Macrothink Institute™

Network Protocols and Algorithms
ISSN 1943-3581
2015, Vol. 7, No. 4

Ad hoc Network for Emergency Rescue System based on Unmanned Aerial Vehicles

Carlos Cambra, Sandra Sendra, Jaime Lloret, Lorena Parra

Universidad Politécnica de Valencia.

Camino de Vera, s/n, Valencia (Spain)

E-mail: carcamb1@doctor.upv.es, sansenco@posgrado.upv.es, jlloret@dcom.upv.es, loparbo@doctor.upv.es

Received: December 3, 2015 Accepted: December 31, 2015 Published: December 31, 2015

DOI: 10.5296/npa.v7i4.8816 URL: http://dx.doi.org/10.5296/npa.v7i4.8816

Abstract

Intelligent Unmanned Aerial Vehicles (UAVs) on emergency rescue are widely used to detect injured mountaineers in inaccessible areas. These systems should satisfy several features. It is important to know the georeference of the injured mountaineers. It is also important to have real-time images of the area where people have suffered the accident. In this paper, we present the development of a UAV integrated within a wireless ad hoc network and the communication protocol that is able to transfer data between several UAV's and the Smartphones carried by the mountaineers. This paper also shows how ad hoc networks extend the wireless coverage for emergency situations in critical areas without GSM cellular coverage. After developing our system, we have focused our effort on demonstrating the correct operation of our UAV and its network performance when the system is used to track someone within a zone. Experimental results show the big potential of this kind of networks working in hostile environments such as big mountains, ravines and river canyons without GSM signal communication.

Keywords: Ad-Hoc Networks, Rescue Systems, Communication Protocols, Unmanned Aerial Vehicles (UAVs), Emergency Situations.

1. Introduction

The areas where mobile communication is being applied is growing day by day, allowing the people to be connected with social networks, blogs, news, etc. [1]. In most situations, the dependency with mobile devices and generic communication protocols, such as IEEE 802.11 b/g/n, on emergency situations can be the key to decrease risks. Ad hoc network allows data transmission between end devices in a distributed manner [2]. One of the main characteristics of ad hoc networks is that they can be used in both cases, when there are few nodes or for large networks where it could be difficult to maintain the connection to all nodes from the access points [3][4].

One of the main hot topics in ad hoc networks is to provide network services, such as audio and video delivery, to end users [5]. But in these services the Quality of Service (QoS) must be guaranteed [6] [7] in order to provide a good experience at the end user side [8].

To make more efficient the utilization of Mobile ad hoc networks (MANETs) and to extend the wireless coverage, there are Android applications when travel in areas with low GSM/3G coverage [9]. In emergency cases, mobile devices can connect to Rescue MANET and transfer the geo-location coordinates, and several sensor data [10].

In this paper, we show the design and development of an emergency rescue system based on Unmanned Aerial Vehicles (UAVs), which is able to scan big areas searching Smartphone signal communications of injured or missing person. The goal of this system is to create a big communication network on a hostile environment to communicate or create Geo tags for missing people in areas without GSM signal, and send it to the Emergency Control Center. Our system uses a quadcopter with two wireless interfaces (5.8GHz and 2.4GHz) and a GPS receiver. The remote control uses NAVIO/RaspberryPi a control board, and the selected control software is the Ardupilot Mega (*APM*) Mission Planner. The system integrates a video camera which transmits real time images. It detects the GPS position of the mountaineers and takes pictures of them. These images are sent to the base station in order to check the status of injured persons and their geographical position. This information is analyzed, and depending on the captured images, the emergency services can design an action plan according to the situation. To assess the potential of this system, the tests are performed in real environments. They allowed us to evaluate the QoS and the correct operation of our rescue emergency system. The motivation of carrying out this analysis comes from the need of reducing the reaction time on emergency services for searching missing people on the big mountains of Spain, called Pyrenees, because there are many areas that do not have GSM signal or 3G coverage.

The rest of this paper is structured as follows. Section 2 presents some previous works about rescue systems. The system and its components are explained in Section 3. It also presents the algorithms and some screenshots of our application. Section 4 explains the test bench performed with our system. Finally, Section 5 presents conclusion and future work.

2. Related Work

This section shows several works about rescue systems and applications based on UAVs.

E. F. Flushing et al. [10] presented a GIS-based Mission Support System (MSS) to assist monitoring and decision-making of mission plans. The proposed framework introduces innovative strategies for estimating the search efficacy according to agent and environment characteristics. The proposal is presented as a powerful integrated tool for managing emergency response scenarios. The system is based on a combination of several systems such as a GIS-based environment modeling, the automatic definition of agents' search efficacies, an optimization-based mission planner and a mission simulator. Authors concluded that the use of this MSS in a scenario where a team of human searchers, air-scent dogs and aerial robotic platforms work together can be very useful to perform the search of a missing person in a mountain area.

L.L. Wang et al. [11] presented an aerial remote sensing detection system based on ATMEGA2560 and the application of Mini-UAVs in emergency rescue of major accidents of hazardous chemicals, especially the application of the system in the emergent processing of oil spill. The system is composed by several modules, i.e., gesture detection module, a GPS module, an ultrasonic ranging module, human infrared sensor module, a wireless remote control module, wireless data transmission module and other components. UAV remote control is based on an open source code developed with Arduino development environment. Finally, authors performed summary about the advantages of using UAVs in emergency rescue situation and pointed out the deficiency of similar systems and how to enhance them.

In [12], authors presented two mobility models for reconnaissance application: “random mobility model” and “distributed pheromone repel mobility model”. These mobility models are based on three actions: going straight, turn left and turn right. Away from the UAV networks, in [13], authors evaluated their new mobility model based on the comparison with real movement traces using two properties: the contact duration (the time interval in which two devices are in radio range) and the inter-contacts time (the frequency and the probability of being in contact with the packet receiver). The idea of comparing the new mobility model with real traces seems to be a good way to validate the behavior of their mobility model.

In [14], authors present a framework for visual exploration of disaster areas where there has been an earthquake, a tsunami or an industrial accident. Its aim is to help rescue teams to locate humans and support to recover people. The designed schema provides an aerial vehicle control system based on several network paradigms. The architecture uses facility of local network to control the mission data for aerial vehicles. The architecture also uses an abstraction layer that facilitates the communication based on the web services [15].

Our proposal combines both networks 2.4 GHz and 5.8 GHz. In addition, we develop a very cheap application able to work on any smartphone. Moreover, because the system sends the images in real time, it provides a better coordination of rescue services than many of the others.

3. System Description

The network architecture for information operations in UAV autonomous teams must support communications at the intra-team, inter-team and global internetworking levels, with individual UAVs acting as network nodes at all levels. In this section, we describe the main parts that form our rescue system, the application operation and the algorithm used by our system to detect the injured climbers.

Figure 1 shows the communication architecture implemented in our communication system. Our system is based on several entities: network adapters, UAVs and a Ground Station Control, software and frameworks oriented to provide enough communication quality. Finally, an Android application is used to gather the data from sensor of the smartphones.

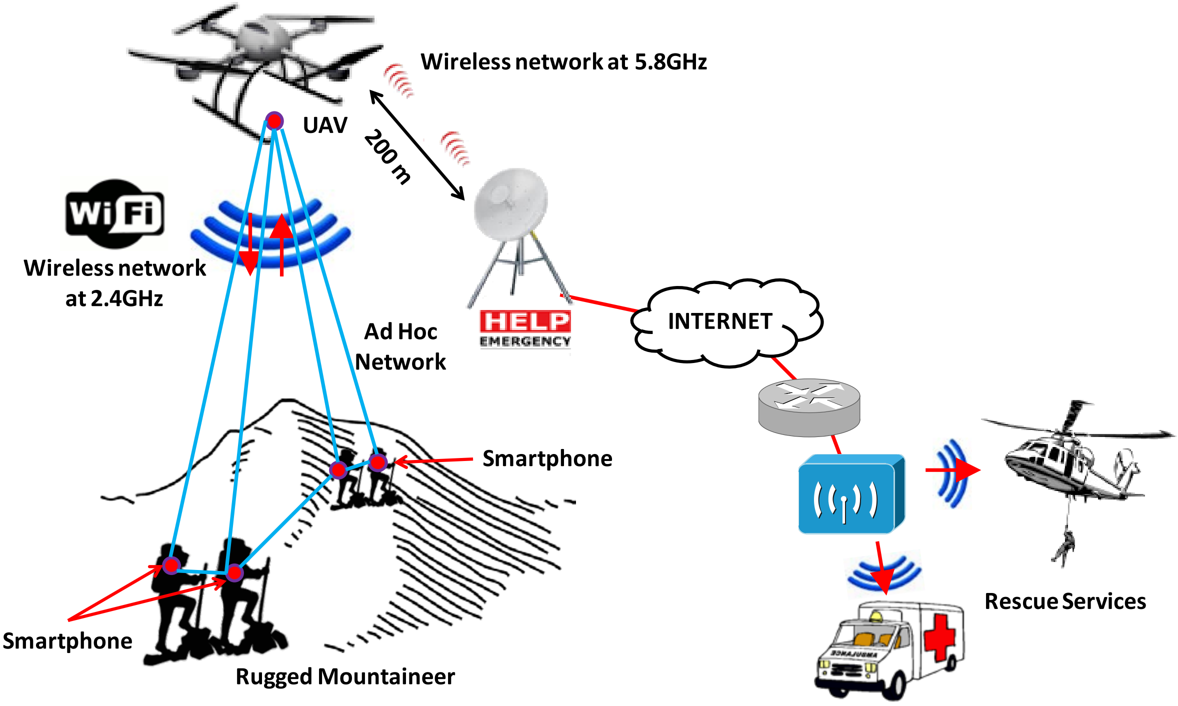


Figure 1. Network scheme communication in high mountains environment

The system uses 2 wireless networks. The first one is an ad hoc network that operates at 2.4 GHz and allows the communication between UAVs. This ad hoc network connects different mobile nodes with the UAVs. The mobile nodes are smartphones with an Android OS application. The application is responsible for taking data from the user's GPS position, which are sent them through the UAV Wi-Fi network using Wi-Fi Direct [16]. Wi-Fi Direct is an emerging technology meant to meet an easy and portable wireless network connectivity. It is peer-to-peer wireless networking. The most significant difference between traditional ad-hoc wireless networking and Wi-Fi Direct is the security. Wi-Fi Direct devices can also simultaneously connect to existing wireless networks. The second network operates at 5.8 GHz and it is responsible of communicating the UAV with the emergency network. The network consists of a series of antennas and servers capable of administering and managing the position data of the injured mountaineers in order to direct the UAV to this position. Depending on the condition of the injured mountaineer and their geographical position, the appropriate type of rescue service is alerted.

1.1 Components

The UAV system is mainly composed by two long range network adapters and the Flight Control board, which also includes a computing system.

ALFA Network AWUS036NH [17] is the IEEE 802.11 b/g/n Wireless USB adapter based on a Ralink Chipset with a maximum power transmission of 2000 mW. It is able to work with a maximum data transfer rate up to 150 Mbps in the 2.4 GHz frequency band. In addition, this device is compatible with MIMO (Multiple Input Multiple Output) technology and allows several methods of data encryption with 64/128-bit WEP, WPA, WPA2, TKIP, AES. AWUS036NH can be configured as ad-hoc mode to connect directly with other device or as infrastructure mode to connect to a wireless AP or router for Internet access.

ALFA Network AWUS051NH [18] is Compatible with IEEE 802.11n draft 3.0 and IEEE 802.11a/b/g wireless standards (at 2.4 GHz and 5.8 GHz frequency bands). It can also work with MIMO technology. This device can work at 150/300 Mbps in the 2.4 GHz/5.8 GHz frequency band and it is able to work at 11/54 Mbps over the IEEE 802.11a/b/g wireless technology. This device can be configured in ad-hoc mode and Infrastructure mode to connect to wireless AP or router to access Internet using different data encryption such as WEP, WPA, WPA2, TKIP and AES. Figure 2 shows both antennae.

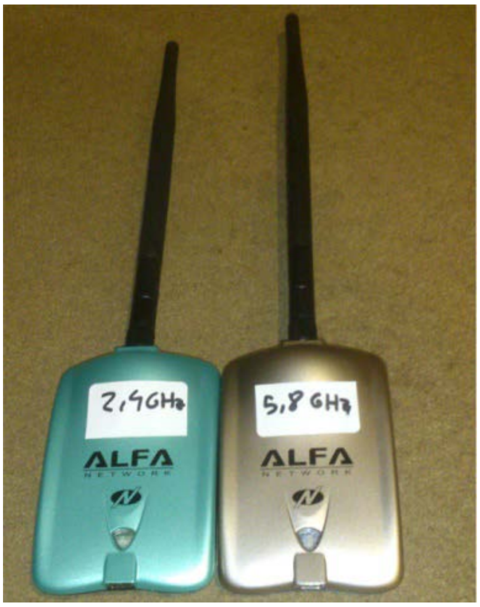


Figure 2. Long range adapters of 2,4 GHz and 5,8 GHz.

NAVIO [19] is an autopilot for raspberry Pi which does not need any other onboard controllers. It has been built for research and education. Its applications can be developed in both C++ and Python. Raspberry Pi and NAVIO are suitable for real time analog video transmission through a wireless key or 3G USB-adapter. This device is able to compress and relay video from a camera First-Person View (FPV). FPV is also known as remote-person view (RPV), or simply video piloting. This method is used to control a radio-controlled vehicle from the driver or pilot's view point. FPV is commonly used to pilot an UAV or a radio-controlled aircraft. NAVIO is equipped with a GPS chip and a RC layer to control servos of all types of platforms: cars, boats, submarines, multi-rotors, planes. Finally, it contains some sensors such as a barometer, an accelerometer or a gyroscope, among others. Figure 3 shows the NAVIO + Raspberry Pi mounted on the UAV.

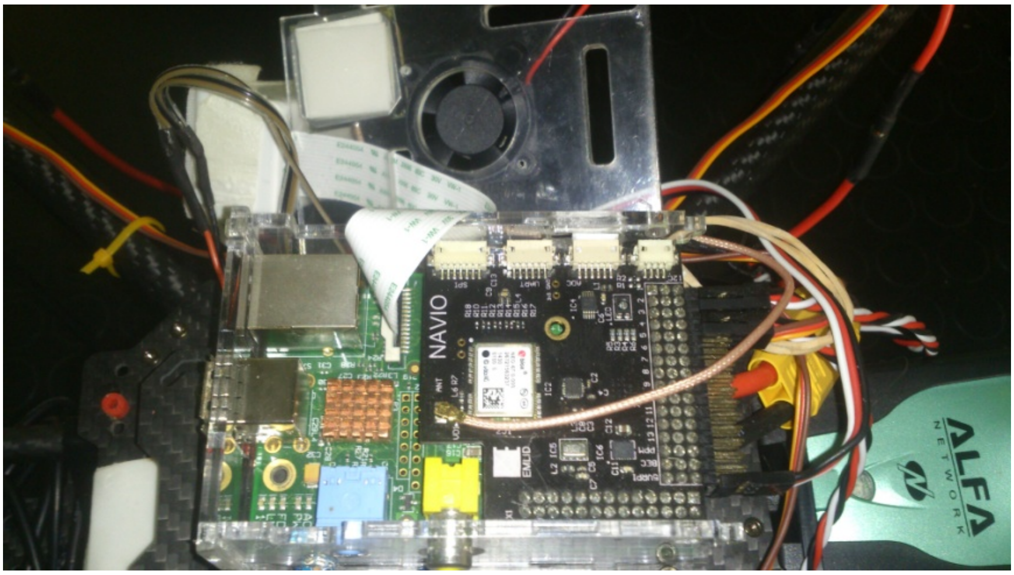


Figure 3. UAV Control board NAVIO/Raspberry Pi

The device is mounted inside a dome (See Figure 4) that protects the device from impacts. It also shows the connection of the wireless cards with the control board.

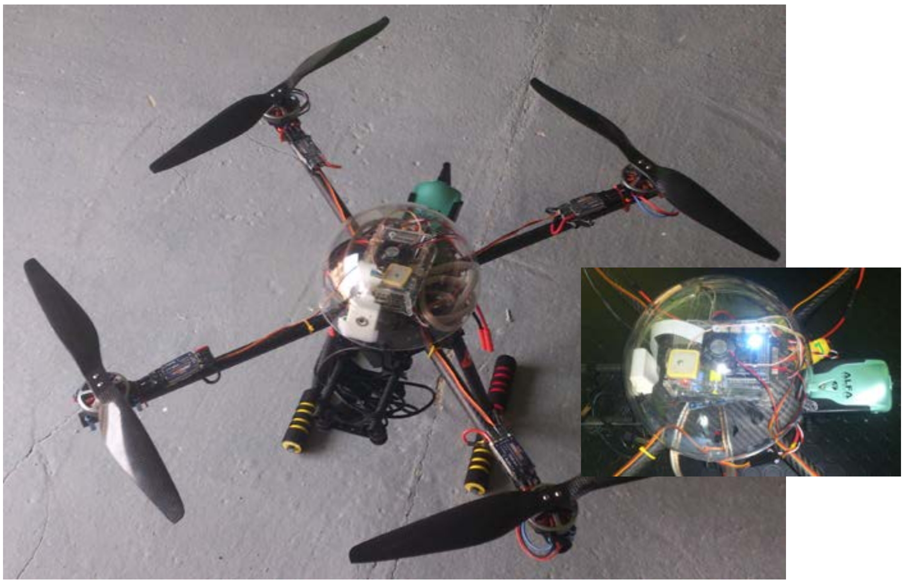
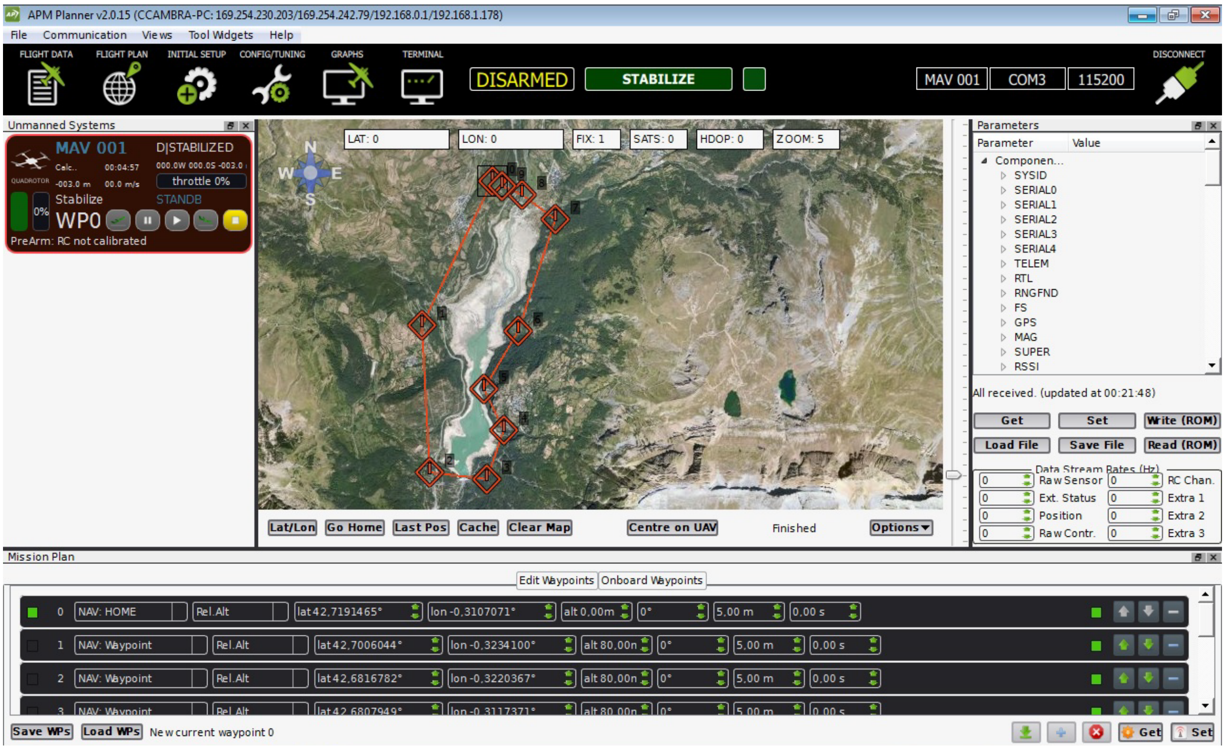


Figure 4. AUV used in our system and the NAVIO/Raspberry Pi protected with a dome

1.2 Flight Control APM

Mission Planner [20] is free and open-source software for ground control stations. This program (See Figure 5) can be redistributed and modified under the terms of the GNU General Public License. It has only been developed for Windows platforms. Mission Planner can be used as a configuration utility or as a dynamic control supplement for autonomous vehicle. Some of the most important features of Mission Planner are that it can plan, save and load autonomous missions into the autopilot using Google Maps. It allows monitoring our vehicle's status while in operation. It is also able to record telemetry logs which contain information logs and, finally, it is possible to download and analyze mission logs created by the autopilot.



APM Planner v2.0.15 (CCAMBRA-PC: 169.254.230.203/169.254.242.79/192.168.0.1/192.168.1.178)

DISARMED STABILIZE MAV 001 COM3 115200

MAV 001 D|STABILIZED
Calc.. 00:04:57 000.0W 000.0S -003.0
-003.0 m 00.0 m/s throttle 0%
Stabilize STANDB
0% WP0
PreArm: RC not calibrated

LAT: 0 LON: 0 FIX: 1 SATS: 0 HDOP: 0 ZOOM: 5

Finished

All received. (updated at 00:21:48)

Data Stream Rates (Hz)

0	Raw Sensor	0	RC Chan.
0	Ext. Status	0	Extra 1
0	Position	0	Extra 2
0	Raw Contr.	0	Extra 3

0	NAV: HOME	Rel.Alt	lat 42,7191465°	lon -0,3107071°	alt 0,00m	0°	5,00 m	0,00 s
1	NAV: Waypoint	Rel.Alt	lat 42,7006044°	lon -0,3234100°	alt 80,00n	0°	5,00 m	0,00 s
2	NAV: Waypoint	Rel.Alt	lat 42,6816782°	lon -0,3220367°	alt 80,00n	0°	5,00 m	0,00 s
3	NAV: Waypoint	Rel.Alt	lat 42,6807949°	lon -0,3117371°	alt 80,00n	0°	5,00 m	0,00 s

New current waypoint 0

Figure 5. APM Mission Planner with real values of our UAV.

We use the APM for monitoring autonomous missions, dynamic control values and dynamic waypoints. It takes information from the ad hoc signal range, FPV and video streaming.

1.3 Android App

Rescue System Android App is programmed to use several smartphone sensors to gather information about the status of the mountaineers. The GPS module and accelerometer sensor are the most powerful in this system. If accelerometer sensor speed up very quickly (a_max value >25, which represent a fast acceleration), we can estimate that the mountaineer is fallen. In this situation, sensor data are packed in a message to send via TCP socket, when UAV is in area of coverage, if not, the packet is waiting to an available ad hoc connection. After connecting, the message is transmitted to the UAV. It is also possible to record the video and audio by the smartphone in order to get information about the fall. In this case, the video is transmitter by UDP. Figure 7 shows the algorithm of our application.

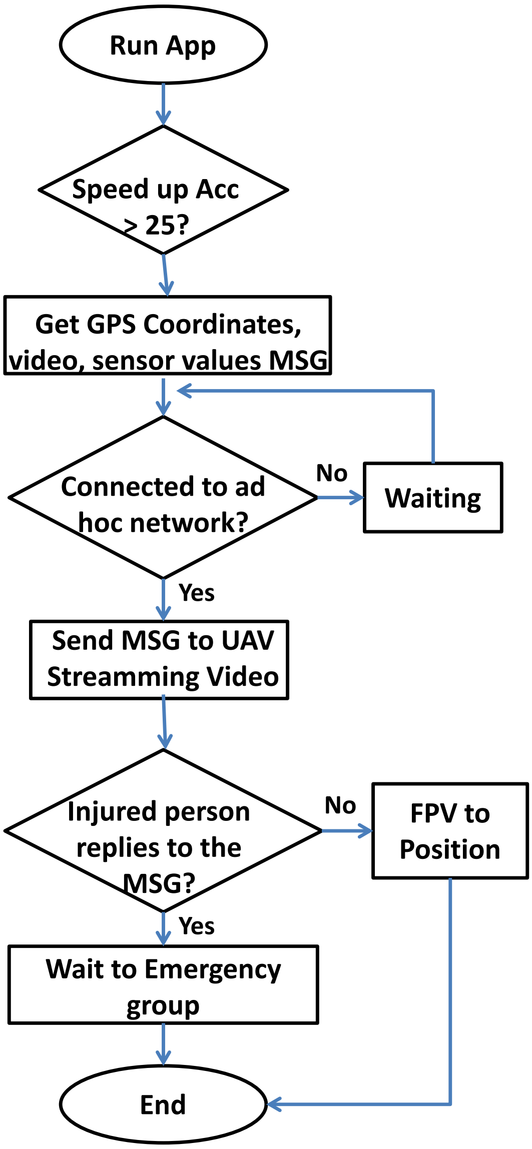
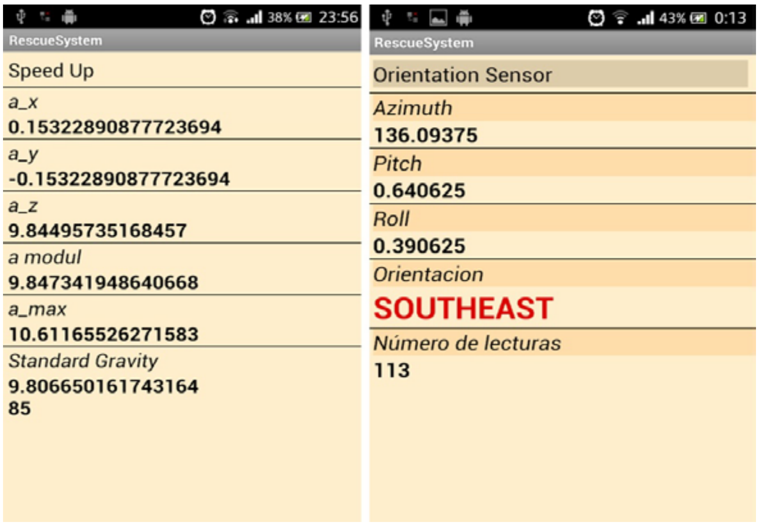

Run App
Speed up Acc > 25?
Get GPS Coordinates, video, sensor values MSG
Connected to ad hoc network?
No
Waiting
Yes
Send MSG to UAV Streamming Video
Injured person replies to the MSG?
No
FPV to Position
Yes
Wait to Emergency group
End

Figure 7. Pseudo code Rescue System

First of all, it is needed an Android smartphone with root privilege. Android Policy does not allow the use of ad hoc network with user privileges. A similar to ad hoc network, but with a single connection is Wi-Fi Direct. Google implements this kind of network to connect Android smartphones, TV or Tablet devices without infrastructure.

As Figure 7 shows, the Android application is in charge of gathering and showing the orientation values of the sensor and the changes in the device speed.



RescueSystem

Speed Up

a_x
0.15322890877723694

a_y
-0.15322890877723694

a_z
9.84495735168457

a modul
9.847341948640668

a_max
10.61165526271583

Standard Gravity
9.806650161743164
85

RescueSystem

Orientation Sensor

Azimuth
136.09375

Pitch
0.640625

Roll
0.390625

Orientacion
SOUTHEAST

Número de lecturas
113

Figure 7. Rescue System Android APP with Sensor values

1.4 Video transmission

In order to register the images, the system presents two cameras. The first one is the front camera which is embedded in the FPV platform of the Raspberry-pi [21]. This camera records images from the front of the UAV. The second camera is an auxiliary camera that can optionally be used. In general, users can use a Sport Cam as sjcam4000 [22]. This camera is connected by USB to the Raspberry-Pi and it is used to take images of the zone under the UAV.

The video transmission is performed by using the Real Time Streaming Protocol (RTSP) that uses the Real-Time Transport Protocol (RTP) sessions. RTP is an IP-based protocol providing support for the transport of real-time data such as video and audio streams. It includes services such as time reconstruction, loss detection, security and content identification. RTP was initially developed for real-time multicast transmission. However, it can also be used in unicast content distribution. It can be used for one-way transport such as video-on-demand as well as interactive services such as Internet telephony. RTSP is a network control protocol designed to be used in entertainment and communication systems to control streaming media servers. RTSP establishes and controls streams of continuous audio and video media between the media servers and the clients. The transmission of streaming data itself is not a task of the RTSP protocol. Most RTSP servers use the RTP in conjunction with Real-time Control Protocol (RTCP) for media stream delivery. However, some vendors and developers implement proprietary transport protocols.

In RTSP, each presentation and media stream is identified by an RTSP URL. In contrast to HTTP, RTSP server has to maintain "session states" in order to correlate RTSP requests with a stream while HTTP is intended as a stateless protocol.

Figure 8 shows the packet exchange when a user requests the UAV to visualize the current video. As we can see in Figure 8, when the user makes a request, this request is performed through HTTP services. After that, the user sends to the UAV a SETUP packet where the user asks to allocate resources for a stream and start a RTSP session. When the session starts, the user asks (using a PLAY packet) the UAV to start sending data on a stream allocated via SETUP. In this moment, the video and audio content are sent using RTP packets and the control process is managed by the RTCP protocol.

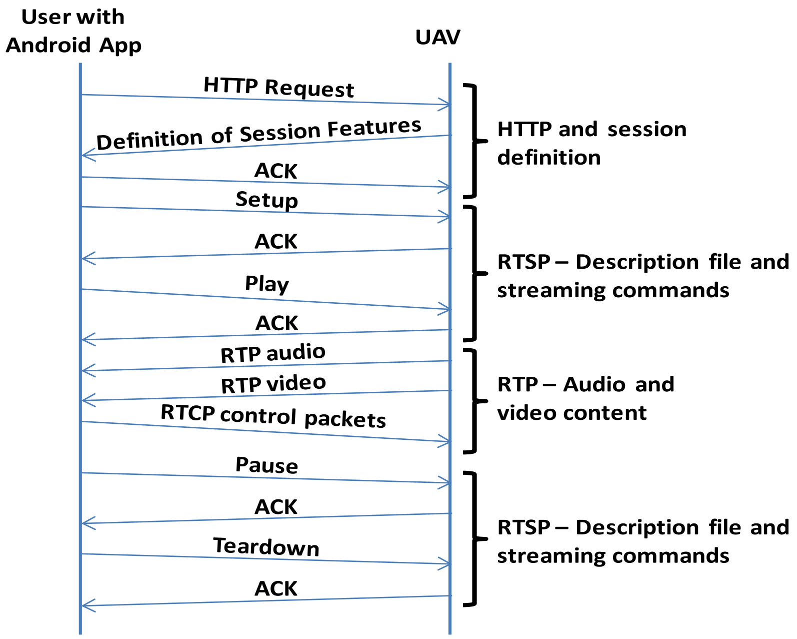

User with Android App
UAV
HTTP Request
Definition of Session Features
ACK
Setup
ACK
Play
ACK
RTP audio
RTP video
RTCP control packets
Pause
ACK
Teardown
ACK
HTTP and session definition
RTSP – Description file and streaming commands
RTP – Audio and video content
RTSP – Description file and streaming commands

Figure 8. Packets exchange between User and UAV

The user can temporarily halt the stream delivery without freeing server resources using PAUSE packets. Finally, when the whole content has been sent, the user sends a TEARDOWN packet to ask the UAV to stop the delivery of the specified stream and release the resources associated with it.

1.5 Routing Protocol used in our system

We have chosen OLSR [23] for the routing protocol of our system. It is a proactive protocol, i.e., it nodes to exchange periodically information in order to constantly update their routing tables. OLSR is link-state based protocol. The disadvantage of OLSR is that it requires an additional network load due to the periodic transmission of control messages. Therefore, it is used a technique called MPR (Multi-Point Relay). When a node receives a control packet, it usually retransmits the data to the neighboring nodes. Thus, there is a phenomenon called flood (flooding), where all nodes receive the packets. However, each one receives the same packet several times from different neighbors, generating an overhead, i.e., the network function overloaded with redundant information. In OLSR, the number of nodes that relay packets will be limited. Among the network nodes, there will be those called MPR. The choice of each MPR is made by a consensus of its neighboring nodes located to a hop. So, when the information should be updated on the network, packets sent by a node will reach all its neighbors, but only those called MPR can relay the information below. This process is repeated with the next nodes that receive the packets. Thus, every node will receive the information only once, that is, no nodes will receive the packet more than once. OLSR protocol is an organized and efficient way to manage the control packet traffic between two nodes, always seeking the shortest path. This is done using an algorithm. It is characterized by the following features:

- Routes are set up based on continuous traffic
- All routes are maintained all the time

As a pro, routes are always available, but as a con, there is constant overhead created by the control traffic, although this problem can be partially solved by using the multipoint relaying method.

Although other protocols could be used in our application, the OLSR protocol is best suited to the requirements of our system.

4. Test Bench

In order to check the network performance of our system, we have performed a test bench during 4 minutes. While the UAV is flying, it sends the images in real-time (See Figure 9). The network traffic is captured using Wireshark [24]. In this section, we are going to show the results of this test bench.

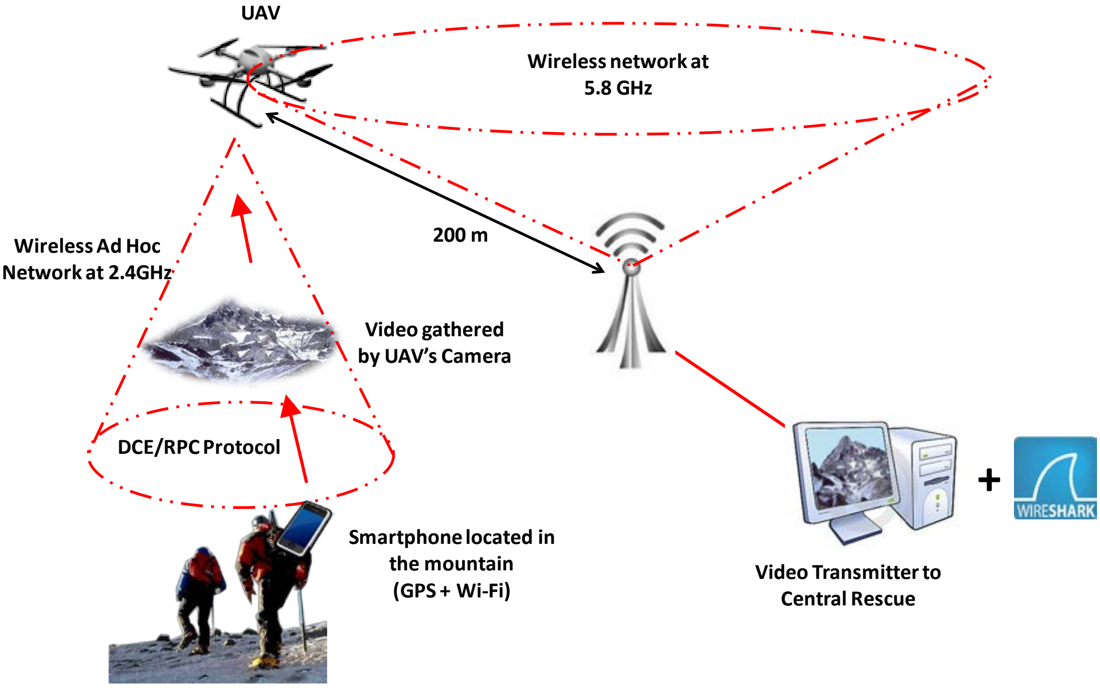

UAV
Wireless network at 5.8 GHz
200 m
Wireless Ad Hoc Network at 2.4GHz
Video gathered by UAV's Camera
DCE/RPC Protocol
Smartphone located in the mountain (GPS + Wi-Fi)
Video Transmitter to Central Rescue
+
WIRESHARK

Figure 9. Architecture developed for performing our test bench

To understand the sort of packets that we detect, we should know that the Distributed Computing Environment/Remote Procedure Calls (DCE/RPC) is a tool for calling a procedure on a remote machine as if it were a local procedure call. For an application, a remote call looks like a local call, but there are several RPC components that work together to implement this utility. The system supports two RPC protocol implementations. The first

RPC protocol works over connection-oriented transports such as TCP/IP and the other RPC protocol operates over connectionless transports such as the UDP/IP. The DCE components can work as client/server applications. The RCP procedure is transparent to the end user. The application programmer is the persons who will be into contact with the RPC components. Figure 10 shows the number of packets (as a function of the elapsed time) generated by the DCE/RPC Protocol transmitting message from Android App to UAV through the ad hoc network. As we can see, the most part of these packets are located at the beginning of the transmission (up to 26th second). During this period, the average number of packets is 8.44 packets/s with a standard deviation of 19.16. The maximum number of received packets during this time is 83 packets/s

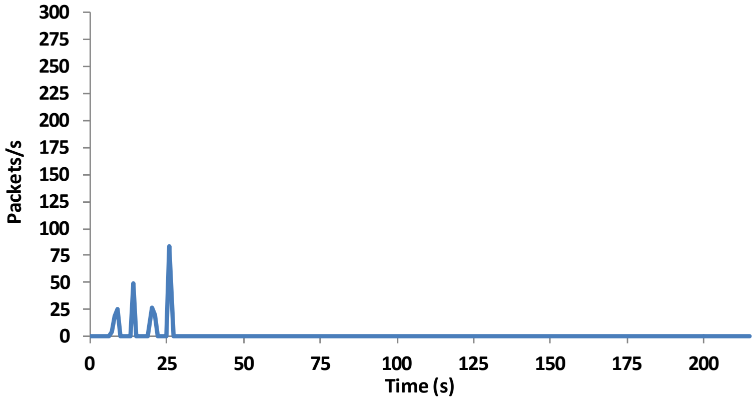


Figure 10. Packets generated by DCE/RPC Protocol transmitting message from Android App to UAV through an ad hoc network.

Figure 11 shows the number of packets/s generated by OLSR. As we can see, the number of packets generated is around 2 packets/s with a standard deviation of 0.72. This flow is maintained for as long as we are working with mobile nodes that constantly change their position because the network needs to update the information about routing tables. The maximum value of received packets is 3 packets/s (is located at 102th and at 110th seconds). The system has exchanged a total number of 175 packets during 216 seconds.

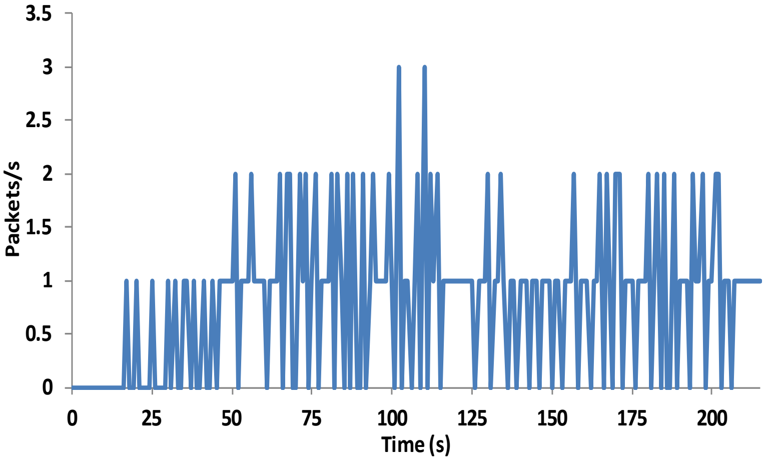


Figure 11. Number of OLSR packets sent by the routing protocol as a function of the time.

Figure 12 shows the number of TCP packets exchanged through the network. In this case, we have detected significant values of packets at the 120th second, instant where the systems starts the transmission and at the 162th second, in which the video deliver stops. The average traffic flow is 0.13 packets/s with a standard deviation of 1.07. The maximum values related to 120th and 162th seconds are 14 packets/s and 5 packets/s respectively. The number of TCP packets transmitted during the simulation is 28.

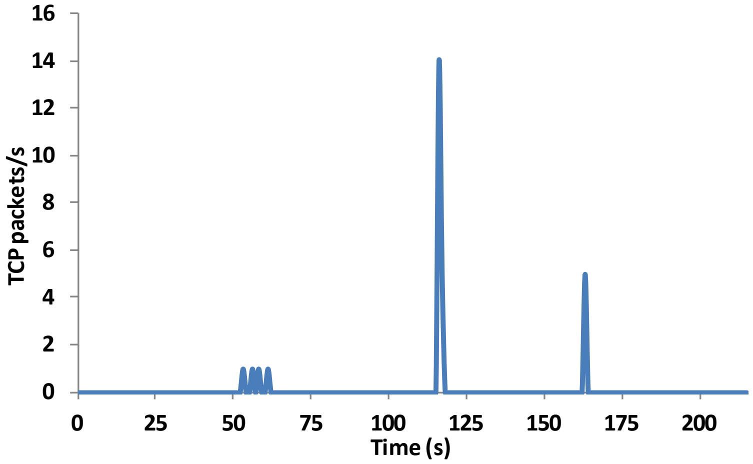


Figure 12. Number of TCP packets as a function of time

Figure 13 shows the results of this test bench in regards to the number of UDP packets exchanged in the network. The average number of packets is 22.62 packets/s during the entire simulation. However, the most of the packets are located between the 116th and 163th seconds. During this period, the video is transmitted. The average number of packets during the video transmission is about 89.90 packets/s. The average traffic during the period where no video is transmitted (from 0 to 115 and from 164 to 215) is about 3 packets/s.

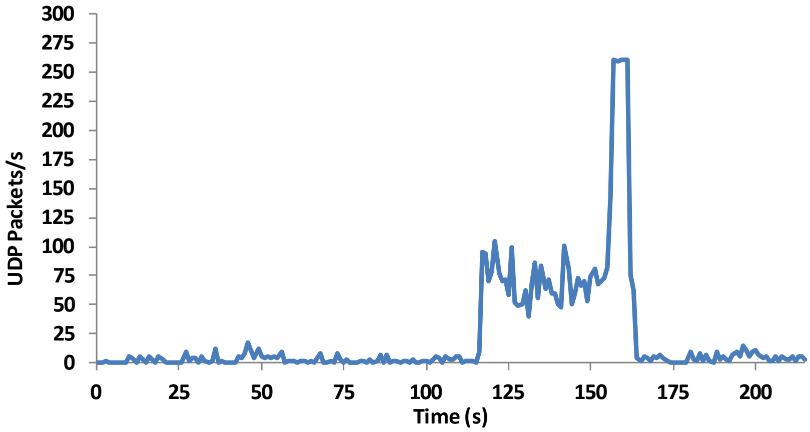


Figure 13. Number of UDP packets as a function of time.

Figure 14 shows the RTP packets as a function of time. These packets are related to the video content transmission. The video transmission starts in the second 116th and finish at the second 163th. During the video transmission, there is an average amount of packets of 87.52

packets/s. During the time that no video is transmitted, the system does not register any packet.

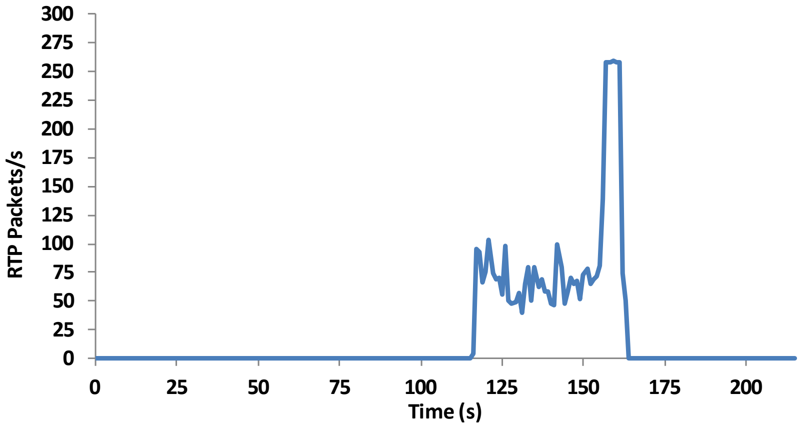


Figure 14. RTP video streaming packets as a function of time

Finally, it is important to estimate the quality of the received video. To know this, we measured different parameters, i.e., the bandwidth consumed by the RTP data flow, the jitter and the delay. As we can see in Figure 15, the average bandwidth registered during this test was around 490 kbps. The maximum bandwidth value is 1089.22 kbps.

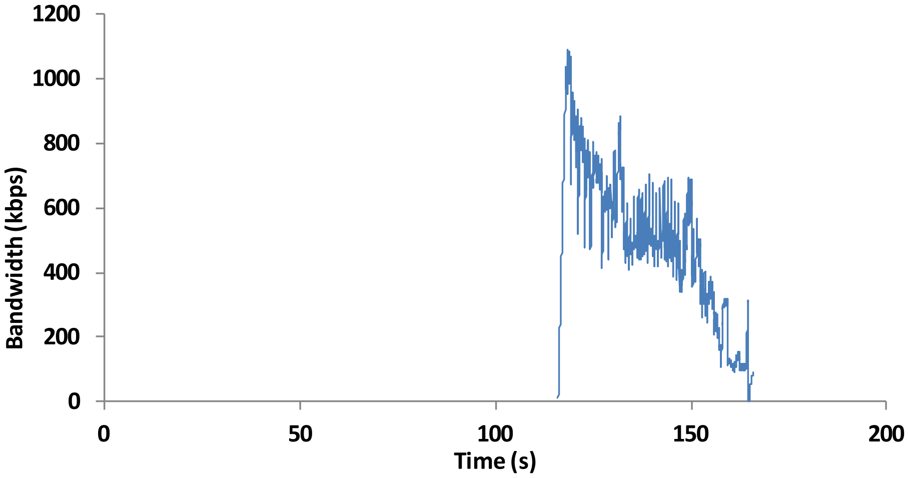


Figure 15. Consumed bandwidth in kbps during the video transmission.

If we analyze Figure 16 and Figure 17, which show the delay and jitter in ms., respectively, we can see that the biggest delay and jitter is located at the beginning of the data transmission. This is because the first seconds are needed to fully buffer the video player.

On the one hand, we can see that the maximum value of delay is about 2450 ms., while on the other hand, the maximum jitter is about 360 ms.

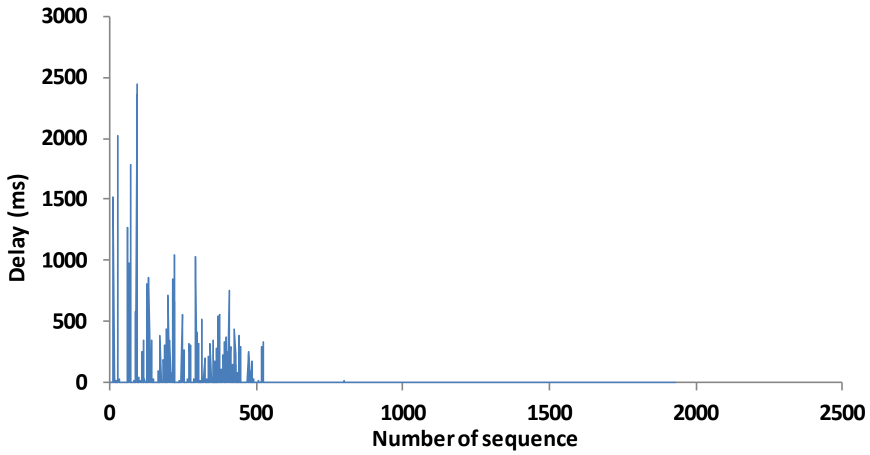


Figure 16. Delay registered in video frames

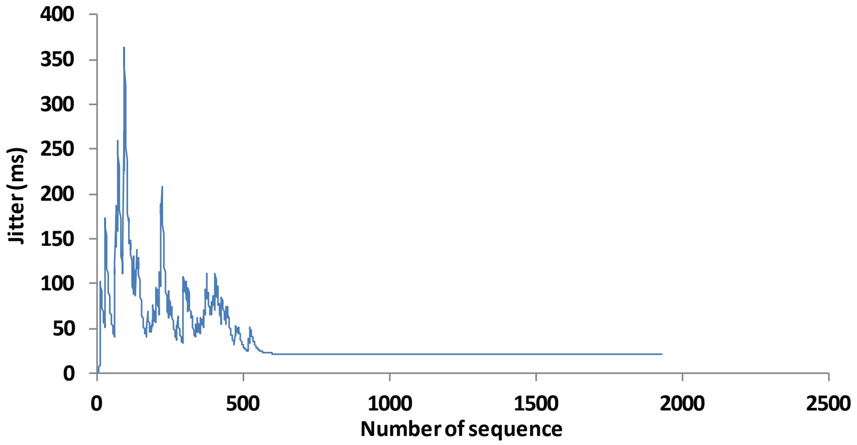


Figure 17. Jitter registered in video frames

The rest of time the delay is near to 0 ms while the jitter is around 40 ms at sequence 500.

5. Conclusion and Future Work

The use of rescue systems will revolution the field of mountain emergency rescues over the air and will make possible some tasks that require high quality images. They will allow higher image processing techniques with better algorithms.

In this paper, we have proposed and developed a system based on an UAV where the ad hoc network technology and UAV are used as a rescue system by using an Android App client node. The proposal has been developed to help emergency services in situations where injured mountaineers are located in special hostile environments. Our system is based on a quadcopter provided with two wireless interfaces of 5.8 GHz and 2.4 GHz, and a GPS receiver. The remote control is made of a NAVIO/Raspberry Pi control board, while the selected control software is the APM Mission Planner.

The test bench was performed in a mountain scenario, working at 200 meters in line to the UAV. Our analysis of the sensors values and OLSR routing protocol demonstrates that they can be used in mobile communication nodes which exchange information while they are moving and using a GPS device.

The main aim of our proposal was focused on the rescue tasks but this system could be used to detect and analyze fire and areas with risk of fire. It would help to control critical weeds areas and cereals in agricultural productions.

As a future work, we are planning to include a dynamic network reservation system providing BoD (Bandwidth on Demand). This is a specification of P2P protocols and allows connecting new devices to the network to exchange information by using UDP and TCP protocols. This protocol could be included in the aerial drones with high level of movement. Moreover, we will add devices authentication and data encryption in order to preserve the privacy of the rescued people.

Acknowledgement

This work has been supported by the "Ministerio de Economía y Competitividad", through the "Convocatoria 2014. Proyectos I+D - Programa Estatal de Investigación Científica y Técnica de Excelencia" in the "Subprograma Estatal de Generación de Conocimiento", (Project Reference: TIN2014-57991-C3-1-P), by the postdoctoral grant "Contratos Postdoctorales UPV 2014 (PAID-10-14)" by the"Universitat Politècnica de València", by the pre-doctoral student grant "Ayudas para contratos predoctorales de Formación del Profesorado Universitario FPU (Convocatoria 2014)" Reference: FPU14/02953 by the "Ministerio de Educación, Cultura y Deporte".

References

[1] Lacuesta, R., Lloret, J., Sendra, S., and Peñalver, L. "Spontaneous Ad Hoc Mobile Cloud Computing Network", The Scientific World Journal, 2014. Vol. 2014, Article ID 232419, 19 pages. http://dx.doi.org/10.1155/2014/232419

[2] Lacuesta, R., García, M., Lloret, J., and Palacios, G., "Study and Performance of Ad Hoc Routing Protocols", chapter 4 of the book Mobile Ad hoc Networks: Current Status and Future Trends, CRC Press, Pp. 71-101. 2011.

[3] Tarique, M., Hossain, A., Islam, R., and Hossain, C. A., "Issues of Long-Hop and Short-Hop Routing in Mobile Ad Hoc Networks: A Comprehensive Study", Network Protocols and Algorithms, Vol 2, No 2. Pp. 107-131. 2010. http://dx.doi.org/10.5296/npa.v2i2.430

[4] Lloret, J., Garcia, M., Bri, D., Diaz, J. R., A cluster-based architecture to structure the topology of parallel wireless sensor networks, Sensors 9 (12), Pp. 10513-10544, 2009.

[5] Lloret, J., Lacuesta, R., Shu, L., and Chen, M. "User-Oriented and Service-Oriented Spontaneous Ad Hoc and Sensor Wireless Networks", Ad Hoc & Sensor Wireless Networks, Vol. 14, Issue 1-2, Pp. 1-8, January 2012.

[6] Atto, M., and Guy, C., “Routing Protocols and Quality of Services for Security Based Applications Using Wireless Video Sensor Networks”, Network Protocols and Algorithms, 2014, Vol. 6, No. 3, pp.119-137. http://dx.doi.org/10.5296/npa.v6i3.5802

[7] Diaz, J. R., Lloret, J., Jimenez, J. M., and Rodrigues, J. J., “A QoS-Based Wireless Multimedia Sensor Cluster Protocol”, International Journal of Distributed Sensor Networks, Vol. 2014, Article ID 480372, 17 pages. http://dx.doi.org/10.1155/2014/480372

[8] Sanchez-Iborra, R., and Cano, M. D., “An approach to a cross layer-based QoE improvement for MANET routing protocols”. Network Protocols and Algorithms, ,Vol. 6, No.3, pp.18-34. 2014. http://dx.doi.org/10.5296/npa.v6i3.5827

[9] Flushing, E. F., Gambardella, L., and Di Caro, G. A., “GIS-based mission support system for wilderness search and rescue with heterogeneous agents”. In Proc. of the 2nd Workshop on Robots and Sensors integration in future rescue INformation system (ROSIN-2012), October 7-12, 2012. Vila Moura- Algarve (Portugal).

[10] EL Mamoune, S., Ezziyyani, M., Lloret, J., Towards a New Approach for Modelling Interactive Real Time Systems Based on Collaborative Decisions Network, Network Protocols and Algorithms, Vol 7, No 1. Pp. 42-63. 2015.

[11] Wang, L. L., Zhou, W. P., and Zhao, S. L., “Application of Mini-UAV in Emergency Rescue of Major Accidents of Hazardous Chemicals”. In 2013 the International Conference on Remote Sensing, Environment and Transportation Engineering (RSETE 2013). July 26-28, 2013. Nanjing, China. Atlantis Press. Pp.152-155. http://dx.doi.org/10.2991/rsete.2013.38

[12] Kuiper, E., and Nadjm-Tehrani, S, “Mobility models for uav group reconnaissance applications”, in Proceedings 2006 International Conference on of Wireless and Mobile Communications (ICWMC '06). July 29-31, 2006, Bucharest, Romania. http://dx.doi.org/10.1109/ICWMC.2006.63

[13] Musolesi, M., and Mascolo, C., “Acommunity based mobility model for ad hoc network research”, in Proceedings of the 2nd international worshop on Multi-hop ad hoc networks, theory of reality (REALMAN ’06). May 26, 2006, Florence, Italy. Pp 31-38.2006. http://dx.doi.org/10.1145/1132983.1132990

[14] Rémy, G., Senouci, S. M., Jan, F., and Gourhant, Y., “SAR.Drones: Drones for Advanced Search and Rescue Missions”, in Les Journées Nationales des Communications dans les Transports (JNCT 2013), Nevers, France, May 29- 30, 2013.

[15] Pastor, E., Lopez, J., and Royo, P., “A hardware/Software architecture for uav pauload and mission control”, In Proc. of the IEEE/AIAA 25th Digital Avionics Systems Conference, 2006, October 15-19, 2006. Portland, Oregon, Pp. 1-8. http://dx.doi.org/10.1109/DASC.2006.313738

[16] Conti, M.; Delmastro, F.; Minutiello, G.; Paris, R., "Experimenting opportunistic networks with WiFi Direct," in 2013 IFIP Wireless Days (WD 2013), pp.1-6, 13-15 Nov. 2013, Valencia. doi: 10.1109/WD.2013.6686501

[17] Features of AWUS036NH- 802.11b/g/n Wireless USB adapter. In ALFA NETWORK Web site. Available at: http://www.alfa.com.tw/products_show.php?pc=34&ps=21 [Last access: December 15, 2015]

[18] Features of AWUS051NH-802.11a/b/g/n Wireless USB adapter, In ALFA NETWORK Web site. Available at: http://www.alfa.com.tw/products_show.php?pc=67&ps=23 [Last

access: December 15, 2015]

[19] Features of Navio + Raspberry PI Control Board. In DIY DRONES Web site. Available at: http://diydrones.com/profiles/blogs/navio-raspberry-pi-autopilot [Last access: December 15, 2015]

[20] APM Planner Ground Control Station. Available at http://planner.ardupilot.com/wiki/mission-planner overview/#what_is_mission_planner [Last access: December 15, 2015]

[21] Features of Raspberry-pi camera. Available at Raspberry website: https://www.raspberrypi.org/documentation/hardware/camera.md [Last access: December 18, 2015]

[22] Features of SJCAM4000 Sport camera. Available at SJCAM website: http://www.sjcam.com/cameras/9-sj4000-full-hd-1080p-waterproof-action-camera-sport-dvr.html [Last access: December 18, 2015]

[23] Clausen, T., and Jacquet, P., RFC 3626: Optimized Link State Routing Protocol (OLSR). October 2003. In IETF Web site. Available at: https://www.ietf.org/rfc/rfc3626.txt [Last access: December 15, 2015]

[24] Wirehark, Network Analyzer. available on-line at http://wireshark.org [Last access: December 15, 2015]

Copyright Disclaimer