

9 数据链路层

数据链路层

用于两个设备(同一种数据链路节点)之间进行传递.

对比理解 "数据链路层" 和 "网络层"

[唐僧例子之白龙马]

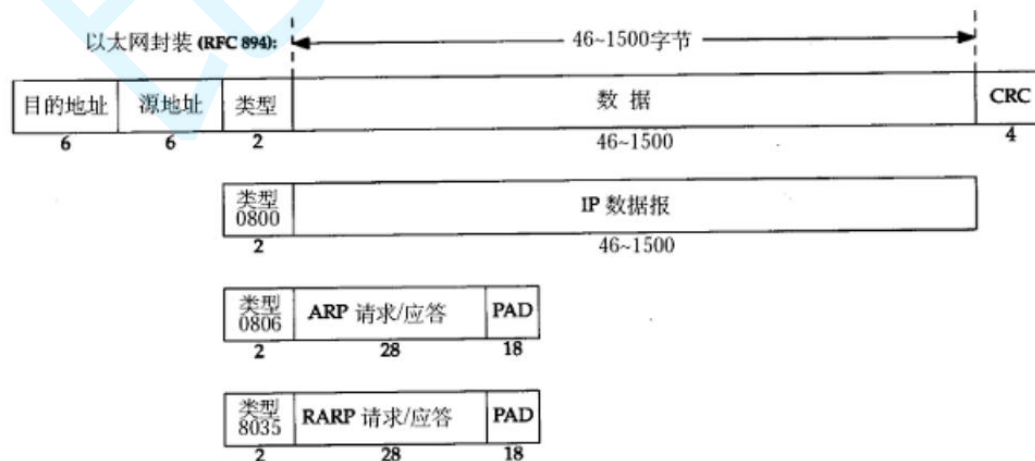
认识以太网

- "以太网" 不是一种具体的网络, 而是一种技术标准; 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;
- 例如以太网中的网线必须使用双绞线; 传输速率有 10M, 100M, 1000M 等;
- 以太网是当前应用最广泛的局域网技术; 和以太网并列的还有令牌环网, 无线 LAN 等;

以太网帧格式

以太网的帧格式如下所示:

以太网的帧格式如下所示:



- 源地址和目的地址是指网卡的硬件地址(也叫 MAC 地址), 长度是 48 位,是在网卡出厂时固化的;
- 帧协议类型字段有三种值,分别对应 IP、ARP、RARP;
- 帧末尾是 CRC 校验码。

认识 MAC 地址

- MAC 地址用来识别数据链路层中相连的节点;
- 长度为 48 位, 及 6 个字节. 一般用 16 进制数字加上冒号的形式来表示(例如: 08:00:27:03:fb:19)
- 在网卡出厂时就确定了, 不能修改. mac 地址通常是唯一的(虚拟机中的 mac 地址不是真实的 mac 地址, 可能会冲突; 也有些网卡支持用户配置 mac 地址).

对比理解 MAC 地址和 IP 地址

还是 [唐僧例子之白龙马]

- IP 地址描述的是路途总体的 起点 和 终点;
- MAC 地址描述的是路途上的每一个区间的起点和终点;

认识 MTU

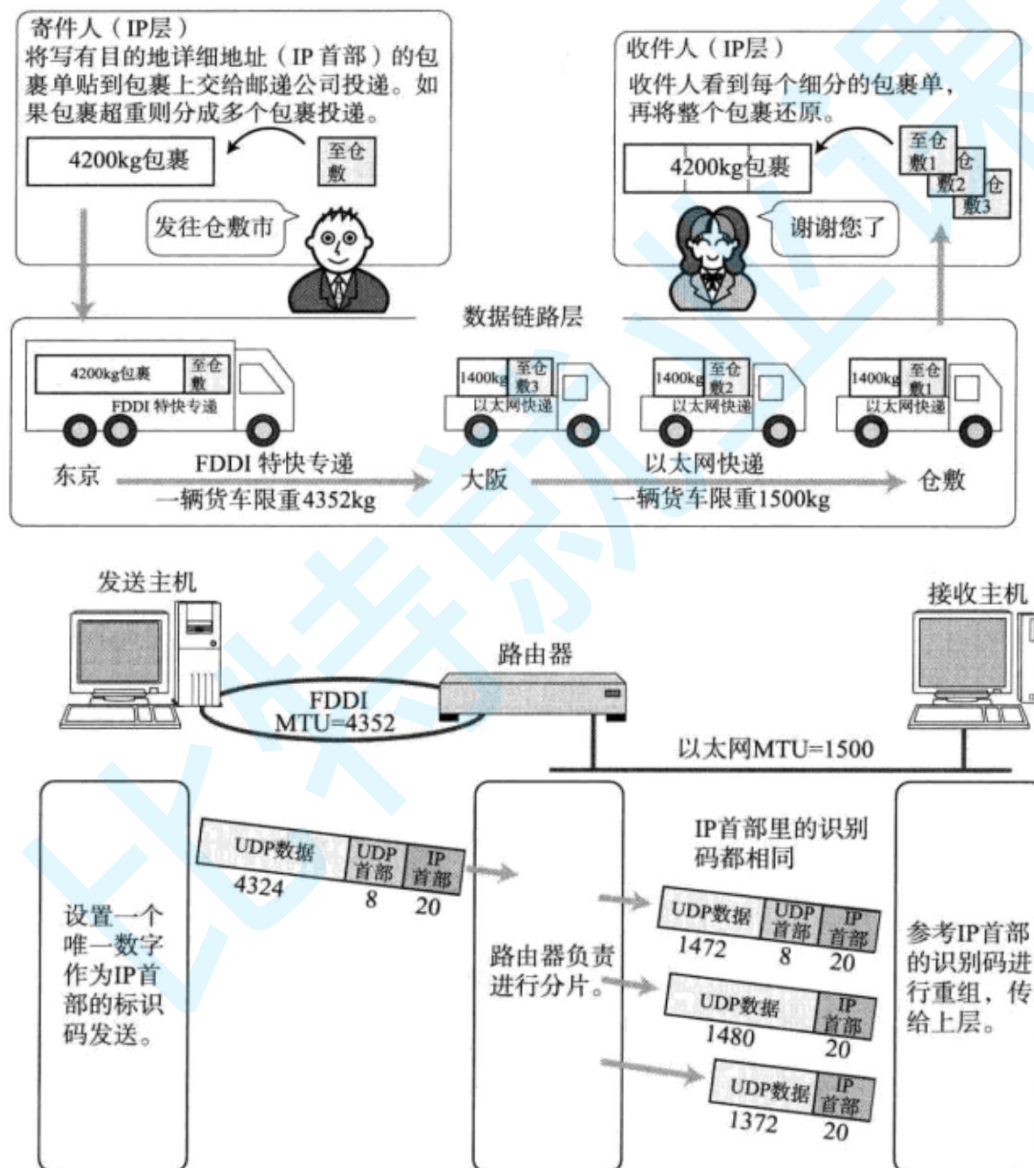
MTU 相当于发快递时对包裹尺寸的限制. 这个限制是不同的数据链路对应的物理层, 产生的限制.

- 以太网帧中的数据长度规定最小 46 字节,最大 1500 字节,ARP 数据包的长度不够 46 字节,要在后面补填充位;
- 最大值 1500 称为以太网的最大传输单元(MTU),不同的网络类型有不同的 MTU;
- 如果一个数据包从以太网路由到拨号链路上,数据包长度大于拨号链路的 MTU 了,则需要对数据包进行分片(fragmentation);
- 不同的数据链路层标准的 MTU 是不同的;

MTU 对 IP 协议的影响

由于数据链路层 MTU 的限制, 对于较大的 IP 数据包要进行分包.

- 将较大的 IP 包分成多个小包, 并给每个小包打上标签;
- 每个小包 IP 协议头的 16 位标识(id) 都是相同的;
- 每个小包的 IP 协议头的 3 位标志字段中, 第 2 位置为 0, 表示允许分片, 第 3 位来表示结束标记(当前是否是最后一个小包, 是的话置为 1, 否则置为 0);
- 到达对端时再将这些小包, 会按顺序重组, 拼装到一起返回给传输层;
- 一旦这些小包中任意一个小包丢失, 接收端的重组就会失败. 但是 IP 层不会负责重新传输数据;



MTU 对 UDP 协议的影响

让我们回顾一下 UDP 协议:

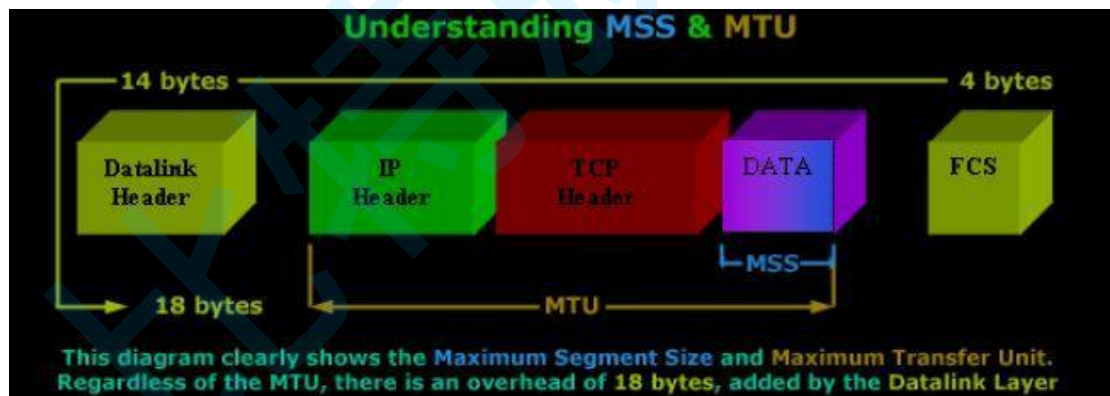
- 一旦 UDP 携带的数据超过 1472(1500 - 20(IP 首部) - 8(UDP 首部)), 那么就会在网络层分成多个 IP 数据报.
- 这多个 IP 数据报有任意一个丢失, 都会引起接收端网络层重组失败. 那么这就意味着, 如果 UDP 数据报在网络层被分片, 整个数据被丢失的概率就大大增加了.

MTU 对于 TCP 协议的影响

让我们再回顾一下 TCP 协议:

- TCP 的一个数据报也不能无限大, 还是受制于 MTU. TCP 的单个数据报的最大消息长度, 称为 MSS(Max Segment Size);
- TCP 在建立连接的过程中, 通信双方会进行 MSS 协商.
- 最理想的情况下, MSS 的值正好是在 IP 不会被分片处理的最大长度(这个长度仍然是受制于数据链路层的 MTU).
- 双方在发送 SYN 的时候会在 TCP 头部写入自己能支持的 MSS 值.
- 然后双方得知对方的 MSS 值之后, 选择较小的作为最终 MSS.
- MSS 的值就是在 TCP 首部的 40 字节变长选项中(kind=2);

MSS 和 MTU 的关系



查看硬件地址和 MTU

```
[tangzhong@tz ~]$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.108 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::fa5f:d814:637d:a6a3 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:03:fb:19 txqueuelen 1000 (Ethernet)
    RX packets 400 bytes 37722 (36.8 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 269 bytes 37356 (36.4 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

使用 ifconfig 命令, 即可查看 ip 地址, mac 地址, 和 MTU;

ARP 协议

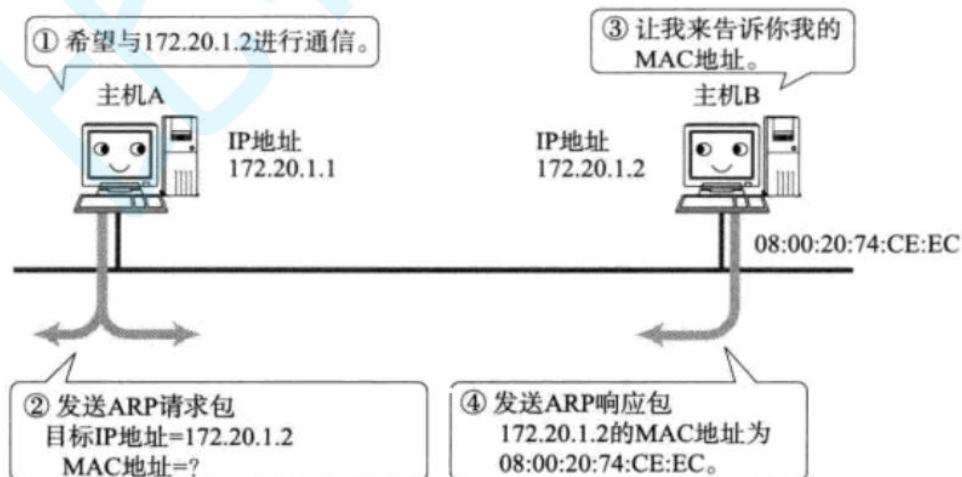
虽然我们在这里介绍 ARP 协议, 但是需要强调, ARP 不是一个单纯的数据链路层的协议, 而是一个介于数据链路层和网络层之间的协议;

ARP 协议的作用

ARP 协议建立了主机 IP 地址 和 MAC 地址 的映射关系.

- 在网络通讯时,源主机的应用程序知道目的主机的 IP 地址和端口号,却不知道目的主机的硬件地址;
- 数据包首先是被网卡接收到再去处理上层协议的,如果接收到的数据包的硬件地址与本机不符,则直接丢弃;
- 因此在通讯前必须获得目的主机的硬件地址;

ARP 协议的工作流程



- 源主机发出 ARP 请求,询问“IP 地址是 192.168.0.1 的主机的硬件地址是多少”,

并将这个请求广播到本地网段(以太网帧首部的硬件地址填 FF:FF:FF:FF:FF:FF 表示广播);

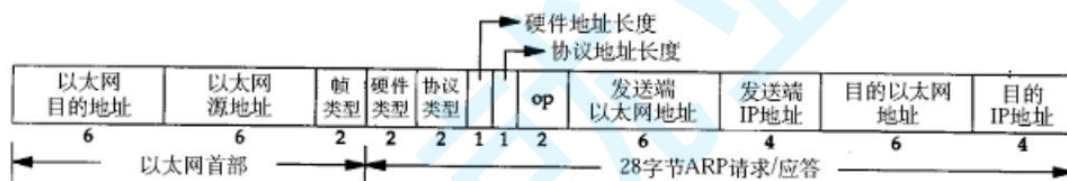
- 目的主机接收到广播的 ARP 请求,发现其中的 IP 地址与本机相符,则发送一个 ARP 应答数据包给源主机,将自己的硬件地址填写在应答包中;
- 每台主机都维护一个 ARP 缓存表,可以用 `arp -a` 命令查看。缓存表中的表项有过期时间(一般为 20 分钟),如果 20 分钟内没有再次使用某个表项,则该表项失效,下次还要发 ARP 请求来获得目的主机的硬件地址

```
[tangzhong@tz ~]$ arp -a
? (192.168.1.107) at e4:f8:9c:be:af:41 [ether] on enp0s3
gateway (192.168.1.1) at 10:bd:18:08:af:62 [ether] on enp0s3
```

想一想,为什么要有缓存表? 为什么表项要有过期时间而不是一直有效?

再想一想, 结合我们刚才讲的工作流程, ARP 的数据报应该是一个什么样的格式?

ARP 数据报的格式



- 注意到源 MAC 地址、目的 MAC 地址在以太网首部和 ARP 请求中各出现一次,对于链路层为以太网的情况是多余的,但如果链路层是其它类型的网络则有可能是必要的。
- 硬件类型指链路层网络类型,1 为以太网;
- 协议类型指要转换的地址类型,0x0800 为 IP 地址;
- 硬件地址长度对于以太网地址为 6 字节;
- 协议地址长度对于和 IP 地址为 4 字节;
- op 字段为 1 表示 ARP 请求,op 字段为 2 表示 ARP 应答。