Adaptive Modulation and HOW IT AFFECTS CELL CAPACITY

CRISTINE KOROWAJCZUK
CCO, CelPlan Technologies, Inc.

I like to tackle problems in parts, break them down, understand what each thing is, and then, how they work together; I think this will also be a good approach to address this topic. Let's start with a quick recap on modulation, then address the adaptive side of it, and, lastly, understand why this knowledge is crucial to properly dimension wireless networks that use adaptive modulation.

Modulation is the process of transmitting information, voice, or data, using a carrier. At this point, I'll assume that the information being transmitted is digital, a stream of bits. I highlight this because it is important to understand what these bits are. When I refer to bits here, I'm talking about the information that we are trying to transmit, i.e. what the device/application is generating at the source that needs to be sent to the destination.

To modulate this carrier, we now need to pack these bits into groups that will be transmitted one at a time; these groups are called symbols. This is where the challenge begins, how many bits can we fit in one symbol? The answer to this question is one that often frustrates: "it depends"; but it does, it really depends. It depends, for example, on the condition of the channel, that is, what is the quality, or signal-to-noise ratio, at the location where the information is being received? What is the technology being used? What are the requirements of the radio being used? And what is the error tolerance of my application?

Rather than trying to discuss the relevance of each of these questions now, I think it is easier to first understand how the mapping of bits into symbols works and then we will have the elements needed to address each of the questions. For simplicity's sake, I will assume in this explanation that the only thing being transmitted is the data bits that we are actually trying to send, ignoring error correction and other overheads that would be applicable in real life.

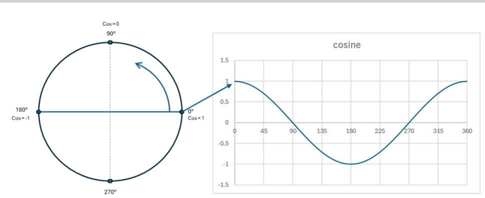

Figure 1 Cosine (sinusoid)

Let's start by working with a very basic modulation scheme: BPSK (Binary Phase Shift Keying). The Phase in the name indicates the position where a sinusoid starts in its oscillation in space, as illustrated in Figure 1.

The fact that a sinusoid can have different phases (Figure 2) allows us to encode it with information by purposefully varying the phase (Shift) to represent different things. This sinusoid can then act as a "carrier" for our information.

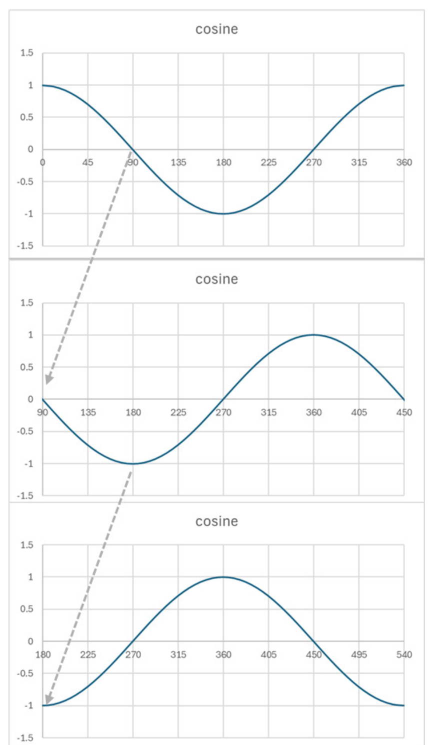

Figure 2 Cosine sinusoids shifted in phase

In BPSK, we use a sinusoid and two distinct phases to identify two binary states that will represent "0" and "1". The phases are usually shifted as far as possible (180°) to make it very easy to differentiate between the bits when receiving the signal. The way the bits are mapped may vary depending on the technology being used (e.g. WiFi, LTE).

Now let's assume I'm trying to transmit a string of 64 bits: 100110010100100010010101001100101100100001110101 0110001010101001

To transmit the first number "1", we'll start the cosine sinusoid with no phase shift, then, for the next bit, "0", we'll invert the sinusoid with a phase shift of 180 degrees and keep doing this for every bit.

The BPSK transmitter typically uses Non-Return-to-Zero (NRZ) coding, where, in the case illustrated in Figure 3, a positive voltage represents a "1" and a negative voltage represents a "0".

Let's assume that to transmit the first bit, in this modulation scheme, the transmitting side sends a voltage of +3V to modulate the carrier to represent the bit "1". At the other end, the receiver will identify the voltage of +3V in the carrier and will decode the bit correctly.

But in our example fading affected the performance of the channel during transmission, and the voltage received was +1.5V. This is where we start to understand the robustness of BPSK modulation; the Decision Boundary is the midway point between the two positions in the map (Figure 4), creating two Decision Regions, one to the left (negative voltages) and one to the right (positive voltages). Thus, even though it did not read the proper voltage, the receiver is still able to "guess" that the bit sent was "1". In fact, any positive voltage will indicate that a "1" was sent; a negative voltage, on the other hand, would indicate that a "0" was sent. Of course, there is still a chance of having such a noisy channel that voltage arrives at the other end beyond the decision boundary and the bit is incorrectly decoded, but this should not happen often; that is, the error rate in this modulation is low, making it very robust.

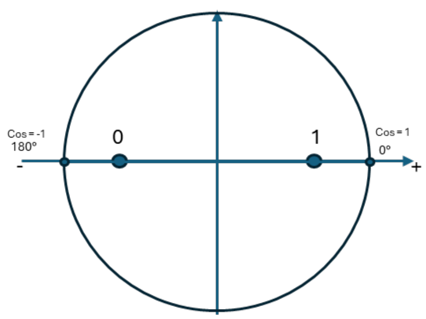

Figure 3 BPSK phase map

...understand why this knowledge is crucial to properly dimension wireless networks that use adaptive modulation.

When using BPSK, one pulse of voltage has to be generated to modulate the carrier for every bit sent, that's because, in this modulation, I'm mapping one bit per symbol. And this introduces us to the beginning of capacity considerations. A bit is a unit of data, but a symbol is a unit of time. The duration of a symbol depends on the technology of transmission and can vary quite a bit: an LTE symbol, for example, is 66.7 µs long, while in Wi-Fi 5 it is 3.2 µs and in WiFi 6 it is 12.8 µs long.

In summary, to send my 64 bits of data in BPSK, we need to transmit 64 symbols (assuming all I'm sending is my data… but that's a topic for later discussion).

To improve efficiency, we can use additional phase shifts to create more states. One way of doing this is using QPSK (Quadrature Phase Shift Keying). In this modulation scheme, we will add a second sinusoid, 90 degrees shifted in relation to the cosine we used for BSPK (Figure 5). To make identification easier, we refer to the cosine sinusoid as in-phase, and the 90-degree shifted one, or the sine sinusoid, as in-quadrature.

When I use QPSK, I introduce the possibility of using more states for mapping bits into symbols; states are the locations in my cartesian chart, identified by the voltage transmitted In-phase (I component) and in-Quadrature (Q component). Each state corresponds to 2 bits so now

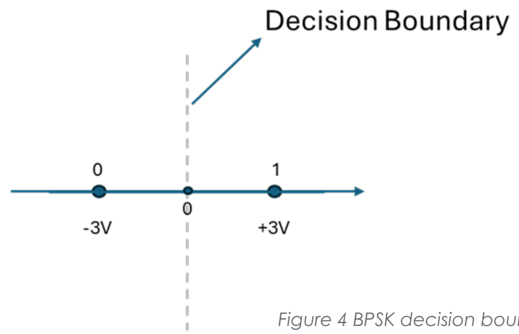

Figure 4 BPSK decision boundary

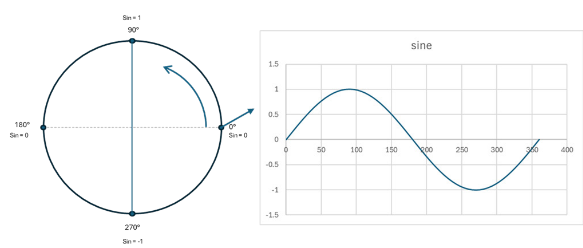

Figure 5 Sine (sinusoid)

we have 4 possible combinations of the bits “0” and “1”: 00, 01, 10, 11. The name “constellation” is used to identify the group of states in a modulation scheme. The position, or quadrant, where each state is located varies with the technology, but in general, a Gray code is followed, that is, only one bit can change between one state and its neighbor; for example, 00 can be adjacent to 01 and 10 but not to 11. This rule has a very useful reason for being: it improves the chance of correctly detecting the state on the receive side and reduces the number of wrong bits received when there’s a misidentification.

Back to our constellation mapping… In QPSK (Figure 6) we can now fit 2 bits per symbol, but addressing each state requires a little more precision; a positive voltage in I is not enough, we also need to identify the voltage in Q correctly to differentiate between “01” and “11”. The advantage is that we can now transmit things twice as fast: with 2 bits per symbol, only 32 symbols are needed to transmit my original 64 bits of data (once again assuming no overhead).

Let’s take a look at 16QAM (Quadrature Amplitude Modulation) next. We saw what Quadrature means when discussing QPSK, but how about the 16 in 16QAM? That refers to the 16 states in the constellation. Again, note the rule of only changing one bit between adjacent states (Figure 7). Also, notice how much closer the states are in the chart. This modulation scheme introduces amplitude in the game, varying the voltage besides the phase to represent different states, which means that the receiver now needs a lot more accurate readings to map the voltages received to the correct state. A “noisy” channel might make life a bit difficult here. What do you get for this accuracy though? Speed. You can now fit 4 bits per symbol, so you only need 16 symbols to transmit your 64 bits of data.

Now we can expand this concept and move up to 64QAM (Figure 8). Based on what we have seen so far, it’s easy to figure out that we will now be dealing with 64 states and that they will be even closer to each other, thus making this a modulation scheme that needs even more precision to be properly decoded.

But there are rewards for the hard life of 64QAM, we get to transmit 6 bits per symbol, that is, in the same amount of time that BPSK transmits 1 bit of data, 64QAM transmits 6! Our string of 64 bits now fits into 11 symbols.

You can see how this continues, the higher the modulation scheme, the more bits you can fit in a symbol but the higher the quality required of the channel to allow the receiver to differentiate between states of the constellation, as illustrated in Figure 9.

Due to the RF quality requirements (i.e. SNR) of the channel for higher modulations, the area where these can be used is limited (Figure 10); for example, the approximate radius of coverage for 4096QAM, a scheme included in the WiFi7 standard, is about 3 feet.

Now that we have a clear understanding of the relationship between increased data rate and stricter signal-to-noise ratio requirement, let’s understand the impact of adaptive modulation on cell capacity.

When designing a network, ideally you want all your devices or end-users to be connected with the highest

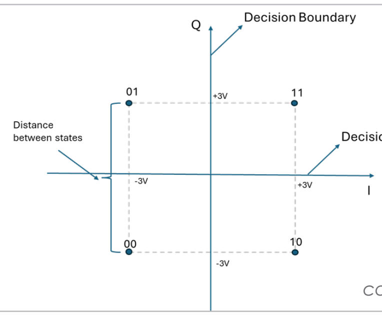

Figure 6 QPSK constellation map

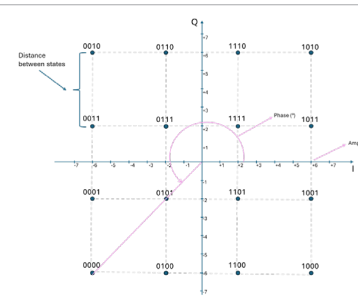

Figure 7 16QAM constellation map

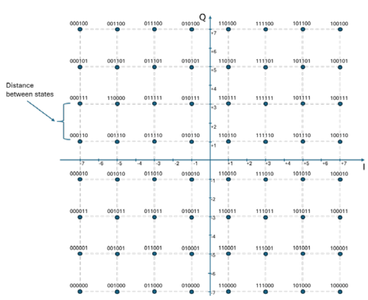

Figure 8 64QAM constellation map

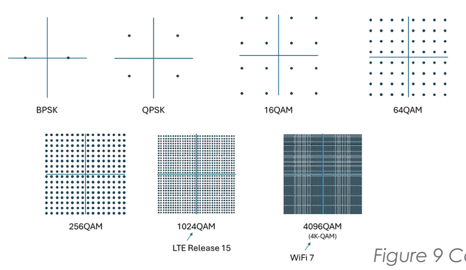

Figure 9 Constellation maps comparison

	Bits / Symbol	
	4096QAM	12
	1024QAM	10
	256QAM	8
	64QAM	6
	16QAM	4
	QPSK	2
	BPSK	1

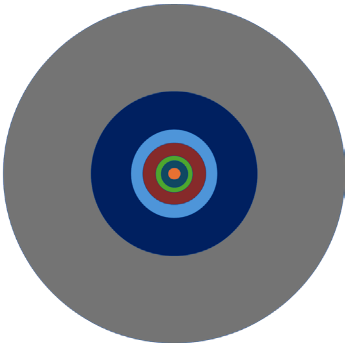

Figure 10 Indication of relative area of cell served per modulation scheme

modulation scheme possible and this is where adaptive modulation thrives, its purpose is to increase cell capacity by optimizing the modulation being used based on channel conditions. This works well when the cell is not completely loaded, but let's take a look at what happens when a fully loaded cell runs into an increased interference scenario.

Start by picturing a single cell with fixed devices as the only users of the cell. You know exactly where they are, and you selected the location of the cell to be ideal to provide service to those users with the best signal level possible trying to maximize signal-to-noise ratio (SNR) and thus achieve higher modulation schemes.

Now, imagine that another cell is introduced in your network, and it causes some interference in your current service area, affecting the SNR at certain locations. Let's say you have 10 devices that were operating on 16QAM that now have a reduced SNR and were automatically downgraded to QPSK operation. Remember that 16QAM has double the spectrum efficiency of QPSK, i.e. it transmits twice the number of bits using the same amount of time. For the cell, this implies that these devices now need double the number of resources to send the same information as before, affecting overall cell capacity. If the cell is not fully loaded and you have some wiggle room, this is not an issue, and it will be accommodated causing the cell to become a little "fuller." Also, due to the interference introduced at the edge, it will now cover a smaller area as devices in the lowest modulation can't downgrade further and will lose service.

If this cell was already full, however, the scenario would be a little different as it would not have the capacity to serve the new requirement of these devices and it would start dropping users (or significantly increasing latency) to cope.

This issue is even more of a problem when there are multiple cells with multiple sectors involved. A technology such as LTE that can use the same carrier for all cells (reuse 1), relies on having some cell capacity always available to coordinate cell edge usage and be able to cope with situations such as this. A fully loaded LTE cell that is affected by a drop in SNR, might trigger issues in all its neighboring cells by increasing the number of conflicts at its cell edge, causing a type of avalanche effect in part of the network: a drop in SNR causes a drop in modulation, which then requires more cell capacity, which then loads the cell more, which then introduces more conflicts at cell edge, which then drops the SNR of more devices...

To summarize, when designing a network, you want to maximize the number of devices in areas served by higher modulation schemes, that is, closer to the center of the cell. If working with LTE, especially when using reuse 1, you want to make sure to save cell capacity to allow the cells to manage carrier usage; the more overlap between cells/sectors and the more devices located in these overlaps, the more capacity you need to save. ■