Cloud Drive Protocol

Inspired by Google Drive (GDrive), we developed a Cloud Drive that allows the user to have his/her files automatically backed up on the cloud and synchronized across several devices.

* **Download**

If the client wants to download a file, then the header will be as the following:

download[one space][file name][Line Feed]

Upon receiving this header, the server searches for the specified file.

If the file is not found, then the server shall reply with a header as the following:

NOT[one space]FOUND[Line Feed]

If the file is found, then the server shall reply

with a header as the following:

OK[one space][file size][Line Feed]

followed by the bytes of the file

* **Upload**

If the client wants to upload a file, then the header will be as the following:

upload[one space][file name][one space][file size][Line Feed]

After sending the header, the client shall send the bytes of the file

* **List**

If the client wants to show the list of the available files locally(ll) or remotely(lr) :

* List on the client side (local):  
  No header needed because we show only the list of files locally without making any request to the  server.
* List on server side (remote):

In this case the header will be as following :

list[one space][Line Feed]

then the server shall reply with a header as the following:

OK[one space][File Name 1][one space][File Name 2][one space][...][File Name N][one space][END][one space][Line Feed] and if there are no files to list then the server shall reply with a header as the following:

OK[one space][END][one space][Line Feed]

Note: The current version of **list** function in this protocol doesn't support file names with spaces, an updated version that supports filenames with spaces will be available soon

* **Backup**

If the client wants to perform a backup of his local repository to  the server:

 This operation is based on the previous C/S communications (**local-list** and **upload**) : first we list all the locally stored files and then we make an upload for each local file based on the  upload function that we already have.

* **Synchronization**

If the client wants to perform a synchronization between all his devices :

 This operation is based on the previous communications (**remote-list** and **download**) :  First we make a **list** call to the server to get the list of  file names on the server, then we perform  a download operation  for each file name from the list that we got using the download function that we already have.