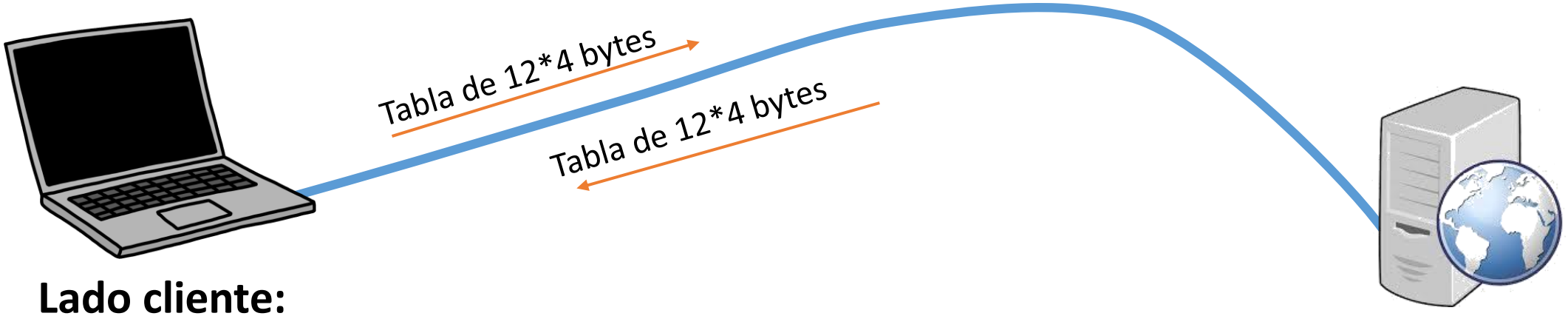
EJEMPLO COMUNICACIÓN UDP ENTRE PC y servidor NTP (tiempo "exacto")



"hora.rediris.es"
130.206.3.166
(otros servidores en https://wiki.bandaancha.st/Lista_de_servidores_NTP_stratum_1_en_Espa%C3%B1a)

Lado cliente:

- Inicializa socket UDP (tipo datagram) → id. ***sock***
- Enlaza (bind) ***sock*** con dirección IP y puerto local (cualesquiera)
- Envía paquete a dirección remota (puerto 123 del servicor) siguiendo protocolo NTP :
  - Tabla de 48 bytes (ó 12 int32_t)
  - Primer byte = 8, resto a 0
- Espera respuesta
- Decodifica respuesta:
  - Comprueba longitud 48 bytes
  - Extrae fecha/hora siguiendo protocolo NTP (la fecha/hora es un time_t que se encuentra en los bytes 40 a 47)
  - Convierte a struct tm para escribir
- Cierra ***sock***

Lado Servidor:

- Inicializa socket UDP (tipo dgram) → id. ***server***
- Enlaza (bind) ***server*** con dirección IP 130.206.3.166 y puerto reservado para NTP (123)
- Espera y lee mensaje recibido en ***server*** según protocolo NTP
- Codifica y envía respuesta siguiendo el mismo protocolo

PROGRAMA PRINCIPAL PC (Windows) 1 de 1

```
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>
#include <winsock2.h>
#include <time.h>

int main()
{
// 1) Preprara socket cliente
    WSADATA wsaData;
    WORD wVersionRequerida = MAKEWORD (2, 2);
    if (WSAStartup (wVersionRequerida, &wsaData)!=0) {
        printf("Error inicializando sockets Windows\n");
        return -1;
    }

    // Variables para gestionar comunicación de socket cliente
    SOCKET sock; // Identif de socket (equivalente a FILE* para archivos)
    struct sockaddr_in local,remoto; // Direcciones de socket local y remoto

    // Obtener identificador para un socket UDP (datagram)
    sock=socket(AF_INET,SOCK_DGRAM,0);
    if (sock==INVALID_SOCKET)  {
        printf("No se puede crear socket. Programa terminado\n");
        return -2;
    }

    // Preparar estructura con datos de IP/puerto local y llamar a bind() para
que enlace el socket local
    local.sin_family=AF_INET;
    local.sin_addr.S_un.S_addr=0;   // 0 = cualquier dirección IP
    local.sin_port=htons(0);        // 0 = cualquier puerto
    memset(local.sin_zero,0,8*sizeof(char));
    if (bind(sock,(struct sockaddr*) &local,sizeof(struct sockaddr_in))!=0) {
        printf("No se puede asignar dirección local. Programa terminado\n");
            return -2;
    }

    // Esperamos a que el usuario pulse INTRO
    printf("Pulse RETURN para conectar con servidor NTP: ");
    getchar();
```

```
    // Conectar con el servidor deseado: ver lista en
    // https://wiki.bandaancha.st/Lista_de_servidores_NTP_stratum_1_en_Espa%C3%B1a
    // 2.1) Preparar estructura con datos de IP/puerto remoto a partir de su nombre
    struct hostent *remoteHost;
    remoteHost = gethostbyname("hora.rediris.es");
    if (remoteHost==NULL) {
            return -3;
    }
    if (remoteHost->h_addrtype != AF_INET) {
            return -3;
    }
    remoto.sin_family=AF_INET;
    remoto.sin_addr.S_un.S_addr=*((u_long*) remoteHost->h_addr_list[0]);
    remoto.sin_port=htons(123);      // htons() -> Convierte a network order
    memset(remoto.sin_zero,0,8*sizeof(char));

    // 3) Preparar solicitud al servidor NTP
    // (https://www.eecis.udel.edu/%7emills/database/brief/arch/arch.pdf)
    uint32_t ntp_send[12],ntp_recv[12];
    int n_bytes=12*sizeof(uint32_t);
    memset(ntp_send,0, n_bytes);
    ntp_send[0]=0x08;

    int nSent=sendto(sock,(const char*) ntp_send, n_bytes , 0 ,
                    (struct sockaddr*) &remoto,sizeof(struct sockaddr_in));

    // 4) Recibir respuesta y decodificar según protocolo NTP
    // (https://www.eecis.udel.edu/%7emills/database/brief/arch/arch.pdf)

    struct sockaddr_in from;
    int fromLen=sizeof(struct sockaddr_in);
    int nRecv=recvfrom(sock,(char*) ntp_recv,n_bytes,0,
                       (struct sockaddr*) &from,&fromLen);

    if (nRecv==12*sizeof(uint32_t)) {
       int pos=10;
       time_t time=ntohl(*((uint64_t*) (ntp_recv+pos)));
       time -= 2208988800U;
       struct tm* tm = localtime(&time);
       printf("Fecha y hora actual: %d/%d/%d , %d:%d:%d\n",
                 tm->tm_mday,tm->tm_mon+1,tm->tm_year+1900,
                 tm->tm_hour,tm->tm_min,tm->tm_sec);
    }

    return 0;
}
```