3GPP TSG CN Plenary Meeting #12 **Tdoc NP-010342**
Stockholm, Sweden, 13th - 15th June 2001

Source:	**Lucent Technologies**
Title:	**Report: Lucent Technologies recommendation regarding issue of checksums and CRCs for SCTP**
Agenda item:	**8.**
Document for:	**Approval**

SCTP, or Stream Control Transmission Protocol, was developed by the IETF to transport PSTN signaling messages over IP networks. SCTP, which is used in place of TCP, overcomes certain disadvantages and shortcomings that are inherent in TCP. SCTP was approved by the IETF in October 2000 as RFC 2960.

3GPP TS 29.205 makes a normative reference to the proposed new ITU-T Recommendation Q.2150.3 which specifies SCTP for transport of BICC signaling messages over IP. In turn, Q.2150.3 refers to RFC 2960 as the normative reference.

Similarly 3GPP TS 29.202 makes a normative reference to RFC 2960 for CAP, MAP and BSSAP+ over IP.

Recently, a problem has been found for short signaling messages with the checksum used in SCTP (please see attachment *SCTPissue* for a detailed description). The IETF is addressing the problem, and it looks like a new algorithm will be adopted. If so, the revised version of SCTP will have a new RFC number, and more importantly, the new SCTP version **will not be backward compatible with RFC 2960**. Any signaling messages received for which the check was generated with a different algorithm than the receiver's would be rejected.

Recommendation:

Therefore, Lucent recommends that complete analysis of this issue should be studied in CN4, and CN4 should return with a recommendation to CN#13 in September. It is important that the checksum issue is resolved so as to ensure that 3GPP's Bearer Independent Core Network Architecture uses the corrected version of SCTP, thus ensuring ease in interoperating 3GPP equipment.

On referencing RFCs in ITU-T Recommendations, checksums and CRCs for SCTP

from : R.A.Adams, Lucent Technologies
raadams1@lucent.com, tel: 01666 83 2503

1 Background

Approval of Q.2150.3 has been delayed because of some concerns that the referenced SCTP RFC, RFC 2960, is possibly not valid in one respect. This is that the Adler 32 bit checksum used to validate SCTP messages end to end possibly fails adequately to detect errors in short messages.

IETF is currently debating whether or not to create a new RFC for this (and possibly other) topic(s).

The problem appears to be that ITU-T normative references to RFCs are equivalent to "dated references" to Recommendations. When the IETF approves a new version of an RFC, it provides a new number for it, and inserts an entry in the RFC index against the old RFC pointing to the new RFC. Hence a normative reference to an RFC in an ITU-T Recommendation can be only to that RFC, and not to later enhancements.

2 Suggestion for references to RFCs in Q.2150.3

It seems sensible in just this one document to add a statement to the effect that
"The concept of the distinction between the use of dated and undated references does not apply to IETF RFCs. The IETF, when enhancing an RFC, issues a new number for the enhanced RFC, and inserts an entry in the RFC index against the entry for the old RFC, which references the new RFC. "

3 On SCTP checksums and CRCs

3.1 Current MTP requirement

The error detection capability required of layers underneath the MTP3 is that implied in Q.706 by the requirement upon the MTP as a whole. This is :
" For all User Parts, the following conditions are guaranteed by the MTP:

a) *Undetected errors*

On a signalling link employing a signalling data link which has the error rate characteristic as described in Recommendation Q.702 not more than one in 10^{10} of all message signal units will contain an error that is undetected by the MTP."

For the MTP level 2, the CCITT CRC 16 is used to detect errors. Given the error rates implied by Q.702, and the fact that a link will fail at a detected SU error rate of ~ 0.004, the maximum SU length of 279 octets, and the characteristics of the CCITT 16 bit CRC (see for example Tanenbaum "Computer Networks", published by Prentice Hall, section 3.5.3) which are detection:

- of all single and double errors,
- of all errors with an odd number of bits,
- of all burst errors of length 16 bits or less,
- of 99.997% of 17 bit error bursts,
- of 99.998% of 18 bit and longer error bursts,

the undetected error probability is met.

The CRC 32s that have variously been proposed for SCTP all have performance considerably better than the CRC 16.

3.2 Implications on SCTP, and email discussions

For the M2UA adaptation layer, which effectively replaces the MTP level 2 signalling data link layer, the SCTP undetected error probability should be no worse than one in 10^{10} end to end.

CRC versus Adler checksum email discussions started in the SIGTRAN group of the IETF somewhat after the SCTP RFC was approved. These discussions later moved to the TSV WG of IETF.
The apparent problem of the Adler checksum was raised by Jonathan Stone and Craig Partridge, Jonathan's response to Randall Stewart's request to summarise the position follows:

" From: Jonathan Stone <jonathan@dsg.stanford.edu>
Sender: tsvwg-admin@ietf.org
Errors-To: tsvwg-admin@ietf.org
X-Mailman-Version: 1.0
Precedence: bulk
List-Id: Transport Area Working Group <tsvwg.ietf.org>
X-BeenThere: tsvwg@ietf.org

randall writes:

>This all started when Craig and Jonathan put forward that the
>Adler-32 WAS weaker for small packets. At least that is
>my understanding... Jonathan and Craig will have to explain...
>
>In effect being weaker for small packets causes great problems
>since this is what are first target was.. signalling messages...
>
>Now I really don't care if Adler-32 is not strong enough for
>iSCSI... They can do something at the app layer... BUT if
>Adler-32 IS weaker than UDP/TCP in small packets then I think
>we do need to do something about it... I put fletcher-16
>(which is a 32 bit sum) in the current BIS since it is already
>an option for TCP...
>
>Jonathan/Craig... could you elaborate the problem for
>small packets....

Briefly, the problem is that, for very short packets, Adler32 is guaranteed to give poor coverage of the available bits. Don't take my word for it, ask Mark Adler. :-).

Adler-32 uses two 16-bit counters, s1 and s2. s1 is the sum of the input, taken as 8-bit bytes. s2 is a running sum of each value of s1. Both s1 and s2 are computed mod-65521 (the largest prime less than 2^16). Consider a packet of 128 bytes. The *most* that each byte can be is 255. There are only 128 bytes of input, so the greatest value which the s1 accumulator can have is 255 * 128 = 32640. So for 128-byte packets, s1 _never_ wraps. That is critical. Why?

The key is to consider the distribution of the s1 values, over some distribution of the values of the individual input bytes in each packet. Because s1 never wraps, s1 is simply the sum of the individual input bytes. (even Doug's trick of adding 0x5555 doesn't help here, and an even larger value doesn't really help: we can get at most one mod-565521 reduction).

Given the further assumption that the input bytes are drawn independently from some distribution (they probably aren't: for filesystem data, it's even worse than that!), the Central Limit Theorem tells us that that s1 will tend to have a normal distribution.
That's bad: it tells us that the value of s1 will have hot-spots at around 128 times the mean of the input distribution: around 16k, assuming a uniform distribution. That's bad. We want the accumulator to wrap as many times as possible, so that the resulting sum has as close to a uniform distribution as possible. (I call this "fairness").

So, for short packets, the Adler-32 s1 sum is guaranteed to be unfair.
why is that bad? it's bad because the space of valid packets-- input data, plus checksum values -- is also small. If all packets have checksum values very close to 32640, then the likelihood of even a `small' error leaving a damaged packet with a valid checksum is higher than if all checksum values are equally likely.

It's not possible to say exactly how bad this effect is, without access to traces of signalling-protocol packets to feed into a simulation. The only high-quality data I have is on the contents of files drawn from filesystems.

Last: If the bytes are not independent, but positively autocorrelated within a packet, that just makes things even worse: intuitively, that makes the s1 value approach a normal distribution even more tightly than the Central Limit theorem suggests.

The story for s2 is more complicated; I'd prefer to leave that for now.

Craig and I both did some fairly crude simulations back in January. I still have some results about how the second-order sum (s2) is distributed, given uniform input. I'm not sure whether i still have on the distribution of s1; anything I do still have would be on archival storage. And it would be low-quality: small samples.

I never went back and did high-quality simulations of 128-byte filesystem data, as I didn't (and still don't) know enough about telephony signalling in general (or SS7 in particular) to know whether filesystem data are a suitable proxy for signalling packets. If it is, I can certainly look at the distribution of Adler32 sums over very short `packets' drawn from filesystem data. I will probably be doing that sometime in June anyway.

hope that helps. For more detail than that, I'll have to start circulating around postscript copies of the draft chapter from my dissertation.

--Jonathan"

Jonathan Stone and Craig Partridge's work followed their part authorship of an article in IEEE/ACM Transactions on Networking, Vol.6 No.5, October 1998, "Performance of Checksums and CRC's over Real Data", in which they claim for real data the failure rate of the CRC 32 investigated was
"almost perfectly consistent with the expected failure rate for random data", whereas the TCP checksum they looked at was considerably worse than that expected, due to the non-uniformity of the data and the checksum's weakness for certain patterns of data.

To sum up on the Adler discussion, for short packets that are required to carry SS7 signalling information, the checksum has "hot spots", which means that the probability of not detecting

an error is considerably worse than 1 in 2^{32} . At present, it is not clear if this will then allow the MTP requirement to be met.

4 End to end checks vs link by link checking

According to various emails, Jonathan Stone and Craig Partridge state that just using link level checks is not sufficient - they examined 500,000 packets which failed the TCP or UDP or IP checksum in the Internet, and they state that "...the highly non-random distribution of errors strongly suggests some applications should employ application-level checksums or equivalents". With M2UA, the whole SCTP path is equivalent to the MTP2 link, so we need the end-to-end SCTP check.

Vern Paxson (April 19) supports this : "...the problem isn't links, it's boxes - routers and end systems. So you want to ensure you have good protection even if the links themselves have strong protection...."

5 Conclusion

It is still uncertain whether or not the undetected error performance requirements on the MTP will be met by signalling paths using SCTP implemented according to RFC 2960.
If they are not, and if the SCTP needs to be enhanced, opinions differ as to:

1. replacing the SCTP RFC 2960's checksum with another, possibly CRC 32 or variant (leading to backwards incompatibility), or
2. leaving in the Adler 32 bit checksum , and adding an optional extra scheme.

2 is not favoured by some, particularly those hoping to be able to use hardware for (at least) generating the check bits. Here, CRCs are particularly suited to hardware generation, especially if they are the last bits transmitted.

Since the suggestion in section 2 of this paper is just a statement of fact, without any pre-judgement of the IETF resolution, and would enable a published Q.2150.3 to remain valid whether or not the referenced RFC 2960 is indeed valid at the time of Q.2150.3's publication, we propose adoption of this text or equivalent in Q.2150.3. This by no means invalidates any other recommendation referring to RFCs.