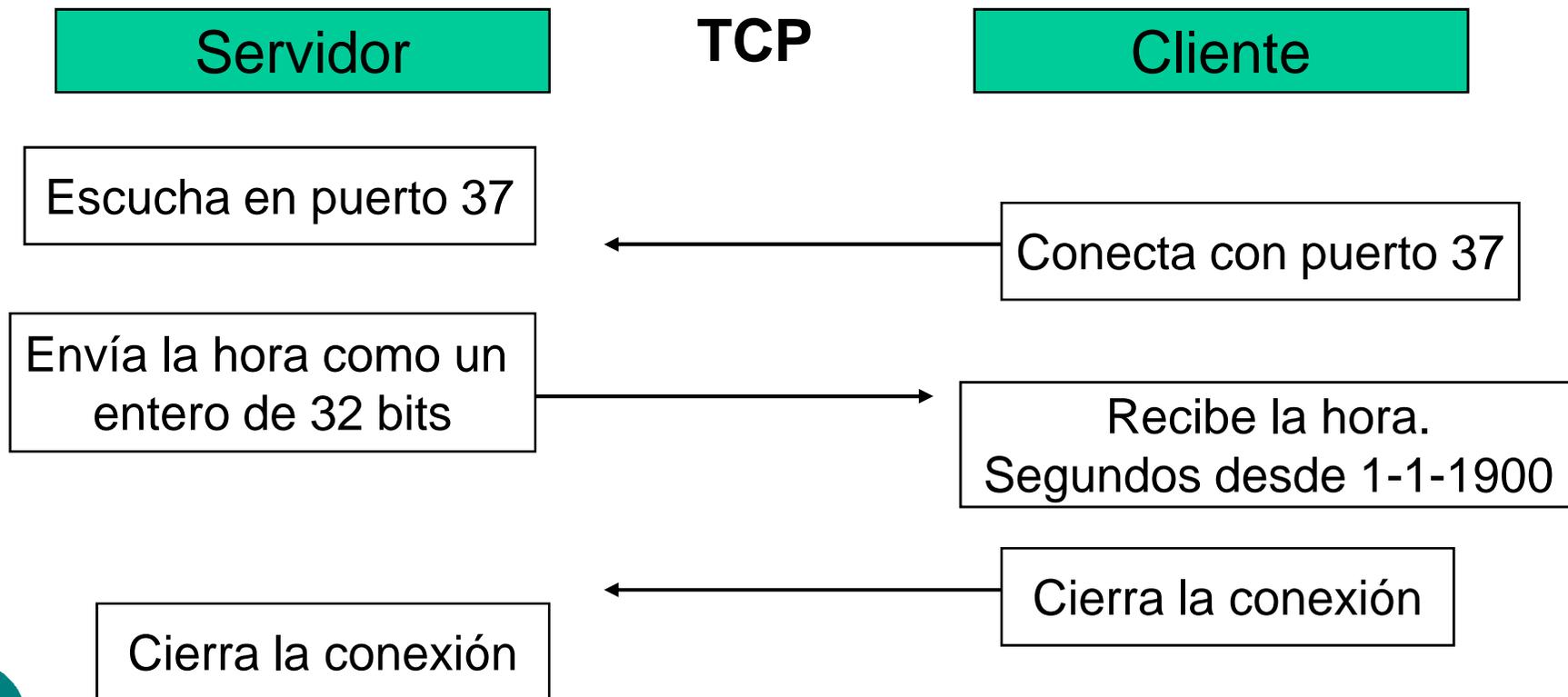


Ejercicio 1: Protocolo Hora

Implementar un programa que muestre la fecha de diversas máquinas.

Utilizar el protocolo de hora: <http://www.faqs.org/rfcs/rfc868.html>.

Esquema de funcionamiento:



Ejercicio 1: Protocolo Hora

Para convertir la respuesta del protocolo de hora a fecha actual:

Función `ctime`.

Prototipo:

```
#include <time.h>

char *ctime (const time_t *clock)
```

Recibe: Un entero largo de 32 bits con el número de segundos pasados desde el 1 de Enero de 1970.

Devuelve: Un puntero a una cadena de 26 caracteres con la fecha equivalente en formato ASCII.

Dato: Segundos entre 1-1-1900 y 1-1-1970 = **2.208.988.800**



Ejercicio 2: Servicio Eco

El servidor escucha en un puerto TCP. Cuando se establece una conexión, el servidor retorna todo lo que recibe. Igual para UDP.

Descripción oficial en: <http://www.faqs.org/rfcs/rfc862.html>.

Cliente de prueba: `telnet`

Uso:

```
telnet maquina puerto
```

Ejemplo:

```
telnet sirio.atc.uniovi.es 7
```

(Para terminar la sesión pulsar <Ctrl-]>)



Ejercicio 2: Servicio Eco

Detalles de Implementación:

El servidor debe recibir el número de puerto por línea de comandos. Ejemplo: `servecho 17323`

Problema: leer un número de datos desconocido. Soluciones:

1. Buffer mínimo (1 byte):

```
read(sock, &buffer, 1);  
write(sock, &buffer, 1);
```

2. Control de lecturas:

```
leido=read(sock, &buffer, sizeof (buffer));  
escrito=write(sock, &buffer, leido);
```



Nota sobre compilación en Solaris

En Solaris (SunOS) es necesario especificar qué bibliotecas contienen la API de sockets:

```
gcc -o ejecutable fuente.c -lnsl -lsocket
```

