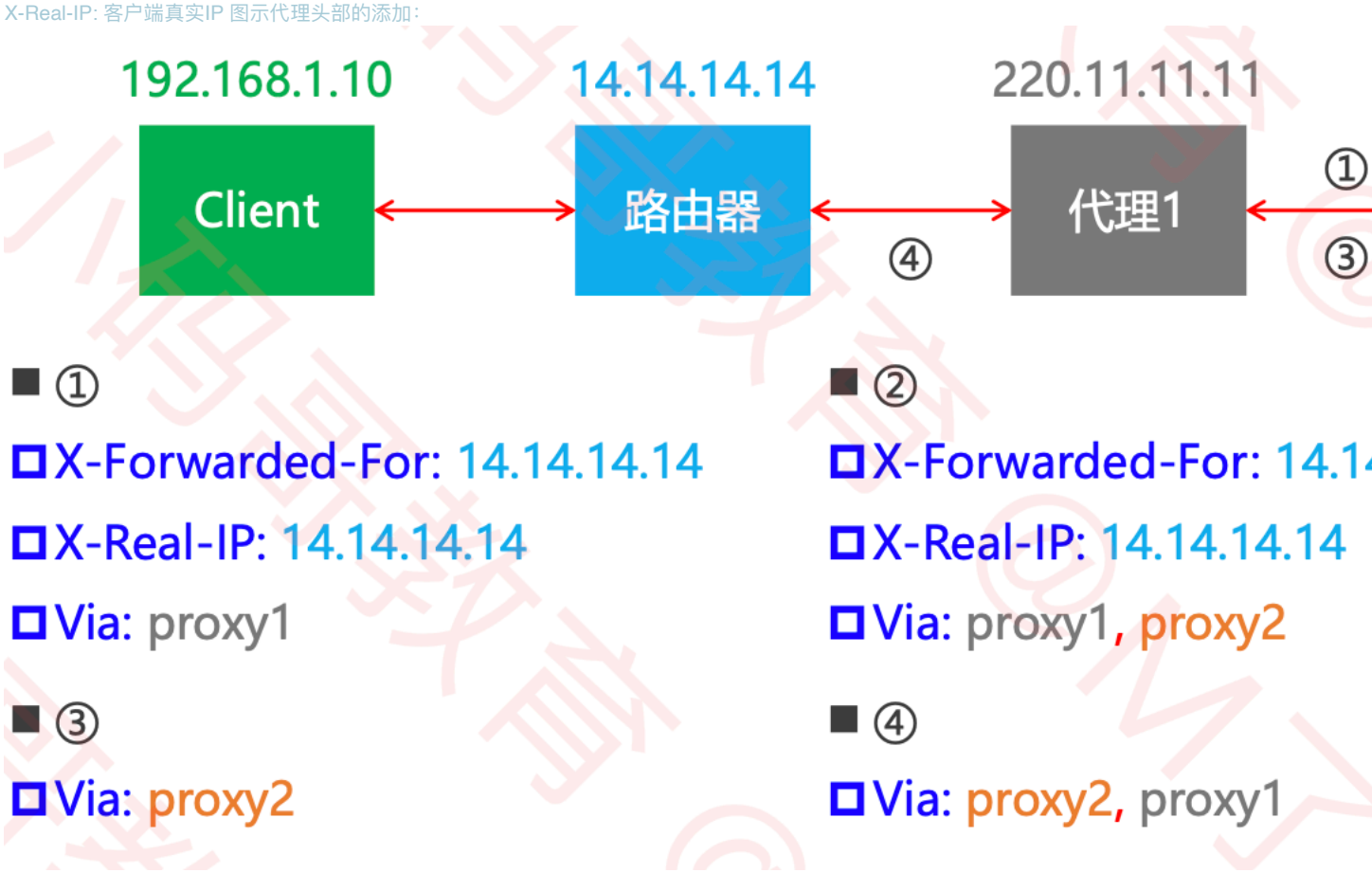
网络协议笔记5-应用层2（代理，加密）

代理

- 特点：不产生数据，只是转发上下游的数据
  - 面向下游的时候是服务器
  - 面向上游的时候是客户端
- 正向代理： 代理对象是客户端
  - 隐藏客户端身份
  - 绕过防火墙
- 反向代理： 代理对象是服务器
  - 隐藏服务器身份
  - 安全防护：
  - 负载均衡： 用户访问的时候访问代理服务器，代理服务器判断真实服务器的负载量，将该请求交给不忙的服务器处理数据

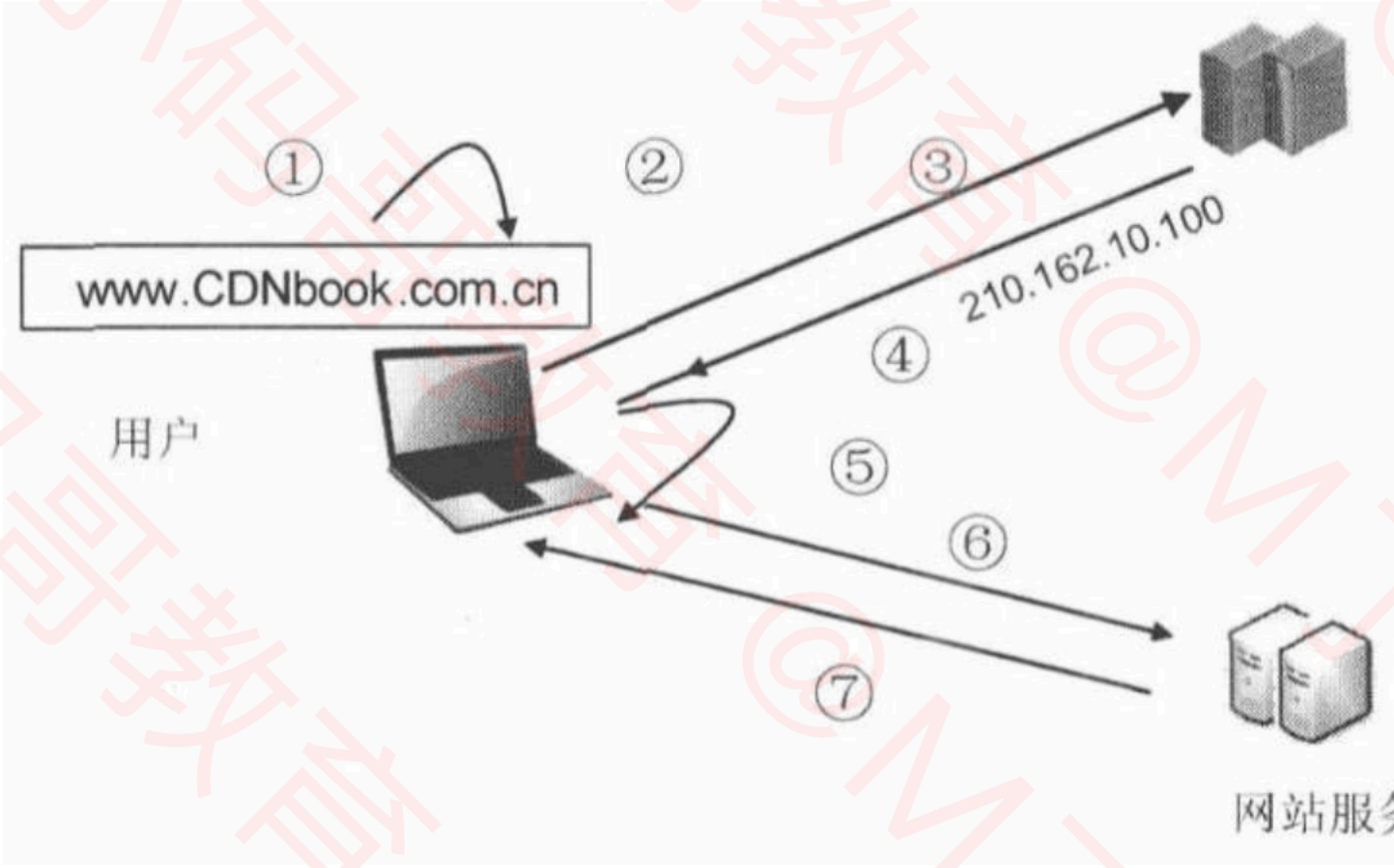
代理的相关头部字段

- Via：追加经过每一台代理服务器的主机名或域名
- X-Forwarded-For: 追加请求方的IP地址
- X-Real-IP: 客户端真实IP 图示代理头部的添加：

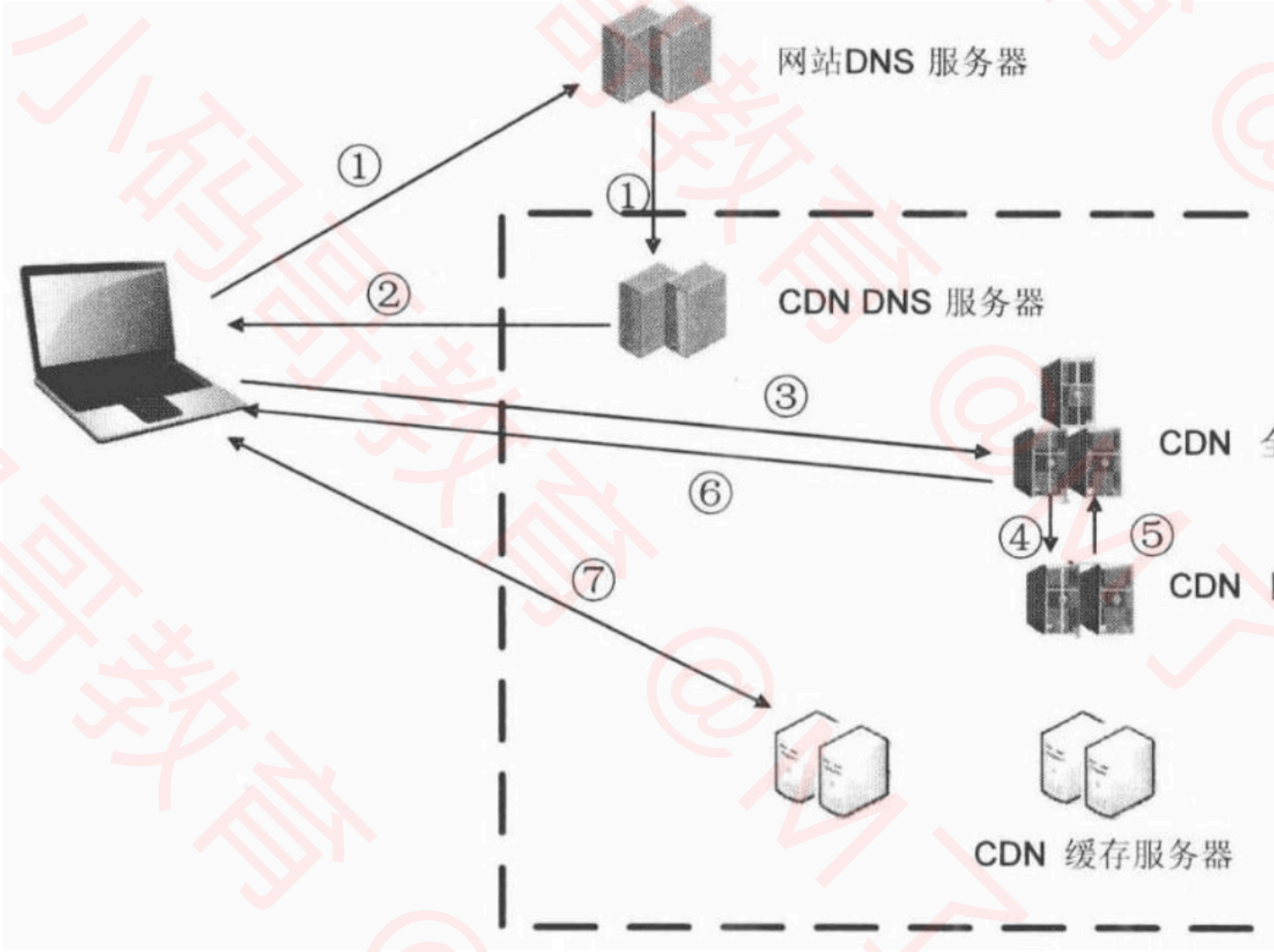


CDN

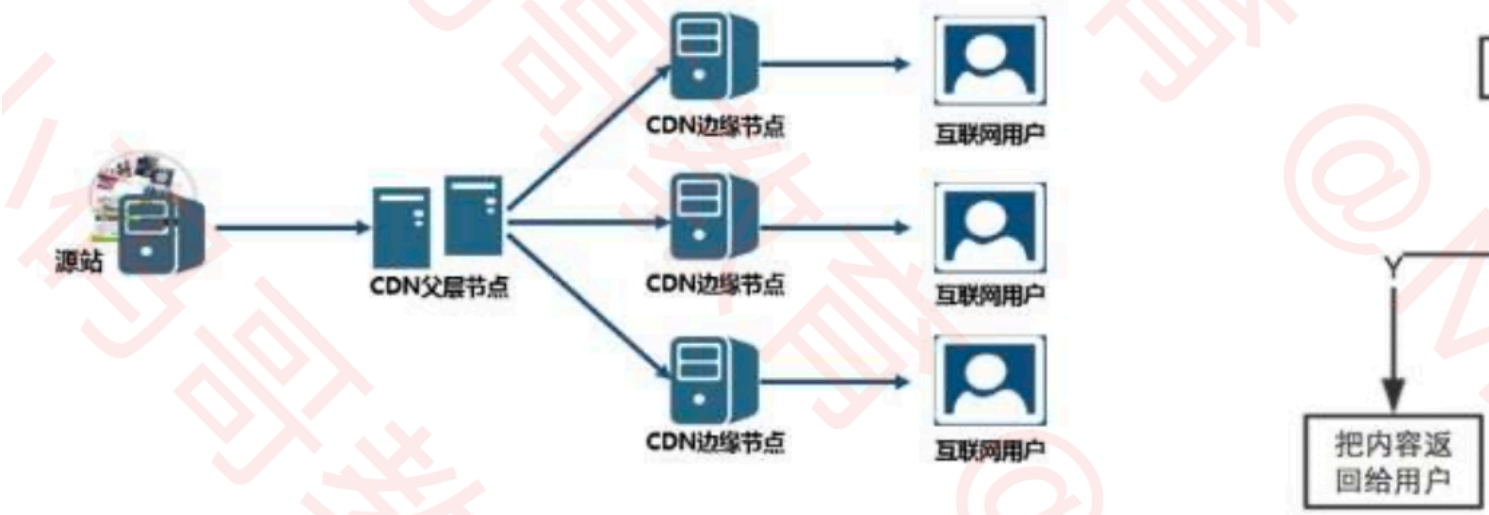
内容分发网络，利用最靠近用户的服务器将用户需要的图片视频等资源传给用户 使用cdn前的请求示意图：



使用cdn前的请求示意图：



请求流程图：



缓存

加密

常见的加密方式：

不可逆：

单向散列函数，虽不是加密可看作加密。不同大小的文件计算后的散列函数的字节数相等。

- MD5：产生128bit的散列值
- SHA：sha-256，sha-384,sha-512散列长度分别为256，384和512bit
- 验证文件是否被篡改的方式：将文件进行散列处理，对比处理后的字节内容与前面是否一致

可逆

对称加密

- 对称加密使用的密钥是同一个
- DES
- 3DES：是将DES加密重复3次的加密算法，过程是 *发送端：明文-> 加密（密钥3）->解密（密钥2）->加密（密钥1）接收端：密文-> 解密（密*

钥3）->加密（密钥2）->解密（密钥1），所以当3个密钥相同时，3DES将退化成DES

- AES
- 对称加密的问题： A将加密过的密文发送给B，需要将密钥发送给B，如果发送密钥过程中遭遇C的窃取，那么C也可以通过该密钥解密，可以通过非对称加密解决问题 #### 非对称加密
- 公钥： 公开的密钥，用于加密。
- 私钥： 私有的密钥，不公开，用于解密
- 公钥和私钥是1对1的，不能单独生成
- 发送方使用接收方的公钥来进行加密，接收方使用只有自己知道的私钥进行解密，保证信息安全
- RSA等
- 非对称加密的速度弱于对称加密

其他

混合密码系统

- 原因：对称加密的密钥容易被窃听，非对称加密的速度过慢，所以采用混合加密方式处理
- A发送给B： A将会话密钥(对称加密的密钥)使用B的公钥进行加密，发送给B的时候将已被非对称加密的会话密钥和密文一同传递给B，B将该密钥使用私钥解密后，使用解密后的会话密钥进行对称解密

数字签名

- 产生原因：无法保证接收到的消息是发送方发送的消息，有可能经过第三方的篡改
- 过程
  1. 消息发送方通过签名密钥生成签名 (使用私钥签名)
  2. 消息接受方通过验证密钥进行解签验证 (使用公钥解签)
- 优化：发送方将消息生成一个散列值，将散列值进行私钥加密发送给接收方，接受方将收到的签名解密，与收到的消息的散列值进行对比。
- 作用
  1. 确认消息的完整性
  2. 识别消息是否被篡改
  3. 防止发送人否认

总结：

	公钥	私钥
非对称加密	发送者加密时使用	接收方解密用
数字签名	验证者验证时使用	签名方签名时使用
持有者	任何人都可以持有	个人持有

证书

- 当A获取B的公钥时，攻击者可以拦截B的公钥后将自己的公钥发送给A，获取A的消息时可以使用自己的私钥解密，然后伪造信息发送给B。所以需要证书验证公钥的合法性
- 证书是权威机构颁发的
- 一般已经存储在浏览器和操作系统本地，不需要通过网络获取
- 验证流程：B是接收者，A是发送者
  1. B生成自己的公钥
  2. B在机构注册自己的公钥（内置了机构的公钥，通过该公钥保证自己的公钥不会被篡改）
  3. 机构使用自己的私钥和B的公钥施加签名并生成证书
  4. A得到包含B公钥和机构签名的证书
  5. A可以通过机构的公钥验证签名，保证B的公钥的合法性
  6. 通过B的公钥加密