Telemetry Data Acquisition and Analysis in Integrated Baseband System Based on TCP/IP Protocol

Tao Zhang¹, a, Yongsheng Gui¹
¹Beijing Aerospace Control Center Beijing, China
azt0530@126.com

Index Terms: TCP/IP; Cortex CRT; Baseband; Telemetry

Abstract. This paper briefly introduced the integrated Command Ranging & Telemetry (CRT) baseband system of a satellite TT&C ground station, and a software was developed with C++ Builder 6.0 to acquire and store the important telemetry data. It effectively resolved the problems of data acquisition and storage for important parameters from baseband system, and it was applied to fault analysis and diagnosis of satellite signals.

I. INTRODUCTION

A satellite TT&C station in Beijing has accomplished dozens of LEOP (Launch Early Orbit Phase) missions since 1988, and it is highly praised by foreign party. At present, this station has become an important part of Inmarsat global TT&C network. The integrated baseband system is one of the most important systems in the whole ground station, equipments on the ground should send telecommand instructions, digital voice, and receive voice, telemetry, data of televisions, images and payloads from satellites, besides accomplish tracking function. The idea of integrated baseband system is to organically integrate ranging, telecommand, telemetry and so on, then the equipments on the ground can be integrated. All the signals whose frequencies are below 70MHz are processed in a box with computer bus platform, and they usually include intermediate frequency (IF) modulator, IF receiver, telemetry simulator, ranging, telemetry and telecommand.

Cortex CRT is adopted by this station because it is the third generation of integrated Command Ranging & Telemetry baseband systems designed and manufactured since 1990. Its missions include station keeping, LEOP, factory and pre-launch testing, and it can be applied to GEO/LEO, three axis or spin stabilized satellites, and its main features are listed as below:
1) PC-based architecture with Windows operating system.
2) User-friendly and intuitive Graphic User Interface.
3) High integration with drastically reduced hardware for increased availability.
4) Enhanced performances, upgradeability and flexibility due to extensive use of digital signal processing techniques.
5) No tuning, no preventive maintenance.
6) ESA/SLE interface available.

Its hardware architecture is shown as below:

The Cortex CRT communicates with the Control Center via an Ethernet port using TCP/IP protocol. This port is used for exchanging telemetry, telecommand, ranging, monitoring and control, Doppler, logging, … data as well as file transfer (for software upgrade and remote maintenance operations), in a multi-clients environment.

The built-in Graphic User Interface uses the same communication interface as the Control Center, allowing full control of any Cortex CRT from any PC workstation connected to the network. But the client software can only display pure digits, and the data cannot be stored on the computer, then the faults of antennas cannot be analyzed and diagnosed according to it. Finally, this software is designed to resolve the problem.
II. BRIEF INTRODUCTION OF TCP PROTOCOL

TCP is a connection-oriented, end-to-end reliable protocol designed to fit into a layered hierarchy of protocols which support multi-network applications. The TCP provides for reliable inter-process communication between pairs of processes in host computers attached to distinct but interconnected computer communication networks [1]. Very few assumptions are made as to the reliability of the communication protocols below the TCP layer. TCP assumes it can obtain a simple, potentially unreliable datagram service from the lower level protocols. In principle, the TCP should be able to operate above a wide spectrum of communication systems ranging from hard-wired connections to packet-switched or circuit-switched networks [2].

The TCP fits into a layered protocol architecture just above a basic Internet Protocol which provides a way for the TCP to send and receive variable-length segments of information enclosed in internet datagram "envelopes"[3]. The internet datagram provides a means for addressing source and destination TCPs in different networks. The internet protocol also deals with any fragmentation or reassembly of the TCP segments required to achieve transport and delivery through multiple networks and interconnecting gateways. The internet protocol also carries information on the precedence, security classification and compartmentation of the TCP segments, so this information can be communicated end-to-end across multiple networks [4].

The process of TCP communication is shown as below:

---

**Fig.1 Hardware Architecture of Cortex CRT**

**Fig.2 Process of TCP communication between server and client**
III. DATA ACQUISITION IN INTEGRATED BASEBAND SYSTEM

The Cortex NT series acts as a TCP/IP server for all types of data transfer. In this case, data are exchanged using the connected-sockets communication protocol.

Some monitoring data flows including the telemetry data flow accept multiple clients at a time, while other flows accept only one client at a time.

Each type of transfer flow is allocated a specific port number which the clients use for connecting themselves to each service.

All TCP/IP messages exchanged between the Cortex product and the host computer(s) connected to the LAN port, are prefixed by a 12-byte message header of the following structure:

<table>
<thead>
<tr>
<th>TABLE I. STANDARD TCP/IP MESSAGE HEADER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offset</td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE II. STANDARD TCP/IP MESSAGE POSTAMBLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offset</td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td>---</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE III. STANDARD MONITORING REQUEST MESSAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offset</td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td>0 to 2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE IV. STANDARD MONITORING MESSAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offset</td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td>0 to 2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4 to N-2</td>
</tr>
<tr>
<td>N-1</td>
</tr>
</tbody>
</table>

All TCP/IP messages exchanged between the Cortex product and the host computer(s) are terminated by a 4-byte message postamble of the following structure:

In this mode, the Cortex product only supplies its status at the request of a remote host computer. On reception of a monitoring request, the CORTEX product checks its validity. If the request is correct, it returns a variable-length message, embedded in a TCP/IP packet. In case of invalid request, the Cortex product returns a negative monitoring message. A monitoring request is a 20-byte message embedded in a TCP/IP packet whose structure is as follows:

If the monitoring request is correct the following message is returned to the Monitoring Client:

In the integrated baseband system of this station, there are 5 Cortex CRT products correspondingly connected to 5 antennas. Each one is a TCP server. By using the existing MCS(Monitoring & Control System) network, a TCP client program is developed with C++ Builder 6.0 on a operating
computer to get important data, such as Eb/N₀ and carrier offset of telemetry unit, from 5 Cortex CRT products by continually sending requests to each one and store the data every seconds. The main process is shown as below.

Firstly, data structure of request messages is defined according to Cortex protocol.

```c
typedef struct
{
    int MsgStart;
    int MsgLen;
    int Flowid;
    int ReqTag;
    int MsgEnd;
} RequestMsg;
```

Secondly, the value of request message is set according to the parameters you want.

```c
RequestMsg iRequestMsg;
iRequestMsg.MsgStart=htonl (1234567890); //header of the message
iRequestMsg.MsgLen=htonl (sizeof (iRequestMsg)); //length of the message, in this case, equals 20
iRequestMsg.Flowid=htonl (2); //flow id
iRequestMsg.MsgEnd=htonl (-1234567890); //standard TCP/IP postamble
iRequestMsg.ReqTag=htonl (0x1062); //Component code of TMU-C is 0x1062
```

```c
sockdata =socket(AF_INET,SOCK_STREAM,0);
server.sin_family=AF_INET;
server.sin_port=htons (3000); //monitoring port of Cortex CRT
server.sin_addr.S_un.S_addr=inet_ addr ("161.30.99.207");
connect (sockdata,(sockaddr*)&server,sizeof(server));
//connect to server
```

After the connection between the server and client is ready, the request message can be sent to the Cortex CRT product, and then immediately the data that you want can be received, we can parse the data according to the protocol to get the values of the required parameters. It should be noticed that the data we receive from Cortex CRT is reverse in byte-orders.

**IV. DATA ANALYSIS AND CONCLUSION**

The value of Eb/N₀ represents the signal-to-noise ratio, generally, when the telemetry from a satellite is normal, the signal should be much greater than noise, otherwise, when the signal is less than noise, the signal is submerged into the noise, and it can’t be received and demodulated, then the telemetry from the satellite is lost.

In this project, the value of Eb/N₀ in db is acquired and stored. For example, the Eb/N₀ curves of an antenna during the same period of time in two different days are shown as below:

![Fig.3 Eb/N₀ curve of normal tracking](image1)

![Fig.4 Eb/N₀ curve of abnormal tracking when telemetry is lost](image2)
In Fig.3, we can see the characteristic curve of $E_b/N_0$ value in db in case of normal tracking satellite. Generally, the $E_b/N_0$ value in db is between 4 and 35, and then we can ensure that the signal is strong enough to be demodulated.

In Fig.4, we can see when the telemetry is lost, the $E_b/N_0$ value in db is suddenly dropped to -127, and when the signal slowly grows up, the $E_b/N_0$ value in db returns to the normal level again. From the two figures above, we can see the changes of $E_b/N_0$, if it happens that the telemetry is lost or other abnormal cases happen, we can find the fault from the curve, and then it can be used to analyze and diagnose the antenna’s or satellite’s faults. This software is running well and stably in actual application, it will be a useful auxiliary tool of the baseband system.

ACKNOWLEDGMENT

During my work on the project, I received great support from my colleagues, directors and my family. I should thank all my colleagues and friends for their helps, especially, I should thank my wife and my new-born baby, without their support, I can’t accomplish my work.

REFERENCES