Ground Segment of Distributed Ground Station System

Katarzyna Dąbrowska *, Marcin Stolarski †
* Warsaw University of Technology, Faculty of Electronic and Information Technology , Institute of Computer Science, Warsaw, Poland, e-mail: *K.Dabrowska.4@stud.elka.pw.edu.pl*
† Warsaw University of Technology, Faculty of Electronic and Information Technology , Institute of Radioelectronics, Warsaw, Poland, e-mail: *M.Stolarski@elka.pw.edu.pl*

***Abstract*— The article discusses an innovative method of data transmission between a satellite and the Earth. The authors suggest using a distributed network of ground stations, which simultaneously receive data from the satellite. The article presents new solutions, which allow improving the quality of receiving channel in comparison to an antenna array. Subsequently, a proposed realization of Distributed Ground Station System with description of transmission protocols, data transmission and storage, authorization system and devices and operators communication is shown. This network is a part of PW-Sat satellite project [5], developed by the students of Warsaw University of Technology.**

Keywords—**space communication, ground station, satellite, GS-SCP protocol.**

I. Introduction

A lot of universities take interest in the aspects of building satellites [1]. Small and cheap satellites are being built, together with ground stations enabling data exchange. Isolated ground stations are used marginally, because of a limited period of radio visibility between the station and a satellite. Seldom are such stations used for supporting satellites from other universities. The very development of a link between the Earth and a satellite is not an easy task and requires building an appropriate antennae system in order to assure a good quality of connection. The authors suggest connecting such stations via a computer network, which would allow greater use of ground stations. Such mechanism enables a multiple extension of a satellite's availability period, and at the same time an increase in the amount of data sent. To increase a number of available stations, the authors also suggest to expand the network by radio amateur stations, since radio amateurs posses appropriate hardware and have access to the Internet. The authors' next suggestion consists in a simultaneous receipt of data from a satellite through many ground stations, and then sending them to the packet voting system. This system enables the improvement of link quality by reducing bit error rate (BER). The article presents the idea of developing a distributed ground station and the detailed proposal of its practical realization.

II. Currently used ground station systems

Presently practiced solutions most often make use of a single ground station. If such a solution is used for communicating with the satellites located at the Low

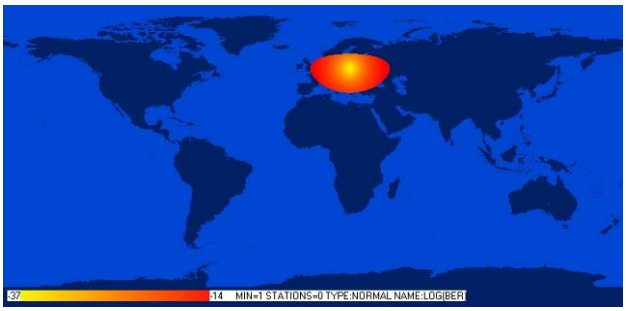

Fig1. The range of a single ground station.

Earth Orbit (LEO), the station has a poor visibility of a satellite (Fig 1), which results in the station being operative merely during 10-20% of a day. Since many institutions (e.g. universities) send smalls satellites and build their own ground stations, the potential of the available mechanism remains unfulfilled. The estimated number of such stations exceeds 100.

The next group of people whose devices can be used for space communication are radio amateurs. The authors have measured [6] data traffic of radio amateur network named APRS-IS. The research has shown that radio amateurs use further 10'000 potential ground stations (Fig 2).

III. Mathematical model of DGSS

When we connect ground stations via the Internet using special software, we can get a tool called Distributed Ground Station System. This solution allows to increase the amount of time during which a satellite is in the range of GS (Fig 3).

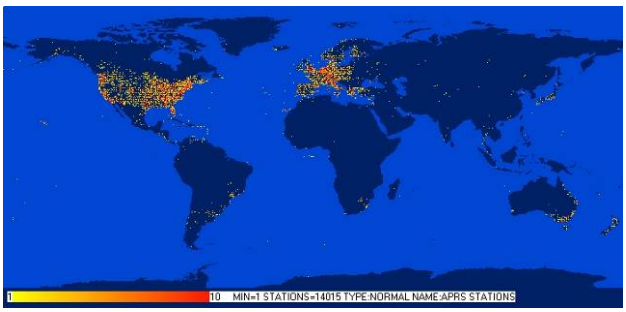

Fig 2. The location of radio amateur stations around the word which use APRS-IS network.

Fig 3. The number of ground stations in the range of a satellite.

The next advantage of the system is the possibility of distributed receipt of information through a couple of stations at the same time. It seems that such simultaneous receipt should improve the radio link quality. The mathematic simulations of authors have shown that with appropriate signal to noise ratio such receipt is better than when using antenna arrays [2,3,4,7,8].

For the simulation a number of scenarios were constructed, and bit error rate (BER) was calculated for each one of them.

Scenarios:

1. Receiving the signal through a single ground station with a 6dB gain of antenna (BERss).
2. Receiving the signal through a single ground station with an antenna array composed of five (6dB gain) antennas (BERn). The summary gain of antenna system was 13 dB.
3. Receiving the signal through five separate receipt systems identical to a single ground station. BER was calculated for the case when none of the station received data correctly (optimal solution, BERo). If one of the stations received correct data it is assumed that it has been interpreted correctly.
4. Receiving the signal through five separate receiving systems identical to a single ground station. BER was calculated for the case when most of the stations did not receive data correctly (voting solution, BERv). If most of the stations received correct data it is assumed that it has been interpreted correctly in the Packet Voting System.

Figure 4. shows the results of BER calculations, depending on the distance. BERss line presents the result for a single ground station. It is a reference line. BERn shows how much better a channel is when we use a five times bigger antenna system. BERo is the theoretical upper boundary of the system's capabilities. It is the case

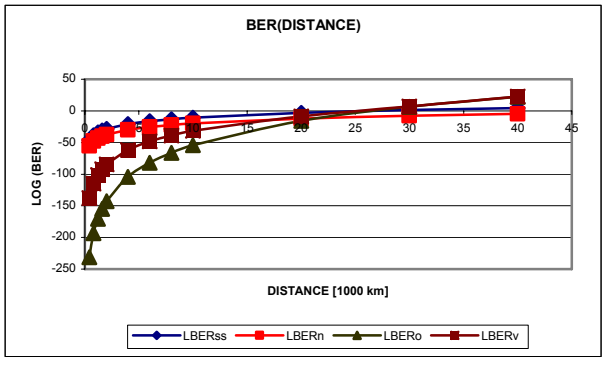

Fig. 4 BER for various realizations of ground stations.

when we get at least one correct result, and somehow we know which station got this result. BERv is the line that shows bit error rate when using the Packet Voting System.

This system assumes that most of the stations receive the data correctly, which would enable to reject the incorrectly received data through majority voting.

As we can see, on distances shorter than 15'000 km, Packet Voting System consisting of five ground stations assures better channel quality than when using a single ground station with a five times more powerful antenna. It is worth pointing out that the distance of 3'500 km is usually a borderline range for Low Earth Orbit satellites, due to radio horizon. Together with a decreasing distance, the advantage of comparing systems increases.

IV. Building DGSS network

The DGSS network is designed to communicate between ground stations located all around the world via the Internet. Currently those stations are a part of the APRS-IS network. Its capabilities, however, are much more limited than those of DGSS (Fig 5).

Ground stations, by means of client applications called DGSS-Client, communicate with DGSS-Server through the protocol layer based on TCP. This architecture allows the centralization of the data received from a satellite by various GSs in the Data Base (DB). The DGSS-Client applications are able to communicate with the server only. Thus, the DGSS-Server plays the administrative role. This is the only part of the system that has access to DB, and is able to communicate with all the client applications in the system.

Because of a short access time to a satellite for the single ground station, the data transmitted by all GSs, connected to our system and stored in DB, also in the aggregated form, is a source of a deeper, generally accessible knowledge. On the other hand people who do not have their own GS are able to communicate with a satellite by using the limited version of DGSS-Client.

The GS Operator's computer has hardware interfaces, which are used by a client application for communicating with outer devices (Fig 6).

Owing to the *Rs232Rotor* interface, program operating of the rotor is possible. For this we use the Orbitron [15] application, which tracks the satellite's orbit parameters by itself, and on their basis it calculates the angles by which a rotor needs to be rotated for the satellite signal to be receivable. The Orbitron is also used for the automatic setting of the frequency of the radio with which it is

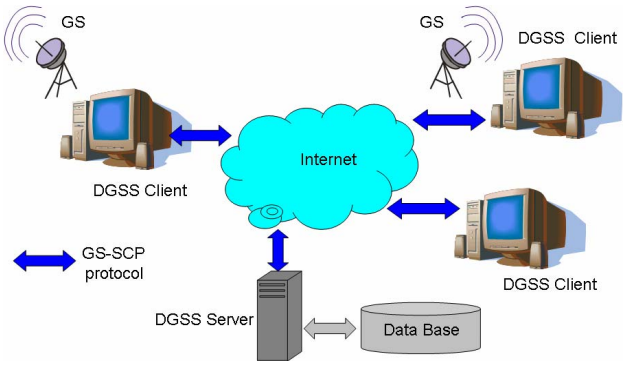

Fig. 5 DGSS schema

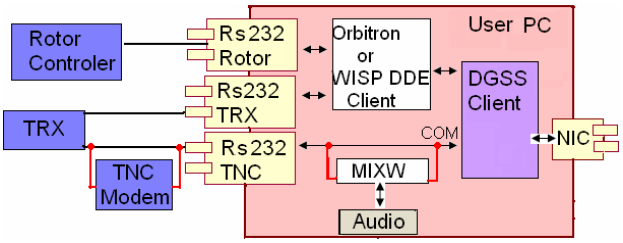

Fig. 6 DGSS Client schema used on User PC

communicating through *Rs232 TRX* interface.

The DGSS-Client has access to the Internet through Network Interface Card (NIC) for communicating with two instances of server – the main DGSS-Server and NORAD-Server.

The NORAD-Server is the FTP server, which periodically downloads, from NORAD, the parameters of the orbit of the satellite being tracked by the system, store them and make them available to DGSS-Clients. The application makes use of this information to visualize the location of the satellite, and controls the radio and the rotor from the level of DGSS-Client without the necessity of using Orbitron.

To do this we will implement the DDE-Lib library, which will calculate the parameters of a satellite's orbit downloaded from NORAD. The resulting values will be sent to the WiSP DDE Client application [16], and translated into correct commands for various models of rotors and radios. Only these settings will appropriately set the rotor and the radio.

DGSS-Client mediates sending the data in two directions: the server, using NIC, and the radio (TRX), using *Rs232 TNC* interface. DGSS-Client sends data to the satellite using COM port. The subsequent way of packets can be of two kinds:

- COM <-> MIXW <-> Rs232 TNC <-> TRX
- COM <-> Rs232 TNC <-> TNC Modem <-> TRX

As it can be seen, MIXW [14] is a software counterpart of the TNC modem.

The simulation of transmission using audio files is also possible (Fig 6).

V. Communication Methods

For the communication between individual system modules, a TCP based protocol called Ground Station - Server Communication Protocol (GS-SCP) is used. The TCP protocol solution is easy to implement, and provides access to fast packet transmission [9, 10, 11]. A very important aspect for the system efficiency is the size of data transmitted. With TCP we have a full control over what we are sending, contrary to HTTP. In the latter case with every new query an expanded header is being sent, which can even double the amount of data transmitted! A project of the GS-SCP protocol packet is given in Fig 7.

Timestamp	Type	Length	Data	Sign Length	Signature
8 bytes	1 byte	4 bytes	-	1 byte	128 lub 256 bytes

Fig 7. The GS-SCP protocol packet.

As it can be seen above, the first field is *Time stamp*. The next one is *Type*, which contains the identifier of one of the below types:

1. *Authorization Packet (AP)* sent by a client in order to be authenticated by the server;
2. *Acknowledgement Authorization Packet (ACK AP)* sent by the server in order to be authenticated by the client;
3. *Change Key Packet (CKP)* – used by the server or client for sending a newly generated public key to the other side;
4. *Initial Order Packet (IOP)* – sent by the client to the server in order to commission a command transmission to a satellite through one of GSs, chosen by the server;
5. Through *Refused Initial Order Packet (RIOP)* the server refuses to transmit a command to a satellite in the case when too many of them awaits the transmission;
6. *Request Order Packet (ROP)* – received by a GS, chosen by the server to transmit a given command to a satellite;
7. *Acknowledgement Send Order Packet (ACK SOP)* – a packet sent by a GS to the server, confirming a successful transmission of a command to a satellite;
8. *Communiqué Packet, CP,* is a packet sent to the server by a GS, containing a message from a satellite received by this GS;
9. *Delivered Order Packet (DOP)* is a packet sent to the client by the server, from which a demand for command transmission to a satellite has been issued;
10. *Share Ground Station Packet (SGSP)* is sent to the server by the GS operator in order to make one's ground station available to other users of the system or to disconnect it, depending on the value of the parameter;
11. *Change Ground Station Parameters Packet (CGSPP)* contains new values of GS attributes, separated with 0; sent in order to introduce changes to the database;
12. *Change User Parameters Packet (CUPP)* has an analogical structure to CGSPP with the only difference being that it concerns the user;
13. *Ping Packet (PP)* is sent both by the client and the server; its function is to detect the connection breakdown from the level of the program as soon as possible;
14. *Quit Packet (QP)* is sent in order to inform the other side about an abrupt disconnection.
15. *SQL Packet (SQLP)* contains the number of a database query about the data needed for the visualization in the DGSS-Client application;
16. *Answer SQL Packet (ASQLP)* is an answer from the database for an SQLP query; its structure has the form: record1\0record2\0record3\0\0record4\0…\0, where \0\0 is the change of verse;

Next to *Type* is *Length*, containing the length of a part of packet data in bytes. Then, in *Data*, proper information in the chars form compatible with APRS [17,18] standard is being sent. If a given packet has an e-signature, it is

contained in the *Signature* field, located towards the end of a packet.

The second reason why TCP is used is the possibility of two-way communication. One of the fundamental assumptions of the system is the possibility of sending commands to a satellite by the users not having a GS. It is thus indispensable to initiate transmission via the server. This is why TCP is far more superior to HTTP, since in HTTP only the client may initiate transmission.

VI. Authorization Methods

Taking into consideration the law governing the question of openness of radio-received APRS [17,18] packet sending, the integrity of data in the system is preserved owing to the mechanism of e-signature based on the RSA algorithm [12].

During the initialization, the server generates two kinds of keys: a private one (Kss) and a public one (Kps), where Kps is saved to the current folder of the server. Subsequently, the administrator transmits it to all registered GS system users via e-mail. Such a solution eliminates the necessity of using Public Key Infrastructure (PKI) and key servers, since it is not required for ensuring the safety of the system developed [13]. Another optional solution would be for the server to send back the packet with its own Kps to every reporting GS. Then, however, a server would be vulnerable to Denial-of-service (DOS) attack, consisting in overflowing the communication channel.

During initialization, each GS, similarly to the server, also generates a pair of keys (Kpg, Ksg).

The integrity of a transmitted packet containing crucial information for the safety of a satellite and data received is confirmed both by the GS and the Server by means of an e-signature [12], which, if it is a part of the packet altogether, is present in the *Signature* field. The length of this field is constant and equals 128 bytes. If it turns out that this length is insufficient for the expected level of safety, it will be extended to 256 bytes.

The decision as to whether a given packet will be signed is up to the server. More precisely, it is on the part of the server where the information about which type of packet should contain a signature is implemented. Sending this information in a packet, for instance as a bool *IsSign* flag, could cause an attack resulting in the removal of the signature and the flag.

In such a case the server would not be able to detect the incompatibility of signature, for it would not know about its existence altogether!

The Client-Server authorization begins with the client being authenticated at the server. For this, the client sends

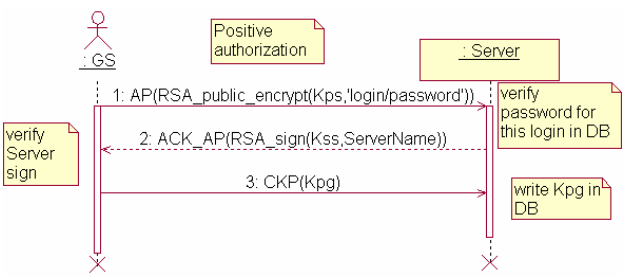

Fig. 8 Authorization

the AP packet ciphered with a login and a password by means of Kps (Fig 8).

The server deciphers it with its own Kss. Then it calculates md5 and compares it with the login and the corresponding hash value in the database. Should the password supplied by the client not match the password in the database, the server breaks the session, thus disabling the DOS attack. When the password is verified, the server gets authenticated at the client. To do this, it sends an e-signed ACK AP packet with the content recognized by the client (e.g. the server's address).

After receiving ACK AP, the client verifies the signature by using Kps. If the operation is successful, the client may be certain that he or she has contacted the server. In the opposite case, the client breaks the session, since it is obvious that it is being intercepted by a third party. When the signature has been verified by both sides, the client sends a CKP packet containing his or hers Kpg to the server, which now can verify e-signatures of a given client.

There is a risk of a packet being intercepted by a third party. This may result either in sending the message by an unprivileged user or the modification, repetition or even failure in delivering the message to the receiver. E-signatures eliminate added or modified packets on the part of the receiver, and the TCP protocol itself signals the deletion of such packet with an error [10]. The possibility of repeating a packet is eliminated by saving last Timestamps for each kind of packet in the database. On receiving each signed packet, the server compares its Timestamp with the one previously saved. If it happens to be larger, the receiver updates the old Timestamp. In the opposite case the packet is rejected.

If the server or one of the clients finds that a private key has been broken, a new pair of keys is generated, followed by the asynchronic sending of a new public key to the other side in the CKP packet. From now on the sender will use the new private key for signing messages.

The privileges control is performed through the access levels. In the system we have four kinds of users:

1. An Operator is a user having his or her own ground station, who is inclined to make it accessible to other system users;
2. A SuperUser is a member of a team, registered in the database, entitled to send control commands to the satellite from the system, due to not having his or her own GS;
3. A RegisterUser is a user who can send marketing telecommands; for this he or she has to provide identification data on the first login;
4. A Guest can only view statistics without being verified;

With each action performed, the server checks whether a given user possesses the privileges adequate to a given action.

VII. API

The user application for the whole system, the Ground Station Application (GSA) consists of two modules: Radio Application (RdA) and User Application (UsA).

Ground Station Module (GSM) is a module for communicating between the GSA and the server through

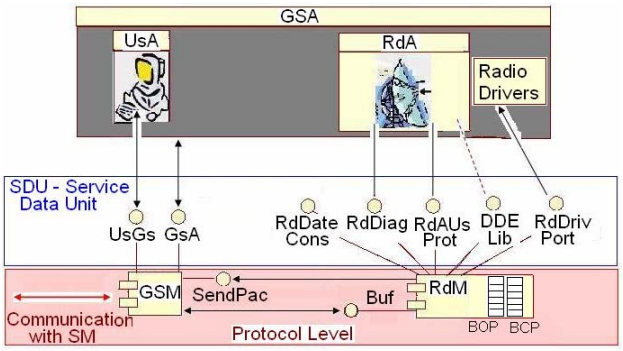

Fig 9. The modularity and program interfaces of the DGSS Client

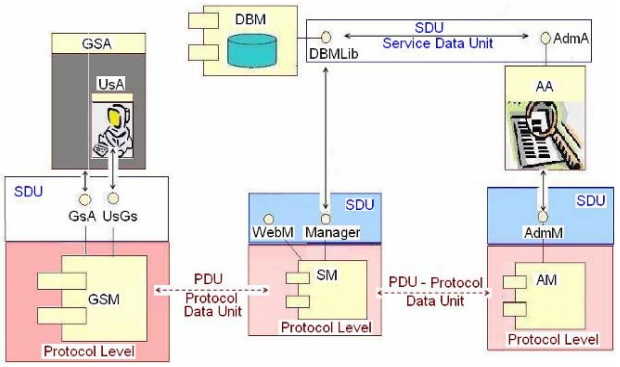

Fig. 10 The modularity and program interfaces of the DGSS system.

the protocol layer, using the following Service Data Unit (SDU):

5. *GsA* – a library for communicating between GSA and the GSM protocol knot;
6. *UsGs* – a library for communicating between UsA and the GSM protocol knot. The whole processing of an SQL query results takes place in the GSA itself, more precisely in the UsA component;
7. *SendPac* is used by the station modules having no connection to the server for sending packets to the server through the protocol layer (Fig 9).

Radio Module (RdM) is a module used for managing RdA application and radio drivers. Owing to the module, they can communicate with the server through the protocol layer, using the GSM module interfaces.

RdDrivProt is a library used for the communication between radio drivers and the RdM knot in the protocol layer. Current plans include the communication between the RdA application and the radio through the COM port, but there are no contraindications to redefining the library in the future, for instance to make it suitable for the USB port communication.

Owing to the *Buf* interface, the GSM module has access to message and commands buffers located in the RdM module.

RdAUsProt is a library used for the communication of the presentation layer of the RdA application with the GSM protocol knot and the SM server. This communication is managed by the RdM module.

RdDateCons is used for confirming data consumption. Before each application shutdown, all messages and commands from BCP and BOP buffers are saved to files and their content deleted. The filenames contain the save time of the content one of the buffers, i.e. the time when the GSA application was shut down.

Placing the buffers in RdM, instead of in RdA, facilitates the access to the messages received by a local station from the GSM module. It is used in sending to the server the UsA level queries about the statistics of messages sent in the whole system. Before sending a packet with an SQL query, the GSM module looks into the BCP buffer and check whether it does not look for a message received by own station. If the message found in the buffer is correct, GSM sends the data to UsA without the server's participation.

RdDiag is a library used for diagnosing the radio applications, i.e. RdA and RdM (Fig 9).

In case a user does not possess a transmitting device, i.e. does not use the RdA module of the GSA application, the RdM module from the protocol layer that manages this application is also not used.

Data Base Module (BM) as a single *DBMLib* interface for communicating with the SM server (used by the *Manager* library) and with the Administartive Application (AA) used by the *AdmA* library (Fig 10.).

Server Module (SM) contains two interfaces (Fig 10.):

1. *WebM* – a library for managing WebModule, a WWW application, which allows viewing the statistics and sending the commands
2. *Manager* – a library used for the communication between the server module and the database.

Administrative Module (AM) is used for the communication between the Administrative Appliaction (AA) and the server through the protocol layer, using *AdmM* program interface. AA makes contact with the database by means of ODBC, but it still needs an interface for the SM server, if only for forcing the change of server's keys for e-signatures.

AA is also equipped with the *AdmA* interface for downloading the statistics from the database (Fig 10.).

VIII. Packet Voting System

To improve the link quality while using a parallel receipt, a system of packet comparison may be used (Fig 12.). Streams of data received from single ground stations are sent to the system. First, it is checked whether any of the streams is correct (e.g. by means of the control sum, contained in the stream). If none of the data streams is correct, the results are sent to the bit comparison system (Fig 11).

In this system, data in the form of single bits, gathered from various sources, as if votes to get the result. If in most cases bit 0 has been received, at the system output we get 0. If most of the source data is bit 1, the system output assumes the value of 1. The control sum is checked again in the output stream. If it is still incorrect, the system informs about a faulty receipt. If, however, the sum turns out to be correct, it means that the system has masked receipt errors and the data is transferred to the

Source packets	*011010010* *100101010* *100100110* *100110101* *010100010*
Result packet	*100100010*

Fig 11. Bit voting system.

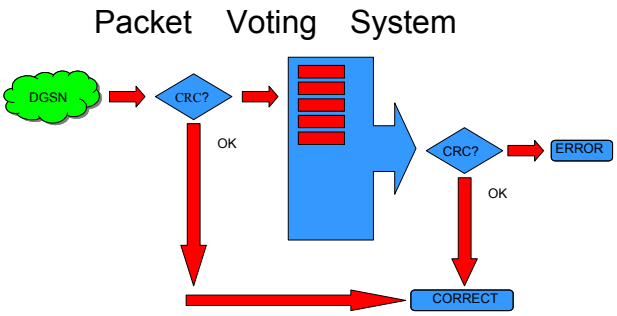

Fig. 12 Packet comparison system.

target system.

The author's previous research has shown that the appropriate transmission of a packet is crucial. The point is that the correct transmission of the header and the packet counter greatly facilitates the working of the system. Admittedly, the distortions in these parts of a packet can also be masked, but it highly increases the demands for computational power, required for checking consecutive variants of packet comparison. It seems that for normal work it is more practical to repeat the transmission of a given packet.

IX. The Method of Choosing TX and RX Stations

A separate discussion is required for the question of whether all the data received via the Internet should be treated equally. A bigger amount of trust should be given to those users who are characterized by a larger coefficient, and consequently achieve better results. The theoretical accessibility of a satellite for the stations, calculated on the basis of a mathematical model, should constitute a part of the coefficient. A similar mechanism should govern the process of choosing the station, which is to send a packet to a satellite. It is unacceptable for a couple of stations to transmit a packet simultaneously, since it would cause a collision in the radio channel. The authors suggest using similar coefficients for choosing the station that has the best chances of delivering the packet correctly.

X. The Flow and Storage of Data

The data transmitted between a satellite and a GS by using the radio module of the client application (RdM) undergoes buffering. This process is meant to increase the reliability of packet flow.

Buffer Order Packet (BOP) is used for gathering the commands from the server (Fig 9). It begins collecting the commands just after the SM server has commissioned to send one (still before adding the final APRS frame header with a given GS's parameters to the command). Such a solution is meant to prevent the loss of packets in case a transmitting device is malfunctioning when GSM has already received the instructions from the server to send the command.

Buffer Communiqué Packet (BCP) is used for gathering the received messages from a satellite (Fig 9). It begins collecting the messages just after having received them from the established port. Such a solution is meant to prevent the loss of received packets in case the GSM-SM connection is malfunctioning.

The sending of commands to a satellite is centralized, i.e. a user has to send such a command (IOP) to the server first (Fig 13). The server, depending on a given user's privileges and the current number of commands waiting to be sent, sends an answer to the user in the form of the ACK IOP packet. If the answer has been positive, the server, basing on the database statistics (a satellite's geographic coordinates in comparison to a GS, the number of ROPs, ACK SOPs and DOPs, the availability of a GS at a given moment, the number of broken CPs, etc.) chooses the best Selected Ground Station (SGS) for sending the packet to a satellite.

If after a defined period of ACK SOP Time Expire (ACK SOP TE) the chosen SGS does not confirm that the Send Order (SO) has been issued to a satellite, the server chooses the next SGS from the list of 'best' GSs. Such a situation may take place for instance when the chosen SGS has a broken transmitting device, and sending the SO to the satellite results in an error.

After a successful transmission of a given command to a satellite, the SGS awaits the confirmation from the server that the satellite has received the command. When the station receives such confirmation it alters the status of a given command in the buffer to "delivered". When the confirmation has not been received, the station repeats the command up to the Maximal Number of Trials (MaxNT) at Trial Time expire (TTe) intervals. After the Order delivered Time expire (OdTe) the SGS, although not certain whether the command has reached the satellite, alters the status of a given command in the BCP buffer to "expired", and does not attempt to send it again.

Only when the server receives the SO confirmation in the form of the ACK SOP packet from the SGS, does it end the iteration of an SGS search, and awaits the satellite's answer to the command in the form of the Communiqué Packet, CP.

If after OdTe from receiving ACK SOP from the station that sent the command to the satellite the server does not receive at least one CP – the satellite's answer to the command, containing the number of that command – it chooses the next SGS, until it receives a CP.

Finally, the server sends the confirmation of having received the satellite's answer to the command to the SGS whose identifier was contained in the CP, and to the user who commissioned the sending of a command. The DOP packet is the proof that the satellite has answered the command, and that this answer has reached the database.

The server is receiving the satellite's answers to the same command from many Received Ground Stations (RGS) until the end of Packet Voting System Time Expire (PVS TE) (Fig 14., CPs 1, 3 and 5).Those that fit into PVS TE are all loaded to the database. The remaining

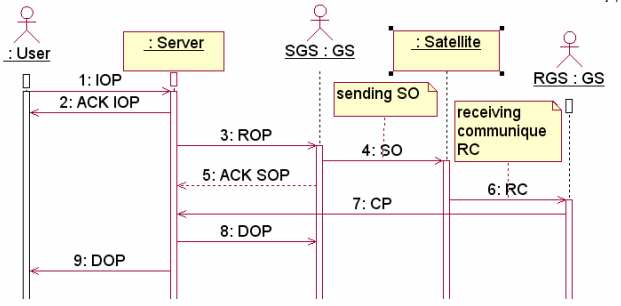

Fig. 13 The schema of sending a command to a satellite.

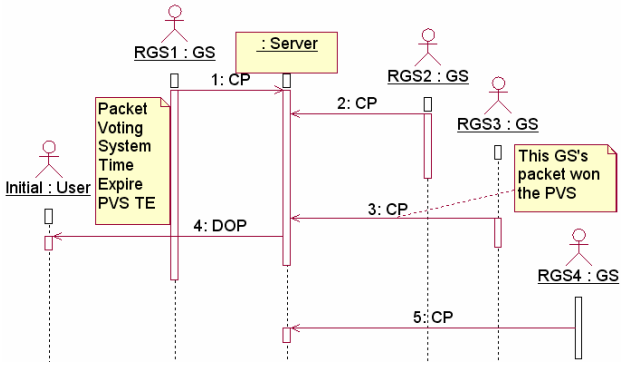

Fig. 14 The schema of satellite messages receipt together with the implementation of the Packet Voting System.

ones take part in the Packet Voting System.

The winning message appears in the database as the official answer of the satellite. The server sends a confirmation DOP packet to the user who initiated sending the command and to the station whose identifier is contained in the packet that won the voting, and then terminates the managing of a given message.

All sorts of SQL queries aiming at visualizing the data in the client application from a satellite's sensors, or the users or GSs activity statistics etc., are sent in the SQLP packets. In the Data field a one-byte identifier of a given query is sent. After receiving such a packet, the server issues a query with a given GS-SCS ID, and sends back the answer in the ASQLP packet. There is no possibility of sending DQL, DML or DDL in the content of a packet. At every moment of the connection with a client, the server accepts only packets of the types it is currently expecting. Such a solution increases the level of security, and prevents the server from being sent the destructive DDLs and DMLs.

XI. Summary

The authors have proposed a new realization of a satellite link, by using a system of distributed receipt. A mathematical model of a distributed satellite link has been presented, and it has been shown when such a system is superior to the one using the same number of antennas forming an antenna array connected to a single receiving station. Subsequently, a program realization of the system on consecutive layers of the IOSI model in the TCP/IP network has been introduced. The algorithms for DGSS network communication as well as gathering and presenting the data have been proposed. The practical verification of the system's mathematical model is currently taking place at the Warsaw University of Technology. Laboratory research and balloon missions are also being realized. The Distributed Ground Station System is also the subject of one of the experiments on the PW-Sat [5] satellite, which is being constructed by the students of the Warsaw University of Technology.

References

[1] "http://www.amsat.org", AMSAT website

[2] Michale O. Kolawole, "Satellite Communication Engineering," Marcel Dekker, Inc., New York 2002

[3] Daniel Józef Bem, "Telewizja satelitarna," Wydawnictwo Czasopism i Książek Technicznych SIGMA NOT, Spółka z o.o, Warsaw 1992

[4] Zdzisław Bieńkowski, "Poradnik Ultrakrótkofalowca," Wydawnictwa Komunikacji i Łączności, Warsaw 1988

[5] Grzegorz Niemirowski, "Cubesat microsatellite with balloon,"56th International Astronautical Congress in Fukuoka, October 2005

[6] M. Stolarski, W. Winiecki, "Building Distributed Ground StationWith Radio Amateurs," Conference materials from Space Technology Workshop STW 2006, Cracow, 23 May 2006, Poland, pp. 49-54

[7] M. Stolarski, "The Use of Distributed Ground Station System for very low power communication," CD-ROM conference materials from The 1st International Workshop on Ground Station Network Tokyo 2006, 18-19 July 2006, Japan

[8] M. Stolarski, „System porównywania pakietów jako metoda poprawiania jakości łącza satelitarnego w Rozproszonej Stacji Naziemnej," Proc. of VII Seminarium stypendystów Fundacji Wspierania Rozwoju Radiokomunikacji i Technik Multimedialnych, Warsaw 6 December 2006, Poland, pp. 35-42

[9] W.R.Stevens, „Programowanie zastosowań sieciowych w systemie UNIX", WNT, 1996

[10] W.R.Stevens, „Biblia TCP/IP", RM, 1998

[11] D.Cormmer, „Sieci komputerowe i intersieci", WNT, 2003

[12] A.J. Menezes, P.C. van Oorschot, S.A. Vanstone, „Handbook of Applied Cryptography", CRC Press, 1997, pp. 425-488

[13] N. Ferguson, B.Schneier, „Kryptografia w praktyce", Helion, 2004

[14] "http://www.mixw.net/", HAM Radio Software website

[15] "http://www.stoff.pl/", Sebastian Stoff's website on „Orbitron – Satellite Tracking System"

[16] "http://www.laboratoriomederos.com/CX6DD/wispdde/",a website on WiSPDDE software for controlling rotors and radios

[17] "http://www.aprs.org", Automatic Position Reporting System websait

[18] The APRS Working Group, „APRS Protocol Reference, Protocol Version 1.0.1", Tucson Amateur Packet Radio Corp, 2000