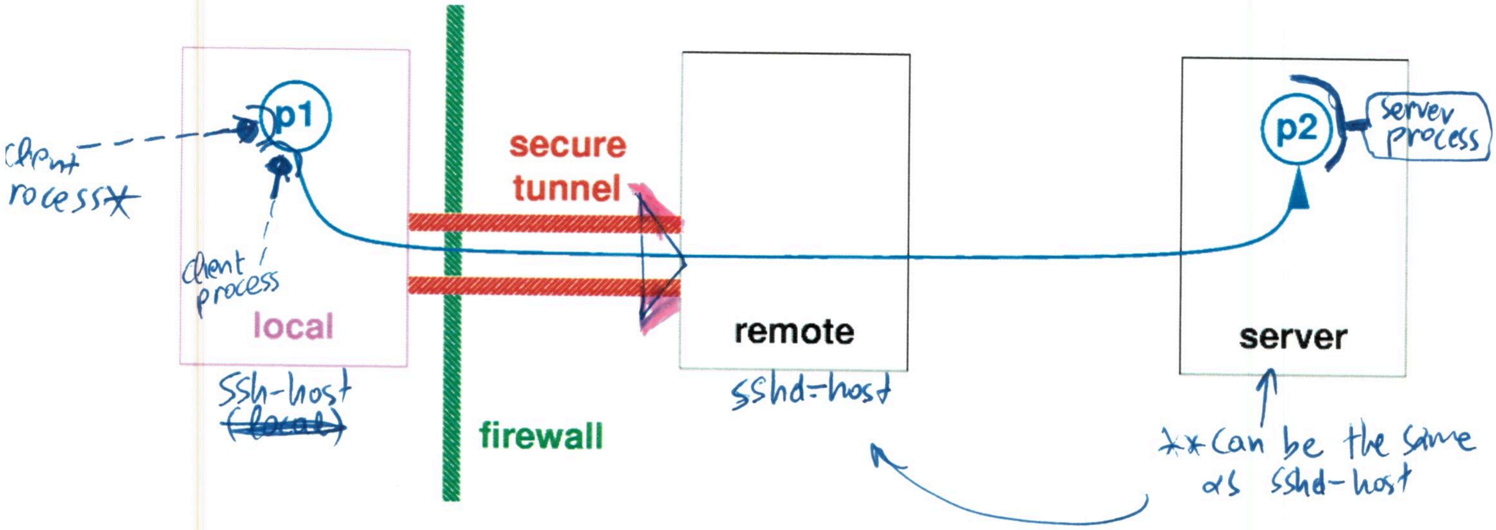
Local port forwarding

Tunneling a client connection is done by way of local port forwarding. A typical example is using an ftp-client from a machine connected to an insecure network, to a server behind a firewall. By forwarding some local port to the remote port 21, and connecting the ftp-client to that port on the local machine, every communication of the ftp-client goes, via that local port, through the ssh connection to the remote machine, where it is forwarded to port 21, on which the ftp server is listening.

Accessing a server in a secure way

```
local$ ssh -L p1:server:p2 remote
```

** localhost



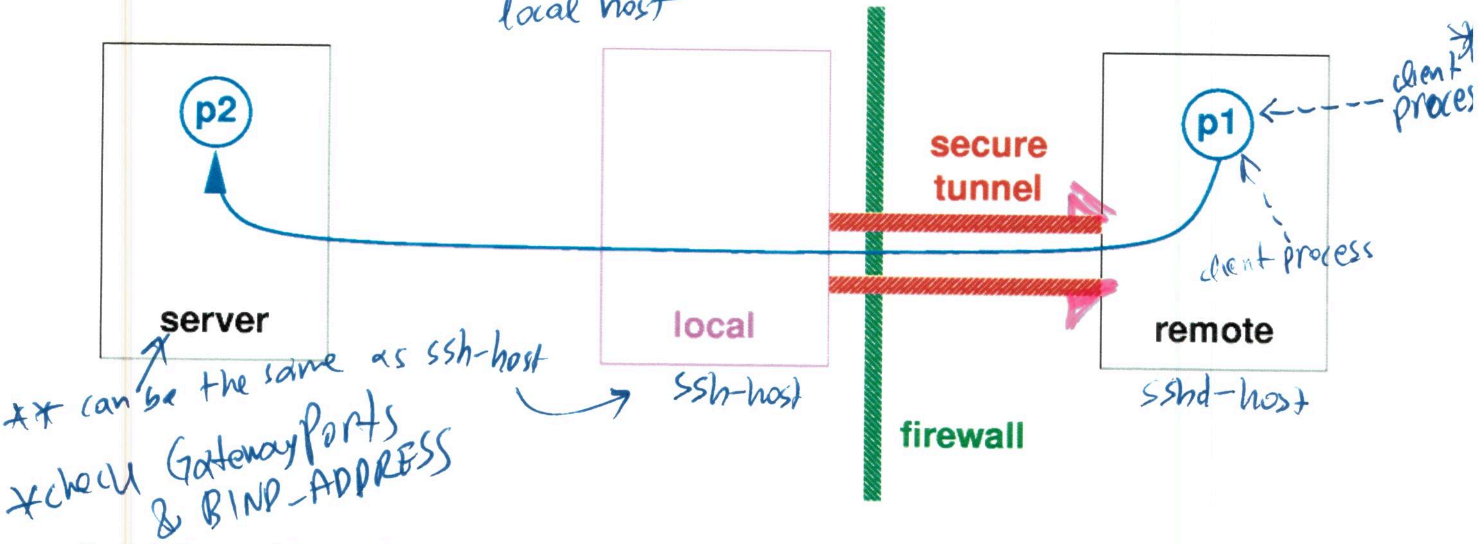
Remote port forwarding

Tunneling a server connection is done by way of remote port forwarding. For the rest it is very similar to local port forwarding.

Providing access to a server in a secure way

```
local$ ssh -R p1:server:p2 remote
```

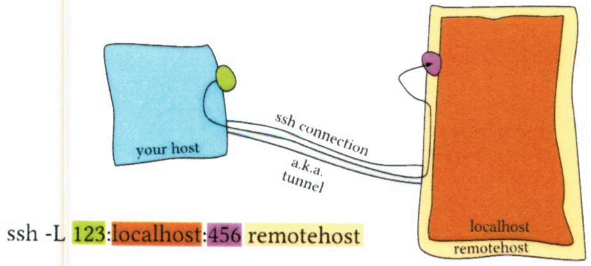
** local host



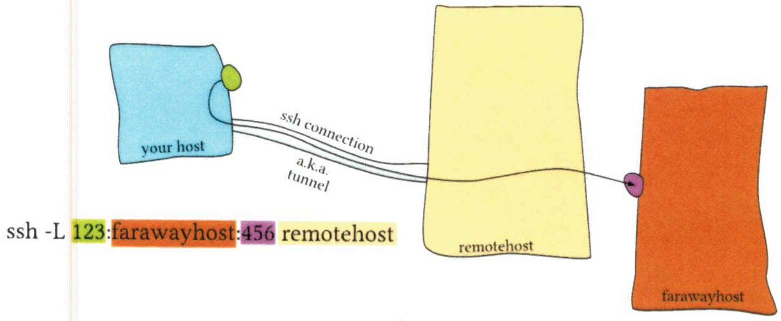
Restrictions with port forwarding

Please make sure that your SSH tunnels are only accessible from your own machine: any configuration items such as *Allow remote hosts to connect to tunnel* should be configured such that only **localhost** can connect to any tunnel you set up ! If your tunnels are accessible for other machines as well, you create huge holes in the departmental security ... so please take care.

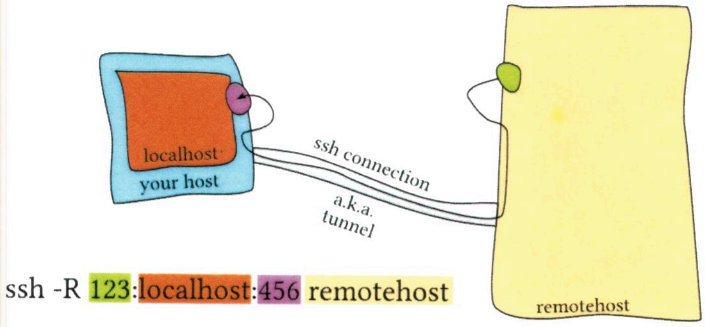
I have drawn some sketches



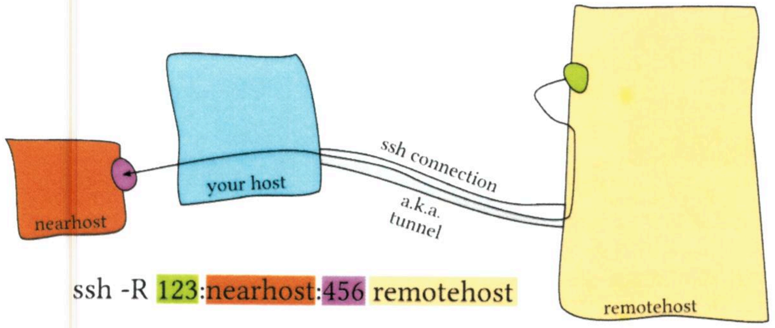
```
ssh -L 123:localhost:456 remotehost
```



```
ssh -L 123:farawayhost:456 remotehost
```



```
ssh -R 123:localhost:456 remotehost
```



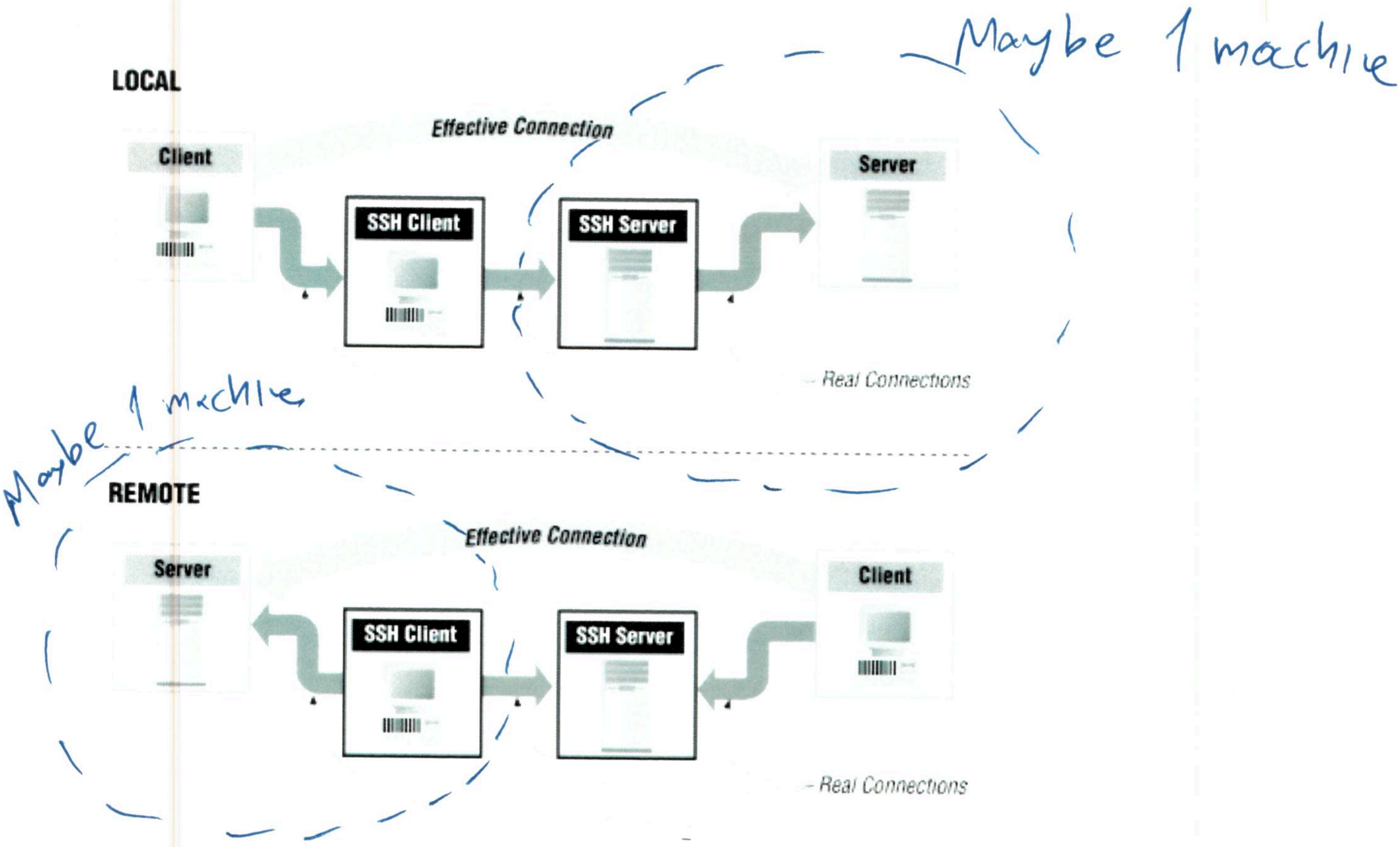
```
ssh -R 123:nearhost:456 remotehost
```

Introduction

1. local: `-L Specifies that the given port on the local (client) host is to be forwarded to the given host and port on the remote side.`

   `ssh -L sourcePort:forwardToHost:onPort connectToHost` means: connect with ssh to `connectToHost`, and forward all connection attempts to the **local** `sourcePort` to port `onPort` on the machine called `forwardToHost`, which can be reached from the `connectToHost` machine.

2. remote: `-R Specifies that the given port on the remote (server) host is to be forwarded to the given host and port on the local side.`

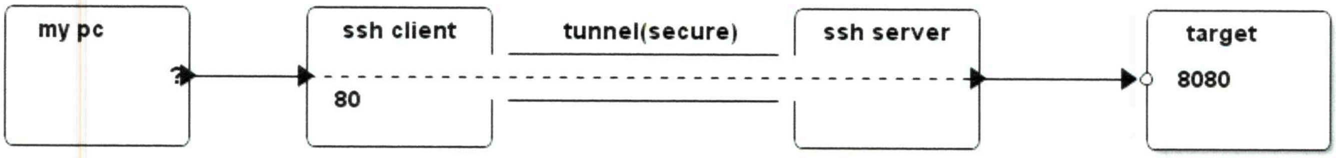


Maybe 1 machine

Maybe 1 machine

Jump three hops

Note that in fact a traffic can be originated from a forth machine. That is, you can "jump 3 hops" with a single SSH tunnel.



According to man page, the listening port will, by default, be bond to the loop back interface only. To forward traffics from other machines, one must

1. Set `GatewayPorts` in `/etc/ssh/sshd_config` to "yes".
2. Specify bind address. For instance (empty or * for any), on ssh client run the following command:

```
ssh -L :80:ssh_server:8080 target
```

But according to my test,

1. GatewayPorts does not take effect.
2. Remote port forward bind to all interface by default.

Option '-N'

> Do not execute a remote command. This is useful for just forwarding ports (protocol version 2 only).