Simple File Upload (Client)

Description
This challenge involves creating a simple client application in Python that can upload files to a server. The client connects to the server, sends a command to upload a file, along with the file's metadata (such as name and size), followed by the file's content. The provided client code skeleton outlines the necessary steps but requires completion and thorough testing to ensure it functions correctly under various scenarios.

Input
The client should take a filename as input, retrieve the file's content from a predefined dictionary (the variable name is files), and send it to the server according to a specified protocol. The server expects a message to initiate the upload process, followed by a header with the file's name and size, and finally, the file's content itself. Please check the format of the header in the given unit test.

Output
The server responds with a confirmation message after successfully receiving the file. The client should display this confirmation to the user. Expected client outputs are as follows. Expected client output:

```
Testing connect to server ...
Connecting to localhost:65432
connect called with: call(('localhost', 65432))

Testing disconnect ...
close called with: call()

Testing attribute ...
test attribute passed: localhost is equal to localhost
test attribute passed: 65432 is equal to 65432

Testing send message ...
test attribute passed: OK is equal to OK
send called with: call(b'Hello')
recv called with: call(1024)

Testing start client ...
Connecting to localhost:65432
OK
send called with: call(b'upload 729.txt')
recv called with: call(1024)

Testing unknown command ...
Connecting to localhost:65432
Testing unknown file name ...
Connecting to localhost:65432
OK
```

```
Testing correct file name ...
Connecting to localhost:65432
OK
sendall called with: call(b'file-name: 729.txt,\r\nfile-size:
18\r\n\r\nContent of 729.txt')
```