

## How to solve TE122 Stability Issues

This guide is mainly related to TE122P stability issues which are being experienced by some users of Asterisk®. Naturally, this also applies to Elastix® and other distributions that use Asterisk®. This guide can also be applied to any other Digium® products that you might be having issues with and is a guide for good practice in any case. This has been confirmed with both Elastix® and some other ISO based distributions.

In the last month, we had two systems that had similar issues, which was dropping of calls at random times during the day. Its one of those problems that is not easy to diagnose, even down to using PRI Debug, it looks like the carrier equipment has failed for a brief period. So naturally you follow up with the carrier, asking them to look at the line to see if they have any issues.

The reason why I mention two machines, was that they were about 14 months apart in motherboard design (although similar), and it would be beneficial to see if the same correction applied to both would correct the issue, giving us some confidence that this was a step in the right direction. It has also been reported by others that this occurs on both AMD® and Intel® Chipset motherboards

To confirm you have a similar issue, the symptoms/issues that you may come across are:

- 1) Calls being dropped (Due to Red Alarm)
- 2) Red Alarms occurring on a regular basis, usually at least 10 per day (as evidenced in the Asterisk Logs)
- 3) Red Alarms occurring for less than a second
- 4) When using a Asterisk® Based distribution such as Elastix®, initial format and install is quite slow (15 min format on 120Gb, and 26 min install of ISO as opposed to 3 min format and 6-8 min install of ISO)
- 5) ZTTool showing an ever increasing increase in the Interrupt misses.
- 6) The faster the machine, the faster the Interrupt misses increase, and also naturally the more often the call drop outs.

Many of us are aware of the interrupt issue where it is always good practice to make sure that the Digium® cards are on their own interrupt (e.g. not sharing with another device), and this includes many of the Digium® Range, as missed interrupts can cause different issues on different cards. As an example the TDM400P with missed interrupts, can result in crackle on the line, or hissing/popping noises. Many of us have become blaise about this interrupt issue as OS software has improved, drivers have improved, APIC has improved and for many it has not presented any issue.

However, with some of the later motherboards, especially with some of the more "generic" boards, they have been making reasonable changes which include less PCI Slots, changing from IDE to SATA, which also includes support of both hard drive types (including mixing), and many other small changes. These changes are making an impact as we found out recently. It is necessary to move back to confirming that the interrupts are not conflicting with the DIGIUM® Cards. At this point it should be noted that these rules should not only apply to Digium® cards but any other cards used for Telephony.

The same emphasis that you place on good network infrastructure, WAN Quality of Service, traffic prioritisation, should equally apply to the telephony card that you install. Those same "realtime traffic" rules that you steadfastly apply to your Local Area Network and Wide Area Network should apply to your Telephony interface.

To resolve this issue, we used a few "tools" that are available on the system. You don't need to install any additional software. The first one is the simple command

```
#cat /proc/interrupts
```

```
          CPU0  CPU1
0:   56986323    0  IO-APIC-edge timer
6:         5    0  IO-APIC-edge floppy
8:         3    0  IO-APIC-edge rtc
9:         0    0  IO-APIC-level acpi
193:  990440    0  IO-APIC-level libata, wcte12x[p]
201: 56975304    0  IO-APIC-level eth0
NMI:         0    0
LOC: 55878552  55878496
ERR:         0
MIS:         0
```

This straight away tells us that the SATA Interface/Driver and the TE122/Driver are sharing the same interrupt on the system.

Another tool is using ZTTEST which runs a test to confirm whether we had some easy to recognise Interrupt issues

```
root@elastix ~]# zttest -v
```

```
Opened pseudo zap interface, measuring accuracy...
```

```
8192 zaptel samples in 8191.664 system clock sample intervals (99.996%)
```

```
8192 zaptel samples in 8191.016 system clock sample intervals (99.988%)
```

```
8192 zaptel samples in 8191.543 system clock sample intervals (99.994%)
```

```
8192 zaptel samples in 8191.520 system clock sample intervals (99.994%)
```

```
8192 zaptel samples in 8191.512 system clock sample intervals (99.994%)
```

```
8192 zaptel samples in 8191.424 system clock sample intervals (99.993%)
```

8192 zaptel samples in 56191.512 system clock sample intervals (0.146%) <==== The blip  
8192 zaptel samples in 8191.480 system clock sample intervals (99.994%)  
8192 zaptel samples in 8191.423 system clock sample intervals (99.993%)  
8192 zaptel samples in 8191.456 system clock sample intervals (99.993%)  
8192 zaptel samples in 8191.472 system clock sample intervals (99.994%)  
8192 zaptel samples in 8191.440 system clock sample intervals (99.993%)  
8192 zaptel samples in 8191.496 system clock sample intervals (99.994%)  
8192 zaptel samples in 8191.464 system clock sample intervals (99.993%)  
8192 zaptel samples in 8191.488 system clock sample intervals (99.994%)  
8192 zaptel samples in 8191.432 system clock sample intervals (99.993%)  
8192 zaptel samples in 8191.528 system clock sample intervals (99.994%)

We run this test over several hours and noted that in general we were on target (on target is generally a figure of 99.97% or better), except for the occasional blip that you will notice in the listing above. Now generally we take the ZTTEST results with a grain of salt, but in this case, this blip was not something we had seen before, and secondly, the amount of times that it appeared over a two hour period, seemed to roughly correlate to the timing that the system dropped the calls. Secondly, it appeared that the IRQ misses that we saw before, seemed to increase each time this blip appeared.

Last but not least we used the

lspci -vb command

What this provided was very detailed information on how the cards/devices were seen at both the bus level and also at the OS level. The output we saw was as follows

```
[root@elastix asterisk]# lspci -vb
```

```
00:1f.2 IDE interface: Intel Corporation 82801GB/GR/GH (ICH7 Family) SATA IDE Controller (rev 01) (prog-if 8f [Master SecP SecO PriP PriO])
```

```
Subsystem: Giga-byte Technology Unknown device b002
```

```
Flags: bus master, 66MHz, medium devsel, latency 0, IRQ 5
```

```
I/O ports at d400
```

```
I/O ports at d800
```

```
I/O ports at dc00
```

I/O ports at e000

I/O ports at e400

Capabilities: [70] Power Management version 2

00:1f.3 SMBus: Intel Corporation 82801G (ICH7 Family) SMBus Controller (rev 01)

Subsystem: Giga-byte Technology Unknown device 5001

Flags: medium devsel, IRQ 5

I/O ports at 0500

02:00.0 Ethernet controller: Digium, Inc. Unknown device 8001 (rev 11)

Subsystem: Digium, Inc. Unknown device 8001

Flags: bus master, medium devsel, latency 64, IRQ 5

I/O ports at a000

Memory at e1000000 (32-bit, non-prefetchable)

Capabilities: [c0] Power Management version 2

02:01.0 Ethernet controller: Realtek Semiconductor Co., Ltd. RTL-8139/8139C/8139

C+ (rev 10)

Subsystem: Realtek Semiconductor Co., Ltd. RT8139

Flags: bus master, medium devsel, latency 64, IRQ 9

I/O ports at a400

Memory at e1001000 (32-bit, non-prefetchable)

Capabilities: [50] Power Management version 2

(\*Note - I have only included a few relevant lines.)

Now the normal routine is to relocate the TE122P into a different slot on the motherboard, which in most cases will change the IRQ that the TE122 will use, usually correcting the issue.

So Our interrupts now look like this

```
cat /proc/interrupts
```

```

          CPU0  CPU1
0:   56986323    0  IO-APIC-edge timer
6:         5    0  IO-APIC-edge floppy
8:         3    0  IO-APIC-edge rtc
9:         0    0  IO-APIC-level acpi
193:   990440    0  IO-APIC-level libata, eth0
201:  56975304    0  IO-APIC-level wcte12x[p]
NMI:         0    0
LOC:  55878552  55878496
ERR:         0
MIS:         0

```

If anyone is watching, you will notice that we switched the position of our Ethernet Card and the TE122P card in the PCI Slots. But we now have the desired results that we were looking for which is the TE122P now sitting on an Interrupt of its own.

Normally this would correct things, but what was interesting was our interrupts continued to climb. It appeared that all that we have been taught about interrupts and TE122P was wrong. Suspicions continued....did we have a faulty card? This was discounted straight away as we had two systems doing the same thing, the only thing that was the same was the carrier, although these were two separate locations, so again this was discounted, and we were left with the machine.

After many, many hours, working through all the BIOS options, disabling devices, trying IDE Drives instead of SATA Drives etc etc...we finally made a change which surprised us. Changing the SATA mode in the BIOS from Auto (which is the default on all these motherboards) to SATA Enhanced resulted in the interrupt misses becoming rock solid after boot up. It was the last thing we wanted to change, as we were using a IDE DVDROM drive (which many of the system builders are still supplying).

If the system is using SATA Hard Drive and a IDE CDROM, the system, in Auto mode selects the PATA/SATA mode to support both devices and it appears that this mode is causing a large amount of the issues. Whether it is a combination of the LIBATA Driver and this PATA/SATA compatibility mode, we cannot be sure which is causing the issue. But taking a huge guess, I suspect that this mode is causing clock cycles to be missed or "stolen" resulting in the interrupt misses.

What further backs up that we had found the device that was causing the issue, was the fact that now, upon installing the ISO Distribution again, the installation was back to lightning fast, with 3 mins for the format, and 5 or so minutes for the install.

Naturally, to confirm that we have found the issue, we put everything back to defaults in the BIOS, we moved the cards

back to the slots that we had previously, and run the same tests, reinstalled the ISO, which went back to the very long install time, and then checked the interrupts and found them climbing again. This time, we made just the change to the SATA settings in BIOS, and whilst this definitely slowed the interrupt misses, but they were still climbing, but only at a rate of 2 or so every minute, which was a lot better, but not the rock steady position that we had with both corrections.

So to summarise, it appears that the two issues together were causing the Interrupt misses that we had, which were

- 1) The LIBATA and the WCTE12x[p] driver cannot be on the same interrupt
- 2) The SATA Driver must be set to ENHANCED MODE (to reduce the possibility that the system selects PATA/SATA mode) which means that you need to be using a SATA (not IDE) CDROM or DVDROM, otherwise it may be disabled

Further evidence has been reviewed on Google, that similar issues are being found on Linux and other Operating Systems with handling of Dual IDE and SATA. What was interesting is that on one motherboard that we applied these changes, we lost access to the IDE Drive, on the other it appeared to still function with and IDE DVD Drive, while in SATA Enhanced mode. It clearly shows that the board manufactures have not got this mode working correctly, and it appears to affect many other boards as well including SuperMicro, ASUS and many others.

Anyhow, we made the changes above to two machines which were definitely suffering from the issue, and we can confirm that they have been in place now for several weeks without an interrupt miss or any issues (which is what we wanted).

Hope this helps others with similar issues.