

UAVNet: A Mobile Wireless Mesh Network Using Unmanned Aerial Vehicles

Simon Morgenthaler, Torsten Braun, Zhongliang Zhao, Thomas Staub, Markus Anwander
Communication and Distributed Systems, Institute of Computer Science and Applied Mathematics
Universität Bern, Switzerland

s.morgenthaler@students.unibe.ch, braun@iam.unibe.ch, zhao@iam.unibe.ch, staub@iam.unibe.ch, anwander@iam.unibe.ch

Abstract— We developed UAVNet, a framework for the autonomous deployment of a flying Wireless Mesh Network using small quadcopter-based Unmanned Aerial Vehicles (UAVs). The flying wireless mesh nodes are automatically interconnected to each other and building an IEEE 802.11s wireless mesh network. The implemented UAVNet prototype is able to autonomously interconnect two end systems by setting up an airborne relay, consisting of one or several flying wireless mesh nodes. The developed software includes basic functionality to control the UAVs and to setup, deploy, manage, and monitor a wireless mesh network. Our evaluations have shown that UAVNet can significantly improve network performance.

Index Terms— Unmanned Aerial Vehicles, Wireless Mesh Networks, Wireless Network Relaying.

I. INTRODUCTION

Wireless Mesh Networks (WMNs) provide an efficient way to deploy large communication networks and to interconnect separated heterogeneous networks. Depending on network scenario and application area, it can be difficult to deploy and maintain the participating wireless mesh nodes. Additionally, it may be crucial to set up a WMN quickly, e.g., in emergency and disaster recovery scenarios. The deployment and maintenance of wireless mesh network equipment may be difficult, inefficient and time consuming or even impossible if the affected area is inaccessible. This paper introduces and evaluates the UAVNet framework, a highly adaptive and mobile WMN using small Unmanned Aerial Vehicles (UAVs). It includes a concept and a prototype implementation of an autonomously and temporarily deployable WMN, using UAVs with attached wireless mesh nodes. The deployed communication network enables the connectivity between different end systems like notebooks, smartphones and tablets and even other wireless or wired networks.

Traditionally, UAVs have been used by the military for surveillance and reconnaissance operations. However, with the advent of robust wireless networking technologies, UAVs equipped with wireless transceivers can thus be enabled to communicate with ground nodes as well as other UAVs. Several research works have been done in this area. However, only a few of them investigated the problem of using multiple UAVs working in an autonomous way as communication infrastructure. Ad-hoc UAV-Ground (AUGNet) [1] is a wireless mobile ad-hoc network with radio nodes mounted at

fixed sites, on ground vehicles, and in small UAVs. It envisioned two different scenarios, where UAVs act as prominent radio nodes, which interconnect disconnected ground stations. In [2] the authors propose an idea of using UAVs as mobile backbone nodes to interconnect backbone nodes on the ground. The UAVs can broadcast link state information efficiently for improving routing performance. The research project AirShield [3] proposes to use autonomous flying robots for reconnaissance purposes and aiding in forecasting and prevention of emergency situations. The mobile UAV equipped with lightweight sensors is deployed to collect relevant data/information from the incident scene. AVIGLE [4] is an EU project, which focuses on the usage of UAVs in the areas of civil security as well as cellular networks, surveying, entertainment and media. In [5], the authors investigate communication during flight and argue that UAVs and ground stations should use omni-directional dipole antennas to achieve the highest throughput under a typical flyover UAV flight path. The work presented in [6] describes field tests with UAVs equipped with WiFi equipment. Those UAVs are small airplanes, which are more powerful than the UAVs used in UAVNet. In the SMAVNET project [7], small fixed-wing flying robots are deployed to rapidly create communication networks for rescuers in disaster scenarios.

In the work presented in this paper, we designed, implemented and evaluated a prototype of networked UAVs. The proposed UAVNet includes a framework focusing on the autonomous deployment of a flying WMN, using small quadcopter UAVs. Each UAV carries a lightweight wireless mesh node, which is directly connected to the flight electronics of the UAV using a serial interface. The flying wireless mesh nodes are autonomously interconnected to each other and are building an IEEE 802.11s WMN. Each wireless mesh node acts as access point and provides network access for regular IEEE 802.11g wireless devices. This work provides a platform for testing sophisticated algorithms for topology control and routing protocols of a UAV network.

The paper is structured as follows. Section II describes the hardware and software components used for developing the UAVNet prototype. Section III presents the software architecture of UAVNet. Section IV discusses experiments and evaluation results of UAVNet. Section I concludes the paper.

II. HARDWARE AND SOFTWARE COMPONENTS

This section describes the hardware and software components used to implement UAVNet. Whenever possible, we used standard components and protocols.

A. Quadcopter

The major components of a flying WMN are the UAVs carrying the wireless mesh nodes. As the flying nodes should stay rather stable in the air, quadcopters are used in this work. Due to their moderate costs, good availability, open software, and community support, quadcopters from the mikrokooper.de community project [8] have been selected.

A standard quadcopter consists of four main components, namely the frame carrying the flight electronics and motors, flight electronics with different components and sensors, four brushless motors controlled by four brushless controllers, and a Lithium-ion Polymer battery. The brushless motors drive the fixed propellers. Flight speed and direction are controlled only by manipulating the speed of the four individual rotors but not by adapting the pitch of their rotor blades. The flight electronics of the quadcopter consists of two modules, namely a flight controller, which controls the speed of the four brushless motors, and a navigation controller, which provides data from a GPS receiver and a 3-dimensional electronic compass. Equipped with all these sensors, the UAV can permanently calculate its exact location and position in the air. Moreover, the UAV can autonomously hold its position or even fly on a predefined route using several GPS waypoints.

The connection between the UAV electronics and the wireless mesh node has been implemented by using a serial connection between the UART interface on the wireless mesh node and the debug port on the navigation controller. As the serial interface on the wireless mesh node works with 3.3V and the navigation controller operates on 5V, a logic level converter is used to establish the connection [9].

B. Wireless Mesh Node

The wireless mesh nodes communicate with each other via an IEEE 802.11s WMN. They are also working as an IEEE 802.11g wireless access point (AP) to connect to end systems like notebooks, smartphones, tablets, which also can be used to control and manage the operation of UAVs. The wireless mesh node used for the UAVNet implementation is the “Professional Mesh OM1P” from Open-Mesh [10]. It contains an IEEE 802.11b/g wireless interface and an internal Universal Asynchronous Receiver/Transmitter (UART) serial port.

Figure 1 shows the quadcopter and wireless mesh node used in the UAVNet. The operating system running on the wireless mesh node is ADAM [11], an embedded Linux distribution developed at University of Bern. ADAM provides a platform to setup, manage, configure and monitor wireless mesh networks of different size. It contains a decentralized distribution mechanism for software updates and remote network configurations. ADAM also provides an intuitive, adaptable and simple build system for embedded Linux systems. It

supports a variety of embedded platforms and architectures, such as the WRAP, Alix [12] and Meraki platforms by using different build profiles.



Fig. 1. UAVNet quadcopter & wireless mesh node

III. UAVNET ARCHITECTURE

A. UAV Network Scenarios

The central communication components are the wireless mesh nodes carried by UAVs. The devices on the ground are divided into two groups playing different roles in UAVNet. The first group includes the end systems intending to communicate with each other. The second group contains monitoring and configuration devices, in our case an iPhone or iPad.

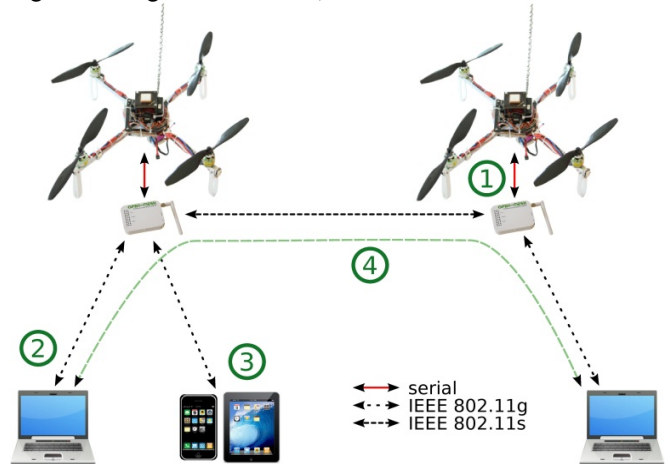


Fig. 2. UAVNet Scenario

Figure 2 shows a typical UAVNet scenario. Two notebooks use two flying UAVs to communicate with each other. The dashed arrows indicate wireless links and the solid arrows represent serial links.

1. Wireless mesh nodes are directly connected to the flight electronics of the UAVs using a serial link, which interconnects the serial interfaces of the wireless mesh node and the navigation controller at the UAV.
2. End systems are connected to the WMN via IEEE 802.11g. The wireless mesh nodes act as mesh access points (MAPs).
3. Configuration devices such as smart phones or tablet computers are used to configure and monitor the UAV network. They are connected using IEEE 802.11g.
4. Traffic between the end systems is relayed via an IEEE 802.11s wireless mesh network established by the UAVs.

B. Software

To keep UAVNet compatible, maintainable, and simple, we decided not to alter any software running on the flight electronics. All our custom software runs directly on the wireless mesh node and communicates with the unaltered software on the flight electronics using the provided serial communication protocol. The system software running on the wireless mesh nodes is a Linux 2.6.37.6 kernel [13] generated by ADAM. The driver from the ath5k project [14] is used for the WLAN chip. For the development of UAVNet, we updated the Linux kernel in the ADAM Linux distribution and extended it with the `compat-wireless` package [15].

Figure 3 shows the software architecture and communication interfaces of UAVNet. The `uavcontroller` written in C is the main component of UAVNet and runs on every UAV. It handles all the communication of the UAVs' wireless mesh nodes with other wireless mesh nodes, the end systems, configuration devices, and the flight electronics. It controls and monitors the different parameters of the UAV and sends appropriate commands and information to the flight electronics. The `uavcontroller` runs as a Linux daemon on the wireless mesh node. It uses two libraries `libuavext` and `libuavint`. The `uavclient` software is installed at monitoring and configuration devices as well as end systems (clients) on the ground and is mainly used for communication of clients with the wireless mesh nodes. It also uses the `libuavext` library.

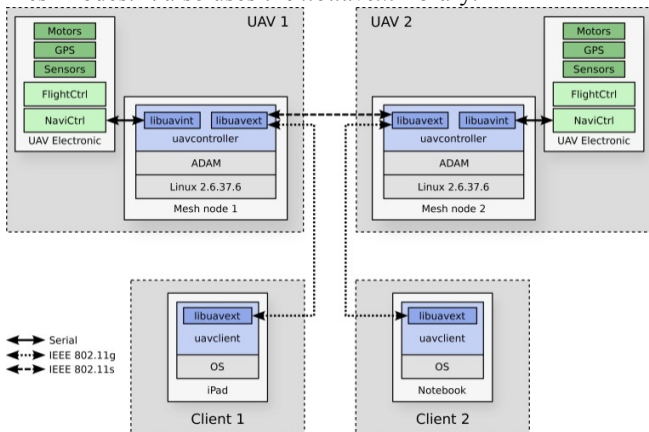


Fig. 3. Architecture and Communication Interfaces of UAVNet

The library `libuavext` in the `uavcontroller` handles the external communication of a UAV using TCP and UDP sockets over IEEE 802.11g and IEEE 802.11s wireless networks. It manages both, the outgoing and incoming traffic. There are two purposes for the communication between the wireless mesh nodes.

1. User data traffic is forwarded between end systems on the ground.
2. The wireless mesh nodes talk to each other and exchange information from the UAVs such as their positions, their flight directions and speeds, or the current status. The `uavcontroller` on one wireless mesh node constantly receives data from its own UAV's flight electronics and from the other UAV's wireless mesh nodes. It uses all

this information to act appropriately and send new data and instructions to the electronics of the UAV or to the other wireless mesh nodes.

The library `libuavext` is also included in the `uavclient` software installed at monitoring and configuration devices as well as end systems (clients) on the ground. `Uavclient` communicates with the wireless mesh nodes at the UAVs using IEEE 802.11g, because most of the clients do not implement the IEEE 802.11s protocol yet.

In a nutshell, `libuavext` is responsible for the following tasks:

- Establishing, listening to, managing, handling and closing all different kinds of sockets on different ports to enable the communication between the different devices.
- Handling and translating host names, IP and MAC addresses, using standard operating system procedures like static host name - IP mappings, Domain Name System (DNS), and Address Resolution Protocol (ARP).
- Sending and receiving all different kinds of TCP and UDP packets on the different sockets including retransmissions.
- Handling, decoding, and encoding outgoing and incoming notifications, ping and control messages. This includes Cyclic Redundancy Check (CRC) calculation and validation, as well as base64 encoding and decoding.

The library `libuavint` handles the internal communication between the wireless mesh node and the flight electronics of the UAV using the serial port. It is a library similar to `libuavext` and included only in the `uavcontroller` software running on the wireless mesh node. Like `libuavext`, it manages both, the outgoing and incoming traffic on the serial interface and is responsible for the following tasks:

- Managing the serial connection between the wireless mesh node and the flight electronics of the UAV.
- Sending and receiving the various messages between the wireless mesh node and the UAV.

The navigation controller is directly connected to the flight controller and has a debug port, which is used for the connection with the wireless mesh node. The navigation controller sends periodically messages to the wireless mesh node that contain information about its current position, flight direction and speed and other important data. The `uavcontroller` receives and processes these messages. Afterwards, it sends them to other wireless mesh nodes and clients or it instructs the UAV with new commands using the serial connection, e.g., the command to fly directly to a specific GPS position.

C. IP Address Assignment

IP address assignment in UAVNet is as follows. All UAVs use static IP addresses, which are defined during the common configuration and setup procedures in ADAM [8]. The end systems and configuration devices use dynamically assigned IP addresses. For this purpose, a Dynamic Host Configuration Protocol (DHCP) server is running on each wireless mesh node. This approach ensures on one side a well-structured and

well-defined wireless mesh network. On the other side, the users on the ground do not need to configure their devices manually to connect to the network.

D. Application Scenarios

UAVNet is currently supporting airborne relay scenarios, where one or several UAVs hover between two end systems to provide a network bridge in the air. The airborne relay scenarios offer a solution to establish a wireless connection between two end systems, which do not get a direct connection, if they are positioned too far away from each other or if there are obstacles between them. Therefore, one or multiple UAVs autonomously position themselves between the two end systems. The wireless mesh nodes attached to the UAVs forward the traffic and thus enable the communication between the two end systems. If the two end systems are too far away from each other and one UAV is not sufficient to establish a connection, multiple UAVs can build a kind of forwarder chain.

1) Airborne Relay Scenario with a Single UAV

In the first scenario, one UAV is sufficient to establish a connection between two end systems on the ground. The UAV flies autonomously between the two end systems and let the attached wireless mesh node forward the traffic between the end systems. The airborne relay scenario with one UAV is shown in Figure 4. The UAV starts near the first end system and begins to broadcast ping messages regularly. The first end system receives these messages and sends its own GPS position to the UAV. Depending on the defined searching mode (see Section III.E), the UAV begins to fly into a given direction or along a spiral track around the first end system to search the second end system. If the second end system receives the broadcasted ping messages, it sends its own GPS position to the UAV. The *uavcontroller* calculates and measures the position between the two end systems and directs the UAV to this calculated centre position. When it arrives at its calculated centre position, it can be reached by both end systems and forward the traffic between them.



Fig. 4. Airborne Relay scenario with a single UAV

2) Airborne Relay Scenario with Multiple UAVs

If one UAV is not sufficient to bridge the distance between the two end systems on the ground, multiple UAVs can be used. They build a chain to forward the traffic between the end systems via several wireless mesh nodes. Such an airborne relay scenario with multiple UAVs is shown in Figure 5. Setting up such a chain of multiple UAVs works as follows:

1. The first starting UAV acts as a scout spotting the exact position of the second end system. Therefore, it uses the same searching algorithms as described in the airborne relay scenario with a single UAV. As soon as it gets the position message from the second end system, it broadcasts this position with the ping and notification messages. This mechanism ensures that all succeeding UAVs

will learn the location of the second end system.

2. It then positions itself in the centre between the two end systems.
3. When it arrives at this centre position, it begins to fly into the direction of the first end system until it receives a predefined signal strength value. This value has been determined by experiments to support both network performance and reliability. This location is locked then as the final position of the first UAV.
4. The next UAV flies directly to the location of the first UAV and moves then into the direction of the second end system until it reaches the pre-defined signal strength to the previously positioned UAV.
5. This is repeated until all UAVs have been positioned accordingly.

This process ensures that the entire chain is built up in the correct direction to ensure optimal connectivity. To avoid collisions, the UAVs can fly at a higher altitude and settle down at a lower altitude, when having reached their final destination.



Fig. 5. Airborne Relay scenario with multiple UAVs

E. Searching Modes

To ensure optimal connectivity between the end systems on the ground, it is important that the UAVs know their exact locations. To achieve this, the end systems transmit their exact GPS coordinates to the UAV. To receive also the position of the second end system, the UAVs must fly into its transmission range to receive their sent position messages. The *uavcontroller* uses two different searching modes to find the second end system.

1. In case of the manual searching mode, the first end system must tell the UAV into which direction it should fly to find the other end system.
2. Using the autonomous searching mode, the UAV can find the second end system automatically without a description into which direction to fly. The UAV calculates multiple waypoints that are on a spiral track around the starting point. Then, it will follow this calculated spiral route until it gets a position message from the second end system. Finally, it positions itself between the two end systems using location based or signal strength based positioning.

For the proper operation of UAVNet, it is important that the UAVs periodically announce their presence and submit some state information about the network and their own status. Therefore, each UAV broadcasts a ping message every few seconds. The ping messages can be received by all other participants like UAVs, smartphones, notebooks, etc. and contain some data about the network and the UAV. By receiving such a message, the devices know that a UAV is within their trans-

mission range.

F. Positioning Modes

UAVNet uses two positioning modes to place the UAVs between the end systems. The first one is the location based positioning mode. It uses the submitted GPS locations of the end systems and directs the UAV to the exact geographical position between these two GPS coordinates. The second one is the signal strength positioning mode. It extends the location positioning mode and includes also the received signal strength of the two end systems to calculate a more accurate position for the UAV. This takes the quality of the wireless connection and other environmental influences into account.

IV. EVALUATION

A. Single-Hop Airborne Relay Scenario

Evaluations with a fully functional single-hop airborne relay setup have been done. This includes two stationary placed notebooks as end systems and a flying UAV with an attached wireless mesh node working as an airborne relay. We evaluated the advantage of our flying UAVNet approach by comparing them to scenarios, where the wireless mesh nodes were placed on the ground. Additionally, the difference between location based and signal strength based positioning have been evaluated. Using the setup shown in Figure 6 several TCP throughput measurements have been performed. Two end systems were placed on the ground, with a distance of 75 m between them. They stayed at the same place during all measurements. The UAV was placed and moved at an altitude of 3-5 m above the ground. Both notebooks were connected to the UAV using IEEE 802.11g. TCP throughput measurements between client1 and client2 have been performed multiple times with the same configuration and parameters, but with the UAVs always located at the following positions:

1. The UAV was placed on the ground, directly beneath client1 to simulate no flying UAV between the notebooks.
2. The UAV was placed on the ground, exactly in the middle between the two notebooks, to simulate a ground based approach using location based positioning.
3. The UAV was hovering exactly in the middle between the two notebooks to evaluate the UAVNet approach. (location based positioning)
4. The UAV was placed on the ground, with the same signal strength to both notebooks to simulate a ground based approach.
5. The UAV was flying with the same received signal strength from both notebooks to evaluate the UAVNet approach. (signal strength positioning)

Figure 7 depicts the average TCP throughput from one notebook to the other one with the corresponding standard deviation (whiskers), depending on the location of the UAV. Measurements with the UAV located at position 1 resulted in a very low TCP throughput of 0.064 Mbps, which indicates that the end systems had almost no connectivity. The evaluation

shows that there is a huge difference between ground-based approaches and our flying UAVNet proposal. With a flying UAV, we reached a 5.3 to 6.3 times higher throughput than when the UAV is positioned on the ground. Signal strength positioning resulted in a 24% - 46% higher throughput compared to location based positioning. The throughput was 127 times higher than without a UAV positioned between the two notebooks.

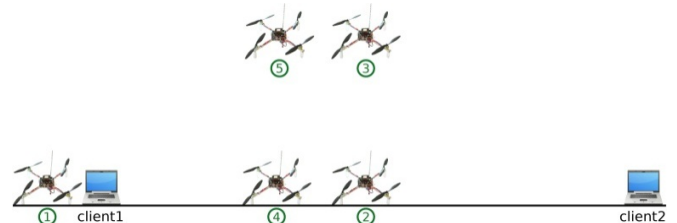


Fig. 6. End-to-end throughput setup

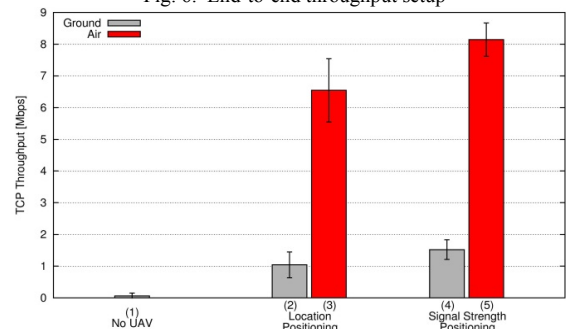


Fig. 7. End-to-end TCP throughput in a single-hop airborne relay setup

B. Multi-Hop Airborne Relay Scenario

We also evaluated a multi-hop airborne relay scenario as depicted in Figure 8. To compare the results to a ground based approach, two different scenarios have been deployed. In scenario 1 two UAVs were placed on the ground. Scenario 2 uses a real multi-hop airborne relay setup with two flying UAVs. Several TCP throughput measurements have been performed between client1 and client2. The two UAVs were deployed in-between the clients. The traffic from client1 was forwarded via both UAVs to client2.

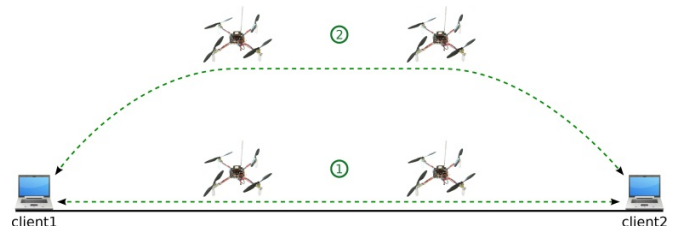


Fig. 8. End-to-end throughput setup in a multi-hop airborne relay scenario

Figure 9 depicts the average TCP throughput of ten measurements from one client1 to client2 with the corresponding standard deviation (whiskers). The measured throughput in scenario 1 is much lower than in scenario 2. With the UAVs placed on the ground we achieved an average throughput of 0.266 Mbps, while the flying UAVs produced an average throughput of 1.868 Mbps. The performance of the multi-hop airborne relay setup is lower than the performance of the single-hop airborne relay scenario, where one UAV is sufficient

to cover the shorter distance between the two end systems. However, the multi-hop airborne relay scenario achieves a significantly higher network throughput compared to single-hop scenarios with too large distances between the nodes.

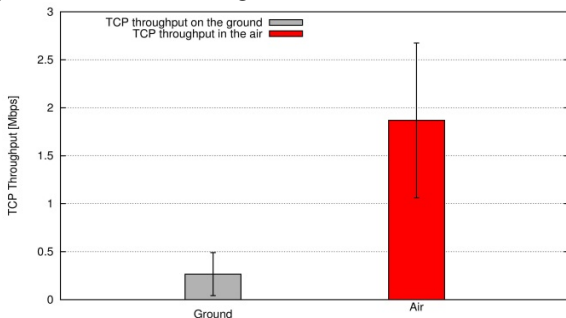


Fig. 9. End-to-end TCP throughput between two notebooks in a multi-hop airborne relay setup

I. CONCLUSIONS

This paper introduced UAVNet, an architecture and prototype implementation of an autonomously deployable temporary and flying IEEE 802.11s WMN. The main purpose of UAVNet is to provide a possibility to deploy a complete communication network in emergency and disaster recovery scenarios in an easy and fast way. The concept is based on small quadcopter UAVs with attached lightweight wireless mesh nodes. The wireless mesh nodes are directly connected to the flight electronics of the UAVs and control the autonomous network deployment. The entire network can be configured, deployed, and monitored by a single user, using a user-friendly remote control application running on an iPad or iPhone. This remote control application monitors the entire UAVNet and displays all involved participants such as UAVs and clients on an interactive map.

We achieved to implement UAVNet without having to modify the original firmware running on the flight electronics of the UAVs. This guarantees optimal compatibility and expandability of the system. The system could be adapted to be deployed on UAVs using different flight electronics. UAVNet uses a uniform communication protocol to exchange commands and data between the flight electronics, attached wireless mesh nodes, and wirelessly connected devices. The integrated IEEE 802.11s wireless mesh network protocol provides improved performance in a highly mobile network.

UAVNet has some significant advantages compared to a ground-based network approach: It has been shown, that the flying wireless mesh nodes result in an up to 6.3 times higher throughput. The UAVNet prototype implementation provides sophisticated searching and positioning algorithms. UAV placement is not restricted by the environment and the deployment is simple and fast and can be performed by a single person. To optimize network performance, an optimal distance between the UAVs has been determined by experimental evaluation. Results of other tests confirm this optimal placement. The nodes are placed close enough to each other to achieve good received signal strength, but sufficiently far away from

each other to cause not too much interference. Further details on this work can be found in [16] and [17].

ACKNOWLEDGMENT

The work described has been partly funded by the Swiss National Science Foundation, grant number 200021-130211.

REFERENCES

- [1] T. X. Brown, B. Argrow, C. Dixon, S. Doshi, R. Thekkekkunnel, and D. Henkel, "Ad hoc UAV-Ground Network (AUGNet)," in *Proc. of 3rd Unmanned Unlimited Technical Conference*, Chicago, 2004.
- [2] D. L. Gu, G. Pei, H. Ly, M. Gerla, B. Zhang and X. Hong, "UAV aided intelligent routing for ad-hoc wireless network in single-area theatre," in *Proc. of Wireless Communications and Networking Conference, IEEE WCNC*, 2000.
- [3] K. Daniel, B. Dusza, A. Lewandowski and C. Wietfeld, "AirShield: A System-of-Systems MUAUV Remote Sensing Architecture for Disaster Response," in *Proc. of 3rd Annual IEEE International Systems Conference*, Vancouver, Canada, 2009.
- [4] S. Rohde, N. Goddemeier, C. Wiefeld, F. Steinicke, K. Hinrichs, T. Ostermann, J. Holsten and D. Moormann, "AVIGLE: A System of Systems Concept for an Avionic Digital Service Platform Based on Micro Unmanned Aerial Vehicles," in *Proc. of IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2010.
- [5] Chen-Mou Cheng, Pai-Hsiang Hsiao, H.T. Kung and D. Vlah: Performance Measurement of 802.11a Wireless Links from UAV to Ground Nodes with Various Antenna Orientations, 15th International Conference on Computer Communications and Networks, 2006. ICCCN 2006., pp.303-308, 9-11 Oct. 2006.
- [6] D. Hague, H.T. Kung and B. Suter, "Field Experimentation of Cots-Based UAV Networking", IEEE Military Communications Conference, MILCOM 2006, pp.1-7, October 23-25, 2006.
- [7] S. Hauert, S. Leven, J.-C. Zufferey and D. Floreano, "Communication-based Swarming for Flying Robots," International Workshop on Self-Organized System, Zurich, Switzerland, 2010.
- [8] H. Buss and I. Busker, "Mikrokoetter Platform," <http://www.mikrokoetter.de>, Nov. 2011.
- [9] "Logical Level Converter," <http://sparkfun.com/products/8745>, Nov. 2011.
- [10] Open-Mesh, "OM1P 802.11g Mid Power Mini Router," <http://www.open-mesh.com/index.php/professional/professional-mini-router-us-plugs.html>, 2011.
- [11] Th. Staub, S. Morgenthaler, D. Balsiger, P. K. Goode, and T. Braun, "ADAM: Administration and Deployment of Adhoc Mesh Networks," in *Proc. of 3rd IEEE Workshop on Hot Topics in Mesh Networking (IEEE HotMESH 2011)*, Lucca, June 2011.
- [12] PC Engines, "ALIX system boards," <http://www.pceingines.ch>, 2011.
- [13] "Linux Kernel," <http://www.kernel.org>, Nov. 2011.
- [14] "Ath5k - Wireless Driver for Atheros Based Wireless Chipsets," <http://www.linuxwireless.org/en/users/Drivers/ath5k>, Nov. 2011.
- [15] "Compat-wireless," <http://www.linuxwireless.org/en/users/Download>, Nov. 2011.
- [16] Th. Staub: Development, Testing, Deployment and Operation of Wireless Mesh Networks, Ph.D. Thesis University of Bern, May 2011.
- [17] S. Morgenthaler: UAVNet: A Prototype of a highly Adaptive and Mobile Wireless Mesh Network Using Unmanned Aerial Vehicles (UAVs), Master Thesis, University of Bern, June 2012