

# **TCP/IP Remote Control of a Ground Station**

**Dale P. Massey  
Universal Space Network, Inc.  
417 Caredean Drive  
Horsham, PA 19044  
215-328-9130**

## **ABSTRACT**

Satellite tracking ground stations are under continuous pressure to automate. Autonomy is generally the desired goal, but if the ground stations are in a Commercial Ground Network (CGN) setup to support many missions simultaneously, remote control of such stations is of much more importance. The proliferation of Low Earth Orbiting (LEO) science, earth resources and eventually global communications satellites either in orbit or planned, requires a much lower cost methodology for ground support. A CGN of TCP/IP remotely controlled ground stations lowers much of the manpower that was historically required to operate such stations. This paper will cover the remote control aspects needed for a satellite ground tracking station and offer a unique remote control topology utilizing TCP/IP.

## **KEYWORDS**

Object-oriented design, control and management object.

## **INTRODUCTION**

What is remote control of a ground station? Remote control, as used in the context of this paper, is the ability to control, configure, and receive status from, all ground station equipment over a network. The network can be a Local Area Network (LAN), Wide Area Network (WAN) or open Internet. To attain this level of remote control several requirements must be levied upon the implementation. These requirements are as follows; single point access, remote equipment control, status, and equipment scheduling interface to all ground station equipment. An additional condition must be imposed on the previous requirements, which is access via a TCP/IP network. Expanding the problem to the next level higher creates a new dimension of problem of how to control each Remote Ground Station (RGS) and all of its respective equipment. Solutions to this problem using

conventional methods are inflexible and usually result in recurring development costs to add another RGS. A more practical and cost effective solution was provided with the application of object-oriented design.

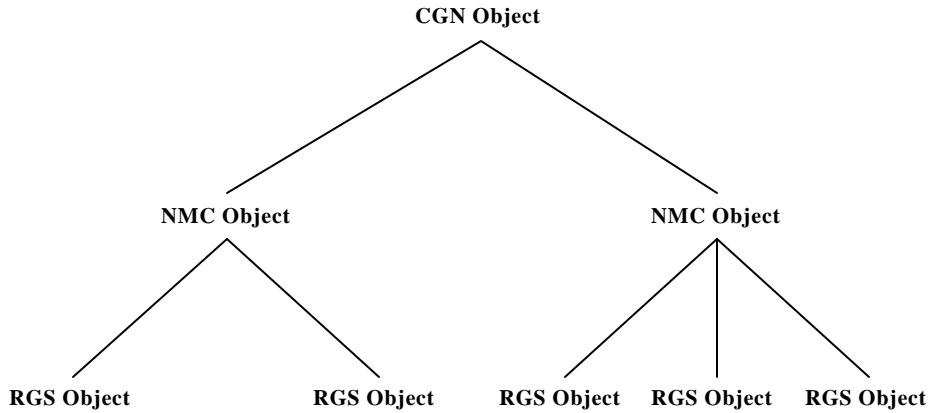


Figure 1- CGN Remote Control Topology

## REMOTE CONTROL TOPOLOGY

Using object-oriented design, every part of the CGN is treated as an object (see Figure 1). Each CGN object is responsible for control and management of the Network Management Center (NMC) objects. At the next lower level a NMC object would be responsible for control and management of the RGS objects. At the lowest level, a RGS object would be responsible for the control and management of all equipment objects (see Figure 2). An equipment object provides the control and management interface to a specific piece of equipment.

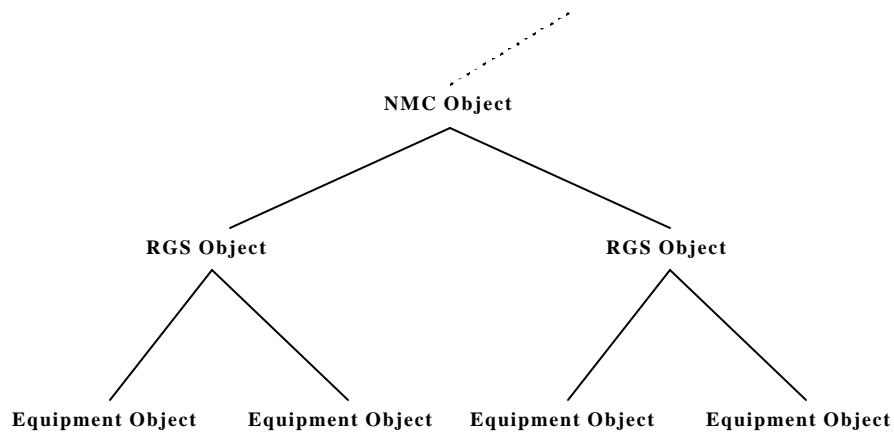
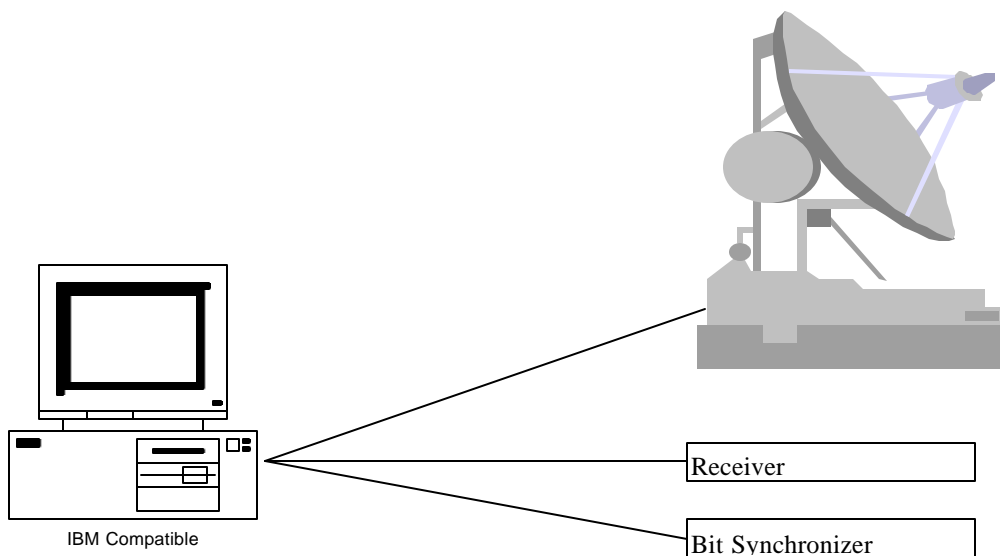


Figure2- CGN Remote Control Topology(Cont.)

## SINGLE POINT ACCESS

Single point access is the ability to access all RGS equipment via a single interface. Control and management of all aspects of an RGS requires the mapping of all ground station equipment to a single entry point. The RGS equipment is physically interfaced to a single computer providing physical single point access.(see Figure 3).



**Figure 3 - RGS Equipment Interface**

The logical interface is implemented using a control and management object called the RGS object. The RGS object maintains an equipment table, where each entry contains a name, an enumerated equipment type identifier, a unit number, an equipment state(i.e. Available, Allocated, etc.), and a pointer to an equipment object. An equipment object is a control and management object, which provides a command set and handles all specific communication protocol issues necessary for communicating to the physical piece of equipment. It could be considered a logical piece of equipment. One equipment object is created for each piece of equipment defined in the RGS configuration file.

BitSync1	COM2 [9600 n 8 1 1 HS_ON]
SignalGenerator1	COM3 [9600 n 8 1 1 HS_OFF]
Receiver1	COM4 [9600 n 8 1 1 HS_OFF]
Antenna1	COM6 [9600 n 8 1 1 HS_OFF]

**Figure 4 - Sample RGS Configuration File**

The RGS configuration file is a map that defines the correlation between the physical and the logical equipment. It contains entries in the following form that defines a name, unit number, communications port(i.e. RS232 port, Socket, etc.), and communication

parameters(Figure 4). The name and unit number define a unique identifier for the piece of equipment.

The RGS object defines several high level functions. These functions are Allocate, Configure, Start, Stop, Transfer Data, Release, GetStatus, and Execute. Each function translates to an equipment object function that performs the equipment specific operation. The Allocate function uses a unique pass identifier to allocate the piece of equipment for a satellite pass. The Configure function uses a satellite identifier to determine the required configuration parameters for the specific equipment object. The Start and Stop functions are used to begin and end the equipment operation. The Release function removes the pass identifier from the equipment table and marks the equipment state as Available. The GetStatus function returns status for all equipment allocated for a specified pass identifier. The Execute function will be covered in the next section.

## **REMOTE EQUIPMENT CONTROL**

Equipment objects have a single point interface similar to that of the RGS object. All functions in an equipment object have enumerated function identifiers. These identifiers are used by the equipment object function called Execute. This function translates the function identifiers into an actual equipment specific function calls.

The RGS object's Execute function uses an equipment identifier, unit number, and function identifier to access a specific equipment object function. Using the RGS object's Execute function, graphical user interfaces can be created for each specific equipment type. This interface provides the facility for modifying equipment parameter in real-time.

## **EQUIPMENT STATUS**

RGS equipment status is obtained using the RGS object function GetStatus. This function allows access, restricted by pass identifier, to the equipment status. This interface returns an array of structures specifying equipment status and a composite RGS status.

## **EQUIPMENT SCHEDULING INTERFACE**

The Equipment Scheduling Interface is a task scheduler dedicated to the control and management of the RGS equipment. The task scheduler executes a pass schedule received from the NMC. The pass schedule file contains several primitive commands for allocation, configuration, etc. of ground station equipment(see Figure 4). These commands translate directly into the RGS object high level functions Allocate, Configure, Start, Stop, Release. Once a schedule file is received, the RGS can execute the scheduled pass without NMC intervention.

Space Network Schedule

Created 10/28/97 at 22:22:02 on NMC1

Start Time: 09-Mar-1998 00:00:00.000

Stop Time: 29-Oct-1997 23:59:59.000

Duration: 1.99999 Days

Begin Tasks # Begin

Satellite COBE

RGS Horsham

# Allocate RGS equipment for pass

09-Mar-1998 11:20:01.000 Allocate Antenna1  
09-Mar-1998 11:20:02.000 Allocate BitSync1  
09-Mar-1998 11:20:03.000 Allocate Receiver1  
09-Mar-1998 11:20:04.000 Allocate SignalGenerator1  
09-Mar-1998 11:20:05.000 Allocate TelemetryProcessor1

# Configure RGS equipment for pass

09-Mar-1998 11:20:10.000 Configure Antenna1 COBE1028972352.pas  
09-Mar-1998 11:20:20.000 Configure BitSync1  
09-Mar-1998 11:20:30.000 Configure Receiver1  
09-Mar-1998 11:20:40.000 Configure SignalGenerator1  
09-Mar-1998 11:20:50.000 Configure TelemetryProcessor1

# Start antenna tracking

09-Mar-1998 11:21:00.000 Start Antenna1  
09-Mar-1998 11:21:05.000 Start SignalGenerator1

# SCHEDULED PASS-

# START: 09-Mar-1998 23:52:07.000

# END: 09-Mar-1998 23:59:07.000

# DURATION: 7 minutes

# Stop RGS equipment

09-Mar-1998 11:22:00.000 Stop Antenna1  
09-Mar-1998 11:22:10.000 Stop TelemetryProcessor1

# Transfer data

09-Mar-1998 11:22:11.000 TransferData TelemetryProcessor1

# Release RGS equipment

09-Mar-1998 11:23:00.000 Release Antenna1  
09-Mar-1998 11:23:02.000 Release BitSync1  
09-Mar-1998 11:23:03.000 Release Receiver1  
09-Mar-1998 11:23:05.000 Release SignalGenerator1  
09-Mar-1998 11:23:07.000 Release TelemetryProcessor1

End Tasks # End

**Figure 4 - Sample Schedule**

## TCP/IP RGS SERVER

The RGS server program creates the task scheduler and RGS objects. The server program provides remote access via a TCP/IP network. The NMC communicates with the server program using messages. A TCP/IP socket is created to handle each unique message type. Each message contains a unique type identifier. This identifier allows the message to be routed to the appropriate part of the server program for processing. The message types map directly to the following objects; the Equipment Scheduling Interface, Remote Equipment Control, and Equipment Status.

## CONCLUSION

In conclusion, the creation of the single point access, remote equipment control, status, and equipment scheduling objects provided a simplified interface to all ground station equipment.

Consolidating these objects in a TCP/IP server provides the desired network availability.

## REFERENCES

1. Coad, Peter and Nicola, Jill., Object-Oriented Programming, YourDon Press, Englewood Cliffs, NJ., 1993
2. Gamma, Erich, Helm, Richard, Johnson, Ralph, and Vlissides, John., Design Patterns, Addison Wesley Press, Reading, MA., 1995
3. Lippman, Stanley B., C++ Primer 2<sup>nd</sup> Edition, Addison Wesley Press, Reading, MA., 1991

## ACRONYMS

CGN	Commercial Ground Network
LAN	Local Area Network
NMC	Network Management Center
RGS	Remote Ground Station
TCP/IP	Transmission Control Protocol/Internet Protocol
WAN	Wide Area Network