

Establishing a p2p connection in python

Source: <http://coding.derkeiler.com/Archive/Python/comp.lang.python/2004-10/1363.html>

From: gs (gs.lists_at_gmail.com)

Date: 10/08/04

Date: 8 Oct 2004 01:26:05 -0700

Hi!

This is my first time posting to a newsgroup so please be gentle to me :)

Introduction to my problem:

I'm studying at the university and a lot of friends and I use MSN to communicate. The problem is that MSN file sending/receiving capabilities are terribly slow and has no resume option. We're also on different subnets and can not access eachothers computers through smb shares, nor can we have local ftp-servers.

My programming background:

I've been taking a few classes in JAVA, enough to get familiar with sockets, threads and the most common sorting algorithms and datastructures (binary trees, stacks etc). I'm also very comfortable programming PHP on a webdeveloper basis.

I recently started out with Python and i immediatly fell in love with it, and now I want to "master" it! :) Alas, my problem is also of educational nature.

Setup:

We're about 30 friends who want the ability to share files.

We're spread out on about 3 different translated external IPs.

I have access to a public IP that has the ability to run python (but not enough bandwidth nor space to host an FTP).

I have limited programming experience.

Proposed solution:

I thought that one approach could be to write a small server for my public server with the sole purpose of keeping track of connected users and establishing p2p connections on demand.

Problems:

1) I've written a threaded server that stores information about connected servers in a "container class". This class holds information

comp.lang.python: Establishing a p2p connection in python

about username, a unique ID, the connecting adress and the actual socket. I have not been able to find any information online on how i could go about to connect two sockets. It would seem like the socket is already "occupied" since it is connected to the server. Could one duplicate, or otherwise get a dedicated socket for sending binary data, to one connected socket from another? A few pointers and/or tips would be greatly appreciated!

2) I've also failed to find python specific information on how abouts one would do file resumming. I would think that you, somehow, use md5-checksums to check file status and somehow skip the first part of the datastream. Also, pointers and maybe an explanation in (short) pseudo-code would be much appreciated.

3) Is this too big of a project for a novice programmer such as myself? I do want to go through with this since it's a genuine problem, but i do not know if i'm ready. I'm not a "quitter" per say, but i'm afraid doing something TOO advanced in the beginning might be overwhelming :)

Thanks for reading,
gs