Client to Server
nc -N host.example.com 1234 < filename.in

nc -l 1234 > filename.out
ssh host.example.com cat > filename.out < filename.in

/usr/sbin/sshd
echo put filename.in filename.out | sftp host.example.com

/usr/sbin/sshd
Server to Client
nc -l 1234 < filename.out

nc -N host.example.com 1234 > filename.in
/usr/sbin/sshd

ssh host.example.com cat < filename.in > filename.out
scp host.example.com:filename.in filename.out

/usr/sbin/sshd
/usr/sbin/sshd
echo get filename.in filename.out | sftp host.example.com
Client to Server to Another Client
Good reasons for this architecture

- Clients go offline often.
- Clients are behind firewall.
- Clients are behind NAT.
Web browser client
CLIENT1 → Amazon → CLIENT2
Free implementations
CLIENT1 $\rightarrow$ chain of free email servers $\rightarrow$ CLIENT2
Or don't run a server
https://fritz.ftp.sh/fossil/send_files
(Upload in web browser)

send_files receive .
send_files publish .

(Download in web browser)
Context menu
Copying files between computers

- Client->Server
- Server->Client
- Client->Server->Client

- send_files
- Context menu