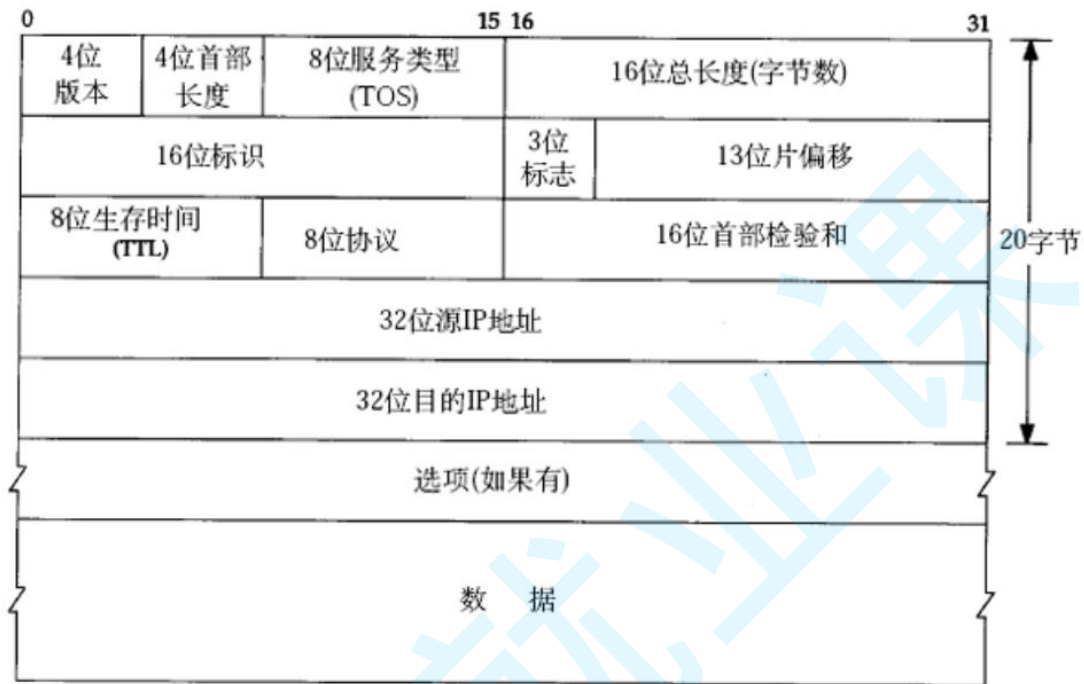


8-1 加餐 - IP 分片和组装的具体过程



- 16 位标识(id): 唯一的标识主机发送的报文. 如果 IP 报文在数据链路层被分片了, 那么每一个片里面的这个 id 都是相同的.
- 3 位标志字段: 第一位保留(保留的意思是现在不用, 但是还没想好说不定以后要用到). 第二位置为 1 表示禁止分片, 这时候如果报文长度超过 MTU, IP 模块就会丢弃报文. 第三位表示"更多分片", 如果分片了的话, 最后一个分片置为 0, 其他是 1. 类似于一个结束标记.
- 13 位分片偏移(fragment offset): 是分片相对于原始 IP 报文开始处的偏移. 其实就是在表示当前分片在原报文中处在哪个位置. 实际偏移的字节数是这个值除以 8 得到的. 因此, 除了最后一个报文之外(之前如果都是 8 的整数倍, 最后一片的偏移量也一定是 8 的整数倍), 其他报文的长度必须是 8 的整数倍(否则报文就不连续了).
- 注意: 片偏移(13 位)表示本片数据在它所属的原始数据报数据区中的偏移量 (以 8 字节为单位)

分片与组装的过程

分片

1. 检查 MTU 限制：

- 当一个 IP 数据报的大小超过了网络的 MTU（最大传输单元）限制时，就需要进行分片。MTU 是数据链路层对 IP 层数据包进行封装时所能接受的最大数据长度。

2. 分割数据报：

- IP 层将原始的 IP 数据报分割成多个较小的片段。
- 对于每个片段，IP 层会设置相应的标识（Identification）、偏移量（Fragment Offset）和标志位（Flags）等字段。
- 标识字段用于标识属于同一个数据报的不同分片，确保所有分片能够被正确地重新组装。
- 偏移量字段指示了当前分片相对于原始数据报的起始位置，以 8 字节为单位。
- 标志位字段包含了 3 个位，其中 **MF（More Fragment）** 位用于指示是否还有更多的分片，**DF（Do Not Fragment）** 位用于指示数据报是否允许进行分片。

3. 添加 IP 头部：

- 每个分片都会加上自己的 IP 头部，与完整 IP 报文拥有类似的 IP 头结构，但 MF 和 Fragment Offset 等字段的值会有所不同。

4. 发送分片：

- 分片在传输过程中独立传输，每个分片都有自己的 IP 头部，并且各自独立地选择路由。

组装

1. 接收分片：

- 当目的主机的 IP 层接收到这些分片后，会根据标识字段将属于同一个数据报的所有分片挑选出来。

2. 排序与组装：

- 利用片偏移字段，IP 层会对属于同一个数据报的分片进行排序。
- 当所有的分片都到达并正确排序后，IP 层会将这些分片重新组装成一个完整的 IP 数据报。

3. 传递给上层协议：

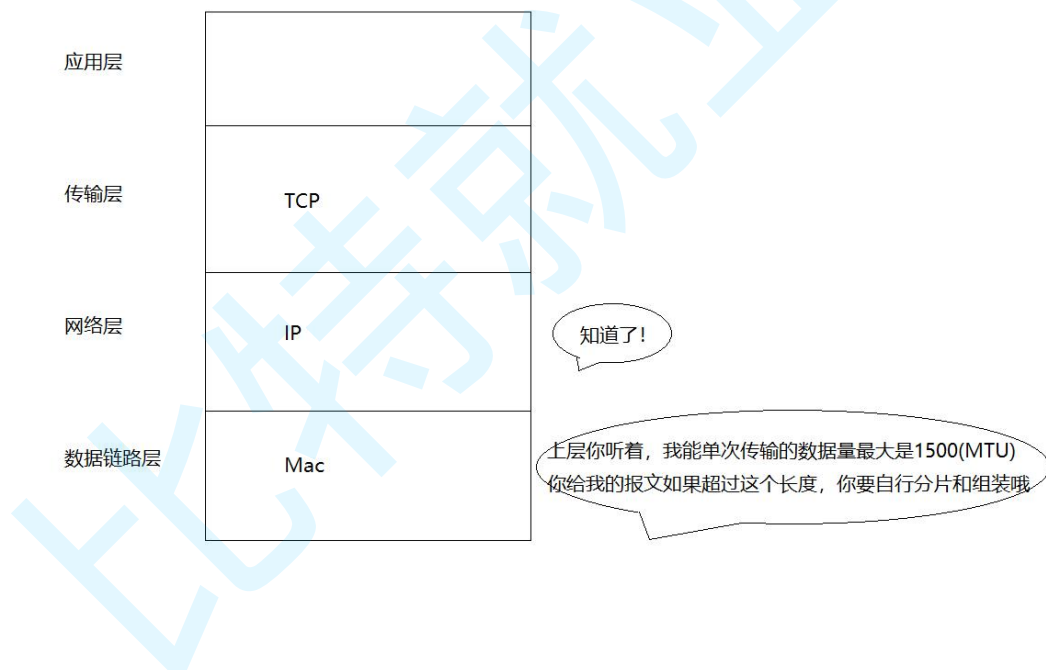
- 组装好的 IP 数据报会传递给上层的协议进行处理。

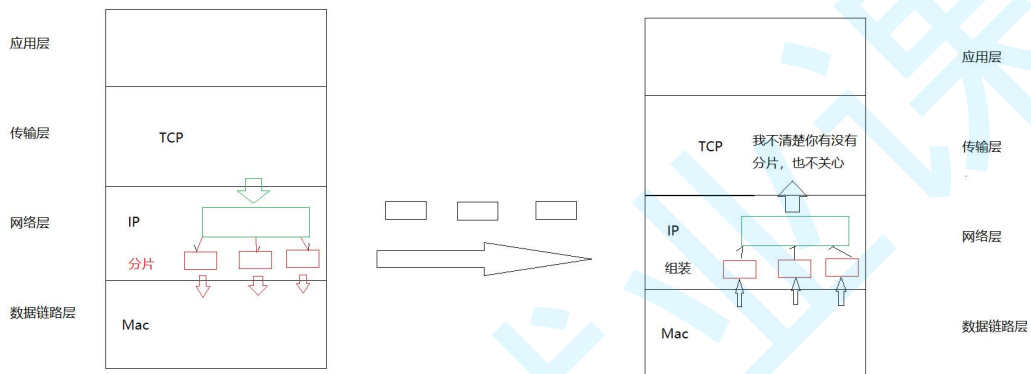
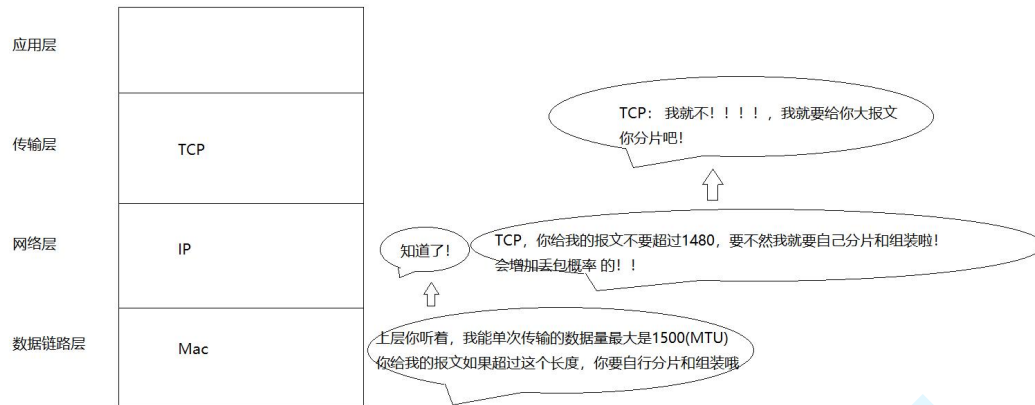
注意：

- IP 分片对传输层是透明的，这意味着传输层无需关心数据是否被分片以及如何重新组装。
- 接收方如何得知自己收到的报文分片了？
- 接收方如何得知自己收到的分片收全了？
- 接收方如何组合形成完整的报文？

分片与组装过程的示意图

分片组装场景





分片组装过程

- 假设在 IP 层，有一个大小为 3000 字节的报文，如何分片？如何组装呢？