

Sistemas Distribuidos y Concurrentes

Sincronización y Relojes

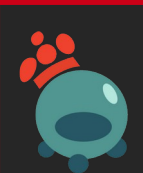
Grado en Ingeniería de Robótica Software

Teoría de la Señal y las Comunicaciones y
Sistemas Telemáticos y Computación

Roberto Calvo Palomino
roberto.calvo@urjc.es

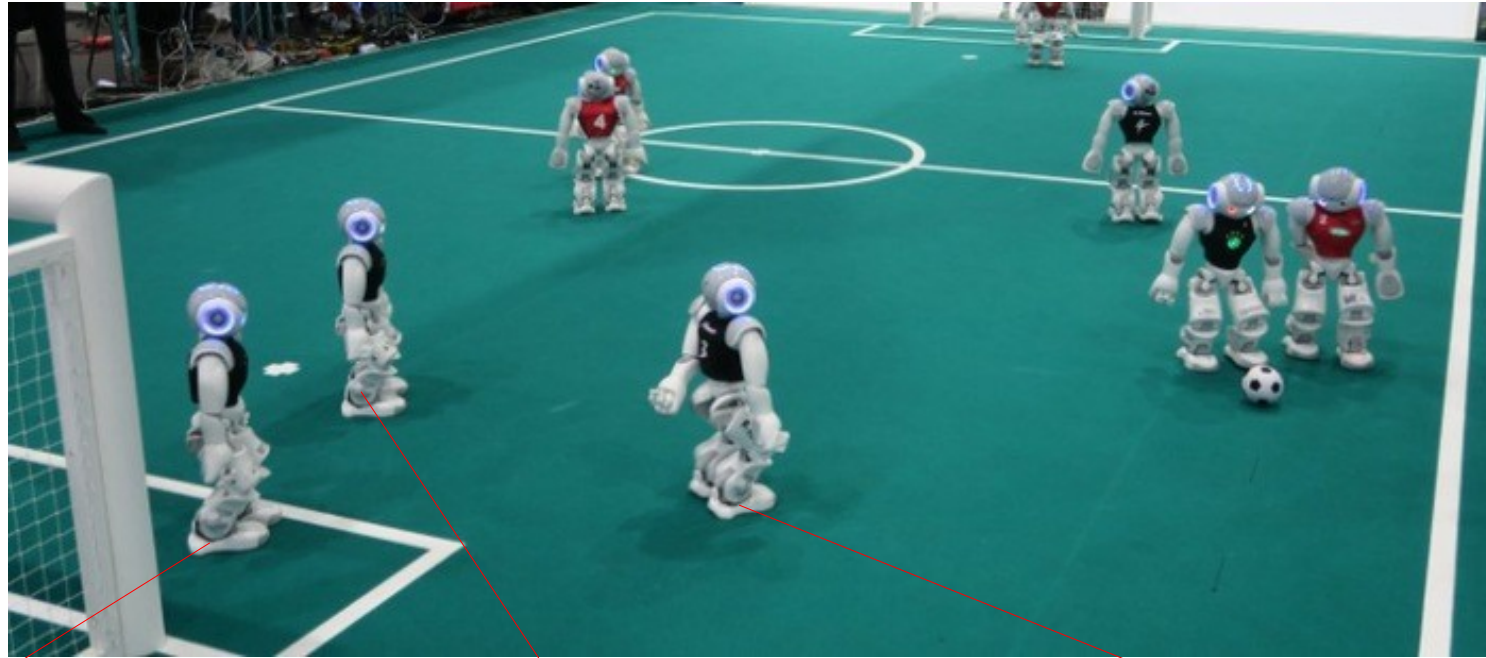
Relojes y Sincronización

- En sistemas distribuidos **NO** existe un único reloj hardware común.
- Cada nodo/componente tiene su **propio** reloj local.
- Necesidad de una **sincronización** para:
 - Aplicaciones en tiempo real
 - Ordenación natural de eventos distribuidos (logs, eventos, ...)
- La sincronización puede ser:
 - Relativa entre los nodos del sistema
 - Absoluta con la realidad (verdad absoluta).
- Multitud de problemas aparecen en sistemas distribuidos cuando se requiere sincronizar relojes.



Sincronización en Robótica

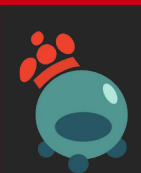
- Distributed Data Logging



Time1: Message1
Time2: Message2
Time3: Message3

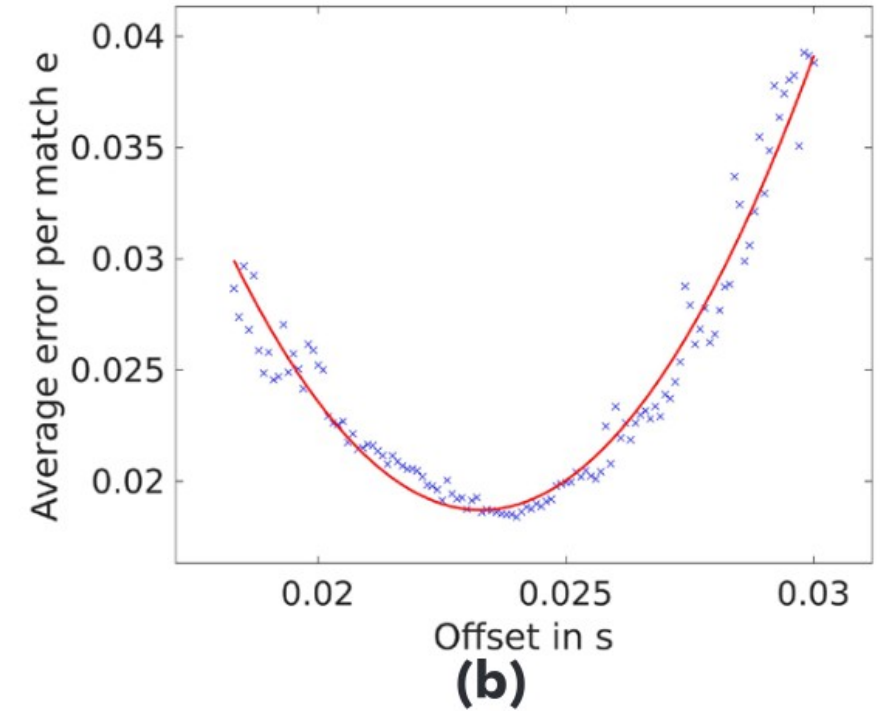
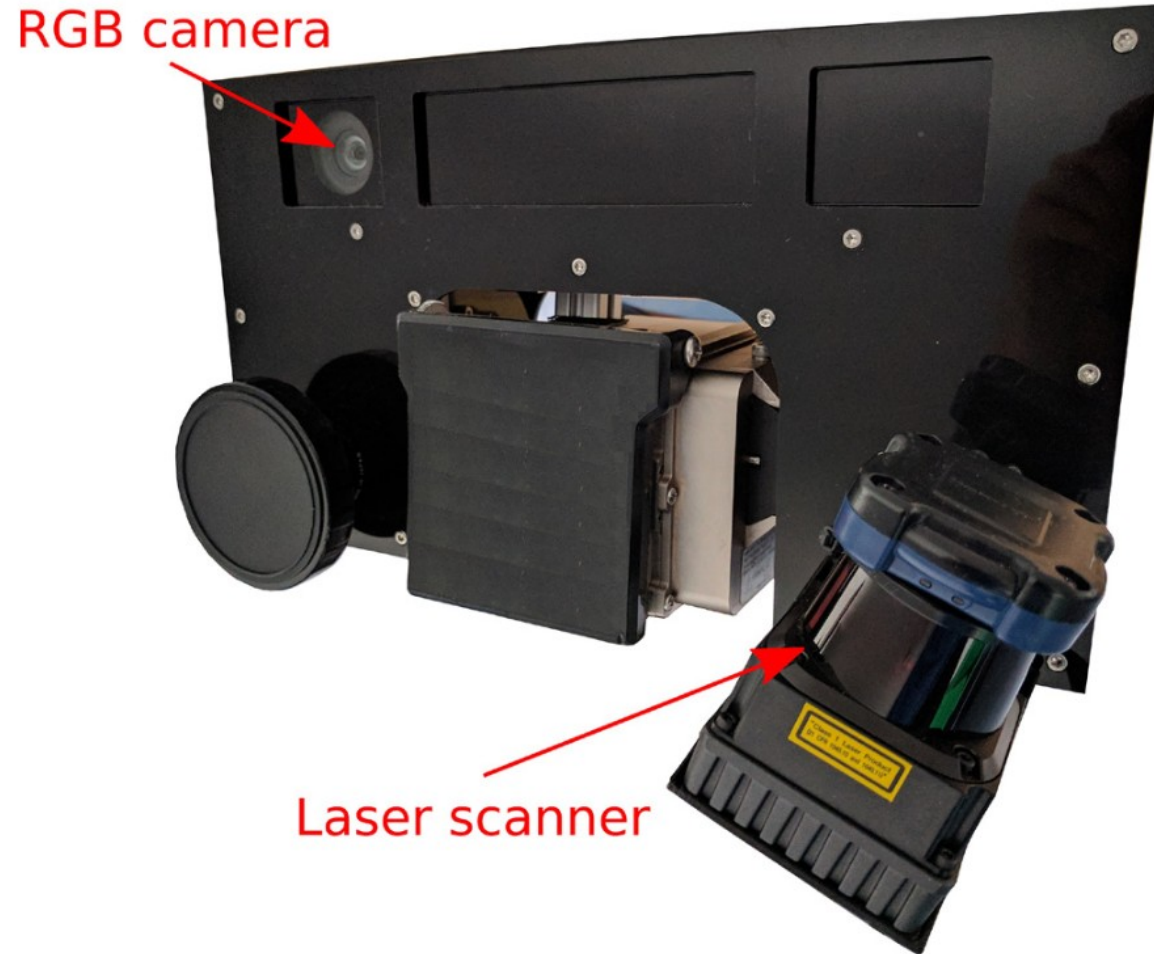
Time1': Message1
Time2': Message2
Time3': Message3

Time1': Message1
Time2': Message2
Time3': Message3



Sincronización en Robótica

SLAM



Sincronización en Robótica

SLAM

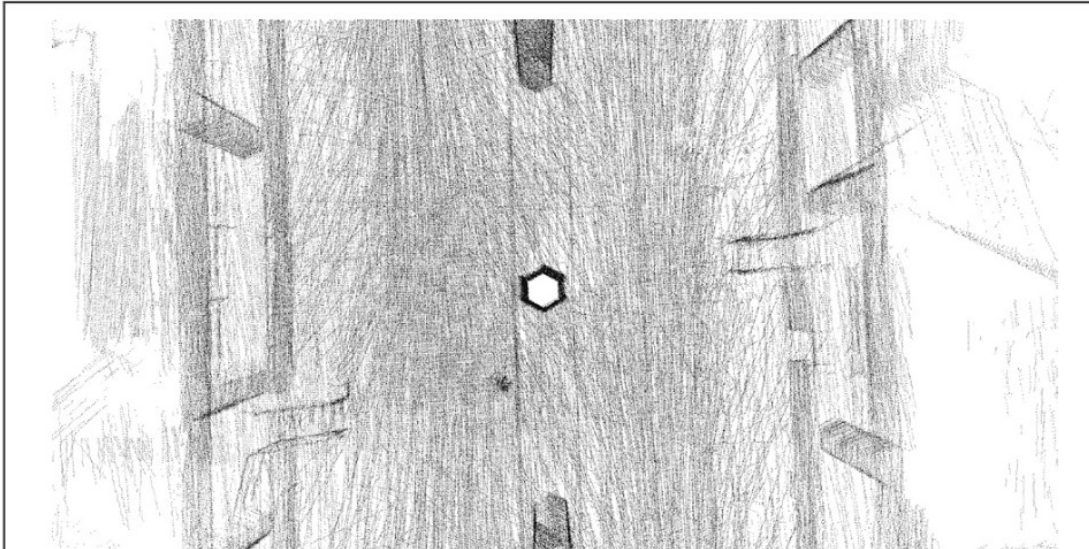


Figure 9. Map section generated by the SLAM approach for the metro station dataset using an appropriate timestamp offset between laser scanner and motor of 24 ms.

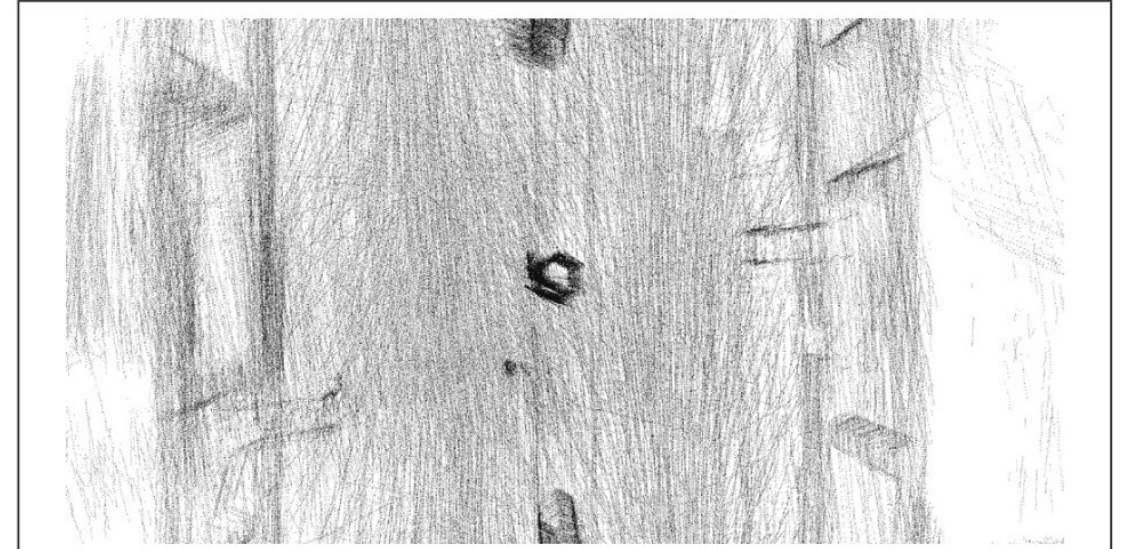
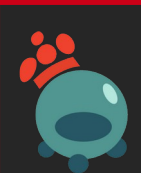


Figure 10. Map section generated by the SLAM approach for the metro station dataset using an inappropriate timestamp offset between laser scanner and motor of 19 ms.

by Indenta



Sincronización en Robótica

Robot-laser welding

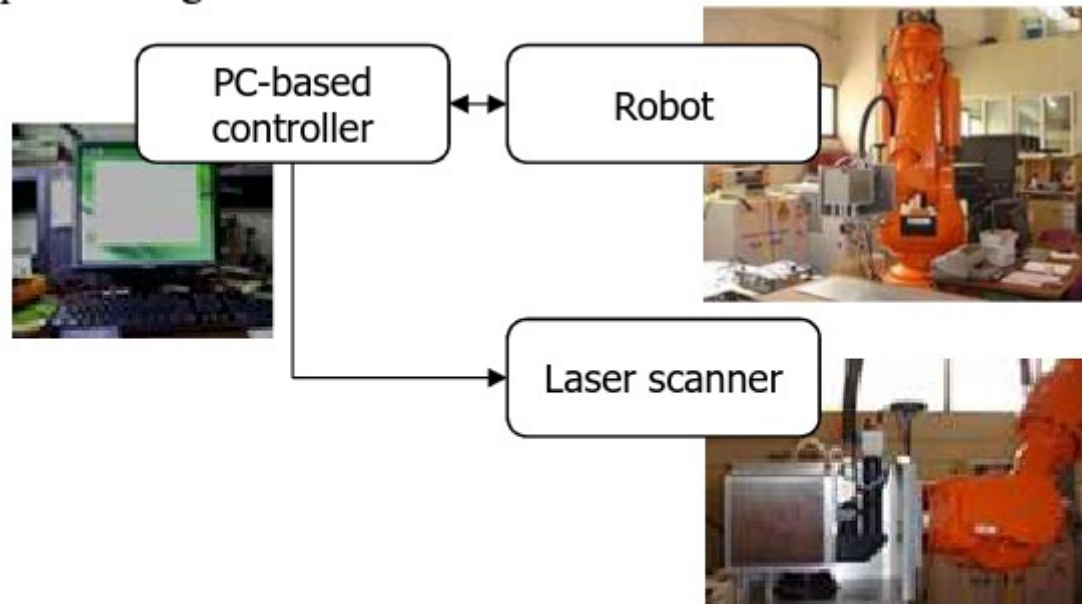


Fig. 1 Robot-laser welding system

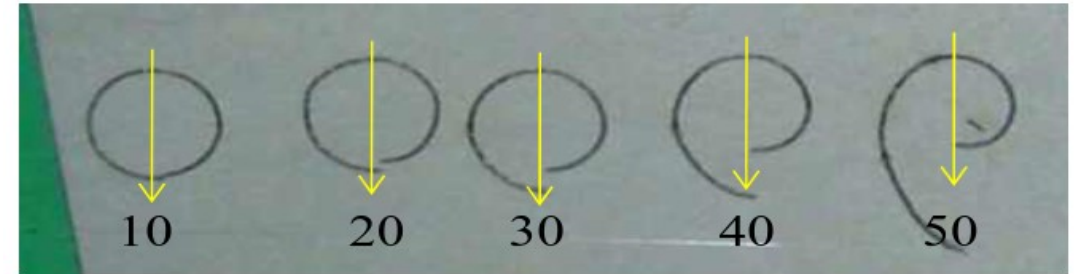


Fig. 6 Patterns welded by the robot speed of from 10 mm/s to 50 mm/s at the weld speed of 60 mm/s

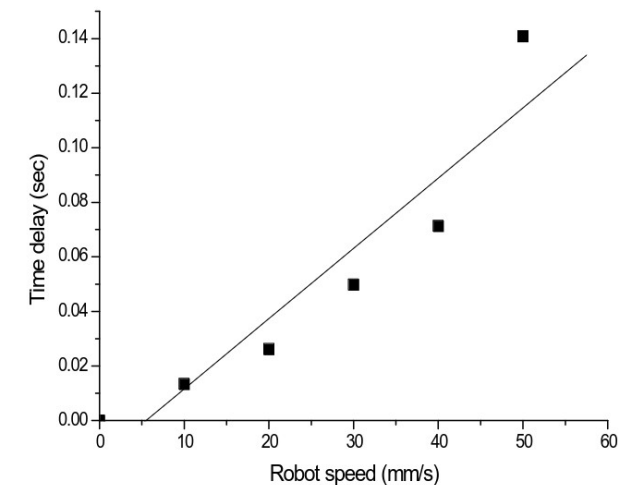
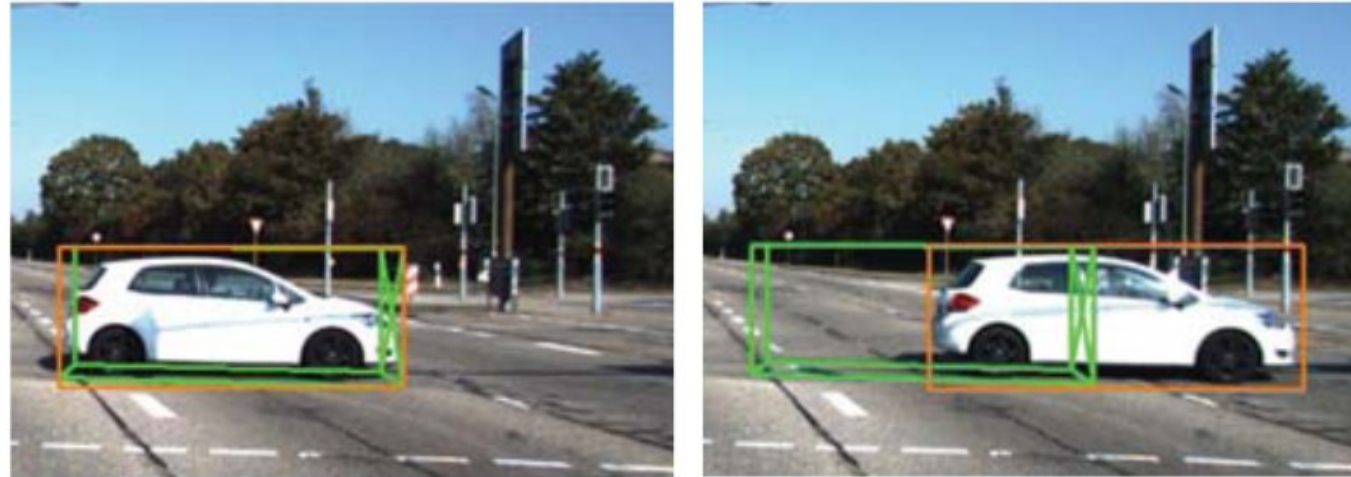


Fig. 7 Time-delay estimation through pre-experiments

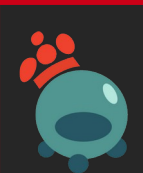
Sincronización en Robótica

Autonomous Driving



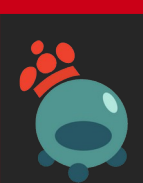
(b) Object detection using synchronized (left) and mis-synchronized (right) data. The green box denotes a 3D detection bounding box projected onto the 2D image and the orange box denotes a 2D detection bounding box.

Fig. 2: Impact of intra-machine camera-LiDAR synchronization on object detection.



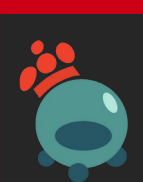
Concepto de Tiempo

- Civilizaciones antiguas se basaban en la observación del cielo y estrellas para contar el tiempo
 - La posición de los planetas y las fases de la Luna se convirtieron en la referencia
 - Cuando la Luna retornaba a la fase inicial de su órbita, el mes había terminado.
- Calendario más antiguo (**8000 a.C.**), trata de un monumento monolítico en Escocia compuesto por 12 piedras que marcan la posición de la luna a lo largo del un año
- Los sumerios (**3000 a.C.**) dividieron el año en 12 ciclos lunares dando paso a la estaciones.
- En Egipto (**1000 a.C.**) descubrieron que el año duraba 365 días, dividieron el calendario en 12 meses de 30 días.
- Julio Cesar (**46 a.C**) instauró el calendario Juliano
- El Papa Gregorio XIII (**1582**) instauró el **calendario Gregoriano**, utilizado actualmente.

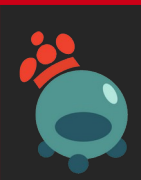


Concepto de Tiempo

- Antiguamente, la escala de tiempo se ligaba al período de **rotación** de la Tierra (1900)
 - El segundo estaba definido como $1/86.400$ del día solar medio
Problema: la rotación de la Tierra no es uniforme (interacciones gravitatorias con el Sol y la Luna).
- **Tiempo de efemérides** (TE), que se basa en el movimiento orbital de los astros y planetas y no tiene en cuenta la rotación de la tierra.
- **GMT** (Greenwich Mean Time), tiempo solar medio en el observatorio de Greenwich.
- El **reloj atómico** fue creado en 1948, es un tipo de reloj que para alimentar su contador utiliza una frecuencia de resonancia atómica normal.
 - Es más preciso y no depende de ningún fenómeno astronómico.

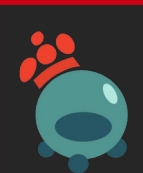


Concepto de Tiempo



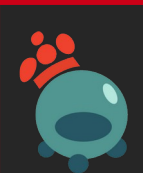
Concepto de Tiempo

- Reloj atómico de Cesio 133.
- Se define el segundo como:
 - El tiempo que necesita el átomo de cesio 133 para efectuar exactamente 9.192.631.770 transiciones.
 - La elección de 9.192.631.770 se hizo para igualar el segundo atómico con el segundo medio solar en el año de su introducción.
- Laboratorios de todo el mundo con relojes atómicos de cesio mandan de manera periódica, las marcas de sus relojes. **Tiempo Atómico Internacional (TAI).**



UTC

- **UTC**: Coordinated Universal Time
- Es el principal **estándar** de tiempo que regula relojes y el tiempo actualmente.
- UTC se obtiene a partir del **TAI** (Tiempo Atómico Internacional).
- UTC además se actualiza con el tiempo medio de Greenwich (tiempo solar medio)
 - A finales de Junio/Diciembre se añade o quita 1 seg. (segundos intercalares) acorde al Servicio Internacional de Rotación de la Tierra y Sistemas de Referencia



UTC

- UTC es el sistema de tiempo utilizado por muchos **estándares** de Internet y la World Wide Web
- Pero UTC presenta **problemas** para sistemas como UNIX
 - Tiempo UNIX/POSIX: guarda el tiempo como un número de segundos a partir de un tiempo de referencia (medianoche UTC del 1 de enero de 1970).
 - No tiene noción de los segundos intercalares.
- Se han diseñado **protocolos** específicos para distribuir el tiempo UTC en Internet:
 - NTP: Network Time Protocol.



¡Tiempo!



Sincronización

- **Resolución** (*resolution*). Se refiere a la mínima magnitud que un sistema puede medir.
- **Precisión** (*precision*). Se refiere a la dispersión del conjunto de valores obtenidos de mediciones en nuestro sistema.
- **Exactitud** (*accuracy*). Se refiere a cuán cerca del valor real se encuentran las medidas del sistema.

Exacto
Preciso



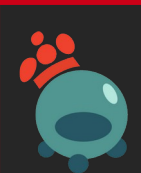
No Exacto
Preciso



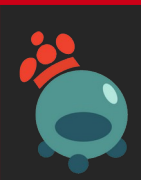
Exacto
No Preciso



No Exacto
No Preciso

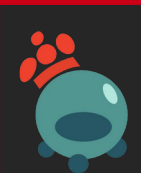


Sincronización



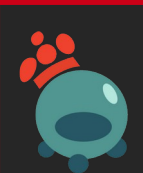
Sincronización

- Asumiendo que los relojes de nuestro sistema distribuido están sincronizados en un instante t , no podemos asumir que seguirán sincronizados en el instante $t+\Delta$
 - No todos los relojes avanzan al mismo ritmo
 - Razones técnicas (temperatura chip, etc.)
 - Razones físicas (relatividad, ej. GPS)
 - Offset: la diferencia con el reloj de referencia.
 - Drift: Los relojes no van a la misma velocidad (frecuencia).
- Es **necesaria** una sincronización para compensar los relojes dentro de un sistema distribuido.



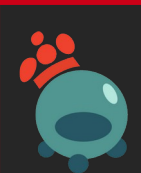
Sincronización de Relojes

- La **sincronización** de relojes en un sistema distribuido consiste en garantizar que los procesos se ejecuten en forma cronológica, y a la vez respetando el orden de los eventos dentro del sistema.
- Relojes físicos: Hardware que tiene noción del tiempo real.
- Relojes lógicos: Abstracción donde lo importante es la relación y orden de eventos/sucesos (no existe el concepto real de tiempo)



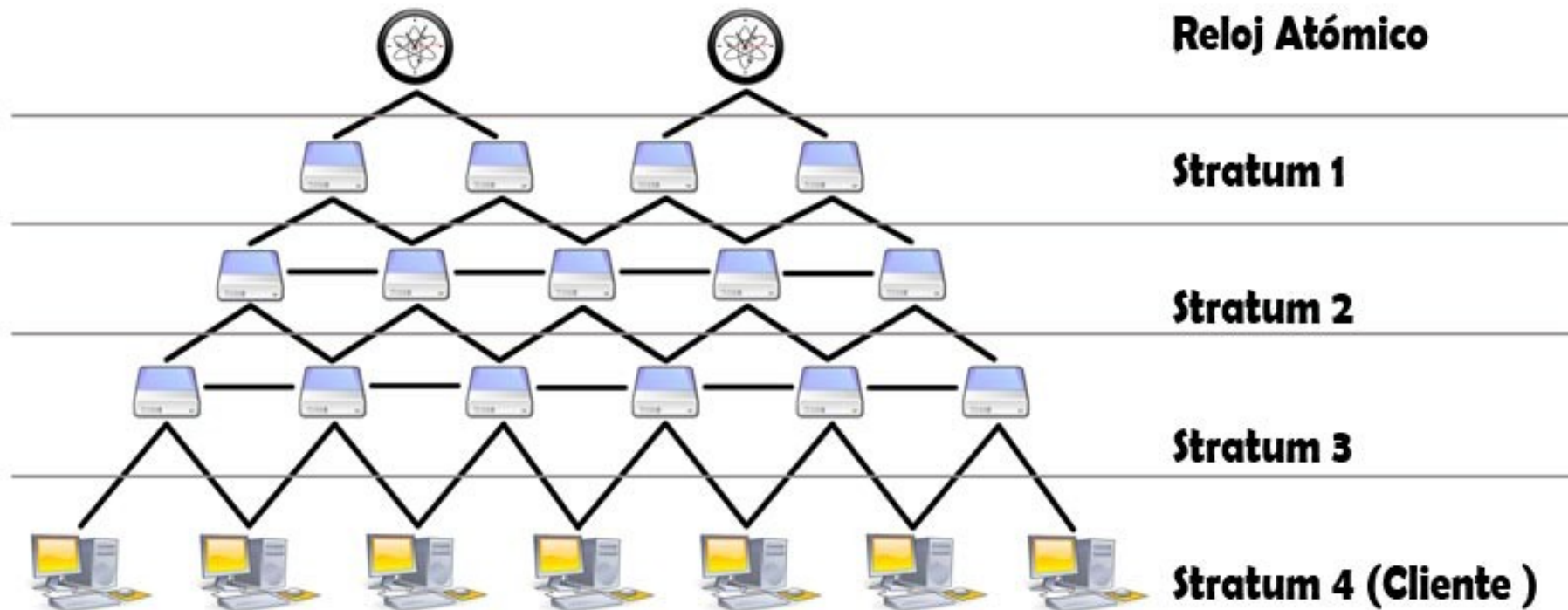
Sincronización de Relojes

- Sincronización Relojes Físicos (basados en UTC)
 - NTP: Network Time Protocol
 - Normalmente puede conseguir una sincronización en torno a decenas de milisegundos en ordenadores conectados a través de Internet.
 - GPS: Global Position System
 - Siempre que haya cobertura, puede conseguir sincronización en torno a ~ 40 nanosegundos.
- Sincronización Relojes Lógicos:
 - Relojes de Lamport
 - Relojes Vectoriales



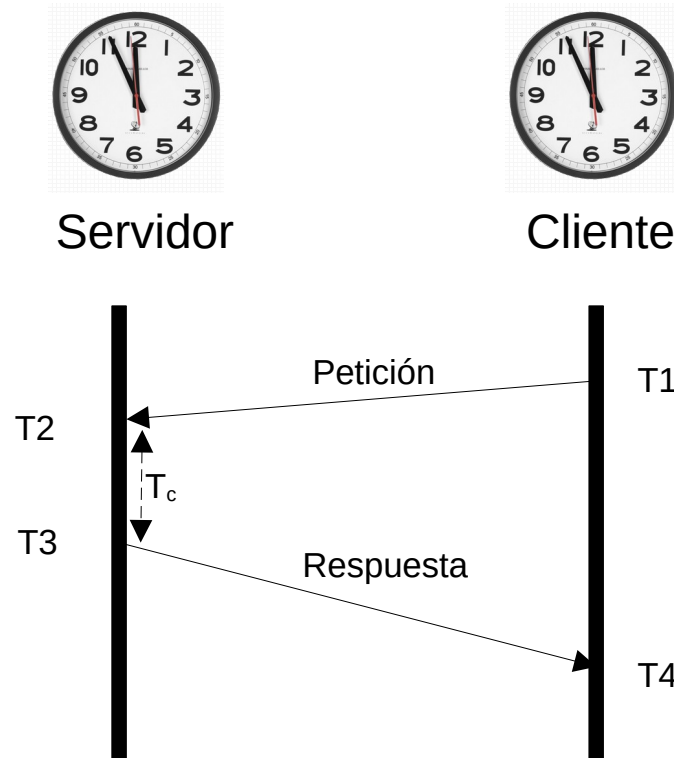
NTP

- Sistema síncrono que depende de relojes o unidades máster.
- Existe una jerarquía de relojes
 - Relojes de Cesio, rubidio en la raíz



NTP

- El cliente se conecta al servidor
- El servidor contesta con una marca de tiempo NTP.
- Los mensajes tardan tiempo en enviarse y llegar.
- El cliente debe compensar el RTT (*round-trip time*)

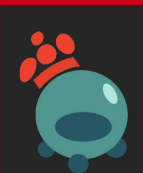


$$RTT = T_{\text{peticion}} + T_c + T_{\text{respuesta}}$$

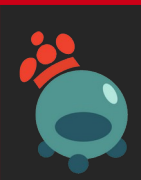
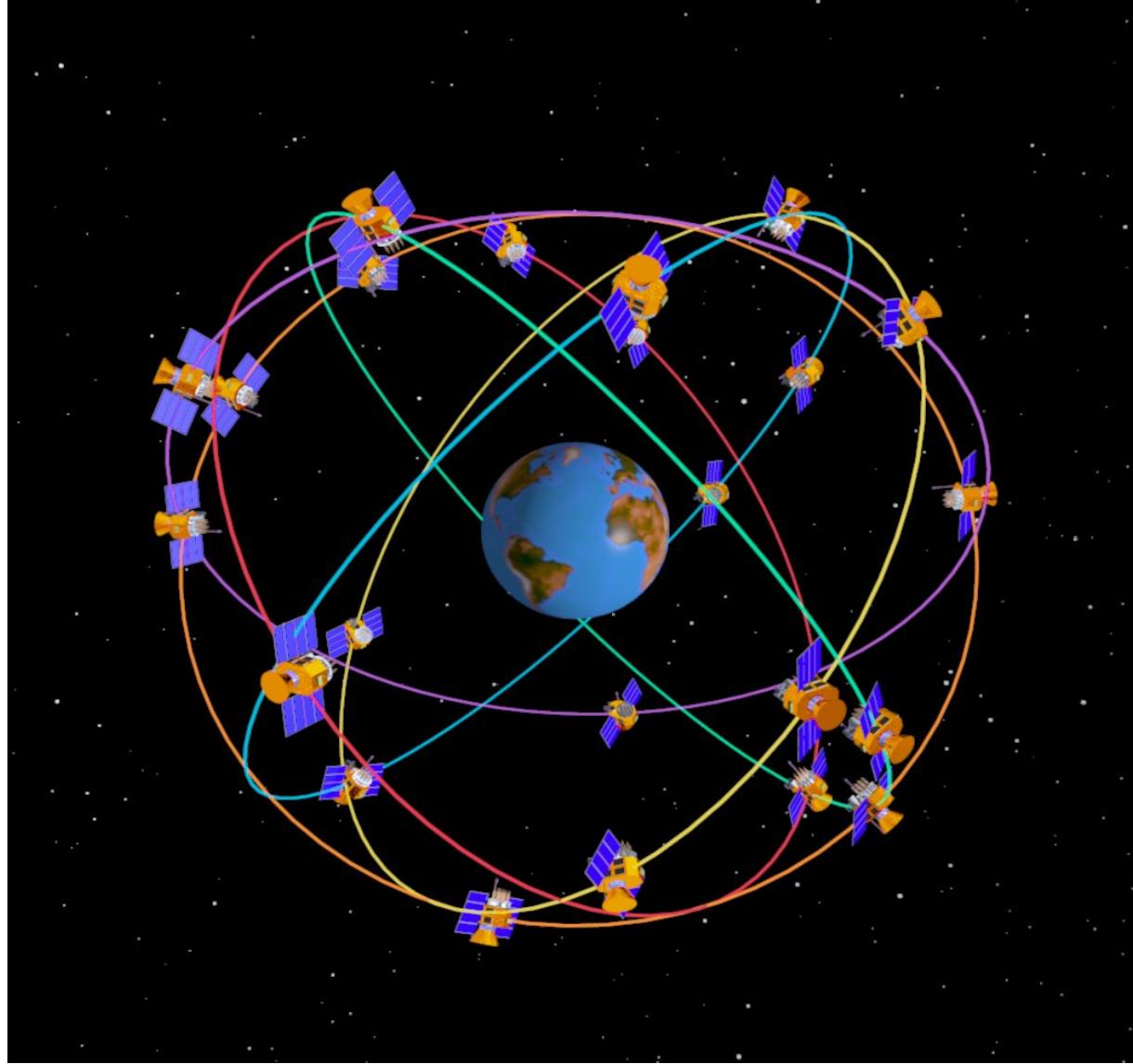
NTP

- *systemd-timesyncd* es el demonio en GNU/Linux que se encarga de actualizar la hora del sistema mediante NTP.

```
$ timedatectl timesync-status
    Server: 193.145.15.15 (0.debian.pool.ntp.org)
Poll interval: 8min 32s (min: 32s; max 34min 8s)
    Leap: normal
    Version: 4
    Stratum: 2
    Reference: C1936B21
    Precision: 1us (-24)
Root distance: 33.218ms (max: 5s)
    Offset: +1.647ms
    Delay: 8.382ms
    Jitter: 1.267ms
Packet count: 4
    Frequency: +4,259ppm
```

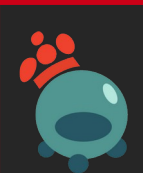


GPS



GPS

- Sistema de satélites que orbitan la Tierra a unos 20.000 km de altura y a una velocidad de unos 14.000 km por hora.
- Cada satélite contiene un reloj atómico.
- Teoría de la **relatividad espacial**
 - Como observadores desde la Tierra, nos parece que el tiempo medido por los relojes de los satélites transcurre más lento. Los relojes se “atrasan” 7 microsegundos / día.
- Teoría de la **relatividad general**
 - La curvatura espacio-tiempo depende del campo gravitacional al que es sometido. Los relojes se “adelantan” 45 microsegundos / día
- Un adelanto diario de 38 microsegundos significaría un error acumulado de 10 km por cada día.



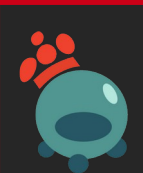
Relojes Lógicos

- Indican el **orden** en que suceden ciertos eventos, no el instante real en el que suceden.
- Muy útiles en sistemas distribuidos donde la **noción** de si un evento ha sucedido antes que otro es importante.
- La **sincronización** es perfecta pero tiene sus limitaciones.
- Se aplica a:
 - Mensajes periódicos de sincronización
 - Ordenación de eventos
 - Detección de violaciones de causalidad.
- Relojes de Lamport y Vectoriales



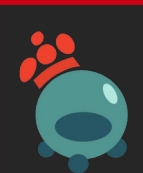
Relojes de Lamport

- Leslie Lamport (1978) definió un algoritmo simple para asegurar la ordenación de eventos en un sistema distribuido. (la sincronización no necesita ser en tiempos absolutos).
- Se basan en la relación “**ocurre antes**” $a \rightarrow b$
- Todos los nodos están de acuerdo en que $a \rightarrow b$
- Se puede observar de forma directa:
 - Siendo a y b eventos del mismo nodo y a ocurren antes que b
 - $a \rightarrow b$
 - Si a es un evento de envío de un proceso P_1 y b es el evento de recepción de un proceso P_2
 - $a \rightarrow b$

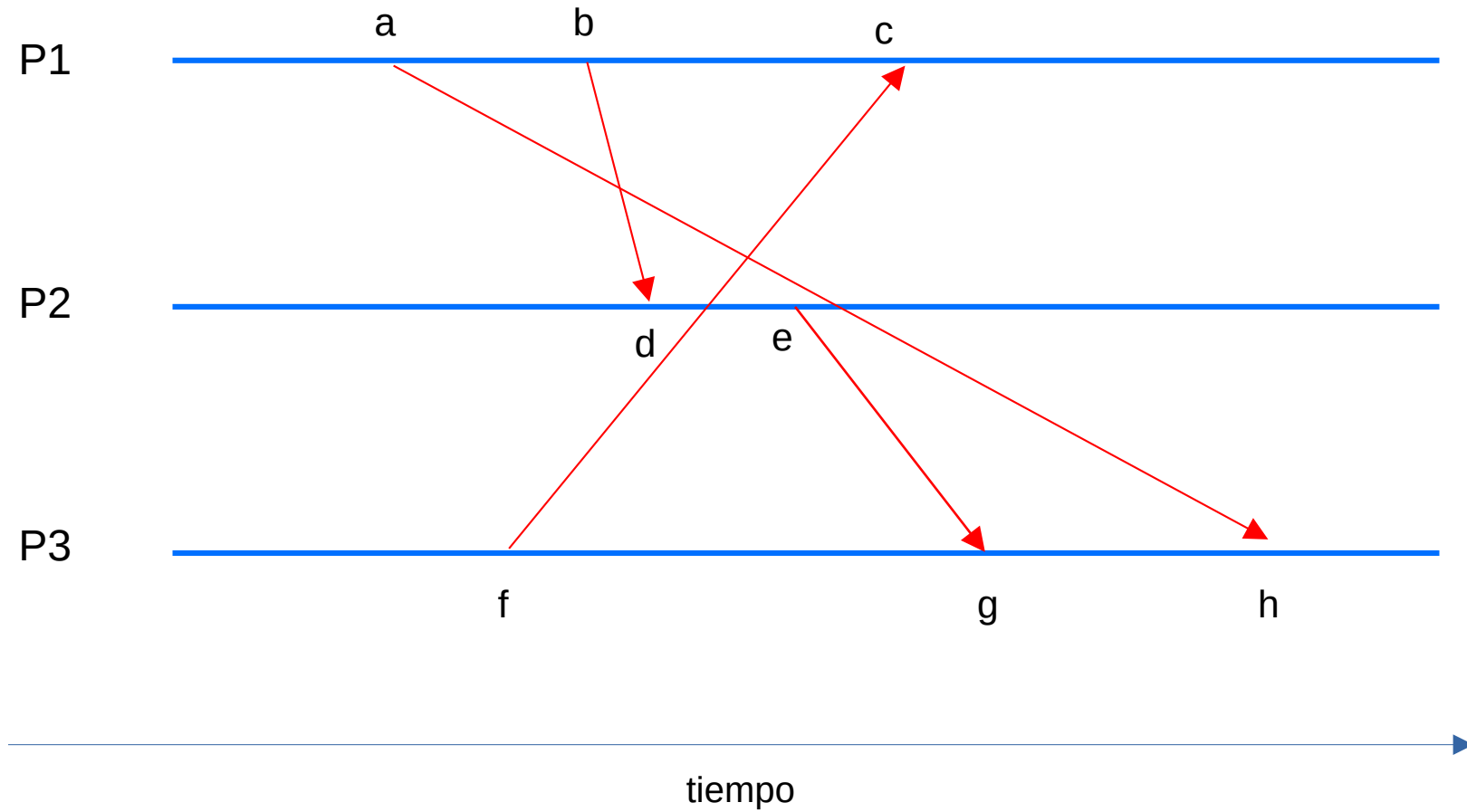


Relojes de Lamport

- La relación “**ocurre antes**” $a \rightarrow b$
 - Es una relación transitiva
 - Si $a \rightarrow b$ y $b \rightarrow c$, entonces $a \rightarrow c$
- Dos eventos x e y son **concurrentes** si no se puede decir con seguridad qué evento ocurre antes
 - $x \parallel y$
- Establecen un orden parcial entre los eventos de un sistema distribuido, ya que no todos los eventos están relacionados entre si.



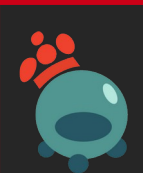
Relojes de Lamport

 $a \rightarrow b$ $b \rightarrow d$ $a \rightarrow e$ $f \parallel e$ $c \parallel e$

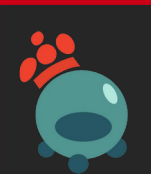
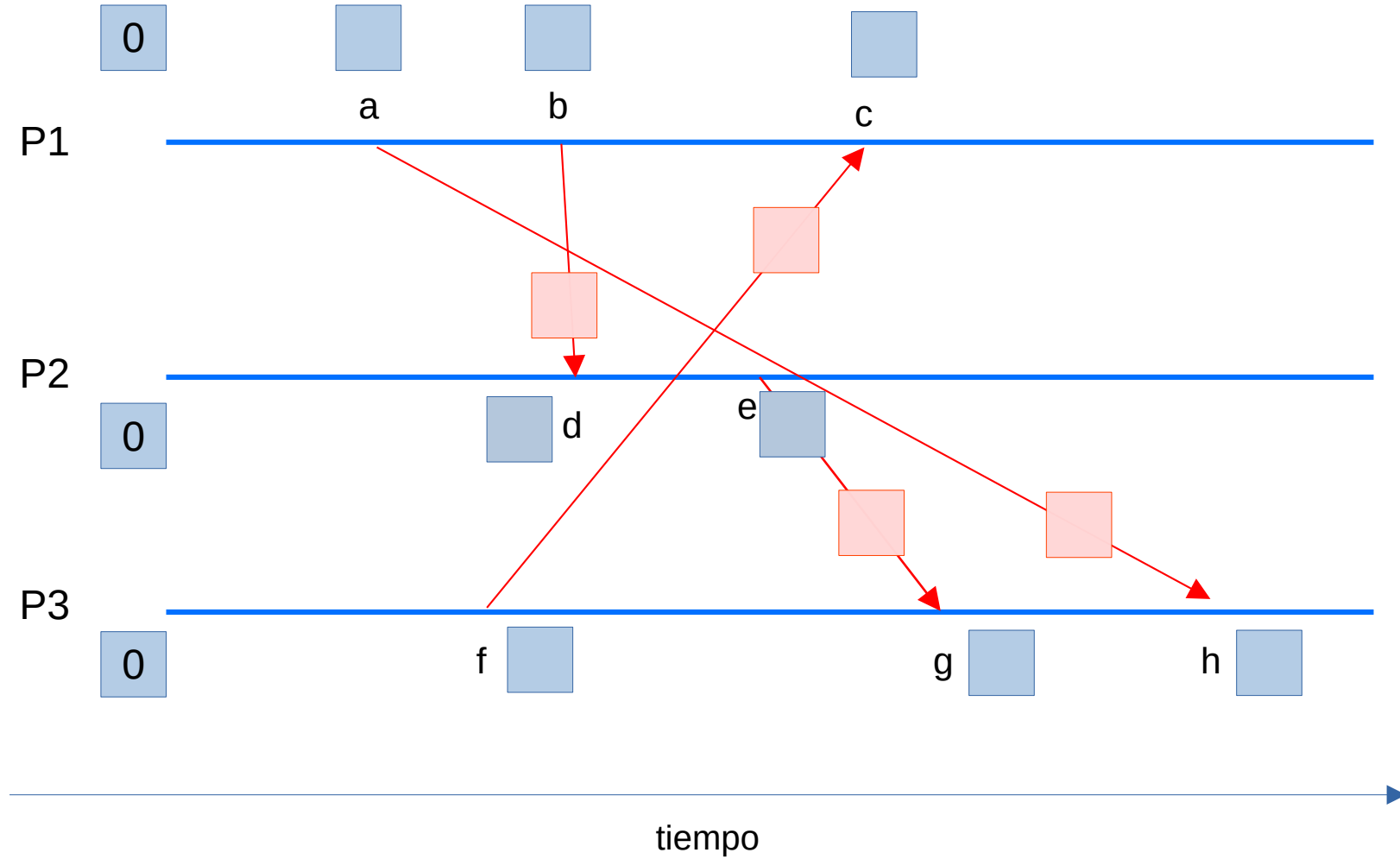
Relojes de Lamport

- Cada proceso P mantiene una variable entera LC_p que se inicializa a 0.
- Cuando un proceso P genera un envío, se actualiza el contador $LC_p = LC_p + 1$
- Cuando un proceso envía un mensaje incluye el valor de su reloj.
- Cuando un proceso Q recibe un mensaje con un tiempo determinado (t), actualiza su reloj local en base a

$$LC_q = \max (LC_q, t) + 1$$

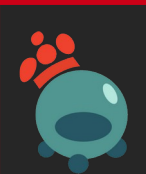


Relojes de Lamport



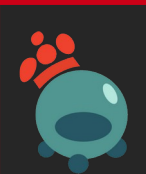
Relojes de Lamport

- Si $a \rightarrow b$, el algoritmo garantiza que
 - $C(a) < C(b)$
- Pero si $C(a) < C(b)$ no se puede asegurar que:
 - $a \rightarrow b$
 - $a \parallel b$
- En el ejemplo anterior $C(a) < C(f)$, pero no sabemos si son eventos concurrentes o $a \rightarrow f$.



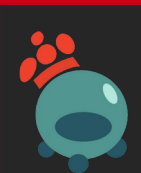
Relojes de Lamport

¿Podríamos reimplementar el sistema GPS utilizando Relojes lógicos de Lamport?



Relojes Vectoriales

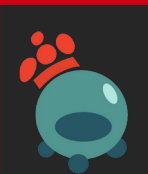
- Asocian un valor **vectorial** a cada evento que se producen en un sistema distribuido
- Los **eventos** suelen ser considerados como el envío y/o recepción de mensajes.
- En un sistema de N nodos, los relojes vectoriales guardan un **vector de N** relojes lógicos en cada nodo.
- Se utilizan para determinar causalidad en sistemas distribuidos
- Soluciona algunas de las limitaciones de los relojes lógicos de Lamport



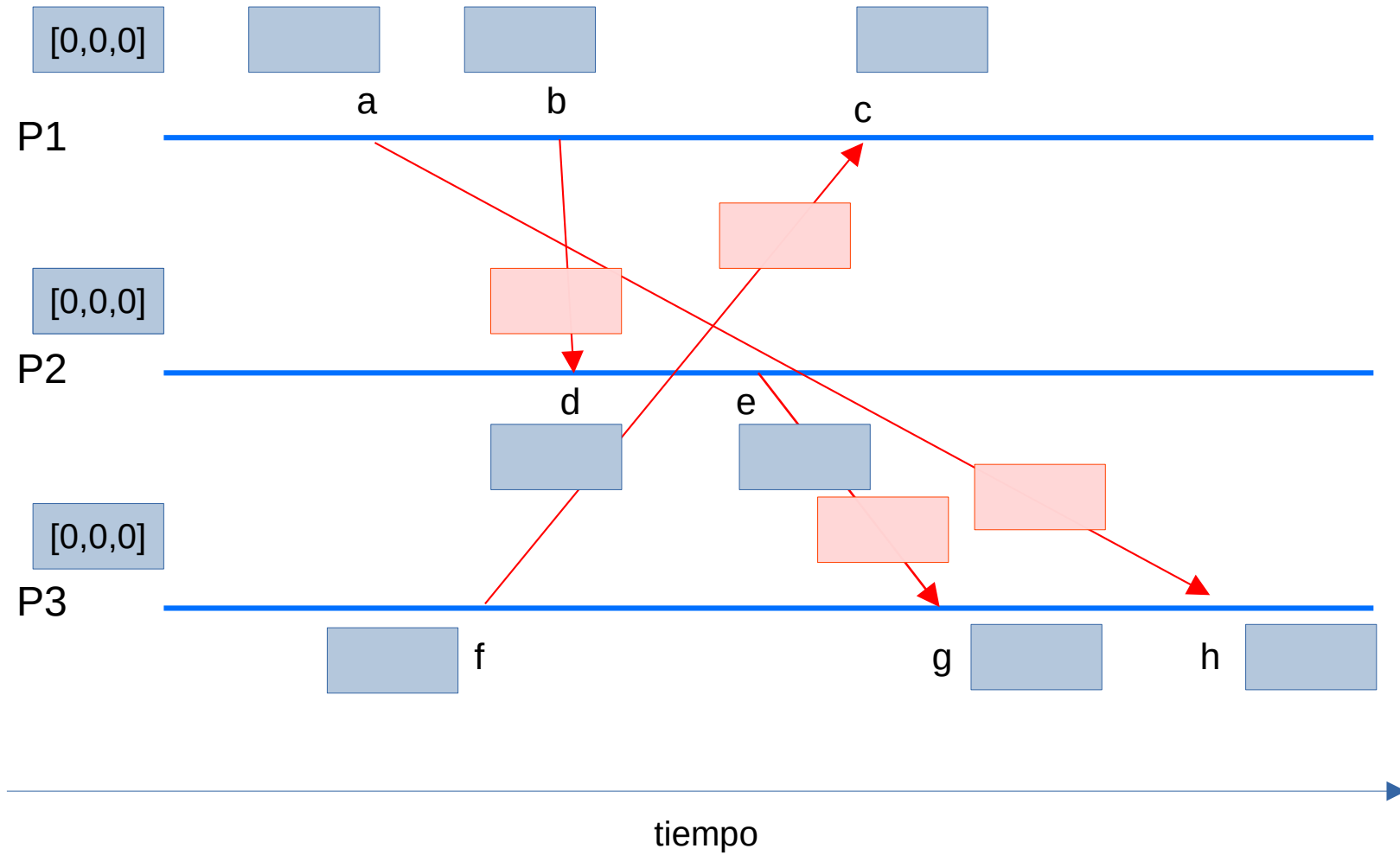
Relojes Vectoriales

- Cada nodo dispone de un reloj vectorial de tamaño N , siendo N el numero de nodos del sistema
- Todos los relojes se inicializan a 0, $V_p=[0,0,0,\dots,N]$
- Un nodo **p** incrementa en 1 $V_p[p]$ antes de que ocurra un evento (envío o recepción)
- Un mensaje enviado lleva asociado su reloj vectorial **$V_m[p]$** .
- Al recibir un nodo **p** un mensaje, incrementa en 1 $V_p[p]$ y

$$\forall i, 1 \leq i \leq N, V_p[i] = \max(V_p[i], V_m[i])$$

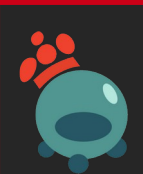


Relojes Vectoriales



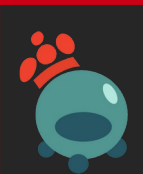
Relojes Vectoriales

- $V(a) < V(b)$ si
 - Cada una de las componentes de $V(a)$ es menor o igual que las componentes respectivas de $V(b)$
 - Y además una de las componentes tiene que ser estrictamente menor.
- Si $a \rightarrow b$ entonces podemos asegurar $V(a) < V(b)$
- Si $V(a) < V(b)$ entonces podemos asegurar $a \rightarrow b$
- Si no se puede demostrar $V(a) < V(b)$ ni $V(b) < V(a)$ entonces $a || b$

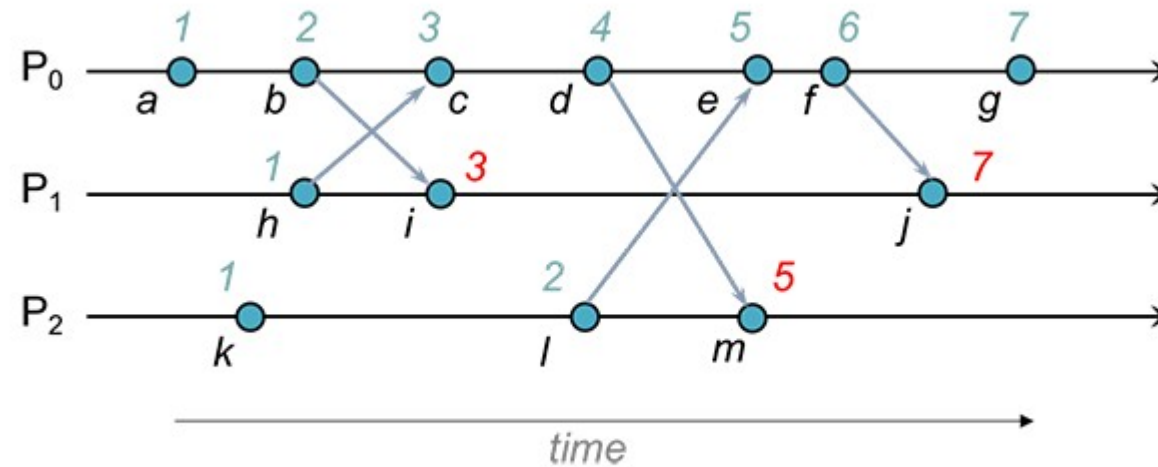


Relojes Vectoriales

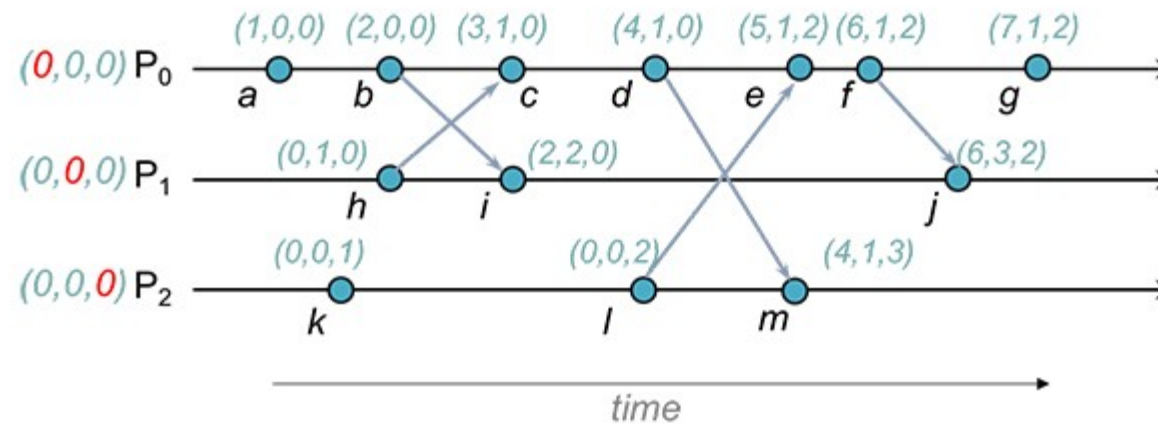
- Presenta algunas limitaciones
- El tamaño del mensaje se incrementa debido al vector de relojes.
- Se puede optimizar únicamente mandando valores que han cambiado.



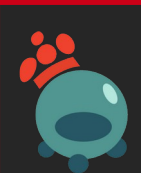
Lamport vs Vectoriales



Lamport Clock Assignment



Vector Clock Assignment



Usos

voldemort/
voldemort

An open source clone of Amazon's Dynamo.

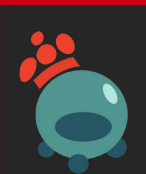


amazon
DynamoDB



Bibliografía

- Sistemas operativos distribuidos (Andrew S. Tanenbaum)
- Time, clocks, and the ordering of events in a distributed system (Leslie Lamport)
 - <https://dl.acm.org/doi/abs/10.1145/3335772.3335934>
- Time Synchronization in modular collaborative robots
 - <https://arxiv.org/pdf/1809.07295.pdf>





Escuela de Ingeniería
de Fuenlabrada



RoboticsLabURJC
Programming Robot Intelligence

