A platform for evaluation of the unmanned aerial vehicles telemetry protocols and their parameters

Svilen Borisov, Jordan Raychev, Georgi Hristov, Ivan Beloev, Plamen Zahariev

This article aims to present an analysis on the modern methods and protocols for transmission of telemetry data to a remote location, as well as to provide a general overview of the telemetry technologies used for this purpose. In order to achieve this objective, the main emphasis of the paper is laid on the open source telemetry protocols, mainly because they provide means for in-depth analysis and also access to their own internal structure. For the purpose of this paper, we provide a comparative analysis between two such telemetry protocols - MavLink and DNP3. It is found that the MavLink protocol has lower overhead (additional added information) compared to the DNP3, and because of this, the MavLink protocol is more suitable for further investigations. For the purpose of our experiments a platform for evaluation of the characteristics and the parameters of the MavLink protocol is proposed. The main objective of this article is to investigate the possibility for modification and improvement of the structure of the protocol, depending on the specific needs of the user.

Платформа за оценка на параметрите на протоколи за предаване на телеметрични данни от безпилотни летателни апарати. (Свилен М. Борисов, Йордан П. Райчев, Георги В. Христов, Иван Х. Белоев, Пламен З. Захариев). *Настоящата статия има за цел да анализира съвременните методи и протоколи за предаване на телеметрична информация на отдалечено разстояние, както и да предостави общ поглед върху използваната технология – телеметрия. За постигането на тази цел е акцентирано върху протоколи с отворен код, поради факта, че те предоставят възможност за по-задълбочен анализ и достъп до тяхната структура. Вследствие на това е извършен сравнителен анализ между протоколите MavLink и DNP3. Получените резултати ясно показват, че допълнителната служебна информация, която се добавя от MavLink протокола, е няколко пъти по-малка от тази при DNP3. Поради тази причина MavLink протокола е избран за по-нататъшни изследвания. За целите на нашият експеримент е предложена платформа за изследване характеристиките на протокола и неговите параметри. Основната цел на текущата статия е изследването на възможностите за модифициране и подобряване структурата на протокола в зависимост от специфичните нужди на потребителя.*

Introduction

Telemetry may be defined as an automated process, by which measurements and data (speed, pressure, acceleration, velocity, etc.) are collected at a remote station, in most cases not easily accessible, and then transmitted to a base station for post analysis and visualization. Although the term telemetry is often referred to as a wireless data transfer, the transmission medium can also be a wired one (e.g., computer networks, optical media and other wired communication carriers). Most of the systems for telemetry data transmission are capable of collecting information from different types of sensors. That requires all measurements to be grouped into format suitable for transmission through a single data stream. In the receiving side the information is converted to its original form and subsequently analyzed.

The unmanned aerial vehicles, also known as UAVs, are becoming more affordable, easy to operate and thus can be used for different applications. This is possible because every single UAV is equipped with some sort of telemetry system, which is used for the transmission and monitoring of the vital for the aircraft information. Due to this reason, the telemetry systems are inseparable part of every unmanned aerial vehicle, and are providing better and safer flights.

General structure

The remote data transmission using telemetry systems is providing an opportunity for safer monitoring of the parameters of the objects, which are located in inaccessible or hazardous places. Nowadays, there are numerous and various types of telemetry systems. Despite their diversity, all telemetry systems for data transmission have many common characteristics [1]. Fig. 1 presents the main components of one such system. Often, the system is divided in two parts – a remote station/unit (UAV) and a base station (Ground Control Station - GCS). The data collected from the sensors at the remote station is converted into single digital data stream, which is then transmitted to the base station. In the receiving side, the information is converted back to its original form and stored for future analysis. Even though there are plenty of methods for data transmission, the most preferred and efficient way is to use the electromagnetic waves. Some of their advantages are:

- The system is more robust, due to the lack of cable transmission media;
- Reduced system cost;
- Wide range of operation condition;
- High degree of portability.

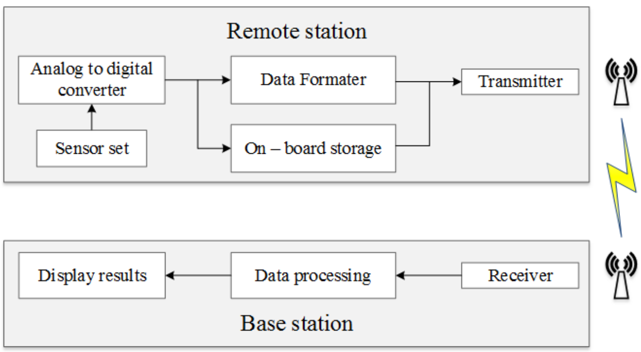

Fig.1. Overview of a system for telemetry data transmission.

At the remote station, a group of sensors (temperature sensor, atmospheric pressure sensor, magnetometer, etc.) is used to collect and measure information about the various system parameters. The data from these sensors is converted into digital form and then multiplexed into a single data stream. The newly formed stream is then modulated and transmitted to the base station. The received stream is then processed and the information is converted to its original form.

In many cases, the base station sends a request to the remote station. The received request indicates that the remote station should start transmitting data back to the base station. By doing that the two stations are switching their modes (base station becomes receiver and the remote station becomes transmitter). After the desired information is collected the stations switch back their modes and wait for further instructions [2], [3] and [4].

Data types

There are several data types, which play essential role for the communication process between the UAV and the ground control station. Every data type has its own specific requirements about the transmission channel – delay, bandwidth, bitrate, channel direction and others. The data types are flight control data, flight status data, payload data and payload control data. All data types and their aforementioned requirements for the communication channel are summarized in Table 1.

Table 1

Data types and channel requirements

Data type	Direction	Data rate	Frequency
Flight control	up	< 30 kbps	1 – 2400 MHz
Payload control	up	> 1 Mbps	1 – 1000 MHz
Flight status	two way	> 1 Mbps	30 – 1000 MHz
Payload data	two way	< 30 kbps	1 – 15 GHz

The **flight control data** is transmitted from the ground station towards the UAV. It usually consists of packets, containing various commands for controlling the remote aircraft. This is mainly accomplished by controlling the UAVs engines and motors, their rotations per minute and etc. Other flight control commands might include commands for changing the positions of the elevators, the position of the flaps, commands for extending or retracting the landing gears and other. All commands are sent from the ground control station to the UAV through a low speed radio uplink, which usually provides speeds of up to 30 Kbps. The flight control data is extremely time sensitive, but its volume is very low and thus represents no challenge for the modern communication interfaces and standards. A very interesting fact is that this type of data is not critical for the mission. Even though it is always good to have means for remote control of the UAV, in the cases when the uplink fails, the aircraft can still continue with its mission. This is possible when the UAV is capable of detecting the failed link and uses its autopilot feature. This feature requires more computational power and can drain the batteries of the UAV more quickly, but can also save the drone when the control uplink fails [5].

The **flight status data** (also known as telemetry data) is used for evaluation of the parameters of the UAV. There are several categories of flight status data based on the observed parameters – UAV location data, battery condition data and spatial positioning data. The UAV location data is used for presenting the global or the relative coordinates of the aircraft, as well as the distance to the home base. This is usually accomplished by reading and processing the data from the on-board GPS module. In the cases when there is no such component available, the received signal strength indicator (RSSI) can be used in combination with radio-based localization algorithms (like TDoA or AoA) to determine the relative position of the UAV. The battery condition data is used for determining the state of the batteries and for estimation of the possible flight time. Probably the most important flight status data types are the ones about the spatial position of the aircraft. They consist of information about the speed and the direction of the drone, its altitude and the vertical and horizontal offsets. Based on the values of the received flight status data, the operator of the GCS can make modifications to the UAV flight plan, can monitor its mission progress or can even issue flight control instructions and commands, which are to be send to the drone. Usually the flight status data is transmitted using a single direction downlink, which is capable of maintaining data rates of about 1Mbps or less. Similarly to the flight control data the flight status data is not mission critical, but provides means for more effective control of the drone [5].

The **payload data** is the only mission critical data and thus the loss of the communication channel, which is used for its transmission, can cause the failure of the mission or any of its tasks. The payload data can be defined as the data streams from the various on-board sensor devices, including the on-board cameras, the temperature and pressure sensors and all other sensing devices. Due to the huge amount of data, which has to be sent to the ground control station, the payload data is usually sent using a separate radio link, which is a broadband high speed link with speeds of over 2 Mbps. This connection is usually bidirectional. This is mainly due to the nature of the data, the need to acknowledge some of the transmitted packets, the necessity to send additional data for maintenance of the sessions between the drone and the base station, etc. Sending and receiving the payload data still presents many challenges, including how to structure the data in the best possible way, how to maintain real-time transmission, how to provide high quality video streams without overconsumption of the channel bandwidth and many other. One of the major problems concerning the transmission of video and audio data is the lack of suitable encoding standards. Currently the most widely used digital video encoding standards are the MPEG standards, but they are designed for video sequences with low dynamics and thus are unsuitable for the real time high-dynamic video streams from the on-board cameras of the UAVs [5].

The **payload control data** is sent rarely and is even sometimes not transmitted at all. This type of data is used when the on-board camera has to be controlled (usually by PTZ commands or some sort of camera control protocols), or when the on-board devices have to be turned on or off. The payload control data can be sent using the radio interface, used for the flight control and the flight status data, or using the less loaded uplink channel, which is used for acknowledging the payload data [5].

The data that is the object of this study is characterized as **Flight status data** (telemetry data) and serves mainly to inform the operator about the state of the aircraft.

The MavLink communication protocol

There are several different options for selection of the most suitable communication protocol for telemetry data transmission. Some of the protocols used nowadays are Modbus, DNP3 (Distributed Network Protocol 3) and MavLink (Micro Air Vehicle Link). Each of these protocols is optimized for its specific purposes. Modbus and DNP3 are best suited for transmission of large amounts of data, which usually leads to increase in the transmission time. MavLink is a lightweight protocol and is optimized for telemetry data transmission for unmanned aerial vehicles. The protocol was developed in 2009 by Lorenz Meier [6]. The current stable version of the protocol is 1.0, while version 2.0 is still under development. Due to its advantages the MavLink protocol has been adopted by several autopilot projects and software developers. The main advantage of this protocol is the lower amount of additional information (called ***overhead***), which is added to the user data when a packet is being sent. This increases the transmission rate, as well as the security and the integrity of the message. The MavLink protocol has a total of 8 bytes of overhead per packet, while DNP3 has 10 bytes static header overhead and 2 bytes overhead for every 16 bytes of user data.

Fig. 2 shows a comparison between the MavLink and the DNP3 overhead. As seen in the figure, the overhead of MavLink is lower by a factor of 2 to 5, compared to the overhead of DNP3, and depends on the size of the user information (bigger overhead when sending small amounts of information and lower overhead otherwise). This is because the overhead of a MavLink packet is always 1 byte for the start frame, payload length, packet sequence, system, component and message identification fields and 2 bytes for the CRC checksum. The overhead of DNP3 is 10 bytes for the header part and 2 additional bytes (CRC checksum) for every 16 bytes of user information (a total of 250 including 1 byte for the transport header).

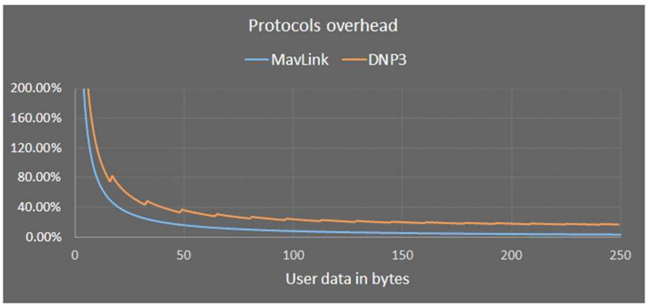

Fig. 2. Overhead of the MavLink and DNP3 protocols.

In addition to the bigger overhead, the maximum size of the DNP3 message is also larger - 292 bytes, compared to only 263 bytes for a Mavlink message, and this leads to increase of the transmission time. Some of the other advantages of the MavLink protocol may be summarized as follows:

- The open source nature of the protocol provides an opportunity for its continuous development by the MavLink community;
- The protocol is independent of the underlying computer hardware or software and does not need to be recompiled during distribution;
- Due to the low overhead, the protocol is suitable for UDP and the UART/Radio-modem transport layer;
- The efficient encoding allows the protocol to be used on various types of microcontrollers, like Arduino, Raspberry Pi, etc.;
- The open source nature of the protocol (library), allows the source code to be modified according to specific user needs.

Table 2 presents the different data fields of a typical MavLink message and provides a brief description for every one of them. The first field of the protocol frame indicates the start of a new packet. The second byte determines the length of the payload (0 – 255 bytes) and the third one represents the packet sequence. The fourth (System ID) and fifth (Component ID) bytes are used to differentiate the systems and their components, thus allowing operations of multiple UAVs on the same network. The sixth byte represents the message identification number, which determines the function of the message. The seventh field holds the payload data, which varies between 0 and 255 bytes in its length, depending on the message ID. The payload may contain variables of the following types – 8, 16, 32, 64 bit signed and unsinged integers, float, double and char. The last two bytes consist of a 16 bit checksum, which is generated by the same means as the ones in the ITU X.25 and SAE AS-4 standards. The only field suitable for modifications, and thus of interest to us, is the Payload field. The performance of the protocol, and more specifically the message rate, directly depends on the size and the value of this field (1).

$$(1)\ msg/s = \frac{Baud\ rate}{Message\ size} = \frac{Baud\ rate}{(Header+Payload+CRC)},$$

where the baud rate is the maximum theoretical transmission rate of the communication link, often referred as symbols per second, and the message size is the amount of bytes per message.

Every message consists of two parts – variable and non-variable part. The non-variable part includes the header, which is 6 bytes, and the CRC, which is 2 bytes. The payload is the variable part of the message and can include the following categories of data – Parameter, Navigation, Status and Other. The Parameter messages or simply Parameters are usually sent at the begging of each flight right after the initialization of the communication link. The parameters are transmitted in both directions (uplink when written to the UAV memory and downlink when being read by the GCS). It can be concluded that lowering the size of the parameter messages will result in faster transmission time.

The other messages of interest to us are the Status messages. These messages contain information about the location of the UAV, its battery condition, spatial position data, readings from the sensors, etc. Although the structure of these messages is often not subject to modification, we can still control the rate, of which the data is being transmitted, and some messages may be modified by either creating a new message or by removing the unnecessary fields.

Table 2

Structure of MavLink's protocol frame

Field name	Byte index	Value	Description
Start frame	0	0xFE for v1.0 0x55 for v0.9	Indicates the start of a new frame
Payload length (n)	1	0 - 255	Indicates the length of the following payload
Packet sequence	2	0 - 255	Shows the packet sequence number in order to prevent packet loss
System ID	3	1 - 255	Identification of the sending system. Allows to differentiate systems on the same network
Component ID	4	0 - 255	Identification of the sending component. Allows to differentiate system's component (GPS, Accelerometer, etc…)
Message ID	5	0 - 255	Identification of the sending message (for example ID = 1 for HEARTBEAT message)
Payload	6 to (6+n)	0 – 255 Bytes	Payload data. Depends on the Message ID
CRC	(n+7) to (n+8)	ITU X.25/SAE AS-4 hashing algorithm	

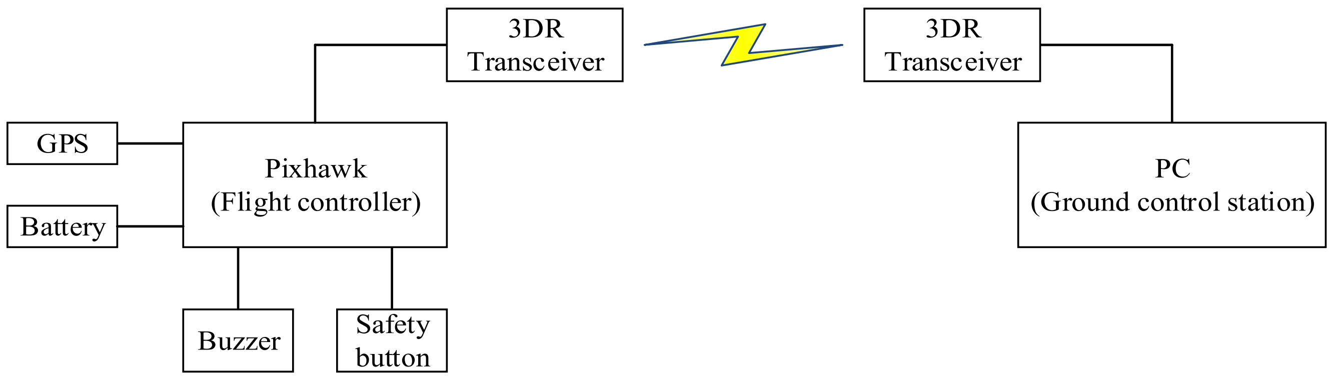

Fig. 3. A platform for evaluation of the MavLink protocol and its parameters

The Navigation messages are messages related to the control of the aircraft and are not subject to modification. The Other messages are related mostly to the means for control of other components of the UAV, like for example the camera, the retractable landing gears, etc. The messages are also not subject to our analysis.

A platform for evaluation the MavLink protocol and its parameters

For the purpose of our analysis and for a deeper examination of the characteristics of the MavLink protocol, an evaluation platform was build, which consist of the following:

- Pixhawk – open source high-performance flight controller, which supports a diverse range of robotic platforms – fixed wings, multi rotor, helicopters, cars and other. For our experiments the Ardupilot firmware was selected, which is available at [9].
- 3DR (3D Robotics) radio modules – the 3DR radio telemetry system is an open source system providing full-duplex radio link between the UAV and GCS. The interface to the modules is made using standard 5V- tolerant TTL serial / FTDI USB serial cables. The operating frequency of the modules is 433 MHz, the data rate - up to 250 kbps and the range is close to 1.6 kilometers. Those radio transceivers were selected due to their flexibility and native MavLink support;
- Ground Control Station – a PC with installed Mission Planner software (current version 1.3.44).

Fig. 3 depicts the architecture of the laboratory platform. As it can be seen from the figure, additional components were added to the flight controller – GPS and compass, buzzer, safety switch and a battery. The GPS and compass are used to collect actual data, which subsequently will be transmitted to the GCS. The buzzer is used to indicate the various states of the vehicle. The safety switch with LED indication is connected for safety reasons (a prerequisite for arming of the UAVs).

Methodology for creating new MavLink messages

One of the main advantages of using the MavLink protocol for telemetry data transmission is the fairly easier and straight forward process for creation of new messages. The use of these new messages can be quite helpful, especially when developing new components for the unmanned aerial vehicles, because it allows the developer (operator) to define individual messages, which suite his or her needs. The process is presented on Fig. 4.

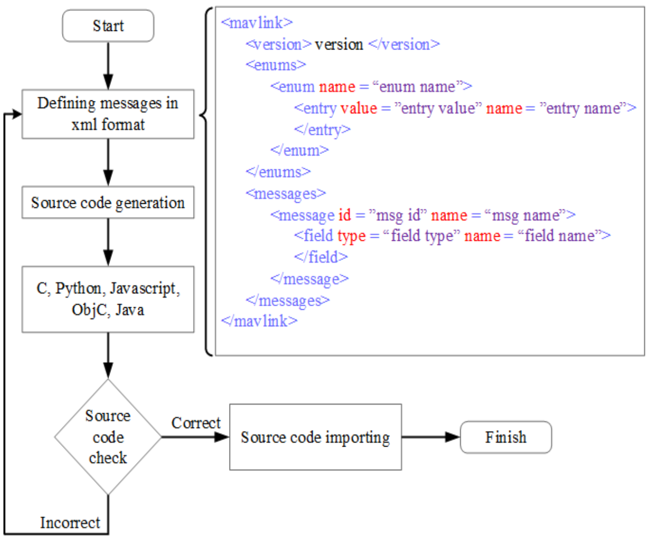

Fig. 4. A methodology for creating new MavLink messages.

The generation of the new MavLink messages starts with the definition of the message structure and in particular the description of the payload fields in xml (e**X**tensible **M**arkup **L**anguage) format. Every message consist of two mandatory fields - version and message, and one optional **enum** fields. The version field is defined at the start of every message and holds information about the protocol version (in our case it is 1.0). The <messages> tag consists of the <message>, <description> (optional) and <field> tags. It has to be noted that a single file may contain several <message> tags, each containing its own "id" and "name" fields.

As the name suggests, the **id** field holds the identification number of the message and the **name** field holds its name. The highest value for the **id** field is 255 (1 byte), and there is one mandatory message, with **id=0**, which is named HEARTBEAT. This message is absolutely necessary to be present within the information transmission process, because it maintains vital information about the aircraft, such as the UAV model, its type, status (if the aircraft is present and responsive), etc. The message **name** field represents the current name of the message, which is being transmitted and is compared to a switch-case statement for determination of how to process it. The **description** field is fully optional and its only purpose is to give a description about the message. The **field** tags (there may be more than one **field** tag in a single message) are used to define the type of information, which is being sent. There are five distinct types, which may be used: unsigned integer and signed integer with length of 8, 16, 32 and 64 bits, float, double and char. As mentioned earlier, the **enum** field is optional, but recommended, and it is used to add additional information to the message itself. For example, the developer might need to define a particular type of message, and then add the actual values as entries of **enum** type, as shown on Fig. 5.

The new MavLink message (Fig. 5) is defined with **id = 150** and **name = TEST**. This message contains just one field (8 bit unsigned integer) and its name is **type**.

```xml
<mavlink>
    <version>1</version>
    <enums>
        <enum name="MAV_TYPE">
            <entry value="0" name="MAV_TYPE_GENERIC">
                <description>Generic micro air vehicle.</description>
            </entry>
            <entry value="1" name="MAV_TYPE_FIXED_WING">
                <description>Fixed wing aircraft.</description>
            </entry>
            <entry value="2" name="MAV_TYPE_QUADROTOR">
                <description>Quadrotor</description>
            </entry>
            <entry value="3" name="MAV_TYPE_COAXIAL">
                <description>Coaxial helicopter</description>
            </entry>
        </enum>
    </enums>
    <messages>
        <message id = "150" name="TEST">
            <description> Here goes the message description </description>
            <field type = "uint8_t" name = "type">
                Unsigned int field, types of which are defined in MAV_TYPE enum
            </field>
        </message>
    </messages>
</mavlink>
```

Fig. 5. Defining an actual MavLink Message.

As shown, the contents of the message are defined as several entries of **enum** type. Every entry has a particular value and a corresponding name. Depending on the sent value for the entry, the receiving side would know the type of the aircraft (in this particular case).

The newly defined message is then converted to an actual programming code (supported languages at the moment are C, ObjC, Swift, CS, JavaScript, Java, Python), using the existing code generators [6]. After validation, the source code can be imported in a preferred working environment.

Experimental results

By using (1) we can calculate the minimum and maximum rate of the messages which are transferred per second. It has to be noted that the rate of our communication link is 57600 bauds per second, which is determined by the serial connection between the personal computer (GCS) and the connected 3DR transceiver. In addition to our header overhead (6 bytes) and the CRC checksum, we have to take into account also the overhead, which is added by the serial connection. In our case for every 8 bits of useful information, additional 2 bits are added (one start bit at the begging of the sequence and one stop bit at its end), which means that for every byte of information, 10 actual bits are being transmitted. The modification of (1) is reflected in (2):

$$(2)\quad \frac{\boldsymbol{msg}}{\boldsymbol{s}} = \frac{\mathbf{57600}}{\mathbf{10}*(\mathbf{8}+\boldsymbol{Payload})} = \frac{\mathbf{57600}}{\mathbf{10}*(\mathbf{8}+\{\mathbf{0}-\mathbf{255}\})},$$

where {0-255} is the size of the payload in bytes.

Based on (2), we can conclude that the minimum and maximum amounts of messages per seconds, which can be transfer by our experimental platform, are respectively 21.9 msg/s and 720 msg/s, as shown on Fig. 6. As seen in the figure, the message rate directly depends on the size of the message. Because of this, it is strongly recommended (if possible) to modify the firmware of the flight controller according to the needs of the project.

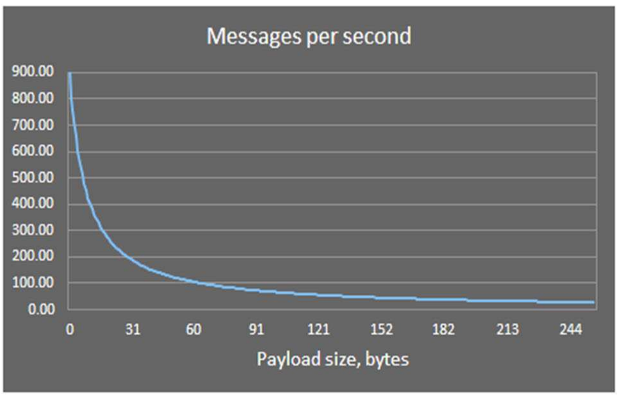

Fig. 6. Transfer rate of messages per second.

Reducing the number of parameters

As mentioned earlier, the reduction of the number of the parameters, which are being transferred prior to the system initialization, will decrease the amount of the information being transmitted and will decrease the boot up time of the autopilot. In order to achieve this, the firmware of the autopilot needs to be modified, subsequently recompiled and uploaded to the system (Fig. 7).

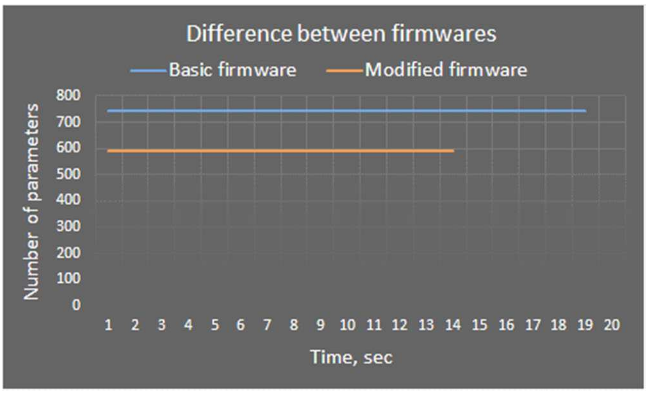

Fig. 7. Difference between basic and modified autopilot's firmware.

Fig. 7 represents the difference between a basic firmware, which is not optimized for any of the robotic platforms and a modified firmware, which in our case is optimized for a quadcopter UAV. Prior to the system initialization, the basic firmware sends out 743 parameters (the value for every parameter could not be accounted at this moment), which takes up to approximately 19 seconds. After recompiling and uploading the modified firmware to the autopilot, the parameter count drops down to 593 and the boot up time was decreased by approximately 5 seconds. Based on the aforementioned modification, it was found that the transmission of each parameter took approximately 23-25ms (depending of the values of the parameters).

Reducing the data rate

The reduction of the data rate, at which the different messages are being transmitted between the GCS and the UAV, is another factor, which can increase the overall performance of the system. All MavLink messages, according to their purpose, are divided into groups, which means that the rate of the individual messages cannot be controlled. There are total of ten message groups, and every one of them reports different information. The groups are: raw sensors, extended status, RC channel, raw control, positions, extra one, two and three, parameters and ADSB (Automatic Dependent Surveillance

Broadcast). The rate of each group is defined in hertz with values from 0 to 10, except the ADSB, which may have its rate increased up to 50 Hz. It has to be noted, that even when the rate of all groups is set to 0, the heartbeat and the radio status messages are still being transferred between the GCS and the UAV.

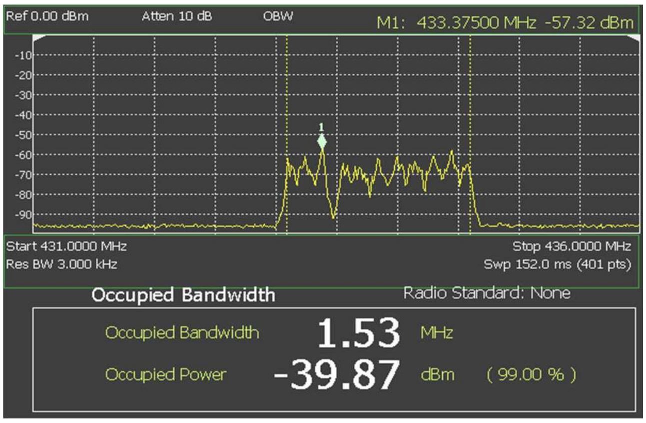

Fig. 8. SA readings while sending information with high data rate.

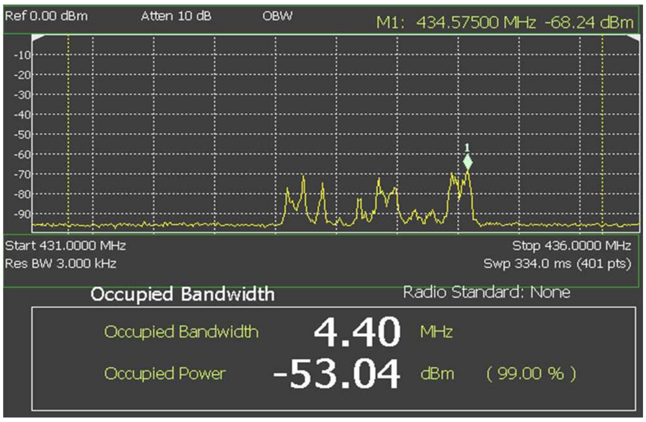

Fig. 9. SA readings while sending information with low data rate.

Fig. 8 and 9 present the readings of a spectrum analyzer while sending information with different data rates. As it can be noted, the high data rate readings show higher utilization of the bandwidth and average occupied power of 40 dBm. On the other hand the readings from Fig. 9, show that low data rate requires less bandwidth and less power. Based on the acquired results, it can be concluded that the minimization of the transmission rate, at which the unimportant information messages are being sent, may result in lower bandwidth utilization and higher overall efficiency of the system.

Conclusion

The presented paper provides a brief overview of the telemetry process and its main characteristics. The different types of protocols and the structure of the messages, which can be used for communication between the GCS and the UAV are also discussed. A comparative analysis on the packet overhead was conducted for the MavLink and the DNP3 protocols, and based on the results it was concluded that the MavLink protocol provides much better results than DNP3. Because of the open source nature of this protocol and its easy implementation, a platform for its further evaluation was proposed. Based on our analysis and experimental results, it was concluded that often times the standard structure of the protocol has to be modified to meet the requirements of the end user. By experimental evaluation, it was shown, that the reduction of the number of parameters in the structure of the messages and the reduction of the message transmission rate can lead to faster boot times, lower link utilization and higher overall system efficiency.

Acknowledgements

The work presented in this paper is completed as partial fulfilment of a project FNI-16-RU-10 “Development of a prototype of a robotic unmanned aerial platform for remote monitoring of critical infrastructure” and project FNI-16-FEEA-04 “Study of the impact of the controller location in the management plane on the performance of software defined networks”, financed under the Scientific Fund of the University of Ruse.

REFERENCES

[1] Madrigal, C. J. Zhang, RSL autonomous rover, Santa Clara University, 2015.

[2] Patranabis, D. Telemetry Principles. Tata McGraw-Hill Education, 1999.

[3] Konyushko, V.M. Features of UAV control systems assessing. In 2013 IEEE 2nd International Conference Actual Problems of Unmanned Air Vehicles Developments Proceedings, 2013.

[4] Johansen, I. H. Autopilot design for unmanned aerial vehicles. In Master Thesis at Norwegian University of Science and Technology, 2012.

[5] Hristov, G., P. Zahariev. Approaches for remote monitoring of the parameters of an unmanned aerial vehicle. 2013.

[6] MavLink Micro Air Vehicle Communication Protocol – http://qgroundcontrol.org/mavlink/start

[7] MavLink Common Message Set – https://pixhawk.ethz.ch/mavlink/

[8] Official Python 2.7. Documentation – https://docs.python.org/2.7/

[9] Ardupilot firmware available at https://github.com/ArduPilot/ardupilot

***Eng. Svilen M. Borisov** - PhD Student, Department of "Telecommunications", University of Ruse "Angel Kanchev". His current research interest include telemetry systems and protocols, unmanned aerial vehicles and robotics.*

tel.: +359 82 888 817 e-mail: sborisov@uni-ruse.bg

***Eng. Jordan P. Raychev** - PhD Student, Department of "Telecommunications", University of Ruse "Angel Kanchev". His current research interest include telemetry systems and protocols, SDNs, communication networks, IoT, unmanned aerial vehicles and robotics.*

tel.: +359 82 888 817 e-mail: jraychev@uni-ruse.bg

***Assoc. Prof. Georgi V. Hristov, PhD** - Department of "Telecommunications" at the University of Ruse "Angel Kanchev". His current scientific interests include communication networks, new generation network architectures, network virtualization technologies, unmanned aerial vehicles, 3D visualization and printing technologies and robotics.*

tel.: +359 82 888 663 e-mail: ghristov@uni-ruse.bg

***Assist. Prof. Ivan H. Beloev, PhD** - Department of "Transport" at the University of Ruse "Angel Kanchev". His current research interests include transport and road traffic systems, unmanned aerial vehicles and telemetry protocols.*

tel.: +359 82 888 605 e-mail: ibeloev@uni-ruse.bg

***Assoc. Prof. Plamen Z. Zahariev, PhD** - Department of "Telecommunications" at the University of Ruse "Angel Kanchev". His current scientific interests include IP and sensor networks, cryptography, wireless communications, IoT, 3D technologies and robotics.*

tel.: +359 82 888 663 e-mail: pzahariev@uni-ruse.bg

Received on: 12.12.2016

THE FEDERATION OF THE SCIENTIFIC-ENGINEERING UNIONS IN BULGARIA /FNTS/

is a professional, scientific - educational, nongovernmental, nonpolitical, nonprofit association of legal entities - professional organizations registered under the Law on non-profit legal entities, and their members are engineers, economists and other specialists in the fields of science, technology, economy and agriculture.

FNTS has bilateral cooperation treaties with similar organizations in multiple countries.

FNTS brings together 19 national associations – Scientific and Technical Unions (STU), 34 territorial associations, which have more than 15 000 professionals across the country.

FNTS is a co-founder and member of the World Federation of Engineering Organizations (WFEO).

FNTS is a member of the European Federation of National Engineering Associations (FEANI), and a member of the Standing Conference of engineering organizations from Southeast Europe (COPISEE), UN Global Compact, European Young Engineers (EYE). The Federation has the exclusive right to give nominations for the European Engineer (EUR ING) title.

Contacts:

108 Rakovsky St., Sofia 1000, Bulgaria

WEB: http: //www.fnts.bg Email: info@fnts-bg.org