

osmo-e1d - Bug #4661

Cisco IOS claims "Far End Block Errors Detected"

07/09/2020 08:40 PM - laforge

Status: In Progress	Start date: 07/09/2020
Priority: Normal	Due date:
Assignee: tnt	% Done: 90%
Category:	
Target version:	
Spec Reference:	
Description	
<p>If I connect the ice40 E1 interface cross-over to a cisco E1 Interface card (I tried several different VIWC and VIWC2 models), it always states something about "Far End Block Errors Detected":</p> <pre>c2811-itp>show controllers e1 0/3/0 E1 0/3/0 is up. Applique type is Channelized E1 - balanced Far End Block Errors Detected No alarms detected. alarm-trigger is not set Version info Firmware: 20060711, FPGA: 13, spm_count = 0 Framing is CRC4, Line Code is HDB3, Clock Source is Line. CRC Threshold is 320. Reported from firmware is 320.</pre> <p>As can be seen, CRC4 is enabled, HDB3 is set. I believe this prevents the Cisco IOS devices to ever sending any real traffic on the line. I can configure as many timeslots with MTP2, HDLC or FR and I never see even a single flag octet on the remote (ice40) end.</p> <p>When connecting a DAHDI card to the unmodified Cisco device, I get:</p> <pre>c2811-itp>show controllers e1 0/3/0 E1 0/3/0 is up. Applique type is Channelized E1 - balanced No alarms detected. alarm-trigger is not set Version info Firmware: 20060711, FPGA: 13, spm_count = 0 Framing is CRC4, Line Code is HDB3, Clock Source is Line. CRC Threshold is 320. Reported from firmware is 320.</pre> <p>I did some research on the web, and it seems that this error message is the result of the remote end (ICE40) reporting CRC errors in the data it received from the Cisco side. So my guess is we might be setting some bits wrong in the TS0 bits.</p> <p>As TS0 generation is in the FPGA part, I'm at a loss here trying to understand what exactly is going on :/</p>	

History

#1 - 07/09/2020 08:42 PM - laforge

I downgraded osmo-e1d to 785476901c0d78960e7bcd82fcc2c5535fadd70c - July 2019, i.e. long before any of my changes to make sure it is not any of my recent osmo-e1d contributions. The behavior is identical, i.e. the "Far End Block Errors Detected" message is presented, too.

#2 - 07/09/2020 09:02 PM - laforge

It might be related to the E-bits?

ITU-T G.704 Clause 2.3.3.4 states:

In those frames not containing the frame alignment signal (see 2.3.2), bit 1 is used to transmit the 6-bit CRC-4 multiframe alignment signal and two CRC-4 error indication bits (E).

...

The E-bits should be set to "0" until both basic frame and CRC-4 multiframe alignment are established (see clause 4/G.706). Thereafter, the

E-bits should be used to indicate received errored sub-multiframes by setting the binary state of one E-bit from 1 to 0 for each errored sub-multiframe.

#3 - 07/10/2020 07:04 AM - tnt

So yeah, the gateway is definitely at fault here: * I don't set the bit properly when not aligned * I inverted them ... I naively thought E=0 meant no errors :/

There is a way to work-around in the firmware. You can disable the 'automatic mode' of setting the E bits by replacing E1_TX_CR_MODE_TS0_CRC_E by E1_TX_CR_MODE_TS0_CRC in the init code for the TX core. It will then use bits [14:13] of the buffer descriptors word to fill the E bits instead and you can set those to 1 by using something like e1_regs->tx.bd = e1f_ofs_to_mf(ofs) | E1_BD_CRC1 | E1_BD_CRC0; when filling the TX buffer descriptor.

It'd be good to confirm that this workaround works and then I'll get to actually fixing the gateway to make sure the automatic mode works as the spec requires.

#4 - 07/10/2020 09:51 PM - laforge

- File fw_app_workaround4661.bin added
- Status changed from New to In Progress
- % Done changed from 0 to 20

update: [tnt](#) provided a firmware workaround build, and I tested it in the exact same test setup. The Cisco is now happy. [tnt](#) indicated he will be fixing the gateway at some later point.

Attaching the firmware update here for the record.

#5 - 09/04/2020 08:43 AM - laforge

- Status changed from In Progress to Stalled

laforge wrote:

[tnt](#) indicated he will be fixing the gateway at some later point.

I didn't yet see any related updates in git. Labelling as 'stalled' for now'

#6 - 10/01/2020 08:17 PM - tnt

- Status changed from Stalled to In Progress
- % Done changed from 20 to 90

This should be fixed in <https://github.com/no2fpga/no2e1/commit/26d99a8519086735dd64d7167265ac88cc3aa004> which is included in the latest gateway.

Remains to be tested though

Files

fw_app_workaround4661.bin	11.4 KB	07/10/2020	laforge
---------------------------	---------	------------	---------