CS3200: Computer Networks

Lecture 3

IIT Palakkad

30 Jul, 2019

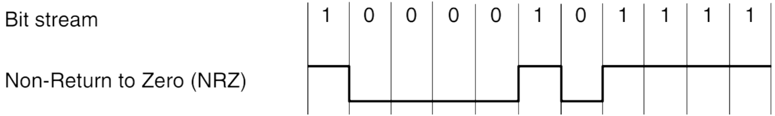
Digital Modulation

- Wires and wireless channels carry analog signals such as continuously varying voltage, light intensity, or sound intensity.
- To send digital information, we must devise analog signals to represent bits. The process of converting between bits and signals that represent them is called **digital modulation**.

Baseband Transmission

Signal occupies frequencies from zero up to a maximum that depends on the signaling rate.

Non-Return to Zero (NRZ)
A positive voltage will be taken to indicate that a 1 was sent and a negative voltage will be taken to indicate that a 0 was sent.



To decode the bits, the receiver maps the signal samples to the closest symbols.
Bandwidth of at least $B/2$ Hz is required when the bit rate is B bits/sec.

Non-Return to Zero (NRZ)

We call the rate at which the signal changes the **symbol rate** to distinguish it from the **bit rate**.

The **bit rate** is the symbol rate multiplied by the number of bits per symbol.

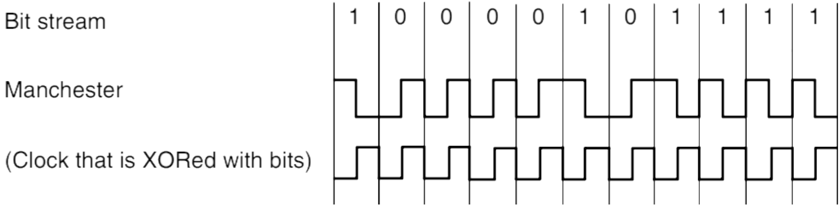
Note that the number of signal levels need to be a power of two. Some levels are often used for protecting against errors and simplifying the design of the receiver.

Clock Recovery

- The receiver must know when one symbol ends and the next symbol begins to correctly decode the bits.
- With NRZ, a long run of 0s or 1s leaves the signal unchanged. After a while it is hard to tell the bits apart, as 15 zeros look much like 16 zeros unless you have a very accurate clock.
- Accurate clocks would help with this problem, but they are an expensive solution for commodity equipment.
- Separate clock signal to receiver?; a waste of bandwidth
- A clever trick here is to mix the clock signal with the data signal by XORing them together so that no extra line is needed.

Manchester Encoding

The clock makes a clock transition in every bit time, so it runs at twice the bit rate.



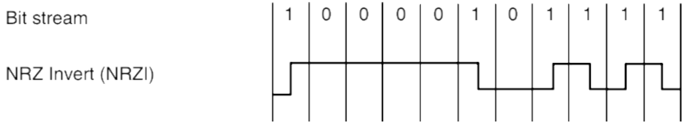
When it is XORed with the 0 level it makes a low-to-high transition that is simply the clock. This transition is a logical 0. When it is XORed with the 1 level it is inverted and makes a high-tolow transition. This transition is a logical 1.

Better decoding, but what at what cost?

Non-Return-to-Zero Inverted

Consider that NRZ will have clock recovery problems only for long runs of 0s and 1s. Can we do something to make NRZ better when there are no long sequences?

A solution: Coding a 1 as a transition and a 0 as no transition, or vice versa.



Receiver stays synchronized with the incoming stream of symbols, without too much increase in bitrate.

Non-Return-to-Zero Inverted

Long runs of 0s still cause a problem. How do we fix it?

A solution: We use **4B/5B** code. Every 4 bits is mapped into a 5-bit pattern with a fixed translation table. The five bit patterns are chosen so that there will never be a run of more than three consecutive 0s.

Data (4B)	Codeword (5B)	Data (4B)	Codeword (5B)
0000	11110	1000	10010
0001	01001	1001	10011
0010	10100	1010	10110
0011	10101	1011	10111
0100	01010	1100	11010
0101	01011	1101	11011
0110	01110	1110	11100
0111	01111	1111	11101