Proyecto Investigación
Protocolo NTP
Network Time Protocol

Sebastián Gallardo
201410006-K

Jean Paul Chánique C.
2923004-8

Mauricio Caceres R.
201666015-1

Amadeus Silva A.
201421071-k

Viernes 23 de Junio

1. Resumen

Los diversos procesos que se ejecutan por dispositivos en la red requieren una buena sincronización de sus relojes para obtener un buen rendimiento. Ante ello, surgen soluciones como el protocolo NTP, protocolo de la capa de aplicación encargado de solucionar este problema. Los servidores que envían paquetes de este tipo están organizados jerárquicamente en diferentes estratos, según la fuente de obtención de la hora universal coordinada (UTC).

Los clientes y servidores participantes estructuran paquetes con diferentes campos, que entregan información acerca del estrato y la IP del servidor de referencia, cada cuanto tiempo envían paquetes NTP, entre otros, siendo los más importantes las *timestamps*. Estos campos entregan información del momento de recepción y envío del paquete, desde su salida desde el computador del cliente hasta la llegada de la respuesta del servidor. Usando estas estampas de tiempo, el cliente puede obtener los valores de *offset*(Diferencia entre su reloj y el del servidor de referencia) y *delay* (Tiempo de ida-vuelta del paquete al servidor). Con ello, y tras diversas consultas y pruebas, elige un valor de *offset* adecuado. Con esto procede a cambiar su hora local de manera gradual. De esta forma, se resuelve el problema de manera simple y coordinada.

2. Introducción

Las operaciones de la red requieren información sincronizada para asegurar un rendimiento optimo. A menudo, cuando hay falta de sincronización, ésta se convierte en el factor clave, ya sea de la falla, o de la capacidad para sobreponerse a una. En otras instancias, los procesos de la red no podrán funcionar sin sincronización de sus relojes.

Los sistemas operativos comunes actualizan su hora local de manera iterativa. Para ello, cada cierto tiempo (normalmente milisegundos) el sistema genera una *interrupción* en las tareas que está ejecutando para proceder a ejecutar la rutina de actualización del reloj, que además de actualizar la hora local agregando el tiempo que ha pasado desde la última actualización, revisa si se han cumplido ciertos plazos y temporizadores que maneja internamente para diferentes procesos. Sin embargo, esto no es suficiente, pues evidentemente la ejecución de esta rutina genera un retraso que no es considerado por el sistema, lo que ocasiona que, a largo plazo, nuestro reloj interno sufra retardos respecto a la hora universal coordinada.

A continuación se procederá a realizar una breve reseña histórica del protocolo NTP, recalcar su importancia, explicar la organización jerárquica de los actores en este tipo de redes, describir la estructura del paquete de este protocolo y sus campos, ilustrar su funcionamiento analizando una captura de paquetes en *Wireshark*, finalmente y calcular con ella los parámetros de corrección del tiempo.

3. El protocolo NTP

El protocolo NTP (*Network Time Protocol*) es el principal protocolo para la sincronización de relojes de diferentes sistemas informáticos en red. Dicho de una forma más coloquial, permite *transferir el tiempo* a lo largo de la red, y entregar referencias de la hora universal coordinada a los diferentes clientes. Corresponde a un protocolo de la capa de aplicación, y debido a la necesidad de una rápida transmisión de las referencias horarias, usa el protocolo de capa de transporte UDP, a través de una red IP. En especifico, usa el puerto 123 de UDP. Es importante recalcar que se usan estampas de tiempo en base al sistema UTC[1]

Es uno de los protocolos más antiguos de Internet usado hoy en día, con un continuo uso por más de 25 años. Fue diseñado para sincronizar ordenadores y procesos dependientes del tiempo en la red. Fue inicialmente diseñado para el sistema operativo Linux, y hasta el día de hoy sigue instalado por defecto en muchos sistemas Unix y distribuciones BSD. A pesar de ello, fue migrado posteriormente a Windows

El protocolo funciona bajo arquitectura *cliente-servidor* provee, básicamente, de 3 datos importantes para la sincronización de reloj: *offset*, *delay* y *dispersión*. El offset especifica la diferencia entre la hora del sistema local y la referencia externa de reloj. El delay especifica las latencias de tiempo medidas durante la transferencias de paquetes dentro de la red. La referencia de dispersión de tiempo especifica el máximo número de errores asociados con la información de tiempo recibido de un reloj externo. Volveremos a estos datos cuando estudiemos el envío y recepción de paquetes NTP, así como el cálculo de estos campos [1].

3.1. Estratificación de Servidores NTP y Sincronización

Los servidores NTP están ordenados en una estructura jerárquica. Los servidores de *estrato 0* corresponden a relojes de cesio, satélites de GPS, entre otros. Quienes se conectan directamente a estas referencias de tiempo físicas constituyen el *Estrato 1*.Así mismo, los clientes que se sincronizan con los de *Estrato 1* constituyen el *Estrato 2*, y así sucesivamente. Esta jerarquía es dinámica, por lo que un cliente puede modificar su nivel y sincronizarse con un servidor de estrato mayor o menor (Fig. 6).

Cuando el sistema se inicia se sincroniza de acuerdo a su pila (estado antes de apagar el sistema), el cual suele ser el estrato 10. Luego, comienza el intercambio paquetes usual con diferentes servidores del estrato anterior, con una frecuencia de 64 segundos, la cual disminuye hasta los 1024 segundos por defecto, a medida que el reloj se sincroniza. Esto no se realiza de manera automática, sino que el reloj local se acerca de manera exponencial a la hora de referencia entregada por el servidor, para evitar tener *saltos* de tiempo que pueden ocasionar problemas en los plazos que el sistema operativo maneja.

[1] Tiempo Universal Coordinado

Una vez realizada la sincronización,se continuarán intercambiando paquetes cada 1024 segundos con los servidores conocidos, descartando los datos de los servidores en que la diferencia entre su reloj y la referencia elegida es superior a 128 milisegundos. Sin embargo, si durante más de 900 segundos todas las respuestas están fueras de ese rango, el sistema considerará su reloj como desincronizado y procederá a sincronizarlo nuevamente, con los servidores conocidos [6].

4. Estructura del Paquete NTP

La cabecera del paquete NTP (Fig. 7) se compone de los siguientes campos.

1. *Leap Indicator* (2 bits): Debido a variaciones en la rotación de la tierra, se utiliza el indicador de salto para señalar si al día actual se le debe sumar o restar un segundo, o mantener la cantidad normal de segundos de 1 día.

2. *Version Number* (3 bits): Número de versión, actualmente el convenio se encuentra en la versión 4.

3. *Mode* (3 bits): Indica el modo en que funciona el paquete.(0: Reserved, 1: Symmetric active, 2: Symmetric passive, 3: Client, 4: Server, 5: Broadcast, 6: NTP control message, 7: Reservado)

4. *Stratum*(8 bits) Indica el nivel de estrato de referencia del reloj local. 0 indica no especificado o inválido, 1 indica servidor primario, del 2 al 15 servidor secundario y 16 indica desincronización. El resto son valores reservados.

5. *Poll* (8 bits): Indica el log_2 del valor correspondiente al máximo intervalo entre sucesivas consultas NTP, en segundos.

6. *Precision* (8 bits): Indica el log_2 de la precisión del reloj del sistema, en segundos.

7. *Root Delay*(32 bits): Tiempo de ida y vuelta (RTT) desde el servidor a la referencia primaria indicada. Corresponde a un número punto fijo con signo que indica el RTT en segundos, con la parte fraccionaria entre el bit 15 y 16. Este campo es relevante solamente en los mensajes del servidor.

8. *Root Dispersion*(32 bits): Mismo formato de *Root Delay*. Indica el máximo error tolerable, propio de la medición. Solo es relevante en los mensajes del servidor.

9. *Reference Identifier* (32 bits): Para servidores primarios (estrato 1), es un código ASCII que describe la referencia externa de sincronización del reloj (Fig. 8). Para servidores secundarios, corresponde a la IPv4 del reloj de sincronización, o los primeros 32 bits del hashing MD5 de la dirección IPv6 del mismo.

Los siguientes 4 campos se expresan en segundos transcurridos desde la fecha de referencia, 1 de enero de 1900.

10. *Reference Timestamp*(64 bits): Hora a la que el reloj fue ajustado o corregido por última vez.
11. *Origin Timestamp*(64 bits): Hora a la que la solicitud es enviada al servidor.
12. *Receive Timestamp*(64 bits): Hora a la que la solicitud llega al servidor. Es *seteado* por el servidor al enviar el paquete de respuesta.
13. *Transmit Timestamp* (64 bits): Hora a la que la respuesta es enviada desde el servidor. Es *seteada* por el servidor al enviar el paquete de respuesta.

El resto de campos corresponden a opciones del paquete [4].

5. Funcionamiento del protocolo NTP, y corrección del reloj

El funcionamiento básico del protocolo, en cada envío de paquetes NTP es bastante simple. El cliente arma el paquete, indicando el valor del campo *Origin Timestamp*(T1), y procede con el envío. Cuando el servidor recibe el paquete, almacena el tiempo UTC a la cual la solicitud fue recibida. Con esto, el servidor ensambla un nuevo paquete, replicando el campo *Origin Timestamp* anterior, indicando el valor del campo *Receive Timestamp*(T2) igual a la hora almacenada, y agregando el valor del campo *Transmit Timestamp* (T3) igual al tiempo en que sale el paquete del servidor, con destino al cliente. Al llegar el paquete de respuesta al cliente, el sistema almacena el tiempo en el cual el paquete llegó de vuelta (Fig. 9).

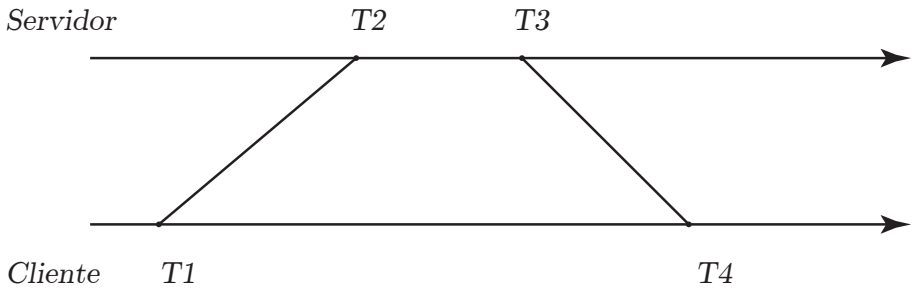

Figura 1: Diagrama temporal de envío de paquetes.

Con estas 4 estampas de tiempo, el sistema procede a calcular el *offset*(θ) y el *delay*(δ) de su reloj local respecto a ese servidor:

$$\delta = (T4–T1)–(T3–T2) \qquad \theta = \frac{(T2–T1)+(T3–T4)}{2}$$

Es importante notar un supuesto importante que realiza el protocolo NTP: este cálculo asume que el “viaje"de ida del paquete tarda el mismo tiempo que el viaje de vuelta. Esto es un supuesto bastante fuerte, lo que puede ocasionar errores importantes en el cálculo del offset. Para esto, el sistema guarda el valor de *jitter*, que corresponde a la desviación entre distintas respuestas del mismo servidor, el cual se usará como explicaremos un poco más adelante.

5.1. Sección Práctica - Analizando NTP con Wireshark

Si usamos Wireshark para obtener paquetes NTP (usando el filtro “ntp"), obtenemos la siguiente respuesta:

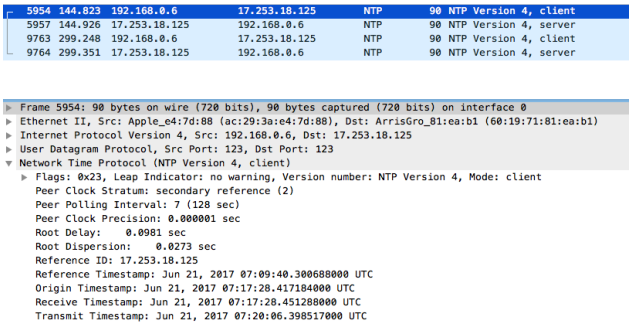

Figura 2: Captura Paquete de salida (Cliente $\Longrightarrow$ Servidor)

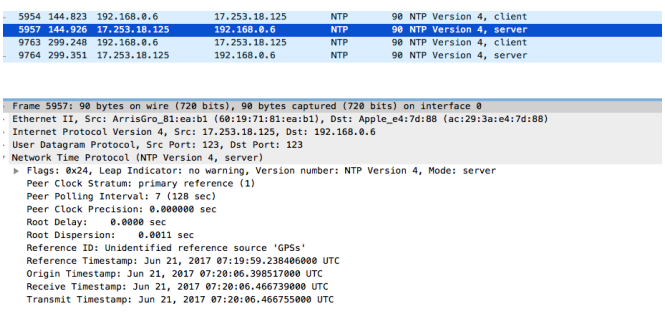

Figura 3: Captura Paquete de entrada (Servidor $\Longrightarrow$ Cliente)

Con estas capturas, podemos observar el estrato al cual pertenece nuestro computador (Estrato de referencia secundaria en este caso), y nuestro servidor de referencia (De referencia primaria). Otro campo importante a observar es el de *Reference Identifier* del cliente, el cual tiene el valor de la IP de nuestro servidor (Lo que se puede comprobar comparando con la IP de origen y de destino). En este caso, corresponde a la IP del servidor de *Apple Computer*, en California. Sin embargo, los campos en que nos concentraremos serán los que nos entreguen las distintas estampas de tiempo. Con esto, obtenemos las siguientes variables para el cálculo de *offset* y *delay* de transmisión:

- T1 = Origin Timestamp
- T2 = Receive Timestamp
- T3 = Transmit Timestamp
- T4 = Destination Timestamp = Receive Timestamp Captura 2

Para la realización del cálculo, usamos un pequeño *script* de *Python*, donde usando una librería especial para operar entre *timestamps*, y luego aplicando las fórmulas explicadas anteriormente, nos entrega el siguiente resultado:

```
MacBook-Pro-de-Sebastian:proyecto sebastian$ python calculo.py
Delay = (T4-T1) - (T3-T2) =  0.102342
Offset = 0.5*((T2-T1)+(T3-T4)) =  -0.017067
```

Figura 4: Resultado Cálculo *Offset* y *Delay*

Finalmente, podemos comparar estos resultados con los entregados por el comando de Linux ntpq −p

```
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
*brsao4-ntp-001. .GPSs.           1 u    1  128  373   98.070   15.007  18.332
```

Figura 5: Ejecución “ntpq -p”

Podemos ver un pequeño error en nuestro cálculo. Es posible que se haya dado por la imprecisión del cálculo manual frente al que realiza el sistema operativo.

NTP resuelve problemas de medición o pérdida de paquetes de manera inteligente; elige el **mínimo de las últimas 8 mediciones de delay**. Luego, el *offset* seleccionado corresponde a la medición con *delay* mínimo. Con esto, el *offset* y *delay* elegido se convierten en los valores de actualización del reloj local.

La forma en la que el sistema actualiza su hora local depende de la configuración del cliente con el/los sistema (s). Si el cliente está configurado con un único servidor de tiempo, el reloj local se actualiza **de manera gradual**, disminuyendo el *offset* poco a poco hasta llevarlo a 0. Esto se realiza para evitar posibles *saltos* de tiempo, que pueden ocasionar errores en el sistema operativo. Si está configurado con más de un servidor de tiempo, el sistema usa el *Algoritmo de Marsullo* para elegir los datos del servidor de menor estrato, y con los valores mínimos de *delay* y *jitter*.

6. Conclusiones

El problema de la sincronización de sistemas y retrasos de la hora local puede ser resuelto de manera simple y robusta, tal como analizamos y demostramos de manera práctica. Sin embargo, se conocen problemas que podrían presentarse en el futuro si no se optimizan algunos campos o se modifica ligeramente el protocolo, como el problema de los 136 años, el cual surge de la cantidad de bits disponibles para expresar el tiempo transcurrido desde la referencia. Una forma de solucionar es la implementación de referencias a Era en el campo opcional NTP (Fig. 10). A pesar de ello, es un protocolo que funciona hasta el día de hoy, después de años de ser adoptado, lo que muestra lo eficiente de su funcionamiento.

7. Referencias

[1] Ordenadores y Portátiles. (2014). “Sincronizando Sistemas Digitales con NTP”.

http://www.ordenadores-y-portatiles.com/ntp.html

[2] D.Mills, U. Delaware, J. Martin, J. Burbank, W. Kasch. (2010). “Network Time Protocol Version 4: Protocol and Algorithms Specification "[Versión Digital] *RFC5905*

https://tools.ietf.org/html/rfc5905

[3] Meinberg Industries. “Technical information - NTP Data Packet"

https://www.meinbergglobal.com/english/info/ntp-packet.htm

[4] Cisco Systems, Inc. , Geoff Huston, APNIC (2012). “Protocol Basics: The Network Time Protocol - The Internet Protocol Journal, Volume 15, No. 4 "

http://www.cisco.com/c/en/us/about/press/internet-protocol-journal/back-issues/table-contents-58/154-ntp.html

[5] Meinberg Industries. “Network Time Protocol - Protocolo de Hora en Red (NTP)"

http://www.meinberg.es/soporte/informacion/network-time-protocol.htm

[6] Enrique V. Bonet Esteban. “ Servicios avanzados III: Sincronización de la hora mediante NTP". Apuntes de Administración y Gestión de Redes. Departamento de Informática. Universidad de Valencia.

http://informatica.uv.es/it3guia/AGR/apuntes/teoria/documentos/NTP.pdf

[7] H3C Technologies Co., “Configuring NTP"Section 1.1.2 *How NTP works*

http://www.h3c.com.hk/Technical_Support___Documents/Technical_Documents/Switches/H3C_S12500_Series_Switches/Configuration/Operation_Manual/H3C_S12500_CG-Release7128-6W710/12/201301/772699_1285_0.htm#_Toc341178391

8. Anexos

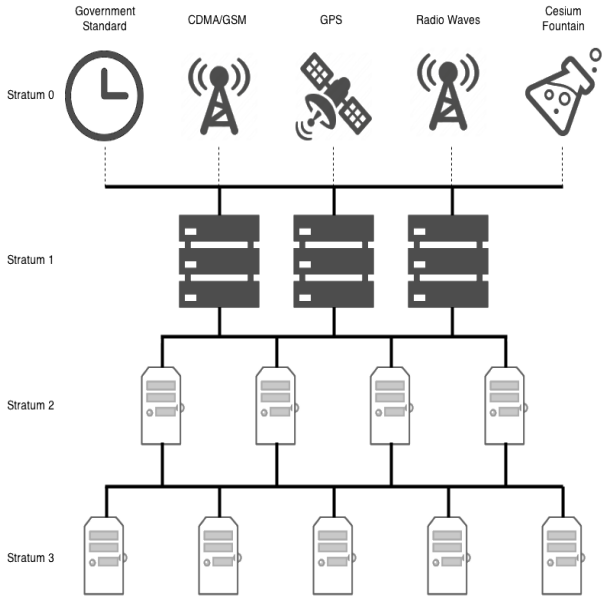

Figura 6: Organización Servidores NTP por Estratos

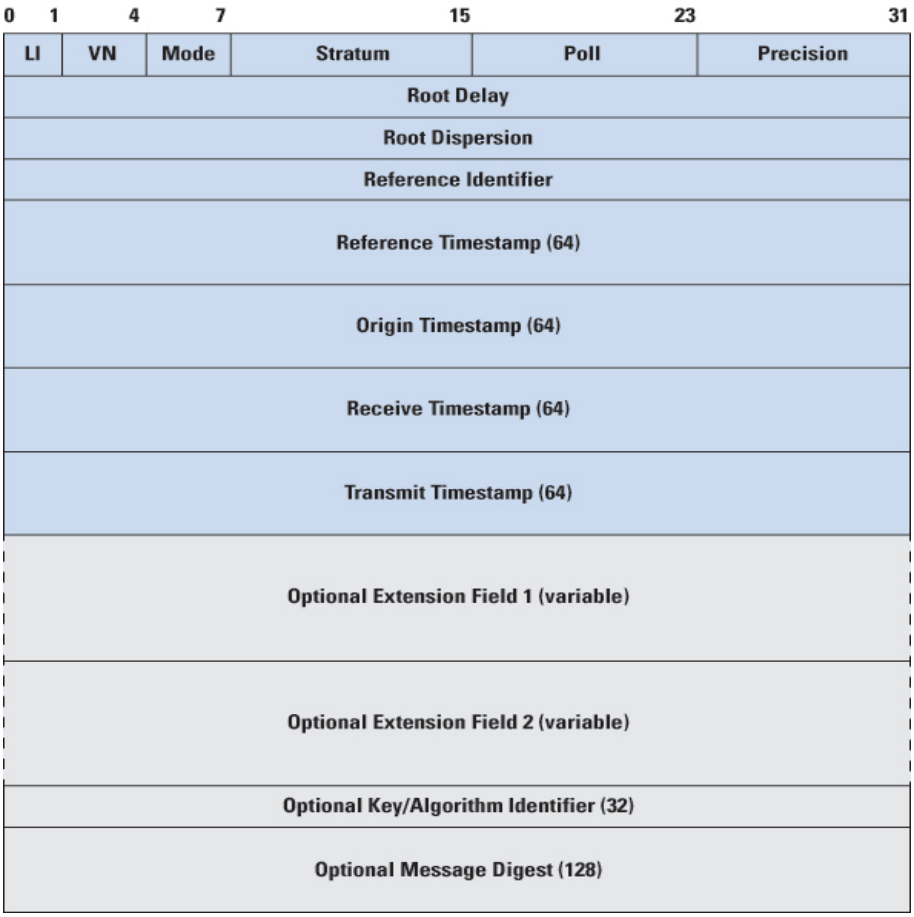

Figura 7: Diagrama de Paquete NTP

Code	External Reference Source
LOCL	uncalibrated local clock
CESM	calibrated Cesium clock
RBDM	calibrated Rubidium clock
PPS	calibrated quartz clock or other pulse-per-second source
IRIG	Inter-Range Instrumentation Group
ACTS	NIST telephone modem service
USNO	USNO telephone modem service
PTB	PTB (Germany) telephone modem service
TDF	Allouis (France) Radio 164 kHz
DCF	Mainflingen (Germany) Radio 77.5 kHz
MSF	Rugby (UK) Radio 60 kHz
WWV	Ft. Collins (US) Radio 2.5, 5, 10, 15, 20 MHz
WWVB	Boulder (US) Radio 60 kHz
WWVH	Kauai Hawaii (US) Radio 2.5, 5, 10, 15 MHz
CHU	Ottawa (Canada) Radio 3330, 7335, 14670 kHz
LORC	LORAN-C radionavigation system
OMEG	OMEGA radionavigation system
GPS	Global Positioning Service

Figura 8: Códigos ASCII de referencias para servidores primarios

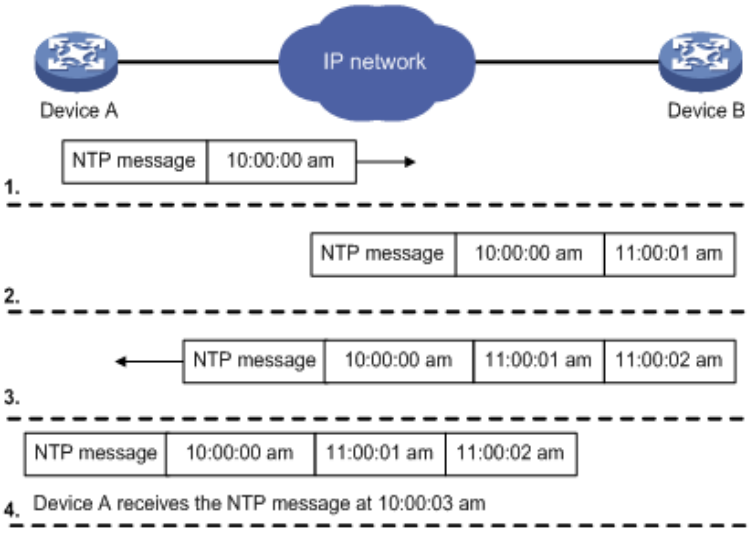

Figura 9: Ejemplo: Envío Paquete NTP Cliente-Servidor

Date	MJD	NTP Era	NTP Timestamp Era Offset	Epoch
1 Jan -4712	-2,400,001	-49	1,795,583,104	1st day Julian
1 Jan -1	-679,306	-14	139,775,744	2 BCE
1 Jan 0	-678,491	-14	171,311,744	1 BCE
1 Jan 1	-678,575	-14	202,939,144	1 CE
4 Oct 1582	-100,851	-3	2,873,647,488	Last day Julian
15 Oct 1582	-100,840	-3	2,874,597,888	First day Gregorian
31 Dec 1899	15019	-1	4,294,880,896	Last day NTP Era -1
1 Jan 1900	15020	0	0	First day NTP Era 0
1 Jan 1970	40,587	0	2,208,988,800	First day UNIX
1 Jan 1972	41,317	0	2,272,060,800	First day UTC
31 Dec 1999	51,543	0	3,155,587,200	Last day 20th Century
8 Feb 2036	64,731	1	63,104	First day NTP Era 1

Figura 10: Fechas para implementación de *eras* en NTP