Quadrotor-based DT-WSNs for Disaster Recovery

Felix Büsching, Keno Garlichs, Johannes van Balen, Yannic Schröder, Kai Homeier, Ulf Kulau,
Wolf-Bastian Pöttner, Stephan Rottmann, Sebastian Schildt, Georg von Zengen, and Lars Wolf
Technische Universität Braunschweig
Institute of Operating Systems and Computer Networks (IBR)
Mühlenpfordtstr. 23, 38106 Braunschweig
[buesching | garlichs | vanbalen | yschroed | homeier | kulau | poettner | rottmann | schildt | vonzengen | wolf]@ibr.cs.tu-bs.de

***Abstract*—How to establish a communication infrastructure when there is no infrastructure at all? After an occurred disaster there is a high demand for functional and working communication. In this paper we propose flying and self-deploying Wireless Sensor Networks (WSNs) to establish disruption tolerant multi-hop communication for disaster recovery and search-and-rescue missions. We also show the implementation of our first WSN-based quadrotor prototype.**

***Index Terms*—Quadrotor; Wireless Sensor Networks; Desaster Recovery; Disruption Tolerant Networking**

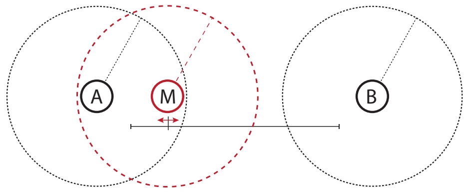

Fig. 1. Node M is moving between nodes A and B. Node M stores, carries and forwards data between nodes A and B.

I. Motivation and Introduction

"Roads? Where we're going, we don't need roads."[1]

Rescue teams rely on images from distant cameras, remote operators try to navigate robots through harsh and inhospitable environment, pollution data recorded by scattered WSNs is analyzed in order to assess the possibility to send humans in contaminated areas:

There are several scenarios for WSNs deployed in unknown territory or after an occurred disaster. Most of these scenarios require the presence of at least some communication infrastructure that is able to transmit data to a remote location, e.g. via the internet. A satellite uplink cannot always be assumed as on the one hand the satellite infrastructure may be damaged as well; on the other hand the communication via satellites requires a free line of sight (to the satellite), which again is not given under heavy smoke or in indoor scenarios like caverns, mines or nuclear power plants.

A. Disruption Tolerant Networks for Disaster Recovery

The network link to a remote control center does not always have to be a continuous end-to-end connection. The concept of a Disruption Tolerant Network (DTN) (or synonym *Delay Tolerant Network*) has its origins in interplanetary communication, where usually a continuous connection cannot be assumed [1]. Traditional communication protocols fail in harsh deep-space environments as they are inappropriate due to several reasons – and they will fail in some disaster recovery scenarios as well: Long distances result in high latency, which again makes connection oriented protocols like TCP unmanageable, and the absence of a continuous end-to-end connection requires a different approach than common communication protocols. The DTN architecture [2] is based on a "store, carry and forward" concept and is able to compensate these shortcomings.

[1]Dr. Emmett Brown in *Back to future* (1985)

In Figure 1 the general functionality of a DTN is shown. There is no need for the nodes to have a continuous connection. The data is organized in so-called Bundles; Bundles can be stored, (physically) carried and forwarded if another node is in communication range. Also WSN projects such as ZebraNet [3] follow a DTN-like approach. However, all these approaches are located in the application layer, using standard protocols and are designed for one special purpose, each.

II. Related Work

The deployment of nodes in a network can be done in several ways – surely depending on size of the deployment, the area, and the environment. In contaminated disaster areas, obviously, a manual deployment is out of question.

Since the beginning of WSNs research large scale deployments via airplanes have been promoted; but – to the best of authors knowledge – have never been performed in research.

In [4] and [5] unmanned helicopters have been used to deploy the nodes of a wireless network. Whereas the drones have either been controlled remotely or have operated autonomously.

In previous works we have shown the basic concept of a vehicle that drops intermediate nodes as soon as the RSSI becomes bad [6]. By this, the vehicle itself maintains the deployment of the wireless sensor network which is used to control the vehicle and to transport data from the vehicle to a distant operator. This concept has been successfully tested at the Eyjafjallajökull volcano [7] in Iceland. In this work the vehicle was controlled by a distant human operator. Nevertheless, even autonomous driving vehicles like [8] need a communication infrastructure to transmit the recorded data.

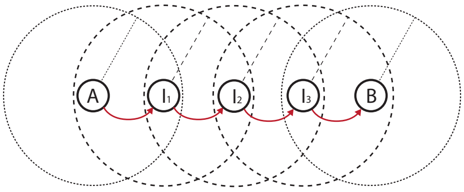

Fig. 2. Several intermediate nodes I_n are needed to finally cover the area of node B.

III. Applications

The basic idea is very simple. A quadrotor – equipped with WSN hardware to establish radio links – is controlled via WSN radio links by an operator who navigates the drone through unknown territory. Like in [7] a camera can transmit images of the current environment. At the moment where the RSSI sinks beyond a certain threshold, the quadrotor holds the position or – if possible – lands to save energy. Afterwards, a second drone is started – following the first until the two drones meet at the same spot. The first – probably landed – drone now acts as a relay and holds the position. Through this relay, the actuation radius of the second drone is enhanced and it can continue to explore the surroundings.

A. Continuous Network

To explore wider areas, each time the RSSI drops below a certain threshold a new drone is started and the current node holds its position. This surely increases the number of hops in the multi-hop network which is formed by this strategy. Thus, also latency will increase and controls will act more delayed with an increasing number of hops. But, after the flying nodes have covered the desired area, the WSN can be used to transmit data relevant for the disaster recovery mission and "normal" network communication can be transmitted via this flying – or once flying now landed – WSN. In Figure 2 this scenario is shown: Node A is the sink and the intermediate I_n nodes start one after another and fly to their designated positions in the multihop network. Finally, node B covers the desired area of interest.

B. Disrupted Network

Depending on the area to be covered, the first approach may be a waste of material, since a lot of drones will be placed in the area. In case the "interesting" spot is at the far end of a chain (node B in Figure 2), most of the flying or landed nodes will only work as relays. This is the point where the DTN protocol really helps to save material and money: In [9] elevators have been used to physically carry data. Here, the quadrotors can be used to shuttle between two – or more – spots and store, carry and forward data, as shown in Figure 1. Thus, the same area which was covered by five nodes in Figure 2) can be covered by only three nodes in Figure 1.

IV. Implementation

The implementation of the quadroter is based on the INGA wireless sensor node [10]. In Figure 3 all components and their interconnections are given.

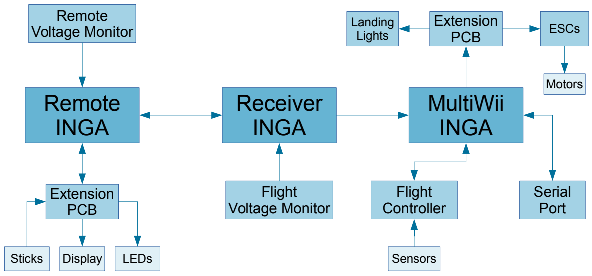

Fig. 3. Hardware diagram of the WSN-based quadrotor.

Fig. 4. The flying WSN-based quadrotor.

All in all, three INGA nodes have been used – one as remote control, one for flight control and one for communication purposes. While the remote control node is located in a standard (toy) remote control, the other two form the actual quadrotor.

The flight controller is based on the MultiWii project[2]. To easily adapt INGA to this Arduino based project, our arduINGA[3] port which makes INGA work as an Ardunio is utilized. The flight controller utilizes INGAs gyroscope and accelerometer for trajectory calculation. It also controls – via an extension with one Electronic Speed Control (ESC) per motor – the four motors and the landing lights.

The receiver runs Contiki [11] and therefore is able to communicate via many existing protocols. Additionally we used this node to monitor the voltage of the flying system.

The remote control node is also running Contiki and interfaces the controller sticks, a display and several LEDs. At first, the RIME communication protocol was used to transfer flight commands and sensor data. To enable a disruption tolerant communication, the μDTN protocol [12] has been used.

V. Conclusion

Figure 4 shows the flying prototype of the WSN-based quadrotor. Unfortunately – until now – we were only able

[2]http://www.multiwii.com

[3]http://git.ibr.cs.tu-bs.de/?p=project-cm-2012-inga-arduingo.git

to build the one quadrotor and thus we were not able to really test our concepts of a disruption tolerant disaster recovery with WSN-based quadrotors.

In contrast to [5] and [4] in our case, the flying vehicles are not used to drop nodes – the quadrotors are meant to be the (relaying) nodes itself: In a *Continuous Network* as (more or less) dumb relays; in a *Disrupted Network* as data mules that store, carry and forward data within the network.

A. Future Work

WSN-capable quadrotors are pretty seldom right now, but, the vehicles presented in [7] rely upon the same technology. Thus, they are compatible and a combined ground- and airborne DTN can be formed. Additionally, some rockets could support this scenario [13].

We also plan to equip the quadrotor with GPS, so that it can search wider areas autonomously with less user interaction. In addition to that, there is the possibility to make use of a computer vision system like the one presented in [14] to make a map of the area. Having received this map via DTN, a ground vehicle could be enabled to navigate in the harsh environment encountered and carry heavier payloads to a target. If this system was able to detect interesting spots (like humans needing help) and send a picture to the operator via DTN this could reduce the network load per drone dramatically. Hence not only the operators would be able to control more drones, but also the network would be capable of handling more quadrotors or other Urban Search and Rescue (USAR)-vehicles.

REFERENCES

[1] Kevin Fall. A delay-tolerant network architecture for challenged internets. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '03, pages 27–34, New York, NY, USA, 2003. ACM.

[2] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss. Delay-Tolerant Networking Architecture. RFC 4838 (Informational), April 2007.

[3] Philo Juang, Hidekazu Oki, Yong Wang, Margaret Martonosi, Li Shiuan Peh, and Daniel Rubenstein. Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet. In *Proceedings of the 10th international conference on Architectural support for programming languages and operating systems*, ASPLOS-X, pages 96–107, New York, NY, USA, 2002. ACM.

[4] P. Corke, S. Hrabar, R. Peterson, D. Rus, S. Saripalli, and G. Sukhatme. Autonomous deployment and repair of a sensor network using an unmanned aerial vehicle. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, volume 4, pages 3602–3608 Vol.4. IEEE, 2004.

[5] Anibal Ollero, Pedro J. Marron, Markus Bernard, Jason Lepley, Marco la Civita, Eduardo de Andres, and Lodewijk van Hoesel. AWARE: Platform for Autonomous self-deploying and operation of Wireless sensor-actuator networks cooperating with unmanned AeRial vehiclEs. In *2007 IEEE International Workshop on Safety, Security and Rescue Robotics*, pages 1–6. IEEE, September 2007.

[6] Sebastian Schildt, Stephan Rottmann, and Lars Wolf. Communication Architecture, Challenges and Paradigms for Robotic Firefighters. In *Proceedings of the 5th Extreme Conference of Communication (ExtremeCom 2013)*, Thorsmork, Iceland, August 2013.

[7] Stephan Rottmann, Sebastian Schildt, Keno Garlichs, and Lars Wolf. Demo: Using DTN for controlling a vehicle with a self deployed WSN. In *Proceedings of the 5th Extreme Conference of Communication (ExtremeCom 2013)*, Thorsmork, Iceland, August 2013.

[8] Fred W. Rauskolb, Kai Berger, Christian Lipski, Marcus Magnor, Karsten Cornelsen, Jan Effertz, Thomas Form, Fabian Graefe, Sebastian Ohl, Walter Schumacher, Jörn-Marten Wille, Peter Hecker, Tobias Nothdurft, Michael Doering, Kai Homeier, Johannes Morgenroth, Lars Wolf, Christian Basarke, Christian Berger, Tim Gülke, Felix Klose, and Bernhard Rumpe. Caroline: An autonomously driving vehicle for urban environments. *J. Field Robot.*, 25(9):674–724, 2008.

[9] Wolf-Bastian Pöttner, Felix Büsching, Georg von Zengen, and Lars Wolf. Data Elevators: Applying the Bundle Protocol in Delay Tolerant Wireless Sensor Networks. In *The Ninth IEEE International Conference on Mobile Ad-hoc and Sensor Systems (IEEE MASS 2012)*, Las Vegas, Nevada, USA, October 2012.

[10] Felix Büsching, Ulf Kulau, and Lars Wolf. Architecture and Evaluation of INGA - An Inexpensive Node for General Applications. In *Sensors, 2012 IEEE*, pages 842–845, Taipei, Taiwan, oct. 2012. IEEE.

[11] Adam Dunkels, Björn Grönvall, and Thiemo Voigt. Contiki - a Lightweight and Flexible Operating System for Tiny Networked Sensors. In *Proceedings of the First IEEE Workshop on Embedded Networked Sensors (Emnets-I)*, Tampa, Florida, USA, November 2004.

[12] Georg von Zengen, Felix Büsching, Wolf-Bastian Pöttner, and Lars Wolf. An Overview of μDTN: Unifying DTNs and WSNs. In *Proceedings of the 11th GI/ITG KuVS Fachgespräch Drahtlose Sensornetze (FGSN)*, Darmstadt, Germany, 9 2012.

[13] Ulf Kulau, Sebastian Schildt, Georg von Zengen, Stephan Rottmann, and Lars Wolf. Ballistic deployment of wsn nodes using model rockets. In *Proceedings of the 6th Extreme Conference of Communication (ExtremeCom 2014)*, Galapagos Islands, Ecuador, August 2014.

[14] D Scaramuzza, MC Achtelik, L Doitsidis, F Fraundorfer, EB Kosmatopoulos, A Martinelli, MW Achtelik, M Chli, SA Chatzichristofis, L Kneip, et al. Vision-controlled micro flying robots: from system design to autonomous navigation and mapping in gps-denied environments. *IEEE Robotics & Automation Magazine*, pages 1–10, 2013.