<https://wiki.filezilla-project.org/Network_Configuration>

**Technical background**

What distinguishes FTP from most other protocols is the use of secondary connections for file transfers. When connecting to an FTP server, you are actually making two connections. First, the so-called *control connection* is established, over which FTP commands and their replies are transferred. Then, to transfer a file or a directory listing, the client sends a particular command over the control connection to establish the *data connection*.

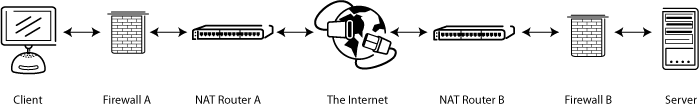
The data connection can be established two different ways, using *active mode* or *passive mode*.

*In passive mode*, which is recommended (see below), the client sends the PASV command to the server, and the server responds with an address. The client then issues a command to transfer a file or to get a directory listing, and establishes a secondary connection to the address returned by the server. *In active mode*, the client opens a socket on the local machine and tells its address to the server using the PORT command. Once the client issues a command to transfer a file or listing, the server will connect to the address provided by the client.

In both cases, the actual file or listing is then transferred over the data connection.

Generally, establishing outgoing connections requires less configuration on the routers/firewalls involved than establishing incoming connections. In passive mode, the connection is outgoing on the client side and incoming on the server side and in active mode this is reversed. Note that the only differences are in establishing a connection. Once established, the connection can be used for uploads or downloads.

A common network setup might look like this:

[](https://wiki.filezilla-project.org/File:FTP1.png)

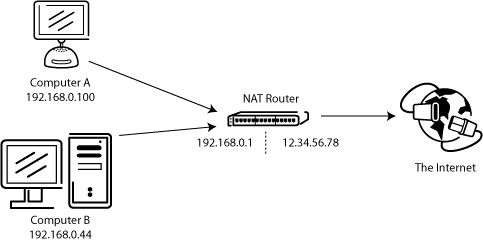
In passive mode, the router and firewall on the server side need to be configured to accept and forward incoming connections. On the client side, however, only outgoing connections need to be allowed (which will already be the case most of the time).

Analogously, in active mode, the router and firewall on the client side need to be configured to accept and forward incoming connections. Only outgoing connections have to be allowed on the server side.

Since in most cases one server provides a service for many users, it is much easier to configure the router and firewall on the server side once for passive mode than to configure the client's router/firewall for each individual client in active mode. Therefore, passive mode is recommended in most cases.

**NAT routers**

Most broadband users will have a NAT (Network Address Translation) router between their computer and the internet. This may be a standalone router device (perhaps a wireless router), or be built into a DSL or cable modem. In a NAT environment, all systems behind the NAT router form a *Local Area Network (LAN)*, and each system in the LAN has a local IP address. The NAT router itself has a local IP address as well. In addition, the NAT router also has an external IP address by which it is known to the Internet. An example might look like this:

[](https://wiki.filezilla-project.org/File:FTP2.png)

The internal IP addresses are only valid inside the LAN, since they would make little sense to a remote system. Think about a server behind a NAT router. Imagine what might happen if a client requests passive mode, but the server doesn't know the external IP address of the NAT router. If the server sends its internal address to the client, two things could happen:

* If the client is not behind a NAT, the client would abort since the address is invalid.
* If the client is behind a NAT, the address given by the server might be the same as a system in the client's own LAN.

Obviously, in both cases passive mode would be impossible.

So if a server is behind a NAT router, it needs to know the external IP address of the router in passive mode. In this case, the server sends the router's external address to the client. The client then establishes a connection to the NAT router, which in turn routes the connection to the server.