Escola Politècnica Superior de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

TREBALL DE FI DE CARRERA

TÍTOL: Implementation of the Optimized TDC in the Gaia TM CODEC

AUTOR: Ricard Castro Navarro

DIRECTOR: Enrique García-Berro Montilla

DATA: 29 de Juny de 2005

Títol: Implementation of the Optimized TDC in the Gaia TM CODEC

Autor: Ricard Castro Navarro

Director: Enrique García-Berro Montilla

Data: 29 de Juny de 2005

Resum

Gaia és la nova missió de l'Agència Espacial Europea (ESA). El seu llançament està programat per al Desembre de 2011 i té com a objectiu la creació del mapa tridimensional més complert i precís de la nostra Galàxia. Des del punt L2, situat a un milió i mig de quilòmetres de la Terra i on les gravetats de la terra i el sol es troben compensades, Gaia observarà més de mil milions d'estels, deu mil nous sistemes planetaris, mig milió de quàsars, i un llarg etcètera.

Per tal de dur a terme aquesta tasca Gaia incorporarà tecnologia pionera. El satèl·lit contindrà 3 telescopis que podran mesurar les posicions, velocitats radials i colors dels estels. Amb aquesta informació serem capaços d'esbrinar la composició química i la formació de la nostra Galàxia, la Via Làctia.

La missió es troba actualment en una etapa de disseny conceptual. La meva aportació en la missió ha estat la d'elaborar un conjunt d'eines software que processen la informació que arriba de l'instrument astromètric (***Astro***). Aquest software es situa abans de l'etapa de compressió de dades i permetrà, entre d'altres coses, la creació de paquets de dades basats en "Time Slots" i l'adaptació de les mides d'aquests TS a la densitat d'estels. En l'elaboració d'aquest software s'ha implementat una nova manera d'establir les mides dels TS de tal manera que he optimitzat els recursos de què disposarà Gaia, arribant a estalviar un percentatge significatiu en la mida de les dades que caldrà transmetre a terra.

Title: Implementation of the Optimized TDC in the Gaia TM CODEC

Author: Ricard Castro Navarro

Director: Enrique García-Berro Montilla

Date: 29 de Juny de 2005

Overview

Gaia is the new mission of the European Space Agency (ESA). The launch is scheduled for December 2011 and the main objective is the generation of the most complete and accurate map of our Galaxy. Gaia will orbit around the L2 point, where the gravities of the Earth and the Sun are balanced. The amount of data that Gaia will manage gives us an idea of the importance of the mission: the observation of about one billion of stars, ten thousand planetary systems, half a million quasars…

In order to accomplish this task, Gaia will include the latest technology. The satellite will have 3 telescopes able to measure the positions, radial velocity and colour of the stars. With this information, we will be able to know the chemical composition and the formation of our Galaxy, the Milky Way.

The mission is now in a conceptual design stage. My contribution to the design of the mission has been the elaboration of a set of software tools that are able to process the information generated by the Astro instrument. This software operates before the compression stage, and it will make possible the creation of source packets based on “Time Slots” and the adaptation of their size to the stellar density. In the elaboration of this software it has been implemented a new way to establish the size of these TS, optimizing the resources of the satellite and saving a significant percentage in the amount of data that Gaia will have to transmit.

“La Tierra vista desde la Luna es el objeto más hermoso que existe. Es un planeta realmente impresionante, porque sus vivos colores destacan mucho sobre el cielo negro.”

- **Edwin E. Aldrin Jr**

Agraïments

Ja fa aproximadament 3 anys que vaig escriure uns agraïments semblants a aquests en el meu treball de recerca de batxillerat (El projecte Apol·lo). Des de llavors, l'única cosa que ha canviat ha sigut que en comptes d'agafar el metro per baixar-me a Verdaguer, he hagut d'agafar el tren cada dia per anar a Castelldefels. Bueno no... la veritat es que he passat de jugar al Flight Simulator (sense saber molt bé el que feia), a estudiar tots els components de la cabina d'un Airbus, a pujar-me en el Concorde, a saber com funciona el sistema elèctric del Aeroport del Prat, a construir una base lunar, a saber per que no s'ha d'encendre el mòbil dins de l'avió, a identificar quins núvols poden produir pluges i quins no... Bueno la veritat es que han canviat moltes coses. Fins i tot amb aquest treball he complert un dels meus somnis; participar en una missió espacial.

Per aquestes i altres coses. Primer de tot voldria agrair a Enrique García-Berro l'oportunitat que m'ha ofert en aquest treball. Amb el qual m'he pogut adonar de la complexitat que comporta la realització d'una missió espacial, així com l'acolliment durant aquests 4 mesos de treball a l'IEEC.

Després agrair a Jordi Portell que, amb tanta atenció i paciència, m'ha guiat en el complicat mon de la programació i de la telemetria. Així com en la correcció i revisió del TFC (s'ha hagut de barallar amb el meu anglès). També li estic agraït per haver-me animat a utilitzar el sistema operatiu Linux, el qual actualment utilitzo.

Tant a Jordi, com a Enrique i com a José Pérez (Company d'aeronàutica) li dono les gràcies pels bons moments passats en les nostres Jamborees Tecnològiques, que hem dut a terme durant cada divendres en els últims quatre mesos per realitzar aquest document. M'ho he passat molt molt molt bé.

També vull agrair als meus pares tot els esforços que han fet durant tota la meva vida (que no han sigut pocs), tot el que soc com a persona us ho dec a vosaltres. Gràcies per confiar en mi, per voler-me oferir un futur de profit, per donar-me tot el que necessitava i espero que algun dia els ho pugui agrair com cal. També gràcies al meu germà, que amb tanta paciència ha estat baixant el volum de la midi-cadena sempre que li he demanat ;-). Però estudia més! Us estimo molt a tots.

També a la meva avia i a la meva tieta, gràcies per acollir-me com un fill.

Bueno i com no, i més ara que conec una mica més el espai. Segueixo pensant que no necessites cap nau per pujar a les estrelles. Però ara ja no ets la meva amiga ehh!!! ☺, jejejeje. Gràcies per la paciència que has tingut durant aquest tres anys, esperant que arribessin els caps de setmana per poder-nos veure.

Gràcies també als meus companys d'universitat, en especial a: Luís, Johny, Marc Torres, Chelu i Juanjo amb quin tants bons i mals moments he passat.

I, finalment, vull dedicar aquest treball a totes aquelles persones que han treballat directa o indirectament en missions espacials. A tots vosaltres, que heu sigut la meva font d’inspiració durant aquests anys de carrera.

A tots aquells que ara esteu a la terra i els que ara no ho esteu. Moltes Gràcies.

TABLE OF CONTENTS

List of tables and figures

INTRODUCTION

Gaia is the new mission of the European Space Agency (ESA), the objective of which is the production of the most ambitious three-dimensional map of our Galaxy, the Milky Way. This map will contain the most complete census of celestial bodies ever obtained, including more than one billion stars (which represents 1% of the Galactic stellar population), thousands of extra-solar planets, galaxies and some 500000 distant quasars.

Gaia is the successor of Hipparcos, the first astrometric mission which was also designed and operated by ESA in the late 1980s. Hipparcos was already a revolutionary mission, although only some hundred thousand stars were catalogued and with a worse precision. Gaia is being designed taking advantage of some of the Hipparcos premises, such as the nominal scanning law (NSL).

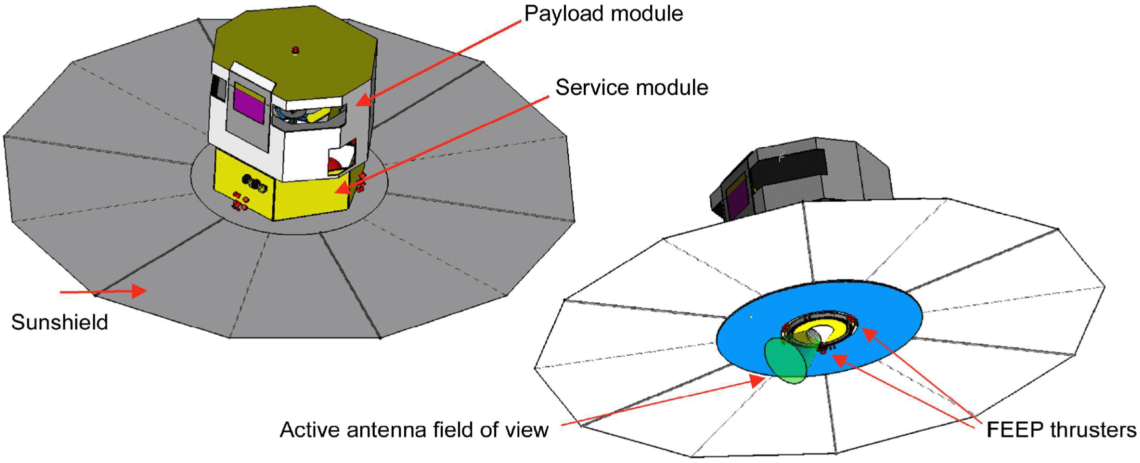
Gaia will be launched in 2011 by a Soyuz-Fregat vehicle. The total launch mass will be about 1700 kilograms, including 800kg of payload, 600kg of service module and 270kg of fuel. The spacecraft will be inserted in an orbit close to L2, the second Lagrangian point, located at about 1.5 million kilometers from the Earth in the direction opposite to the Sun. This will make possible a continuous eclipse-free scanning of the sky during five years, which is the nominal lifetime of Gaia; the objective is to determine the position, colour, composition and proper motion of the astronomical sources being observed. About three years will be needed in order to process on ground all the information gathered from Gaia. These data will provide us the necessary acknowledge to determinate the star formation and evolution of the Milky Way. The mission will conclude nominally on 2020. The entire mission, including launch, ground operations and payload, will cost about 450 million Euros.

An exciting aspect of this mission is that, owing to its extremely high sensitivity and systematic scanning, Gaia will be able to detect and measure a large amount of stars and objects currently unknown. For example, small objects such as asteroids will be measured and catalogued. These asteroids sometimes move very close to the Earth and, hence, may suppose a danger. A quick-look reduction process, combined with an alerting system, will make possible the discovery of such objects with high potential of danger, as well as the discovery of supernovae or other interesting events.

In order to achieve all these objectives Gaia will include state-of-the-art technology. The spacecraft contains three optical telescopes, two of which combine their images onto an astrometric focal plane in order to obtain the high-precision position measurements. The third telescope is used for spectrometric and photometric measurements. The full spacecraft can be divided in two main sections: the payload module and the service module.

The service module contains the propulsion system and the communication units, which are essential components that allow the spacecraft to operate correctly and return the data to the ground –taking into account that it will be located at 1.5 million kilometers away. Gaia is expected to communicate with the Earth during eight hours per day. During this time, it will transmit the science data and ***housekeeping*** telemetry. Although the power of the transmitter is about 17 Watts, the relatively weak signal sent by the phased-array antenna of Gaia will be easily received by the Cebreros antenna currently being built at ESAC (Spain), which is the main ground station baselined for this mission.

The payload module is fixed to a high-stability SiC hexagonal structure, housed inside a dome called the thermal cover. Here is where the three telescopes are located; the two astrometric telescopes are called Astro-1 and Astro-2 and their FOV are separated about 106º each other. We will focus all of our work on the measurements made with them. Both of these telescopes use a 1.4m primary mirror, while the latter mirrors will be common to both telescopes in order to combine the images onto an array of electronic detectors; more specifically, about 180 CCDs (Charge Coupled Devices). They will measure the position and brightness of the celestial objects being observed by Gaia. This will be the largest CCD focal plane ever built, leading to an equivalent resolution of about 1.5 Gigapixels –although the image acquisition will not be the “typical” shutter-based one. This high-end system will make possible the accurate measurement of star positions with an accuracy of a few microarcseconds, which is equivalent to the diameter of a human hair as seen from 2 thousand kilometers.

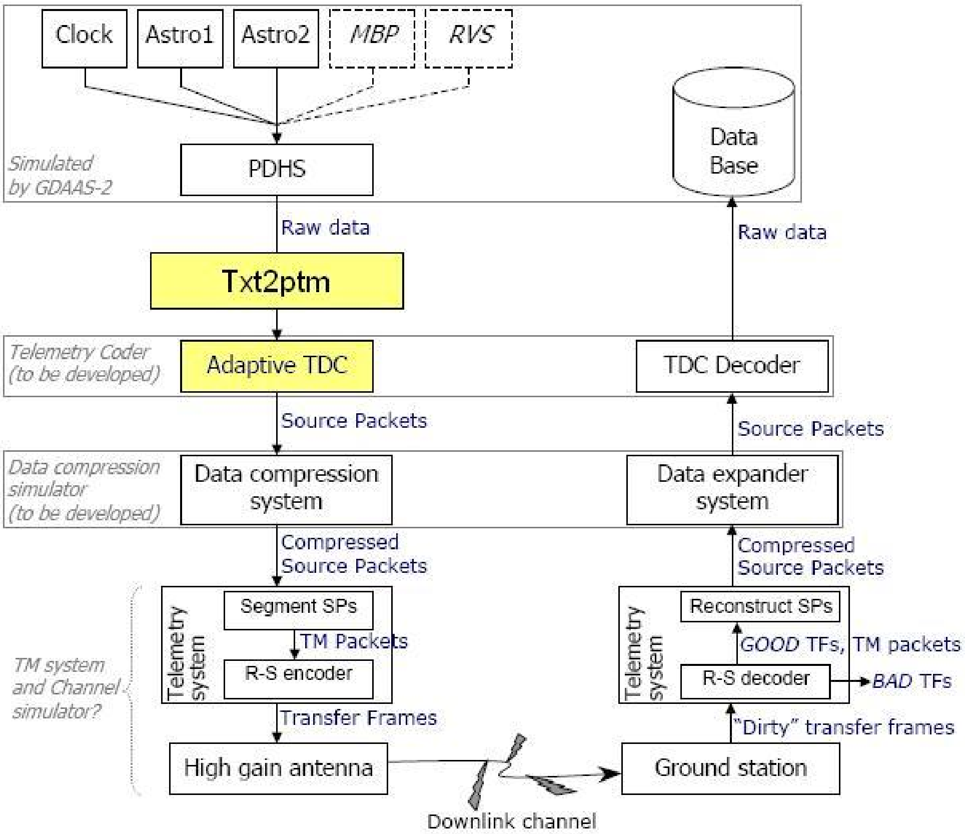


The third telescope, called Spectro, will measure the radial velocities of the sources owing to the shift in their spectral lines. This information can then be combined with that one from the Astro instruments to give a full picture of how the celestial object is moving through space. The Spectro instrument will also be able to do medium-resolution photometry of the celestial objects, which will be used in order to determine the physical parameters of the sources.

In order to reach such high levels of accuracy, it is important to keep a uniform temperature in the payload –and especially in the most critical elements, such as the CCD focal plane and the telescopes. A large sunshield (of 11-meter diameter) will be included as seen in the figure of above, which will also include part of the solar array. The payload will be about 3 meters wide per 2 meters high, while the service module will be about 1 meter high.

The mission is currently in the conceptual design stage, where detailed scientific requirements are being fixed. During next year the detailed design and implementation phases will start, after deciding the prime contractor of the mission –which will be EADS/Astrium or Alcatel/Alenia. This is a crucial phase since the scientific capability of Gaia will be fixed. For this, detailed designs and simulations will be needed from the scientific teams.

The contribution of this work to the mission consists in the preparation of the science data generated by the Astro instrument, in order to be correctly fed into the telemetry and communication systems. This will make possible the transmission of all the science data to the ground. This is an important task because this software must fulfil the requirements and standards fixed by ESA to space missions. The main product offered by our work will be a reliable simulation software named ***txt2ptm***, able to convert simulated science data to a flight-realistic sequence of data packets. The location of this software in the data pipeline of Gaia is shown in the following figure, in the yellow boxes.



The main requirements fixed for the development of ***txt2ptm*** include:

- The creation of a Time Slot-based codification scheme
- An Adaptive Codification method

This software tool has been designed and implemented as additional software modules for the Gaia TM CODEC that will offer a flight-realistic Packet Telemetry (PTM) format. It includes not only the GASS-to-PTM[1] converter, but also a source sorting module that solves some limitations of the simulated data. The system implements the optimized Time Data Codification (TDC), using the Time Slot concept and an adaptive scheme depending on the density of the sky region being observed. In this way, an optimal codification is always achieved, thus avoiding and unnecessary occupation of the downlink channel.

The purpose of this document is to explain the solutions that have been chosen in order to implement a new TDC software, able to receive the files generated by GASS, modify them and insert the adequate new fields needed to obtain a flight-realistic PTM format. We will not only follow the guidelines previously fixed for the creation of time slots, but we will also explore new improvements to the current design. The code to be implemented will be described as visually as possible, avoiding any inclusion of large code listings in benefit of block diagrams and examples. It will make possible a better understanding of the software developed here. We recall that only the Astro instrument will be considered in this study.

[1] GASS is the Gaia Simulator, which produces simulated sources according to a Galaxy model.

CHAPTER 1. WHERE AND WHY?

As already explained in [RD.5.], the currently available TM data for Gaia (generated by GASS) does not fulfil any flight-realistic standard. In [RD.4.] an optimized Time Data Codification (TDC) was introduced, which, moreover, fulfills the CCSDS/ESA Packet Telemetry standard, as well as some CCSDS Time Code Formats. Nevertheless, the current version of the Gaia TM CODEC (see [AD.2.] and [AD.3.]) does not include any of these standards. In other words, there is a TM simulator available (GASS and TM CODEC) and a flight-realistic design available (the optimized TDC), but not an implemented software combining both. It is therefore imperative to develop a software able to simulate and convert such data into flight-realistic TM data. The best solution is obviously the inclusion of the designs already available in [RD.4.] in the already available software [AD.2.].

This new software, which we call ***txt2ptm*** (text to packet telemetry), is conceived as a tool that can receive data from GASS and deliver it to the rest of TM CODEC software using the appropriate TDC structure as shown in Fig. 1.1 a hybrid software implementation will be used, this is, with the data field formats configured in an XML file but using a software-fixed transmission scheme. We must note that the currently available modules of the TM CODEC shall be modified in order to accept this new TM format, although these modifications will be minimal.

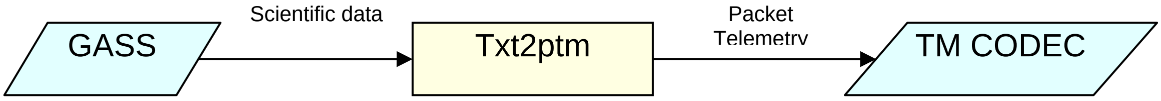


Fig. 1.1 From GASS to a flight-realistic TM CODEC

1.1 Preliminary step: preparing the data for txt2ptm

As explained in [RD.4.], the timing used in the science telemetry has an important impact in the overall telemetry size. The simulated telemetry data generated by GASS [RD.5.] follow an elementary scheme based on data sets, coding the time data of every source wrt the last reference time. There is a better optimized way to do this, which is to divide each data set in several time slots, labelling every time slot with a ***time slot mark*** (TSM). Then, every source will have its time data coded wrt the last TSM, thus needing fewer bits. We will describe this method hereafter. The use of one or another TDC method must be fixed before starting developing the PTM converter software, so that both processes could be done in a single step –this is, converting to PTM and using a new TDC scheme.

When starting the implementation of this optimized TDC we found a shortcoming in the files generated by GASS: the detected transit times (DTT) of the sources in any arbitrary data set did not increase uniformly, i.e., the sources were not sorted by their DTT, as they should be in a flight realistic environment. Instead they realeased according to their Hierarchical Triangular Mesh (HTM) coordinates. The reason may be the internal implementation of GASS, and is beyond the scope of this study. In the real mission, the PDHS of Gaia will release the DTTs of every source in a ***first detected – first output*** basis, i.e., sorted by their DTT within every data set (which significantly eases the calculation and generation of time slots). For this reason, a sort algorithm must be implemented prior to the ***txt2ptm*** block, so that all of the sources in a data set will be sorted by their DTT. This previous add-on software will be named ***txt2sort***.

CHAPTER 2. TXT2SORT

As noted in chapter 1 before, we need to sort all the sources in every data set in order to correctly create their respective time slots. A software tool has been developed, which is able to get a telemetry file from GASS simulated data and generate a ***.txt** file with the sources of every data set sorted according to their respective DTTs.

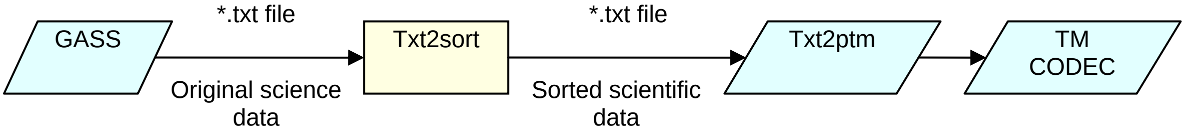


Fig. 2.1 Location of txt2sort in the TM simulation pipeline

Although the PDHS of Gaia will, in principle, release the source data with a correct DTT order, we have devised the ***txt2sort*** tool to operate as flight-realistic software (just in case it may be needed for the real mission). This is, it should be able to process the data in real time. We must have in mind the structure of a GASS TM file in order to understand the explanation below, so we have included a sample in Annex B as an example.

It is not easy to process the real time data coming from the instruments if the sources are not sorted by their DTT. One of the libraries developed in [AD.2.] (the I/O library) makes use of a memory buffer to save all the field values of each data set. We can imagine this buffer as a vector, with well-known positions for every data field in the GASS file. A variable named ***bufpos***, continuously incremented with the field sizes in bytes, makes possible this accurate knowledge of the data locations in the vector. Saving the ***bufpos*** value at a given instant, while storing GASS data to memory, will make possible determining the first buffer position of the data from a given source. In our case, we must save the position of the source identifier. The other important value to take into account is the Detected Transit Time or DTT (cf. Annex B, Ref. A). Hence, if the set of DTT values of a data set is sorted and their associated sources (stored in the memory buffer) are written using the corresponding ***bufpos*** values, all the sources in the GASS file will be sorted.

Figure 2.2 graphically displays how the program runs for every data set. First of all, it creates a vector with a simple data structure in each of its positions: ***bufpos*** and DTT. The vector has a size equal to the number of sources in the data set. Afterwards, the vector is filled with the adequate positions of each Source Identifier and DTT. Finally, the vector is sorted with respect to DTT but, obviously, keeping the ***bufpos*** values with their respective DTT. After this process, the program follows the “script” given by the DTT+***bufpos*** pair and writes every source stored in the memory buffer to the output file.

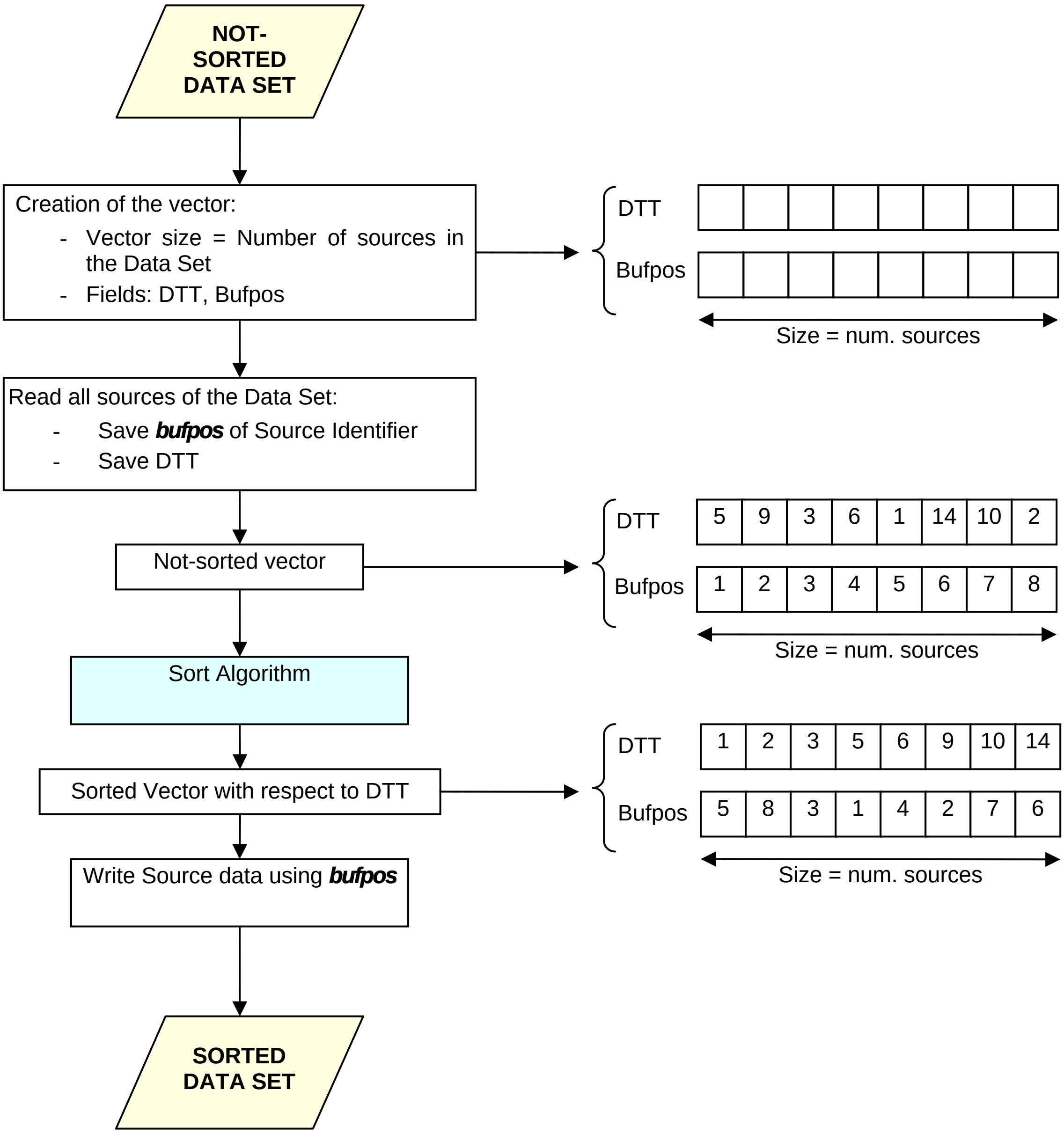


Fig. 2.2 Data Set sort process

2.1 Sort algorithms

In this section we briefly describe and test some of the most popular sort algorithms available so far, determining their reliability when applied to our particular problem. These third-party algorithms have been adapted for being executed in ***txt2sort***, in order to keep the ***bufpos*** values with their corresponding DTT values.

2.1.1 Bubble

The ***bubble*** algorithm sorts a vector moving the high values to the right. The algorithm loops ***n***-1 times, being ***n*** the number of sources. An example is shown below, where we have highlighted the highest values that have been moved every step:

Iteration 1:	6	9	5	8	4
Iteration 2:	6	5	8	4	**9**
Iteration 3:	5	6	4	**8**	9
Iteration 4:	5	4	**6**	8	9
Iteration 5:	4	**5**	6	8	9

2.1.2 Direct Insertion

The ***Direct Insertion*** algorithm consists in moving every value to its correct position with respect to the elements below it. The algorithm needs ***n***-1 steps as well.

Iteration 1:	6	9	8	5	4
Iteration 2:	6	9	8	5	4
Iteration 3:	6	**8**	**9**	5	4
Iteration 4:	**5**	**6**	8	9	4
Iteration 5:	**4**	5	6	8	9

2.1.3 Direct Selection

This algorithm detects the element that has the lowest value and sorts it with respect to the fields below it. This algorithm again has ***n***-1 stages.

Iteration 1:	6	9	4	7	3
Iteration 2:	**3**	6	9	4	7
Iteration 3:	3	**4**	6	9	7
Iteration 4:	3	4	**6**	9	7
Iteration 5:	3	4	6	**7**	9

2.1.4 Quicksort

The theory of the implementation of a ***quicksort*** algorithm is complex and is beyond the scope of this document. Anyway, in a few words, ***quicksort*** is based on the “divide and win” method: it consists of several loops, selecting a random value in each of them; the values next to this one are then sorted accordingly. [RD.6.] contains a thorough description of this algorithm. We have adapted an already available quicksort implementation to the needs of ***txt2sort***.

2.2 Statistics

Since Gaia will eventually measure a large amount of sources every second, the execution speed of our sorting algorithm was the most important statistic to obtain. The standard C++ library `<time.h>` has been used in order to retrieve the time that an algorithm needs for sorting a given number of sources. The star density ranges from 267 to 32000 sources per second, which is a realistic range of the number of sources that can be contained in a Data Set [AD.3.]. The following figure shows that the Quicksort algorithm has been the fastest one.

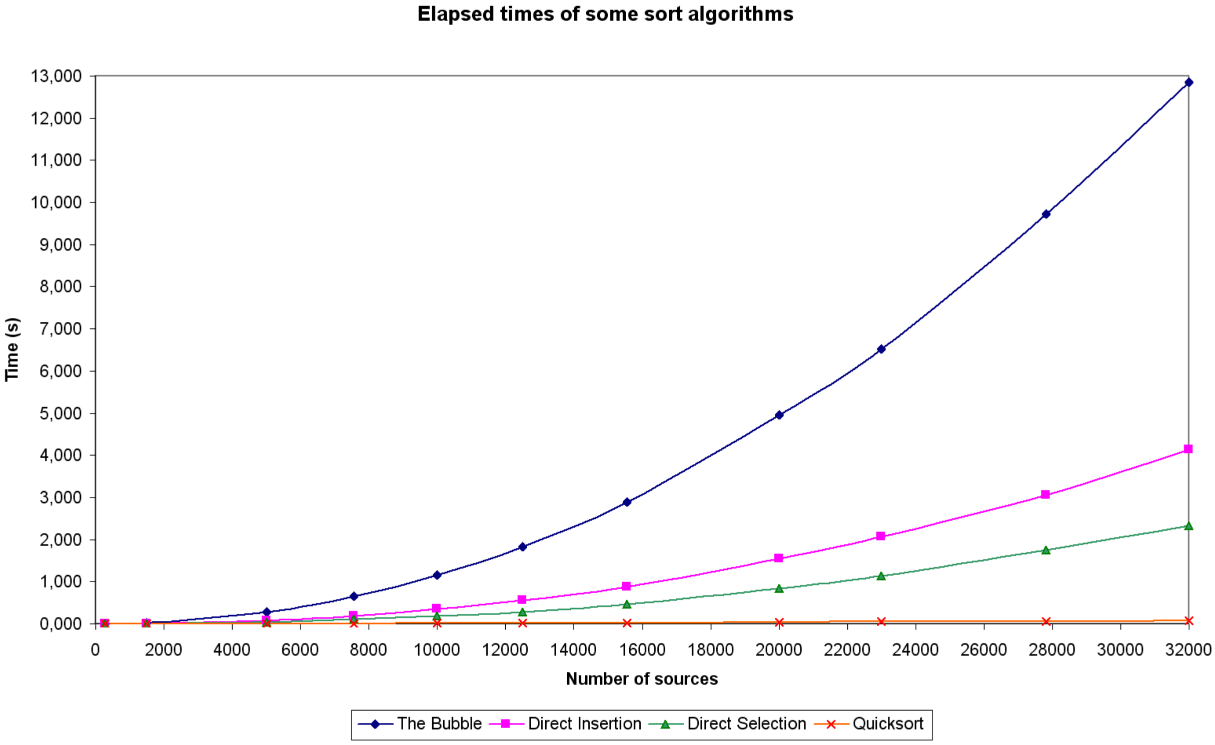


Fig. 2.3 Elapsed times of the sort algorithms tested

The values used for generating this plots are included in Annex C. We include in figure 5 a separate plot of the Quicksort elapsed times, which offers a spectacularly faster execution: it is able to sort 32000 sources in only 70 milliseconds. Therefore, it will be the sort algorithm used in ***txt2sort***.

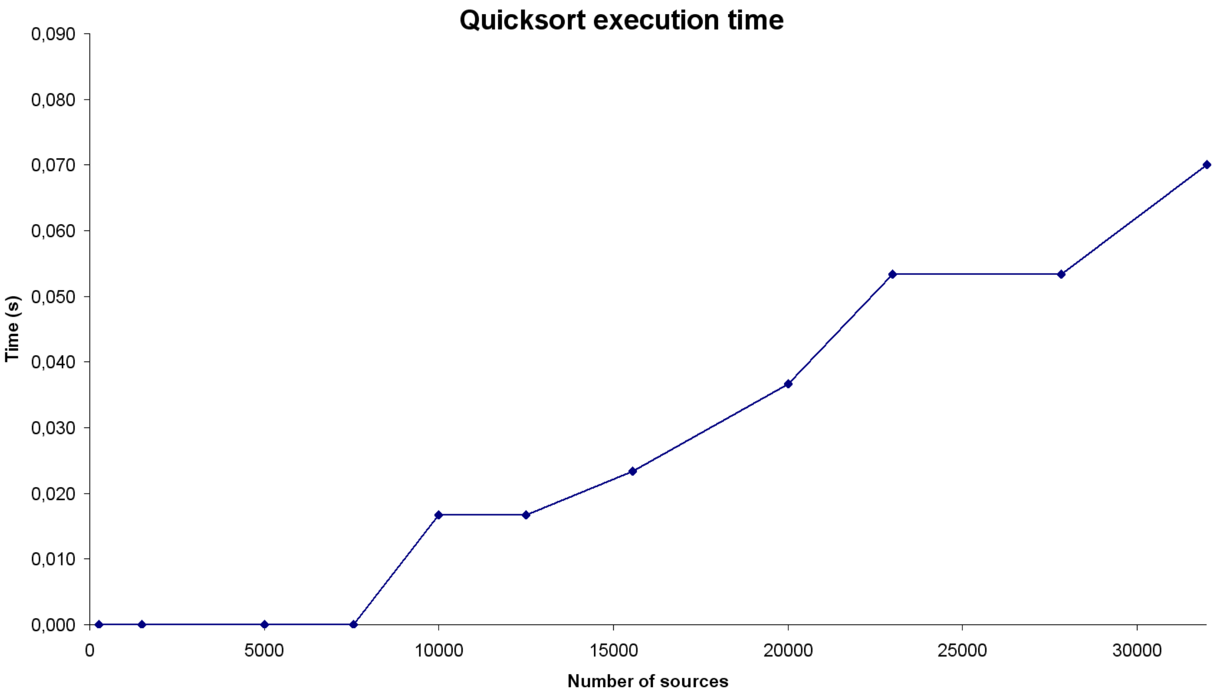


Fig. 2.4 Elapsed time of the Quicksort algorithm

For the sake of completeness, in Table 2.1 we also show a comparison between the different sort methods when processing a large amount of sources –which is a reasonable maximum for Gaia. We can see that Quicksort runs about 33 times faster than the best of the rest of algorithms, and that the Bubble algorithm is the slowest one.

Table 2.1 Speed of several sorting algorithms applied to a large vector of data

Sort Method	**Number of sources**	**Time (seconds)**
Quicksort	32000	0.070
Direct Selection	32000	2.327
Direct Insertion	32000	4.140
Bubble	32000	12.843

Consequently, we have chosen to implement the Quicksort algorithm for the calculations reported in this technical note.

CHAPTER 3. TXT2PTM

3.1 A new timing scheme

The main objective of the ***txt2ptm*** software is to change the current base of time to an optimized Time Slots (TS) philosophy. First of all, let us review the currently used scheme. The files generated by GASS organize the sources in the way shown in figure 3.1.

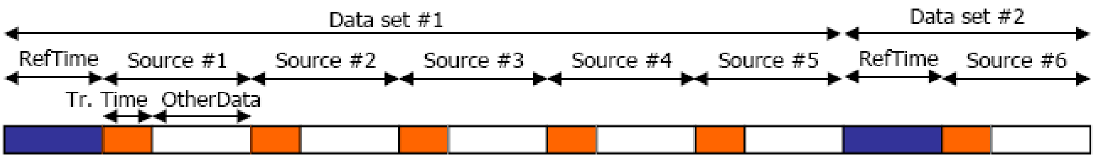


Fig. 3.1 Current codification of time data

In this figure it can be seen that the sources are packed in blocks named Data Sets (DS). Every data set starts with a reference time, and every source inside that data set contains time data relative to this reference time. Using this structure already saves an important amount of telemetry size, since it avoids including a large time label in each of the sources.

On the other hand, ***txt2ptm*** aims to implement the structure defined in [RD.4.], which is illustrated in Fig. 3.2.

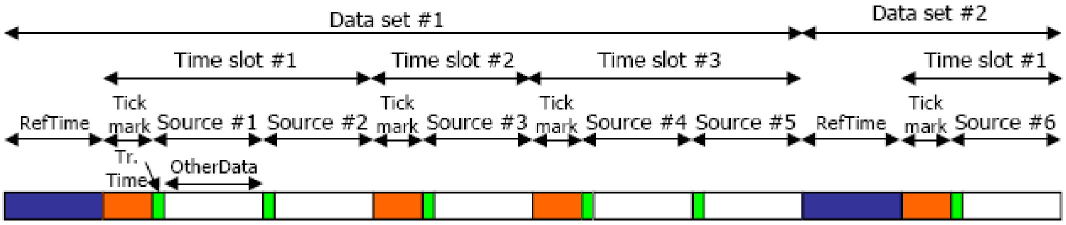


Fig. 3.2 New codification of time data, using time slots

In this case, the sources within a Data Set are structured in smaller blocks named Time Slots (TS), each one starting with a ***tick mark*** (a time label) that is used as the new time reference for the sources. At the same time, these time labels (named Time Slot Marks, TSMs) are coded wrt the last reference time. In this way, fewer bits are used to label the measurement time of every source. There are obviously other ways to code the timing data and to do it in an optimized way, but ***txt2ptm*** will use this scheme.

3.2 The creation of Time Slots

The first step needed to implement an optimized TDC is to change the base of time while at the same time adapting it to the current star density. This will be explained in the following subsections.

However, it is worth mentioning at this point that as already noted in [RD.4.], a Time Slot is "closed" (and a new one is created) when:

- The total amount of sources within that time slot is 15
- A given number of TDI periods since the last TSM is reached (this is, the Maximum TSM Offset or MTO)
- The current Data Set is finished (because a new reference time has been generated)

3.2.1 Adaptive TS length

The adaptive system presented in [RD.4.] is able to adapt the size of the TS to the star rate. The star rate is a value that indicates the number of sources measured every second, which equals to the number of sources contained in one data set (of 1-second length). In the case of GASS, it is clearly noted in a given data field of the TM file (see Ref. B in Annex B). Table 3.1 shows which interval of sources corresponds to every TS length (TS length equals to the Maximum TSM offset, or the time difference between the first and the last source of the same TS):

Table 3.1. Adaptive TS values (extracted from [RD.4.])

Number of Stars/second	Maximum TSM offset	MTO Flag value
<void data set>	<void data set>	"111"
< 61	524 TDIs	"110"
61 – 20	262 TDIs	"101"
120 – 256	131 TDIs	"100"
256 – 526	65 TDIs	"011"
526 – 1066	32 TDIs	"010"
1066 – 2145	16 TDIs	"001"
> 2145	8 TDIs	"000"

3.2.2 The absolute method

Our first objective is the creation of time slots as described in [RD.4.]. In this scheme, the DTT of the first TS source is used as the Time Slot Mark (TSM), relative to the reference time of the Data Set. On the other hand, the DTTs of the rest of sources in the TS are referenced to this first DTT, or TSM. We can better understand this with the help of the diagram shown in figure 3.3.

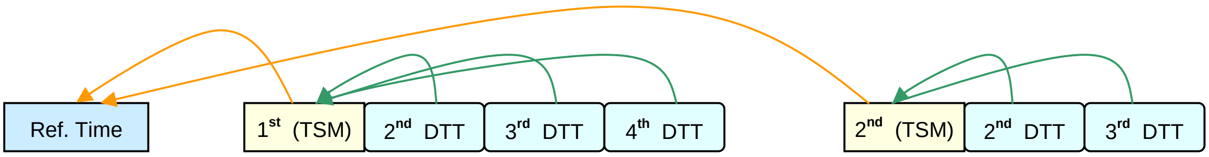


Fig. 3.3 Representation of the original TimeSlot-based method

A new TS is created when the value of a DTT (when coded as relative to the last TSM) reaches the maximum number of TDI periods allowed by the adaptive TS system. The following expression is used in order to obtain the new value of a DTT when converted to this TS-based system:

$$\boxed{\#TDI_n = \left(DTT_n - DTT_{1st}\right)} \qquad \textbf{(3.1)}$$

Where:

- $\#TDI_n$ is the new DTT value, as the number of TDI periods since the last TSM.
- DTT_n is the original DTT value of the current source.
- DTT_{1st} is the original DTT value of the 1st source of the time slot, which equals to the TSM.

Figure 3.4 illustrates this operation with an example, where the values shown in blue are the ones taken into account for determining when the time slot range has been exceeded.

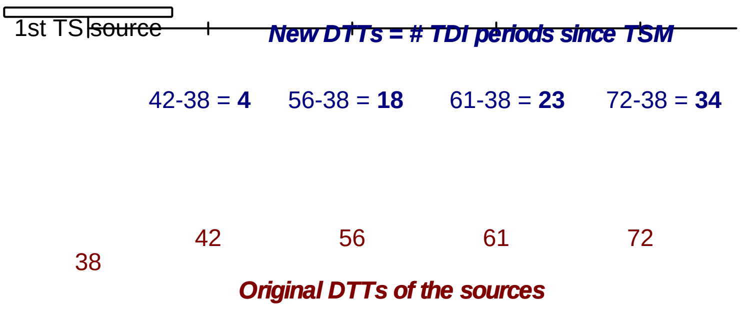

1st TS source
New DTTs = # TDI periods since TSM
42-38 = 4
56-38 = 18
61-38 = 23
72-38 = 34
38
42
56
61
72
Original DTTs of the sources

Fig. 3.4 Example of a TS-based absolute codification

3.2.3 The differential method

While implementing the TS-based method it was realized that there exists a possibility for going one step further in the optimization of time data, using a differential method for coding the DTTs of the sources. This alternative implementation was called the ***TS-based Differential Method***.

The differential method applies a differential coding to ***each*** of the DTTs of the sources. This is, the new DTT values are now relative to the last DTT value, rather than relative to the last TSM. Figure 3.5 shows this new operation.

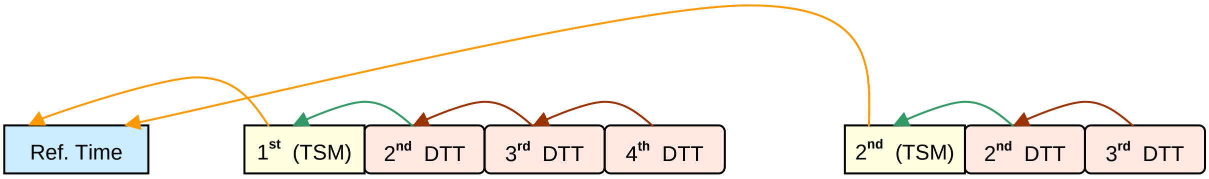


Fig. 3.5 The TS-based differential method

With this method it must be used a different expression for updating the counter of TDI periods (this is, the new DTT values). The new rule is the following:

$$\boxed{\#TDI_n = (DTT_n - DTT_{n-1})} \tag{3.2}$$

where a new variable has been introduced wrt the previous case:

- ***DTT**$_{n\text{-}1}$*: It is the value of DTT of the previous source

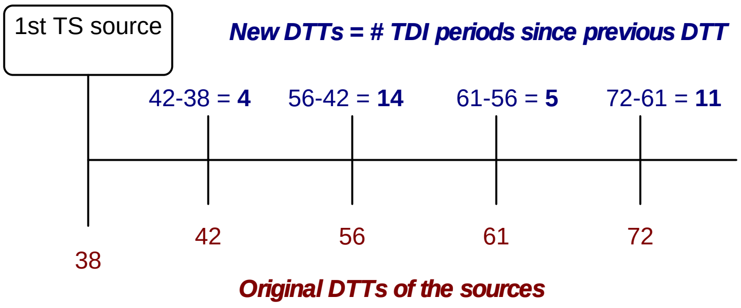

1st TS source
New DTTs = # TDI periods since previous DTT
42-38 = 4
56-42 = 14
61-56 = 5
72-61 = 11
38
42
56
61
72
Original DTTs of the sources

Fig 3.6 Example of a TS-based differential codification

The main advantage of this differential method is that the resulting DTT values are always smaller than (or, at most, equal to) the values obtained with the original (absolute) method. For instance, by visual inspection of Figs. 3.4 and 3.6, it can be easily seen that:

- DTTs in Figure 3.4: 4, 18, 23, 34
- DTTs in Figure 3.6: 4, 14, 5, 11

The preliminary sorting process of the sources wrt their DTT makes it possible. The main implication of this result is that, since these values are compared to the Maximum TSM Offset (MTO) in order to determine the end of a Time Slot, lower values will usually imply more sources in a Time Slot (i.e., in a Source Packet, see [RD.4.]). In other words, fewer time slots will be generated for the same amount of sources which, in turn, implies fewer headers (of the source packets) and, therefore, fewer bytes to be transmitted, resulting thus in an overall additional saving of telemetry occupation. A second implication of this result, which will only be seen in a later stage of the TM CODEC, is that lower values will be fed into the compression stage and, hence, we foresee that the final compression ratio achieved will slightly increase.
As it will be seen later, the DTT time information is transmitted at the beginning of a time slot (i.e., of a source packet). First, the Time Slot Mark (TSM) corresponding to the DTT of the first source is transmitted. Afterwards, a series with the several DTT values will be transmitted. The CCSDS-compatible codes presented in [RD.4.] will not be implemented yet at this stage, since we prefer focusing on the creation of time slots. Figures 3.7 and 3.8 show an example of the output file segment containing these values.

```
(...)
38             /*TSM*/
4 18 23 34     /*DTT*/
(...)
```

Fig. 3.7 Absolute DTT transmission in AP #B / #C

```
(...)
38             /*TSM*/
4 14 5 11      /*DTT*/
(...)
```

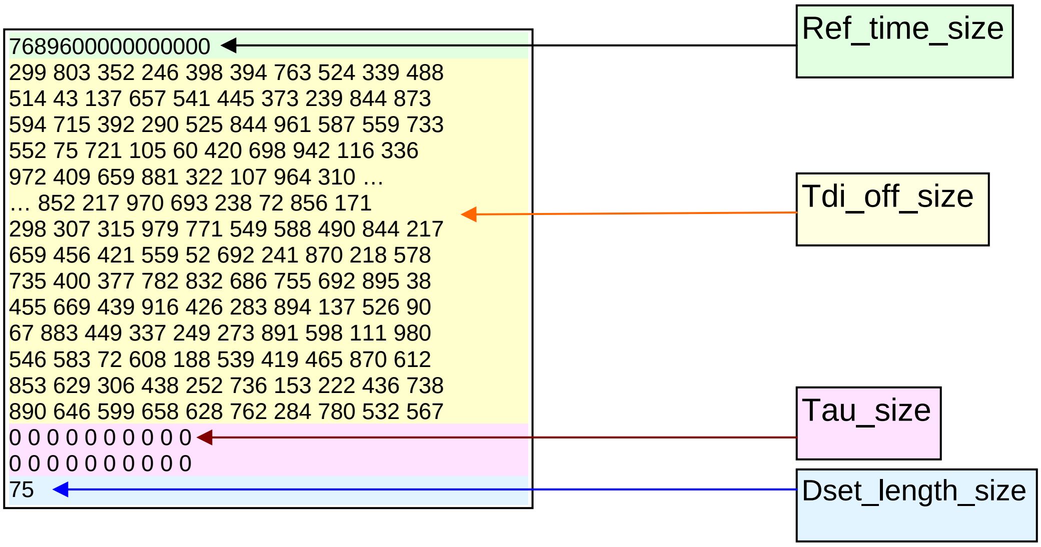
Fig. 3.8 Differential DTT transmission in AP #B / #C

3.3 Optimized packet telemetry structure

In this section we will study the work done for the implementation of the final packet telemetry (PTM) structure for AP #A and for AP # B and # C as well. [RD.4.] will be used as the reference for the packet structure. We recall that the final implementation of these data structures and their headers must fulfill the requirements of both the mission and the ESA telemetry standards.

3.3.1 PTM for AP #A

The output files of ***txt2sort*** are very similar to a GASS TM file, except for the quantization of some data fields (particularly the flux data). Figure 3.9 shows an example of a Data Set header, which will have to be converted to the AP #A data format.



```
7689600000000000
299 803 352 246 398 394 763 524 339 488
514 43 137 657 541 445 373 239 844 873
594 715 392 290 525 844 961 587 559 733
552 75 721 105 60 420 698 942 116 336
972 409 659 881 322 107 964 310 …
… 852 217 970 693 238 72 856 171
298 307 315 979 771 549 588 490 844 217
659 456 421 559 52 692 241 870 218 578
735 400 377 782 832 686 755 692 895 38
455 669 439 916 426 283 894 137 526 90
67 883 449 337 249 273 891 598 111 980
546 583 72 608 188 539 419 465 870 612
853 629 306 438 252 736 153 222 436 738
890 646 599 658 628 762 284 780 532 567
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
75
```

Fig. 3.9 Typical contents of a Data Set header

Txt2ptm will have to rearrange these data in order to follow the structure for AP #A proposed in [RD.4.], shown in figure 3.10. This structure is formed by a series of data fields like counters, timing flags and telemetry control data.

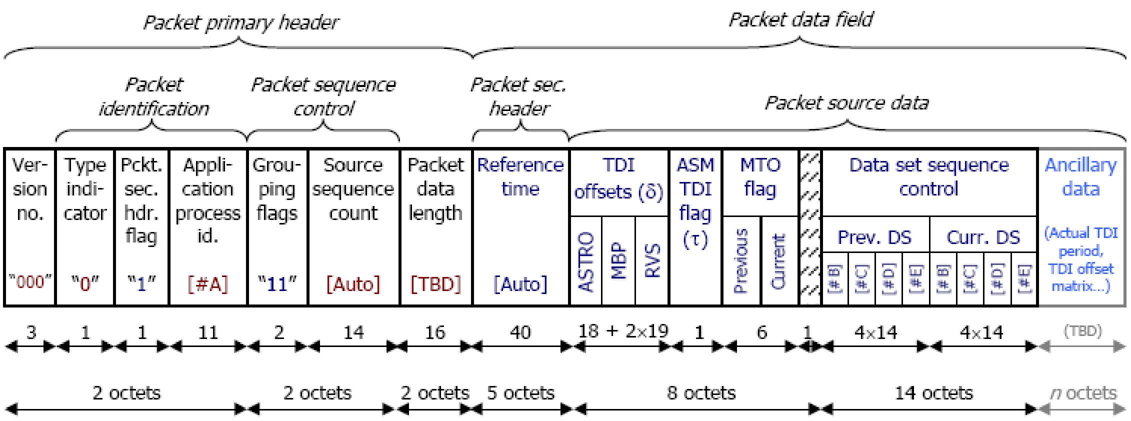


Fig 3.10 Final packet structure for AP #A data

Txt2ptm uses a structure named ***primary_h*** which contains every field of this packet structure. The following are its main fields:

- **Version no**., **Type indicator** and **Pckt. sec. hdr. flag** are standard-fixed values.
- **Application process id.** is a mission-fixed value. Since the final API values are under discussion and have not been fixed yet, we will use an arbitrary number such as 101. With this number we will easily identify a header of a Data Set in a ***txt2ptm*** output file.
- **Source sequence count**: it is a counter of the number of AP #A packets generated.
- **Packet data length** indicates the size of the packet data field (this is, excluding the packet primary header) in bytes. In the case of AP #A it is a fixed value (248 Bytes):

$$40+18\boldsymbol{x}10\boldsymbol{x}10+20\boldsymbol{x}1+6+14\boldsymbol{x}8=1978(\boldsymbol{bits})$$

$$1978/8=247{,}25(\boldsymbol{Bytes})\xrightarrow{\boldsymbol{round}}248(\boldsymbol{Bytes})$$

- **MTO flags**: The MTO Flag value used in the previous and in the current Data Set. It is determined by the adaptive timing scheme.
- Some **empty bytes** must be included in order to generate a source packet with an integer number of bytes (which is forced by the standard). In other words, it corresponds to the decimal value to be added to the packet data length in order to round to the next byte. In this case it is a fixed value:

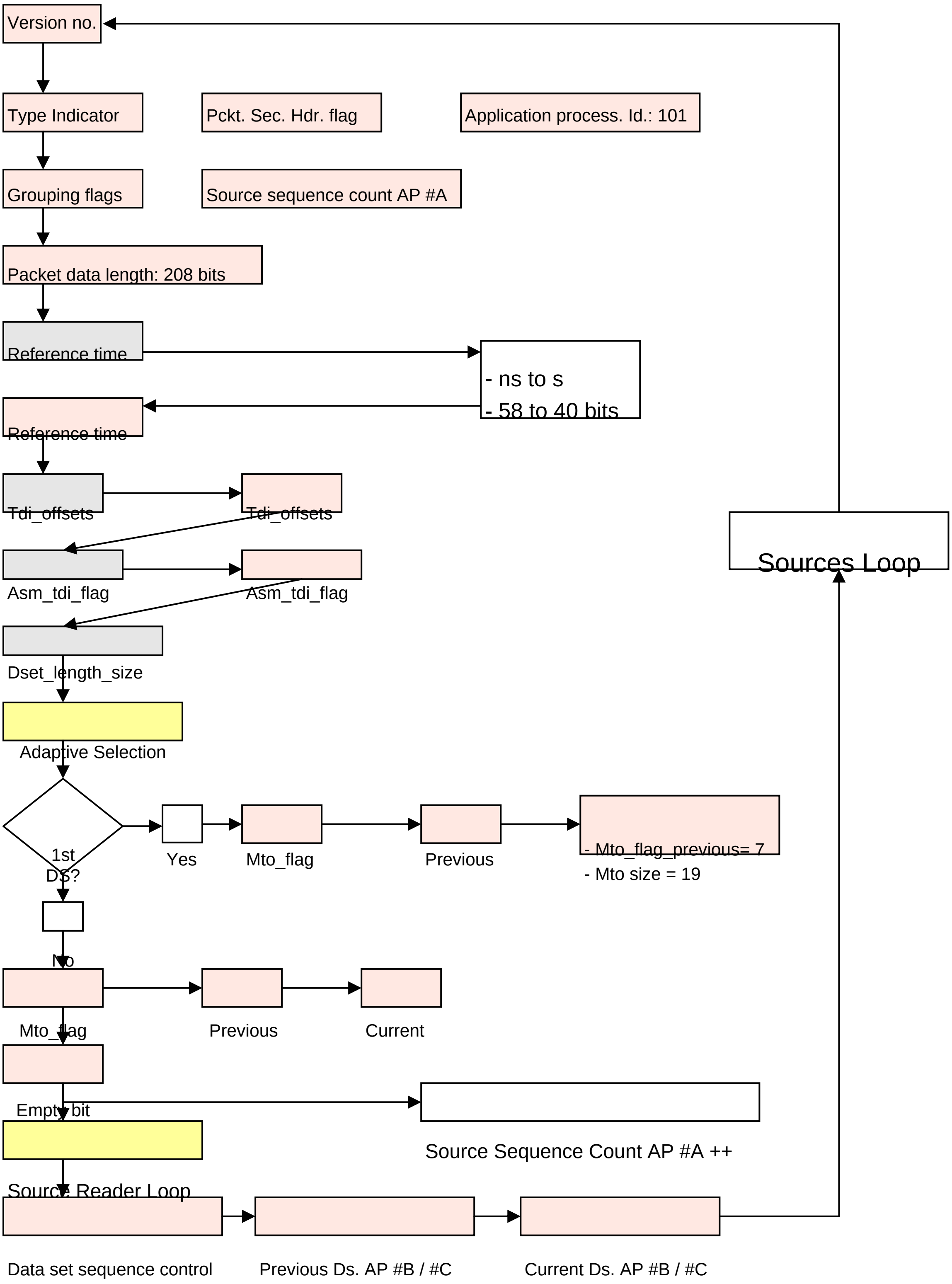


Fig. 3.11 Flux diagram of the generation of AP #A packets

- **Data set sequence control**, divided in two parts: the ***Prev. DS*** values include the ***source sequence count*** values of AP #B to #E corresponding to the first packet of the last Data Set, while ***Curr. DS*** contain these values for the current data set. In our case, AP #D and #E will always have a ***source sequence count*** value of 0, because we are only working with Astro-1 and Astro-2 Data.

The diagram shown in figure 3.11 describes how ***txt2ptm*** runs in order to create the AP #A packet structure. The orange boxes represent the values written to the output file, while the boxes in grey represent the values read from the input file. White and yellow boxes are the processes and loops executed.
It is important to note that the ***Dset_length_size*** data field (Ref. B in Annex B), included in the GASS file [RD.5.], has disappeared in this new data structure. As it will be seen in the next section, every TS header already notes the Number of Sources transmitted in that TS. If we know the number of sources of every TS in the data set, then we will know the total number of sources of the data set. The final aspect of the AP #A output data is illustrated in figure 3.12, where it can be seen in red the new values added to build the header and in blue the values that have been modified.

```
0                                  /* Version no.*/
0 1 101                            /*Type indicator*//*Pckt. Sec. hdr. flag*//**/
3 0                                /* Grouping flags *//* Source sequence count */
248                                /* Packet data length */
7689600                            /* Reference Time*/
301 806 352 249 396 396 762 521 337 491   /* TDI offsets */
513 44 139 660 543 447 374 242 843 872
594 718 389 293 528 843 960 587 557 733
550 73 718 103 59 418 696 938 117 337
975 411 660 880 323 110 960 308 916 880
462 220 249 117 616 630 777 411 667 718
381 887 345 411 755 491 543 161 433 73
88 242 609 117 396 909 572 367 345 557
733 858 374 960 345 271 257 660 183 579
572 110 850 220 968 696 235 73 858 169
301 308 315 982 770 550 587 491 843 220
660 455 418 557 51 689 242 872 220 579
733 403 374 784 828 689 755 689 894 37
455 667 440 916 425 286 894 139 528 88
66 880 447 337 249 271 894 601 110 982
543 587 73 609 191 543 418 462 872 609
850 630 308 440 249 733 154 220 433 740
887 645 601 660 630 762 286 777 535 565
0 0 0 0 0 0 0 0 0 0                       /* ASM TDI Flag */
0 0 0 0 0 0 0 0 0 0
7 5                                       /* MTO flag previous*/ /* MTO flag current */
0                                         /* Empty Bits */
0 0 0 0       /*Data set sequence control, Prev.DS: AP #B, AP #B, AP #C, AP #D*/
0 0 0 0       /*Data set sequence control, Current.DS: AP #B, AP #B, AP #C, AP #D*/
```

Fig. 3.12 Example of Data Set header contents

3.3.2 PTM for AP #B and AP #C

The instruments Astro-1 and Astro-2 transmit as AP #B and AP #C science data. MBP and RVS instruments would transmit with AP #C and #D, but they will not be treated here. The new PTM that Gaia will transmit is shown in figure 3.13.
The most interesting parameters which we will work on are:

- **Application process Id.:** This is the parameter with which we distinguish the TS containing Astro-1 or Astro-2 TS data. The value for Astro-1 TS is 102 and for Astro-2 is 103.
- **Source sequence count:** it is a counter of packets generated by Astro-1 or Astro-2.

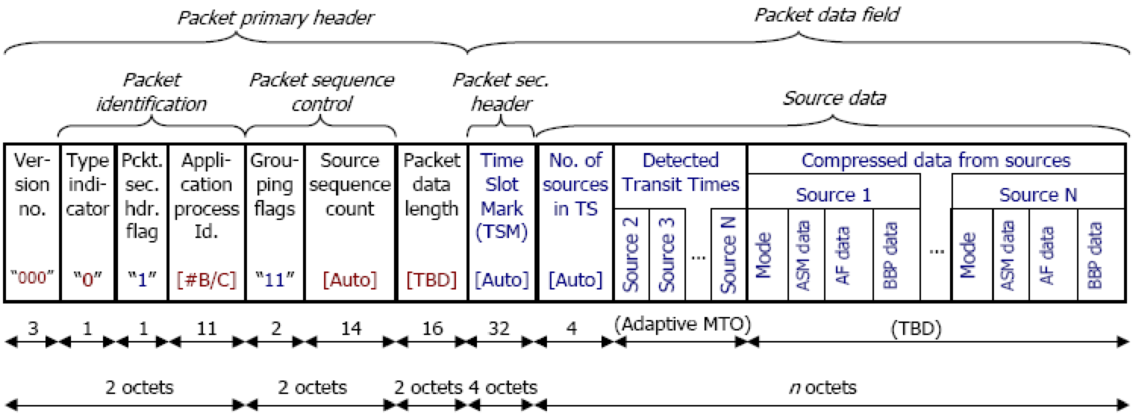
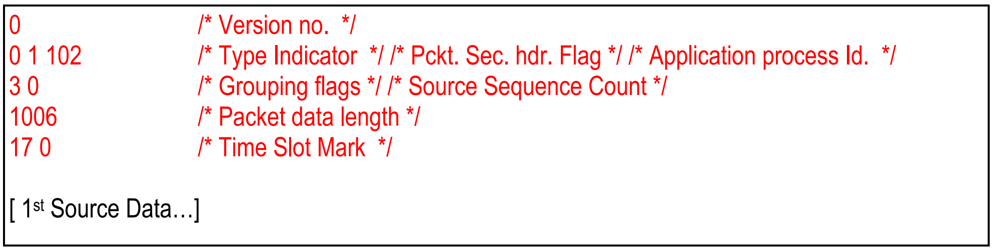


Fig. 3.13 Final PTM structure for AP #B and AP #C data

- **Packet data length:** It represents the size of the current TS, again omitting the packet primary header.
- **Time Slot Mark:** In the current codification it represents the time between the Reference Time of the data set and the current TS start. It is the DTT and SUBDTT of the current source. It is important to note that in the current implementation of ***txt2ptm*** we will keep the original codification (as DTT+SubDTT) and field sizes. Future improvements to this software will implement the adequate CCSDS-compliant formats.
- **No. of sources in TS:** It contains the number of sources transmitted in the current time slot.
- **Detected Transit Time:** The values mentioned in 3.2.2. and 3.2.3, this is, the DTT of the sources transmitted in the current time slot.
- **Version no. / Type indicator / Pckt. Sec. hdr. Flag / Grouping flags**: They contain fixed values.

- In **Compressed data from sources** the scientific data of the sources is transmitted, excluding the DTT and SubDTT values because they are already included in the header of the packet.

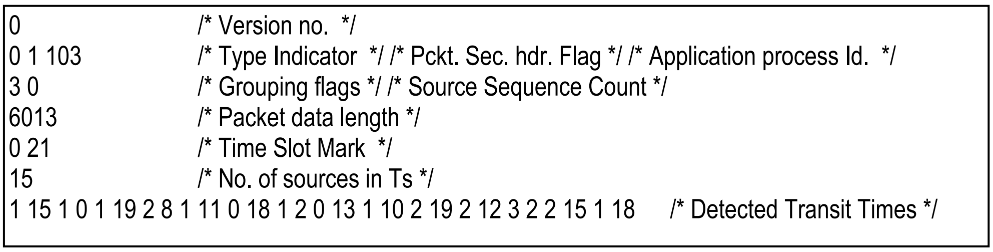
We will show now the organization of the headers for AP #B and AP #C data. Figure 3.14 shows an example taken from a file output by ***txt2ptm*** using the Differential Method.



```
0                /* Version no.  */
0 1 102          /* Type Indicator  */ /* Pckt. Sec. hdr. Flag */ /* Application process Id.  */
3 0              /* Grouping flags */ /* Source Sequence Count */
1006             /* Packet data length */
17 0             /* Time Slot Mark  */

[ 1st Source Data…]
```

Fig. 3.14 AP #B header



```
0                /* Version no.  */
0 1 103          /* Type Indicator  */ /* Pckt. Sec. hdr. Flag */ /* Application process Id.  */
3 0              /* Grouping flags */ /* Source Sequence Count */
6013             /* Packet data length */
0 21             /* Time Slot Mark  */
15               /* No. of sources in Ts */
1 15 1 0 1 19 2 8 1 11 0 18 1 2 0 13 1 10 2 19 2 12 3 2 2 15 1 18      /* Detected Transit Times */
```

Fig. 3.15 AP #C header

As can be seen, the fields ***Version no.***, ***Type Indicator***, ***Pckt. Sec. hdr. Flag*** and ***grouping flags*** are fixed values. The ***application process Id.*** provides the necessary information to know if the time slot contains sources from Astro-1 (102) or Astro-2 (103). On its hand, ***Source Sequence Count*** indicates the correct sequence of packets. The ***packet data length*** field indicates the size in bytes of the current packet data field. The ***Time Slot Mark*** indicates the DTT and SubDTT of the 1st source of the TS, relative to the ***Ref. Time*** to the current data set. And finally, depending on the method used (Absolute or Differential), the Detected Transit Times are transmitted relative to the TSM or to the last DTT. In this first ***txt2ptm*** implementation the SubDTT values have been kept unchanged.

Figure 3.14 displays a especial case of Time Slot: the fields of Detected Transit Time are empty. The reason for this is that the difference between the first and the second source is larger than the Maximum TSM Offset, and therefore only the first source is included in the TS. That is., this time slot contains only one source.

3.4 Performance of the system

The creation of AP #B and AP #C headers are mandatory because of the use of the Packet Telemetry standard. This is the most complex process described in this document. Now we present some of the problems found in the creation of Time Slots:

- The Time Slots are packets of either Astro-1 or Astro-2 sources (exclusively, i.e., not mixed). The input file contains sources from Astro-1 and Astro-2 completely mixed. We need to associate a series of counters to a Time Slot of Astro-1 or Astro-2.
- Another problem is to ensure that the correct number of sources is included in the Time Slot. It will depend on the maximum number of sources for a TS, the Maximum TSM Offset, and the creation of a new Data Set.

In order to better understand the followings explanations it is useful to see a ***.sort*** file (although a GASS file will also be useful). Let us recall how a ***.sort*** file is composed: a sequence of Data Sets each containing a given number of sources of Astro-1 or Astro-2 data, each referring to the Reference Time of the Data Set. In order to solve the first problem we have created a new library, which we have named ***gaia_ts.h***. This library has been created in the ***/misc*** folder of the TM CODEC.

3.4.1 Gaia_ts library

We have created a new C++ class named Cts (TS Class). We have used a philosophy similar to ***txt2sort***. A Cts object has the following fields:

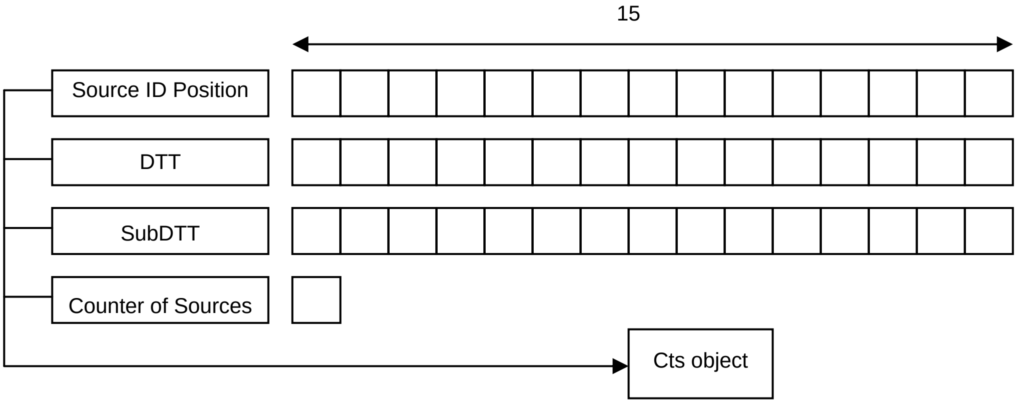


Fig. 3.16 Cts object

The maximum number of positions in the vectors is fifteen, because it is the maximum number of sources that can be saved in a TS; on the other hand, we only need a position for the counter of the number of sources in the TS. A Cts object has a set of functions associated that get and send the values explained before. The advantage of using an object like this one is that it can create a Cts object for Astro-1 or Astro-2 that works independently, having its associated group of functions.

3.4.2 Running

The objective of ***txt2ptm*** is to achieve the new time slot base codification and the insertion of the appropriate headers for AP #A, AP #B and AP#C. The diagram shown in figure 3.17 describes how ***txt2ptm*** operates with the file received from ***txt2sort***. ***Txt2ptm*** creates a Cts object for Astro-1 and Astro-2 data. It also creates a set of counters that are needed for the different Source Sequence counts. After that, it receives the first AP #A data set and it is processed using the diagram of Figure 3.11. When AP #A is written to the output file with the new headers, the first source is then received. During all the Data Set, the objects Astro-1 or Astro-2 are filled with the sources depending of the Source Identifier (see Annex B, Ref. D). When the conditions for creating a new time slot occur in one of the objects, this is sent to a function that adds the necessary headers and prints all the sources of the Time Slot.

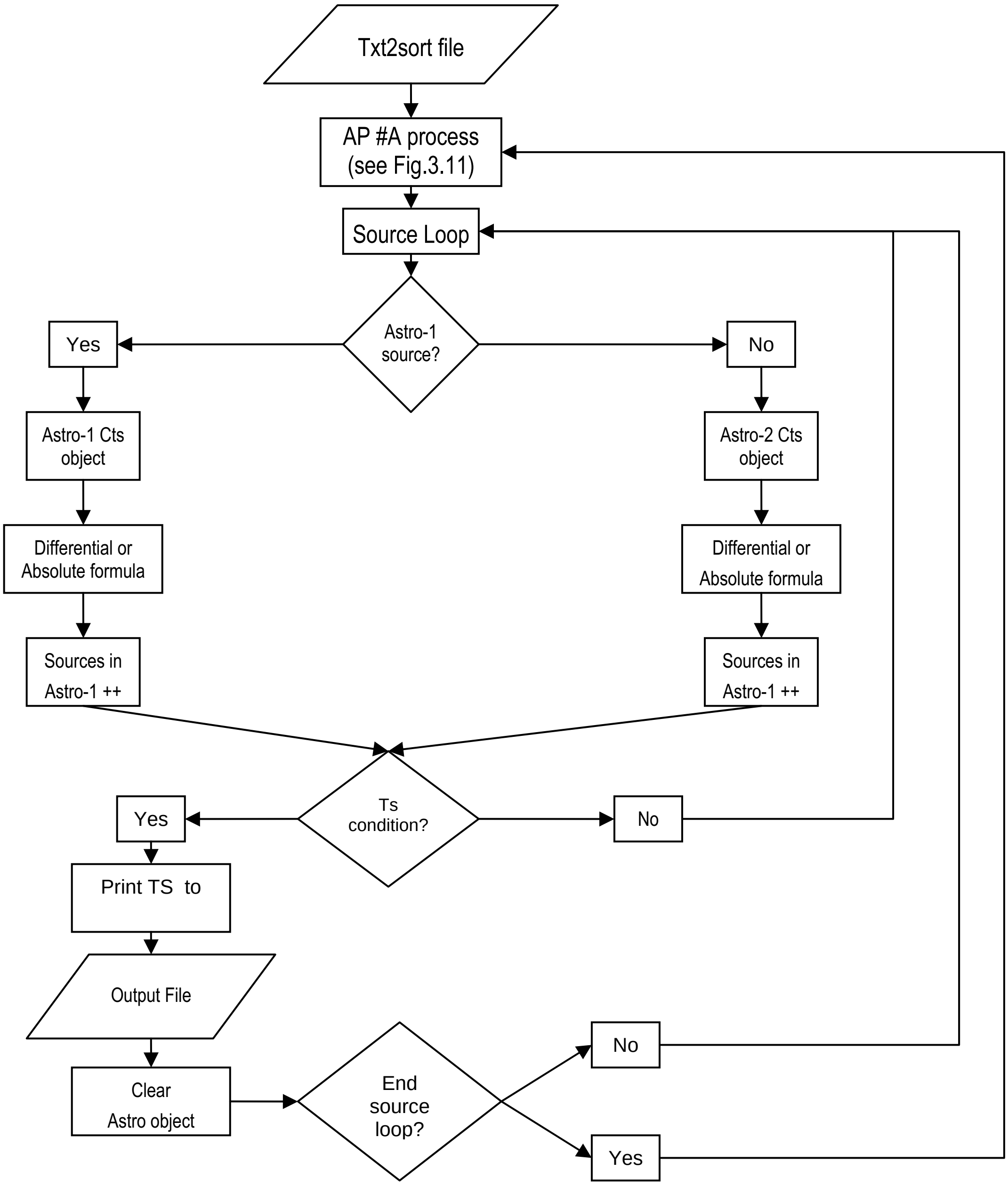


Fig 3.17 Txt2ptm operational flux diagram

3.5 Test using 6 hours of Gaia simulated data

In this section we will show some of the statistics obtained in order to determine the reliability of ***txt2ptm***. The followings graphics have been made with the information that ***txt2ptm*** outputs to the user (see section 3.8 below). The simulations have been run in an AthlonXP 2600+ using the Gentoo GNU/Linux distribution. The files used in the statistics are listed in Table 3.2.

Table 3.2 TM files containing a simulated Great Circle

File Name	Density (stars/second)
d80.00-80.05.tm	484,4143
d80.05-80.10m20.tm	290,5115
d80.10-80.15m20.tm	82,661
d80.15-80.20.tm	268,8704
d80.20-80.25m20.tm	462,831

Each TM file shown in table 3.2 contains 1.2 hours of mission, and hence the five files sum a total of 6 hours. The combination of all five files corresponds to a “Great Circle”, this is, a complete revolution of Gaia on its axis. On the other hand, the processing time, has required an amount of about 3 computational hours per file, yielding a total of 15 hours of computer time. This corresponds only to the computer time required by ***txt2ptm.*** A larger amount of computer time was really required since we have not added the computer time necessary to generate the TM files with GASS. Since the current implementation of ***txt2ptm*** makes possible the selection (in compilation time) of an absolute or a differential operation, it was interesting to establish a comparison between the two codifications. Therefore each file has been processed twice.

In order to better understand the behaviour of Gaia in this simulation (and the “system load” or star density), an FDI plot (Field Density Index, [AD.3.]) has been included in Annex D, where the density peaks corresponding to the scans of the Galactic Plane (where much more stars can be observed) can be observed.

3.5.1 First statistic: bps per second saved by the Differential Method

The first statistic we have devised shows the average amount of bps that the Differential Method saves with respect the Absolute codification. The expression used to obtain it is the following:

(3.3)

Where:

TotalSizeAbsolute→ It is the total size of the file processed with the Absolute Method (in bits)

TotalSizeDifferential→ It is the total size of the file processed with the Differential Method (in bits)

ObservationTime→ It is the time that Gaia has been observing (in seconds)

Figure 3.18 shows the resulting downlink saving in bits per second. The point where the density is about 82,661 sources/second (corresponding to d80.1-80.15.tm.) is highlighted, comparing it to the equivalent period of time in Annex D. It coincides with a point where both Astro-1 and 2 register fewer sources/second. On the other hand, the peak of maximum bps saved is found when both Astro instruments have approximately the same density of sources (see Annex D). Although Astro-1 reaches a peak of roughly 1400 sources/second, the adaptive selection compensates this effect.

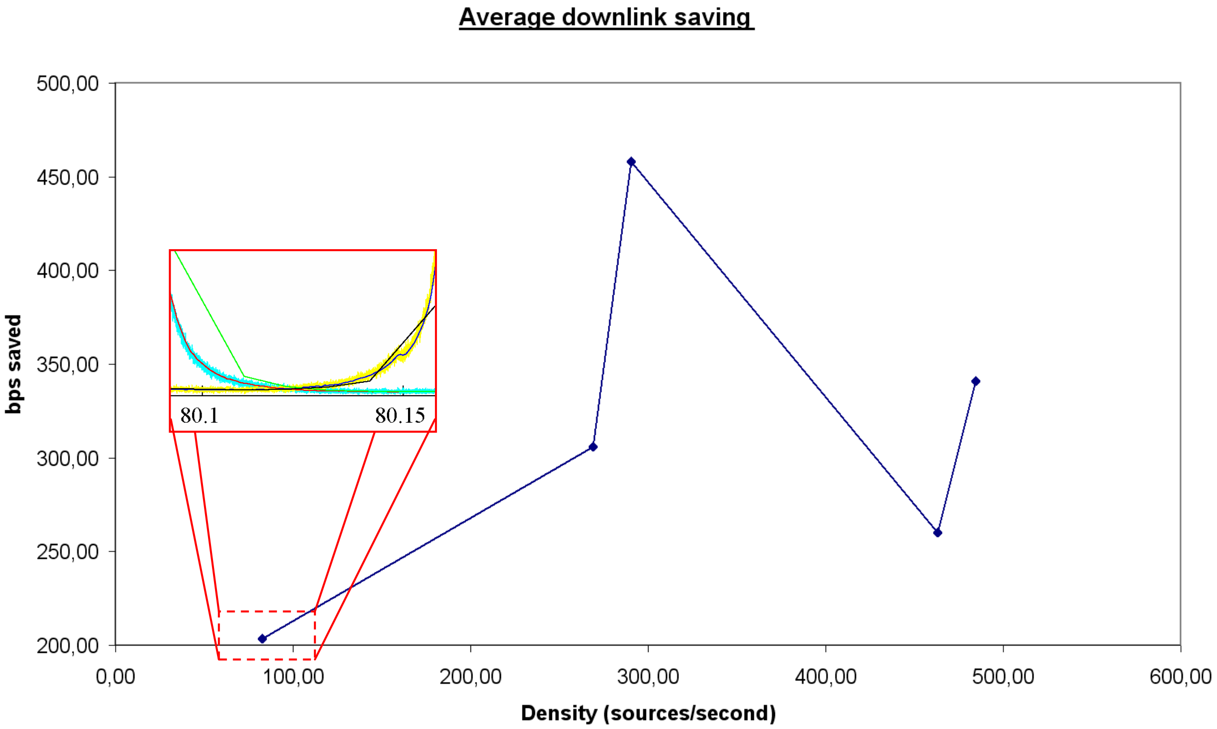


Fig. 3.18 Average bps saved using the differential TDC

3.5.2 Second statistic: Average sources in TS

The difference between the two methods has been studied in section 3.1. There we have shown that the difference lies on the way the sources are referenced: while in the Absolute Method all the sources of the TS are referenced to the first source of the TS, the Differential Method takes advantage of the fact that the DTT of the sources always increase with time and, consequently, the difference between one source and the next one is always smaller than in the Absolute Method. As the creation of TS will depend on this difference, a Differential Method will always fit more sources in a given TS. This has been already verified, and it is graphically shown in figures 3.19 and 3.20. These figures show the difference in the number of sources per TS that appear when we use each one of the two algorithms independently, either the Differential or the Absolute method. While, for example, on average in Astro-1 only 7 sources are transmitted in every TS, the Differential Method includes up to 11 per TS, which is a considerable increment. For instance, in Astro-1, at about 462 sources/second we find the maximum number of sources in a time slot. Astro-2, at the same density, offers the minimum number of sources/TS. It can be seen in Annex D that in this interval of time (d80.20-80.25) the highest peak of density for Astro-1 is reached while a minimal amount of sources/second is measured by Astro-2. We can see as well the opposite case at about 485 sources/second. In this case, Astro-2 is the instrument which has more sources per TS, while Astro-1 has fewer sources per TS.

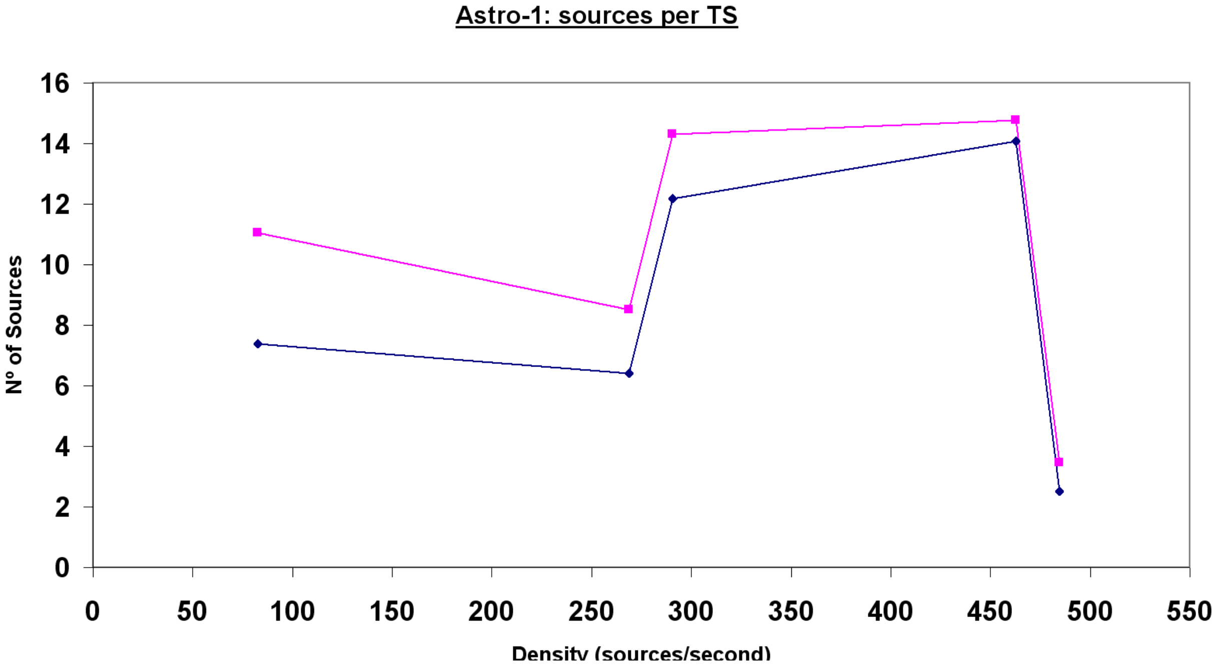


Fig. 3.19 Astro-1 Sources per TS

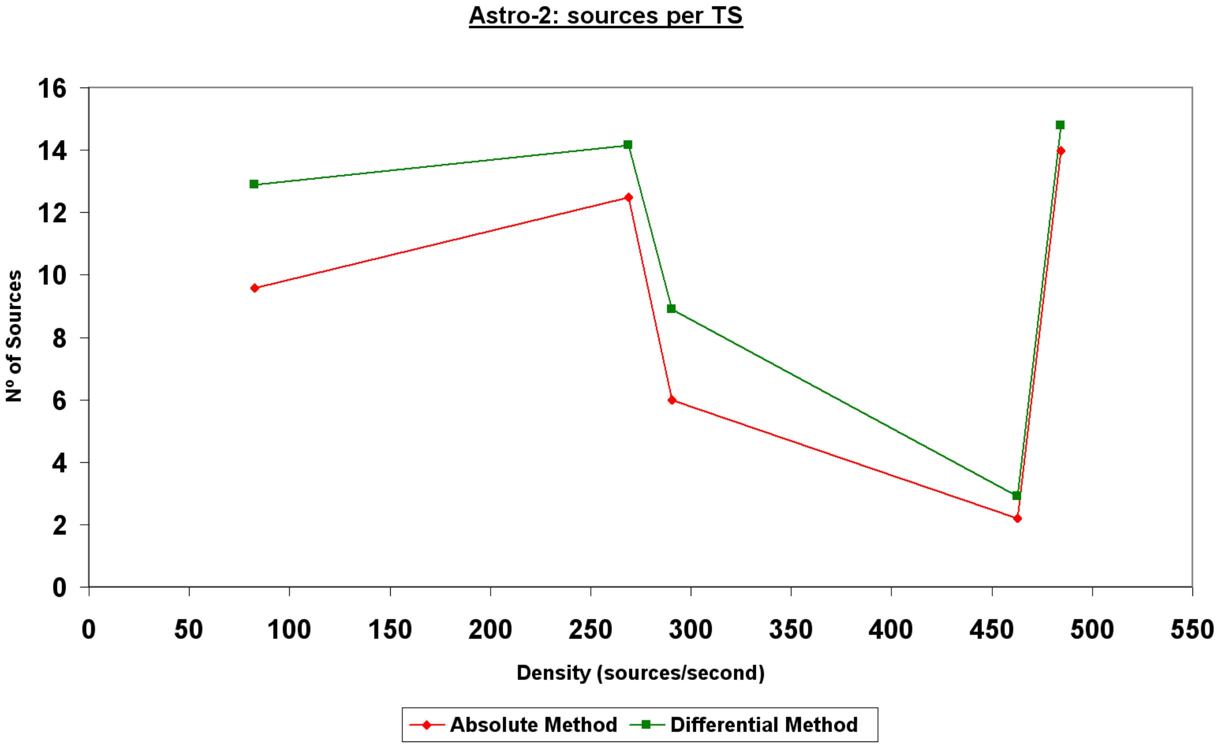


Fig. 3.20 Astro-2 Sources per TS

3.5.3 Third Statistic: Number of TS created

It is as well straightforward to see that since the total number of stars measured in a simulation is fixed, if we pack more sources in a TS then fewer TSs will be generated. This can be seen in Figs. 3.21 and 3.22, which show the number of TS/second in Astro-1 and Astro-2.

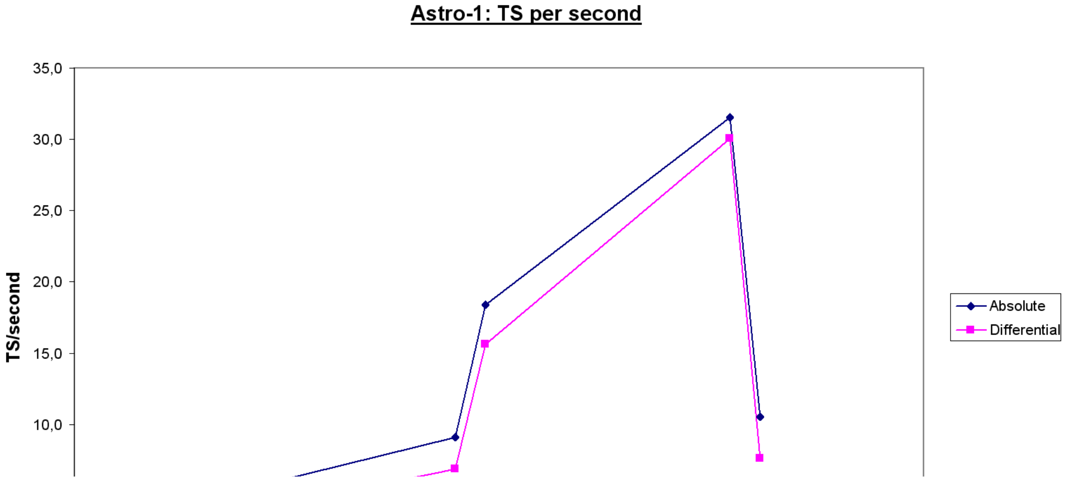


Fig. 3.21 Astro-1 TS/second

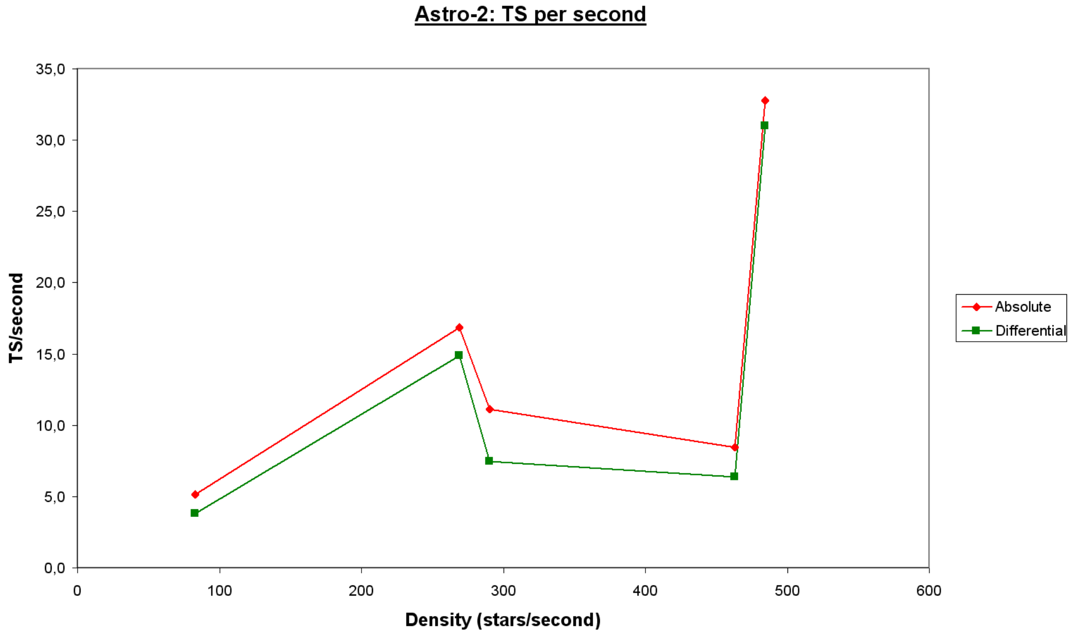


Fig. 3.22 Astro-2 TS/second

As it can be seen in these plots, the Differential method creates fewer TSs every second. Between 462 sources/second and 484 sources/second the two lines are very close; this situation is similar to that one previously found in d80.10-d80.15, where a valley of density is reached. Clearly, the smaller the number of sources/second is, the smaller the amount of TS created. The value registered for a density of 462 is approximately 33 TS/second which is the maximum value of these statistics, whereas the point at 82 sources/second corresponds to the minimal value of density of TS. The maximum value corresponds in Annex D to a peak of Astro-1 sources, while the minimum corresponds to the instant where the valley of density is reached. On the other hand, in Astro-2 the peak of generated TS/second is found where in Annex D we see the maximum value of the density index for Astro-2 (d80.00-d80.05). Additionally, the minimum value is located in 80.10-80.15 where the valley in the density index appears. Both figures have in common that the new Differential Method creates fewer TS/second, as we already have demonstrated that more sources are packed in the same TS. Consequently, the overhead introduced is smaller.

3.5.4 Fourth statistic: Overhead added by the headers

As explained before, every TS is accompanied by a header and, consequently, a certain amount of bits is added to the file size. Obviously, the fewer TS created, the lower the amount of bits we add. Consequently, the size of the output file will be smaller as well. Figure 3.23 illustrates the percentage of headers added to the input file when processing it with ***txt2ptm*** using the Absolute or the Differential method. It is evident that the Differential Method creates fewer TS/second. It is the codification that generates fewer headers, so its files have a smaller size.

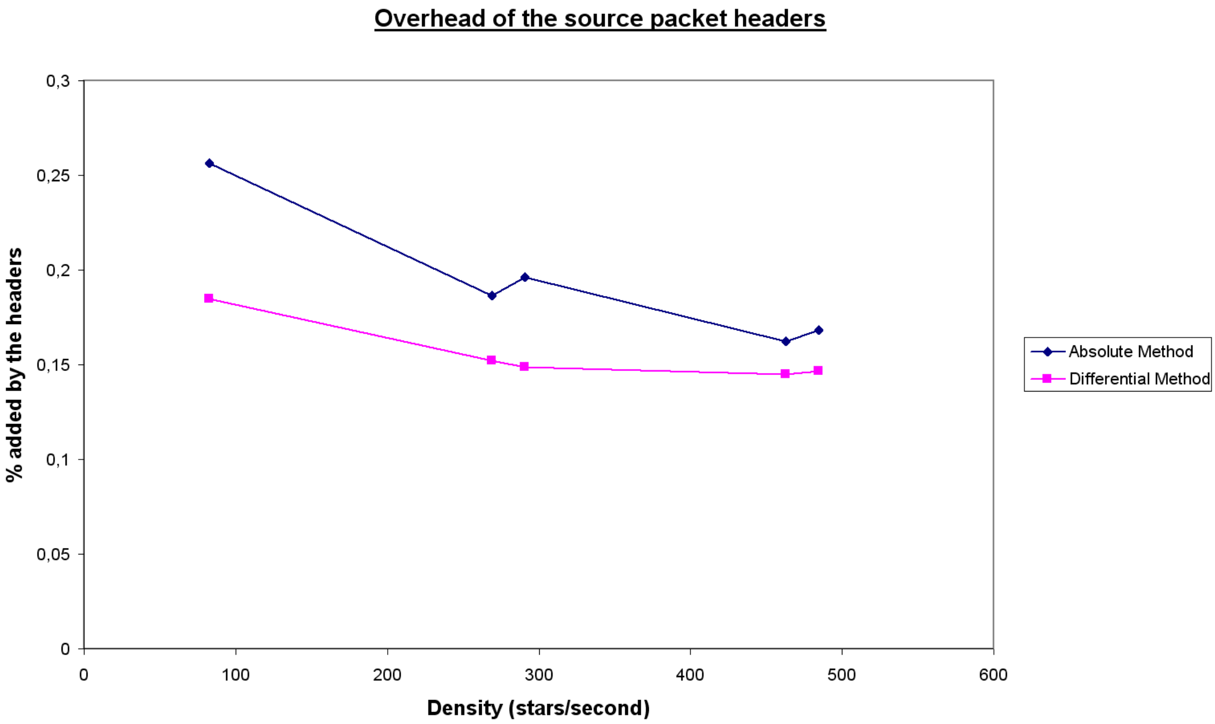


Fig. 3.23 Overhead introduced by the headers

3.5.5 Fifth statistic: File size saved

In Figure 3.24 it is shown the percentage that the Differential Method saves using its codification with respect to the same file using the Absolute Method. It is interesting to see that it is in the valley of Fig. 3.24 in Annex D (corresponding to d80.10-d80.15) where the two Astro instruments have in average the same number of sources, where more bits are saved. On the other hand, when one of the two Astro instruments is more loaded than the other, fewer bits are saved. The reason resides on the fact that the adaptive method determines the MTO depending of the total amount of sources (Astro-1 plus Astro-2) that are included inside the Data Set. Since in general one of the two Astro instruments is more loaded than the other, this parameter runs more efficiently with the Astro instrument that has more sources.

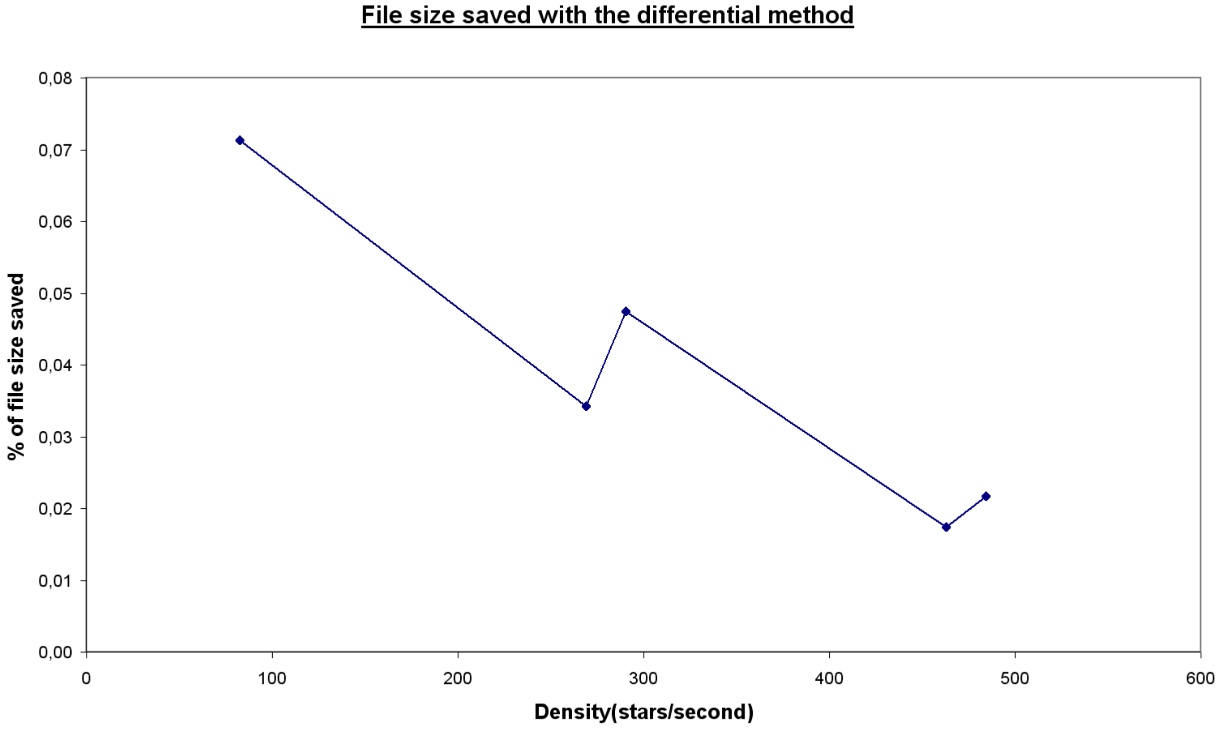


Fig. 3.24 File size Saved

3.6 Other Tests of the System

It is interesting to study as well not only a typical case as the Great Circle analyzed before. For this reason, we present now a series of small simulations, where the densities cover a much larger range. The statistical analyses will be the same as before. The files used are:

Table 3.3 Short simulation files

File name	Start Density
d91m20.tm	71,1667
d80m20.tm	74,9537
d80.065m20.tm	137,2778
d80.087m20.tm	973,7315
d91.045m20.tm	2314,3518
d120.185m20.tm	2452,4167
d120.0.tm	6664,6665
d120.06m20.tm	11165,7998

3.6.1 First statistic: bps per second saved by the Differential Method

As it can be seen in Fig. 3.25 at a density of about 6664 stars per second it appears a low value of bps saved. As it can be observed in in Figs. 3.26 and 3.27 there is a small difference between the number of TS created by Astro-1 and Astro-2.

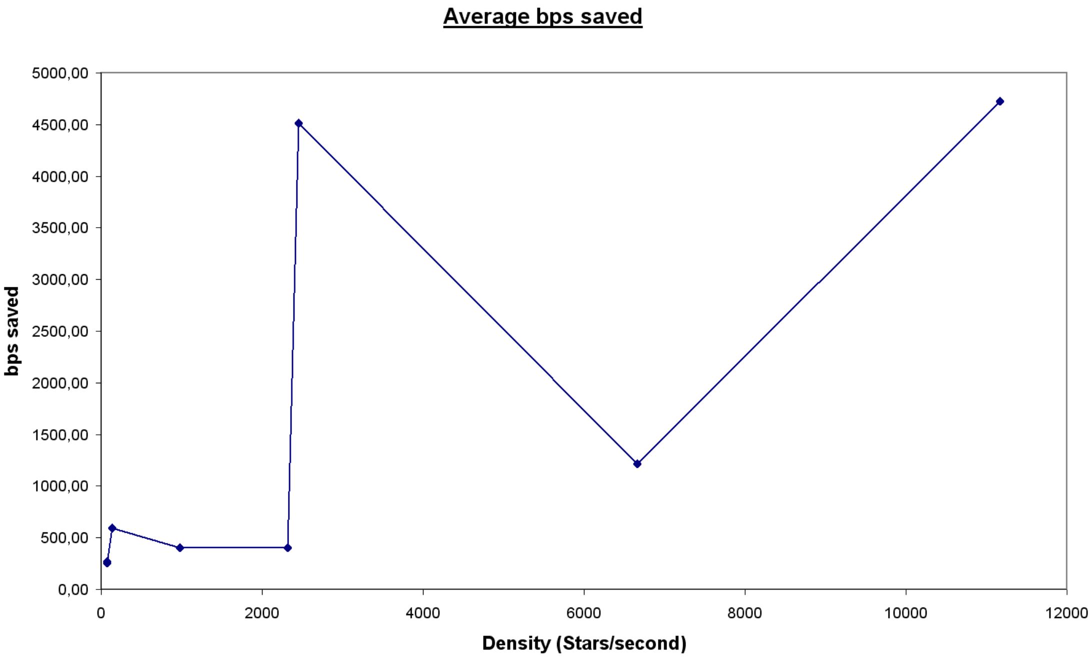


Fig. 3.25 Average bps saved

3.6.2 Second statistic: Average sources per TS

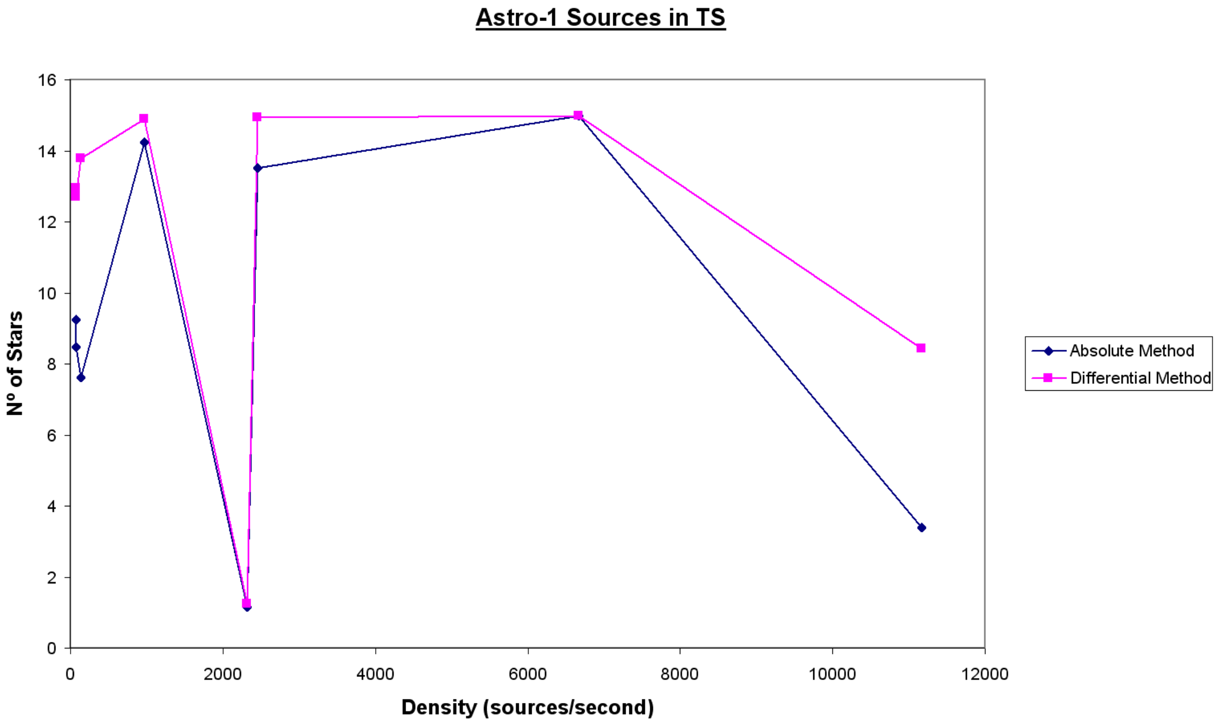


Fig. 3.26 Average sources per TS in Astro-1

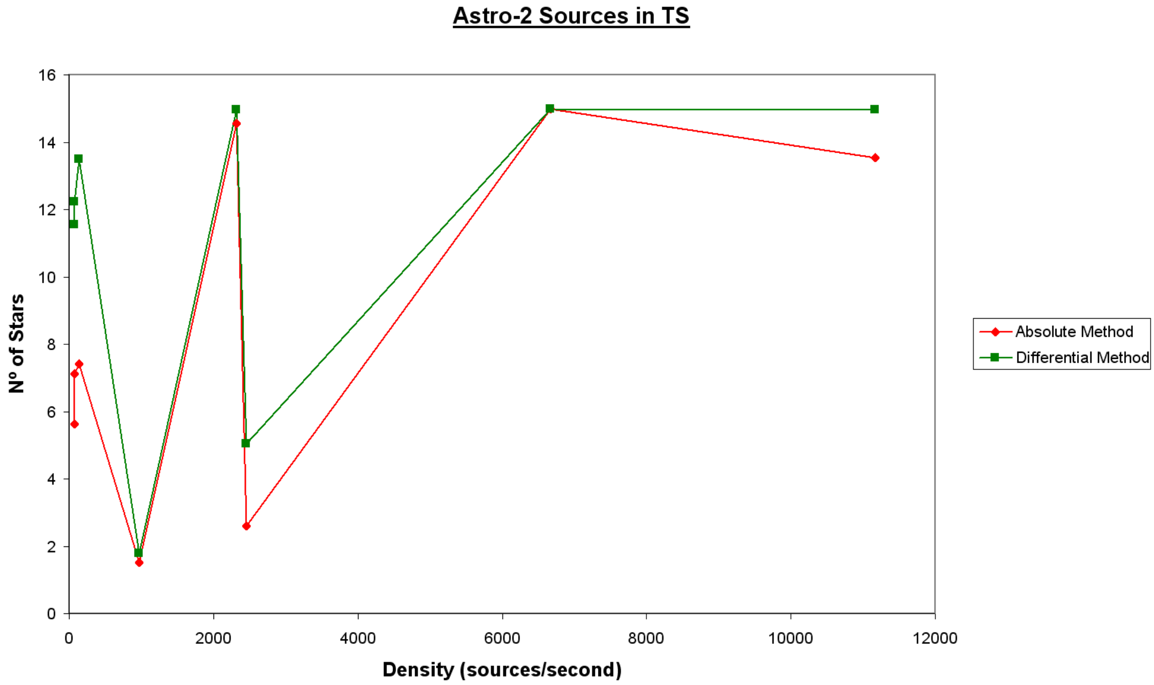


Fig. 3.27 Average sources per TS in Astro-2

In Figs. 3.26 and 3.27 it can be observed that the Differential Method is able to fit more sources in every time slot, and curiously, this happens when it is subject to extreme conditions. For example this occurs when there are high densities and high differences between the load of the two Astro instruments. In particular, this is the situation found when the respective loads of Astro-1 and Astro-2 are 72 sources/second and 11,166 sources/second. The differences between the peaks of the graphics consequently arise because of the load of the Astro instruments (see the Annex E for more details).

3.6.3 Third statistic: Number of TS created

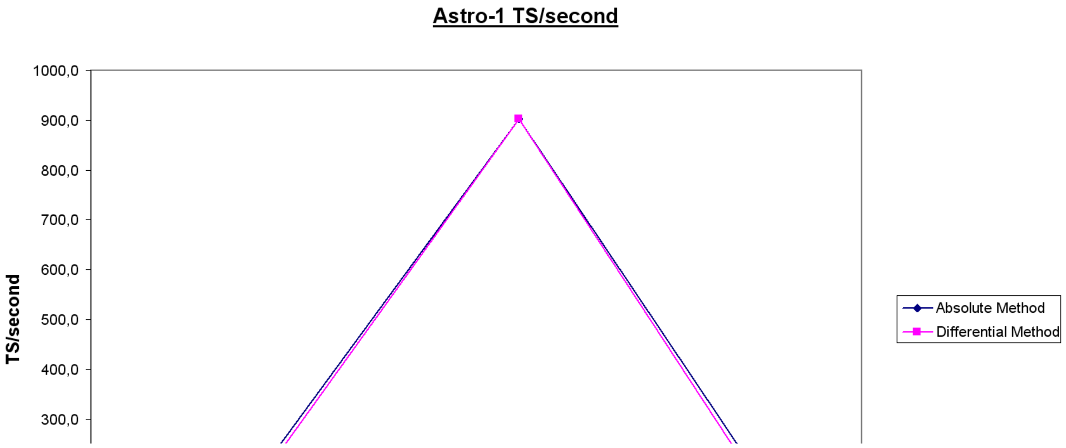


Fig. 3.28 Astro-1 TS/second

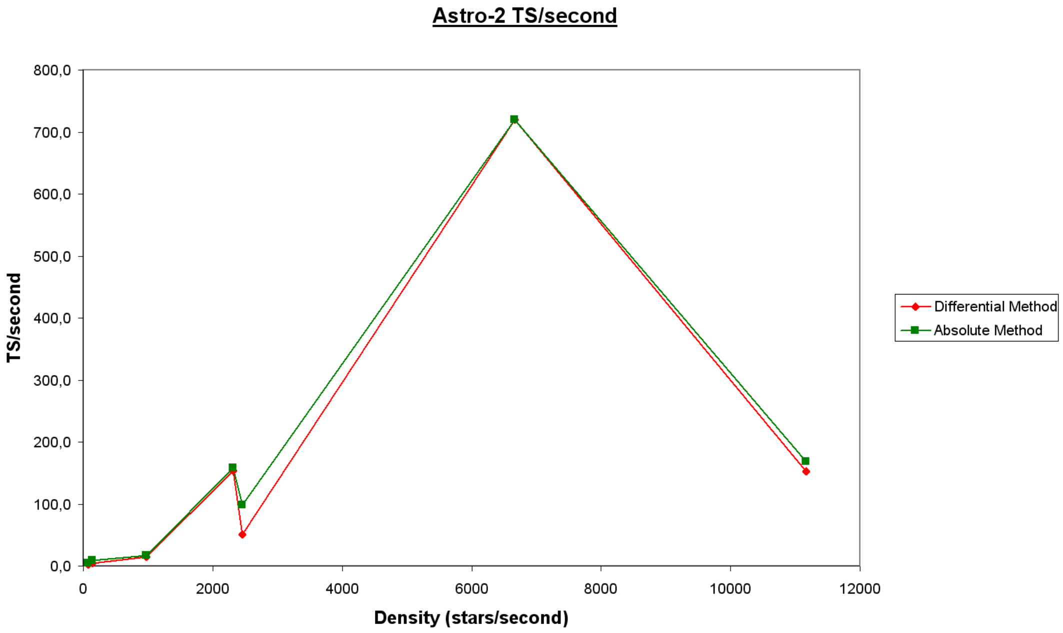


Fig. 3.29 Astro-2 TS/second

These figures have a similar behaviour to those already discussed in section 3.6.2. When the two Astro instruments are highly loaded, the differences between the Differential and the Absolute Method are minimal. However, when one of the two Astro instruments generates more TS/second than the other, the Differential Method behaves considerably better than the Absolute Method.

3.6.4 Fourth statistic: Overhead added by the headers

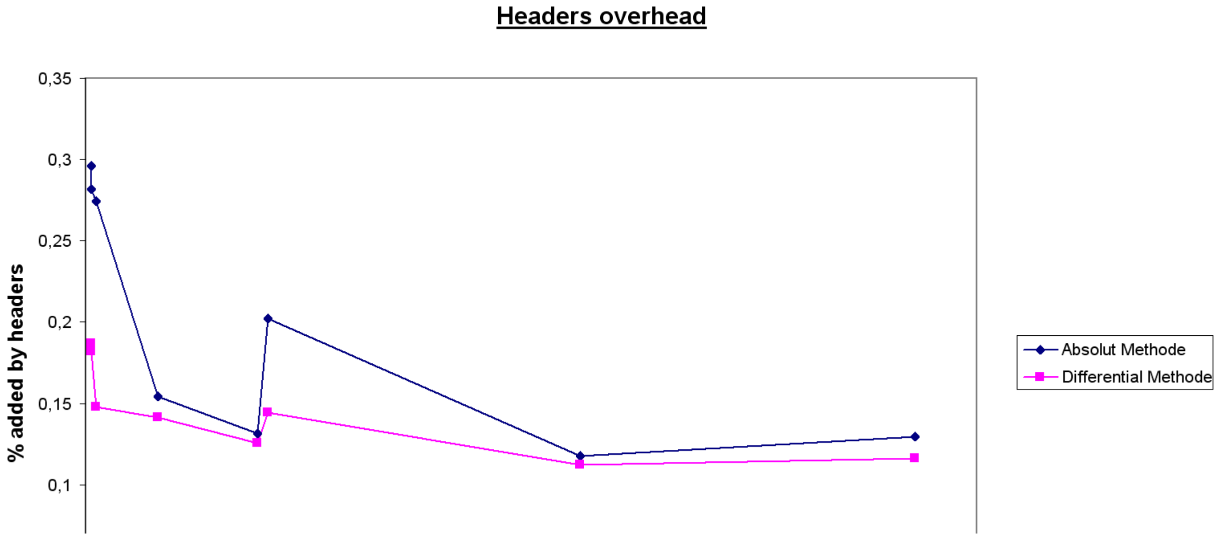


Fig. 3.30 Percentage of overhead added by the headers

Obviously, fewer bits are added as fewer TS are created. As it can be corroborated from Fig. 3.30 this is the reason why the Differential Method adds fewer bits to the file. It is important to see that at low densities (where less TS would have to be created), a large saving in the percentage of headers is obtained.

3.6.5 Fifth statistic: file size saved

As Fig. 3.31 shows, it is at about 134 sources/second where it is registered the maximum saving that the Differential Method can do with respect to the Absolute Method. The reason is that in this point, where the density is particularly low, the two Astro instruments deal with a similar quantity of sources (particularly, the Astro-1 instrument is processing 7594 sources/second, whereas the Astro-2 instrument is processing 7232 sources/second). When both Astro instruments process the same quantity of sources per second, the average saving is higher than if one of the two Astro instrument has fewer sources than the other.

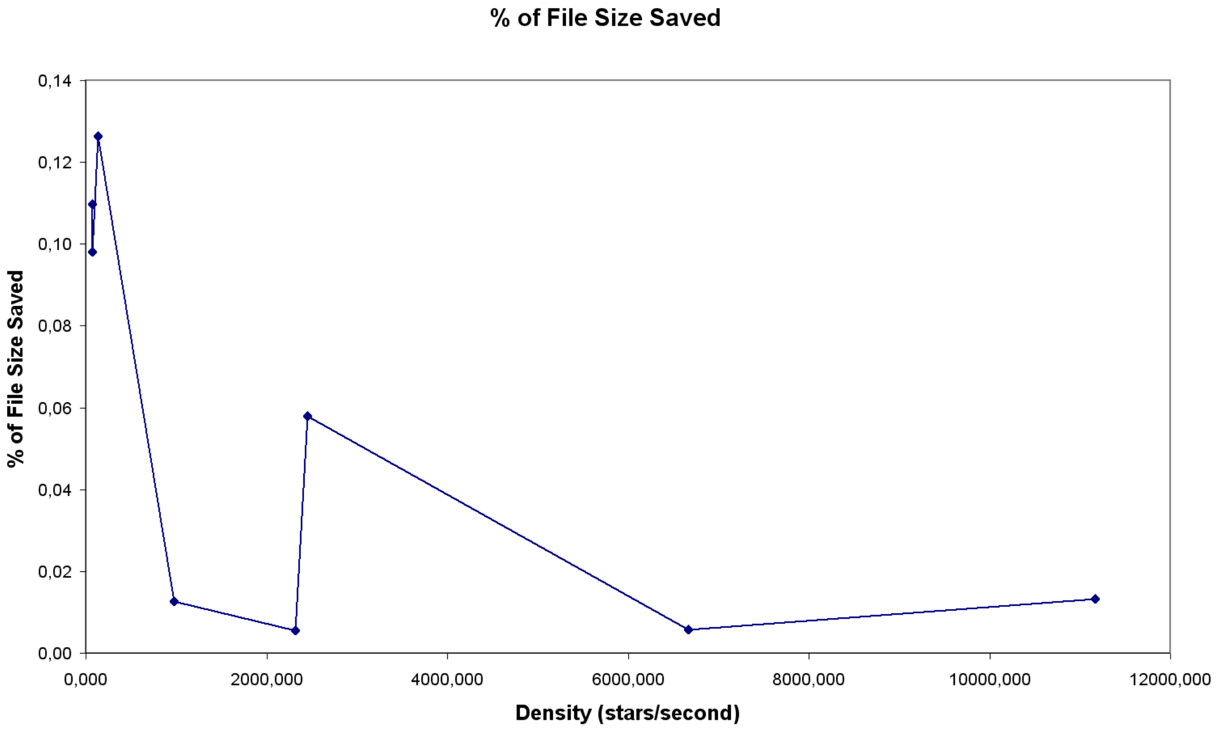


Fig. 3.31 Percentage of file size saved

3.7 General behaviour

Finally, it is interesting to study the general behaviour of points 3.5 and 3.6, this is, compiling all the simulations in a single plot. Here we show the amount of bps saved by the Differential Method when compared to the Absolute Method. This is done in Fig. 3.32.

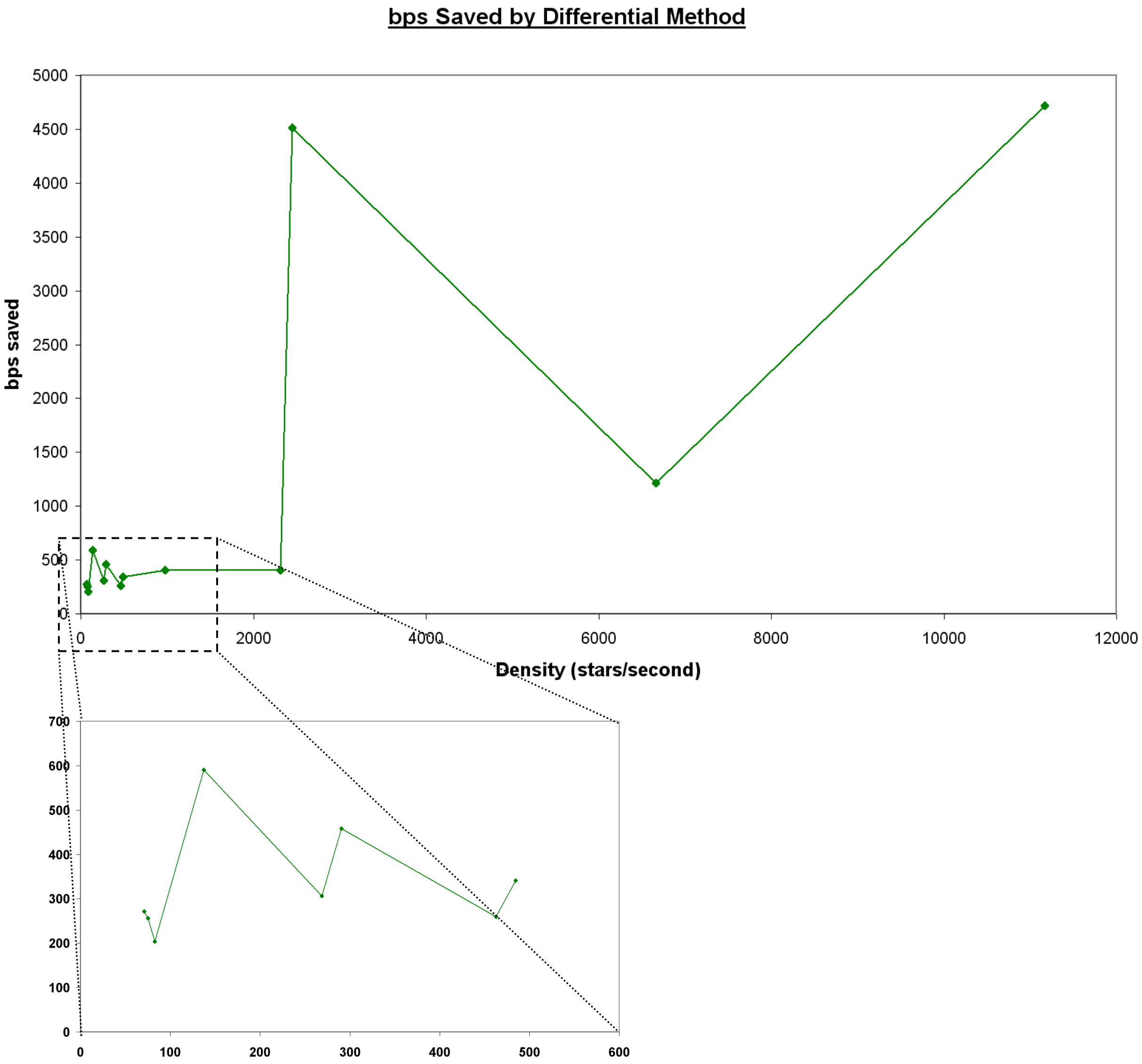
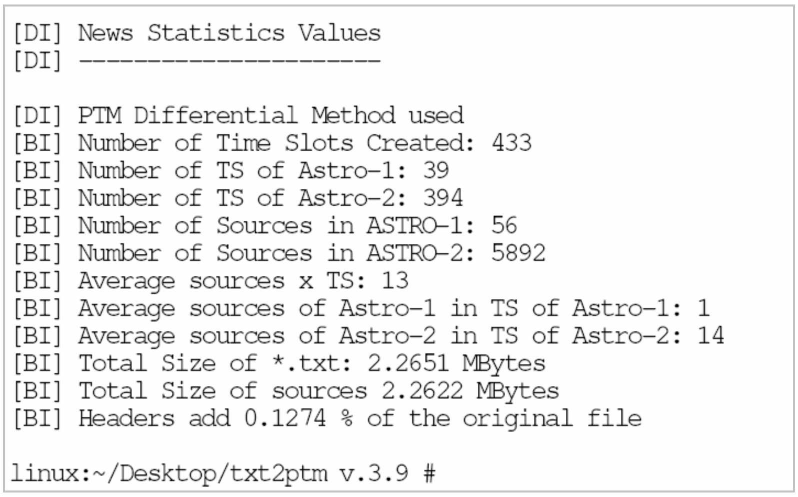


Fig. 3.32 Amount of bps saved by the Differential Method

This graphic combines the graphics of points 3.5.1. and 3.6.1. In the high density zone the values of 3.6.1. appear, where the simulations have a high density whiles the low density zone is dominated by the point 3.5.1.

3.8 New statistical information

Following the philosophy of ***txt2bin***, ***txt2ptm*** also presents a new set of statistics that give us helpful information about the creation of TS. The fields that we can see in a linux console when ***txt2ptm*** has run are shown in Fig. 3.33.



```
[DI] News Statistics Values
[DI] ----------------------

[DI] PTM Differential Method used
[BI] Number of Time Slots Created: 433
[BI] Number of TS of Astro-1: 39
[BI] Number of TS of Astro-2: 394
[BI] Number of Sources in ASTRO-1: 56
[BI] Number of Sources in ASTRO-2: 5892
[BI] Average sources x TS: 13
[BI] Average sources of Astro-1 in TS of Astro-1: 1
[BI] Average sources of Astro-2 in TS of Astro-2: 14
[BI] Total Size of *.txt: 2.2651 MBytes
[BI] Total Size of sources 2.2622 MBytes
[BI] Headers add 0.1274 % of the original file

linux:~/Desktop/txt2ptm v.3.9 #
```

Fig. 3.33 Console Statistics presented by txt2ptm

CHAPTER 4. CONCLUSIONS

4.1 Technical conclusions

With ***txt2ptm*** the TM CODEC has achieved the objective of offering a more realistic TDC in this concept phase of the mission. Following with the philosophy of ***txt2bin***, two main programs have been developed to achieve this objective:

- **Txt2sort**: It was developed as a consequence of the necessity of sorting all the sources of a data set by the DTT field. The sort algorithm selected was the Quicksort algorithm, which was the fastest algorithm, as seen in a comparative study, performed in Chapter 2. Possibly the best solution would be that the GASS simulator could offer the sources already sorted, so an implementation of ***txt2sort*** inside GASS would be an interesting option in order to get closer to the real operation that Gaia will have.

- **Txt2ptm:** It is the core of the implementation of the TDC. Its main functions are:
  - Define a Time Slot base philosophy inside the Data Sets, where every TS contains only sources of Astro-1 or Astro-2.
  - The implementation of the adaptive timing scheme, which was able to adapt the size of a TS depending on star density. The difference between the DTT of two sources and its comparison with the Maximum TSM Offset defines the amount of TS created.
  - The addition of the necessary headers for the creation of the TS and the flight-realistic TDC.

While the Adaptive Selection was being developed, we realized that there was a new way to determine the references between the sources: the Differential Method. We have implemented then this method as well. This method is able to take advantage of the resources and can pack more sources in the same TS. The consequences of using this method, as the statistics report, have been an approximate saving of 400 bps in star densities of 290 sources/second, and savings of 0,2% of the total size of a file. Although these values may seem small at first glance, it must be taken into account that this is achieved by simply saving packet headers: the variable code sizes presented in [RD.4.] have not been implemented whatsoever yet. Also, it has been achieved by only introducing relatively small changes in the code of the Gaia TM CODEC. Finally, taking into account the size of the information that Gaia will manage during the mission lifetime (about 1 TB of information), the Differential Method represents an interesting option that should be further considered.
The best statistical values reached by the Absolute Method occur when the two Astro instruments are equally loaded. In this case the two codifications have a similar behaviour. However, when one of the Astro instruments is more loaded than the other, the Differential Method offers the best efficiency. Finally, ***txt2ptm***

represents a step further towards the inclusion of the compression stage in the packet telemetry and the TDC.

4.2 Personal conclusions

My main motivation for choosing this master thesis was that it merged my two favourite preferences: space sciences and computers. I have been working in the IEEC (Institut d'Estudis Espacials de Catalunya) in a scientific group composed of a Doctor in Physics, an Electronics Engineer and two Aeronautics Engineers. I learned to work using different philosophies, but always inside a rigorous research center. I achieved new knowledge on data compression tools, as well as the basic knowledge on Linux which was absolutely unknown to me.

CHAPTER 5. REFERENCES

5.1 Applicable Documents

Reference	Title	Code/Authors	Version	Date
[AD.1.]	Proposal of a Payload Data Handling Unit and internal data flux	GAIA-BCN-004	2.1	2003/06/17
[AD.2.]	Definition of a Telemetry CODEC	GAIA-BCN-011	1.2	2004/12/22
[AD.3.]	Telemetry and statistical tests with GASS data and TM CODEC	GAIA-BCN-012	1.2	2005/01/10

5.2 Reference Documents

Reference	Title	Code/Authors	Version	Date
[RD.4.]	Timing and transmission schemes for Gaia	GAIA-BCN-006	1.1	2004/03/13
[RD.5.]	Telemetry data formats in simulations	GAIA-BCN-007	1.5	2004/03/17
[RD.6.]	http://pages.stern.nyu.edu/~panos/java/Quicksort/index.html			

Annex A Definitions and Acronyms

AP	Application Process
Bps	Bits per second
CODEC	Coder/Decoder
CCSDS	Consultative Committee for Space Data Systems
Cts	Time Slot Class
DS	Data Set
DTT	Detected Transit Time
FDI	Field Density Index
GASS	Gaia System Simulator
HTM	Hierarchical Triangular Mesh
IEEC	Institud d'Estudis Espacials de Catalunya
Kb	Kilo bits
Mb	Mega bits
MB	Mega bytes
MBP	Medium Band Photometer
MTO	Maximum Time Slot Offset
PTM	Packet Telemetry
RVS	Radial Velocity Spectrometer
TBD	To be determinate
TDC	Time Data Codification
TDI	Time Delayed Integration
TM	Telemetry
TS	Time Slot
TSM	Time Slot Mark
UB	Universitat de Barcelona
UPC	Universitat Politècnica de Catalunya
wrt	With Respect To

Annex B Data files, example of a GASS file

```
7689600000000000
2195 5885 2579 1800 2919 2888 5592 3839 2487 3578
3771 313 1005 4815 …
[…More TDIs Offset values…]
… 4272 530 4454 1378 3954 3070 3407 6377 4487
6253 4609 2244 3210 1844 5395 1122 1631 3196 5409
6525 4735 4394 4823 4602 5585 2085 5721 3899 4157
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
75

0002341225000035
0
1 0 0
99
0 0 0 1.2860841038032103 4.094151406068948 134
6 729 12
965 9
401 101 121 112 405
101 104 176 142 107
121 176 1519 884 231
112 142 884 533 173
405 107 231 173 430
20097
12 12
8859 738
679 846 963 1391 4190 11301 11018 4003 1523 1310 976 754
13760 740
261 472 732 2702 9374 11844 4870 1179 876 477 305 160
18660 742
240 428 639 2084 8099 12397 6032 1364 902 533 326 187
23559 744
223 401 769 1467 5396 11585 8289 2369 1269 888 434 222
28459 747
236 388 470 2029 9104 12696 5499 1089 842 498 268 168
52958 757
217 283 381 647 3421 11199 11387 3406 923 728 420 254
57831 761
226 227 218 221 247 241 244 243 259 245 271 301 288 371 222 307 411 826 4648 12373 9700 2375
842 742 333 220 358 309 253 257 246 238 246 244 237 219 275 218 230 210
62756 762
108 99 107 116 110 138 310 1285 1193 263 161 117 107 114 87 114
65756 763
[…More AF values…]
```

Ref. C: Reference Time

Ref. B: DSET_LENGTH_SIZE or # of stars/second

Ref. D: Source Identifier

Ref. A: DTT

Annex C TXT2SORT statistics, speed of execution

Table C.1 Bubble statistics

The Bubble

# Sources	Sample 1	Sample 2	Sample 3	x
267	0,000	0,000	0,000	0,000
1500	0,020	0,020	0,020	0,020
5000	0,290	0,300	0,270	0,287
7550	0,620	0,660	0,660	0,647
10000	1,140	1,140	1,180	1,153
12500	1,850	1,850	1,800	1,833
15550	2,940	2,890	2,850	2,893
20002	4,870	4,920	5,050	4,947
23000	6,570	6,480	6,510	6,520
27830	9,860	9,580	9,740	9,727
32000	12,650	13,020	12,860	12,843

Table C.2 Direct Insertion statistic

Direct Insertion

# Sources	Sample 1	Sample 2	Sample 3	x
267	0,000	0,000	0,000	0,000
1500	0,000	0,000	0,000	0,000
5000	0,080	0,080	0,090	0,083
7550	0,190	0,190	0,190	0,190
10000	0,360	0,350	0,360	0,357
12500	0,560	0,580	0,540	0,560
15550	0,850	0,860	0,930	0,880
20002	1,520	1,570	1,520	1,537
23000	2,060	2,030	2,110	2,067
27830	3,050	3,000	3,100	3,050
32000	4,110	4,180	4,130	4,140

Table C.3 Direct Selection statistic

Direct Selection

# Sources	Sample 1	Sample 2	Sample 3	x
267	0,000	0,000	0,000	0,000
1500	0,000	0,000	0,000	0,000
5000	0,040	0,050	0,040	0,043
7550	0,110	0,100	0,110	0,107
10000	0,180	0,180	0,200	0,187
12500	0,270	0,290	0,290	0,283
15550	0,490	0,470	0,460	0,473
20002	0,850	0,800	0,840	0,830
23000	1,100	1,130	1,160	1,130
27830	1,750	1,750	1,750	1,750
32000	2,330	2,320	2,330	2,327

Table C.4 Quicksort statistic

Quicksort

# Sources	Sample 1	Sample 2	Sample 3	x
267	0,000	0,000	0,000	0,000
1500	0,000	0,000	0,000	0,000
5000	0,000	0,000	0,000	0,000
7550	0,000	0,000	0,000	0,000
10000	0,020	0,020	0,010	0,017
12500	0,020	0,010	0,020	0,017
15550	0,020	0,030	0,020	0,023
20002	0,040	0,030	0,040	0,037
23000	0,050	0,050	0,060	0,053
27830	0,060	0,060	0,040	0,053
32000	0,070	0,070	0,070	0,070

Annex D Features of a Great Circle of Gaia

D.1 FDI from d80.00 to d80.25

The following graphic is useful to understand the statistics of 3.5:

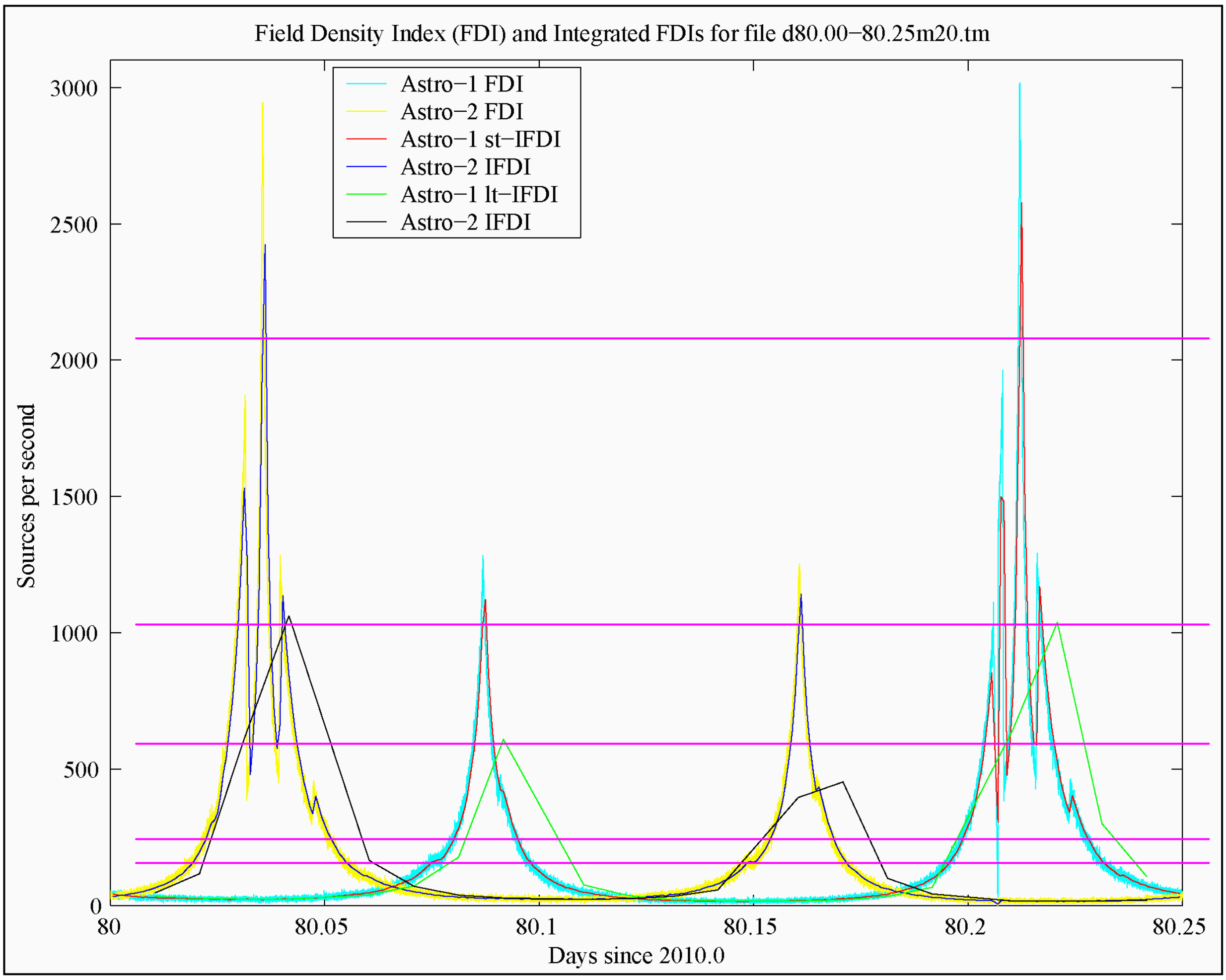


Fig. D.1 FDI and Integrated FDIs from d80.00 to d80.25

The integration times of the different curves are:

- FDI: 1 second
- st-IFDI: 1 minute
- 1t-IFDI: 15 minutes

The main feature of this graphic is that while one of the instruments is observing a high density field, for example Astro-1 between 3000 sources/second from the 80th mission day to 80.05, the other instrument is observing a low density field of a few sources/second. This difference of values is a consequence of the intrinsic operation of Gaia, but the differences in the field densities in these instruments makes possible to study the instruments independently.

The purple horizontal lines note the changes in the MTO. The values of MTO can be found in Table 3.1. We recall that the MTO value is able to adapt the size of a TS taking into account the star density. It is an important value to consider in the statistics.

D.2 Statistics Tables

These tables have the values used in the graphics of 3.5.

Table D.1 Values used in 3.5.1

Name	Star Density	bps saved
d80.00-80.05.tm	484,41	340,79
d80.20-80.25m20.tm	462,83	260,01
d80.05-80.10m20.tm	290,51	457,99
d80.15-80.20.tm	268,87	306,03
d80.10-80.15m20.tm	82,66	203,26

Table D.2 Values used in 3.5.2 and 3.5.3.

Absolute

File Name	Density	Stars in Astro-1	Stars in Astro-2	TS A1/second	TS A2/second	NofStars/TS Astro-1	NofStars/TS Astro-2
d80.00-80.05.tm	484,41	114062	1978608	10,52	32,78	2,51	13,97
d80.20-80.25m20.tm	462,83	1918830	80600	31,56	8,46	14,07	2,20
d80.05-80.10m20.tm	290,51	967140	287579	18,37	11,12	12,19	5,99
d80.15-80.20.tm	268,87	252318	909202	9,10	16,85	6,42	12,49
d80.10-80.15m20.tm	82,66	145097	212081	4,55	5,13	7,37	9,57

Differential

File Name	Density	Stars in Astro-1	Stars in Astro-2	TS A1/second	TS A2/second	NofStars/TS Astro-1	NofStars/TS Astro-2
d80.00-80.05.tm	484,41	114062	1978608	7,62	30,99	3,47	14,78
d80.20-80.25m20.tm	462,83	1918830	80600	30,07	6,40	14,77	2,92
d80.05-80.10m20.tm	290,51	967140	287579	15,64	7,48	14,31	8,90
d80.15-80.20.tm	268,87	252318	909202	6,85	14,86	8,52	14,16
d80.10-80.15m20.tm	82,66	145097	212081	3,04	3,81	11,05	12,89

Table D.3 Values used in 3.5.4. and 3.5.5.

			Absolute Method	Differential Method
File Name	Density	File size Saved	% of Headers	% of Headers
d80.00-80.05.tm	484,4143	0,02	0,1684	0,1467
d80.20-80.25m20.tm	462,831	0,02	0,1624	0,145
d80.05-80.10m20.tm	290,5115	0,05	0,1963	0,1488
d80.15-80.20.tm	268,8704	0,03	0,1864	0,1521
d80.10-80.15m20.tm	82,661	0,07	0,2562	0,1849

Annex E Other Tests of The System

Table E.1 Files Characteristics of 3.6.

File name	Start Density	bps Saved
d91m20.tm	71,1667	271,85
d80m20.tm	74,9537	256,32
d80.065m20.tm	137,2778	590,31
d80.087m20.tm	973,7315	403,90
d91.045m20.tm	2314,3518	403,90
d120.185m20.tm	2452,4167	4512,76
d120.0.tm	6664,6665	1211,69
d120.06m20.tm	11165,7998	4721,59

Table E.2 Values used in 3.6.2 and 3.6.3

Absolute

File Name	Density	Stars in Astro-1	Stars in Astro-2	TS A-1/second	TS of Astro-2/second	NofStars/TS A1	NofStars/TS A2
d91m20.tm	71,17	4871	2815	4,88	4,63	9,24	5,63
d80m20.tm	74,95	4461	3634	4,87	4,72	8,48	7,13
d80.065m20.tm	137,28	7594	7232	9,22	9,03	7,62	7,42
d80.087m20.tm	973,73	102392	2771	66,58	17,02	14,24	1,51
d91.045m20.tm	2314,35	1727	248223	13,94	157,93	1,15	14,55
d120.185m20.tm	2452,42	237423	27438	162,75	97,75	13,51	2,60
d120.0.tm	6664,67	473274	377526	902,94	719,63	14,98	14,99
d120.06m20.tm	11165,80	13277	246510	36,34	168,52	3,38	13,54

Differential

File Name	Density	Stars in Astro-1	Stars in Astro-2	TS A-1/second	TS of Astro-2/second	NofStars/TS A1	NofStars/TS A2
d91m20.tm	71,17	4871	2815	3,48	2,26	12,95	11,54
d80m20.tm	74,95	4461	3634	3,25	2,75	12,71	12,24
d80.065m20.tm	137,28	7594	7232	5,10	4,96	13,78	13,49
d80.087m20.tm	973,73	102392	2771	63,69	14,39	14,89	1,78
d91.045m20.tm	2314,35	1727	248223	12,91	153,69	1,24	14,95
d120.185m20.tm	2452,42	237423	27438	147,06	50,52	14,95	5,03
d120.0.tm	6664,67	473274	377526	902,80	719,63	14,98	14,99
d120.06m20.tm	11165,80	13277	246510	14,58	152,62	8,43	14,96

Table E.3 Values used in 3.6.4. and 3.6.5.

			Absolut Methode	Differential Methode
File Name	Density	File Size Saved	% of Headers add	% of Headers add
d91m20.tm	71,167	0,11	0,296	0,187
d80m20.tm	74,954	0,10	0,282	0,182
d80.065m20.tm	137,278	0,13	0,274	0,148
d80.087m20.tm	973,732	0,01	0,154	0,141
d91.045m20.tm	2314,352	0,01	0,131	0,126
d120.185m20.tm	2452,417	0,06	0,202	0,144
d120.0.tm	6664,667	0,01	0,118	0,112
d120.06m20.tm	11165,800	0,01	0,129	0,116