```
"""Functions and constants that are common to both client.py and
server.py"""

import os

MIN_PORT_NUM = 1024
MAX_PORT_NUM = 64000
BLOCK_SIZE = 4096

TIMEOUT = 5.0    # Timeout in seconds

BAD_PORT_NUMBER_ERR = "ERROR port number is not in not in the range {} to {} or it i
s a bad format.".format(MIN_PORT_NUM, MAX_PORT_NUM)
COULDNT_BIND_ERR = "ERROR on binding to socket."
COULDNT_CREATE_ERR = "ERROR on createing socket"
COULDNT_CONNECT_ERR = "ERROR on connecting to socket."
SOCKET_LISTEN_ERR = "ERROR on listening to socket."
MISSING_ARG_ERR = "ERROR missing one or more command line arguments."
FILE_ALREADY_EXISTS_ERR = "ERROR the file {} already exists locally."
CANT_CONVERT_ADRESS_ERR = "ERROR nodename nor servname provided, or not known."
TIMOUT_ERR = "ERROR timeout"
INVALID_FILE_REQUEST_ERR = "ERROR invalid FileRequest"
INVALID_FILE_RESPONSE_ERR = "ERROR invalid FileResponse"
COULDNT_WRITE_FILE_ERR = "ERROR couldn't write file to disk."

LOCAL_HOST = "127.0.0.1"  # or locallhost.com


def error(message="", *sockets, exit_all=True):
    """exits with an error message."""
    for socket in sockets:
        if socket is not None:
            socket.close()
    if exit_all:
        exit(message)
    else:
        print(message)

def convert_portno_str(port_num):
    """Check that port number is a number is is in correct
    range, else call error()."""
    if isinstance(port_num, int):
        return port_num
    elif isinstance(port_num, str) and port_num.isdigit() and MIN_PORT_NUM <= int(po
rt_num) <= MAX_PORT_NUM:
        return int(port_num)
    else:
        error(BAD_PORT_NUMBER_ERR)


def file_exists_locally(file_name):
    """Returns True if file_name exists AND it can be opened locally."""
    file_exists = False
    # Test if the file can be opened
    if os.path.exists(file_name):
        # Test if the file can be opened
        try:
            infile = open(file_name)
            file_exists = True
        except IOError:
            file_exists = False
        finally:
            infile.close()

    return file_exists
```