A Communication Layer for UAV/UGV Swarm Applications

Hannes Harms[1], Jan Schattenberg[1], Julian Schmiemann[1], Ludger Frerichs[1]
[1]Institute of Mobile Machines and Commercial Vehicles, TU Braunschweig, Germany
hannes.harms@tu-braunschweig.de

Keywords: Search and Rescue Robotics, Middleware, MANET, Machine Cluster

Abstract: In this paper we present an approach to connect a machine cluster of unmanned ground and aerial vehicles inside a MANET to exchange sensor data requiring a high bandwidth. The introduced communication layer extends the messaging system of an underlying middleware to work with non-ideal wireless links. Every connection can be configured with a user defined data rate and protocol to control the overall traffic load in the network. Moreover the message transport system contains mechanisms for lossless and lossy data compression.

1 INTRODUCTION

One key challenge in civil protection or search and rescue tasks is to provide an information base of the environment on which rescue forces can rely on. Within the joint development project "ANKommEn" a semi-automatic machine cluster consisting of unmanned aerial vehicles (UAV) and unmanned ground vehicles (UGV) is developed to collect, process and visualize different sensor data on-the-fly. The consortium of "ANKommEn" (german acronym for: "Automated Navigation and Communication for Exploration") consists of the Institute of Mobile Machines and Commercial Vehicles, the Institute of Flight Guidance – both Technische Universität Braunschweig – and the AirRobot GmbH & Co. KG. It is funded by the German Federal Ministry of Economic Affairs and Energy and administrated by the Space Administration of the DLR.
The main concept of "ANKommEn" is, that each swarm unit is equipped with a different sensor setup e. g. a high resolution camera for aerial images, a 3D-LiDAR sensor for mapping or a thermal camera for e. g. firefighting support. The sensor data is used to create a data base to decide about adequate rescue actions or to coordinate human rescue forces. This paper will focus on the communication system to connect all swarm participants among themselves, as well as to connect the UAV/UGVs with a ground station for controlling the vehicles and visualization purposes including sensor live streams.
The therefore used network is based upon a VHT (Very High Throughput) 802.11ac mesh network in ad-hoc mode. Mobile ad-hoc networks (MANETs) can be used without any fixed networks infrastructure, which makes them suitable for disaster scenarios or highly mobile nodes like Car2Car networks. Moreover, MANETS provide a very flexible mechanism to form a network between participants that meet in a certain area. With regard to the use cases of "ANKommEn" it is possible to immediately connect rescue forces with the UGVs or UAVs that are in operation.

In "ANKommEn" the swarm consist of two UGVs, both with a 3D-LiDAR sensor (Velodyne VLP-16) and three UAVs, equipped with different sensors (Velodyne VLP-16, high resolution optical camera and thermal camera). The wireless communication has to deal with a high network load and should be as robust as possible e. g. to deal with lossy links and link failures. Therefore a communication architecture is proposed, which is capable to deliver high throughput traffic within a fine grained mechanism to control the underlying transport protocol and the delivery rate of each data channel.

2 RELATED WORK

The communication system for a distributed machine cluster has to meet different requirements depending on the application. Usually some of the main challenges are the physical network setup and configuration, the underlying routing protocol, which has to take node mobility into account and user space

applications that have to be aware of the behavior of a MANET.
Another research area for data exchange between mobile machines and especially autonomous systems is a software stack for passing messages form one process to another in a so called *middleware.* With increasing software complexity a middleware becomes more and more important.
Both issues, the middleware system and MANET setup will be discussed shortly in the following.

2.1 Middleware

During the last years several frameworks for inter process communication have been developed. Some of them with the most valuable impact on the community are ROS (Robert Operating System) (Quigley, 2009), LCM (Lightweight communications and marshalling) (Huang, 2010), and especially for UAV the Pixhawk MAVCONN Middleware (Meier, 2012). ROS uses a TCP or UDP publisher/subscriber concept for message transport, but the framework has never been designed to run on a distributed network including non-ideal wireless links. The centralized approach of the *roscore* makes it inadequate for swarm communication. LCM also implements the publisher/subscriber concept, but in contrast to ROS a decentralized approach is used and LCM is capable of soft real-time message transmission. The decentralized approach of LCM is achieved by sending all messages as UDP multicast which would be a very expensive method in respect of bandwidth usage when dealing with MANETs under high traffic load. The MAVCONN middleware has an interface for ROS and LCM to exchange data with a ground station using UDP messages, but the swarm capabilities of the framework are limited.

2.1 MANET Routing

A major challenge for MANETs is the routing protocol, which determines the routes that network packages have to travel to be successfully delivered. In a MANET routes can change, because nodes are not stationary. Several solutions for this problem were developed in the last years, e. g. the Optimized Link State Routing protocol (OLSR) (Clausen, 2003), Better Approach To Mobile Adhoc Networking (BATMAN) (Johnson, 2008) or Babel (Chroboczek, 2011).
We chose BATMAN, because operating on the data link layer (layer 2) offers the possibility to apply different protocols on the transport layer (layer 3).

3 NETWORK SETUP

For the network configuration two different approaches should be mentioned here. One possibility is to establish a mobile ad-Hoc networks in 802.11 Infrastructure mode as described by Wirtz et al. (2011). Another common approach is to set up the ad-hoc network with client using the IBSS mode (Independent Basic Service Set), as used herein.
Each swarm node is equipped with a central communication board (Gateworks GW 5520, 800 Mhz i.MX 6) that interfaces with a processing unit (Intel Nuc i5) and a navigation unit (Phytec Mira), as shown in Figure 1. The main sensor is attached to the processing computer and GNSS and IMU are connected to the navigation board. We are using two WiFi modules on the communication board, operating at 5.5 GHz (80 MHz channel width) and 2.4 GHz (40 MHz channel width). The WiFi interfaces are linked to a virtual network device (bat0) that is created by the used routing protocol BATMAN. Both ethernet adapters (eth0, eth1), as well as the virtual mesh interface bat0 are then bridged together to an overall network interface, to which the IP-address is assigned to.

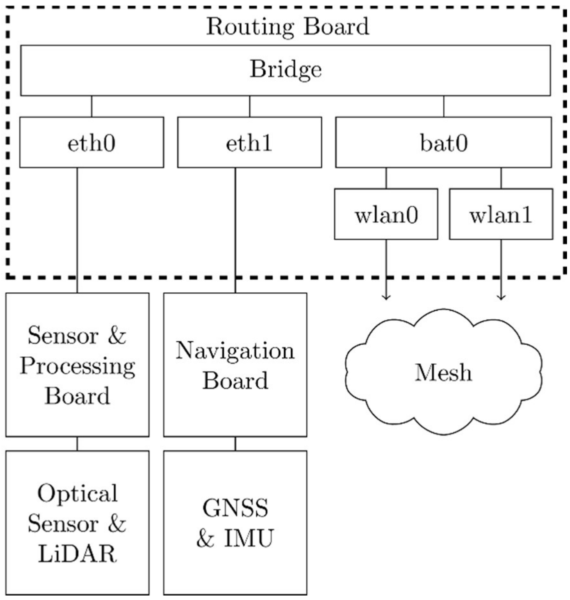

Figure 1: Network structure for a single mesh node (UAV/UGV)

3 COMMUNICATION ARCHITECTURE

The presented communication layer is designed to work on top of a middleware system that handles the inter-process communication between applications in a subnetwork that could reliable communicate, e. g. some clients connected through Ethernet. Currently ROS is used for the inter process communications, but it is also thinkable to interface with other middleware systems. Each swarm client runs an independent *roscore* on the communication board. ROS nodes executed on other wired computers connected to the communication unit hook up with the local *roscore*. For transmitting and receiving ROS topics with wireless connected mesh neighbors the publisher and subscriber mechanism of ROS has to be extended. Therefore the regular topic transport mechanism is spitted up and enlarged with a node for sending and receiving topics via an unreliable WiFi connection. In a similar approach Schwarz et al. (2016) build up the communication for the NimbRo rescue robot this way.

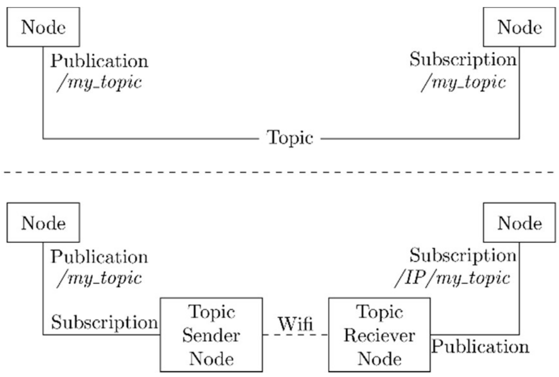

Figure 2: ROS topic transport extension

In our approach an additional set of state nodes (sender state and receiver state node) to distribute the local available topics of a single *roscore* to all other swarm neighbours is used, similar to the ROS master discovery developed by Tiderko et al. (2016). The sender state node requests the current state of the *roscore* and publishes a message containing a list of all available topics and services. The topic sender node subscribes the sender state messages for transmitting the message via broadcast periodically to all other network participants. To avoid an endless loop of topics transmissions, the sender state node takes also account to the already received state messages coming from other swarm clients. By doing this the sender state message only contains the original topics and services of the local *roscore*, therefore retransmissions of external topics are avoided.

Based on the status message the topic sender node is capable of establishing a unique connection through TCP or UDP for each data channel that is requested. As outlined in Figure 3 the topic transport process can be subdivided into some subroutines. First of all, the message is serialized into a byte array and if required the message can also be compressed using a standard zlib DEFLATE algorithm. Before publishing the message on the receiver side, the original message has to be recovered through decompressing and deserialization of the incoming byte stream.

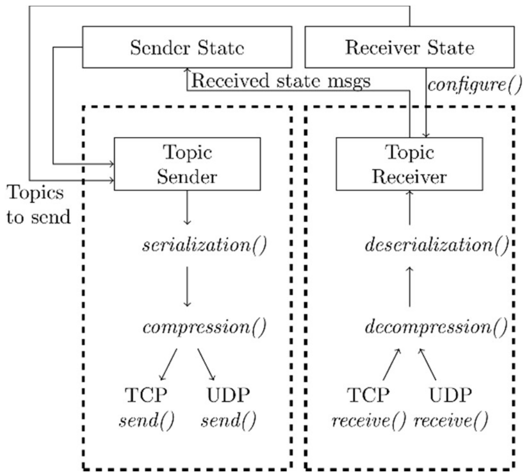

Figure 3: Software layout for the topic transport

Every connection can be configured with a user defined data rate to control the overall traffic load in the network. Furthermore each connection can be adjusted with an individual buffer size on the sender side. In case of a connection loss data can be buffered for a certain period of time. If the connection recovers the buffered data is send out with an appropriate rate to make a trade-off between the configured transmission frequency and the available network capacity. Image topics like aerial or thermal images could be either transferred using the raw data, or if the compression mode is enabled, a JPEG image compression is used by subscribing the ROS built-in compressed image topic.

4.1 Data Compression

One of the main problems in "ANKommEn" is that the utilized sensors can easily overload the network capacity. To prevent this the topic sender and topic receiver nodes contain the already mentioned mechanism for compressing and decompressing messages. Figure 4 shows some exemplary throughput measurements for live streams of a 3D-LiDAR sensor (Velodyne VLP-16), a stereo camera (Multisense S21) and an optical camera. For compression *zlib* was chosen, because the algorithm ensures a fair balance between compression speed and compression ratio.

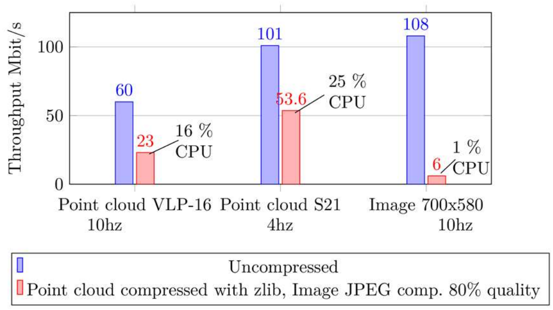

Figure 4: Sensor throughput – raw and compressed

The required bandwidth for the point cloud streams could be reduced by 61 % / 47 % and the corresponding CPU load stays at an acceptable level (16 % / 25 %) for a single core on an Intel Core i5-4250U.

4.2 Video Streaming

To save an additional amount of bandwidth, image topics can also be transferred as a h.264 encoded video stream. For this purpose the already introduced sender state nodes sends a configuration message to a video server node that subscribes the image topic. The video server node is able to inject every image in a Gstreamer pipeline by using a Gstreamer *appsrc* element. The h.264 video streams can be transferred with the RTP (Real-Time Transport) protocol from the UAV/UGV to the ground station.

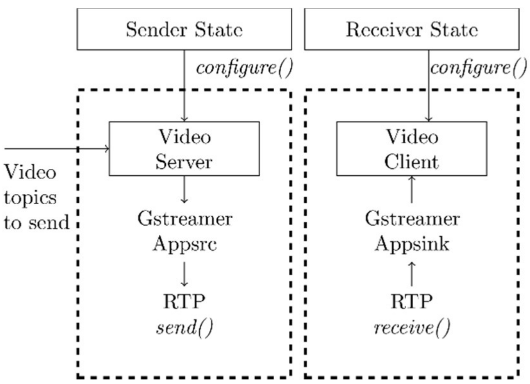

Figure 5: Software layout for video streaming

On the receiver side a video client node is set up with a Gstreamer *appsink* element that handles the incoming RTP stream. The client node is able to convert each frame into an ROS image message for further publishing. First results have shown that the network load for video streaming can be significantly decreased compared to a JPEG stream, as expected. Due to the high amount of processing power needed for h.264 encoding and decoding we are also experimenting with a hardware accelerated Gstreamer pipeline. Using the build-in hardware acceleration of an Intel core i5-3320M the CPU usage could also be reduced from nearly 100 % to 9 % for single core.

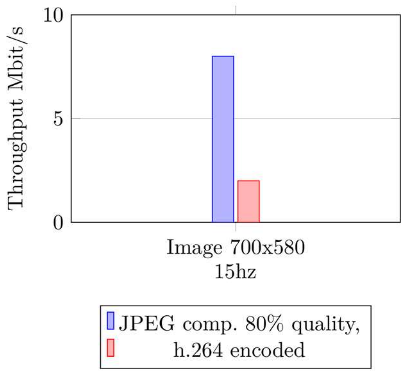

Figure 6: Throughput comparison for a JPEG and h.264 image stream

5 USER INTERFACE

The message transfer between each swarm participant can either be established automatically through predefined configuration files for the sender and receiver state nodes or manipulated via a user interface. The interface is implemented as a plugin for *rviz* and lets the operator switch between the UDP, TCP and RTP protocol for each topic. Apart from this the sending rate, the buffer size and the compression mode can be adjusted. For each data channel the actual bitrate is calculated and displayed to give the user a better understanding of the current network load.

Figure 7: User interface for controlling network traffic

5 THROUGHPUT EXPERIMENTS

In order to estimate if the ad-hoc network is capable to handle the network load first throughput experiments with the UAV and UGV were made. The tests for the ground vehicle were performed in an urban area with a small building in the line of sight to the vehicle. The measurements for the aerial vehicle were done with a free line of sight. In the distance range of 0-160 *m* the throughput for the UAV is lower than the throughput for the UGV due to another antenna shape and different antenna assembly positions, as show in Figure 8. At larger distances the UAV performs better than the UGV. The maximum range for a stable link to the UAV is about 350 *m* using a transmission power of 27 dBm. Figure 8 also contains bandwidth requirements for some of the utilized sensors in ANKommEn.

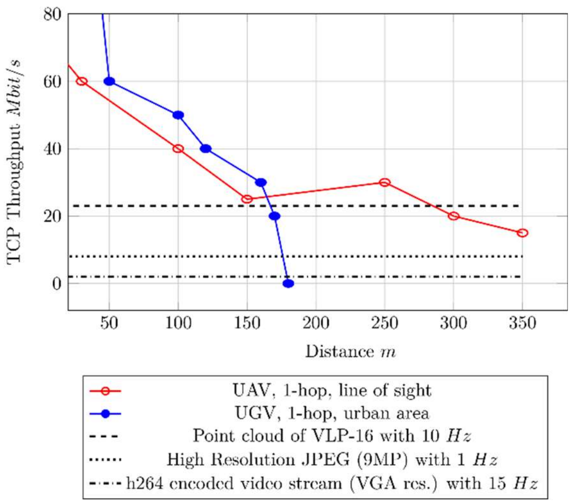

Figure 8: Single hop throughput for UAV and UGV

Of course the network demand highly depends on the number of participants, link distances and type of sensors, but it can be estimated that it is possible to deal with the traffic of imaging sensors in a smaller network. For streaming point clouds, e. g. data of the Velodyne VLP-16, transmission rates have to be limited to an appropriate frequency.

7 CONCLUSION

It is shown that the proposed communication architecture is capable to extend the underlying middleware architecture to exchange messages in an unreliable network. In addition some methods were presented to reduce network load and meet bandwidth requirements. The presented user interface is another option to control data streams and manage different situations. In the future we would like to test if sensor streams can be transferred via multiple hops to exploit the full potential of the mobile ad-hoc network. Further tests for detecting link failures and reconnection time are necessary, especially for TCP connections. It is also thinkable to spent more work on adjusting transmission rates automatically depending on the available network bandwidth in order to avoid preconfigured rates.

ACKNOWLEDGEMENTS

All presented work was done within the joint development project “ANKommEn” funded by the German Federal Ministry of Economic Affairs and Energy administrated by the Space Administration of the DLR (funding code: 50NA1518).

REFERENCES

Quigley, M., Conley, K., Gerkey, B, J. Faust, Foote, T., Leibs, J., Berger, E., Wheeler, R., Ng, A., 2009. ROS: an open-source Robot Operating System. In: ICRA Workshop on Open Source Software.

Huang, A., Olson, E. and Moore, D. LCM: Lightweight communications and marshalling, 2010. In: Proc. of the IROS, pp. 4057–4062

Meier, L., Tanskanen, P., Heng, L., Lee, G., Fraundorfer, F. and Pollefeys, M., 2012. Pixhawk: A micro aerial vehicle design for autonomous flight using onboard computer vision,” Autonomous Robots, vol. 33, pp. 21–39

Clausen, T., Jacquet, P. , 2003. Optimized link state routing protocol. In: Network Working Group RFC 3626

Chroboczek, J., The Babel Routing Protocol, 2011. In: Network Working Group RFC 6126

Johnson, D., Ntlatlapa, N., Aichele, C., 2008, A simple pragmatic approach to mesh routing using BATMAN, 2nd IFIP International Symposium on Wireless Communications and Information Technology in Developing Countries, Pretoria, South Africa

Wirtz, H., Heer, T., Backhaus, R., Wehrle, K., 2011. Establishing mobile ad-hoc networks in 802.11 infrastructure mode. In WiNTECH '11, Proceedings of the 6th ACM international workshop on Wireless network testbeds, experimental evaluation and characterization. Pages 89-90

Tiderko, A., Hoeller, F., Röhling, T., The ROS Multimaster Extension for Simplified Deployment of Multi-Robot Systems, Robot Operating System (ROS): The Complete Reference (Volume 1), Pages 629-650, Springer International Publishing

Schwarz, M., Beul, M., Droeschel, D., Schüller, S., Periyasamy, A., Lenz, C., Schreiber, M., Behnke, S., 2016. Supervised Autonomy for Exploration and Mobile Manipulation in Rough Terrain with a Centaur-like Robot, Accepted for Frontiers in Robotics and AI, section Humanoid Robotics