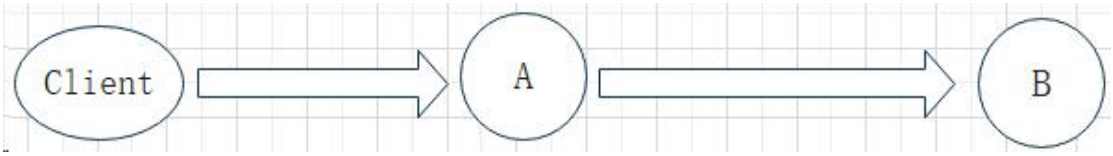
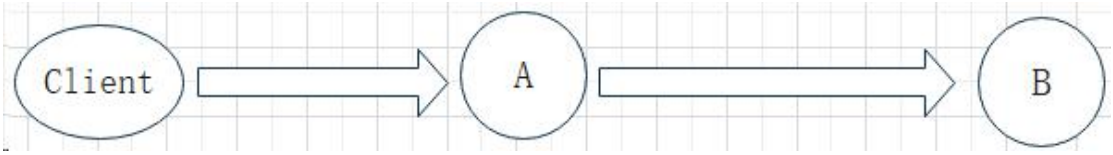
代理_端口转发_端口映射_NAT_概念释疑

①端口转发



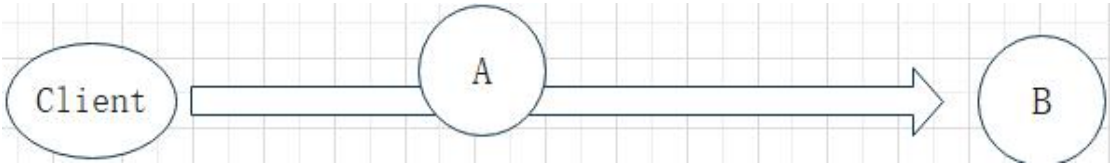
Client 访问的目标 ip:port 为 A，　　（这个 ip 是网络层的 ip 地址，port 为传输层 port）
A 收到报文后将目标 ip:port 改为 B 的，源 ip:port 不变
B 响应的报文是发给 Client，
B 发给 client 的流量不一定原路返回（即可能不经过 A）

②端口代理



Client 访问的目标 ip:port 为 A，
A 收到报文后将源 ip:port 改为 A 自己，将目标 ip:port 改为 B 的，
B 响应的报文是发给 A，
A 收到响应报文后再根据映射关系修改响应报文的源 ip:port 为 A，目的 ip:port 为 client
（响应流量原路返回）

③sNAT 源地址转换



Client 访问的目标 ip:port 为 B，
流量经过 A，A 将请求报文的源 ip:port 改为 A 自己，目标不变，
B 响应的报文是发给 A
A 收到响应报文后再根据映射关系修改响应报文的目的 ip:port 为 Client
（响应流量原路返回）

④dNAT 目的地址转换

也叫作 **端口映射**，原理同 **端口转发**
（响应流量不一定原路返回）

http(s)/socks 代理（一般指浏览器/web 服务器的代理），分 2 种方向:

⑤正向代理：

- Client 访问的目标不管是谁（只要符合走代理的规则），就统统把流量发给代理服务器，让代理帮 Client 去访问目标服务器，
- Client 到代理之间的（http(s)/tcp/udp）流量是封装在代理协议层之上的，也就是说套了一层代理协议的壳，
- Client 知道自己是把流量发给了代理服务器
- 代理服务器收到流量后，再解开这个壳，得到（http(s)/tcp/udp）流量，再根据此流量里的相关信息（如域名）去找目标服务器的 ip，最后代理再做 **sNAT 源地址转换** 把流量发给目标服务器。
- 目标服务器响应的报文是发给代理服务器，
- 代理收到响应报文后再根据映射关系修改响应报文的目的 ip:port 为 Client

（响应流量原路返回）

⑥反向代理：

- Client 就正常访问目标服务器 B，Client 并不知道自己访问的是代理，
- 结果 B 是一个反向代理服务器，它把收到的 client 发来的流量再进行某些修改（如 修改 Http 的报头字段，当然也可不修改），再做 **端口代理** 发给后面的真实服务器
- 后面的真实服务器的响应报文是发给反向代理，
- 反向代理收到响应报文后再根据映射关系修改响应报文的源 ip:port 为自己，目的 ip:port 为 client

（响应流量原路返回）

总结：

根据报文的源 ip:port 及目标 ip:port 的修改情况以及应用场景的不同 可以得出以上六种不同的网络术语。仅根据报文的源 ip:port 及目标 ip:port 的修改情况 只需要区分 3 种情况：

1. **只修改目标 ip:port**，即端口转发、端口映射，其中 ip:port 可以只改其一，必改其一
   目标的响应报文不一定原路返回，所以 Client 那边会得不到正确的响应，因为响应如果不经过中间的 A（修改报文 ip:port 的始作俑者），则返回时的报文 ip:port 得不到修改，所以 client 认为此响应报文不是正确的，所以要确保响应报文在路由上能经过 A
2. **只修改源 ip:port**，即 sNAT 源地址转换，其中 ip:port 可以只改 ip，也必须改 ip
   响应报文一定原路返回
3. **修改源 ip:port 以及修改目标 ip:port**，即端口代理，源中的 ip:port 可以只改 ip，也必改 ip
   目标 ip:port 可以只改其一，即改 ip 或 port 都行，必改其一
   响应报文一定原路返回

作者：Cof-Lee

2020-07-28