

Are Bees Bugs?

This article is about one of the strangest problems I ever worked on. It involved a part of the 3G cellular network infrastructure known as the Modem Processor (just MP for short). That's a bit of hardware that lives in the cell tower, and manages the moment by moment communication between your smart phone and the tower.

One day in October we got notice that MPs were going down and not coming back up. Without a working MP a cell tower is just a piece of abstract modern art. Now a simple crash and reboot wouldn't have been too bad – that would only take a minute or two. But the only way the cellular company had found to return the cell site to an operational state was to send a technician out to the cell tower and have him physically remove and reinsert the MP – basically doing a forced power cycle of the equipment. The big problem was that it could take several hours for a technician to drive to the affected tower. Having degraded cellular service for hours is a big no-no (Note to hardware designers – always provide a way to do remote power cycles!), so this was a serious issue.

By the luck of the draw I ended up being assigned to work on the problem. The service provider arranged things so that when the problem occurred I could log into the affected cell tower and look around (until the technician got there and started messing with the hardware). The service provider had said that the MP just went completely dead, but I noticed that the Ethernet link to the MP was bouncing. Mostly it was down, but periodically it would come up for a little bit before going down again. So the MP was NOT completely dead! Something was going on!

Well, it turned out the MP was in fact rebooting. It was in a cycle where it would come up, run for maybe 30 seconds, and then die, and then try to come up again. The cycle would repeat ad infinitum. That brief period of uptime was a God send. Using my fastest typing fingers, it was possible for me to log into the MP and issue a few commands before the rug was pulled out from under me.

Now presumably some piece of running software was doing something that was causing the MP to go down. It would have been nice if the MP had been crashing – then we could simply have started a normal software analysis – looking for the bug that was causing the crash. But I was quickly able to determine that the MP was not crashing – the processor was simply being reset, as if there was a gremlin at the cell tower physically pushing the MP's reset button. Now if you're an experienced computer professional you realize this has all the earmarks of a hardware bug – the biggest clue being that recovery required a power cycle. But hardware people always want you to prove that the problem is their fault, so I had more work to do.

Even though it was known that the cause was likely going to ultimately be hardware, the fact that the software would run for a relatively specific amount of time before the processor reset meant that there was still possibly a software component to the issue. Now the MP had well over 100 tasks running on it – with luck one of them was doing something that triggered the problem. Fortunately, I had had the foresight during development (blowing my own horn here) to provide a way of suspending tasks. So I would log into the cell site, wait for the MP to come up, then log into the MP and suspend as many tasks as I could before the MP died. I went through numerous iterations of this, until finally, one time ... the MP did not reset. A minute passed, then two, then three. Success! I had suspended the task that was initiating the reset. A few more iterations narrowed the list of suspects down to a single task. Suspend it, no reset. Un-suspend it, get a reset.

The task in question was responsible for loading the proper cell tower configuration data into some specialized hardware. At this point the debug process ended up chasing (one of many) red herrings. Now this software/hardware combination was being run by many service providers around the world, at tens of thousands of cell towers, yet only one service provider was seeing this issue. So the theory became that maybe it had something to do with the particular configurations that this service provider (who happened to be in Canada by the way) was using, so people started spending a lot of time trying to correlate the problem with configuration specifics.

Now I've described all this activity in a few paragraphs, but the actual process took months. Time passed. The end of January rolled around – and the problem

stopped happening. Nobody had changed anything – the issue just went away on its own. That left a mystery in the air of course, but telecom people can be very pragmatic. The network was running fine – that’s all that mattered.

More time passed. Spring. Summer. October rolled around, and again, MPs started dying. At this point the service provider was sufficiently desperate that I was allowed to load debug images onto cell towers in the problem state. This was always a bit of a rush job, as I only had until the technician got to the cell tower to do my work. I had to load the debug image, do my debugging, and then reload the official software, before the technician arrived to reseat the MP. Since the problem was very sporadic and rarely occurred twice at the same cell tower, there was not much prep that could be done ahead of time in terms of pre-delivering the debug images. Eventually though I was able to isolate the issue down to a single machine instruction. Don’t execute that instruction and everything’s fine. Execute the instruction and the MP resets.

The mystery though, was that the instruction had absolutely no connection with processor reset. Furthermore that exact instruction, in exactly the same context, would be executed in an MP that was booting normally. Somehow, some external factor was causing a normal, useful instruction to act as a processor killer.

At this point the hardware people had to admit that it really was a hardware problem. Being a software person I was no longer heavily involved in the debug process, but I did assist at times. One possibility considered was that the voltage out of range sensor was somehow triggering the reset. That turned out to be a dead end, but it did lead to the discovery of another issue. The interaction between the sensor and the software was through one of the processor’s GPIO (general purpose input/output) pins. When the sensor detected a voltage problem it would signal that fact via a GPIO pin. The software would be watching that GPIO pin, and when it saw the signal it would know that there was a voltage problem. Unfortunately, the hardware was wired to use one GPIO in (say pin X) while the software was watching a different GPIO pin (say pin Y). So the voltage sensor was useless. You wonder how things like that happen, since preventing such issues is not exactly rocket science – you have the hardware people prepare a list of the GPIO pins that they are using, you have the software people prepare a list of the GPIO pins that they are using, and then you compare the two lists!

Anyway, as always, time passed. Eventually the end of January rolled around, and again the problem stopped occurring.

What happened the following October? The problem returned. This time management did something they probably should have done during the first year the problem showed up – they had someone actually examine the cell sites (as opposed to simply having a tech reseal a card and then leave). Do you know what they found? Bees! The equipment enclosures at the cell tower were not sufficiently tight, and when the weather turned cold the bees would find their way into the equipment, presumably looking for warmth. I never got the details, but somehow the bees screwed up the electrical grounding, and that somehow induced the strange behavior that I saw with the “killer instruction”. By January things would be a lot colder, and I suppose by then the bees were either dead or inactive. Ultimately, the fix to this issue was not in computer software or computer hardware, but in site preparation.

So, to answer my titular question, in this particular context, bees are definitely bugs!