**1.- Modos de Sincronización y algoritmos**

Existen dos modos principales de sincronización.

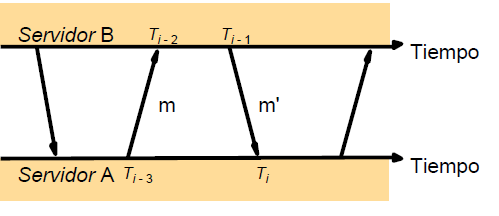
* **Externa**, el cual se caracteriza por que un reloj C se sincroniza con una fuente UTC (Tiempo Universal Coordinado) exacta S, la cual normalmente está basada en un tiempo atómico, con lo que es muy precisa.

Dentro de la sincronización externa, existen entre otros varios, dos algoritmos de sincronización, que son:

* **Método de Cristian:** Consiste en la petición de tiempo por parte de un proceso a un servidor S, que recibe señales UTC. Consiste en:

1. Un proceso solicita el tiempo en un mensaje mr y recibe t en mt de S.
2. El proceso establece su tiempo a , donde Tround es el tiempo de ida y vuelta.
3. Precisión ±

* **Método** **NTP**: Sincroniza a los clientes directamente con UTC.
  + - * Se caracteriza por su estructura de **árbol**: los nodos **raíz** (servidores) intercambian mensajes de tiempo con UTC. Estos se lo pasan al nivel siguiente por diversos mecanismos. Y estos a los siguientes, de manera recursiva.
      * Los nodos hojas son las máquinas **clientes**.
      * **Funciona**:



Donde:

o=compensación entre los dos relojes

d=tiempo transmisión entre los mensajes (t-t’)

* + - * Los servidores NTP mantiene pares del tipo **<oi, di>** (**compensación, diferencia** **tiempo de transmisión**) estimando la fiabilidad de las variaciones.
      * Existen 3 formas de **comunicación**:

**Multicast**: LANs de alta velocidad.

**Llamada a procedimiento**: similar a Cristian, el servidor acepta peticiones

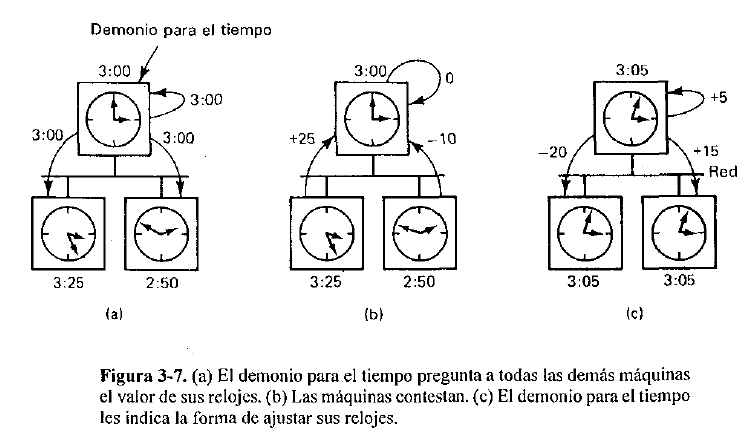
**Simétrica**: pares de servidores se intercambian mensajes conteniendo información de tiempo.

* + - * Los 3 modos usan **UDP** para el intercambio de mensajes.
        + Cada mensaje lleva marcas de tiempo de los **eventos recientes**.
* **Interna**, el cual se realiza entre cualquier par de computadores.

Dentro de la sincronización interna, existen entre otros varios, dos algoritmos de sincronización, que son:

* **Primera Aproximación**: Consiste en conocer el **tiempo de transmisión** entre dos procesos (que se define con límite inferior y superior (min, max)) y establecer el reloj como el tiempo local (t) y el tiempo de transmisión, que es (max – min)/2. Por tanto:
* **Método de Barkeley:** Consiste en una serie de pasos que promedia los resultados de varios sistemas. Pasos:

1. Un maestro consulta y recoge los valores del resto de esclavos.
2. Se utilizan los tiempos de ida/vuelta para estimar el valor de relojes.
3. Promedia los resultados incluyéndose y eliminando valores no consistentes.
4. Envía la magnitud de ajusta de cada reloj.

****

**1** Algoritmo Berkeley

**2.- Explicar porque el escenario 2 ("el del servidor") no es apto para el negocio electrónico/transacciones en un negocio electrónico**

Porque obliga al servidor de autenticación a tener controlados a todos los ***proveedores y usuarios***. El servidor de autenticación debería conocer todas las claves secretas de todos los clientes.

**4.- ¿El algoritmo TEA se podría llevar a otra arquitectura hardware?**

Sí, siempre y cuando el hardware admita arrays de **enteros largos** (32 bits) y operaciones con puerta **xor (**damos por hecho que los desplazamientos está intrínseco en cualquier hardware).

**5.- Definir entre los algoritmos simétricos y asimétricos criptográficos cuáles son mejor o peor**

Los algoritmos asimétricos son más seguros, ya que usan dos claves distintas para encriptar y desencriptar. Usan la vulnerabilidad que representa el algoritmo (puerta falsa) para generar dos claves.

**6.- Definir los pasos de Chandy-Lamport, cuando se acaba y finalidad**

Cada proceso registrar su estado y para cada canal entrante también un conjunto de mensajes enviados a él. El proceso **registra**, ***para cada canal***, todos los mensajes que entraron después de que él registrar el estado y antes de que el emisor registrara y propio estado.

**7- Define brevemente los siguientes conceptos referidos al tema Tiempo y estados globales:**

* **Tasa de deriva**

**Diferencia** por unidad de tiempo en que el reloj del computador difiere del reloj perfecto.

* **Reloj correcto**

Un reloj se considera correcta si su límite de **deriva** es **conocido** ρ > 0

* **Sistema distribuido asíncrono**

Puesto que sabemos cuándo un sistema distribuido es **síncrono**, podemos inferir que un sistema **asíncrono** se da cuando en un sistema no está definido **uno o más** de los límites siguientes:

1. Tiempo máximo y mínimo para **ejecutar** cada **paso** de un proceso
2. Tiempo máximo y mínimo de **recepción** de **mensaje**
3. Los **límites** de **deriva** de cada reloj local donde se ejecuta cada proceso son conocidos.

* **Certificado**

Sentencia firmada por un principal que sirve de credencial o autenticación. Éste necesita: un formato estándar acorado, acuerdo sobe las cadenas que lo forman y fecha de expiración.

* **Corte de la ejecución del sistema**

Un corte es un subconjunto de la historia global, que es la unión de los prefijos de las historias de los procesos.

* **Corte consistente**

Se refiere a lo anterior y si para cada evento que contiene, también contiene todos los sucesos que sucedieron antes de él se considera un corte consistente.

* **Reloj Lógico**

Un contador software monótono creciente. Cada proceso **p** tiene su reloj lógico que se utiliza para fijar las marcas temporales de los eventos.

**8- Desarrolla el ataque del cumpleaños indicando en que se basa, en que contexto se puede dar y qué aspectos son determinantes para protegernos de ese ataque.**

El ataque de cumpleaños se basa en la probabilidad de encontrar un par idéntico de resumen. Es decir, si existe si la función de dispersión no es muy efectiva, se puede dar el caso de que **H(M)=H(M’).**

Siguiendo lo anterior, con paciencia se puede encontrar dos mensajes cuyo resumen al aplicar la función de dispersión sea el mismo.

Si M’ es malicioso, y tenemos un M firmado digitalmente, podemos sustituir los mensajes y actuar con un mensaje que no era el esperado.

**9-Desarrolla el algoritmo de acceso a sección crítica llamado "Ricart-Agrawala". ¿Se cumplen en todos los casos las tres exigencias de exclusión mutua? Razona la respuesta. Además indica y justifica el número de mensajes que se necesitan en su funcionamiento.**

El **algoritmo** funciona de la siguiente manera:

Cuando un proceso desea acceder a un recurso compartido, elabora un mensaje que contiene el **nombre del recurso**, su **número de proceso**, y el **tiempo actual**. Entonces envía el mensaje a todos los demás procesos, incluyéndose de manera conceptual. Se supone que el envío de los mensajes es confiable; es decir, no se pierde mensaje alguno.

Una vez le han **contestado todos**, pone el estado de la sección crítica en **liberada/buscada/tomada.**

Si el proceso está en **tomado** o está **buscado** y además el tiempo del proceso p es inferior al del proceso p’, pone p’ en la cola de petición.

Si no está tomada o está buscada y el tiempo de p es superior al de p’, responde a la petición de p’.

**10-Desarrolla el algoritmo Nedham-Schroeder**

A -> S: pide B | NA

S -> A: devuelve {M, Kab, {Ticket}Kb}Ka| NA

A -> B: {Ticket}Kb={KAB, A}Kb

B -> A: {NB}KAB

A - > B: {NB - 1}Kab