*Protocolo de aplicação*

O protoloco de aplicação implementado tem como aspetos principais:

* O envio dos pacotes de controlo START e END. Estes contêm o nome e o tamanho do ficheiro a ser enviado;
* A divisão do ficheiro em fragmentos quando se trata do emissor e a concatenação dos fragmentos recebidos, quando se trata do recetor;
* Encapsular cada fragmento de dados com um header contendo o número de sequência do pacote (módulo 255) e o tamanho do fragmento;
* Leitura do ficheiro a enviar, quando se trata do emissor, e criação do ficheiro, quando se trata do recetor.

Estas funcionalidades foram implementadas usando as seguintes funções:

unsigned char \*controlPackageI (unsigned char state, off\_t sizeFile, unsigned char \*filename, int sizeOfFilename, int\* sizeControlPackageI);

Esta função retorna um pacote START ou END. Recebendo como argumentos um caracter “state” para identificar se o pacote pretendido é START ou END, o nome do ficheiro e o tamanho do ficheiro. Este pacote será enviado usando a função LLWRITE, pertencente ao protocolo de ligação de dados.

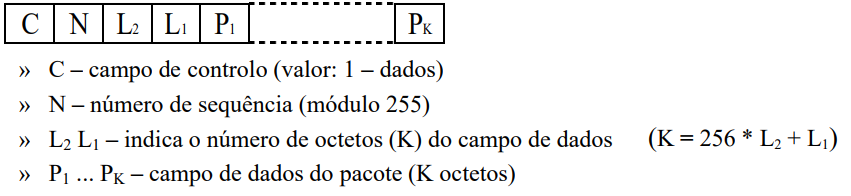
unsigned char \*splitMessage (unsigned char \*mensagem, off\_t \*indice, int \*sizePacket, off\_t sizeFile);

Esta função recebe como argumentos o conteúdo do ficheiro, o índice do primeiro elemento pertencente ao pacote, o tamanho do pacote e o tamanho do ficheiro.

Esta função retorna um pacote de tamanho sizePacket que poderá ser reduzido dentro desta função se não existir sizePacket elementos desde o índice até ao final do conteúdo do ficheiro.

unsigned char \*headerAL (unsigned char \*mensagem, off\_t sizeFile, int \*sizePacket);

Esta função retorna a mensagem a ser enviada concatenando o header com o pacote recebido. A mensagem vai então ficar da seguinte forma:



O header é constituído pelos caracteres: C, N, L2 e L1 e o pacote recebido são os caracteres de P1 a PK.

unsigned char \*openReadFile (unsigned char \*fileName off\_t \*sizeFile);

Esta função retorna o conteúdo do ficheiro a ser enviado e, como argumento, o tamanho do ficheiro. Recebe como argumento o nome do ficheiro que é escrito aquando da chamada do programa.

void createFile (unsigned char \*mensagem, off\_t \*sizeFile, unsigned char \*filename);

Esta função é responsável pela criação do ficheiro, usando o conteúdo recebido através dos pacotes de dados. O ficheiro criado tem o mesmo nome do ficheiro transmitido, obtido através da trama START.