

8B10B编码详解

8B/10B，也叫做8字节/10字节或8B10B。8B/10B方式最初由IBM公司于1983年发明并应用于ESCON(200M互连系统)，由Al Widmer和Peter Franaszek在IBM的刊物“研究与开发”上描述。

8b/10b编码的特性之一是保证DC平衡，采用8b/10b编码方式，可使得发送的“0”、“1”数量保持基本一致，连续的“1”或“0”不超过5位，即每5个连续的“1”或“0”后必须插入一位“0”或“1”，从而保证信号DC平衡，它就是说，在链路超时不致发生DC失调。通过8b/10b编码，可以保证传输的数据串在接收端能够被正确复原，除此之外，利用一些特殊的代码(在PCI-Express总线中为K码)，可以帮助接收端进行还原的工作，并且可以在早期发现数据位的传输错误，抑制错误继续发生。

8b/10b编码是将一组连续的8位数据分解成两组数据，一组3位，一组5位，经过编码后分别成为一组4位的代码和一组6位的代码，从而组成一组10位的数据发送出去。相反，解码是将1组10位的输入数据经过变换得到8位数据位。数据值可以统一的表示为DX.Y或KX.Y，其中D表示为数据代码，K表示为特殊的命令代码，X表示输入的原始数据的低5位EDCBA，Y表示输入的原始数据的高3位HGF。

8b/10b编码是目前许多高速串行总线采用的编码机制，如USB3.0、1394b、Serial ATA、PCI Express、Infini-band、Fiber Channel、RapidIO等总线或网络等。

8B/10B编码是目前高速串行通信中经常用到的一种编码方式。直观的理解就是把8bit数据编码成10bit来传输，为什么要引入这种机制呢？其根本目的是“直流平衡（DC Balance）”。当高速串行流的逻辑1或逻辑0有多个位没有产生变化时，信号的转换就会因为电压位阶的关系而造成信号错误，直流平衡的最大好处便是可以克服以上问题。

将8bit编码成10bit后，10B中0和1的位数只可能出现3种情况：

- 1.有5个0和5个1
- 2.有6个0和4个1
- 3.有4个0和6个1

这样引出了一个新术语“不均等性（Disparity）”，就是1的位数和0的位数的差值，根据上面3种情况就有对应的3个Disparity 0、-2、+2。

工作原理

8bit原始数据会分成两部分，其低5位会进行5B/6B编码，高3位则进行3B/4B编码，这两种映射关系在当时已经成为了一个标准化的表格。人们喜欢把8bit数据表示成Dx.y的形式，其x=5LSB(least significant bit最低有效位)，y=3MSB(most significant bit最高有效位)。

例如一个8bit数据101 10101，x=10101(21) y=101(5)，现在我们就把这8bit数据写成D21.5，明白了吧！

Dx.y形式在进行5B/6B和3B/4B编码中表示更直观，下面我们来看看两张编码表：

对于8bit数据，它在表中的位序为HGFEDCBA，即H为最高位，A为最低位，EDCBA经过5B/6B编码为abcdei，HGF经过3B/4B编码为fghj。传送10bit编码的顺序为abcdeifghj。

5B/6B code							
input		RD = -1	RD = +1	input		RD = -1	RD = +1
	EDCBA	abcdei			EDCBA	abcdei	
D.00	00000	100111	011000	D.16	10000	011011	100100
D.01	00001	011101	100010	D.17	10001	100011	
D.02	00010	101101	010010	D.18	10010	010011	
D.03	00011	110001		D.19	10011	110010	
D.04	00100	110101	001010	D.20	10100	001011	
D.05	00101	101001		D.21	10101	101010	
D.06	00110	011001		D.22	10110	011010	
D.07	00111	111000	000111	D.23 †	10111	111010	000101
D.08	01000	111001	000110	D.24	11000	110011	001100
D.09	01001	100101		D.25	11001	100110	
D.10	01010	010101		D.26	11010	010110	
D.11	01011	110100		D.27 †	11011	110110	001001
D.12	01100	001101		D.28	11100	001110	
D.13	01101	101100		D.29 †	11101	101110	010001
D.14	01110	011100		D.30 †	11110	011110	100001
D.15	01111	010111	101000	D.31	11111	101011	010100
				K.28	11100	001111	110000

8B/10B编码 <wbr>详解zz

† 3B/4B使用K.x.7

3b/4b code

input		RD = -1	RD = +1	input		RD = -1	RD = +1
	HGF	fghj			HGF	fghj	
D.x.0	000	1011	0100	K.x.0	000	1011	0100
D.x.1	001	1001		K.x.1 ‡	001	0110	1001
D.x.2	010	0101		K.x.2 ‡	001	1010	0101
D.x.3	011	1100	0011	K.x.3	011	1100	0011
D.x.4	100	1101	0010	K.x.4	100	1101	0010
D.x.5	101	1010		K.x.5 ‡	001	0101	1010
D.x.6	110	0110		K.x.6 ‡	001	1001	0110
D.x.P7 †	111	1110	0001				
D.x.A7 †	111	0111	1000	K.x.7 †‡	111	0111	1000

8B/10B编码 <wbr>详解zz

† 对于D.x.7，当和5B/6B组合时D.x.P7和D.x.A7编码必须选择一个来避免连续的5个0或1。遇上连续5个0或1的情况下使用“逗号码”来进行校准。D.x.A7用在x=17 x=18 x=20当RD=-1时，x=11 x=13 x=14 当RD=+1时。当x=23 x=27 x=29 x=30时，使用K.x.7进行编码。其他情况下x.A7码不能被使用，他将会导致和其他“逗号序列”产生冲突。

‡ 候补编码K.x.y允许K.28.1 K.28.5 K.28.7作为“逗号码”来保证数据流中的唯一性。

你们也许注意到了表中有个RD标志，它是Running Disparity的缩写，它的目的就是保持8B/10B编码中的直流平衡。它跟上面提到的Disparity其实是一样的意思，+1用来表示1比0多，-1用来表示0比1多，-1是它的初始化状态。下面我们来看一张表来加深理解：

Rules for Running Disparity

Previous RD	Disparity of 6 or 4 Bit Code	Next RD
-1	-2	not used (choose +2 encoding instead)
-1	0	-1
-1	+2	+1
+1	-2	-1
+1	0	+1
+1	+2	not used (choose -2 encoding instead)

8B/10B编码 <wbr>详解zz

上面我们提到的“逗号码”和“逗号序列”，其实都是当初在规划8B/10B编码机制的时候，所谓的控制代码（Control Characters）的其中之一。8B/10B标准中使用了12个特殊的控制代码，他们能在数据中被发送，还可以组合成各种“原语”。

Control symbols			
input		RD = -1	RD = +1
	HGF EDCBA	abcdei fghj	abcdei fghj
K.28.0	000 11100	001111 0100	110000 1011
K.28.1 †	001 11100	001111 1001	110000 0110
K.28.2	010 11100	001111 0101	110000 1010
K.28.3	011 11100	001111 0011	110000 1100
K.28.4	100 11100	001111 0010	110000 1101
K.28.5 †	101 11100	001111 1010	110000 0101
K.28.6	110 11100	001111 0110	110000 1001
K.28.7 ‡	111 11100	001111 1000	110000 0111
K.23.7	111 10111	111010 1000	000101 0111
K.27.7	111 11011	110110 1000	001001 0111
K.29.7	111 11101	101110 1000	010001 0111
K.30.7	111 11110	011110 1000	100001 0111

8B/10B编码 <wbr>详解zz

† 在控制代码中，K.28.1 K.28.5 K.28.7 是逗号序列，逗号序列是用来校准用的，如果K.28.7没有被使用，序列00111111 或者 11000000 是不会出现在任何编码中的。
‡ 在实际编码中如果K.28.7可以被使用，一种更复杂的校准规范需要†被使用，它们能组合成各种“原语”，在任何情况下多个K.28.7序列不允许被同时使用，它将导致不可探测的逗号序列。

提定端口就是根端口到达根桥路径上的非根端口和离根网桥最近的网段上的端口

生成树协议主要有两个重要的作用：1、避免在二层交换网络中产生路径回环 2、能够在二层交换网络中实现冗余备份。本文将为您详细介绍生成树协议配置中的选举过程。

第一步:选举根桥(Root Bridge)

在一个生成树域内, 开始的时候, 每台交换机都认为自己是根桥,都发送BPDU, 用于选举根桥。选举根桥, 根据BridgeID的值。Bridge ID由两个部分组成:

- 交换机的优先级(priority), 在Cisco交换机上默认为32768(长度为2个字节);
- 交换机的MAC地址(长度为6个字节);

选完Root之后,只有Root Bridge可以周期性的发送BPDU,所有Nroot没有资格发送BPDU.

第二步: 选举根端口(RootPort)

首先介绍STP的端口成本

连接类型	新成本值	原成本值
10Gbit/s	2	1
1Gbit/s	4	1
100Mbit/s	19	10
10Mbit/s	100	100

路径成本计算方法: 路径成本是从根桥计算的。通常是从根桥到拓扑中其他交换机的端口成本累加。

根桥通过其接口通告BPDU时, BPDU帧中的默认路径成本值是0.连接的交换机接收此BPDU时, 会将本地传入端口的成本加到路径成本中。如果端口是一个快速以太网端口, 那么路径成本的计算方式如下: 0(根桥的路径成本)+19(交换机的端口成本)=19.在将BPDU通告到它之后的交换机之后, 该交换机将包含更新后的路径成本。因为BPDU从根交换机传播得越来越远, 所以累加的路径成本值会越来越高。

此处要注意路径成本是按端口成本累加的。如果一条千兆链路跑百兆网络, 端口是G口的话, 成本还是按4算。

根端口的定义：非根桥用于到达根桥的路径成本最小的端口。

需要注意的是，根桥本身绝对不会有根端口，因为它是根桥，因此它不需要到达自身的端口。

选择根端口的步骤：

- 1.有两条或者多条到达根交换机的路径时，选择具有到达根交换机的最低累加路径成本的路径；
- 2.如果到达根交换机的多条可用路径具有相同的累加路径成本，那么交换机会选择具有最低Bridge ID的相邻交换机(通过交换机可到达根桥)；
- 3.如果多条路径都通过相同的相邻交换机，那么会选择具有最低优先级值的本地端口；
- 4.如果端口优先级值相同，那么会选择交换机上具有最低物理编号的端口。例如在2960系列交换机上，该端口将是FastEthernet0/1或者Gigabit0/1.

完成此步骤后，非根交换机将有一个且仅有一个端口成为其根端口。

第三步：选举指定端口(DesignatedPort)

每台交换机都有一个用来到达根交换机的根端口。除了每台交换机有一个根端口外，每个网段(Segment，应该说成介质比较准确，可以理解成每段线路)还有一个用来到达根交换机的端口，该端口成为指定端口(Designated Port)。

注意：根桥上的每个活动端口都是指定端口。因为所连网段到达根交换机的成本是0，即最低累加成本值。换句话说，这些LAN网段中的每个网段中的每个网段都直接与根交换机相连，因此，这些网段到达根交换机就没有任何成本。

如果一条链路上有Root Port,那么对端一定是Designated Port.

如果一条链路上没有Root Port,那么Designated Port对端一定是Nondesignated Port.

下面列出在所有交换机链路两端选举一个唯一的DP的步骤：

1.第一步，比较链路两端的接口的发送方向的COP(Cost of Port),谁小谁是DP

2.第二步，比较链路两端接口的发送者BID，谁小谁DP

3.第三步，除非接口自己连自己，否则不用，该步骤比较链路两端接口的发送者Port-ID，谁小谁DP。

其他的既不是根端口也不是指定端口的端口就是NDP(Nondesignated Port)

Relationship between Bandwidth, Data Rate and Channel Capacity

This posts describes the relationship between signal bandwidth, channel bandwidth and maximum achievable data rate. Before, going into detail, knowing the definitions of the following terms would help:

- **Signal Bandwidth** – the bandwidth of the transmitted signal or the range of frequencies present in the signal, as constrained by the transmitter.
- **Channel Bandwidth** – the range of signal bandwidths allowed by a communication channel without significant loss of energy (attenuation).
- **Channel Capacity** or **Maximum Data rate** – the maximum rate (in bps) at which data can be transmitted over a given communication link, or channel.

In general, information is conveyed by change in values of the signal in time.

Since frequency of a signal is a direct measure of the rate of change in values of the signal, **the more the frequency of a signal, more is the achievable data rate or information transfer rate**. This can be illustrated by taking the example of both an analog and a digital signal.

If we take analog transmission line coding techniques like Binary ASK, Binary FSK or Binary PSK, information is tranferred by altering the property of a high frequency carrier wave. If we increase the frequency of this carrier wave to a higher value, then this reduces the bit interval $T (= 1/f)$ duration, thereby enabling us to transfer more bits per second.

Similarly, if we take digital transmission techniques like NRZ, Manchester

encoding etc., these signals can be modelled as periodic signals and hence is composed of an infinite number of sinusoids, consisting of a fundamental frequency (f) and its harmonics. Here too, the bit interval (T) is equal to the reciprocal of the fundamental frequency ($T = 1/f$). Hence, if the fundamental frequency is increased, then this would represent a digital signal with shorter bit interval and hence this would increase the data rate.

So, whether it is analog or digital transmission, an increase in the bandwidth of the signal, implies a corresponding increase in the data rate. For e.g. if we double the signal bandwidth, then the data rate would also double.

In practice however, we cannot keep increasing the signal bandwidth infinitely. The telecommunication link or the communication channel acts as a police and has limitations on the maximum bandwidth that it would allow. Apart from this, there are standard transmission constraints in the form of different channel noise sources that strictly limit the signal bandwidth to be used. So the achievable data rate is influenced more by the channel's bandwidth and noise characteristics than the signal bandwidth.

Nyquist and Shannon have given methods for calculating the channel capacity (C) of bandwidth limited communication channels.

Nyquist Criteria for maximum data rate for noiseless channels

Given a noiseless channel with bandwidth B Hz., Nyquist stated that it can be used to carry atmost $2B$ signal changes (symbols) per second. The converse is also true, namely for achieving a signal transmission rate of $2B$ symbols per second over a channel, it is enough if the channel allows signals with frequencies upto B Hz.

Another implication of the above result is the **sampling theorem**, which states that for a signal whose maximum bandwidth is f Hz., it is enough to sample the signals at $2f$ samples per second for the purpose of quantization (A/D conversion) and also for reconstruction of the signal at the receiver (D/A conversion). This is because, even if the signals are sampled at a higher rate than $2f$ (and thereby including the higher harmonic components), the channel would anyway filter out those higher frequency components.

Also, symbols could have more than two different values, as is the case in line

coding schemes like QAM, QPSK etc. In such cases, each symbol value could represent more than 1 digital bit.

Nyquist's formulae for multi-level signalling for a noiseless channel is

$$C = 2 * B * \log M,$$

where C is the channel capacity in bits per second, B is the maximum bandwidth allowed by the channel, M is the number of different signalling values or symbols and log is to the base 2.

For example, assume a noiseless 3-kHz channel.

1. If binary signals are used, then $M = 2$ and hence maximum channel capacity or achievable data rate is $C = 2 * 3000 * \log 2 = 6000$ bps.
2. Similarly, if QPSK is used instead of binary signalling, then $M = 4$. In that case, the maximum channel capacity is $C = 2 * 3000 * \log 4 = 2 * 3000 * 2 = 12000$ bps.

Thus, theoretically, by increasing the number of signalling values or symbols, we could keep on increasing the channel capacity C indefinitely. But however, in practise, no channel is noiseless and so we cannot simply keep increasing the number of symbols indefinitely, as the receiver would not be able to distinguish between different symbols in the presence of channel noise.

It is here that Shannon's theorem comes in handy, as he specifies a maximum theoretical limit for the channel capacity C of a noisy channel.

Shannon's channel capacity criteria for noisy channels

Given a communication channel with bandwidth of B Hz. and a signal-to-noise ratio of S/N, where S is the signal power and N is the noise power, Shannon's formulae for the maximum channel capacity C of such a channel is

$$C = B \log (1 + S/N)$$

(log is to base 2)

For example, for a channel with bandwidth of 3 KHz and with a S/N value of 1000, like that of a typical telephone line, the maximum channel capacity is

$$C = 3000 * \log (1 + 1000) = 30000 \text{ bps (approx.)}$$

Using the previous examples of Nyquist criteria, we saw that for a channel with bandwidth 3 KHz, we could double the data rate from 6000 bps to 12000 bps., by using QPSK instead of binary signalling as the line encoding technique. Using Shannon's criteria for the same channel, we can conclude that irrespective of the line encoding technique used, we cannot increase the channel capacity of this

channel beyond 30000bps.

In practise however, due to receiver constraints and due to external noise sources, Shannon's theoretical limit is never achieved in practise.

Thus to summarize the relationship between bandwidth, data rate and channel capacity,

- In general, greater the signal bandwidth, the higher the information-carrying capacity
- But transmission system & receiver's capability limit the bandwidth that can be transmitted

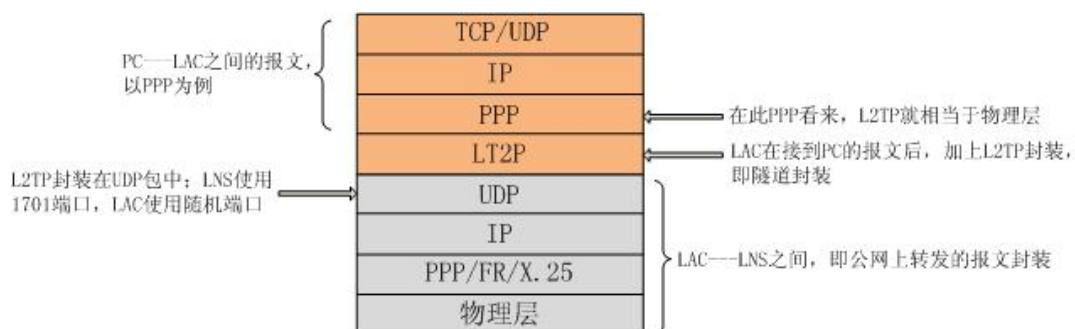
Hence data rate depends on

- Available bandwidth for transmission
- Channel capacity and Signal-to-Noise Ratio
- Receiver Capability

More the frequency allotted, more the channel bandwidth, more the processing capability of the receiver, greater the information transfer rate that can be achieved.

如下图3所示，从图中至上而下的分析，为PC的报文在PPP内网环境中发送到LAC，由LAC封装L2TP，再通过外网的报文正常转发给LNS的报文封装过程。

注：设计的网络环境为PC---LAC为内网使用PPP协议，LAC---LNS为外网使用协议由服务供应商自定。



作者：NicDino

链接：<https://www.zhihu.com/question/23115645/answer/34232671>

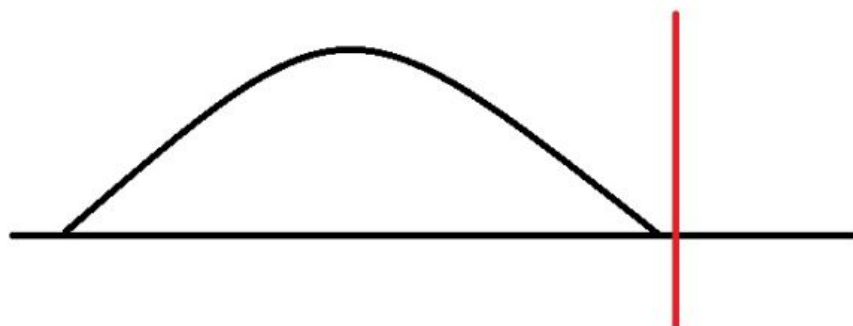
来源：知乎

著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。

1. 为什么要有调制？

因为实际要传输的信号通过的是模拟信道，想要将发送端的信号从一边传到另一边，发送的信号必须具有适合相应信道的物理形式（囧，好拗口）。比如说我想通过一个电话网络发一段数字信号，

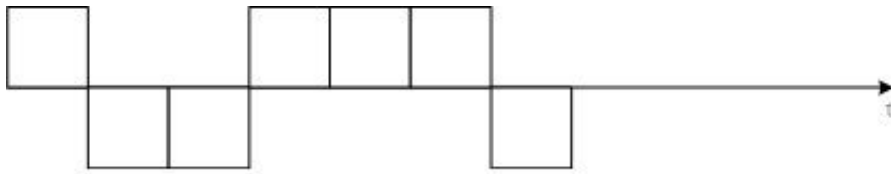
但是由于电缆自身构造的问题，频率太高的信号衰减得很快，那么我就要控制发送端的波形，让它在频谱图上的宽度不至于太宽以至衰减太快。就像下面这样。



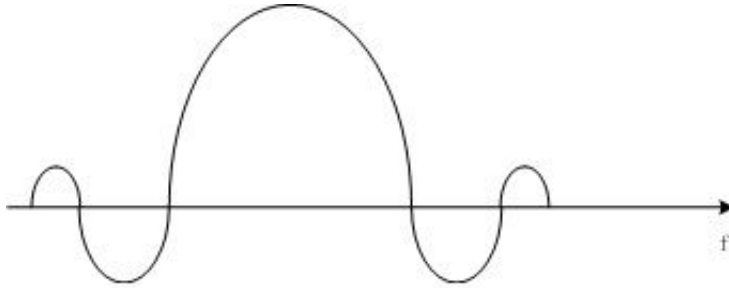
横轴是频率 f ，红线往右表示对高频有很大的衰减。那么作为发送端，就要控制信号的形式，使其尽量不超过红线。

2. 基带信号。

什么是基带信号？假设我的通信系统用的是电话线，那么传输的信号就是基带信号，因为频率很低，基本上集中在零频。下图是一段基带信号。



然后做一个Fourier Transform，频谱大概是下面这个样子。

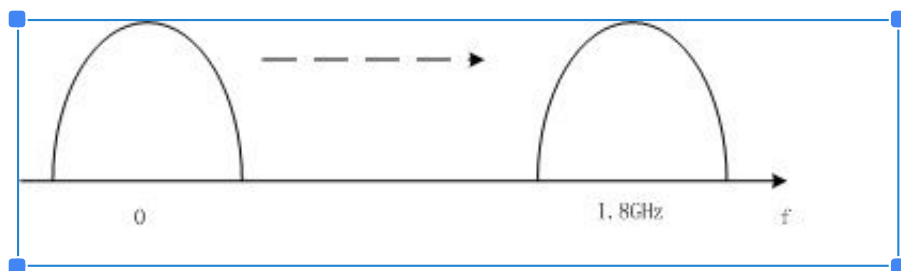


当然，具体在频谱上呈现什么样子取决于基带信号选择什么波形。一般来说，信号在频率轴上面占的宽度和在时间轴上每个脉冲波的时长成反比。所以，不同的调制模式会导致信号占用不同的带宽。BPSK在一个时隙里能传一个信号，QPSK在一个时隙能传两个信号，16QAM在一个时隙能传4个信号（大概是4个，记得不太清楚）。然后导致16QAM占用的带宽高于QPSK，QPSK高于BPSK。

3. 载波。

为什么要有载波？

如果你用的是电话线这样的有线传输信道，基本上就没载波什么事了，这条电缆上的频谱都是你的，想怎么用就怎么用。但是！如果用的是无线信道，那载波就很有必要。因为无线频谱是公共的，谁都可以在上面传输，如果不加任何限制的话，你传这个信号我传那个信号，大家相互干扰然后最后结果就是谁都别想有好的接收信号。所以政府会规定哪一段信号可以用来干什么，比如说GSM只能在900MHz上面的某一段传输，3G下面WCDMA用1.8GHz上面5MHz。但是基带信号基本只集中在0Hz附近，怎么弄到900MHz或者1.8GHz呢？这个时候就要用到载波了。载波自身不具有任何信号，它只是信号的搬运工。就像下图。



一般来说，加载波就是对原信号乘以一个 $\cos(2\pi f t)$ 。注意一下，不同的调制模式决定的是占用的带宽，所谓WCDMA的5Mhz，TD-SCDMA的1.6Mhz，LTE的1.4-20Mhz，802.11的20-160Mhz 指的是上面这幅图里那个信号的宽度有多大。而载波决定的是频率中心点，就是上图中那个波形的中心位于频率轴的哪个地方。

最后，一语以蔽之，无线信号刚从发送端的天线出来的时候，会具有这样的数学表达式：

$$x(t) = \sum_{n=1} a_n p(t - nT) \cos(2\pi f t)$$

a_n 是真正有意义的信号，也就是0101之类的东西；T是每个波形占用的时隙，就是每个波形会持续多长时间，然后 $p(t-nT)$ 取决于调制的模式，不同的调制模式会导致 $p(t)$ 有不同的表达式；最后 $\cos(2\pi f t)$ 就是载波了，f 决定了你要把这个信号搬到频谱轴上的哪个地方。

最最后提一下，其实调制的定义很模糊，你可以说把信号变成合适的波形是调制，加载波是另外的事情；也可以说把信号变成合适的波形和加载波是一个东西。反正最终只要能通过一定的方法把信号从一端发出去，再在另一端收回来，就完成了调制解调啦。