链滴

（三）数据链路层

作者：function001

原文链接：https://ld246.com/article/1637505125055

来源网站：链滴

<img src="https://ld246.com/images/img-loading.svg" alt="" data-src="https://b3logfile
com/bing/20210829.jpg?imageView2/1/w/960/h/540/interlace/1/q/100">
<h2 id="一-数据链路层的功能">一，数据链路层的功能</h2>
数据链路层在物理层所提供服务的基础上向网络层提供服务，即将原始的、有差错的物理线路改
成逻辑上无差错的数据链路，从而向网络层提供高质量的服务。它一般包括 3 种基本服务：<strong
无确认的无连接服务、有确认的无连接服务和有确认的有连接服务
具体地说，数据链路层的主要功能如下。

链路管理：负责数据链路的建立、维持和释放，主要用于面向连接的服务。
帧同步：接收方确定收到的比特流中一帧的开始位置与结束位置。
差错控制：用于使接收方确定接收到的数据就是由发送方发送的数据。
透明传输：假设在图 3-1 中的透明传输区间里出现了 01111110 这样的比特组合，也就是与帧
界符相同，岂不是会被误认为是传输结束而丢弃后面的数据？显然，这样的情况
是绝对不允许发生的，于是就发明了透明传输来解决此问题。其实，透明传输就是不管数据是什么样
比特组合，都应当能在链路上传送。

当两个主机互相传送信息时，网络层的分组必须将封装成帧，并以帧的格式进行传送。将一段数
的前后分别添加首部和尾部，就构成了帧。首部和尾部中含有很多控制信息，这些信息的重要作用之
是确定帧的界限，即帧定界。例如，在 HDLC 协议中的帧格式使用标志 F （01111110）来标识帧的
始和结束。
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https:
/b3logfile.com/file/2021/10/image-cf0b38c3.png?imageView2/2/interlace/1/format/jpg"></
>
<h2 id="二-组帧">二，组帧</h2>
4 种组帧方法包括字符计数法、字节填充的首尾界符法、比特填充的首尾标志法、物理编码违例
。
1，字符计数法
字符计数法是用一个特殊的字符来表示一帧的开始，然后用一个计数字段来表明该帧包含的字节
。当目的主机接收到该帧时，根据此字段提供的字节数，便可知道该帧的结束位和下一帧的开始位，
图 3-2 所示。
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https:
/b3logfile.com/file/2021/10/image-9ce5ca30.png?imageView2/2/interlace/1/format/jpg"></
>
字符计数法存在的问题：如果计数字段在传输中出现差错，接收方就无法判断所传输帧的结束位
当然也无法知道下一帧的开始位，这样就无法帧同步了。由于此原因，字符计数法很少被使用。
2，字节填充的首尾界符法
其实可以将其拆开理解，首先讨论一下首尾界符法。 由 C 语言的知识可以知道 ASCII 码是 7 位
码，可以组成 128 个不同的 ASCII 码，但是可以打印（就是可以从键盘输入的字符）的只有 95 个字
，那么当传送的帧是文本文件（都是从键盘输入的）时，就可以在剩下的 33 个控制字符中选定 2 个
符（教材中选用了 SOH 与 EOT 分别作为帧开始符和帧结束符）作为每一帧的开始和结束，这样接收
只需要判断这两个控制字符出现的位置就能准确地分割成帧，如图 3-3 所示。
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https:
/b3logfile.com/file/2021/10/image-7fcff7ac.png?imageView2/2/interlace/1/format/jpg"></p

字符 SOH 代表 Start of Header（首部开始），EOT 代表 End of Transmission（传输结束）。
/p>
这种方式对于帧数据为文本文件是绝对没有问题的。但是还有一种情况就比较麻烦了， 假设要
送的帧不是文本文件，即帧数据部分可能包含控制字符，就不能仅仅使用控制字符去进行帧定界了，
则将会导致错误地“找到帧的边界”，把部分帧收下（误认为是一个完整的帧），如图 3-4 所示。</
>
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https:
/b3logfile.com/file/2021/10/image-1303ec85.png?imageView2/2/interlace/1/format/jpg"></
>
字节填充的首尾界符法设法将数据中可能出现的控制字符“SOH”和“EOT”在接收端

不解释为控制字符。
其方法如下： 在数据中出现字符“SOH”或“EOT”时就将其转换为另一个字符，而这个字符不会被错误解释的。但所有字符都有可能在数据中出现，于是就将数据中出现的字符“SOH”转换为ESC”和“x”两个字符，将数据中出现的字符“EOT”转换为“ESC”和“y”两个字符。而当数据出现了控制字符“ESC”时，就将其转换为“ESC”和“z”两个字符。这种转换方法能够在接收端将到的数据正确地还原为原来的数据。“ESC”是转义符，它的十六进制编码是 1B。 如图 3-5 所示，上方的数据中出现了 4 个控制字符“ESC” “EOT” “ESC”和“SOH”。 按以上规则转换后的数即为图 3-5 下方的数据。

3，比特填充的首尾标志法
比特填充的首尾标志法是使用 01111110 作为帧的开始和结束标志，似乎帧定界又解决了，但如果帧数据部分出现了 01111110 怎么办？透明传输仍然是个问题。其解决方法如下： 不难发现 01111110 中有 6 个连续的“1”，只要数据帧检测到有 5 个连续的“1”，马上在其后插入“0”，而接方做该过程的逆操作，即每收到 5 个连续的“1”，自动删除后面紧跟的“0”，以恢复原始数据。因，此方法又称为零比特填充法，具体可见下面的模拟过程。 模拟过程：

原始数据：01101011111100101111111011 （数据中出现两次 01111110）。
零比特填充后的数据：01101011111210010111112101 1 （加下画线的 0 表示填充的 0）。
接收方收到数据，一旦遇到 5 个连续的“1”就将后面的“0”去掉，即可得到原始数据。

4，物理编码违例法
物理编码违例法利用物理介质上编码的违法标志来区分帧的开始与结束，例如，在曼彻斯特编码，码元 1 编码成高-低电平，码元 0 编码成低-高电平，而高-高和低-低电平的编码方式是无效的，可分别用来作为帧的起始标志和结束标志。
注意：在使用字节填充的首尾界符法时，并不是所有形式的帧都需要帧开始符和帧结束符，如 MC 帧就不需要帧结束符。
<h2 id="三-差错控制">三，差错控制</h2>
<h2 id="1-检错编码">1，检错编码</h2>
<h3 id="1-奇偶校验码">1）奇偶校验码</h3>

奇偶校验码就是在信息码后面加一位校验码，分奇校验和偶校验。 奇校验：添加一位校验码后，得整个码字里面 1 的个数是奇数。接收端收到数据后就校验数据里 1 的个数，若检测到奇数个 1，则为传输没有出错；若检测到偶数个 1，则说明传输过程中，数据发生了改变，要求重发。
偶校验：添加一位校验码后，使得整个码字里面 1 的个数是偶数。接收端收到数据后就校验数据 1 的个数，若检测到偶数个 1，则认为传输没有出错；若检测到奇数个 1,则说明传输过程中，数据发了改变，要求重发。

可见，当数据中有一位数据发生改变时，通过奇偶校验能够检测出来，但并不知道是哪一位出错；如果数据中同时有两位数据发生了改变，此时奇偶校验是检测不到数据出错的， 所以它的查错能力限。
<h3 id="2-循环冗余码">2）循环冗余码</h3>
CRC 算法的基本思想是将传输的数据当做一个位数很长的数。将这个数除以另一个数。得到的数作为校验数据附加到原数据后面。
例：10110010000 / 11001 的结果：

<h2 id="2-纠错编码">2，纠错编码</h2>
纠错编码：在接收端不但能检查错误，而且能纠正检查出来的错误。常见的纠错编码是海明码。
海明码：又称为汉明码，它是在信息字段中插入若干位数据，用于监督码字里的哪一位数据发生

变化，具有一位纠错能力。假设信息位有 k 位，整个码字的长度就是 k+r 位；每一位的数据只有两状态，不是 1 就是 0,有 r 位数据就能表示出 2^ 种状态。如果每一种状态代表一个码元发生了错误， k+r 位码元，就要有 k+r 种状态来表示，另外还要有一种状态来表示数据正确的情况，所以 2r-l^k+r 才能检查一位错误，即 2r^k+r+lo 例如，信息数据有 4 位，由 2r^k+r+l Mr^3,也就是至少需要 3 监督数据才能发现并改正 1 位错误。 例如，给 8 个学员进行编号，可以用 3 位数来编码：学号为 00 、001、...、111；也可以用 5 位数来编码：学号为 00000、0000K 00010、...、00111,但是没有必用 5 位，只要能满足编码的要求就可以了，所以只需求出满足条件的最小的 k 值即可。 海明码求解具体步骤如下：

确定校验码的位数 r。
确定校验码的位置。
确定数据的位置。
求出校验位的值

<h2 id="四-流量控制与可靠传输机制">四，流量控制与可靠传输机制</h2>
<h2 id="1-流量控制">1，流量控制</h2>
流量控制就是要控制发送方发送数据的速率，使接收方来得及接收。一个基本的方法是由接收方制发送方的数据流。常见的有两种方式:停止-等待流量控制和滑动窗流量控制。

停止-等待流量控制

它是流量控制中最简单的形式。停止-等待流量控制的工作原理就是发送方发出一帧，然后等待答信号到达再发送下一帧；接收方每收到一帧后，返回一个应答信号，表示可以接收下一帧，如果接方不返回应答，则发送方必须一直等待。
<ol start="2">
滑动窗口流量控制

滑动窗口流量控制允许一次发送多个帧。滑动窗口流量控制的工作原理就是在任意时刻，发送方维持了一组连续的允许发送的帧的序号， 称为发送窗口。同时，接收方也维持了一组连续的允许接收帧的序号，称为接收窗口。发送窗口和接收窗口的序号的上下界不一定要一样，甚至大小也可以不同发送方窗口内的序列号代表了那些已经被发送但是还没有被确认的帧，或者是那些可以被发送的帧。送端每收到一个帧的确认，发送窗口就向前滑动一个帧的位置。当发送窗口尺寸达到最大尺寸时， 发方会强行关闭网络层，直到有一个空闲缓冲区出来。在接收端只有当收到的数据帧的发送序号落入接窗口内才允许将该数据帧收下，并将窗口前移一个位置。若接收到的数据帧落在接收窗口之外（就是收到的帧号在接收窗口中找不到相应的该帧号），则一律将其丢弃。
<h2 id="2-滑动窗口协议">2，滑动窗口协议</h2>
只有在接收窗口向前滑动时（与此同时也发送了确认），发送窗口才有可能向前滑动。
可靠传输机制包括停止-等待协议、后退 N 帧协议和选择重传协议。
从滑动窗口的层次上看，该 3 种协议只是在发送窗口和接收窗口大小上有所差别。
停止-等待协议：发送窗口大小=1，接收窗口大小=1。
后退 N 帧协议：发送窗口大小 >1，接收窗口大小=1。
选择重传协议：发送窗口大小 >1，接收窗口大小 > 1。
当接收窗口的大小为 1 时，一定可保证帧按序接收。
<h2 id="3-停止等待协议">3，停止等待协议</h2>
发送方传输一个帧后，必须等待对方的确认才能发送下一帧。 若在规定时间内没有收到确认，发送方超时，并重传原始帧。看到这里也许有人会问，停止-等待流量控制技术（这里是停止-等待流控制技术而不是停止-等待协议）为什么要一直等待？为什么不设置一个规定时间？这里就要回到第 1 章协议的制定。首先协议需要建立在一定技术（停止-等待流量控制技术）之上，然后在此技术之上要考虑一切可能突发的不利状况（可以这么理解：协议=技术 + 考虑不利因素，即停止-等待协议=停-等待流量控制技术 + 不利因素），设置规定时间重传就是为了解决这些不利因素。如果不设置时间会造成死锁， 这样就无法推进，在这里可以联系到操作系统的死锁，如果没有外力参与去打破死锁，会一直等待下去，而这里的外力就是重传计时器。
<h2 id="4-后退N帧-GBN--协议">4，后退 N 帧（GBN ）协议</h2>

后退 N 帧协议基于滑动窗口流量控制技术。若采用 n 个比特对帧进行编号，其发送窗口尺寸 Wt 必须满足 1 < Wt <= 2 的 n 次方 - 1　接收窗口尺寸为 1。若发送窗口尺寸大于 2 的 n 次方 - 1 会造成接收方无法分辨新、旧数据帧的问题。由于接收窗口尺寸为 1，因此接收方只能按序来接收数帧。
后退 N 帧协议的基本原理：发送方发送完一个数据帧后，不是停下来等待确认帧，而是
可以连续再发送若干个数据帧。如果这时收到了接收方的确认帧，那么还可以接着发送数据
帧。如果某个帧出错了，接收方只能简单地丢弃该帧及其所有的后续帧。发送方超时后需重
发该出错帧及其后续的所有帧。由于减少了等待时间，后退 N 帧协议使得整个通信的吞吐量
得到提高。但接收方一发现错误帧，就不再接收后续的帧，造成了一定的浪费。据此改进， 得到了选重传协议。
<h2 id="5-选择重传-SR--协议">5，选择重传（SR ）协议</h2>
选择重传协议也是基于滑动窗口流量控制技术的。它的接收窗口尺寸和发送窗口尺寸都
大于 1,以便能一次性接收多个帧。若采用 n 个比特对帧进行编号，为避免接收端向前移动
窗口后，新接收窗口与旧接收窗口产生重叠，发送窗口的最大尺寸应该不超过序列号范围的
一半：Wt <= 2 的(n-1)次方 （请参考下面的补充知识点）。当发送窗口取最大值时，WR=Wt=2 的(n-1)次方（大部分情况都是发送窗口等于接收窗口，且等于 2 的(n-1)次方，因为这样可达到最大率，记住就好）。此时，若 WT 取大于 2 的(n-1)次方的值，可能造成新、旧接收窗口重叠。
选择重传协议的基本思想：若一帧出错，则其后续帧先存入接收方的缓冲区中，同时要求发送方传出错帧，一旦收到重传帧后，就和原先存在缓冲区的其余帧一起按正确的顺序送至主机。选择重传议避免了重复传输那些本来已经正确到达接收方的数据帧，进一步提高了信道利用率，但代价是增加缓冲空间。
<h2 id="6-发送缓存和接收缓存">6，发送缓存和接收缓存</h2>

从图 3-10b 中可以看到，按序到达的且没有被交付给主机的帧被放在接收缓存（接收窗
口外的那一部分接收缓存，以下讲的接收缓存都是指这部分）里面（因为已经发送过确认帧
了，仅仅是等主机的应用程序来取），而不是接收窗口里面。那些不是按序到达的数据且没有错误的一定是要放在接收窗口里面，因为这些帧不能直接给主机，而放在接收缓存的帧是要给主机的，等到少的帧收到后，再一起放到接收缓存。
缓存就是在计算机的存储器中设置的一个临时存放数据的空间。发送进程将欲发送的数据先写入存，然后接收进程在合适的时机读出这些数据。
<h2 id="五-介质访问控制">五，介质访问控制</h2>
<h2 id="1-介质访问控制分类">1，介质访问控制分类</h2>
介质访问控制分为以下 3 类：

信道划分介质访问控制。
随机访问介质访问控制。
轮询访问介质访问控制。

其中，1）是静态分配信道的方法，而 2）和 3）是动态分配信道的方法。
<h2 id="2-信道划分介质访问控制">2，信道划分介质访问控制</h2>
首先介绍多路复用技术的基本概念。当传输介质的带宽超过了传输单个信所需的带宽时，人们就通过在一条介质上同时携带多个传输信号的方法来提高传输系统的利用率，这
是所谓的多路复用，也是实现信道划分介质访问控制的途径。多路复用技术能把多个信号组
合在一条物理信道上进行传输，使多个计算机或终端设备共享信道资源，从而提高了信道的
利用率。

频分多路复用

将一条信道分割成多条不同频率的信道，就类似于将一条马路分割成多个车道，尽管同一时间车都在这条马路上行驶，但是分别行驶在不同的车道上，所以不会发生冲突。现在假设每个车道的宽度能改变了，但是需要加车道，所以马路就必须变宽，类似于使用频分复用时，如果复用数增加，那么道的带宽（此时的带宽是频率带宽，不是数据的发送速率）必须得增加。
<ol start="2">
时分多路复用

假设现在只有一个玩具，却有 10 个小孩要玩，这时候只能将一个固定的时间分割成 10 份，10 小孩轮流玩这个玩具，即时分多路复用。 所以当使用时分多路复用时，复用数加并不需要加大信道带宽，只需将每个信道分得的时间缩小即可。也许很多人在这里会有疑问，如果好某个时间轮到一个小孩玩了，但是这个小孩现在睡着了，岂不是这段时间就浪费了吗？没错，是浪了，这时候就需要把时分复用改进，于是引入统计时分复用。继续上面的例子，在如果该玩具轮到某个小孩玩，但是他睡着了，立刻跳过他，给下一个小孩玩，这样就基本可以保证具没有空闲时刻。可见每个孩子下次轮到自己玩的时间都是不确定的，如果睡觉的人多了，很快就轮了；如果睡觉的人少，就很慢。因此，统计时分复用是一种动态的时间分配，同时又是异步的（每个子玩玩具的时间周期是不固定的），所以统计时分复用又称为异步时分复用。普通的时分复用就是同步时分复用（因为每个孩子都在一个固定的周期才能得到玩具，即使中间有孩睡觉也要等）。
<ol start="3">
波分多路复用

波分多路复用就是光的频分多路复用，在一根光纤中传输多种不同频率（波长）的光信号，由于路光的频率（波长）不同，因此各路光信号不互相干扰。最后，再用分波器将各路波长不一样的光分出来，

/b3logfile.com/file/2021/10/image-9dcac717.png?imageView2/2/interlace/1/format/jpg"></
>
<ol start="4">
码分多路复用

码分多路复用又称为码分多址（CDMA），它既共享信道的频率，又共享时间，是一种真正的动复用技术。
每个站点都维持一个属于该站点的芯片序列，并且是固定的。假如站点 A 的芯片序列为 000110 1,则 A 站点发送 00011011 表示发送比特 1;而将 00011011 每位取反，即发送 11100100 表示发送特 0。习惯将芯片序列中的 0 写为-1，1 写为 +1,所以 A 站的芯片序列就是(-I -1-1 +1 +1 -1 +1 +1),一般将该向量称为该站的码片向量。

任意两个不同站的码片向量正交，即任意两个站点的码片向量的规格化内积一定为 0。
任意站点的码片向量与该码片向量自身的规格化内积一定为 1;任何站点的码片向量
和该码片的反码向量的规格化内积一定为-1

总结：码分多路复用技术具有抗干扰能力强、保密性强、语音质量好等优点，还可以减
少投资和降低运行成本，主要用于无线通信系统，特别是移动通信系统。CDMA 手机就使用
了此技术。
<h2 id="3-随机访问介质访问控制">3，随机访问介质访问控制</h2>
随机接入的意思是所有用户都可以根据自己的意愿随机地发送信息，这样会产生冲突
(或者称为碰撞)，从而导致所有冲突用户发送数据失败。为了解决随机接入发生的碰撞， CSMA/CD 协议被引入。
受控接入就是不能随机地发送数据，一定要得到某种东西才有权发数据。<
p>
随机接入即 ALOHA 协议、CSMA 协议、CSMA/CD 协议和 CS A/CA 协议。 以上 4 种协议的核心思想是通过争用，胜利者才可以获得信道，从而获得信的发送权。 正因为这种思想，随机访问介质访问控制又多了一个绰号：争用型协议</stron

>。
<h2 id="4-轮询访问介质访问控制">4，轮询访问介质访问控制</h2>
轮询访问介质访问控制主要用在令牌环局域网中，目前使用得很少。在轮询访问介质访问控制中用户不能随机地发送信息，而是通过一个集中控制的监控站经过轮询过程后再决定信道的分配。典型轮询访问介质访问控制协议就是令牌传递协议。
<h2 id="六-局域网">六，局域网</h2>
<h2 id="1-局域网的基本概念与体系结构">1，局域网的基本概念与体系结构</h2>
局域网（Local Area Network，LAN）是指一个较小范围（如一个公司）内的多台计算机或者他通信设备，..通过双绞线、同轴电缆等连接介质互连起来，以达到资源和信息共享目的的互联网络。/p>
局域网最主要的特点

局域网为一个单位所拥有（如学校的一个系使用一个局域网）。
地理范围和站点数目有限（双绞线的最大传输距离为 100m,如果要加大传输距离， 则在两段双线之间安装中继器，最多可安装 4 个中继器。例如，安装 4 个中继器连接 5 个网段，则最大传输距可达 500m，所以地理范围有限。局域网一般可以容纳几台至几千台计算机，所以站点数目有限）。
与以前非光纤的广域网相比，局域网具有较高的数据率、较低的时延和较小的误码率
（现在局域网的数据率可以达到万兆了；传输距离较短所以时延小；距离短了失真就小，误码率自然低）。

局域网的主要优点

具有广播功能，从一个站点可很方便地访问全网。局域网上的主机可共享连接在局域网上的各种件和软件资源。
便于系统的扩展和演变，各设备的位置可灵活地调整和改变。
提高了系统的可靠性、可用性。
各站为平等关系而不是主从关系。

局域网的主要技术要素
局域网的主要技术要素包括网络拓扑结构、传输介质与介质访问控制方法。其中，介质
访问控制方法是最为重要的技术要素，决定着局域网的技术特性。
局域网的主要拓扑结构
局域网的主要拓扑结构包括星形网、环形网、总线型网和树形网（星形网和总线型网的结合）。
局域网的主要传输介质
局域网的主要传输介质包括双绞线、铜缆和光纤等，其中双绞线为主流传输介质。
局域网的主要介质访问控制方法
局域网的主要介质访问控制方法包括 CSMA/CD、令牌总线和令牌环。前两种作用于总
线型网，令牌环作用于环形网。IEEE 的 802 标准定义的局域网参考模型只对应于 OSI 参考模
型的数据链路层和物理层，并且将数据链路层拆分为两个子层：逻辑链路控制（LLC）子层
和媒体接入控制（MAC）子层。与接入到传输媒体有关的内容都放在 MAC 子层，而 LLC 子
层与传输媒体无关。
<h2 id="2-以太网工作原理">2，以太网工作原理</h2>
以太网采用总线拓扑结构，所有计算机都共享一条总线，信息以广播方式发送。为了保证数据通的方便性和可靠性，以太网使用了 CSMA/CD 技术对总线进行访问控制。考虑到局域网信道质量好以太网采取了以下两项重要的措施以使通信更加简便。

采用无连接的工作方式。
不对发送的数据帧进行编号，也不要求对发送方发送确认。 因此以太网提供的服务是不可靠的服，即尽最大努力交付，差错的纠正由传输层的 TCP 完成

<h2 id="3-以太网的MAC帧">3，以太网的 MAC 帧</h2>
局域网中的每台计算机都有一个唯一的号码，称为 MAC 地址或物理地址、硬件地址。 每块网卡厂即被赋予一个全球唯一的 MAC 地址，它被固化在网卡的 ROM 中，共 48bit(6B)。

由于总线上使用的是广播通信，因此网卡从网络上每收到一个 MAC 帧，先要用硬件检查 MAC 中的 MAC 地址，如果是发往本站的帧就收下，否则丢弃。

MAC 帧组成部分的详细分析如下。

前导码：在帧的前面插入 8B，使接收端与发送端进行时钟同步。这 8B 又可分为前同
步码(7B)和帧开始定界符(1B)两部分。

注意：在讲解组帧的时候，特意说明了 MAC 帧不需要帧结束符，因为以太网在传送帧时，各帧间必须有一定的间隙。因此，接收端只要找到帧开始定界符，其后面连续到达的比特流就都属于同一个 MAC 帧，所以图 3-13 中只有帧开始定界符。
<ol start="2">
目的地址、源地址:均使用 48bit (6B)的 MAC 地址。

注意：地址字段包括目的地址和源地址两部分。处于前面的地址字段为且的地址，处于后面的地字段为源地址。
<ol start="3">
类型：占 2B,指出数据域中携带的数据应交给哪个协议实体处理。例如，若类型字
段的值为 0x0800,就表示上层使用的是 IP 数据报等，这个无需记忆，知道是怎么回事就行。
数据：占 46∽1500Bo 46 和 1500 是怎么来的？首先，由 CSMA/CD 算法可知，以太
网帧的最短帧长为 64B，而 MAC 帧的首部和尾部的长度为 18B，所以数据最短为 64B - 18B=46B 其次，最大的 1500B 是规定的，没有为什么。
填充：前面讲过，由于 CSMA/CD 算法的限制，最短帧长为 64B，因此除去首部 18B， 如果数长度小于 46B，那么就得填充，使得帧长不小于 64B。当数据字段长度小于 46B 时， 需要填充至 46B 当数据字段长度大于或等于 46B 时，则无需填充。因此，填充数据长度的范围为 0∽46B。
校验码（FCS）：占 4B,采用循环冗余码，不但需要校验 MAC 帧的数据部分，还要校验目的地、源地址和类型字段，但是不校验前导码。

<h2 id="4-以太网的传输介质">4，以太网的传输介质</h2>
传统以太网可使用的传输介质有 4 种，即粗缆、细缆、双绞线和光纤。对应的，MAC 层下面给了这 4 种传输介质的物理层，即 10Base5 （粗缆）、10Base2 （细缆）、10Base-T （双
绞线）和 10Base-F （光纤）。其中，Base 指电缆上的信号为基带信号，采用曼彻斯特编码； Base 面的 10 表示数据传输速率为 10Mbit/s；Base 后面的 5 或 2 表示每一段电缆最长为 500m
或 200m （实为 185m）； T 表示双绞线，F 表示光纤。

<h2 id="七-广域网">七，广域网</h2>
<h2 id="1-概念">1，概念</h2>
广域网通常是指覆盖范围很广（远远超出一个城市的范围）的长距离网络。广域网由一
些节点交换机以及连接这些交换机的链路组成。节点交换机将完成分组存储转发的功能。互
联网虽然覆盖范围也很广，但一般不称它为广域网。因为在这种网络中，不同网络（可以为
局域网也可以为广域网）的“互连”才是其最主要的特征，它们之间通常采用路由器来连接。 而广域只是一个单一的网络，它使用节点交换机连接各主机而不是用路由器连接各网络。 虽然节点交换机和由器都是用来转发分组的，它们工作原理也类似，但区别是节点交换机在单个网络中转发分组，而路器在多个网络构成的互联网中转发分组。广域网和局域网的区别与联系见表 3-5。

注意:
1)从层次上考虑，广域网和局域网的区别很大，因为局域网使用的协议主要在数据链路
层(包含少量物理层的内容)，而广域网使用的协议主要在网络层。

2)广域网中存在一个最重要的问题，即路由选择和分组转发。路由选择协议负责搜索分
组从某个节点到目的节点的最佳传输路由，以便构造路由表。从路由表再构造出转发分组的
转发表，分组是通过转发表进行转发的。
为了方便理解，可以将广域网看成一个大的局域网，从专业角度来讲就是通过交换机连接多个局网，组成更大的局域网，即广域网。因此，广域网仍然是一个网络。而因特网是多个网络之间互连，因特网由大局域网（广域网）和小局域网共同通过路由器相连。因此，局域网就可通过因特网与另一相隔很远的局域网进行通信。
<h2 id="2-PPP">2，PPP</h2>
PPP 主要由下面三部分组成：

一个将 IP 数据报封装到串行链路的方法。
一个链路控制协议（LCP）。其用于建立、配置和测试数据链路连接，并在不需要时
将它们释放。
一套网络控制协议（NCP）。其中每个协议支持不同的网络层协议，用来建立和配置
不同的网络层协议

PPP 的帧格式：

/b3logfile.com/file/2021/10/image-e6f9c46e.png?imageView2/2/interlace/1/format/jpg"></
>

标志字段（F）：首部和尾部各占 1 个字节，规定为 Ox7E。
地址字段（A）:占 1 个字节，规定为 OxFF,没有为什么。
控制字段（C）:占 1 个字节，规定为 0x03，没有为什么。
协议字段：占 2 个字节。例如，当协议字段为 0x0021 时，PPP 帧的信息字段就是 IP
数据报；若为 0xC021,则信息字段是 PPP 链路控制数据；若为 0x8021,则表示这是网络控
制数据。
信息部分：占 0∽1500 个字节。为什么不是 46∽1500 个字节？因为 PPP 是点对点的， 并不是线型，所以无需采用 CSMA/CD 协议，自然就没有最短帧。另外，当数据部分出现
和标志位一样的比特组合时，就需要采用一些措施来实现透明传输（上面的补充知识点已讲）。
帧检验序列（FCS）:占 2 个字节，即循环冗余码检验中的冗余码。检验区间包括地
址字段、控制字段、协议字段和信息字段。

PPP 的工作状态
当用户拨号接入 ISP 时，路由器的调制解调器对拨号做出确认，并建立一条物理连接。 这时，人计算机向路由器发送一系列的 LCP 分组（封装成多个 PPP 帧）。这些分组及其响应选择了将要使的一些 PPP 参数。接着就进行网络层配置，网络控制协议（NCP）给新接入的个人计算机分配一个时的 IP 地址。这样，个人计算机就成为因特网上的一个主机了。 当用户通信完毕时，NCP 释放网络连接，收回原来分配出去的 IP 地址。接着，LCP 释放数据链路层连接，最后释放物理层连接。
总结：

PPP 是一个面向字节的协议。
PPP 不需要的功能：纠错（PPP 只负责检错）、流量控制（由 TCP 负责）、序号（PPP
是不可靠传输协议，所以不需要对帧进行编号）、多点线路（PPP 是点对点的通信方式）、半双工或工（PPP 只支持全双工链路）。
