



IAR Memo *xxx*:

05 de Mayo de 2004 (v. 0.1)

Driver para Network Time Protocol del GPS Trimble Thunderbolt Disciplined Clock

Fernando Pablo Hauscarriaga¹ fernandoph@iar.unlp.edu.ar

1: Instituto Argentino de Radioastronomía (IAR).

PDF: http://www.iar.unlp.edu.ar/~fernandoph/pub/docs/thunderbolt_driver.pdf
PostScript: http://www.iar.unlp.edu.ar/~fernandoph/pub/docs/thunderbolt_driver.ps

Índice

1. Introducción	1
section.2.1 <i>Network Time Protocol: Una breve introducción</i>	2
section.3.2 Instalación y configuración de la máquina Server NTP	3
section.4.3 Trabajos sobre el código obtenido	3
section.4	
4.1. Introducción	3
subsection.4.1	
4.2. Breve descripción de las funciones de <code>refclock_palisade.c</code>	4
subsection.4.2	
4.3. Modificaciones en las Funciones	4
subsection.4.3	
4.3.1. <code>palisade_start</code>	4
subsubsection.4.3.1	
4.3.2. <code>TSIP_decode</code>	5
subsubsection.4.3.2 Resultados	6
section.5	
5.1. Gráficos de <i>offset</i>	6
subsection.5.1	
5.2. Problemas de sincronismo	7
subsection.5.2 Conclusión	8
section.7.6 Propuesta	8
section.7	

1. Introducción

El sistema de sincronización de computadoras que vamos a detallar, se constituye de dos elementos claves: el primero es el Reloj de Referencia, con el cual llevaremos a cabo la sincronización; el segundo es el protocolo que usaremos para que todos los equipos de la red puedan sincronizarse con el Reloj de Referencia.

La sincronización de los equipos en una red como la que posee el Instituto juega un papel importante, ya que muchos de los cálculos que se realizan para llevar a cabo las observaciones implican operaciones con factores de tiempo (tanto medios como sidéreos), por eso es menester poder tener una red capaz de mantenerse en sincronización sin ninguno otro medio que por sí misma, sin tener que depender de servidores de tiempo externos, por supuesto que estos servidores se utilizaran también, tomándolos como referencia para detectar con más precisión el *offset* de la red del **IAR**; pero lo realmente importante es que la red, aún en el supuesto caso de que falle el enlace con Internet jamás pierda sincronismo.

Utilizaremos como Reloj de Referencia, el **Thunderbolt GPS Disciplined Clock**, un modelo de la firma **TRIMBLE**. Para la sincronización de los equipos en la red, utilizaremos un protocolo llamado **Network Time Protocol (NTP)** (RFC-1305) desarrollado por **David L. Mills**, que posee licencia GPL.

En la Figura 1, se puede ver un diagrama general del sistema, describiendo a grandes rasgos cada una de las partes. Mas adelante, conforme se valla desarrollando cada etapa, se ira detallando con mayor exactitud el funcionamiento y desarrollo de todos los componentes del sistema.

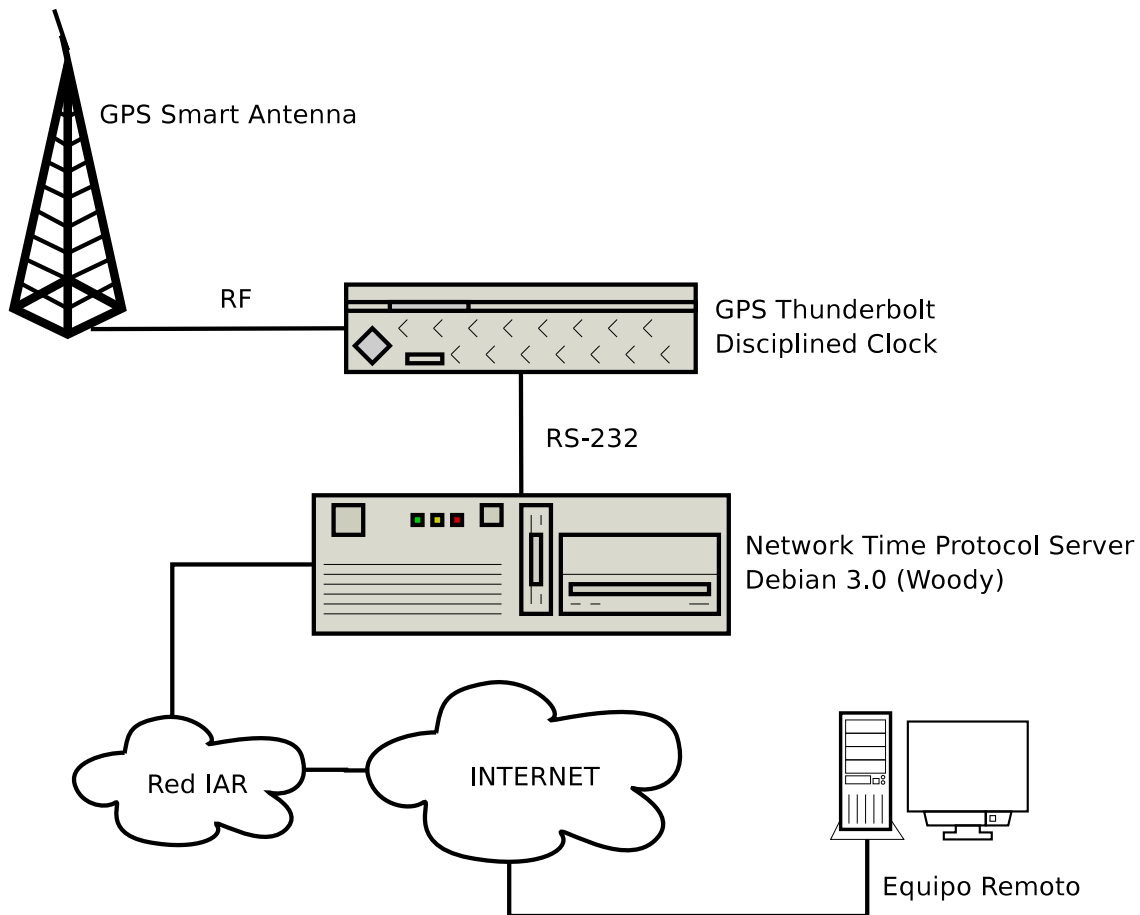


Figura 1: Diagrama general de sistema de sincronización de PCs mediante NTP

2. Network Time Protocol: Una breve introducción

Network Time Protocol (NTP), ahora establecido como un protocolo *standard* de *internet*, se usa para organizar y mantener un conjunto de clientes, servidores de tiempo y caminos de transmisión como una red de sincronización. *NTP* está construido sobre las bases del protocolo de *internet* (IP) y el protocolo UDP (*User Datagram Protocol*), los cuales proveen un mecanismo de transporte libre de conexiones; sin embargo, es también adaptable a otros protocolos. *NTP* está específicamente diseñado para mantener la confiabilidad y estabilidad en rutas de *internet* típicos que involucran varios *gateways* (o *routers*) y redes poco confiables.

En *NTP* uno o más servidores primarios se sincronizan directamente a fuentes de referencia externas (por ej: GPS). Los servidores de tiempo secundarios se sincronizan con los servidores de tiempo primarios y con otros servidores en la red. Una topología de red típica se muestra en la figura 2a, en la cual, los nodos representan servidores de la red, con números de *stratum* en cuenta regresiva hacia la raíz (*stratum 1*: servidor de tiempo sincronizado con un reloj de referencia externo). Las flechas indican los caminos de sincronización activos y la dirección del flujo de la información de tiempo. Las líneas representan caminos de sincronización de respaldo donde si bien se intercambia información de tiempo, no necesariamente se usa para sincronizar los relojes locales. La figura 2b muestra la misma red, pero con la línea marcada “x” (en la figura 2a) fuera de servicio. La red se ha reconfigurado automáticamente para usar los caminos de respaldo, con el resultado de que uno de estos servidores ha pasado de *stratum 2* a *stratum 3*.

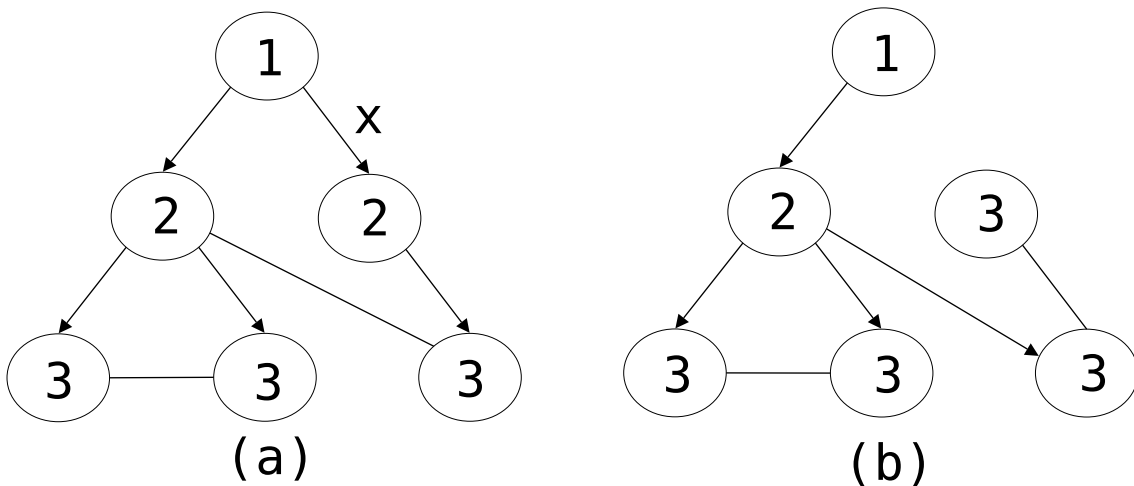


Figura 2: Red de Sincronización

3. Instalación y configuración de la máquina Server NTP

Se armó el servidor de tiempo dentro de un gabinete de PC rackeable.

Se instaló en la maquina servidor, un sistema Debian GNU/Linux Woody 3.0. La versión del Kernel es 2.4.18-bf2.4.

Se le asignó una IP: 163.10.43.42 y un nombre de host: gps.iar.unlp.edu.ar



Figura 3: Servidor de NTP junto al Trimble Thunderbolt GPS Disciplined Clock

4. Trabajos sobre el código obtenido

4.1. Introducción

Al momento de escribir este informe, la última versión estable de *NTP* es 4.2, así es que los trabajos se realizaron sobre esta versión del software.

El software provee soporte para una variedad importante de relojes de referencia externos. Lamentablemente el *GPS* del **IAR** no encajaba en esta categoría de afortunados, por lo cual se decidió agregar el soporte para el equipo.

Cuando estábamos estudiando la manera de agregar el *driver* al *set* ya existente, nos topamos con el *driver* del modelo **Palisade** de la firma **Trimble**, con lo cual el siguiente paso fue obtener la documentación del mismo.

Con la documentación ya en la mano, vimos que el modelo **Palisade** era un equipo concebido especialmente para la utilización en bases de tiempo de precisión y fue pensado desde el momento

de diseño para ser usado en servidores de tiempo *NTP*, de ahí que el driver que viene en el paquete *NTP* está originalmente escrito por la firma del fabricante.

Vimos también que la estructura de los paquetes, tanto del **Thunderbolt** como del **Palisade**, era exactamente la misma, por lo tanto, la decodificación de los paquetes entrantes estaba totalmente resuelta y lo único que quedaba por resolver era pasarle al NTP los datos correctos que necesitaba para poder sincronizarse con el *GPS* y así usarlo como reloj de referencia externo.

Ahora, es importante enfatizar el hecho de que una de las grandes diferencias entre los dos modelos, reside en que el modelo **Palisade** cuenta con la posibilidad de entregarle al usuario fracciones de tiempo menores al segundo, mientras que el **Thunderbolt** no posee esta característica.

Los archivos de código fuente del *driver* para el modelo **Palisade** son los siguientes:

refclock_palisade.c
refclock_palisade.h

Ambos se encuentran en el directorio `/ntpd` del código fuente.

Estos archivos son los que se modificaron para que el nuevo *driver* del modelo **Palisade** suportara además al modelo **Thunderbolt**.

Puede encontrar el patch en:

http://www.iar.unlp.edu.ar/fernandoph/pub/projects/thunderbolt_ntp/refclock_palisade.patch.tar.gzhttp://www.iar.unlp.edu.ar/fernandoph/pub/projects/thunderbolt_ntp/refclock_palisade.patch.tar.gz

4.2. Breve descripción de las funciones de `refclock_palisade.c`

- `palisade_start` - Abre los dispositivos e inicializa los datos para ser procesados.
- `palisade_shutdown` - Apaga el Reloj de Referencia.
- `day_of_year` - Obtiene el día, el mes y el año del número de días transcurridos en el año.
- `TSIP_decode` - Decodifica los paquetes de datos.
- `palisade_receive` - Recibe los datos de la interfaz serie.
- `palisade_poll` - Es llamada por el procedimiento `transmit` y detecta si el dispositivo contesta o no a los pedidos de datos del protocolo.
- `palisade_io` - Inicializa punteros, lee el `TIMECODE` y genera un `TIMESTAMP` del reloj interno de la PC, también se encarga de armar el paquete de tiempo.
- `HW_poll` - Dispara la entrada de eventos del Dispositivo, toma un `TIMESTAMP` del reloj de la PC para igualarlo al del *GPS*.

4.3. Modificaciones en las Funciones

A continuación se expondrá sin entrar en detalles de programación estrictos, las modificaciones que se hicieron sobre las funciones del *driver* existente.

4.3.1. `palisade_start`

Se agregó en esta función la posibilidad de que el *driver* pueda detectar que tipo de *GPS* tiene conectado, y configurar en consecuencia la interfaz serie, ya que las formas de comunicación a través del puerto serie difieren entre los dos modelos de *GPS*.

4.3.2. TSIP_decode

En esta función se agregó la decodificación de los paquetes del *GPS* Thunderbolt, el *0x8FAB* (Paquete de tiempo), el *0x8FAC* (Paquete de Posición) y el *0x6D* (*status* del *tracking*).

El *GPS* envía los paquetes *0x8FAB* y *0x6D* junto con el *PPS* (pulso por segundo). El paquete *0x8FAB* contiene información detallada del tiempo del *GPS* y el paquete *0x6D* contiene información sobre el estado general del *tracking* de los satélites. Luego, cada 6 (seis) *PPS* o si se detectan cambios, el *GPS* envía el paquete de posición *0x8FAC* que contiene información detallada sobre la posición, así como también sobre *leap second*.

A continuación se muestran los paquetes que se usaron para llevar a cabo la sincronización; sólo se muestran aquellos datos de cada paquete que realmente son utilizados por el protocolo *NTP*, para una descripción completa de estos y todos los paquetes disponibles en el protocolo de comunicación del *GPS*, ver el manual del usuario del mismo.

Paquete 0x8FAB

<i>Byte</i>	<i>Item</i>
1-4	<i>GPS Seconds of Week (TOW)</i>
5-6	<i>UTC Offset</i>
10	Segundos (BYTE)
11	Minutos (BYTE)
12	Horas (BYTE)
13	Día del Mes (BYTE)
14	Mes (BYTE)
15-16	Año(INTEGER)

Paquete 0x8FAC

<i>Byte</i>	<i>Item</i>
36-43	Latitud (DOUBLE) en Radianes.
44-51	Longitud (DOUBLE) en Radianes.
52-59	Altura (DOUBLE) en Metros.

Paquete 0x6D

<i>Bit en Byte 0</i>	<i>Item</i>
0:2	<i>Fix Dimension</i> (2D o 3D)
3	<i>Fix Mode</i> (auto o manual)
4:7	Número de satélites usados para el ajuste

5. Resultados

5.1. Gráficos de *offset*

En los siguientes gráficos se representa la variación del *offset* conforme el paso del tiempo. La escala de las abscisas representa el tiempo en horas, y el eje de las ordenadas representa el *offset* en segundos entre el reloj interno de la computadora y el *GPS*; estos datos fueron obtenidos de los archivos de log que genera automáticamente *NTP*.

Durante los primeros días de funcionamiento, el patch estuvo sometido a varias sesiones de *debug*. Como consecuencia del continuo arrancar - detener del servidor de tiempo, la calidad del sincronismo fue bastante pobre, e incluso nula en algunos momentos. Como ejemplo de esta situación, podemos ver la Figura 4, donde se ve el gráfico de *offset* del primer día de funcionamiento del driver. En este, se puede ver una brecha de aproximadamente 2 horas (entre las 9 y 11 horas) en el que el sincronismo se pierde alrededor de las 9 horas y lentamente empieza a recuperarse alrededor de las 11 horas, estando más estable alrededor de las 13 horas.

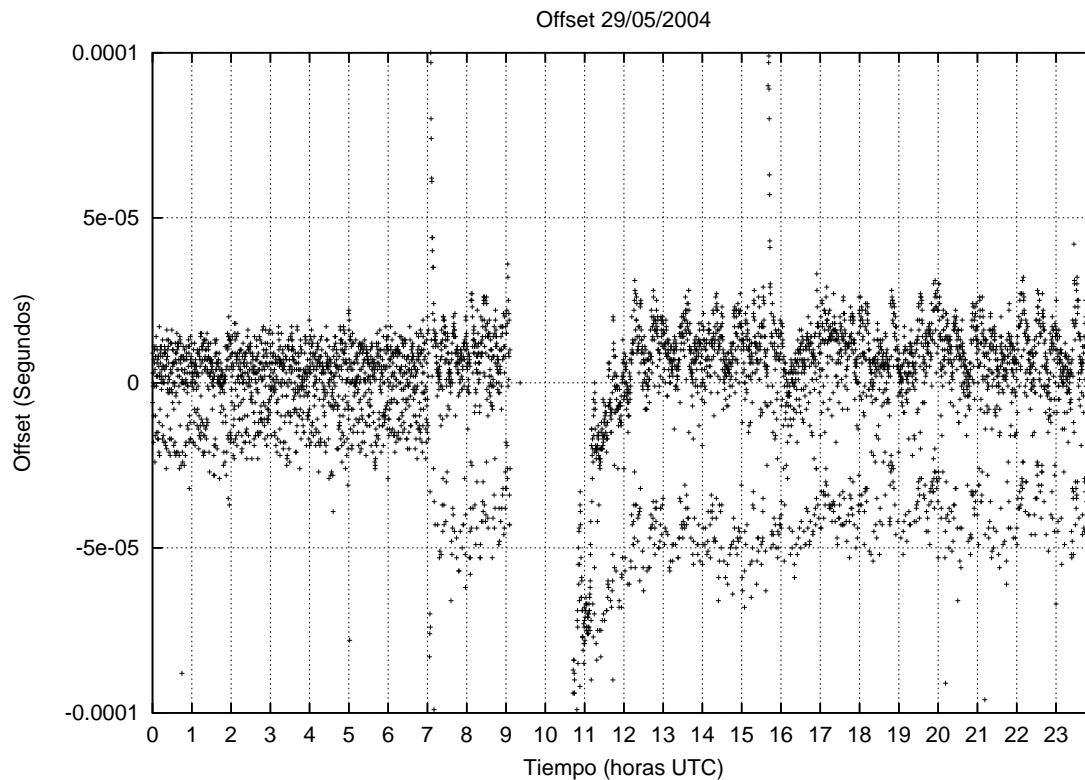


Figura 4: El primer gráfico, con cortes debido a *tests* y al continuo arrancar - detener del programa durante los primeros *testeos*.

La capacidad de *NTP* de mantener un *offset* de tiempo entre dos relojes alrededor de 0 depende casi exclusivamente del tiempo de *uptime* que éste tenga. En la Figura 5, puede verse nuevamente un gráfico de *offset*, donde la estabilidad del tiempo es mucho mas notable.

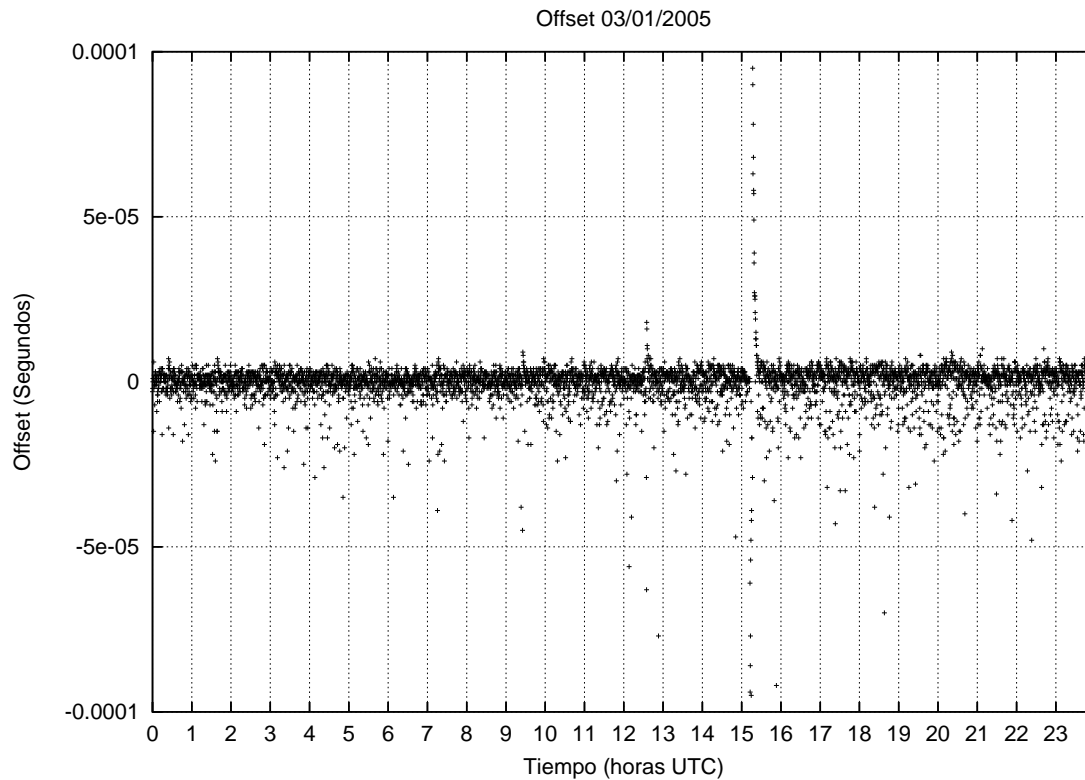


Figura 5: Datos de Offset

5.2. Problemas de sincronismo

Finalmente nos hemos encontrado con un problema de sincronismo persistente entre el *GPS* y *NTP*. Cada final de semana (Domingo a las 00:00 hrs. UTC, semana *GPS*), el *GPS* pierde sincronismo, para luego recuperarlo sin otro medio que por sí mismo, 18 horas después.

Esta falla se hizo evidente a través del archivo de *log* que automáticamente genera el demonio *NTPD*. A continuación vemos un fragmento en el cual se presenta la falla el día Sábado 6 de Noviembre, donde aproximadamente un minuto y medio después de las 21 (UTC -3), tiempo que tarda *NTPD* en declarar muerta una fuente, la leyenda “*No servers reachable*” nos indica que ha perdido sincronismo con el *GPS*, recobrándolo al día siguiente (7 de Noviembre) a las 15:12:45 (UTC -3).

```

3 Nov 09:45:25 ntpd[4073]: frequency initialized -0.002 PPM from /etc/ntp.drift
3 Nov 09:46:12 ntpd[4073]: synchronized to GPS_PALISADE(0), stratum=0
3 Nov 09:46:11 ntpd[4073]: time reset -1.584153 s
3 Nov 09:46:11 ntpd[4073]: kernel time sync disabled 0041
3 Nov 09:47:14 ntpd[4073]: synchronized to GPS_PALISADE(0), stratum=0
3 Nov 09:47:49 ntpd[4073]: kernel time sync enabled 0001
3 Nov 22:49:01 ntpd[4073]: time reset +1.000025 s
3 Nov 22:50:07 ntpd[4073]: synchronized to GPS_PALISADE(0), stratum=0
5 Nov 06:09:11 ntpd[4073]: time reset +1.000024 s
5 Nov 06:10:17 ntpd[4073]: synchronized to GPS_PALISADE(0), stratum=0
6 Nov 13:09:35 ntpd[4073]: time reset +1.000015 s
6 Nov 13:10:43 ntpd[4073]: synchronized to GPS_PALISADE(0), stratum=0

```

```
6 Nov 21:01:29 ntpd[4073]: no servers reachable
7 Nov 15:12:45 ntpd[4073]: synchronized to GPS_PALISADE(0), stratum=0
7 Nov 22:00:53 ntpd[4073]: time reset +1.000021 s
7 Nov 22:01:56 ntpd[4073]: synchronized to GPS_PALISADE(0), stratum=0
9 Nov 06:07:28 ntpd[4073]: time reset +1.000026 s
9 Nov 06:08:30 ntpd[4073]: synchronized to GPS_PALISADE(0), stratum=0
```

De ninguna manera podemos permitirnos una falla de estas características, perdiendo prácticamente un día entero de sincronismo con el *GPS* y degradando así en forma importante, la calidad de todo el sistema de la Base de Tiempo del Instituto; motivo principal por el cual este proyecto esta en funcionamiento.

6. Conclusión

En la implementación para el soporte del *GPS Thunderbolt* en el *driver* para *NTP* del *GPS Palisade* se tuvieron que agregar algunas de las características propias (resueltas ya en el *hardware*) del *GPS Palisade* que el *Thunderbolt* no poseía. Algunas de las más importantes son:

- La posibilidad de que *NTP* sepa las fracciones de segundo del *GPS* a partir del *TOW (Time of Week)*.
- La implementación del *status* del *tracking* fuera del paquete primario de sincronización, y que con esto, *NTP* pueda saber no solo el tiempo, sino también la calidad del mismo.
- La posibilidad de que el *driver* active la alarma de *leap* para que *NTP* sepa cuando aplicar el salto de 1 segundo. Esto también fue implementado fuera del paquete de sincronización primario.

Finalmente para encarar el problema de pérdida de sincronismo presentado anteriormente intentamos consultando a las listas de correo electrónico que existen para el desarrollo de nuevos *drivers* para *NTP* de relojes de referencia externos, ya que lo que nos replicó el fabricante al consultarlo fue que el modelo de *GPS* que usamos para este proyecto, no es el indicado, pues éste fue pensado desde el momento de fabricación para el sincronismo electrónico de redes *wireless* y comunicaciones digitales, no para la sincronización de sistemas informatizados.

Las respuestas desde las listas de correo fueron nulas, nadie desarrolló, ni planea hacerlo al menos por ahora, un *driver* para el modelo de *GPS* como el que posee el Instituto.

7. Propuesta

El único y principal objetivo de este trabajo fue siempre la implementación de una Base de Tiempo estable y perfectamente sincronizada de acuerdo con los estándares usados en otros observatorios. Justamente por esta razón, no podemos desarrollar un sistema de sincronización de alta precisión partiendo de la base que el equipamiento que vamos a usar, no es el adecuado. Por lo tanto proponemos la adquisición de un equipamiento acorde a las necesidades del Instituto.

Paralelamente, el equipo ya disponible no quedara en desuso ni obsoleto, sino que se utilizará en las áreas a las que aplica, manteniendo en sincronismo todo el equipamiento digital de Sala de Control.

Para esto, estamos buscando la salida mas conveniente, tecnológica y económica, para satisfacer las necesidades de la Base de Tiempo, y por ende las del Instituto.

Tenemos entendido que los medios tecnológicos hoy en día utilizados para la sincronización son estándares y de gran soporte en varias corporaciones, por lo tanto, vemos viable una solución a corto plazo y en la medida que este a nuestro alcance, económico.