
CS 224 — COMPUTER NETWORKS

Atin Bainada

Contents

1	Line Coding	2
1.1	Multiple types of line coding	2
1.1.1	Non-Return to Zero (NRZ)	2
1.1.2	Return to Zero (RZ)	3
1.1.3	Manchester Encoding	3
1.1.4	Differential Manchester Encoding	4
1.1.5	Other encodings and Baud rate	4
2	Extra Notes	5
2.1	Framing	5
2.1.1	Byte-Oriented Protocols (PPP)	5

1 Line Coding

Line coding is the process of converting data into signal. At the sender's end we convert sequence of bits into digital signal and at the receiver's end the signal is recreated into bits by decoding the signal.

1.1 Multiple types of line coding

1.1.1 Non-Return to Zero (NRZ)

In this line coding 1 is represented by a positive voltage and 0 by a negative voltage.

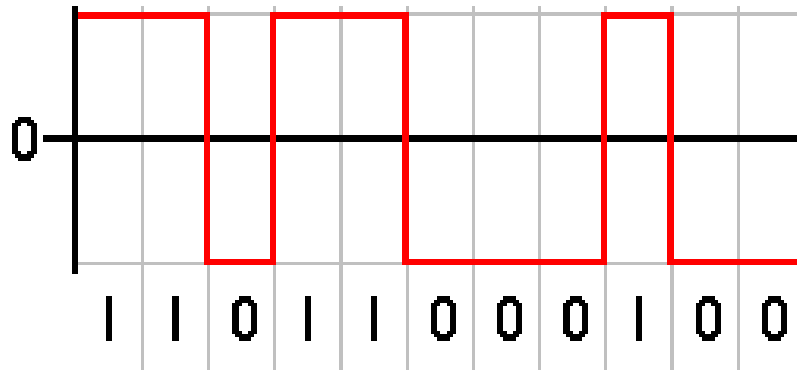


Figure 1: NRZ line coding

The problem with NRZ is that a sequence of several consecutive 1s or 0s means the signal stays constant for a long period of time. There are two problems caused by continuous long strings of 0s or 1s.

The first problem is known as *clock recovery* problem. The clock recovery problem is that both the sender and receiver's clocks have to be precisely synchronized in order for the receiver to recover the same bits the sender transmits. The receiver tries to synchronize its clock using the received signal- *The Clock Recovery Process*. Whenever the signal changes from 1 to 0 or 0 to 1, then the receiver knows it is at a clock cycle boundary and it can resynchronize itself. However, a long period of time without any change in signal leads to *clock drift*. Thus clock recovery depends on having a lot of transitions in the signal.

The second problem is known as *baseline wander*. During the signal transfer the receiver keeps an average of the signal it has received so far and then compare the incoming signal with the average to distinguish between low and high signal. Whenever the signal is significantly lower than the average the receiver concludes that it's a 0; similarly when the signal is much higher than the average the receiver interprets it to be a 1. And when there are too many consecutive 0s or 1s the average voltage wanders from the baseline (hence the name), making it more difficult to detect significant change in the signal.

One approach that addresses this problem is called *non-return to zero inverted* (NRZI). In this encoding the sender makes a transition from current signal to encode a 1 and stays at the current signal to encode a 0. This only solves problem of consecutive 1s.

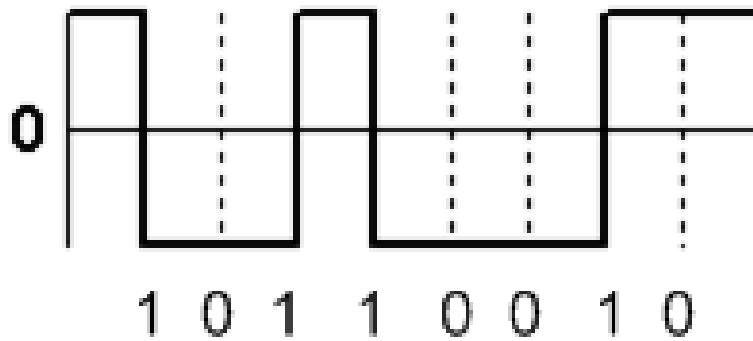


Figure 2: NRZI line coding

1.1.2 Return to Zero (RZ)

In this line code the signal drops to zero voltage between each bit interval. In this case the signal is self-clocking i.e. the receiver can always resynchronize its clock using the clock recovery process hence it eliminates the problem of baseline wander.

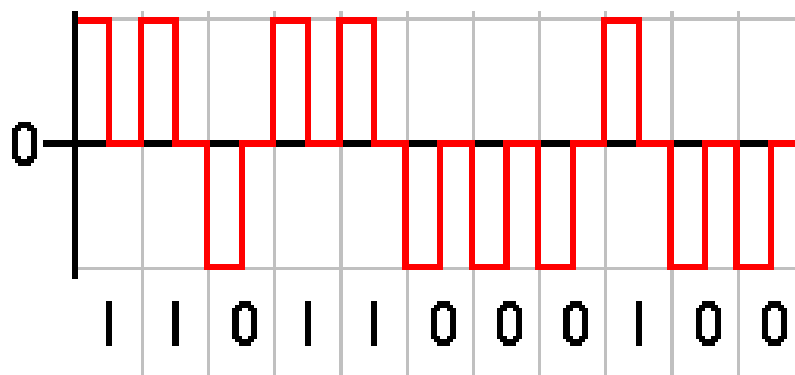


Figure 3: RZ line coding

Main disadvantage of RZ encoding is that it requires twice the bandwidth to get the same data transfer rate as compared to **NRZ encoding**. Another problem is that it uses three levels of voltages. Due to these deficiencies this encoding is not used today. Instead, it has been replaced by Manchester and differential Manchester encodings.

1.1.3 Manchester Encoding

In this encoding each bit is represented as either low in first half and high in second half or high in first half and low in second half. It is a self clocking signal. It is also known as **Phase Encoding** or **PE**

There are two conventions for the representation of data. First one specifies 0 as low-high and 1 as high-low. In the second 0 is represented by high-low and 1 by low-high. The second one is used in **IEEE 802.3**.

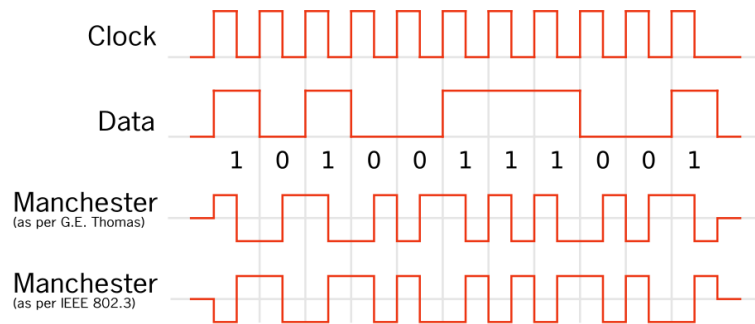


Figure 4: Manchester Encoding (Both conventions)

There are two problems in Manchester encoding.

The first is that the receiver needs to know where the signal starts. For solving this problem, a special, known signal called preamble is sent before the actual signal. The receiver figures out the start of the actual signal using the preamble.

The another problem is with the polarity of the signal. If a Manchester encoded signal gets inverted during communication, it is transformed from one convention to other. This ambiguity can be overcome using differential Manchester encoding.

1.1.4 Differential Manchester Encoding

In this encoding there is always a transition in the middle of the bit interval. 0s and 1s are determined using the beginning of the bit interval. If there is a transition at the beginning of the interval then the bit is 0. If there is no transition at the beginning then the bit is 1.

This encoding solves the problem of polarity. Unlike Manchester encoding, only presence or absence of transition is used to encode/decode the signal. This coding scheme works same if the signal is inverted.

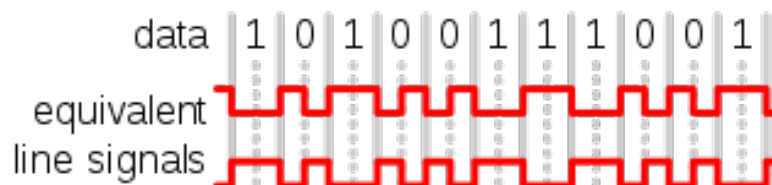


Figure 5: Differential Manchester Encoding (0 represented by transition, 1 by no transition)

It is also known as Biphase Mark Code, digital frequency modulation, F2F. And it is used in the IEEE 802.5 standard for Token Ring LANs.

1.1.5 Other encodings and Baud rate

4B/5B Encoding

As the name implies, in this encoding each 4 bit block of data is mapped to a 5 bit block to avoid the problem of consecutive 1s or 0s.

8B/6T Encoding

In this encoding 6 voltage levels are used to represent 8 bits on a single signal.

Baud Rate

Baud rate is the number of signal/symbol changes per second. A symbol refers to one of the several voltage levels in the signal.

2 Extra Notes

2.1 Framing ¹

In the [line coding](#) lecture we saw how to transmit a sequence of bits over a point-to-point link, or from adaptor to adaptor. From previous chapters we know that blocks of data (called *frames* at this level), not bit streams, are exchanged between nodes. The network adaptor at the nodes handles the exchange of frames. Recognizing exactly what set of bits constitute a frame, or determining start and end of a frame, is the central challenge faced by the adaptor. There are many ways to address this framing problem. Here three different protocols are described to address this problem. Here we are discussing framing in context of point-to-point link but this is a fundamental problem that must also be addressed in multiple access networks like Ethernet and Wi-Fi.

2.1.1 Byte-Oriented Protocols (PPP)

It is one of the oldest approach to framing. In this protocol each frame is viewed as a collection of bytes (characters) instead of bits. At high level, there are two approaches to this type of protocols.

The first is to use special characters known as *sentinel characters* to indicate start and end of a frame. The data portion of the frame is sometimes contained between two or more special characters STX (start of text) and ETX (end of text). BISYNC used this approach. The problem with this is that one of the sentinel characters might appear in the data portion of the frame. The standard way of solving this problem is by "escaping" the character by preceding it with a DLE (data-link-escape) character Whenever it appears in the body of a frame. This approach is often called character stuffing because extra characters are inserted in the data portion of the frame. The **Point-to-Point Protocol** (PPP), which is commonly used to carry Internet Protocol packets over various sorts of point-to-point links, uses this method.

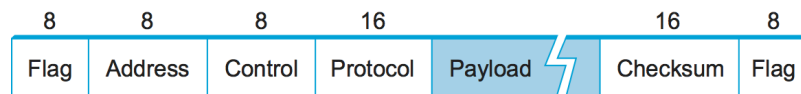


Figure 6: PPP frame format

The alternative of this is to include the number of bytes in the frame at the beginning of the frame, in the frame header. DDCMP used this approach. One danger with this approach is that a transmission error could corrupt the count field, in which case the end of the frame would not be correctly detected. (A similar problem exists with the sentinel-based approach if the ETX field becomes corrupted.)

¹Read more about Framing [here](#).