A technical report

Computer clocks at GMRT - A

12 June 2003

N.G. Kantharia, A. Adoni, A.P. Rao

1 Background

Time-keeping is an important activity of an observatory. Most observatories invest in GPS and hydrogen masers to maintain their time standards. At GMRT, a GPS receiver is used for maintaining an accurate time standard. A Pentium personal computer operating on Linux, *dual1* is used to determine a time equation. Since delays, fringe stopping, antenna pointing etc. depend on the time, accurate timekeeping is necessary. At GMRT, the data acquisition system, delay-correction system and correlator time-keeping is done by *dual1*. However, the antenna control system gets its time standard from a sun work station, presently *bhaskar*.

At GMRT, a time equation is determined on the data acquisition computer, *dual1* using the GPS 1 PPM (pulse per minute) signal. The equation maintains time on duall to a high accuracy of $100\mu s$ (Rajiv Singh 1999). Since the equation is determined by using the GPS 1 PPM without comparision with the actual time, the *dual1* time can drift in time. For example, if *dual1* time was 12h15m15s when the GPS sends out a minute pulse, then the actual time will be 12h15m0s and the server time is updated to that. This will fail only if the *dual1* time has drifted by more than 30s in which case an error of 1m might be introduced. However, this is not likely to happen. The data acquisition software 'acq' (Rajiv Singh) has an in-built time-keeping. A client-server pair 'tcli-tsr' is used to update the time on other computers. The time server 'tsrv' is run on *dual1* and 'tcli' on the client computer. The time offset on other computers is compared to the *dual1* time and the offset printed when the client software 'tcli' is run on the client computer. The time on the client can be corrected by root permissions by running 'tcli-s'. Thus, the server time on *dual1* which is in tune with the GPS time should be used regularly to update the times on other machines at the observatory to maintain accurate time-keeping on all computers.

In March 2003, we noticed a fast rate of change in the time of the new control machine '*bhaskar*' in addition to a 2 sec jump every few minutes. To study this, we undertook a couple of experiments and studied the behaviour of the clocks of a few computers at the observatory.

2 Experiment 1

The time offsets on the local clocks of computers *bhaskar* (control machine), *aditya* (backup control machine), *mithuna* (data recording machine), *dhanishtha* (old control machine) and *chitra* (old backup control machine) as compared to *dual1* were recorded at a regular interval. These were obtained by executing 'tcli' on these computers and noting down the difference in time between the server on *dual1* and the local computer clock. A short script was written to set up a cron job of running tcli every minute on all the aforementioned machines. The recorded offsets were written into an output file. This script was run simultaneously on the above machines.

2.1 Results

2.1.1 Sun work stations

The above experiment demonstrated that clocks on different computers behave differently. Clocks on sun work stations jump by 2 seconds at regular intervals (Fig. 1 and Fig. 2). It is noticed that when the computer clock drifts by more than 2 seconds within some time frame, those 2 seconds are corrected for by an internal mechanism. This ensures that the clock does not drift too fast and that time accuracy is maintained to $\sim$ 2 seconds for a significant length of time. This drift of 2 seconds on the old sun (*dhanishtha* and *chitra*) clocks is observed to be much slower and hence the correction is infrequent. The clocks are more stable and reliable. On *dhanishtha*, the clocks drift by 2 seconds every 8 hours or so whereas on *chitra*, the 2-second correction was corrected once every 6 hours. In contrast, the new sun machines (*aditya* and *bhaskar*) drift by 2 seconds within 25 minutes! Thus a 2 second jump is observed every 25 minutes or so. Please note that the 2 second correction is not effected instantaneously. The rate at which the clocks on *aditya* and *bhaskar* drift is $\sim$ 0.1 sec/min. Once this drift has added up to 2 seconds, the clock is corrected at a rate of $\sim$ 0.4 sec/min so that in two minutes, a 2 second jump is recorded. While a slow drift which is seen on *dhanishtha* and *chitra* and corrected by the system is beneficial, the fast variation as recorded by *aditya* and *bhaskar* and corrected by the system should be investigated.

Superposed on the slow 2-second drift, which is regularly corrected by the sun systems, there is a slow residual drift of the clock. Fig 2 clearly shows that the *chitra* system corrects for the 2 sec drift in its clock within $\sim$ 6 hours but there is a $>$ 2 sec drift over 1600 minutes which is not corrected by the machine and needs external intervention. A slow residual drift of $<$ 1 sec over $\sim$ 1600 minutes is also noticed in *dhanishtha*. Due to the regular 2-second correction,

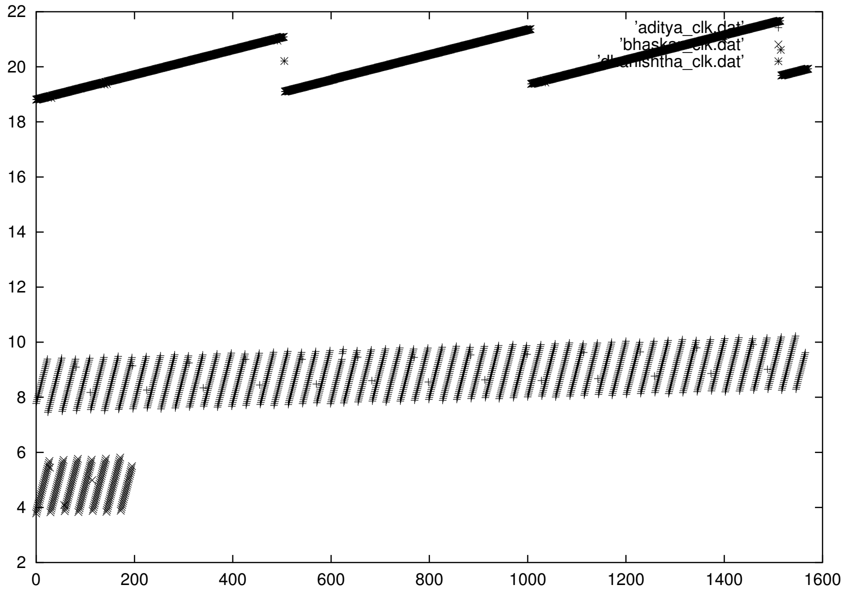

Figure 1: This figure shows the clock offsets for the sun work stations at GMRT: *dhanishtha*, *bhaskar* and *aditya*. The topmost curve is for *dhanishtha*, middle one is for *aditya* and the lowermost curve is for *bhaskar*. The slope shows the drift whereas the intercept on the y-axis shows the existing offset when we started the experiment. X-axis is labelled in minutes whereas the y-axis is in seconds of time.

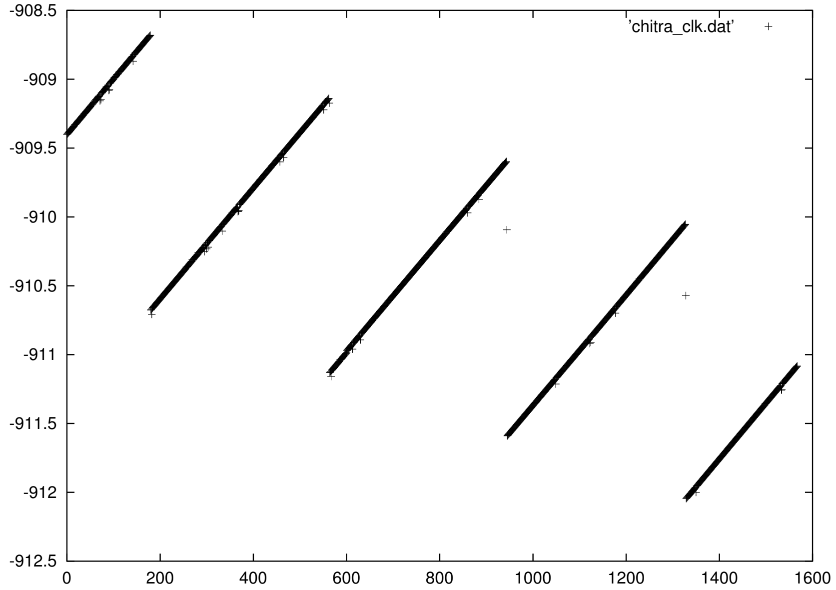

Figure 2: This figure shows the clock offset for the sun computer *chitra*. The intercept on the y-axis is ~ -910 sec which indicates the clock offset when we started the experiment. X-axis is labelled in minutes whereas the y-axis is in seconds of time.

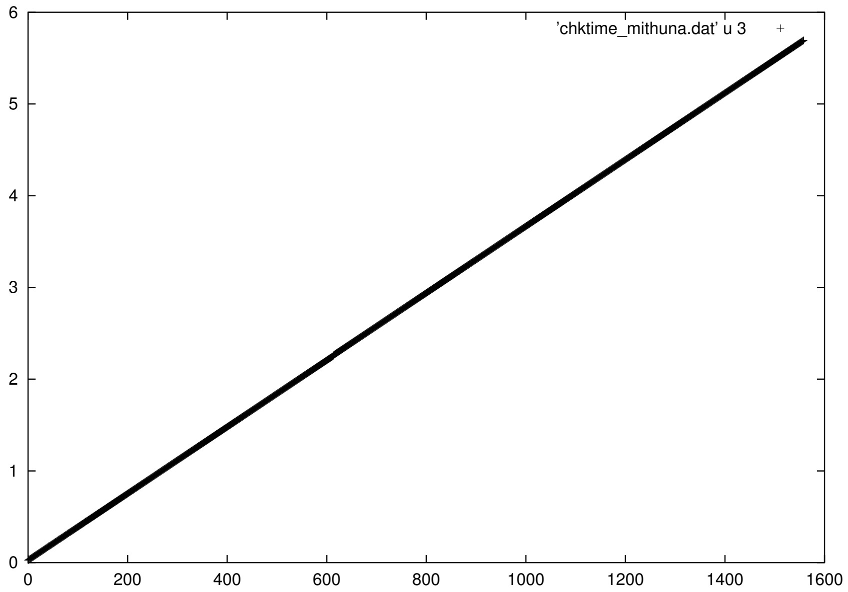

Figure 3: This figure shows the clock offset for a Linux-running personal computer, *mithuna*. We corrected the existing clock offset before starting the experiment. The clock shows a drift of $\sim$ 6 sec in 24 hours. X-axis is labelled in minutes whereas the y-axis is in seconds of time.

the residual drift is reduced making the clock more stable over a longer time scale. In case of *bhaskar* and *aditya*, the residual time drift is also < 1 sec over $\sim$ 1600 min. Although over a long time scale, *bhaskar* and *aditya* are stable clocks, the jumps of 2 seconds recorded every 25 minutes or so should be understood and if possible a clock behaviour similar to the *dhanishtha* clock established.

2.1.2 Pentium Personal Computers

The Pentium personal computers operating with Linux OS exhibit a different clock behaviour. In case of *mithuna* (interferometric upper sideband data recording computer), there is a linear drift of $\sim$ 6 sec over 1600 min as shown in Fig. 3. There is no internal system correction for this clock and all the corrections have to be effected from external time-servers. A linear drift of $\sim$ 0.25 sec over $\sim$ 5 hours is recorded for mithunb (interferometric lower sideband data recording computer). We also ran 'tcli' on *dual1* and recorded the offsets. As expected the duall time is stable within 0.01 sec over 5 hours.

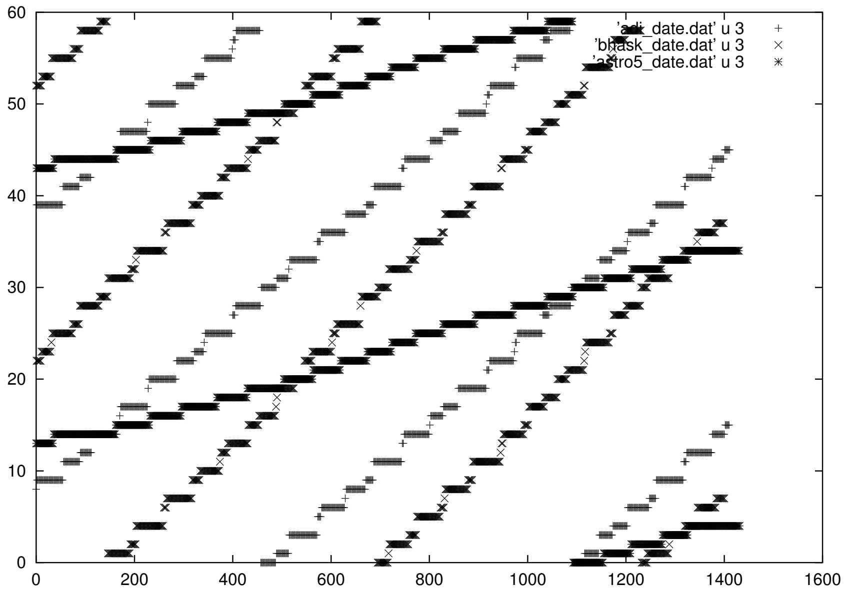

Figure 4: This figure shows the output from the system command 'date' run every 30m on the sun work stations, *bhaskar* & *aditya* and on the linux personal computer, *astro5*. The '+' symbol represents the *aditya* time, 'x' shows *bhaskar* time and '*' shows the variation in *astro5* time. X-axis is labelled in record number whereas the y-axis is in seconds of time.

3 Experiment 2

Since the time offsets discussed above are recorded with the server running on a Pentium computer, we also conducted another experiment to record the system time independent of the server. We recorded the time displayed by running the system command 'date' every 30 seconds. Here an external server was eliminated and only the system clock was monitored. This was done to verify that the linux server does not show any peculiar problem with monitoring the sun work station time. We recorded the time on *dhanishtha*, *chitra*, *aditya*, *bhaskar*, *astro5*, *mithuna* mithun over $\sim$ 13 hours. The results are shown in Fig 4.

The clock of the sun work stations i.e.*dhanishtha*, *chitra*, *aditya*, *bhaskar* clearly showed 2 second jumps as noticed in Experiment 1 (see Fig 4). Thus similar behaviour is exhibited by the sun work stations and the time server does not have any peculiar behaviour on determining the time offsets for the sun work stations.

4 Discussion

The sun work stations and Pentium computers employ different time-keeping methods. It is not clear which is the better of the two. In case of the sun work stations, the clock does not show a large drift over 24 hours and is within 2 seconds for all the computers studied here, while in case of *mithuna* (Pentium), time drifts by $\sim$ 6 seconds within 24 hours. However, time drifts by 2 seconds at a faster rate on *aditya* and *bhaskar* and is frequently corrected by the system whereas the drift is slow for *dhanishtha* and *chitra*. All the sun work stations require infrequent time correction using an external time server whereas the Pentium computers require a correction from an external time server frequently.

In addition to above, we would like to point out that many clocks in the observatory show offsets ranging from many seconds to many minutes at any given time. For example, one of us, has twice noticed that the *mithuna* time has drifted by 20m from the *dual1* time. Although all the computers are not crucial for antenna control or data acquisition, an observatory should not have such widely deviant times, and we should evolve a method of correcting the clocks frequently. Since the 'tcli-tsrv' software is available, it should be made a daily/weekly cron job of correcting the time on all machines. This can be done either between two observing runs by the telescope operator on a daily basis or on one of the maintenance days as a cron job where the operator intervention is removed.

4.1 Practical Implications of Clock errors

We briefly describe the implications of time offsets on (1) the control system and subsequently on antenna pointing/positioning (2) the data acquisition system on observations.

4.1.1 Control system

An error in the clock of the control computer *bhaskar/aditya* will directly translate to a pointing error for the antennas. For a time offset t sec, the pointing error for a source at declination δ will be (t * 15)" cos(δ). The worst case will be for a source near the equator (δ=0^o). The following table gives a quick look at the pointing offsets produced by seemingly 'insignificant' time offsets for a source near the equator.

Since the sun work stations frequently show clcok offsets of 2 seconds, pointing offsets can be accurate only within 30" at any given time. Since the desired pointing accuracy so that the intensity fluctuations within the beam due to to pointing errors are not significant, is $\Delta\theta < HPBW(1420MHz)/20$ (See ASP Conference Series 6). For GMRT antennas, $\Delta\theta = 24'/20 = 1.2'$. The 2 second time offset will produce an uncertainty which is just within the required accuracy for the low frequencies that GMRT operates at. At lower frequencies, the effect of the 2 second time offset will be less.

Clock offset (seconds)	pointing offset (arcmins)
2	0.5
6	1.5
10	2.5
20	5

However, in addition to the fast drift noticed in the control machines, there is a slow drift which is about a second in 24 hours. If this is not corrected regularly then the pointing offsets will change with time as the clock drifts. These errors cannot be corrected by pointing models and will translate into phase errors in the data which the user might be able to remove by using self-calibration techniques. However, if the clock drifts by a few seconds without being corrected, this would lead to dynamic range limitations in the image. The intensity fluctuations at the half power points at 1420 MHz due a $\sim 1.2'$ pointing error would be $\sim 13\%$. With increasing the pointing offsets, the fluctuations at the half power points will increase leading to poor dynamic range effects, especially for extended sources. Hence, it is advisable to correct the time once a day between observations and can even be set up as a cron job. This way the time offset is always maintained at about 2 seconds and pointing offsets to less than 30" which is sufficient for the low frequencies at which GMRT operates till the 2 second fast drift can be understood and resolved.

4.1.2 Correlator and Data acquisition system

The time standard on *dual1* computer which is used for data acquisition and time-keeping for the correlator is maintained to an accuracy of $100\mu sec$ (Rajiv Singh 2003) by using the GPS. Thus the time-keeping for the correlator system required for delay corrections and fringe-stopping is maintained to a high accuracy at GMRT and is not an issue.

However, for completion we would like to mention that if there was a clock error in computing the instrumental delays leading to incorrect instrumental delays being applied to the data, then this would translate to a phase gradient across the observed band. Anyway, as noted by Chengalur and Kudale (2002), this is not the case for the GMRT antennas which show changes in instrumental delays between the two polarizations, between different IF filters and between different feeds due to reasons other than time offsets.

4.1.3 Difference in control system time and data acquisition system time

Frequently, we find that this time difference has drifted to 6-8 seconds. This means that the antennas are pointing in a direction which is 90-120" (for a source at the equator) offset from the direction for which delay correction and fringe stopping are getting implemented. This would certainly lead to phase errors for the L band. Since an error of 2 sec between the two times seems inavoidable with the present control computers, it is essential that the difference is not allowed to become larger than this number. Again this translates to an urgent need to update the times on the observatory machines at least once a day between two observing runs or set up an hourly cron job.

5 Progress Report

Since the above experiments were done, progress has been achieved in fixing the clock drift problem on the sun work stations, *bhaskar* and *aditya*. We could contact the Sun Microsystems user help and have applied a cluster of patches to correct the fast drift in the local clocks and the subsequent frequent 2 sec jumps. This has been implemented on *bhaskar* and it has been found to function satisfactorily as shown in Fig 5. The fast drift has been replaced by a slow drift of less than one second over 24 hours. More details of the procedure will be given in the second part (B) of this report by A. Adoni.

6 Summary

In this report, we have described a couple of experiments we conducted to study the behaviour of the system clocks on sun work stations which are used as antenna control computers and the pentium computers which are used for data acquisition and recording. The old control machines, *dhanishtha* and *chitra* had a fairly stable clock registering a drift of 2 seconds over 6-8 hours which are internally corrected by the system. However the new control machines, *bhaskar* and *aditya* register a drift of 2 seconds within half an hour and which

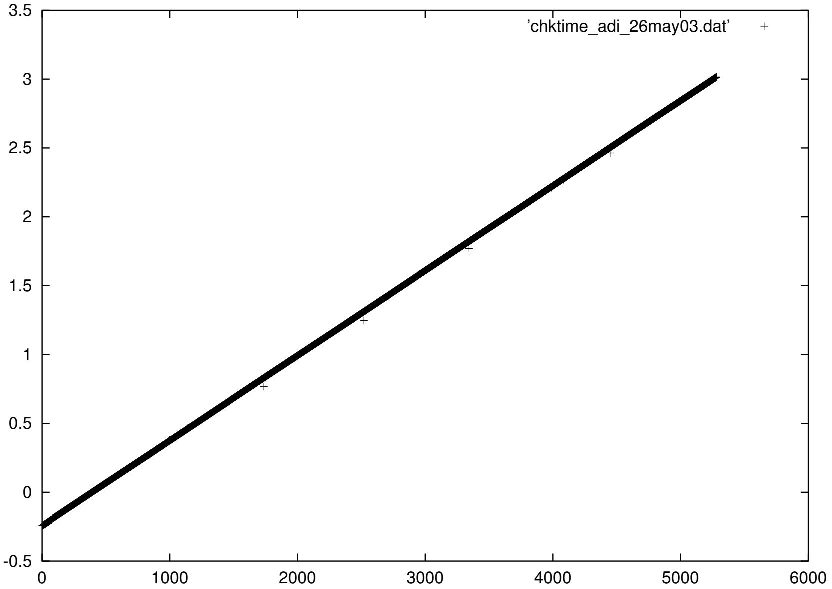

Figure 5: This figure shows the clock offsets for the sun work station, *aditya* at GMRT after applying a cluster of patches. The 2 second drifts and jumps which were visible in 1 are not present and a linear drift of about 2 seconds over 5000 minutes is only seen. The clock has become fairly stable. The slope shows the drift whereas the intercept on the y-axis shows the existing offset when we started the experiment. X-axis is labelled in minutes whereas the y-axis is in seconds of time.

is internally corrected by the system. In other words, a 2 second jump in the clocks is registered every half an hour or so, which is not desirable. In addition to this, the sun work station clcoks show a slow drift. On the other hand the Pentium machines running Linux OS show a linear drift in the clock with no internal system correction. We measured the drift for one of the machines used for data recording, *mithuna* and found a drift of about 6 seconds in 24 hours. Although it does not show 2 second jumps, the drift is fairly significant and requires to be corrected once in 24 hours. The data acquisition computer, *dual1* has got GPS synchronised time. We suggest that rest of the computers at the observatory are synchronised to *dual1* time using the server-client software *tsrv-tcli* developed by Rajiv Singh.

As follow-up action, we contacted the Sun Microsystems user help and have implemented a cluster of patches for the new control machines to slow down the fast drift and also rid them of the 2 second jumps. After the application of these patches, the drift in the *bhaskar* clock has been reduced to less than a second in 24 hours.

7 References

Rajiv Singh, 1999 in Radio Astronomy at Low Frequencies, SERC School held at NCRA in 1999.

Jayaram N. Chengalur, Sanjay Kudale in Technical Report in 2002, 'Instrumental Delays at GMRT'.

NRAO Summer School on Synthesis Imaging in Radio Astronomy, ASP Conference Series 6.