How 2.4GHz Spread Spectrum works

Taking a few liberties with the details to save time and space, here's a brief overview of the way the 2.4GHz spectrum and timelines are used by existing 2.4GHz RC systems:

Note, if you start getting bored or confused reading what follows, skip ahead to the **ANALOGY** section further down the page, it's much easier going ;-)

With our existing 27/35/36/40/41/50/72Mhz radio systems we require each flier to have exclusive use of a fixed frequency (their "channel"). In order to ensure this, most clubs operate a system whereby fliers must obtain a pin or peg before they can turn on their transmitter. There is only one pin/peg for each frequency so (in theory) there should be no chance of two transmitters operating simultaneously on the same channel/frequency.

However, this sometimes doesn't quite work out and, if someone inadvertantly turns on their transmitter while another is flying a model on the same frequency, a crash will almost certainly result.

Spread spectrum (SS) effectively solves that problem by not only dividing a new frequency band (2.4GHz) up into a range of frequencies but also something called a "timeline".

With SS it becomes possible for many people to fly on the same frequency without causing interference to each other.

How can this happen?

Well because they operate at a much higher frequency (2.4GHz is 33 times higher than 72MHz), the time it takes to send all the various servo positions to a model is dramatically decreased. This means that instead of transmitting continuously, a SS transmitter only spends a very short period of time transmitting and then turns off -- leaving the frequency free for other transmitters to use.

In effect, if a transmitter only takes 2 thousandths of a second to send the servo position information to its receiver, and does this 50 times a second it will have been transmitting for only 10% of the time, the other 90% of the time it just sits idle without transmitting at all. In such a case, it is said to have a timeline utilization of just 10% -- meaning that up to 9 other transmitters could also share the same frequency without interfering with each other (each consuming another 10% of the remaining timeline).

Remember, if this is getting a bit heavy, skip to The ANALOGY further down the page

And, when you realize that there are dozens of frequencies that can be used, this means you could (in theory) fly hundreds of models simultaneously without fear of being shot down.

So how can *your* receiver tell which is *your* transmitter and therefore which signal it should respond to?

That's done by a process called "binding". When you bind a SS receiver to a transmitter, they form an association that ensures that only that transmitter will be responded to. All other transmitters will be completely ignored, even if they're on the same frequency.

However...

If two transmitters did actually transmit on the same frequency at exactly the same time a collision will occur and data will be lost.

Fortunately, most SS systems work very hard to avoid such collisions so they do not intentionally transmit simultaneously.

Instead, they transmit only when they think the frequency is clear. Of course as a particular frequency becomes more congested with multiple transmitters, the chances of collisions occurring increases.

What's more, it's virtually impossible to achieve 100% of the theoretical maximum throughput pm a given frequency (although you can get quite close) because as the timeline becomes more cluttered, the number of collisions (or deferred transmissions following a carrier-sense) increases.

If you're using a 10% utilization of the timeline on a specific frequency for each transmitter then an attempt to run 10 transmitters will result in increased latency (ie: servo response will get sluggish) for all users -- even though (in theory) there is time for them all to transmit if they just wait their turn. Once the timeline becomes saturated then someone is going to miss out and the latency rates really soar.

The Spektrum system does not actively use timeline sharing -- it appears to be solely frequency division (grabing sole ownership of two frequencies and denying any other Spektrum transmitter the right to use those frequencies). In this respect it's little different to a smart synthesized 72MHz system that checks what channels are free before firing up then chooses the quietest one and uses it. Of course the Spektrum also only uses a small percentage of the available time on its chosen channels so other systems could (in theory) slot into the gaps without having too much effect but no two Spektrum 2.4GHz systems will share frequencies with each other in the way that Futaba or XPS do.

The Futaba FAAST uses both time division and frequency division by performing pseudo-random frequency hopping -- meaning that it not only transmits in short bursts like the rest but also keeps constantly changing its frequency so as to dodge interference and reduce the effect of other signals from other transmitters (or interference) that might appear on some of those frequencies.

The XPS system (according to Jim Drew) is primarily a time-division system with frequency shifting only occurring "as required" due to congestion of the timeline or excessive noise on the chosen operating channel.

THE ANALOGY

I like to use a freeway analogy to explain frequency and time multiplexing. Imagine that each lane of the freeway is a frequency and the timeline is the distance between the start and the end of the freeway.

If a timeline is full, that lne of the freeway will be bumper-to-bumper and progress will be slow -- in fact there may not even be any room for your car without creating a collision. If the timeline is empty, there are no cars and you can scoot along as fast as you like.

Okay, here's how our 2.4GHz RC systems fit onto that freeway...

The Spektrum chooses two lanes of the freeway and has those lanes all to itself. Any other Spektrum wanting to use the freeway will have to find two clear lanes of its own. If the Spektrum discovers one of its chosen lanes is blocked with other traffic, it'll continue on in the second lane it's chosen. If that lane gets congested -- uh-oh! Fortunately the Spektrum freeway has about 80 lanes to choose from.

Futaba's FAAST system has no dedicated lane. It weaves and darts between lanes continuously, so as to avoid any other traffic that might be also using the freeway. The FAASt system only starts to slow down when all the freeway lanes are so congested with other traffic that it can no longer find a gap to move into. I'm not sure how many lanes the FAAST freeway has but it's quite a few.

XPS (based on what Jim Drew has explained) starts off on what it believes is the least busy lane and sticks to it -- unless that lane becomes so clogged with traffic that it begins to slow down. It'll then look for a lane with fewer vehicles and change to that so that it can continue at full speed. If XPS can't find a clear lane, it'll use whichever is the least congested. Unfortunately, the XPS freeway only has 11 lanes but under normal circumstances, that should be plenty.

These explanations have been somewhat abridged and simplified from the comprehensive (but I hope "easy to understand") article I've written about the use of 2.4GHz for RC systems. It'll be published online as soon as I've finished some of the loose ends and tidied up a few details.

I hope this at least clarifies things a little for those who wonder what goes on behind the smoke and mirrors.

And as to which system is best, which is most vulnerable to interference and which has the most potential weaknesses... well you'll just have to wait for the full article to be published at RCModelReviews.com