



Introducción a la práctica: conf



Prácticas introductorias

- ◆ **Práctica 0: probar herramientas básicas para el desarrollo de las siguientes prácticas**
 - ❖ Sobre todo, LEER transparencias de programación, y código
 - ❖ Familiarizarse con el funcionamiento y programación de la tarjeta de sonido
- ◆ **Práctica 1: pruebas más exhaustivas de funcionamiento de la tarjeta de sonido**
 - ❖ Ejemplo de cómo analizar el comportamiento de un dispositivo o herramienta de programación

Ni Práctica 0 ni Práctica 1 generan ningún resultado evaluable



Aplicación conf

Aplicación de audioconferencia

- ◆ Tarjeta de sonido en modo *duplex*
- ◆ Dos *modos* en la aplicación para permitir el inicio de comunicación
 - ❖ Modo *first* (*primero* – en ser arrancado desde la línea de comandos), espera a recibir datos de algún nodo (no tiene por qué conocer su dirección).
 - ✓ Cuando recibe el primer dato, “aprende” qué dirección IP tiene su interlocutor, y le empieza a enviar a datos
 - ❖ Modo *second* (*segundo*), directamente datos a una dirección especificada en la línea de comandos
- ◆ Podría ser utilizada en entorno multicast
 - ❖ Aunque cada aplicación está preparada para tener sólo un vecino
 - ❖ ¡No se puede utilizar TCP!

Aplicación conf

◆ Dos formas de arrancar first y second

◆ Comunicación unicast

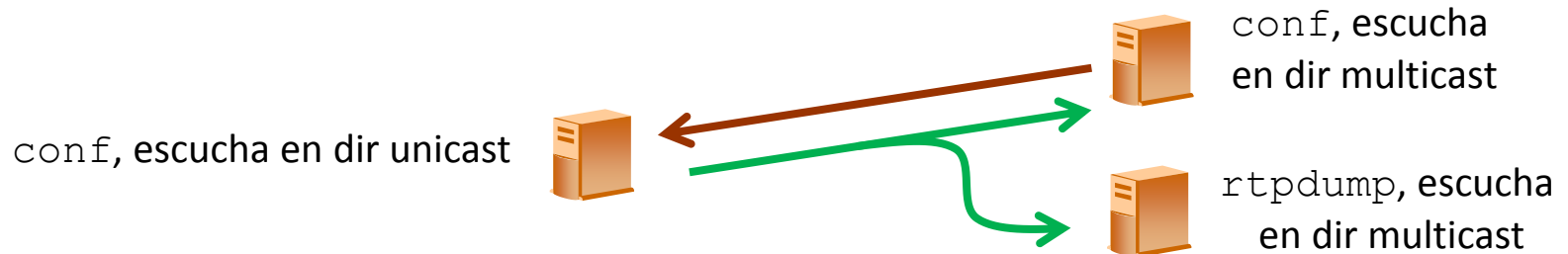
```
equipo1> ./conf first
```

```
equipo2> ./conf second 163.117...
```

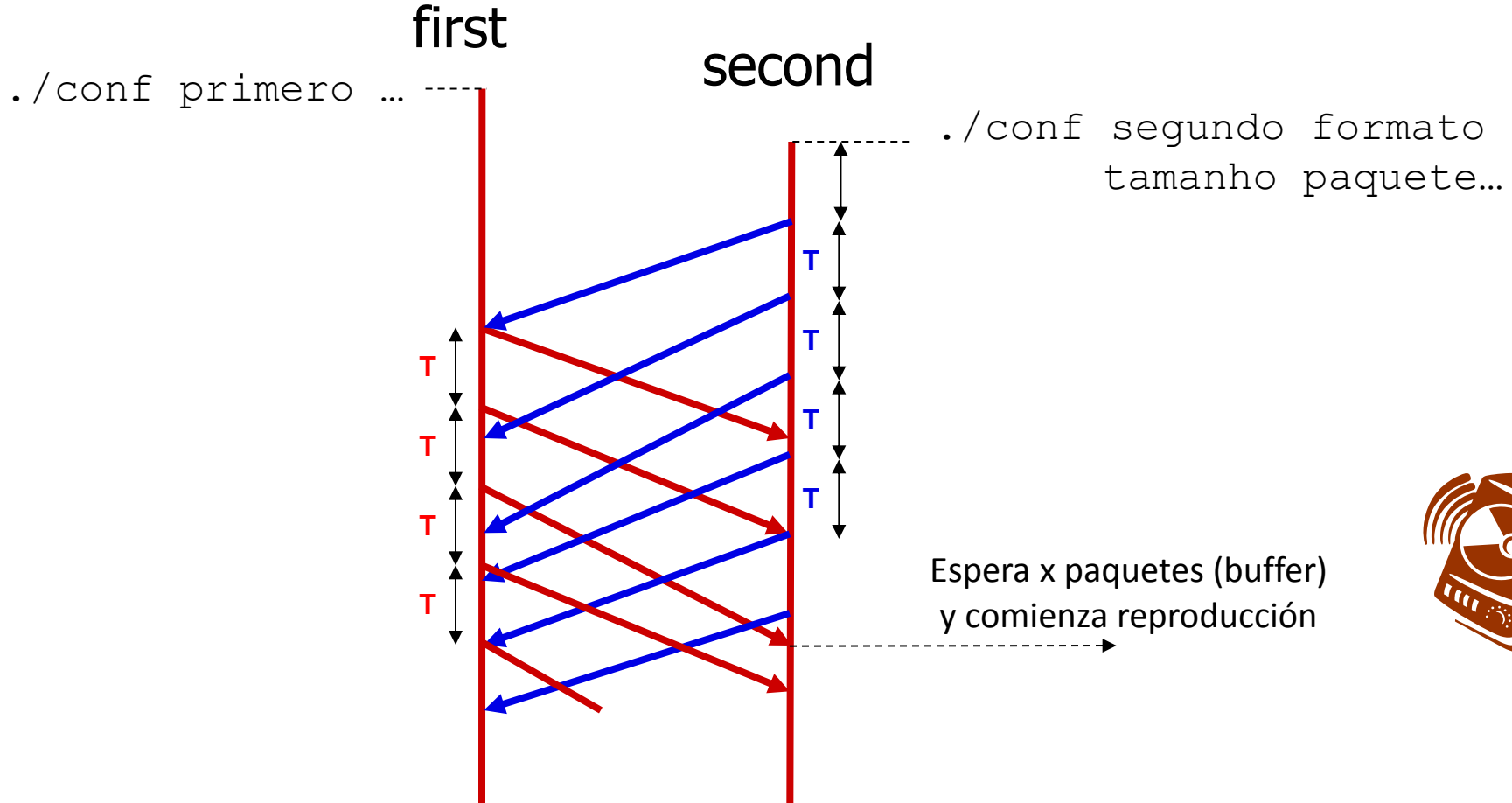
◆ First escucha en multicast

```
equipo1> ./conf first -m227.1...
```

```
equipo2> ./conf second 227.1...
```



Aplicación conf



◆ **T viene determinado por el formato y el tamaño del paquete**

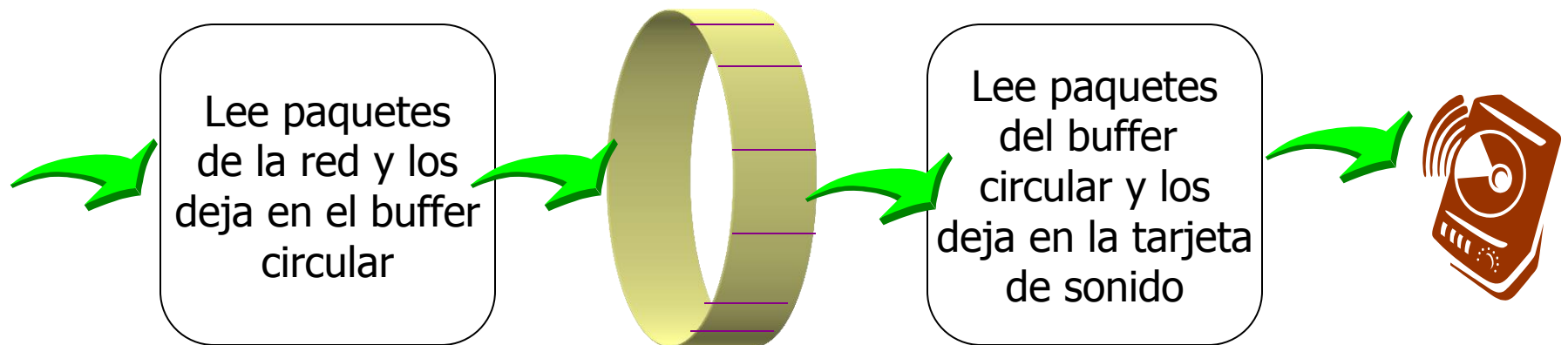
Aplicación conf

◆ Es necesario desacoplar

1. Recepción de paquetes de audio
2. Reproducción de los paquetes de audio recibidos
3. Grabación y envío de audio local
4. Generación y envío de mensajes RTCP
5. Recepción y tratamiento de mensajes RTCP

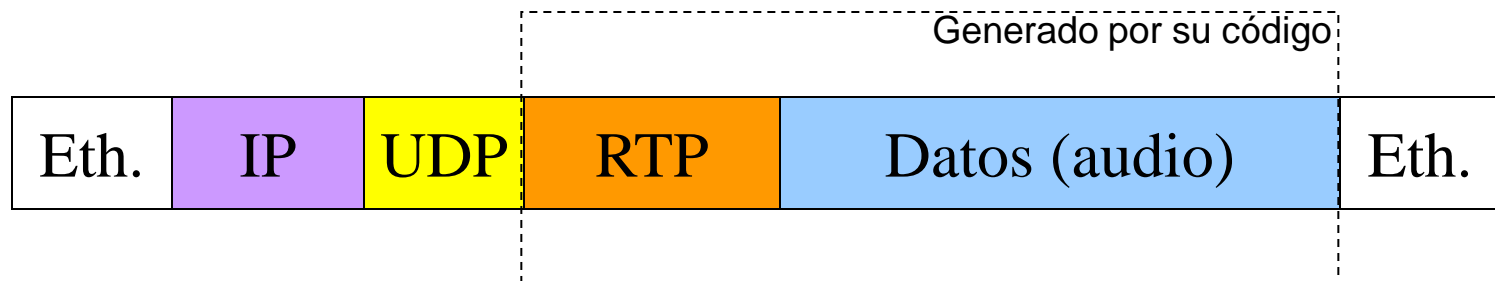
Cualquier retardo en los 3 primeros procesos resulta en un retardo que puede hacer que un contenido no se pueda reproducir en su momento

◆ Para desacoplar recepción de paquetes y reproducción (1 y 2) se utiliza **select + buffer circular**



Aplicación conf

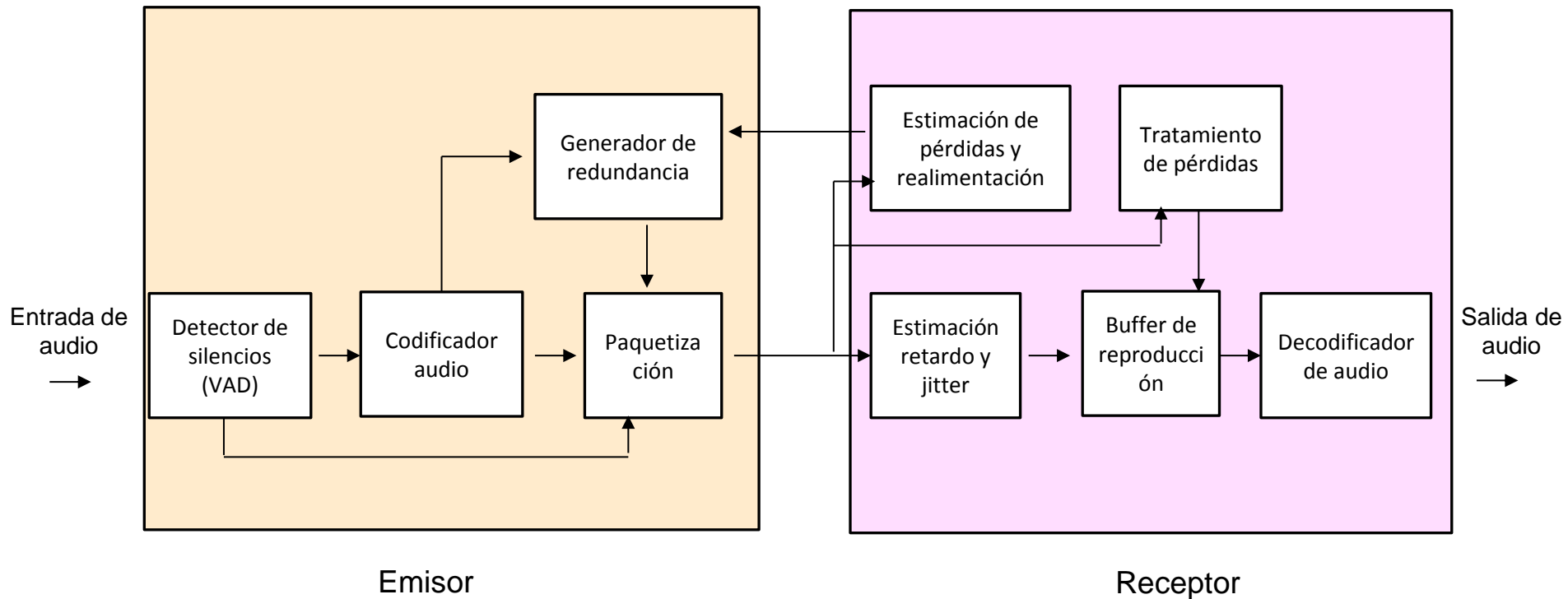
- ◆ **Su aplicación se debe encargar de construir la cabecera RTP en el envío de paquetes**
 - ❖ **Rellena estructura en C**
 - ❖ **Lee los datos de la tarjeta de sonido, y los deja a continuación**
 - ❖ **El mensaje resultante es empaquetado con UDP y IP**
 - ✓ Esta tarea la realiza el SO, a través de la interfaz de sockets (haciendo `sendto`)
 - ✓ Puerto estándar para RTP=5004; puerto estándar para RTCP=5005
 - ❖ **La tarjeta de red añade por hardware la cabecera y la cola para Ethernet**



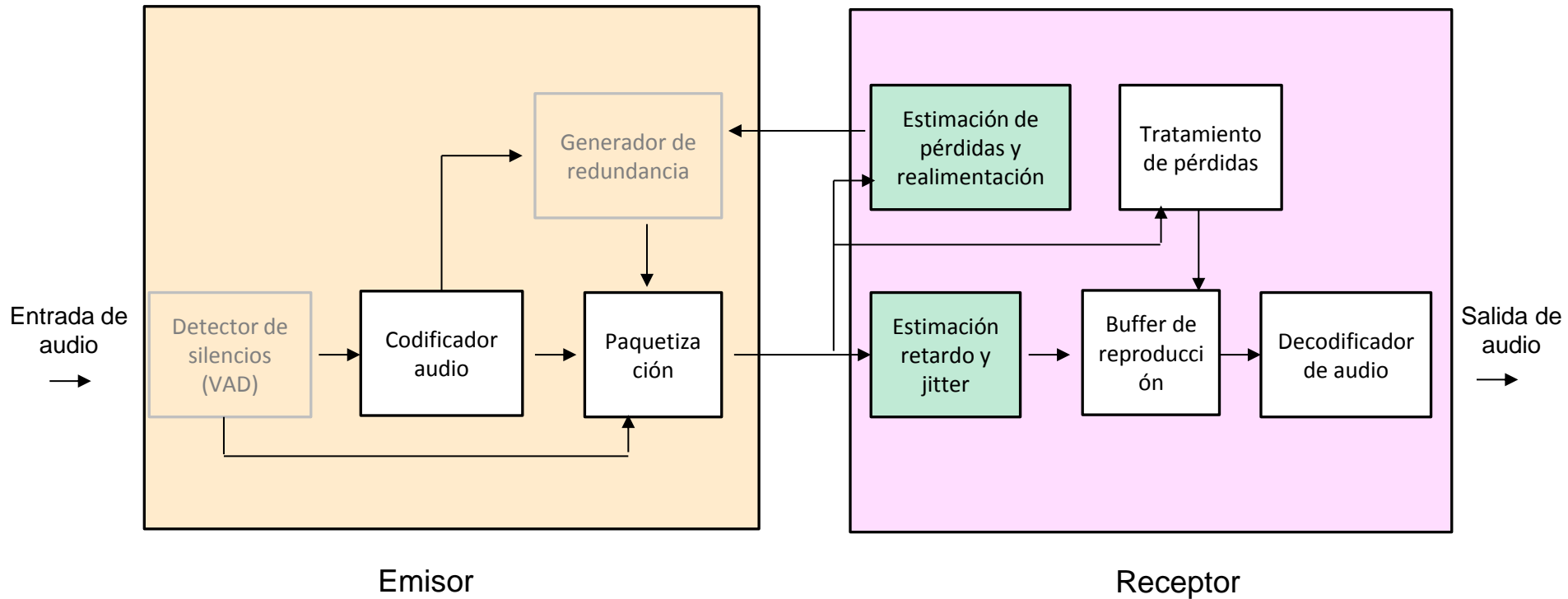
Aplicación conf

- ◆ **Optimice la reproducción configurando los *fragmentos* de la tarjeta de sonido**
 - ❖ De esta forma se evitan añadir retardos innecesarios
- ◆ **Implemente comportamiento razonable (para una aplicación interactiva) frente retrasos o pérdidas de paquetes**
 - ❖ Si un paquete no ha llegado a tiempo (llega tarde o se pierde), se reproduce el último paquete recibido
 - ✓ Detectamos que se acaban los datos preguntando a la tarjeta de sonido, SNDCTL_DSP_GETODELAY
 - ❖ Si se reciben de la red paquetes para los que se ha pasado su tiempo de reproducción, se descartan para no alargar la duración de la representación
 - ✓ Utilice RTP para numerar los paquetes e insertar marcas de tiempo
 - ❖ Si llegan paquetes desordenados, suponemos que se han perdido

Arquitectura de un sistema de VoIP



Arquitectura de conf



Comparación de conf y Skype

(de acuerdo a datos de Analysis and Evaluation of the Skype and Google-Talk VoIP Systems, B. Sat, B. W. Wah)

◆ Trama de audio

- ❖ conf permite cualquier tamaño de trama que sea potencia de 2, controlado desde la línea de comandos
- ❖ Skype utiliza tamaños de trama de 20 o 30 ms

◆ Codec:

- ❖ conf utiliza codificación PCM sin compresión, 8 KHz y 8 bits/muestra, 44100 KHz y 16 bits/muestra
- ❖ Skype utiliza codec Internet Low Bit-rate Codec (iLBC), con el que codifica muestra de 8 KHz con 16 bits/muestra
 - ✓ Codec especializado en habla humana
 - ✓ Entre 15.2 Kbps (trama 20 ms) y 13.3 Kbps (trama 30 ms)
 - Mucho más eficaz que conf

◆ Detección de silencio

- ❖ conf : no
- ❖ Skype: sí

◆ Paquetización

- ❖ conf manda una trama en un paquete UDP
- ❖ Skype manda 2 tramas en un paquete UDP
 - ✓ También puede utilizar TCP, si tiene problemas para atravesar Firewalls

◆ Buffer en recepción

- ❖ conf : configurable en línea de comandos
- ❖ Skype: 60 ms



Comparación de conf y Skype

◆ Redundancia de datos

- ❖ conf no tiene
- ❖ Skype sí permite redundancia para recuperar paquetes perdidos

◆ Ante pérdidas de paquetes

- ❖ conf repite paquete anterior
- ❖ Skype recupera 2 primeras tramas, y luego inserta silencio
 - ✓ Si pasan 6 segundos, intenta comunicación por TCP

◆ Ante paquetes desordenados

- ❖ conf detecta pérdida, pero no reproduce los paquetes desordenados
- ❖ Skype los reproduce como se reciben (no detecta desorden)

◆ Información de receptor a emisor (feedback)

- ❖ conf : opcional, con RTCP
 - ✓ No está definido para qué utilizarlo
- ❖ Skype: manda 4 u 8 bytes cada 2 segundos
 - ✓ Se utiliza para modificar la cantidad de redundancia en la emisión, para tolerar 2 paquetes perdidos en vez de 1
 - En este caso, el retardo da un salto adicional de 60 ms