Gus Sinnis
P-23 LANL
Milagro Memo
4/25/00

Milagro Data Integrity I: The Clock

Introduction: In Milagrito we had many problems with the recorded times of events. Some of these problems were traced to problems reading out the FASTBUS latch module. We believe that we have corrected the problem associated with the latch. In this memo I report on the quality of the Milagro data as regards the event times. I conclude that while the data is in general much freer of errors than was the Milagrito data, there still exist residual timing errors. In the Milagro data we do not see repeating events or repeating buffers of events. These errors are characterized and a plan of action is suggested.

Characterization of Clock Errors: The clock errors are categorized as follows:

- Time reversal (the time of event n+1 is precedes that of event n)
- Time gap (the time of event n+1 is > 10s + that of event n)
- Time repeat (the time difference between 2 successive events < 1μs)
- Time null (the recorded time is 0.000000)
- Event null (all event data is 0)
- Time sequence as reported from the online (should be identical to time reversals)
- Time with seconds > 86400.
- Day error (GPS clock and kahuna time differ by > 1/2 day)
- Hour error (GPS clock and kahuna time differ by > 1 hour)
- Minute error (GPS clock and kahuna time differ by > 1 minute)
- BCD error (BCD bits from clock decode to > 9)

One clock error can cause several of the above errors. For example a bad read of the clock can lead to a time gap, then the next event (supposedly a good read of the clock) will appear as a time reversal. In addition a BCD error can lead to an apparent time reversal or a gap and a subsequent reversal. Figure 1 below shows the distribution of the total number of errors per sub run. Remember, one bad event can lead to 1, 2, or 3 entries in the histogram. Figure 2 is the same data re-binned to show the distribution near zero errors. There are two features of interest in these histograms. 1) There are a large number of sub runs with a few errors. 2) There are relatively few sub runs with a large number of errors. At the moment I do not understand why the time sequence errors as encoded in the reconstructed data differs from what I infer by examining the event times.

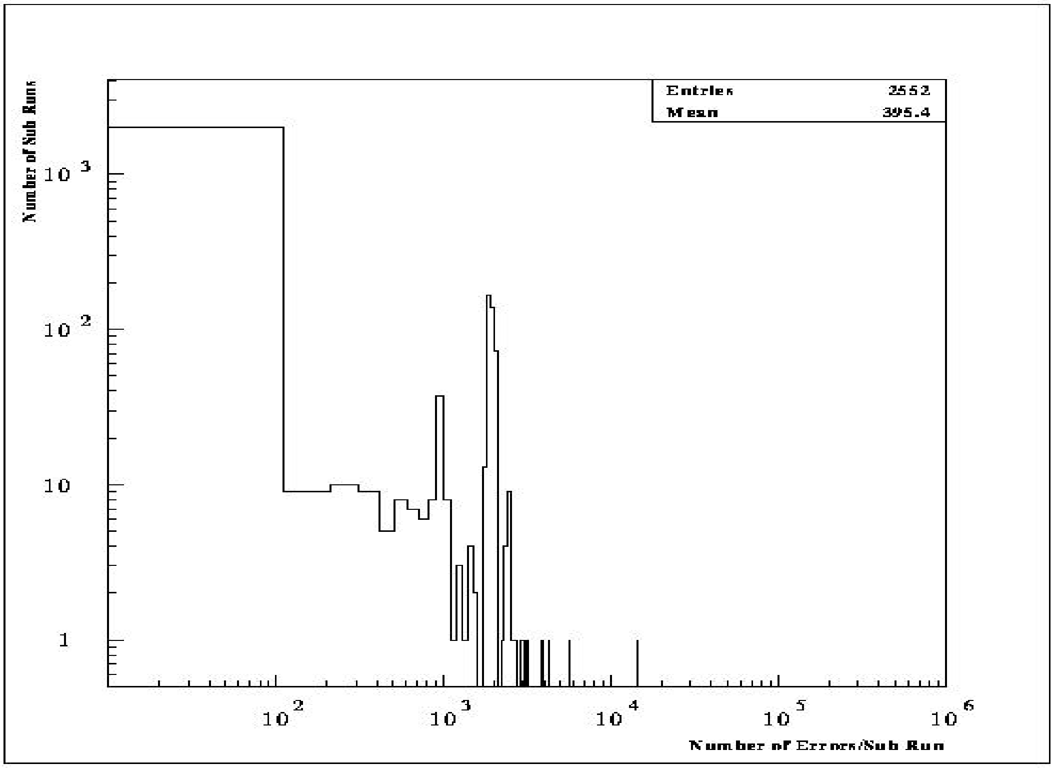

Figure 1 Number of bad events per sub run. (One entry for each sub run.)

<u>The Data and Some Statistics:</u>

For this analysis I examined all the REC files residing on disk at Maryland, these spanned runs 1150 to 2179. There were a total of 14.59 billion events in 32,431 sub runs.

The recorded errors can be broken up into three groups:

1. The data itself was crap (i.e. all information not just the time)
2. The clock on kahuna was wrong (causes GPS "errors")

3. Very rare errors reading the latch module (1/25,000,000 events)

Item 3 is rare because we read the latch 3 times and require that at least 2 reads agree. Even with the requirement we do occasionally see incorrect times being recorded. This is the cause of the time reversals, gaps, and bcd errors. Item 1 was a problem for runs 1203-1288 and is due to debugging of the online reconstruction code. These runs consist of ~1000 events each and do not constitute an appreciable amount of data. Item 2 is isolated to a few days of operation. In one case the clock on kahuna was inadvertently set to 1970 and on another occasion it was off by 2 days for unknown reasons. In addition there are instances where it drifted off by more than a few minutes.

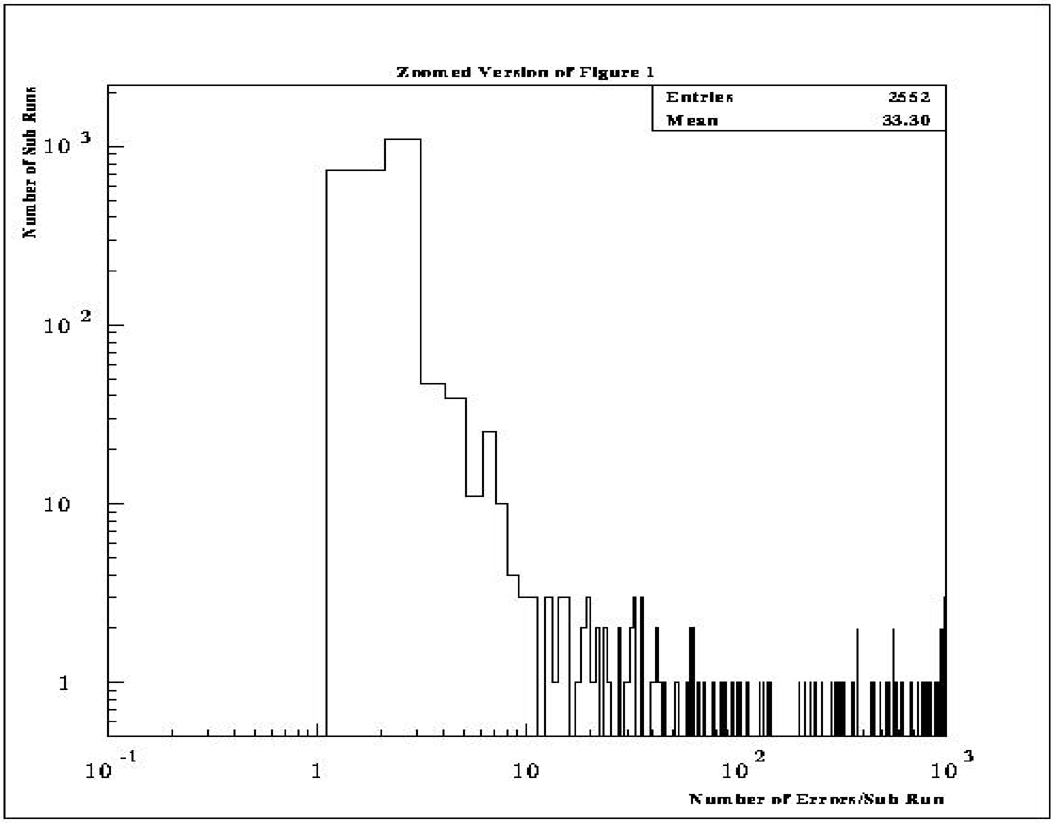

Figure 2 Zoom of Figure 1 near small number of errors.

While the total number of bad events is only 913483 ($6x10^{-5}$ of all events) in physics analyses it may well be prudent to exclude entire sub runs when the error rate is too high. Table I gives the total number of events excluded if we exclude entire sub runs based on the error count.

Table 1 Number and Fraction of Total events excluded as a function of error threshold in sub run. Where all events in the sub run are excluded if the error threshold is exceeded.

Number of errors	Number of events excluded	Fraction of all events excluded
>0	1,055,769,344	0.0723
>1	666,032,896	0.0456
>2	82,667,760	0.00566
>3	57,790,224	0.003959
>4	38,128,424	0.002612
>5	36,105,564	0.002474
>6	27,869,312	0.001909
>7	23,775,860	0.001629
>8	22,347,360	0.001531
>9	21,391,700	0.001466

Frequency of Error Types: In Figure 3 I show the frequency distribution of the various errors. The definition of the error type (the x-axis in the figure is given in the list below.

1. Time reversal
2. Time gaps
3. Time repeats
4. Time nulls (time = 0.)
5. Event nulls (all data = 0.)
6. Time sequence (as reported by data bits)
7. Seconds > 86400.
8. GPS minute off from kahuna
9. GPS hour off from kahuna
10. GPS day off from kahuna
11. GPS BCD error

Note that in this figure the number of BCD errors is incorrect as is the number of GPS day errors. There was/is a bug in the compression of the online information to form a compressed reconstructed event. While 9 bits are needed to encode the complete error information only 8 bits are allocated for this information. In the compressing of the data an overflow (bit 9 high) causes all bits to go high.

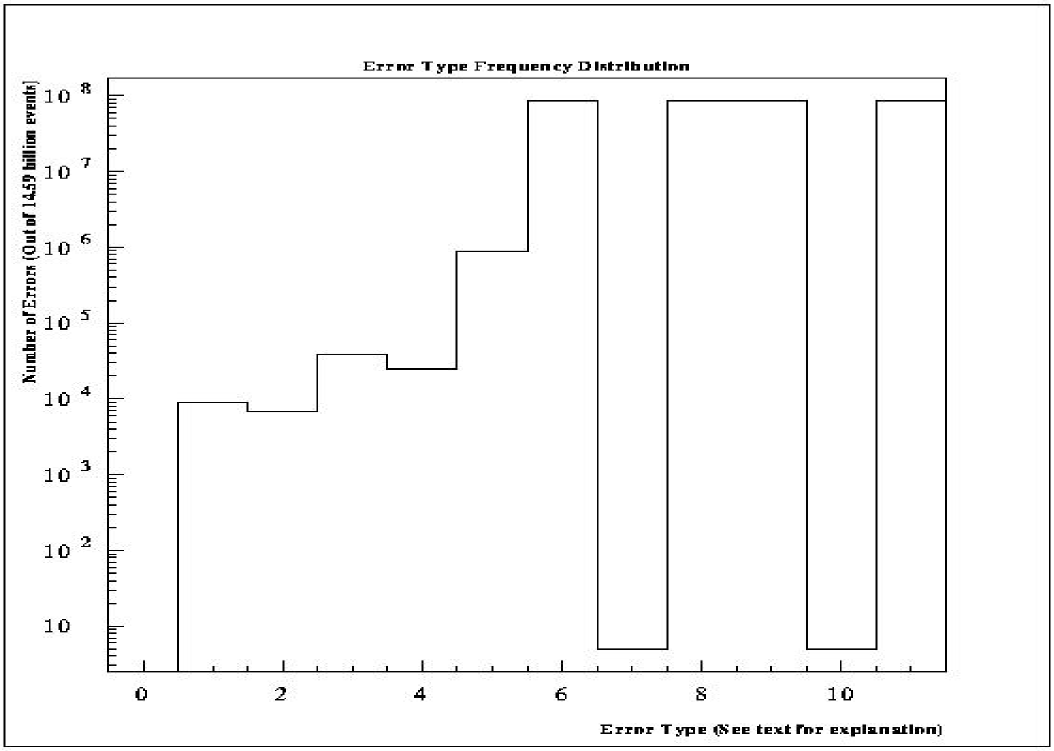

Figure 3 Number of each type of error over entire data set. Note absence of day number errors and large number of BCD errors is due to compression bug in reconstruction code.

Error History: The final plot (Figure 4) shows the number of errors/run (integrated over all the sub runs within the run) as a function of run number. Note that for the later runs (> 2000) a run corresponds to a day and contains about 150 sub runs. For the early runs (with a large error rate) there are many fewer events per run. So the obvious trend to improved data quality is actually greater than it appears to be in the plot. This plot does not include the GPS min, hr, or day errors.

Conclusions and Recommendations: After an initial burn-in period the data quality in Milagro is quite good. Because of the large number of errors in the Milagrito data, we established a complex and imperfect algorithm for retaining events that were surrounded by bad data. The imperfection of the algorithm used is due to the inherent difficulty in determining which events had the correct time assigned to them given a string of inconsistent times. I propose that for Milagro we take the following approach.

- As part of the data integrity check performed at Maryland we remove entire sub runs with more than 3 errors. These sub runs may be placed in a separate directory for possible use in a triggered GRB search. But it should be understood that they are somewhat suspect.
- For the remaining runs we use a simplified version of a time checking routine that simply removes events whose times are not monotonically increasing. While this will usually mean that we remove 2 events for each error, given the paucity of errors this will have a negligible impact on any analysis.

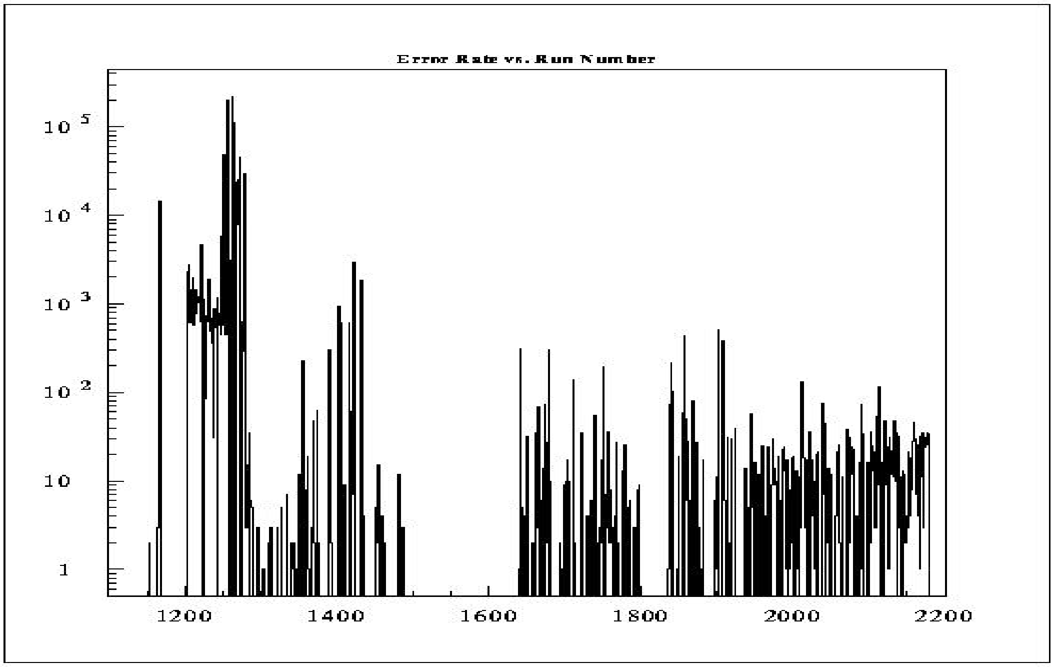

Figure 4 Number of errors/run (integrated over all sub runs) as a function of run number.