10 NAT、代理服务、内网穿透

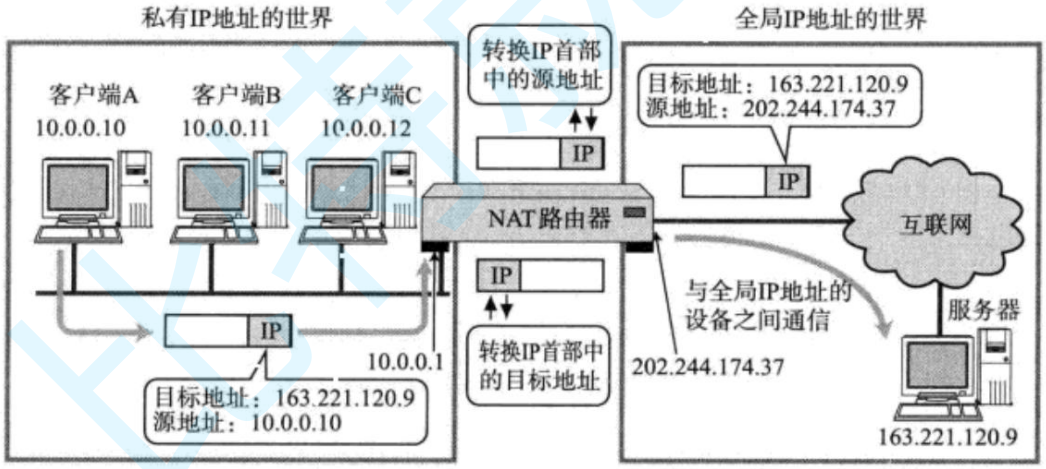
NAT 技术背景

之前我们讨论了, IPv4 协议中, IP 地址数量不充足的问题

NAT 技术当前解决 IP 地址不够用的主要手段, 是路由器的一个重要功能;

- NAT 能够将私有 IP 对外通信时转为全局 IP. 也就是就是一种将私有 IP 和全局 IP 相互转化的技术方法:
- 很多学校, 家庭, 公司内部采用每个终端设置私有 IP, 而在路由器或必要的服务器上设置全局 IP;
- 全局 IP 要求唯一, 但是私有 IP 不需要; 在不同的局域网中出现相同的私有 IP 是完全不影响的;

NAT IP 转换过程

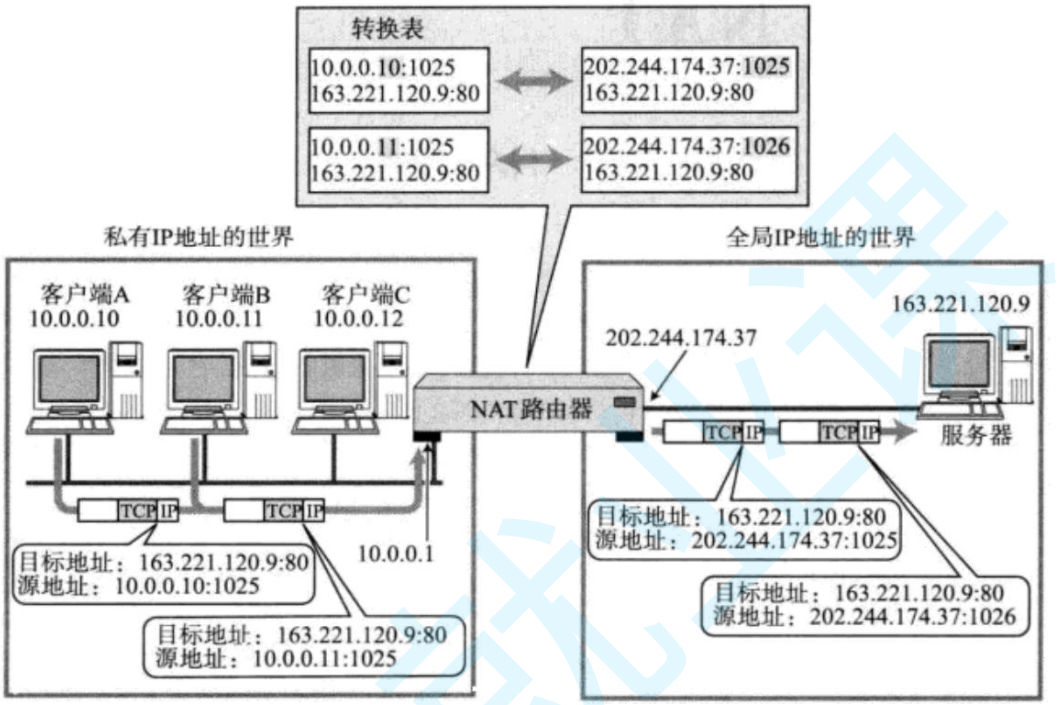


- NAT 路由器将源地址从 10.0.0.10 替换成全局的 IP 202.244.174.37;
- NAT 路由器收到外部的数据时, 又会把目标 IP 从 202.244.174.37 替换回 10.0.0.10;
- 在 NAT 路由器内部, 有一张自动生成的, 用于地址转换的表;
- 当 10.0.0.10 第一次向 163.221.120.9 发送数据时就会生成表中的映射关系;

NAPT

那么问题来了, 如果局域网内, 有多个主机都访问同一个外网服务器, 那么对于服务器返回的数据中, 目的 IP 都是相同的. 那么 NAT 路由器如何判定将这个数据包转发给哪个局域网的主机?

这时候 NAPT 来解决这个问题了. 使用 IP+port 来建立这个关联关系



这种关联关系也是由 NAT 路由器自动维护的. 例如在 TCP 的情况下, 建立连接时, 就会生成这个表项; 在断开连接后, 就会删除这个表项

NAT 技术的缺陷

由于 NAT 依赖这个转换表, 所以有诸多限制:

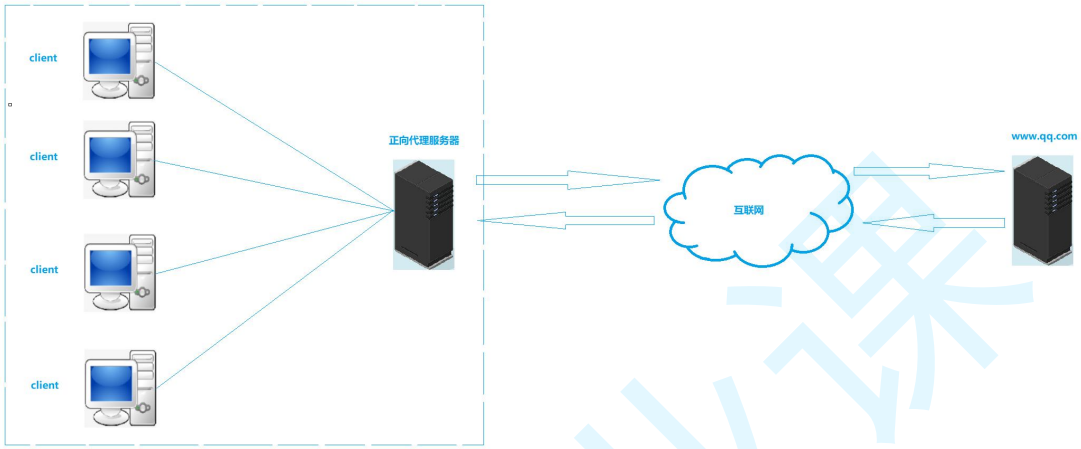
- 无法从 NAT 外部向内部服务器建立连接;
- 装换表的生成和销毁都需要额外开销;
- 通信过程中一旦 NAT 设备异常, 即使存在热备, 所有的 TCP 连接也都会断开;

代理服务器

正向代理

概述

- 正向代理（Forward Proxy）是一种常见的网络代理方式，它位于客户端和目标服务器之间，代表客户端向目标服务器发送请求。正向代理服务器接收客户端的请求，然后将请求转发给目标服务器，最后将目标服务器的响应返回给客户端。通过这种方式，正向代理可以实现多种功能，如提高访问速度、隐藏客户端身份、实施访问控制等。



工作原理

- 客户端将请求发送给正向代理服务器。
- 正向代理服务器接收请求，并根据配置进行处理，如缓存查找、内容过滤等。
- 正向代理服务器将处理后的请求转发给目标服务器。
- 目标服务器处理请求，并将响应返回给正向代理服务器。
- 正向代理服务器将响应返回给客户端。

功能特点

- 缓存功能：正向代理服务器可以缓存经常访问的资源，当客户端再次请求这些资源时，可以直接从缓存中获取，提高访问速度。
- 内容过滤：正向代理可以根据预设的规则对请求或响应进行过滤，如屏蔽广告、阻止恶意网站等。
- 访问控制：通过正向代理，可以实现对特定网站的访问控制，如限制员工在工作时间访问娱乐网站。
- 隐藏客户端身份：正向代理可以隐藏客户端的真实 IP 地址，保护客户端的隐私。
- 负载均衡：在多个目标服务器之间分配客户端请求，提高系统的可扩展性和可靠性。

应用场景

企业网络管理：企业可以通过正向代理实现对员工网络访问的管理和控制，确保员工在工作时间内专注于工作，避免访问不良网站或泄露公司机密。

公共网络环境：在公共场所如图书馆、学校等提供的网络环境中，通过正向代理可以实现对网络资源的合理分配和管理，确保网络使用的公平性和安全性。

内容过滤与保护：家长可以通过设置正向代理来过滤不良内容，保护孩子免受网络上的不良信息影响。

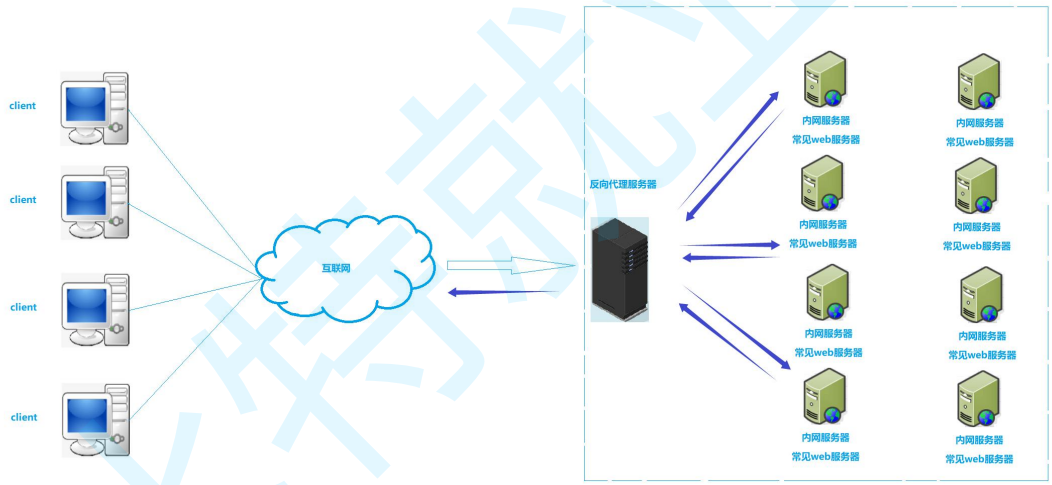
提高访问速度：对于经常访问的网站或资源，正向代理可以通过缓存机制提高访问速度，减少网络延迟。

跨境电商与海外访问：对于跨境电商或需要访问海外资源的企业和个人，正向代理可以帮助他们突破网络限制，顺畅地访问海外网站和资源。

反向代理

概述

- 反向代理服务器是一种网络架构模式，其作为 Web 服务器的前置服务器，接收来自客户端的请求，并将这些请求转发给后端服务器，然后将后端服务器的响应返回给客户端。这种架构模式可以提升网站性能、安全性和可维护性等



基本原理

- 反向代理服务器位于客户端和 Web 服务器之间，当客户端发起请求时，它首先会到达反向代理服务器。反向代理服务器会根据配置的规则将请求转发给后端的 Web 服务器，并将 Web 服务器的响应返回给客户端。在这个过程中，客户端并不知道实际与哪个 Web 服务器进行了交互，它只知道与反向代理服务器进行了通信。

应用场景

- 负载均衡：反向代理服务器可以根据配置的负载均衡策略，将客户端的请求分发到多个后端服务器上，以实现负载均衡。这有助于提升网站的整体性能和响应速度，特别是在高并发场景下。
- 安全保护：反向代理服务器可以隐藏后端 Web 服务器的真实 IP 地址，降低其被

直接攻击的风险。同时，它还可以配置防火墙、访问控制列表（ACL）等安全策略，对客户端的请求进行过滤和限制，以保护后端服务器的安全。

- 缓存加速：反向代理服务器可以缓存后端 Web 服务器的响应内容，对于重复的请求，它可以直接从缓存中返回响应，而无需再次向后端服务器发起请求。这可以大大减少后端服务器的负载，提升网站的响应速度。
- 内容过滤和重写：反向代理服务器可以根据配置的规则对客户端的请求进行过滤和重写，例如添加或删除请求头、修改请求路径等。这有助于实现一些特定的业务需求，如 URL 重写、用户认证等。
- 动静分离：在大型网站中，通常需要将静态资源和动态资源分开处理。通过将静态资源部署在反向代理服务器上，可以直接从反向代理服务器返回静态资源的响应，而无需再次向后端服务器发起请求。这可以大大提升静态资源的访问速度。
- CDN（Content Delivery Network，内容分发网络）就是采用了反向代理的原理

NAT 和代理服务器

路由器往往都具备 NAT 设备的功能, 通过 NAT 设备进行中转, 完成子网设备和其他子网设备的通信过程.

代理服务器看起来和 NAT 设备有一点像. 客户端像代理服务器发送请求, 代理服务器将请求转发给真正要请求的服务器; 服务器返回结果后, 代理服务器又把结果回传给客户端.

那么 NAT 和代理服务器的区别有哪些呢?

> - 从应用上讲, NAT 设备是网络基础设备之一, 解决的是 IP 不足的问题. 代理服务器则是更贴近具体应用, 比如通过代理服务器进行翻墙, 另外像迅游这样的加速器, 也是使用代理服务器.
> - 从底层实现上讲, NAT 是工作在网络层, 直接对 IP 地址进行替换. 代理服务器往往工作在应用层.
> - 从使用范围上讲, NAT 一般在局域网的出口部署, 代理服务器可以在局域网做, 也可以在广域网做, 也可以跨网.
> - 从部署位置上看, NAT 一般集成在防火墙, 路由器等硬件设备上, 代理服务器则是一个软件程序, 需要部署在服务器上.

代理服务器是一种应用比较广的技术.

> - 翻墙: 广域网中的代理.
> - 负载均衡: 局域网中的代理.

代理服务器又分为正向代理和反向代理.

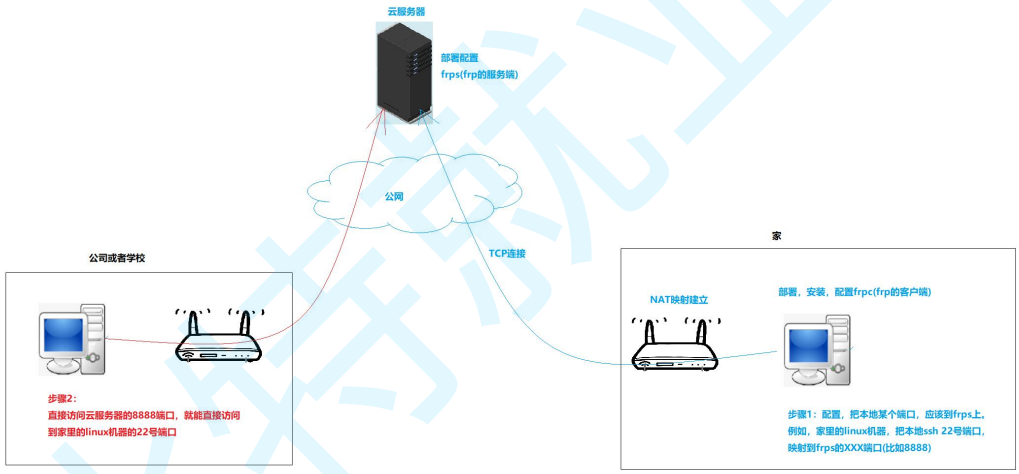
代购例子

```
C
花王尿不湿是一个很经典的尿不湿品牌，产自日本.
我自己去日本买尿不湿比较不方便，但是可以让我在日本工作的表姐去超市买了快递给我．此时超市看到的买家是我表姐，我的表姐就是 "正向代理";
后来找我表姐买尿不湿的人太多了，我表姐觉得天天去超市太麻烦，干脆去超市买了一大批尿不湿屯在家里，如果有人来找她代购，就直接把屯在家里的货发出去，而不必再去超市．此时我表姐就是 "反向代理"
```

正向代理用于请求的转发(例如借助代理绕过反爬虫).

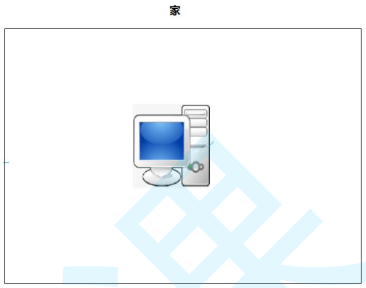
反向代理往往作为一个缓存.

内网穿透



内网打洞

- 现场画图解释原理



总结

数据链路层

- 数据链路层的作用: 两个设备(同一种数据链路节点)之间进行传递数据
- 以太网是一种技术标准; 既包含了数据链路层的内容, 也包含了一些物理层的内容. 例如: 规定了网络拓扑结构, 访问控制方式, 传输速率等;
- 以太网帧格式
- 理解 mac 地址
- 理解 arp 协议
- 理解 MTU

网络层

- 网络层的作用: 在复杂的网络环境中确定一个合适的路径.
- 理解 IP 地址, 理解 IP 地址和 MAC 地址的区别.
- 理解 IP 协议格式.
- 了解网段划分方法
- 理解如何解决 IP 数目不足的问题, 掌握网段划分的两种方案. 理解私有 IP 和公网 IP
- 理解网络层的 IP 地址路由过程. 理解一个数据包如何跨越网段到达最终目的地.

- 理解 IP 数据包分包的原因.
- 了解 ICMP 协议.
- 了解 NAT 设备的工作原理.

传输层

- 传输层的作用: 负责数据能够从发送端传输接收端.
- 理解端口号的概念.
- 认识 UDP 协议, 了解 UDP 协议的特点.
- 认识 TCP 协议, 理解 TCP 协议的可靠性. 理解 TCP 协议的状态转化.
- 掌握 TCP 的连接管理, 确认应答, 超时重传, 滑动窗口, 流量控制, 拥塞控制, 延迟应答, 捎带应答特性.
- 理解 TCP 面向字节流, 理解粘包问题和解决方案.
- 能够基于 UDP 实现可靠传输.
- 理解 MTU 对 UDP/TCP 的影响.

应用层

- 应用层的作用: 满足我们日常需求的网络程序, 都是在应用层
- 能够根据自己的需求, 设计应用层协议.
- 了解 HTTP 协议.
- 理解 DNS 的原理和工作流程.