## Digital Data Communication Techniques

### Asynchronous and Synchronous Transmission

Data are transferred over a single signal path rather than a parallel set of lines, as is common with I/O devices and internal computer signal paths. With serial transmission, signaling elements are sent down the line one at a time. Each signaling element may be

• Less than one bit: This is the case, for example, with Manchester coding.

• One bit: NRZ-L and FSK are digital and analog examples, respectively.

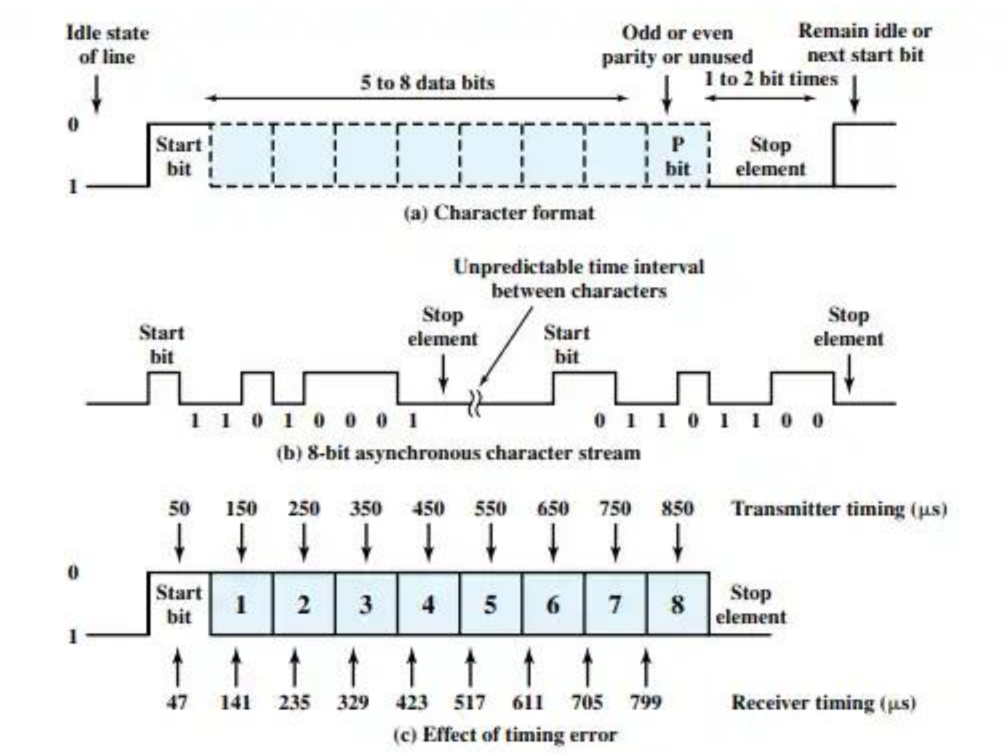
• More than one bit: QPSK is an example.

For simplicity in the following discussion, we assume one bit per signaling element unless otherwise stated. The discussion is not materially affected by this simplification.

One of the difficulties encountered in such a process is that various transmission impairments will corrupt the signal so that occasional errors will occur. This problem is compounded by a timing difficulty: In order for the receiver to sample the incoming bits properly, it must know the arrival time and duration of each bit that it receives. Suppose that the sender simply transmits a stream of data bits. The sender has a clock that governs the timing of the transmitted bits. For example, if data are to be transmitted at one million bits per second (1 Mbps), then one bit will be transmitted every microsecond as measured by the sender's clock. Typically, the receiver will attempt to sample the medium at the center of each bit time. The receiver will time its samples at intervals of one bit time. In our example, the sampling would occur once every If the receiver times its samples based on its own clock, then there will be a problem if the transmitter's and receiver's clocks are not precisely aligned. If there is a drift of 1% (the receiver's clock is 1% faster or slower than the transmitter's clock), then the first sampling will be 0.01 of a bit time away from the center of the bit (center of bit is from beginning and end of bit). After 50 or more samples, the receiver may be in error because it is sampling in the wrong bit time For smaller timing differences, the error would occur later, but eventually the receiver will be out of step with the transmitter if the transmitter sends a sufficiently long stream of bits and if no steps are taken to synchronize the transmitter and receiver.

### Asynchronous Transmission

Two approaches are common for achieving the desired synchronization. The first is called, oddly enough, asynchronous transmission. The strategy with this scheme is to avoid the timing problem by not sending long, uninterrupted streams of bits. Instead, data are transmitted one character at a time, where each character is five to eight bits in length.1Timing or synchronization must only be maintained within each character; the receiver has the opportunity to resynchronize at the beginning of each new character.



(a) Character format

(b) 8-bit asynchronous character stream

(c) Effect of timing error

First, the last sampled bit is incorrectly received. Second, the bit count may now be out of alignment. If bit 7 is a 1 and bit 8 is a 0, bit 8 could be mistaken for a start bit. This condition is termed a framing error, as the character plus start bit and stop element are sometimes referred to as a frame. A framing error can also occur if some noise condition causes the false appearance of a start bit during the idle state. Asynchronous transmission is simple and cheap but requires an overhead of two to three bits per character. For example, for an 8-bit character with no parity bit, using a 1-bit-long stop element, two out of every ten bits convey no information but are there merely for synchronization; thus the overhead is 20%. Of course, the percentage overhead could be reduced by sending larger blocks of bits between the start bit and stop element.

Synchronous Transmission

With synchronous transmission, a block of bits is transmitted in a steady stream without start and stop codes. The block may be many bits in length. To prevent timing drift between transmitter and receiver, their clocks must somehow be synchronized. One possibility is to provide a separate clock line between transmitter and receiver. One side (transmitter or receiver) pulses the line regularly with one short pulse per bit time.The other side uses these regular pulses as a clock.This technique works well over short distances, but over longer distances the clock pulses are subject to the same impairments as the data signal, and timing errors can occur. The other alternative is to embed the clocking information in the data signal. For digital signals, this can be accomplished with Manchester or differential Manchester encoding. For analog signals, a number of techniques can be used; for example, the carrier frequency itself can be used to synchronize the receiver based on the phase of the carrier.

With synchronous transmission, there is another level of synchronization required, to allow the receiver to determine the beginning and end of a block of data. To achieve this, each block begins with a preamble bit pattern and generally ends with a postamble bit pattern. In addition, other bits are added to the block that convey control information used in the data link control.