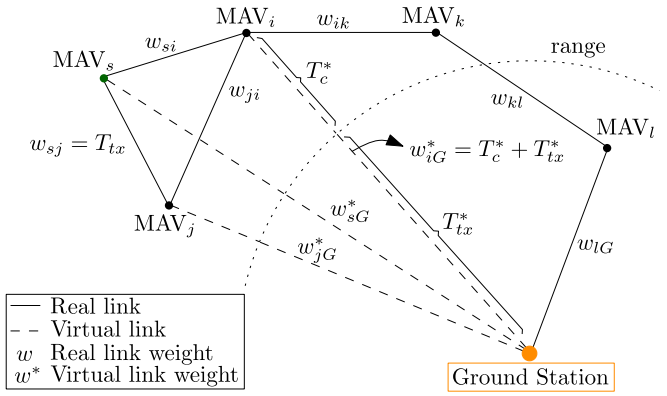


Fig. 3. Example weighted MAV graph (schematic view): Path options of MAV$_s$ to send data to the ground station; w_{xy} represents the weight of the real link between MAV$_x$ and MAV$_y$ and w_{xG}^* represents the weight of the virtual link between x and the ground station.

Virtual links. If no real link exists from node n_i to the destination of the message G, a hypothetical *virtual link* is defined for this node. The weight of the virtual link is calculated as the expected time node n_i needs to carry the data from its current position to come in transmission range of the destination G plus the transmission delay:

$$\begin{aligned} w_{iG}^* &= T_c^* + T_{tx}^*, \\ T_c^* = \frac{d_G}{v}, T_{tx}^* &= \frac{M_{data}}{s(\overline{n}_i, G)}. \end{aligned} \tag{3}$$

The expected carry time T_c^* depends on the distance d_G traveled by the node to reach the communication range $\mathcal{D}$ of G (assuming that G is not moving during this time), and the node's default speed v. The transmission delay is calculated similarly to Equation (2), $\overline{n}_i$ denotes node n_i when entering the transmission range of G.

On this MAV graph model, any traditional graph traversal algorithm can be invoked. Fig. 3 visualizes the path options of an MAV that wants to send data to the ground station in a sample graph. In the example, an end-to-end-path of real links exists between the sending MAV_s and the ground station. In addition, virtual links are depicted for all neighbors of MAV_s; the calculation of the weight of a virtual link is exemplified for MAV_i.

2.5 Modeling Link Throughput

We estimate the throughput of a link as a function of the known geographical distance d between the sending and the receiving MAV and denote it by $s(d)$. Due to the lack of a given throughput versus distance function, we model $s(d)$ as a parameterized logarithmic function derived from free space path loss. We fit it to empirical measurement results derived with two quadcopters in line-of-sight conditions flying at about the same altitude[1] as detailed in Fig. 4. At closest distance, the median measured throughput is about 60 Mbit/s. The Pearson correlation coefficient of throughput

1. Note that adaptations of the link throughput model may be needed when operating the copters at significantly different altitudes.

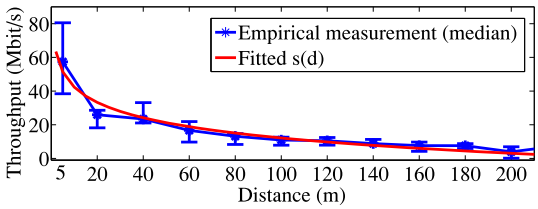


Fig. 4. Throughput versus distance measurement test between two flying quadcopters at a relative altitude of about 20 m (with a deviation of about 5 m in altitude due to flight dynamics and GPS errors), and fitted $s(d)$ function (cf. Equation (4)). Min and max error bars are respectively the 25 and 75 percent quartiles.

versus logarithm of distance yields -0.967. This result shows that the throughput has a very strong negative correlation to the logarithm of the distance and statistically approves the use of a logarithmic function. The derived empirical $s(d)$ function is given as follows ($s(d)$ in [bit/s], d in [m]):

$$s(d) = 10^6 \times (-9.09 \times \log_2(d) + 72.58). \tag{4}$$

The resulting R^2 error of the fitted function is 0.9496, which proves the very good fit of $s(d)$.

3 Motion-driven Packet Forwarding

We exploit sensor information of MAVs to design a location-aware packet forwarding algorithm that makes use of physical motion of MAVs. The algorithm works in both connected and intermittently connected networks. When connected, the algorithm routes a packet along the shortest path. Otherwise, greedy geographic forwarding extended by DTN mechanisms is used and data are carried by MAVs. The algorithm is termed $\mathrm{DTN}_{\mathrm{geo}}$. The DTN part of the algorithm is then extended by two heuristics that make use of anticipated future locations, as well as estimated link capacity and connection time. While the basic algorithm and the first heuristic that anticipates only future locations do not adapt their decision to the load or message queue size, the second heuristic prefers links that master the load best, i.e., that allow to transmit most of the available data. Both heuristics are light-weight by design in order not to challenge the embedded processing capabilities of MAVs.

3.1 $\mathrm{DTN}_{\mathrm{geo}}$ Algorithm

$\mathrm{DTN}_{\mathrm{geo}}$ is a location-aware packet forwarding approach with DTN support. Each MAV is aware of the geographic position of all MAVs and further of the global topology of the multi-hop MAV network. The MAV maintains a topology table, which is periodically updated by MAV status messages transmitted through the out-of-band channel. The topology table contains MAV IDs and real links with a weight of $w = T_{tx}$ (transmission time), as well as virtual links with a weight of $w^* = T_c^* + T_{tx}^*$ (carry time and transmission time), as detailed in Section 2.4. We employ a single-copy model, i.e., only one copy of a message exits in the network at a time.

In case the current list of neighbors and the sending queue are not empty, the MAV executes Algorithm 1, which first tries to find the shortest path from source to destination (*end-to-end routing*). If this path does not exist, it forwards each message in the queue to a neighbor determined by the shortest virtual link (*DTN-based forwarding*):

- *End-to-end routing*: The MAV analyzes the network topology to find the shortest path to the destination of the message by implementing Dijkstra's algorithm. If such a path exists, the MAV forwards the message to the neighbor that is a part of the shortest path.
- *DTN-based forwarding*: In case no end-to-end path is found, the MAV forwards the message to the neighbor with the smallest virtual link weight or keeps the message in case the MAV's own weight is equal or less than the weight of its neighbors. A smaller weight of a virtual link basically expresses physical proximity to the destination. Inspired by greedy geographic forwarding [9], a node is selected that can physically move data faster to the destination. In its simple form, the scheme has been proven to be effective in [8], [9].

Algorithm 1. $\mathrm{DTN}_{\mathrm{geo}}$ Algorithm.

```
procedure DTN_GEO(src, dst, M_data)        ▷ Sending M_data from src to dst
  if ∃ shortest path via N ∈ {Neighbors} then
    ForwardTo(N, M_data)                   ▷ Route
  else
    W* ← GetVLWeight(src, dst)
    H ← src
    for N ∈ {Neighbors} do
      if GetVLWeight(N, dst) < W* then
        W* ← GetVLWeight(N, dst)
        H ← N
      end if
    end for
    if H ≠ src then
      ForwardTo(H, M_data)                 ▷ DTN Transmit
    else
      StoreInQueue(M_data)                 ▷ DTN Carry
    end if
  end if
end procedure
```

3.2 Anticipatory Forwarding Heuristics

$\mathrm{DTN}_{\mathrm{geo}}$ considers only current positions and not future ones. Thus, the expected future capacity of a link and the connection time is not factored in. To counteract these limitations, two heuristics are introduced.

$\mathrm{DTN}_{\mathrm{close}}$ – *future proximity to destination.* We extend $\mathrm{DTN}_{\mathrm{geo}}$ by estimating the trajectory of an MAV in the prediction time frame $\mathcal{F}$ based on the linear mobility model introduced in Section 2.3. The node that is predicted to be then the closest node to the destination is selected.

Algorithmically, we compare all neighbors N of node n_i at current time t and select the next node n_j that fulfills the following condition:

$$\begin{aligned} &\underset{n_j \in N}{\arg\min}\ d_{n_j}(dst, t+\mathcal{F}), \\ &d_{n_j}(dst, t+\mathcal{F}) < d_{n_i}(dst, t+\mathcal{F}). \end{aligned} \tag{5}$$

In the formula, $d_{n_j}(dst, t+\mathcal{F})$ is the anticipated geographical distance of node n_j to the destination at the time $t+\mathcal{F}$, and $d_{n_j}(dst, t+\mathcal{F}) < d_{n_i}(dst, t+\mathcal{F})$ ensures that

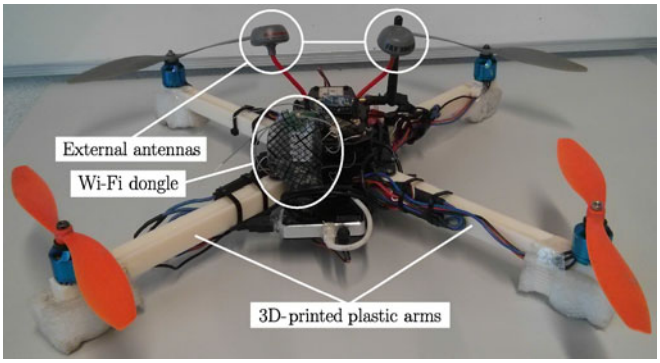


Fig. 5. Used quadcopter platform with 3D-printed plastic arms and on-board wireless package with two external circular antennas.

the selected node n_j will also be closer than n_i itself to the destination.

$\mathrm{DTN}_{\mathrm{load}}$ – *capacity to master load within connection time.* The connection time of a link and the throughput of that link determine the amount of data that can be transmitted. Again, we observe a prediction time period of $\mathcal{F}$. Assuming that a load of M_{data} [bit] are in the queue of MAV n_i ready to be sent, we calculate the following for each neighbor n_j:

$$\mathcal{B}_{n_j} = \frac{1}{M_{data}} \sum_{\tau=t}^{t+\mathcal{F}} s(d_{n_j}(n_i,\tau))\Delta\tau \tag{6}$$
$$d_{n_j}(n_i,\tau) < \mathcal{D}, \quad d_{n_j}(dst, t+\mathcal{F}) < d_{n_i}(dst, t+\mathcal{F}).$$

In the formula, $s(d_{n_j}(n_i,\tau))$ denotes the throughput [bit/s] as a function of the distance $d_{n_j}(n_i,\tau)$ between node n_i and its neighbor n_j; $\mathcal{D}$ is the maximum transmission range; τ represents the number of discrete time steps of duration $\Delta\tau$ between t and $t+\mathcal{F}$; and $d_{n_j}(dst, t+\mathcal{F}) < d_{n_i}(dst, t+\mathcal{F})$ enforces that the selected neighbor n_j is closer than the node itself to the destination at the time $t+\mathcal{F}$.

The resulting $\mathcal{B}_{n_j}$ is the expected capability of a link to handle M_{data} within the connection time. A larger value of $\mathcal{B}_{n_j}$ indicates (i) a better communication channel that would provide higher data rates, and/or (ii) a longer connection time that would permit more data exchange. In particular, a value equal or larger than one indicates that the queue can be depleted; the larger the value, the sooner the queue will be depleted. This heuristic therefore aims at selecting a neighbor that will most likely exhibit best transfer conditions during the time of data transmission.

4 MAV Network Testbed

The used MAV network testbed is set up by quadcopters with autonomous flight and hovering capabilities commercially available at a reasonable price. We optimize the flying platform as well as the used communication technologies for wireless transmission in the air.

4.1 Flying Platform

We use a platform called "Arducopter" [18], which possesses an Arduino-based autopilot with GPS, IMU, pressure sensors, etc. The copter's typical cruise speed is 4.5 m/s and it is able to fly safely at altitudes up to 100 m. The autopilot enables autonomous take-off and landing, and navigating through defined GPS waypoints. To set waypoints conveniently at the beginning of or during a mission, a graphical user interface is leveraged.

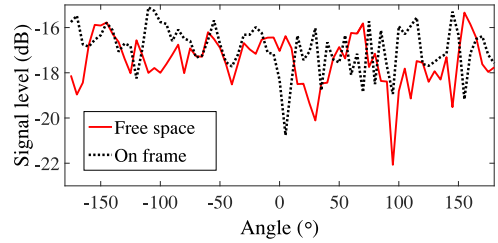


Fig. 6. Radiation pattern with bare antenna ("free space") and antenna mounted on the copter frame with plastic arms ("on frame") in an anechoic chamber.

The original metal-arm copter is modified by applying 3D-printed plastic arms and an on-board wireless package with two external circular antennas, see Fig. 5. We showed in [1] that the metal arms cause high signal blockage with losses between 15 and 20 dB. Fig. 6 shows the measured radiation pattern of antennas mounted on the MAV frame with plastic arms versus bare antennas; basically no signal loss by the copter's new frame is observed.

4.2 Hybrid Wireless Network

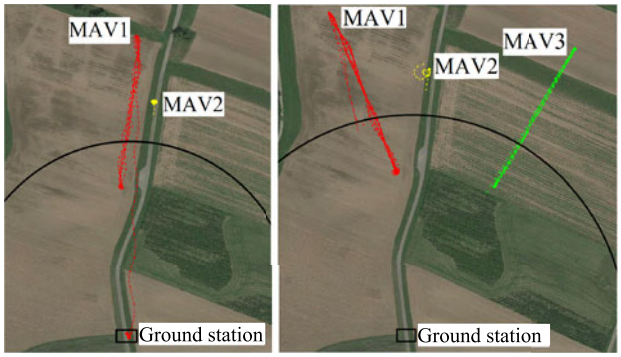
We make use of the following two radio technologies:

XBee-PRO. XBee-PRO (IEEE 802.15.4) provides a long-range (up to 1.5 km), low-throughput (less than 80 kbit/s, shared among all MAVs) communication channel reserved for light-weight data such as control commands, telemetry data, and acknowledgment of data reception. XBee-PRO serves as the out-of-band channel in our network architecture. This technology operates in 2.4 GHz frequency band and connects every quadcopter to the ground station. We operate XBee-PRO as a broadcast channel. Telemetry data including GPS (latitude, longitude, altitude), orientation, and speed are broadcasted periodically with a tunable period (here, about 50 bytes are sent by each MAV per second).

Wi-Fi. Wi-Fi (IEEE 802.11n) is a shorter range, high-throughput communication technology that is well suited for transfer of large-sized data. The performance characteristics depend on the concrete hardware set-up, yet, MAVs of similar type are typically exposed to similar weight and embedded system limitations. In our copter testbed, the communication range is up to $200-300$ m and the UDP throughput is up to $80-100$ Mbit/s (cf. Fig. 4). To avoid interference with XBee-PRO, Wi-Fi is configured in 5 GHz frequency band. The Wi-Fi network connects the copters to one another in ad-hoc mode. We select SparkLAN WUBR-507N USB dongles with Ralink 3572 chipset due to its flexibility (both 2.4 and 5 GHz bands are supported) and acceptable performance.

4.3 Data Traffic

For rigorous analysis of packet forwarding over the Wi-Fi network, MAVs create specially structured application-layer messages that enable in-depth performance inspection. The message header includes a *sequence number* that identifies the respective message and an *algorithm ID* to differentiate between the different forwarding algorithms. The *TTL (time to live)* field defines the maximum number of hops allowed in



(a) Scenario S1 (b) Scenario S2

Fig. 7. Scenarios used in field experiments: GPS positions and trajectories of (a) two and (b) three quadcopters and one ground station. Arcs indicate the cut-off range $\mathcal{D} = 200$ m of Wi-Fi communication.

order to prevent infinite message looping. Message *creation time* is used to calculate the message delay (all nodes are synchronized via NTP). Finally, the *path* field is used to store the sequence of node IDs traversed by the message. Each message has an overall message size (including header and payload) of about $1,400$ bytes (i.e., less than the maximum transmission unit to avoid fragmentation). By configuring the frequency of message creation/sending, we generate the desired traffic load and can mimic any application's data traffic. All messages are transmitted using UDP.

5 Field Experiment Results

Leveraging the introduced quadcopter testbed, we evaluate the proposed packet forwarding algorithms in real world. We use up to three quadcopters and one ground station; all copters generate data destined to the ground station.

5.1 Scenarios

The basic set-up is inspired by a search and rescue mission. One stationary ground station is placed together with one hovering copter at a distance where no direct link between ground station and copter is provided. Two scenarios of intermittent connectivity are defined as depicted in Fig. 7:

- *Scenario S1*: One ferry (MAV_1) moves in and out of communication range of one hovering MAV (MAV_2) and the ground station and establishes connectivity by carrying data (own and data from MAV_2) to the ground station.
- *Scenario S2*: Another copter (MAV_3) is added to the setting of scenario S1. This copter also moves back and forth between the ground station and MAV_2, but in opposite direction of MAV_1. Different to scenario S1, now multiple path options exist to reach the ground. In particular the hovering MAV_2 may now choose between forwarding data to MAV_1 or to MAV_3.

We select these scenarios to expose the forwarding algorithms to different path options even in a setting with only few nodes. A scenario starts when all quadcopters have arrived at their first waypoint.

All quadcopters periodically log necessary parameters for post-flight analysis including the Wi-Fi topology and the current status of various on-board sensors including GPS position, speed, and orientation. Due to safety requirements each copter must always remain in visibility range. Limited by this constraint and further to avoid wide-distance links with low quality, we limit the communication range of the wireless links to $\mathcal{D} = 200$ m (even though the external antennas would provide larger ranges [1], yet at lower quality). The safe flight altitude used is 20 m. Each test is executed for about 8 min (the maximum flight time of the quadcopter is 10 min).

TABLE 2
Test Settings in Field Test and Simulation

	Field test	Simulation
Scenario		
Test area	$400\ \mathrm{m} \times 400\ \mathrm{m}$	$800\ \mathrm{m} \times 800\ \mathrm{m}$
Number of nodes/ferries	up to 4/1–2	up to 14/4
Number of scenarios	2	10
Experiment time	~ 8 min	~ 8 min
Mobility		
MAV speed	4.5 m/s	4.5 m/s
Mobility (betw. waypoints)	Real	Trace driven
Communication		
Wi-Fi IEEE 802.11n	Full featured	Limited, no MIMO
Wi-Fi cut-off range	200 m	200 m
Out-of-band channel	XBee-PRO	Shared memory
Message creation rate	25/s	5 (10 and 20)/s

Each MAV generates 25 messages per second (280 kbit/s) yielding a total load of 560 kbit/s (scenario 1) and 840 kbit/s (scenario 2) that is destined to the ground station. To compare the different packet forwarding algorithms under similar test conditions, we send multiple copies of the same message at each point in time, one for each forwarding algorithm instead of repeated experiments which would be prone to altered measurement conditions (due to changing GPS accuracy, wind, automatic flight behavior, etc.). Table 2 summarizes the settings.

5.2 Metrics

The following metrics are used to evaluate the performance of the packet forwarding algorithms:

- *Delivery ratio*: The delivery ratio is defined as the fraction of messages that have been successfully delivered to the destination out of the messages that have been generated. This metric is a measure of the reliability of the forwarding algorithm.
- *Delay*: The delay is calculated for each message successfully received at the destination. It is the sum of the communication delay (C_{delay}, Equation (1)) occurred on each hop a message traverses to reach the destination. C_{delay} includes carry time as well as transmission time.
- *Hop count*: The hop count is the number of hops a message passes until it reaches the destination. This metric allows to discuss the efficiency of a forwarding algorithm.

5.3 Results of $\mathrm{DTN}_{\mathrm{geo}}$ Forwarding

$\mathrm{DTN}_{\mathrm{geo}}$ is studied first to provide a baseline for further investigations. Table 3 summarizes the measured delivery ratio and delay per MAV and in total, in both test scenarios.

TABLE 3
Field Experiment: Delivery Ratio and Delay of DTN_{geo}, DTN_{close}, and DTN_{load}, for Each MAV Under Test, Scenarios S1 and S2

	Delivery ratio (%)			Delay (s): mean/median/std.		
	DTN_{geo}	DTN_{close}	DTN_{load}	DTN_{geo}	DTN_{close}	DTN_{load}
Scenario S1						
MAV_1 (ferrying)	98.60	100	99.60	23.77/22.67/19.62	22.23/20.30/18.52	21.42/19.72/17.40
MAV_2 (hovering)	99.60	100	99.60	22.61/20.33/19.63	21.20/18.05/18.49	20.26/17.43/17.39
Total average	**99.10**	**100**	**99.60**	23.19/**21.50**/19.62	21.71/**19.17**/18.51	20.84/**18.27**/17.40
Scenario S2						
MAV_1 (ferrying)	96.01	97.48	99.02	1.41/0.11/2.86	1.10/0.08/2.76	1.19/0.11/2.44
MAV_2 (hovering)	96.12	95.31	99.45	1.58/0.13/3.0	1.03/0.08/2.63	1.11/0.12/2.43
MAV_3 (ferrying)	95.76	97.30	100	1.38/0.07/2.85	0.77/0.07/1.98	1.03/0.07/2.42
Total average	**95.96**	**96.69**	**99.49**	1.46/**0.10**/2.90	0.97/**0.08**/2.46	1.11/**0.01**/2.43

Delivery ratio. In both scenarios, the delivery ratio is above 95 percent. These results demonstrate the very reliable message forwarding behavior of DTN_{geo}. The delivery ratio is generally lower in scenario S2 compared to scenario S1. Possible causes are the higher load in scenario S2 and the higher dynamics generated by three MAVs compared to the setup with two MAVs, leading to more 802.11n link losses.

Delay. The total median delay observed is 21.50 s in scenario S1. This large delay is due to the disconnection time period of about 60 s of the ferry MAV in each round. Adding an additional ferry (scenario S2) decreases the median delay to 0.1 s. Fig. 8 shows the delay of a sample sequence of messages of ferry MAV_1 (scenario S2). The peak delays correspond to the points in time when MAV_1 is temporary disconnected from the ground station. In this situation MAV_1's messages are queued and transmitted later, in a slower store-carry-forward manner. We further observe a large spreading of delay, which is due to the situation that some messages are carried away from the destination ground station before transmitting while others are transmitted immediately.

Hop count. The number of hops traveled by each message is generally low due to the small network size. The average hop count of all messages is 2.18 in scenario S1, and 1.79 in scenario S2. The higher fraction of direct link options to the ground station is the cause for the lower hop count in scenario S2. The hop count distribution of both scenarios for all MAVs is shown in Fig. 9. We remark that the hop count varies depending on the MAV's task, which defines its waypoints. The messages of ferry MAV_1 require at least one hop in scenarios S1 and S2 (similarly, ferry MAV_3 in scenario S2 can send messages to the ground in one hop), whereas the minimum hop count is 2 for the hovering MAV_2.

We further observe that transmission is not always efficient. In scenario S1, Fig. 9a, still 59 percent of MAV_1's messages require three hops to reach the ground station, which means that they are sent from MAV_1 to MAV_2 and back to MAV_1 before being transferred to the ground station. Similarly, the fraction of messages of MAV_2 that take four hops in scenario S1 (9 percent) and three hops in scenario S2 (8 percent) are inefficiently sent back and forth. In the following we will show that the proposed heuristics provide effective countermeasures against this ping-pong effect.

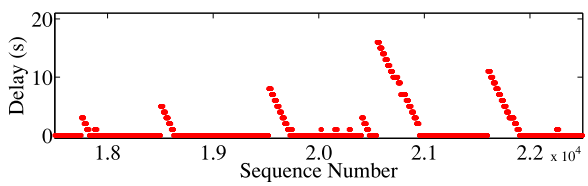


Fig. 8. Sample message sequence generated by MAV_1 in scenario S2: communication delay per message sequence number using DTN_{geo}.

5.4 Results of Heuristics

We now study the effects of each predictive heuristic, DTN_{close} and DTN_{load}, in isolation. We discuss the results in relation to the results of DTN_{geo}. Again, scenarios S1 and S2 are used, and Table 3 details the measurements. The prediction time frame is set to $\mathcal{F} = 4$ s.

Delivery ratio. The average delivery ratio of DTN_{geo} is already very high. Both heuristics achieve a similar, slightly improved average delivery ratio in both scenarios (cf. Table 3). The largest improvement is achieved by DTN_{load} in scenario S2, where the total average delivery ratio is increased by 3.68 percent. A reason for this improvement is the specific property of DTN_{load} to select a neighbor with the best communication condition to deliver (most of) the message load successfully.

Delay. In terms of total median delay, we observe that DTN_{close} and DTN_{load} can slightly outperform DTN_{geo} by

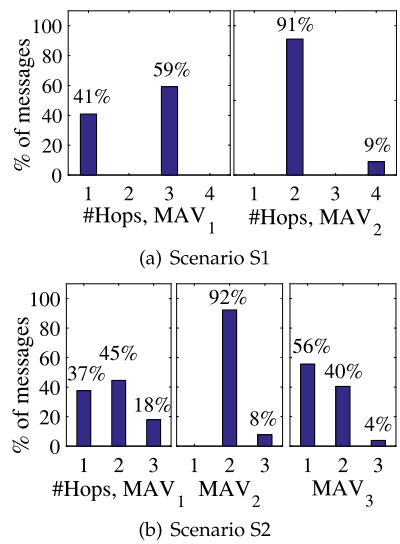


Fig. 9. Field experiment: hop count distribution of DTN_{geo}.

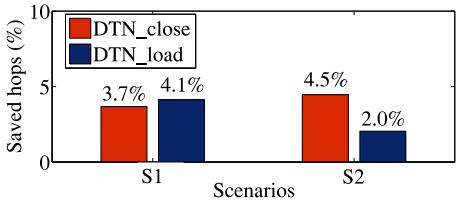


Fig. 10. Hop count improvement in terms of percentage of saved hops for DTN_{close} and DTN_{load} algorithms compared to DTN_{geo}.

respectively 1.8 and 3.23 s in scenario S1. As the delay is already very low in scenario S2, the respective improvements are low as well: respectively 20 ms for DTN_{close} and 90 ms for DTN_{load}.

In particular in scenario S1, only MAV_1 ferries data back to the ground station, therefore the messages of all algorithms have to pass through the same ferry (path) irrespective of inefficient forwarding loops and thus the delay can in principle not be significantly improved. This changes for scenarios where more path options exist (cf. Section 6).

Hop count. Fig. 10 visualizes the fraction of hops saved by each heuristic in comparison with DTN_{geo}. The fraction is calculated as the total number of hops saved per scenario over the total number of hops of DTN_{geo}[2]. Though the fraction might not seem impressive, for instance, an improvement of 4.5 percent in scenario S2 corresponds to a saving of 2,250 hops (single hop transmissions) compared to DTN_{geo}. We observe that both heuristics can save a substantial number of non-necessary transmissions, thus, reducing the network load and mitigating possible message loss due to overload. DTN_{load} achieves this result by forwarding data to a node closer to the destination with predicted ability to master the load best in the estimated connection time, whereas DTN_{close} decreases the hop count as the prediction of future locations alleviates ping-ponging of messages.

6 Simulation Results

We now expose the forwarding schemes to a larger number of nodes (up to 13 MAVs) that is not easily feasible in field experiments and use the same evaluation metrics as introduced for the field test. We make use of the state-of-the-art network simulator ns-3 [19]. A benefit of selecting a common, open network simulator is that we can repeat scenarios rapidly and make our results comparable with other works. However, ns-3 lacks proper MAV mobility and aerial communications models, and does so far not support full-featured IEEE 802.11n (no support for MIMO, for instance). In order to compensate for these limitations, we provide a simulation mobility model and a communication model.

The *simulation mobility model* is derived from real in-flight traces of quadcopters as observed by GPS and IMU sensors. First, traces of a straight flight are selected. Then, by applying basic geometric translation and rotation operations, we are able to create the movement between mission-given waypoints (including turns). Additionally, we implement hovering at locations. The *communication model* is also based on real world observations. The throughput between two

2. In total, DTN_{geo} results in 40×10^3 hops in scenario S1, and 50×10^3 hops in scenario S2.

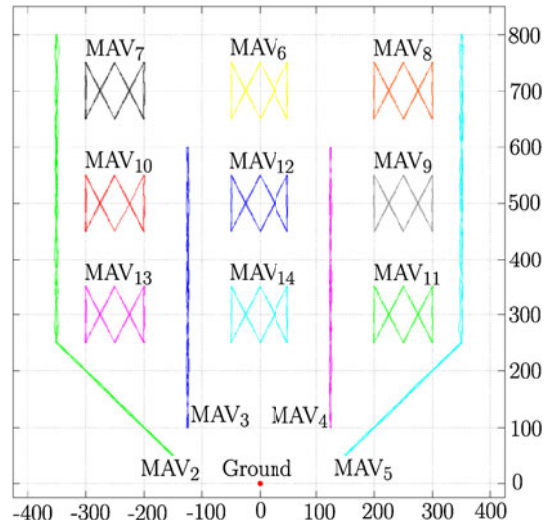


Fig. 11. Simulation scenario with a ground station, trajectories of four ferry MAVs (MAV_{2-5}) and nine searching MAVs (MAV_{6-14}). The fuzziness of the MAV trajectories is a result of the simulation mobility model based on real MAV trajectories.

MAVs is modeled as a function of distance derived from measurements, cf. Section 2.5. This way, we assure that simulation meets reality although simulation not fully implements all communication features.

6.1 Simulation Setup

DTN_{geo}, the two heuristics DTN_{close} and DTN_{load}, and epidemic routing [20] are implemented in simulation. We use a shared memory to emulate the out-of-band channel (XBee-PRO in field tests) for sharing MAVs' position and direction information. Nodes connect to each other via ad-hoc Wi-Fi 802.11n at 5 GHz. Similar to the field test, the Wi-Fi communication range is set to 200 m. On the physical layer, the transmission gain is set to 20 dB, which provides the required throughput for the intended Wi-Fi range. Each MAV generates 5 messages per second (56 kbit/s, overall 728 kbit/s are generated in the largest scenario with 13 MAVs) addressed to the ground station. Note that the reduction from 25 messages generated per second in the field test, cf. Section 5, to 5 messages is due to creating a comparable total load in a larger fleet and to speed up simulation (cf. the discussion on higher loads in Section 6.5). Messages are sent using UDP sockets with a TTL set to 20 hops. In compliance with the field tests, the experiment time of each simulation test run is about 8 min. A summary of parameters used in simulation is given in Table 2.

6.2 Scenarios

The simulation scenarios are – as in the field test – inspired by search and rescue missions (cf. Fig. 1). The simulation area is about 800 m×800 m. Fig. 11 visualizes the placement of all MAVs and their trajectories. The scenarios are constructed by using a basic setting of one ground station (node number 1) and four ferry MAVs (MAV_{2-5}), and up to nine additional searching MAVs (MAV_{6-14}). The searching MAVs are placed out of communication range of the ground station but within the communication range of at least one ferry MAV. Each searching MAV is assigned a 200 m×200 m region, defining the area the MAV should

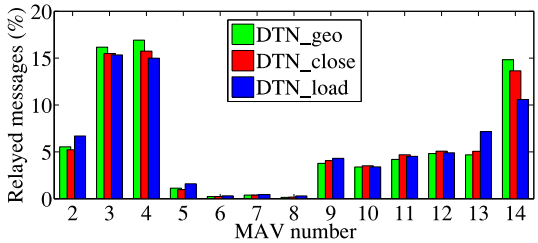


Fig. 12. Popularity of MAVs in the 14-node scenario: percentage of relayed messages by each MAV for all three geo-based algorithms.

survey and collect data from. As depicted in the figure, a typical search zigzag movement pattern is implemented to effectively cover a region. Furthermore, the symmetric ferries $\mathrm{MAV}_{2,5}$ and $\mathrm{MAV}_{3,4}$ always fly in opposite directions to generate more path options for forwarding decisions.

To investigate effects of different MAV densities and placements, we introduce ten different scenarios with varying number of nodes n ($5 \leq n \leq 14$). To define the scenarios, we look at the number of messages that are traversed per MAV. Typically not all MAVs are equally popular with respect to forwarding messages. We define the *popularity of an MAV* as: $\text{Popularity} = \frac{\text{Number of messages relayed}}{\text{Total number of messages}}$.

To construct a scenario $n-1$, we remove the respective most popular searching MAV in scenario n in order to change a significant aspect of scenario n. We start with the largest scenario $n = 14$; the popularity of all MAVs in Scenario $n = 14$ is visualized in Fig. 12. Most popular MAVs are the ferries MAV_{2-4}, and central MAVs close to the ground station (MAV_{11-14}), with MAV_{14} being the most popular searching MAV. Scenario $n = 13$ is now constructed by eliminating MAV_{14} from scenario $n = 14$. To easily identify the scenarios, we selected indexes of the searching MAVs to reflect the order of elimination. In other words, each scenario n consists of the nodes $\{\mathrm{MAV}_2 \cdots \mathrm{MAV}_n\}$; the placement of each MAV is depicted in Fig. 11.

6.3 Performance of $\mathrm{DTN}_{\mathrm{geo}}$

$\mathrm{DTN}_{\mathrm{geo}}$ combines end-to-end shortest path routing and DTN-based forwarding if needed. Fig. 13 shows how often packets are routed along the shortest end-to-end path or using DTN-based connections making use of ferries. We find that in our sparse node deployment, the overwhelming majority of messages is forwarded using DTN. At most, 12.9 percent of the messages can make use of an existing end-to-end path in all test scenarios.

The performance of basic $\mathrm{DTN}_{\mathrm{geo}}$ is evaluated in comparison with epidemic routing [20], a basic packet forwarding approach that makes use of message replication. Its spreading principle is central to many state-of-the-art

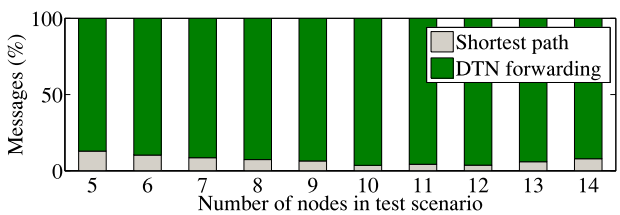


Fig. 13. $\mathrm{DTN}_{\mathrm{geo}}$ algorithm: Fraction of packets received at the ground station by DTN forwarding versus shortest path delivery.

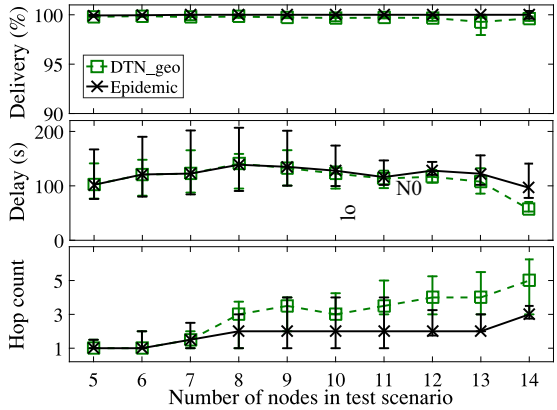


Fig. 14. Comparison of $\mathrm{DTN}_{\mathrm{geo}}$ with epidemic forwarding (median, 25 and 75 percent quartiles as error bars). The *median* values are calculated over all MAVs acting in one scenario.

and more sophisticated DTN algorithms (such as Spray and Wait [21]). Epidemic routing uses an approach analogous to the spreading of infectious diseases. "Infected" nodes forward a packet when another node is encountered that does not yet have a copy of the packet ("not yet infected"). The transmission is successful when the first packet copy is received at the destination. Epidemic routing explores exhaustively all routes in the network. In an ideal situation, the first received copy shows the minimum delay possible (likely at low hop count), and the delivery ratio of transmission is optimal as well. In reality, epidemic forwarding considerably consumes network and processing resources [22]. The generated load tends to cause congestion, losses, and delays. We implement epidemic routing in a configuration, that this scheme operates almost always loss-free with a delivery ratio close to 100 percent and thus, can serve as a benchmark (cf. Fig. 14 (top)).

Delivery ratio. As visualized in Fig. 14 (top), $\mathrm{DTN}_{\mathrm{geo}}$ achieves a very high delivery ratio varying between 99.2 and 100 percent. It is worth noting that $\mathrm{DTN}_{\mathrm{geo}}$ achieves a delivery result similar to epidemic routing, but with lower overhead, since $\mathrm{DTN}_{\mathrm{geo}}$ utilizes only a single message copy while epidemic forwarding relies on forwarding many copies of each message (as discussed later in detail).

Delay. Fig. 14 (middle) shows the delay of $\mathrm{DTN}_{\mathrm{geo}}$ and epidemic forwarding. Interestingly, in scenarios with 10 nodes or more, $\mathrm{DTN}_{\mathrm{geo}}$ outperforms epidemic routing and achieves a smaller delay. In particular in the 14-node scenario, the achieved delay is 39 s less. This is because even in a favorable configuration, epidemic spreading leads to large message queues and nodes are often not able to fully deplete the outbox queues during connection time. A significant portion of messages remains in the queues, which are delivered with larger delays in one of the next encounters. Moreover, as shown by the figure, the delay generally decreases for a better connected network, with larger number of nodes.

Hop count. The measured hop count is depicted in Fig. 14 (bottom). $\mathrm{DTN}_{\mathrm{geo}}$ messages have traversed up to 5 hops to reach the ground station. As expected, epidemic routing reaches the best hop count achievable in the scenarios by exploring all possible path options. Still, $\mathrm{DTN}_{\mathrm{geo}}$ achieves good results, in worst case the difference in hop count is 2.

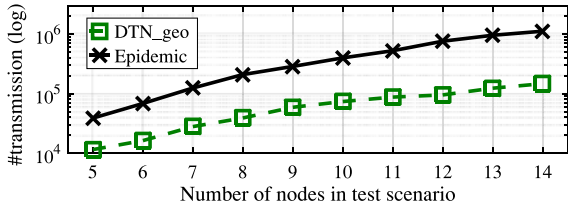


Fig. 15. Number of transmissions (in log scale) of DTN_{geo} and epidemic forwarding.

In principle, the hop count increases with increasing number of nodes, simply because many packets traverse more hops to reach the ground station from further distances.

Message overhead. Fig. 15 compares DTN_{geo} and epidemic routing in terms of the number of transmissions each algorithm generates. Compared to epidemic routing, DTN_{geo} needs between 27×10^3 and 970×10^3 less transmissions in the different scenarios.

To summarize, DTN_{geo} achieves convincing performance results, yet there is room for improvement concerning the forwarding efficiency (hop count).

6.4 Improvements with Heuristics

We perform the same simulation scenarios as before and compare the results of the heuristics with the results of DTN_{geo}. A time frame of $\mathcal{F} = 4$ s is used for future prediction. Table 4 summarizes the improvement of delay, delivery ratio, and hop count. The median values are calculated over all MAVs acting in one scenario. Improvement is calculated as the difference between the median values.

Delivery ratio. DTN_{geo} with and without the heuristics achieves a very high delivery ratio of > 99 percent in all the scenarios. There is only small room for improvement as shown by average improvement of both DTN_{close} and DTN_{load}, in Table 4. In single test cases the heuristics perform slightly worse.

Delay. On the total average (see Table 4), the delay is improved by 1.04 s (DTN_{close}) and 1.87 s (DTN_{load}), in single cases up to 4.44 s (DTN_{close}) and 12.59 s (DTN_{load}) – with a degradation in single cases at most up to 1 s. We find that in some cases DTN_{geo} forwards along long paths. A major cause for the long routes is the disadvantageous ping-pong effect which can be alleviated by using prediction of future positions (DTN_{close}). DTN_{load} can improve delay by further anticipating how much data can be transferred over a link in the near future.

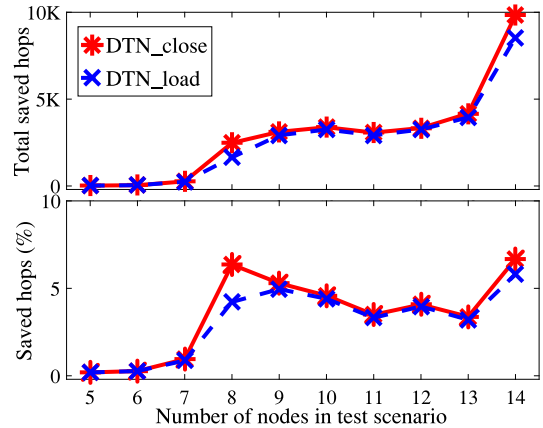


Fig. 16. Hop count improvement of DTN_{close} and DTN_{load} with respect to DTN_{geo}: (top) total number of saved hops and (bottom) the fraction of saved hops out of the total number of hops required by DTN_{geo}.

Hop count. The hop count results show that both DTN_{close} and DTN_{load} either improve the hop count of DTN_{geo} by up to 1 hop or do not impair it (median values, cf. Table 4). Fig. 16 (top) shows the total amount of saved hops compared to DTN_{geo}, and (bottom) presents the saved hops related to the number of hops needed for the respective message by DTN_{geo}. Up to 7 percent of the total number of hops can be saved in the scenarios. In the largest scenario (14 nodes), the saving amounts to 147×10^3 hops for DTN_{close}.

6.5 Discussion of Results

We have shown that DTN_{geo} provides a practical geographic packet forwarding scheme for intermittently connected networks. The performance evaluation in terms of delivery ratio, delay, and hop count reveals that DTN_{geo} achieves results close to epidemic routing, yet with substantially lower messaging overhead. The major improvement achieved by the heuristics DTN_{close} and DTN_{load} is the reduction of inefficient transmissions that can be detected when anticipating future positions. We now want to discuss factors impacting our approach.

Prediction time frame. The prediction time (in our study 4 s) should be adapted to the characteristics of the MAV's trajectory. As we employ a linear mobility model, the prediction time can be increased for trajectories with few turns without increasing the prediction error, yet, leading to improved packet forwarding results. For instance, we performed additional tests in a sample scenario with ferries

TABLE 4
Improvement of DTN_{geo} by the Two Heuristics in Terms of Delay, Delivery Ratio, and Hop Count (Difference in Median Values); the Symbols "+"/"–"/"=" Refer, Respectively, to Improvement/Degradation/Same Results as Achieved by the Heuristics in Comparison to DTN_{geo}

	Scenarios (n)	5	6	7	8	9	10	11	12	13	14	**Average**
DTN_{close}												
	Delivery ratio	=	=	– 0.04	=	+ 0.04	+ 0.05	=	– 0.04	+ 0.41	– 0.10	**+ 0.032**
	Delay (s)	=	=	+ 1.87	+ 3.35	– 0.64	+ 0.33	– 0.28	+ 2.64	+ 4.44	– 1.33	**+ 1.04**
	Hop count	=	=	=	+ 1.0	=	=	=	=	=	+ 1.0	**+ 0.2**
DTN_{load}												
	Delivery ratio	=	=	– 0.02	=	+ 0.04	+ 0.09	=	– 0.18	+ 0.34	– 0.14	**+ 0.013**
	Delay (s)	=	=	+ 1.87	+ 3.90	– 0.61	– 0.01	– 0.79	+ 2.76	+ 12.59	– 1.00	**+ 1.87**
	Hop count	=	=	=	=	=	=	+ 0.5	=	=	+ 1.0	**+ 0.15**

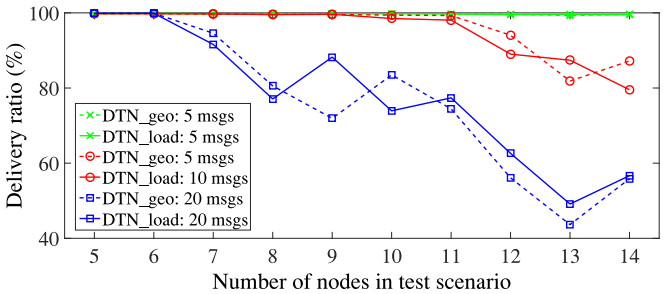


Fig. 17. Delivery ratio results of $\mathrm{DTN}_{\mathrm{geo}}$ and $\mathrm{DTN}_{\mathrm{load}}$ under different loads (median values of ~ 8 min simulation).

MAV_{2-5}, and searching MAVs MAV_{11}, MAV_{13}, and MAV_{14} (cf. Fig. 11) with different prediction times. Compared to a prediction time of 4 s, with a prediction time of 8 s the delay of both heuristics is improved by 0.54 s and the number of hops saved is increased by 1.36×10^3 ($\mathrm{DTN}_{\mathrm{close}}$) and by 0.57×10^3 ($\mathrm{DTN}_{\mathrm{load}}$); the delivery ratio remains high.

Location sensor noise. Real mobility traces come with instantaneous noise as a result of GPS inaccuracy (IMU inaccuracies are small), impacting the performance of location-aware algorithms. To prevent false forwarding decisions due to positioning inaccuracies in practice, we introduce a safety margin of $d_{min} = 3$ m minimum difference in distance to the destination that has to be exceeded before another node is considered to be closer to the destination than the sending node (value of d_{min} has been empirically derived and can be configured). With improved positioning technologies, it is expected that the safety margin d_{min} can be relaxed.

MAV placement. Our extensive simulations showed that the placement of the MAVs together with the trajectories of the ferry MAVs significantly influences the achievements of prediction-based heuristics. Anticipation of the future can be exploited best when multiple path options are available that comprise ferries with differing mobility vectors (some heading for the ground station, some moving away, etc.). In such settings, predictive approaches can avoid that messages are physically carried long ways with large delays or that messages are looping (ping-pong effect). In cases with limited path variety, pure $\mathrm{DTN}_{\mathrm{geo}}$ is sufficient.

Higher load. We now study our algorithm under different loads. The simulation setup is as before, consisting of 5 to 14 nodes (cf. Fig. 11). Three cases are compared: every MAV generates 5 messages per second (base scenario, 56 kbit/s per MAV), 10 messages per second (112 kbit/s per MAV), and 20 messages per second (224 kbit/s per MAV). During disconnection times, the generated load accumulates leading to extensive message transfer and temporary heavy load on the links as soon as they are established.

Fig. 17 summarizes the delivery ratio of $\mathrm{DTN}_{\mathrm{geo}}$ and further $\mathrm{DTN}_{\mathrm{load}}$ – the heuristic which prefers links that allow to transfer the most of the data in the queue. It can be observed that the delivery ratio of $\mathrm{DTN}_{\mathrm{geo}}$ decreases significantly with increasing load, from almost 100 percent (5 messages per second) to almost 44 percent (20 messages per second) in the worst performing scenario ($n = 13$). We further observe an increase of (median) delay by up to 53 s in some scenarios when the load is quadrupled (note that the measured delay is already about 100 s under base load). Table 5 shows the delay statistics of all scenarios.

TABLE 5
Measured Delay Under Different Loads (mean/median/std.)

	$\mathrm{DTN}_{\mathrm{geo}}$ (s)	$\mathrm{DTN}_{\mathrm{load}}$ (s)
5 messages	113.7/**118.6**/26.2	111.7/**117.4**/25.1
10 messages	126.2/**125.8**/31.4	120.2/**123.1**/25.3
20 messages	138.3/**132.4**/29.9	130.8/**132.2**/28.8

$\mathrm{DTN}_{\mathrm{load}}$ mitigates the impact of higher load and outperforms $\mathrm{DTN}_{\mathrm{geo}}$ in the majority of the test scenarios in terms of delivery ratio and delay. Yet, $\mathrm{DTN}_{\mathrm{load}}$ shows a similar trend of performance decrease as $\mathrm{DTN}_{\mathrm{geo}}$ when exposed to higher load.[3] We conclude that packet forwarding in MAV networks should implement additional countermeasures to mitigate loss caused by (accumulated) heavy loads. Options to consider are load balancing mechanisms and reliable one-hop transfer (up to future work).

7 Related Work

Packet forwarding in MAV networks relates to routing protocols in mobile ad-hoc networks (MANETs). Yet, the frequent topology changes of an MAV network, fast movement, and unstable wireless link conditions make the existing MANET routing algorithms impractical. In [7], OLSR is evaluated in a network of two micro airplanes and a ground station. The authors conclude that OLSR can not cope properly and quickly enough with the fast changing topology. In own previous work [8], we show that routing based on B.A.T.M.A.N. is impaired by the changes of the topology due to its long route convergence time.

A different, more promising paradigm is followed by delay-tolerant networks (DTNs). Traditional DTNs set up by human-carried devices have been exhaustively studied. One difference to MAV networks is the usually long inter-contact time that is not given in mission-oriented MAV networks. Another major difference – and advantage – of autonomous aerial robot networks is that MAV pathways are more predictable than human device trajectories. Multi-copy schemes of DTNs may be avoided and a single-copy protocol may be employed, which has also been pursued in (few) DTN networks [23]. When considering the transfer of large-sized data, traditional spreading with multi-copy protocols comes with a large overhead (and potential packet loss) [24]. In our work, we do not require but also not exclude multi-copy schemes. Although our algorithms use motion and load context to improve single-copy transmission, it is in principle possible to generate multiple copies on top of our forwarding scheme. Having investigated both MANET and DTN forwarding, we believe that the MAV network is a hybrid DTN/MANET and therefore both connected and disconnected cases should be supported. R3 supports this point of view [25]; R3 is a special case of epidemic routing [20], which leverages replication to improve delay. However, replication-based solutions lack scalability.

3. For reasons of completeness, we note that $\mathrm{DTN}_{\mathrm{close}}$ shows a similar trend both in delivery ratio and delay as the other schemes. Further, the hop count does not change under varying load.

Geographic routing is a promising approach for aerial networks [8]. For ground vehicles, navigation information and the store-carry-forward concept are leveraged by Geo-DTN+Nav [26]. Yet, the approach cannot utilize the characteristics of MAV networks such as the use of controlled mobility in free space (as provided by ferries). LAROD is a location-based algorithm design for aerial vehicles [10]. Similar to our algorithm, LAROD combines geographic routing with store-carry-forward. The achieved results confirm a sufficiently good delivery ratio comparable to epidemic routing at a substantially lower (still considerable) overhead. Performance results of geographic forwarding in sparse networks confirm that greedy geographic forwarding is in general only suitable for non-critical applications [11]. This result is further supported by later work of the authors that includes an estimation of the probability of successful communication between source and destination pairs in geographic forwarding, thus suggesting to combine greedy forwarding with additional mechanisms [27]. We follow this thought by combining the concepts of geographic routing and ferrying in our work. In particular for sparse, partitioned networks, ferries are a well-suited concept mitigating some of the effects described in this related performance study [4].

AeroRP [28] presents a geographic routing algorithm that uses velocity-based heuristics to cope with the very fast vehicles in aeronautical networks (around $1200\ \mathrm{m/s}$), where the considered speed is much higher than the speed of an MAV (up to $25\ \mathrm{m/s}$). AeroRP requires full trajectory knowledge and uses the *time to intercept*, i.e., the time two MAVs are in transmission range, as the primary metric for routing decisions. Using simulation, it is shown that variants of AeroRP outperform the traditional MANET routing algorithm OSLR and AODV. Our heuristic $\mathrm{DTN}_{\mathrm{load}}$ utilizes a similar concept, although it does not require full trajectory knowledge.

Motion information is also used in [29] for ad-hoc routing with full MAV trajectory knowledge. The algorithm requires large memory for maintaining the path information of all the MAVs and comes at high computational costs. Yet, simulation results show improved throughput in comparison to two MANET algorithms, AODV and LAR. To mitigate exhaustive search in the forwarding path space, we introduce an A*-based search algorithm in [30]. A major advantage of trajectory-aware routing is the optimization of the multi-hop path, however this comes with a disadvantage as the knowledge of the path of all MAVs during the whole mission is required. In contrast, the scheme introduced in this paper requires only current motion and location information from an MAV's neighboring nodes to make a forwarding decision for the next hop.

The mentioned related approaches and studies either evaluate the routing algorithms entirely in simulation or lack leveraging MAV system characteristics (ferries, out-of-band control channel). To the best of our knowledge, we are the first to practically investigate an anticipatory location-aware forwarding algorithm with DTN support using a fleet of real MAVs, supported by a realistic simulation, which has been customized for MAV networks by communication and mobility models derived from real observations.

8 Conclusion

We developed a concept for multi-hop micro aerial vehicle networks addressing the main challenges for packet forwarding from a practical perspective. The core of the networking solution is a motion-driven packet forwarding algorithm that applies delay-tolerant networking in case of disconnections. By taking advantage of location and motion sensors provided by MAVs and a realistic mobility model, near future MAV positions are predicted. Also, adding a realistic link throughput characterization allows for anticipating near future link capacity and connection time.

In an evaluation study comprising experiments in a real testbed and simulation, we demonstrated that the algorithm achieves a delay and delivery ratio comparable with ideal epidemic forwarding. With prediction, the algorithm further counteracts message ping-ponging of location-aware forwarding. Our results reveal that in particular in more complicated topologies of larger MAV fleets where multiple path options exist, anticipating future positions has an observable positive effect.

Acknowledgments

The authors thank Simon Egli, Mario Burger, and Fabian Schuiki for helping with field tests, and Bertold Van den Bergh for the tests in the anechoic chamber. This article has been partially supported by the SNSF under the contract of CRSI22_133059 (SWARMIX project), the Madrid Regional Government through the TIGRE5-CM program (S2013/ICE-2919), and by Ministerio de Economía y Competitividad grant TEC2014-55713-R.

References

[1] M. Asadpour, B. Van den Bergh, D. Giustiniano, K. Hummel, S. Pollin, and B. Plattner, "Micro aerial vehicle networks: An experimental analysis of challenges and opportunities," *IEEE Commun. Mag.*, vol. 52, no. 7, pp. 141–149, Jul. 2014.

[2] T. Andre et al., "Application-driven design of aerial communication networks," *IEEE Commun. Mag.*, vol. 52, no. 5, pp. 129–137, May 2014.

[3] M. Asadpour, D. Giustiniano, K. A. Hummel, S. Heimlicher, and S. Egli, "Now or later? - delaying data transfer in time-critical aerial communication," in *Proc. 9th ACM Int. Conf. Emerg. Netw. Exp. Technol.*, 2013, pp. 127–132.

[4] M. Bin Tariq, M. Ammar, and E. Zegura, "Message ferry route design for sparse ad hoc networks with mobile nodes," in *Proc. 7th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2006, pp. 37–48.

[5] M. Won, R. Stoleru, H. Chenji, and W. Zhang, "On optimal connectivity restoration in segmented sensor networks," in *Proc. Wireless Sens. Netw.*, 2013, pp. 131–148.

[6] N. Banerjee, M. D. Corner, and B. N. Levine, "An energy-efficient architecture for DTN throwboxes," in *Proc. 26th IEEE Int. Conf. Comput. Commun.*, 2007, pp. 776–784.

[7] A. Jimenez-Pacheco, D. Bouhired, Y. Gasser, J.-C. Zufferey, D. Floreano, and B. Rimoldi, "Implementation of a wireless mesh network of ultra light MAVs with dynamic routing," in *Proc. Globecom Workshops*, 2012, pp. 1591–1596.

[8] M. Asadpour, S. Egli, K. A. Hummel, and D. Giustiniano, "Routing in a fleet of micro aerial vehicles: First experimental insights," in *Proc. 3rd ACM MobiHoc Workshop Airborne Netw. Commun.*, 2014, pp. 9–10.

[9] B. Karp and H.-T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proc. 6th Annu. Int. Conf. Mobile Comput. Netw.*, 2000, pp. 243–254.

[10] E. Kuiper and S. Nadjm-Tehrani, "Geographical routing in intermittently connected ad hoc networks," in *Proc. 22nd Int. Conf. Adv. Inform. Netw. Appl. Workshops*, 2008, pp. 1690–1695.

[11] R. Shirani, M. St-Hilaire, T. Kunz, Y. Zhou, J. Li, and L. Lamont, "The performance of greedy geographic forwarding in unmanned aeronautical ad-hoc networks," in *Proc. 9th Annu. Commun. Netw. Serv. Res. Conf.*, 2011, pp. 161–166.
[12] B. Van den Bergh, T. Vermeulen, and S. Pollin, "Analysis of harmful interference to and from Aerial IEEE 802.11 systems," in *Proc. 1st Workshop Micro Aerial Vehicle Netw., Syst. Appl. Civilian Use*, 2015, pp. 15–19.
[13] M. Asadpour, D. Giustiniano, K. A. Hummel, and S. Egli, "UAV networks in rescue missions," in *Proc. 8th ACM Int. Workshop Wireless Netw. Testbeds, Exp. Eval. Characterization*, 2013, pp. 91–92.
[14] M. Asadpour, D. Giustiniano, and K. A. Hummel, "From ground to aerial communication: Dissecting WLAN 802.11n for the drones," in *Proc. 8th ACM Int. Workshop Wireless Netw. Testbeds, Exp. Eval. Characterization*, 2013, pp. 25–32.
[15] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Commun. Mobile Comput.*, vol. 2, no. 5, pp. 483–502, Aug. 2002.
[16] M. Y. S. Uddin, D. M. Nicol, T. F. Abdelzaher, and R. H. Kravets, "A post-disaster mobility model for delay tolerant networking," in *Proc. Winter Simul. Conf.*, 2009, pp. 2785–2796.
[17] S. M. George, W. Zhou, H. Chenji, M. Won, Y. O. Lee, A. Pazarloglou, R. Stoleru, and P. Barooah, "DistressNet: A wireless ad hoc and sensor network architecture for situation management in disaster response," *IEEE Commun. Mag.*, vol. 48, no. 3, pp. 128–136, Mar. 2010.
[18] "Arducopter: Arduino-based autopilot," (2013). [Online]. Available: http://copter.ardupilot.com/, [Accessed 14-July-2015]
[19] "NS-3 Network Simulator," (2014). [Online]. Available: http://www.nsnam.org, [Accessed 14-July-2015]
[20] A. Vahdat and D. Becker, "Epidemic routing for partially-connected ad hoc networks," Duke Univ., Durham, NC, USA, Tech. Rep. CS-200006, Apr. 2000.
[21] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: An efficient routing scheme for intermittently connected mobile networks," in *Proc. ACM SIGCOMM Workshop Delay-Tolerant Netw.*, 2005, pp. 252–259.
[22] X. Zhang, G. Neglia, J. Kurose, and D. Towsley, "Performance modeling of epidemic routing," *Comput. Netw.*, vol. 51, no. 10, pp. 2867–2891, 2007.
[23] M. J. Khabbaz, C. M. Assi, and W. F. Fawaz, "Disruption-tolerant networking: A comprehensive survey on recent developments and persisting challenges," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 2, pp. 607–640, Apr.–Jun. 2012.
[24] A. Balasubramanian, B. N. Levine, and A. Venkataramani, "Replication routing in DTNs: A resource allocation approach," *IEEE/ACM Trans. Netw.*, vol. 18, no. 2, pp. 596–609, Apr. 2010.
[25] X. Tie, A. Venkataramani, and A. Balasubramanian, "R3: Robust replication routing in wireless networks with diverse connectivity characteristics," in *Proc. 17th Annu. Int. Conf. Mobile Comput. Netw.*, 2011, pp. 181–192.
[26] P.-C. Cheng, K. C. Lee, M. Gerla, and J. Haerri, "GeoDTN+Nav: Geographic DTN routing with navigator prediction for urban vehicular environments," *Mobile Netw. Appl.*, vol. 15, no. 1, pp. 61–82, 2010.
[27] R. Shirani, M. St-Hilaire, T. Kunz, Y. Zhou, J. Li, and L. Lamont, "Quadratic estimation of success probability of greedy geographic forwarding in unmanned aeronautical ad-hoc networks," in *Proc. 75th IEEE Veh. Technol. Conf.*, 2012, pp. 1–5.
[28] K. Peters, A. Jabbar, E. K. Cetinkaya, and J. P. Sterbenz, "A geographical routing protocol for highly-dynamic aeronautical networks," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2011, pp. 492–497.
[29] R. Muzaffar and E. Yanmaz, "Trajectory-aware ad hoc routing protocol for micro aerial vehicle networks," in *Proc. Eur. Conf. Netw. Commun.*, 2014, pp. 1–7.
[30] M. Asadpour, M. Burger, F. Schuiki, and K. A. Hummel, "Needle in a haystack: Limiting the search space in mission-aware packet forwarding for drones," in *Proc. 1st MobiCom Workshop Experiences Des. Implementation Smart Objects*, 2015, pp. 31–36.

Mahdi Asadpour received the BSc degree in computer software engineering from the Sharif University of Technology, and the MSc and PhD degrees in computer science, and information technology and electrical engineering from ETH Zurich, respectively. His current research areas include wireless communications for MAV networks.

Karin Anna Hummel received the PhD degree in computer science from the Vienna University of Technology with honors in 2005. She is an assistant professor at JKU Linz. She is the author of more than 80 papers. Her research interests include opportunistic networking, mobilty characterization, and wireless communication.

Domenico Giustiniano received the PhD degree in telecommunication engineering from the University of Rome Tor Vergata in 2008. He is a research associate professor at IMDEA Networks Institute. He is the author of more than 60 papers and inventor of five patents. He research interests include emerging areas in the field of pervasive wireless systems.

Stefan Draskovic received the BSc degree from the University of Novi Sad, Serbia, and the MSc degree in electrical engineering from ETH Zurich. He is currently working toward the PhD degree in the Computer Engineering Group of ETH Zurich. His research interests cover multiprocessor system design and performance analysis.