REMOTE CONTROL OF UNMANNED DEVICES

Petr STODOLA
University of Defence
Brno, Czech Republic
Petr.Stodola@unob.cz

Abstract

This paper deals with communications protocol used for data transfer in a specialized wireless network. This network is intended for connection between remote-controlled reconnaissance devices. The protocol serves to arrange mutual communication between constituent devices (clients) and a control station (server). The protocol ensures automatic sending a GPS position of constituent devices, sending control commands and information, text or data messages. The protocol can be used for remote control of reconnaissance devices. This paper deals with the protocol on application layer.

Keywords

Protocol, message, command, wireless network, UDP, client, server, remote-controlled device, reconnaissance, remote control, application layer

1. Introduction

The present era is characterized by the tendencies of modern armies to implement the special remote-controlled devices and systems which ensure reconnaissance and fight operations without a human sources waste risk. These devices are controlled from a control station. In the case of their destruction by the enemy there is only damage to material. These and similar systems gain importance especially in connection with the growth of terrorism [1].

Figure 1 presents the scheme presenting the communication of remote-controlled devices system. The system is composed of the group of devices controlled from the control station. Communication is ensured (in the first place) through a communication router (however, it is possible to communicate between constituent devices). The number of devices (*n*) is limited by network throughput.

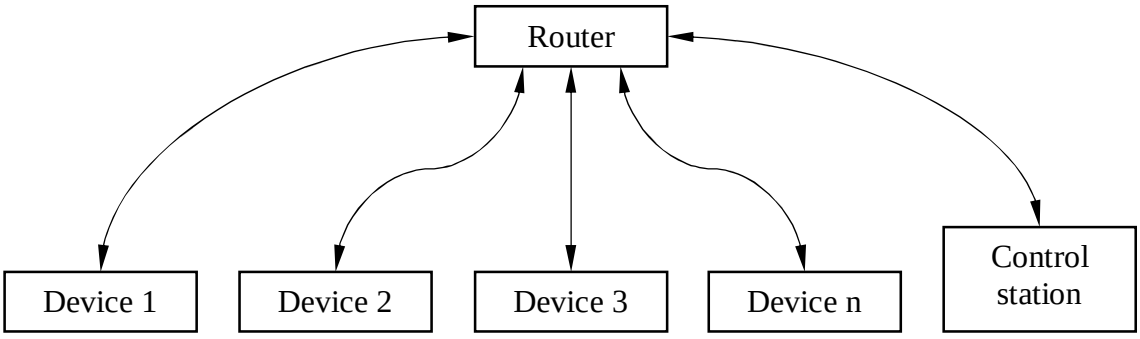


Figure 1: Block scheme of remote-controlled devices system

Remote-controlled device can be represented by ground reconnaissance or fight vehicle or pilotless device (glider or helicopter). Each remote-controlled device is equipped with the sensorial system (cameras, laser range-finder and gun sight) and it can be equipped with the fight system. The communication can work even between other than remote-controlled devices (e.g. individual soldiers who are equipped with PDA).

On the control station there is an overview and surveillance system with 3D digitized maps. Here we can see the positions of all devices of the system in the real time and we can take an easy control of them. The router can be a pilotless device with a relay station. It flies in the air about several kilometers above the ground and creates a communication umbrella for other devices and a control station. According to practical experience it is possible to cover about several tens of square kilometers.

This paper deals with description of the communications protocol which is used and verified in the remote-controlled devices system developed at the University of Defence in Brno. The protocol is based on application layer (according to ISO/OSI network model [3]). On physical layer, the data can be transfer on wi-fi network [6], radio network by digital transmitter and it is experimented with adaptive laser connections. The used network must guarantee needed data throughput (for transfer of video or audio from the constituent devices). On data link, network and transport layer, the data transfer is ensured via classic protocols (IP, UDP) [3].

2. Communication architecture

The principle of communication in the system can be inscribed with client-server architecture [4]. The constituent devices are clients, the control station is a server. The clients are controlled from server and they periodically send basic information (especially their positions) to the server. Communication always proceeds through the server (even communication between constituent devices). The router ensures data distribution.

It is important to secure a communication even in highly demanding conditions (complex terrain, contrary meteorological conditions, jamming). In these conditions, connection loss or data corruption can occur. Hence, there was chosen the protocol UDP for data transfer on transport layer [4]. UDP is a very simple protocol which gives very good results in similar conditions. It is very fast connectionless protocol with minimal demands to protocol direction. Sending UDP packets is ensured via Windows Socket network interface.

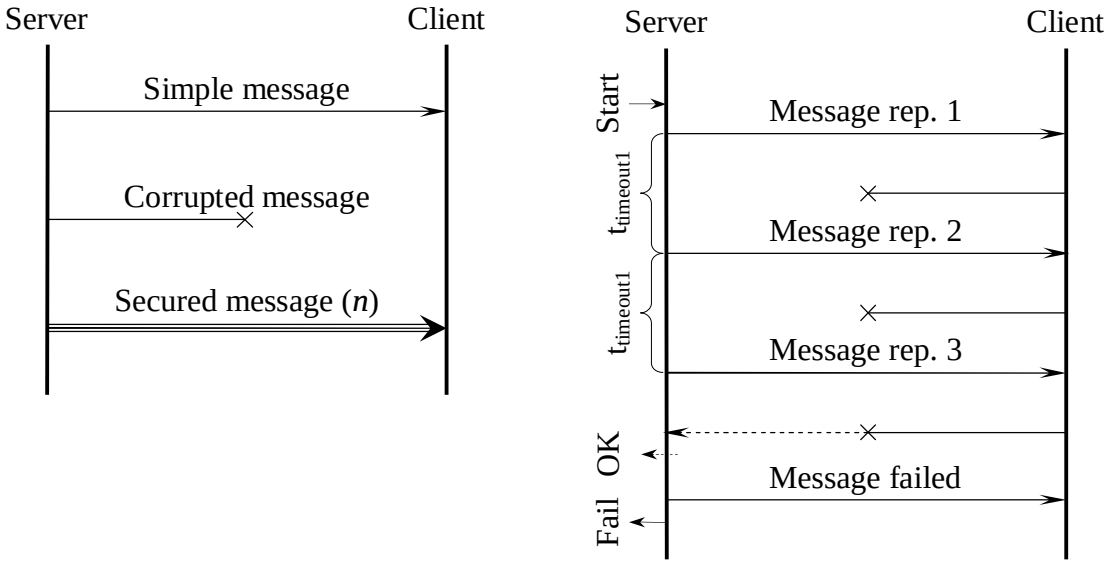
The TCP protocol is the option to UPD protocol. This is the protocol maintaining the connection between a client and a server and it can guarantee data delivery and order of received packets. But the disadvantage partly consist in the biggest demands to protocol direction and especially, experience shows that in demanding transfer conditions there are problems with connection maintenance.

Protocol UDP does not guarantee packet delivery (and sender does not find out whether a packet is not delivered) and delivery order. Furthermore, it can offer a situation when the packet is delivered twice to the same address. When the important data are transferred, the data transfer is ensured on a software level. Protocol UDP is very suitable for transfer of stream data (online video and audio from device cameras).

3. Packet transfer and used symbols

The above mentioned protocol ensures transfer of data block within several packets which are sent subsequently via UDP protocol. The size of packet about 1 KB offers good result (bigger size means the risk of increase of packet corruption or its loss).

Figure 2a presents used symbols. Simple continuous line represents transfer of a simple message (packet) from a server to a client (of course, transfer can be sent even in the opposite direction). Broken line figures a corrupted (undelivered) message. Triple line represents a software secured transfer of a message. Figure 2b presents the scheme of a secured transfer. The principle consists in sending a simple message to a client (or in opposite direction to a server) and waiting for its answer. If the answer does not come in interval $t_{timeout1}$, the data transfer from the source to destination is repeated. The number of repeating is defined by parameter *n* which can be set arbitrarily according to situation (the number in figure is set to value 3).



a) Used symbols *b) Scheme of a secured message*

Figure 2: Simple and secured messages

4. Login of clients

Figure 3 presents clients login to a server and the principle of the software connection maintenance. Each client who wants to login to the system sends a login message to a server. The client sends this message in time interval t_{login} periodically. When the server obtains a login message, it considers the client being entered the system. When the server does not obtain a new login message in the time interval t_{logout}, the client is automatically checked out.

A similar system of message sending works even in the opposite direction. The server periodically sends information messages to all clients. These messages acquaint clients with server existence. If clients do not obtain information message within particular time, the server is considered to be disconnected. There is also information about other clients in the information message, therefore each client has a view of status and position of other clients.

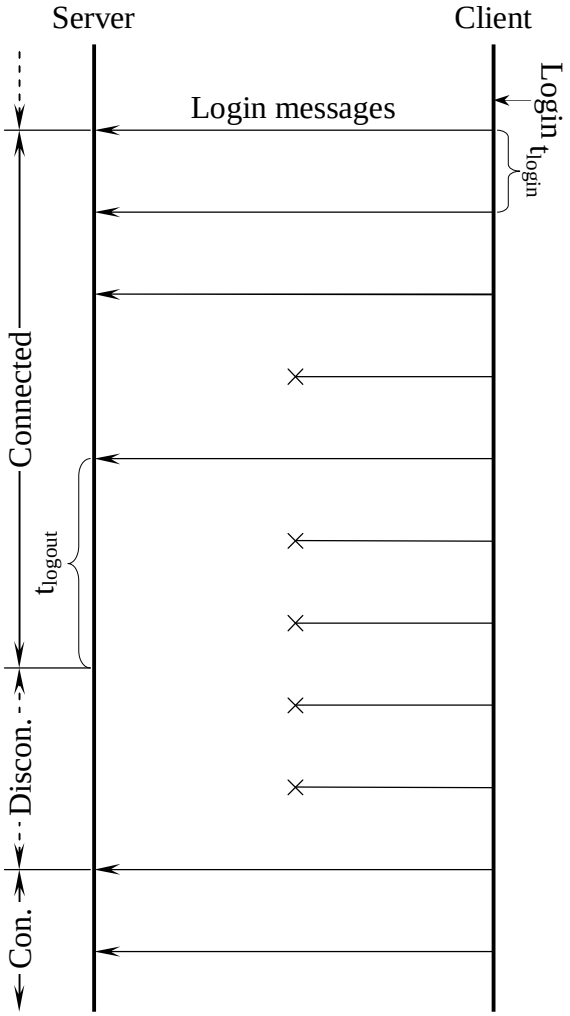
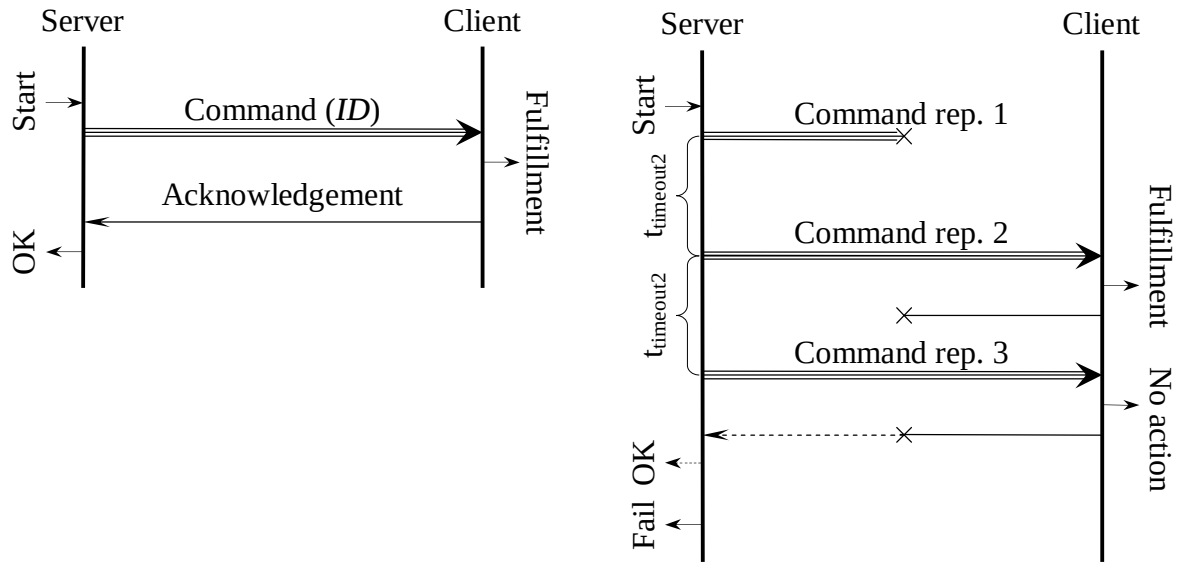


Figure 3: Client login

5. Command messages

Command messages serve to transfer commands and data messages between clients and a server. In command message there is possible to send the commands for control of remote-controlled devices, information messages with devices status, text-based messages for text communication and data messages with arbitrary content.

The scheme of the command message sending is presented in figure 4a. Command must get in the one packet and it is sent via secured message (each command has a unique ID which serves for its discernment). Command receiving is confirmed by a simple message. The command is sent again in case its confirmation does not occur in time interval $t_{timeout2}$. It is possible to set the number of repeating arbitrarily according to command importance. Figure 4b presents the example of sending a command message. The number of repeating is set to value 3.



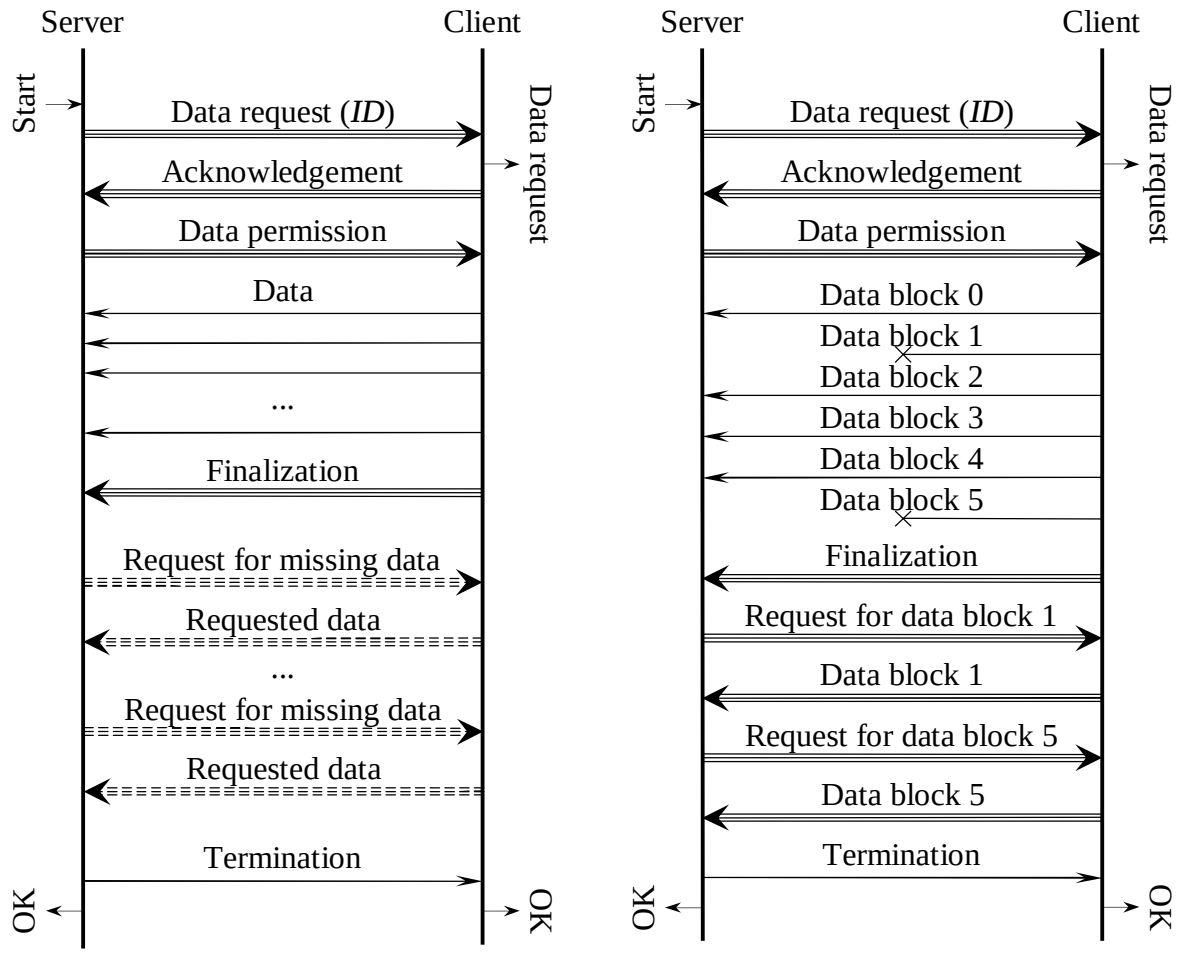
a) Scheme of command message *b) Example of command message*

Figure 4: Command message

6. Sending secured data

In case of necessity to send important data between a client and a server (for example digital photography etc.), it is necessary to use the principle of sending secured data. The scheme based on this principle is presented in figure 5a. Data transfer starts by data request (with unique ID), the other side confirms it. Next, permission for data transfer start is sent. Data are divided into blocks and they are transferred in the simple messages in sequence. The receiver is informed when the transfer is over. When certain packets are damaged, there are requests for their restoring (dashed lines in figure). Finally, a termination message is sent.

The particular example of sending secured data is presented in figure 5b. Of course, the protocol is protected against certain undelivered control messages. In this case, the whole process is stopped after certain time expiration and error is reported to the operator.



a) Scheme of sending secured data *b) Example of sending secured data*

Figure 5: Secured data sending

7. Sending stream data

The scheme of sending stream data (online video or audio data) from a client to a server is presented in figure 6. The server asks for data (likewise the client can ask for) which a client sends via simple messages in certain time intervals. In case of damage of the certain packet, failure of video or audio segment occurs. But usually, it is only a short time segment which is not important for the data transfer purpose. The whole communication can be in progress at the opposite direction at once (full duplex). The quality of transferred images and sounds is dependent on the wireless network throughput.

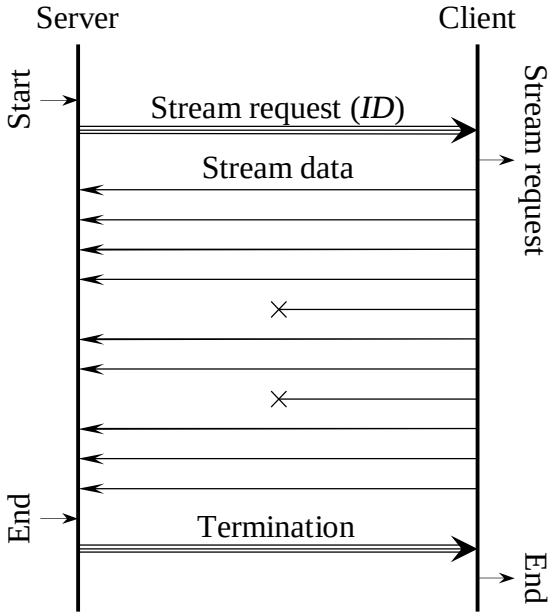


Figure 6: Scheme of sending stream data

8. Conclusion

The values of some parameters of described protocol (for example time intervals t_{login}, t_{logout}, $t_{timeout1}$, $t_{timeout2}$, the number of repeating secured message sending etc.) must be set according to the situation. For example, remote control of a device asks shorter intervals for commands repeating (milliseconds) comparing to text-based communication (tenths of seconds).

The protocol is used for communication and remote-controlled devices control which are developed at the intelligence and electronic warfare group at the University of Defence in Brno. Recently, there are some pilotless devices (gliders and helicopters) available and we are working on developing the ground remote-controlled reconnaissance device capable of a target destruction.

The results of this project were published and presented at the military fair IDET 2007 in Brno [1]. Mentioned above reconnaissance device is based on a terrain four-wheel YAMAHA YFM 450. The platform with sensorial and fight system is fixed on a four-wheel chassis. Sensorial system consists of two stereo cameras. The image is transferred to the overview and surveillance system on the server. Fight system is represented by a kind of rifle.

Nowadays, we are working on the automation of the decision-making process. The closest aim is focused on automatic searching, watching and sighting of the targets and their further destruction. The estimated calculations and experience demonstrate that the device can sight and destroy the target in a very short time interval (couple of tenths of seconds).

References

[1] RÝZNAR, B., MAZAL, J. The battlefield of the 21st century and process automation (and its role in future forces). In: Proceeding of the International Conference on Military Technologies ICMT ’07. Brno, 2007. pp. 231-237. ISBN 978-80-7231-238-2.

[2] RYBÁR, M., et al. Warfare modeling and simulation (Modelovanie a simulácia vo vojenstve). Bratislava: MO SR, 2000. ISBN 80-88842-34-4.

[3] JIROVSKÝ, V. Vademecum of network administrator (Vademecum správce sítě). Praha: Grada, 2001. ISBN 80-7169-745-1.

[4] PIRKL, J. Network and Internet programming in Windows (Síťové programování pod Windows a programování Internetu). České Budějovice: Kopp, 2001. ISBN 80-7232-145-5.

[5] BRISBIN, S. Build Your Own wi-fi Network (Postavte si svou vlastní wi-fi síť). Praha: Neocortex, 2003. ISBN 80-86330-13-3.

[6] GUIZANI, M. Wireless communications systems and networks. New York: Kluwer, 2004. ISBN 0-306-48642-3.