NEAR-REAL-TIME STATUS EXCHANGE

Werner Gurtner

Astronomical Institute, University of Berne, Switzerland

Abstract

For several years now up to about 10 mostly European laser stations have been exchanging their tracking status, especially the current time bias of successfully tracked satellites, in real-time. The use of this information can help stations to acquire satellites with poor orbit predictions more easily and more rapidly, and it can be the basis for a more sophisticated coordination of the satellite tracking The paper describes the communication protocol, the message format, and the utilities needed to participate in this program.

1. Introduction

The ILRS laser stations in Europe are located in a cluster relatively close together. A consequence of this geometry is the fact that very often part or all of the European stations will track the same satellite simultaneously or in rapid succession.

Occasionally the orbit predictions for some satellites can be fairly poor, and it might be rather difficult and time-consuming to get first successful range observations by searching the area in the sky around the predicted positions.

The largest component of the prediction error tends to be along the orbit. Therefore the station operators will mainly search for the satellite in the along-track direction. The along-track error is equal to a time error, i.e., the satellite is either early or late within the predicted orbit. Comparing the first successful range observations with the predicted ranges and taking into account the geometry between the satellite orbit and the tracking station it is easy to compute the apparent along track error in time, assuming that the observed difference is generated by an along-track error, only. Alternatively a station could apply a series of controlled time biases to its tracking procedure until it gets successful returns from the satellite and note the actual time bias.

This prediction error could be transmitted to all other stations nearby to be used for a faster acquisition of the respective satellite.

Simultaneously tracking the same satellite by many stations nearby might not be the optimum scenario if another satellite happens to pass the area in the same time. It might be more effective if some stations concentrated their tracking to this other satellite, especially if due to geometry or prediction quality acquisition was difficult and did not allow for rapid pass interleaving. If the information about the current tracking status were shared among the stations, real-time decisions about changes in the tracking priority could be easily taken by the operators.

Several years ago we proposed and successfully implemented a status exchange scheme and server program at the Zimmerwald station computer as described in the following sections.

2. Architecture of the Status Exchange

2.1 Client-Server Concept

A simple solution for an organized status exchange among an arbitrary number of stations is a client-server concept:

- Each station (the client) sends its status to a server
- The server collects the individual status messages and merges them into a simple table
- Each station receives in return the tabulated status information

2.2 Communication Protocol

Rather simple and easy to use for communication between computers is the Internet TCP/IP protocol: Information (e.g., character strings) can be sent to another computer by specifying

- the IP address (a number or a name, e.g., `130.92.25.24` or `aiuas3.unibe.ch`) and
- the so-called port number on which the receiving computer expects to get the data

to specific communication subroutines available for many programming languages and most operating systems (often assembled in the *socket library*).

On the other end of the communication line the target computer receives the messages on the specified port by means of the respective communication subroutines.

Usually the connection between two computers is established in a client-server mode:

- The server starts to listen on the specified server port for incoming connection requests
- As the need arises a client asks the server to establish a connection for data transmissions
- Request granted and connection established either one of the two partners can start sending data, depending on the rules previously specified and agreed upon.

After completion of the planned data transmission (one or more uni- or bilateral exchanges of data messages) the connection can be stopped either by the client or the server.

The communication rules for our implemented status exchange are as follows:

1. The communication between a client (an SLR station) and the server starts with the request by the client
2. The connection is established by the server
3. The server accepts status messages from the client anytime whenever the connection is open
4. The client (station) generates status messages periodically, e.g., every 15 seconds and sends them to the server
5. The server sends the compiled status message back to the client every 15 seconds as long as the connection remains open
6. The server can simultaneously establish connections to a number of different clients (currently 50).
7. Individual status messages consist of an ASCII character string delimited by a line-feed character (see below).

8. The compiled status message consists of a series of individual status messages plus a final record of a string of dashes, also delimited by a line-feed character.
9. The client may leave the connection open for as long as it likes or can close it after each successful transmission cycle.
10. Messages older than 30 minutes are discarded by the server.

As there is no need for a client to actually send any status messages to the server the connection can also be used by anybody to just receive the compiled status messages for monitoring purposes.

2.3 Format of the Status Messages

The status messages contain

- the station name (no pre-defined names)
- the current date and time (UTC)
- the satellite currently tracked (names according to the standard time bias file distributed through the same channels as the satellite orbit prediction, i.e., no hyphens, blank space or underscores, first letter uppercase, except names like ERS2, GFZ1)
- the station status:
  - NXT : Pass to come next, the system is waiting for this pass
  - CUR : Currently tracking
  - LST : Most recent pass, no new pass initiated
  - CAL : Calibration
  - OUT : Not operating
- the number of successful returns
- the used set of satellite predictions (label formed from an acronym for the prediction center and the sequence number of the current prediction set, see also in the standard time bias file)
- the current time bias in seconds (w/r to the distributed predictions) applied or determined in real-time from the observed ranges

```
Graz          2004-06-21 07:11:00  Topex      CUR  9786  HON172 -0.005
Zimmerwald    2004-06-21 07:11:07  Topex      CUR   384  HON172 -0.006
Potsdam       2004-06-21 07:11:01             OUT
Wettzell      2004-06-21 07:11:06  Topex      CUR   124  HON172 -0.004
MLRO-Matera   2004-06-21 07:11:07             OUT
Ftlrs_Sfdo    2004-06-21 07:05:00             OUT
Grasse_slr    2004-06-21 07:02:12             OUT
Herstmonceux  2004-06-21 07:10:50  Topex      CUR   391  HON172 -0.005
Yarragadee    2004-06-21 07:11:04             OUT
-----------------------------------------------------------------------
```

Table 1: Example of a compiled status message

Table 2 contains the format of all data fields of a status message (in Fortran notation).

```
A13           I4   I2 I2 I2 I2 I2 A10        A3  I5    A6     F6.3
------------- ---- -- -- -- -- -- ---------- --- ----- ------ ------
station       date       time     satellite  stat obs  irvset t-bias
              yyyy-mm-dd hh:mm:ss                      cccnnn
```

Table 2: Format of a status message

2.4 Generation of the Status Message

The client has to generate periodically (e.g., every 15 seconds) the current status of the station and to prepare and send the status message as described in sections 2.3 and 2.5. Of course this is only possible if such a program has access to the respective system data automatically and in real time. Depending on the procedures used for the data transmission the program might store the message intermediately for later use, pipe it into the communication program or send it directly to the server.

The time bias should refer to the prediction set as declared in the status message, i.e., it should be the sum of any a priori time bias used and the additional time bias as determined from the range observations or as applied to the introduced range predictions.

2.5 Status Exchange

Messages are to be sent to port `7810` at the server `aiuas3.unibe.ch`.

There are several possibilities for a client how to actually send and/or receive messages:

Receive only:

`Telnet` to port `7810` can be interactively invoked. Messages will be received and displayed or sent to the standard output as long as the `telnet` connection is alive:

```
telnet aiuas3.unibe.ch 7810
telnet aiuas3.unibe.ch /port=7810
```

Send and receive:

- Write your own program (C language: Use the *socket* libraries)
- Use the program `eurostat`, see chapter 3, below.

2.6 Exchange of General Text Messages

Each station may also upload a text message line (< 70 characters) to the server containing a message to be appended to the station status lines. The first character of such a line has to be an exclamation mark.

Example:

```
!This is a general text message
```

More details can be found at ftp://ftp.unibe.ch/aiub/slr/slr_stat.txt.

3. The `eurostat` Program

In order to simplify the real-time status exchange between the laser stations we have prepared a C program to be run under UNIX or on a Windows PC. It directly connects to port `7810` on our system, sends a line of text and receives the current merged status messages.

The program is called as follows:

```
eurostat aiuas3.unibe.ch 7810 'n' [<textfile] [>statusfile]
```

The program reads from the standard input the locally generated status line and writes the received status to the standard output.

The standard input (i.e., the local status line) can of course be piped into the program, the received merged status on the standard output can be piped into some display program if necessary.

'**n**' is the number of receive cycles to be run, a zero means an infinite number. Usually '**n**' should be set to 1 and the program should be called in a loop every 15 to 30 seconds.

The program can also be used to just receive the compiled status file indefinitely by omitting any standard input and by setting 'n' to zero.

The source code, a DOS executable as well as a more detailed description can be found at ftp://ftp.unibe.ch/aiub/slr.

4. Conclusions

The real-time status exchange as described above and implemented at the Zimmerwald server `aiuas3.unibe.ch` and as clients at a number of (mostly European) laser stations has proven to be an easy to use tool for the rapid exchange of real-time satellite time bias and status information. Operators at the tracking stations also started to like the possibility to "see" what the other stations are currently tracking, it may improve the "corporate feeling" among the ILRS station operators and may even generate a friendly competition for successful tracking.

We encourage all ILRS stations currently not taking part in the status exchange program to join the group. For questions and support please contact the author.

Acknowledgment

I would like to thank Martin Ploner of our institute for the generation of the `eurostat` C program.