

Re: Opinions on 16-bit checksums.

Source: <http://coding.derkeiler.com/Archive/General/comp.arch.embedded/2004-08/1831.html>

From: Grant Edwards (grante_at_visi.com)

Date: 08/26/04

Date: 26 Aug 2004 14:38:48 GMT

On 2004-08-26, Neil Bradley <nb_nospam@synthcom.com> wrote:

>> *There are plenty of changes that "cancel themselves out" for a
>> CRC as well.*
>
> *But not as many as checksum.*

People keep saying things like that, but nobody ever has any real evidence that it's true.

Let's say you've got a 1024 byte block you're summing. There are 2^{8192} possible combinations of bits. For any well-behaved[1] 16-bit checksum (CRC or otherwise), There are $2^{8176} - 1$ bit patterns that are wrong but generate the correct sum.

[1] By "well behaved" I mean that all output values are equally likely for a random input stream.

> *You can reverse two bytes in a file and a traditional checksum
> won't catch it, but a CRC most likely will. If you don't
> believe me, try it yourself.*

Possibly, but what makes you think there aren't just as many other types of errors that a CRC will miss and a different algorithm will catch?

> *Consider yourself lucky, then. You yourself have already
> proven that checksums aren't the best things. Not to say that
> CRCs are the ultimate, but I'd bank \$\$ on it that CRCs are
> better than traditional checksums.*

I'm sure if it was settled by majority vote CRC would win, but in math, things aren't settled by majority vote.

>> *Therefore, there are billions of different
>> changes to a ROM that "cancel out" and generate the same CRC --
>> just as there are billions of changes that can occur and*

comp.arch.embedded: Re: Opinions on 16-bit checksums.

>> *generate the same IP checksum.*

>

> *The question becomes one of the nature of "naturally occurring"*

> *failures and*

That's my point. CRCs are well suited to the errors than naturally occur in serial communications. Nobody seems to care that ROMs probably do not not have the same naturally occurring failure modes.

> *I think you're really talking about "check codes" rather than*

> *traditional checksums.*

OK, whatever. Such "check codes" are traditionally ca