The data tranfer consists of exchange of packets initialted by the client, the server responds to each client packet, all the retransmision work is done by the client, the server just tries to respond to each client packet at its best.

The packet consistes of a header that is always present, and a data part. Data part is present in some packets, the length of the data part is determined by the packet length minus the size of the header.

```
#define MAX_PACKET_DATA 1024

typedef struct
{
	unsigned int packettype;///packet type
	unsigned int seqnumber;
	long long fileoffset;
	long long chunksize;
}UDP_PACKET_HEADER;


typedef struct
{
	UDP_PACKET_HEADER header;
	char data[MAX_PACKET_DATA];

}UDP_PACKET;
```

Each request of the client is responded by the erver using the same seqnumber. This is used to guard against reception delayed old packets – each received packet that doesnt have the expected seqnumber is discadred... Indeed seq numbers are generated at random...

Packet type may be :
0 – error message of server
1 – file size requestof client, file name is contained as data
2 – file size response of server, size is returned in chunksize field
3 – read file data request– file offset and chunk size determine the data to be read. To be the packet fully descriptive the filename is send as data.
4 – read file data response – server responds witht data from the file, fileoffset is the start offset, and chunk size is the size.

The communication is as follows.

To get a file size the client sends a **packettype** 1 containing the filename as **data**. Other fields are not important.
The server responds with **packettype** 2, filling in the **chunksize**. If the output size cannot be determined by 'stat' the server responds by **packettype** 0 and a string 'stat error'

To transfer data the client sends a **packettype** 3 containing the filename as **data**, the **fileoffset** and the **chunksize**.

If the file size cannot be determined by 'stat' the server responds by **packettype** 0 and **TextString** 'stat error', if the file cannot be openned the server returns **packettype** 0 and **data** 'fopen error' and it the startposition cannot be set server returns **packettype** 0 and **data** 'fseek error'. Otherwise the server reads the data and returns it using packet type 4.