

INMARSAT SATELLITE DATA COMMUNICATION CHANNEL SIMULATOR

Sorin-Robertino Sinte

Technical Department, CERONAV, 69A, Baba Novac str., 900366 Constanta, Romania,
phone: +40-241-639595, sorin@ceronav.ro

This paper describes Inmarsat Satellite data communication infrastructure for a simulator. This communication channel implements GMDSS (Global Maritime Distress and Safety System) and data transfer functions between INMARSAT-C compatible terminal and a Land Earth Station (LES). Thrane & Thrane compatible software, installed on embedded controller is used for simulation of Inmarsat terminal. This controller transfer data to simulated Inmarsat satellite using a wireless network. The satellite functions are implemented into another embedded controller. This satellite communication controller is linked with Inmarsat terminal via wireless network and with another land earth station using a secondary wireless channel. The land earth station is software implemented in a PC linked with simulated satellite using a wireless network. This center is able to manage messages received from Inmarsat terminal, via simulated satellite, and convert them into short text messages, automatically delivered to local network workstations, in faxes format or mail messages. The system can be consists as base for a GMDSS simulator.

Keywords: satellite communications, maritime data communications

1. INTRODUCTION

In debugging process of GMDSS (Global Maritime Distress and Safety System) radio terminal we need to test messaging and DSC (distress call alert) message passing between Inmarsat-C terminal and terrestrial operator. For solving this problem we implement a tool for testing delivery of messages from Inmarsat-C terminal to a terrestrial end-point without connection with satellite communication channel.

Also this tool can be used as infrastructure for a GMDSS message passing simulator. The system is able to manage 15 Inmarsat-C compatible terminals, one LES (Land Earth Station) center, and one simulated satellite system.

The structure of system is presented in figure Fig. 1 it is consists of three levels:

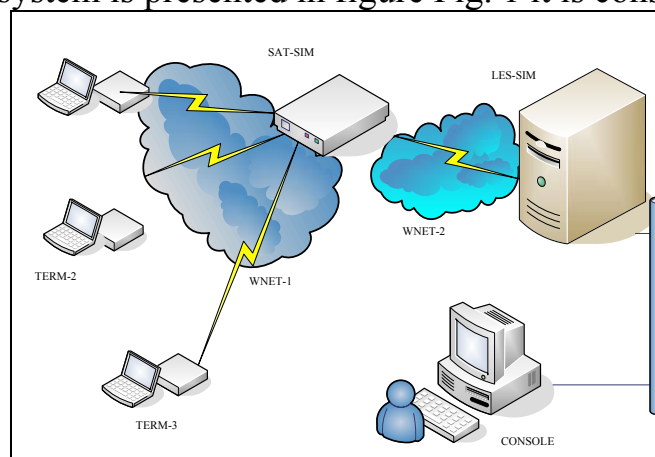


Fig. 1

1. first level is represented by INMARSAT-C terminals, connected with communication controller (notated with TERM-1...TERM-3 in figure Fig. 1).

2. second level is represented by Inmarsat-C simulated satellite (SAT-SIM notation in Fig. 1).

3. and the third level is represented by simulated LES system (LES-SIM notation in Fig. 1) and the operator console connected to LES computer.

Between levels we have wireless connections. First wireless connection (WNET-1 notation in Fig. 1) is a small network communication controllers (TERM-1...TERM-3) and simulated Inmarsat satellite (SAT-SIM). This wireless network uses same frequency channel for all connections (433.1580 MHz). The second wireless connection (WNET-2) is a wireless direct link between the simulated satellite (SAT-SIM) and LES system (LES-SIM). This second network use another frequency channel (433.3020 MHz). The each wireless network doesn't disturb communication of the other one (the communication channels are aliens).

2. SYSTEM COMMUNICATION PROTOCOL

The simulated network protocol is nearby Inmarsat communication protocol. The

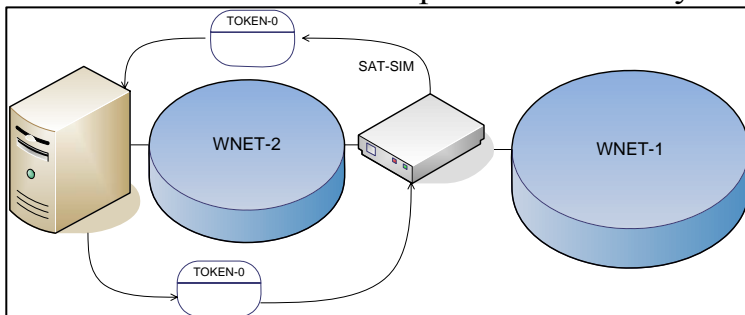


Fig. 2

both networks use a Token-Ring / ALOHA like communication protocol. Each network is consists of ring communication protocol. The protocol is controlled by LES center. The LES centers initiate a token in his network (WNET-2),

named TOKEN-0. The token will be delivered by satellite simulator (SAT-SIM) into the second network (WNET-1) (see Fig. 2). Each communication controller listen the frequency channel (in his network). If no controller will request to communicate with satellite, after a time the satellite will reply to LES center with same token TOKEN-0. In this manner the ring will be closed. If a communication controller need to login on satellite, after receiving the token TOKEN-0,

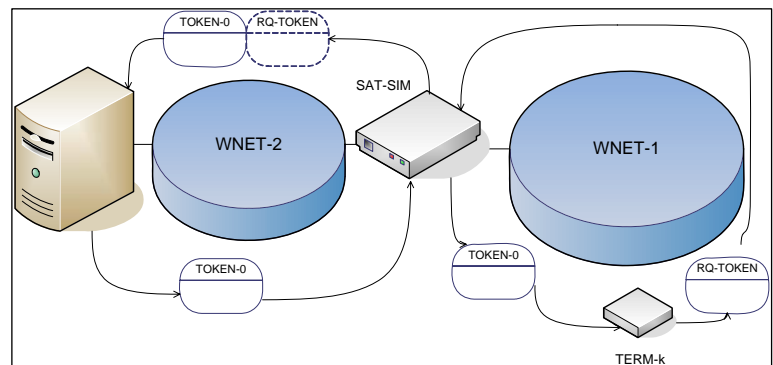


Fig. 3

the controller will send in his network a request channel frame named RQ-TOKEN (Fig. 3). The satellite will process this request and forward it to LES center. The LES center will insert a new communication token in his chain. On receiving of this token the communication controller will be able to send data to LES center. For disconnecting of communication controller from his network, the controller must send a RQ-LOGOUT, or if controller doesn't respond for a period equal with $5 \times \tau$,

where τ is period of token TOKEN-0 to travel the ring (at LES communication interface level).

3. SYSTEM STRUCTURE

The system structure is composed from:

1. communication controller (TERM-1 ... TERM-3),
2. simulated satellite subsystem (SAT-SIM),
3. simulated LES system (LES-SIM).

3.1 Communication controller

The hearth of communication controller is a microcontroller (TERM_k in Fig. 1) with two serial compatible RS232 interfaces. One serial communication interface will be connected with Inmarsat-C terminal (which is consists of a CAPSAT software installed on a embedded 386SX CPU board) – this serial will be notated with ConCom- and the other one is connected with a wireless serial data radio module – notated with NetCom-. The serial radio module is configured to work in 433.1580 MHz (frequency adopted for WNET-1 network).

The communication controller software is consists of two threads:

- one is responsible with network communication (NetworkThread) and
- the other one is responsible with terminal communication (CommThread).

The implementations of these threads are:

```
NetworkThread () {
    if (Event'Timmer [Schedule_for_Testing_Satellite_Time_Out]) {
        Satellite.Status = 0;
    }
    if (Event'NetCom.Rx) {
        switch (NetCom.Rx.Type) { // any token
            case TOKEN-0:
                if (! NetCom.IsLink && Satellite.LogonRq) {
                    NetCom_Send (RQ_TOKEN);
                    Satellite.LogonRq = 0;
                }
                break;
            case TOKEN-1:
                if (Satellite.Data.Size)
                    NetCom_Send (Satellite.Data);
                break;
        }
        Satellite.Status = 1;
    }
}
```

```
CommThread () {
```

```

    if (Event'ConCom.Rx) {
        switch (message = ConCom_Read()) {
            case 'LogonRequest':
                Satellite.LogonRq = 1;
                break;
            case 'LogoffRequest':
            case 'Data':
                if (Satellite.Status)
                    Satellite.Data = message;
                break;
        }
    }
}

```

3.2 Simulated satellite subsystem

The base of simulated satellite subsystem is a microcontroller with two RS232 compatible serial interfaces (SAT-SIM in Fig. 1). Both serial channels will be connected to data radio modules. First RS232 serial interface, is named Net1Com, and will be linked with a radio module, on WNET-1 network - radio module is configured to work on 433.1580 MHz. The secondary communication interface is named Net2Com and will be linked with another data radio module, configured to work on 433.3020 MHz, in WNET-2 network.

The simulated subsystem is managed by a single thread SatelliteThread described below:

```

SatelliteThread () {
    if (Event'Timmer [Schedule_Testing_Time_Out]) {
        LES.Online = FALSE;
        for each TERM_k do {
            TERM_k.Online = FALSE;
        }
    }
    if (Event'Net2Com.Rx) {
        switch (message = Net2Com_Get ()) {
            case 'TOKEN_0':
            case 'TOKEN_k':
                Net1Com_Send (message);
                LES.Online = TRUE;
                break;
            case data:
                k = message.to;
                if (TERM_k.Online) {
                    Net1Com_Send (message);
                }
        }
    }
}

```

```

    }
    if (Event'Net1Com.Rx) {
        switch (message = Net1Com_Get ()) {
            case 'TOKEN_k':
                k = message.from;
                TERM_k.Online = TRUE;
            case 'TOKEN_0':
                Net2Com_Send (message);
                break;
            case data:
                if (LES.Online) {
                    Net2Com_Send (message);
                }
        }
    }
}

```

3.3 Simulated LES system

This system (LES-SIM in Fig. 1) is implemented on a compatible PC-AT, with a Windows XP/2000/NT operating system installed on it. The computer has a serial RS232 interface installed on it (on board or via a USB bus system) – this interface is notated with SERCOM device -. Using this interface the system is linked with WNET-2 network, via a data radio module (programmed to work on 433.3020 MHz frequency channel). Also the system has Ethernet network card plug, for connection with a local are network. The LES functions are implemented in a windows task named LesCenter. The algorithm of this task is described below:

```

LesCenter () {
    static integer LastStation = 0;

    if (Event'Timer [Token_Schedule]) {
        k = LastStation;
        if (LastStation == 0) {
            Satellite.IsOnline = FALSE;
            SerCom_Send ('TOKEN_0');
        }
        else if (Station[k].IsOnline) {
            SerCom_Send ('TOKEN_k');
        }
        LastStation ++;
    }
    if (Event'SerCom.Rx) {
        switch (message = SerCom_Get ()) { // something received from serial
            case 'TOKEN_0':
                Sattelite.IsOnline = TRUE;

```

```

        break;
    case 'TOKEN_k':
    case message.IsLogon: // Logon Request
        k = message.from;
        Station[k].IsOnline = TRUE;
        break;
    case message.IsLogoff: // Logoff Request
        k = message.from;
        Station[k].IsOnline = FALSE;
        break;
    case message.DSC: // distress call received
        k = message.from;
        log ("DSC message from station k");
        for each k from Station[1..n-k].IsOnline
            SerCom_Send (DSC for station k);
        eMail ("les@localhost", "DSC for station k");
        break;
    case data: // normal message received
        log (message);
        if (message.to is 'INMARSAT Station') {
            SerCom_Send( message, message.to);
        } else {
            eMail (message.to, message);
        }
        break;
    };
}
}

```

4. CONCLUSIONS

This kind of center is able to manage messages received from Inmarsat terminal, via simulated satellite, and convert them into short text messages, automatically delivered to local network workstations, in fax format or mail messages. The system can be considered as base for a GMDSS simulator.

5. REFERENCES

- [1] Pelton J.N., A.U. Mac Rae, K.B. Bhasin i.e., *Global Satellite Communications Technology and Systems*, International Technology Research Institute, pp. 89-112, pp169-170, 1998.
- [2] Elbert B.R., *The Satellite Communication. Applications Handbook*, Second Edition, Artech House Inc, pp. 84-93, 2004.
- [3] ***, *Manual for use by the Maritime Mobile and Maritime Mobile-Satellite Services*, International Telecommunication Union, Radio Communication Bureau, pp. 81-96, pp 267-271, pp. 385-427, pp. 567-569, pp 571-574, 2005.
- [4] ***, *HAC Data Radio Modules*, Shenzen Hac Technology Co. Ltd, 2006.