

# Datastream

The datastream process loads baseband data from a source (file, mark5 module, or network socket) into a big ring buffer. The length of this buffer is an integer number of “subintegration” lengths (which is itself an integer number of FFTs). This ring buffer is divided into segments. There are 3 relevant variables controlling the size and slicing of this buffer:

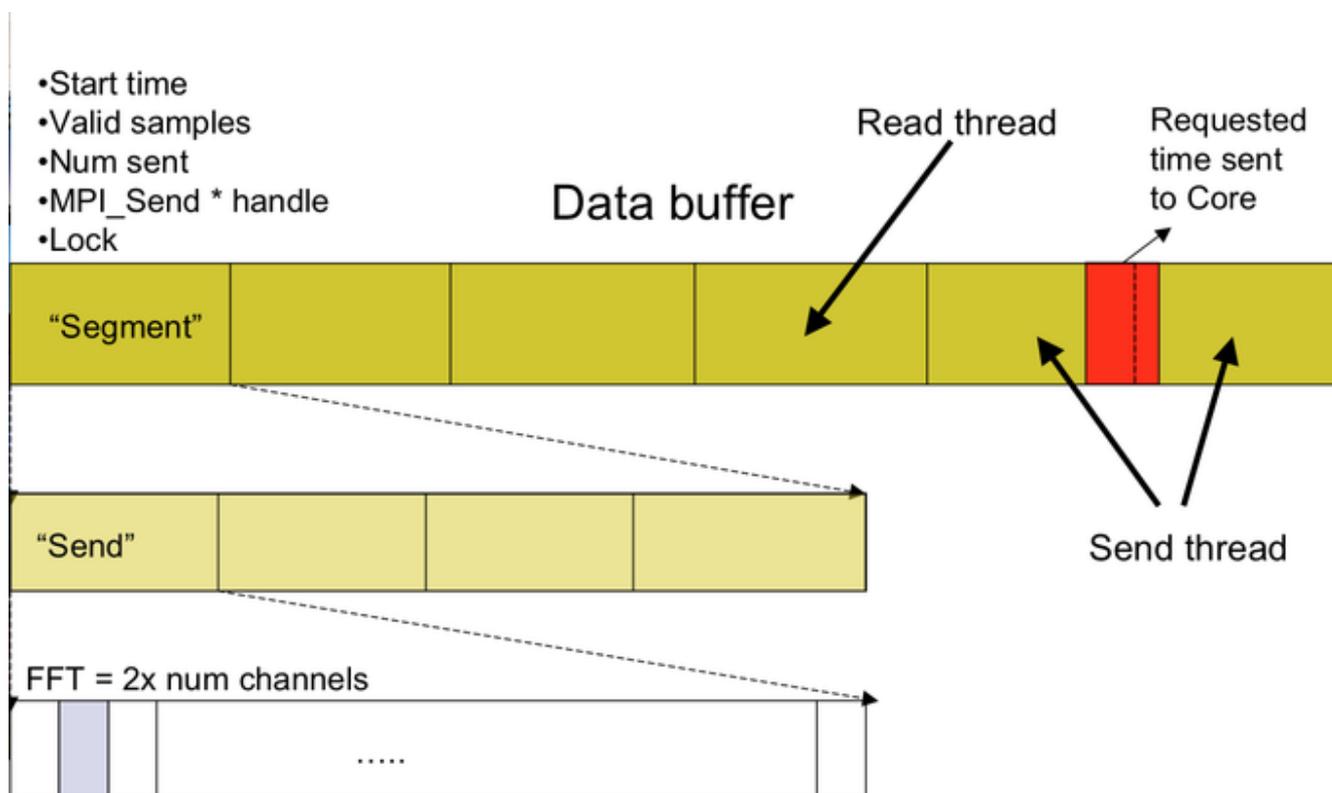
SUBINT\_NS: The length of a subintegration in nanoseconds (must be an integer number of FFTs)

DATA\_BUFFER\_FACTOR: The number of subintegrations in the data buffer

NUM\_DATA\_SEGMENTS: The number of segments which the data buffer is sliced into

If the correlation has more than one configuration, the one which would create the largest data buffer is used.

The ring buffer is shown in more detail as a picture below:



## Controlling the buffer size

The Datastream process has one buffer of consequence; the databuffer shown in the image above. There are a number of conflicting requirements which influence a good size for this buffer:

1. It must fit in memory (obviously)
2. It should be large enough to allow for some latency as the segments are filled
3. The time duration of an individual segment must be an integer number of nanoseconds
  1. Thus for funny sampling rates, the segment size must be carefully chosen
  2. And for low data rates the segment size cannot be too large, otherwise its duration in nanoseconds would not fit in a 32 bit int

4. The size of a segment should be such that the filling method (reading from a file, network stream or module) is efficient; this typically means not too small, at least ~10 MB
5. But some limitations with the Mark5 system can conflict with this; on certain linux kernels, a read size >8 MB from a Mark5 module will lead to corrupted data

If you start with an appropriate subintegration size, then choosing an appropriate databufferfactor to match the requirements above is usually not too hard. However, figuring out a subint size that is also well-matched to the requirements of the `core` process can make this procedure somewhat iterative and challenging.

From:

<https://www.atnf.csiro.au/vlbi/dokuwiki/> - **ATNF VLBI Wiki**

Permanent link:

<https://www.atnf.csiro.au/vlbi/dokuwiki/doku.php/difx/mpifxcorr/datastream>

Last update: **2016/08/26 13:34**

