Internet Engineering Task Force (IETF)
Request for Comments: 7820
Category: Experimental
ISSN: 2070-1721

T. Mizrahi
Marvell

March 2016

UDP Checksum Complement in the One-Way Active Measurement Protocol (OWAMP) and Two-Way Active Measurement Protocol (TWAMP)

Abstract

The One-Way Active Measurement Protocol (OWAMP) and the Two-Way Active Measurement Protocol (TWAMP) are used for performance monitoring in IP networks. Delay measurement is performed in these protocols by using timestamped test packets. Some implementations use hardware-based timestamping engines that integrate the accurate transmission time into every outgoing OWAMP/TWAMP test packet during transmission. Since these packets are transported over UDP, the UDP Checksum field is then updated to reflect this modification. This document proposes to use the last 2 octets of every test packet as a Checksum Complement, allowing timestamping engines to reflect the checksum modification in the last 2 octets rather than in the UDP Checksum field. The behavior defined in this document is completely interoperable with existing OWAMP/TWAMP implementations.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for examination, experimental implementation, and evaluation.

This document defines an Experimental Protocol for the Internet community. This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are a candidate for any level of Internet Standard; see Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc7820.

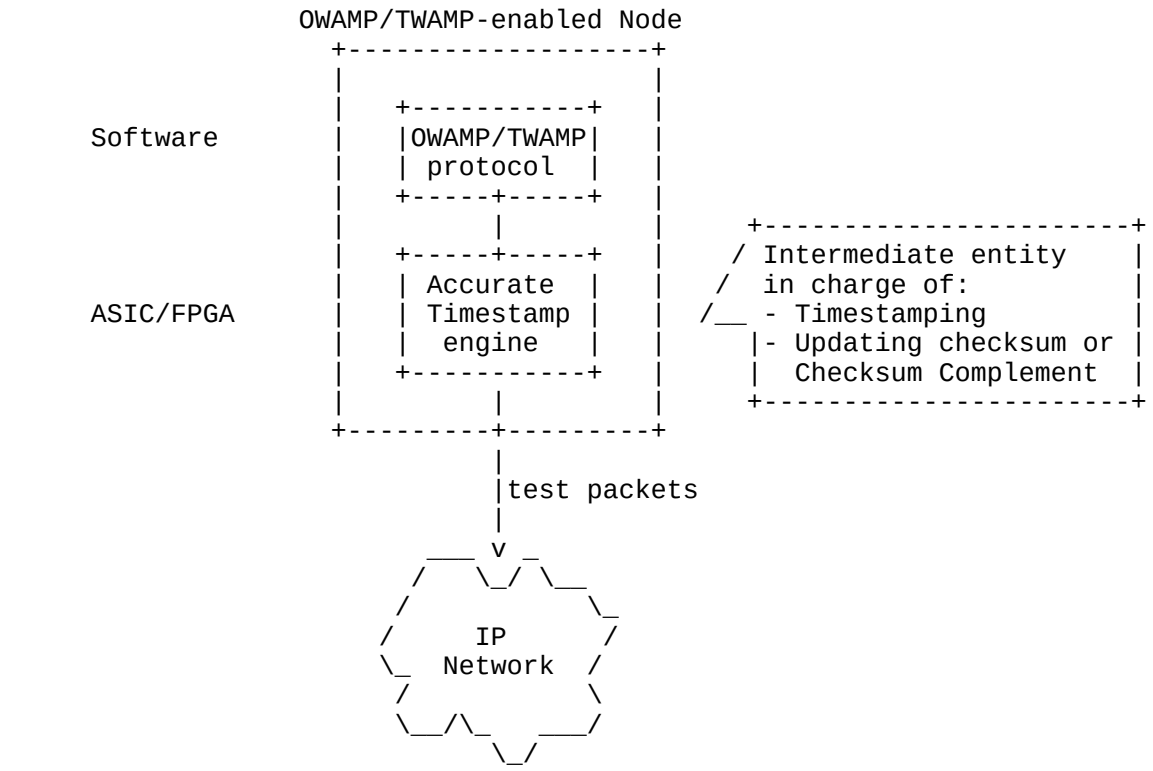
Table of Contents

1. Introduction

The One-Way Active Measurement Protocol [OWAMP] and the Two-Way Active Measurement Protocol [TWAMP] are used for performance monitoring in IP networks.

Delay and delay variation are two of the metrics that OWAMP/TWAMP can measure. Measurement is performed using timestamped test packets. In some use cases, such as carrier networks, these two metrics are an essential aspect of the Service Level Agreement (SLA) and therefore must be measured with a high degree of accuracy. If packets are timestamped in hardware as they exit the host, then greater accuracy is possible in comparison to higher-layer timestamps (as explained further below).

The accuracy of delay measurements relies on the timestamping method and its implementation. In order to facilitate accurate timestamping, an implementation can use a hardware-based timestamping engine, as shown in Figure 1. In such cases, the OWAMP/TWAMP packets are sent and received by a software layer, whereas the timestamping engine modifies every outgoing test packet by incorporating its accurate transmission time into the Timestamp field in the packet.



ASIC: Application-Specific Integrated Circuit
FPGA: Field-Programmable Gate Array

Figure 1: Accurate Timestamping in OWAMP/TWAMP

OWAMP/TWAMP test packets are transported over UDP. When the UDP payload is changed by an intermediate entity such as the timestamping engine, the UDP Checksum field must be updated to reflect the new payload. When using UDP over IPv4 [UDP], an intermediate entity that cannot update the value of the UDP Checksum has no choice except to assign a value of zero to the Checksum field, causing the receiver to ignore the Checksum field and potentially accept corrupted packets. UDP over IPv6, as defined in [IPv6], does not allow a zero checksum, except in specific cases [ZeroChecksum]. As discussed in [ZeroChecksum], the use of a zero checksum is generally not recommended and should be avoided to the extent possible.

Since an intermediate entity only modifies a specific field in the packet, i.e., the Timestamp field, the UDP Checksum update can be performed incrementally, using the concepts presented in [Checksum].

A similar problem is addressed in Annex E of [IEEE1588]. When the Precision Time Protocol (PTP) is transported over IPv6, 2 octets are appended to the end of the PTP payload for UDP Checksum updates. The value of these 2 octets can be updated by an intermediate entity, causing the value of the UDP Checksum field to remain correct.

This document defines a similar concept for [OWAMP] and [TWAMP], allowing intermediate entities to update OWAMP/TWAMP test packets and maintain the correctness of the UDP Checksum by modifying the last 2 octets of the packet.

The term "Checksum Complement" is used throughout this document and refers to the 2 octets at the end of the UDP payload, used for updating the UDP Checksum by intermediate entities.

The usage of the Checksum Complement can in some cases simplify the implementation, because if the packet data is processed in serial order, it is simpler to first update the Timestamp field and then update the Checksum Complement, rather than to update the timestamp and then update the UDP Checksum residing at the UDP header.

The Checksum Complement mechanism is also defined for the Network Time Protocol in [RFC7821].

2. Conventions Used in This Document

2.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [KEYWORDS].

2.2. Abbreviations

HMAC	Hashed Message Authentication Code
OWAMP	One-Way Active Measurement Protocol
PTP	Precision Time Protocol
TWAMP	Two-Way Active Measurement Protocol
UDP	User Datagram Protocol

3. Using the UDP Checksum Complement in OWAMP and TWAMP

3.1. Overview

The UDP Checksum Complement is a 2-octet field that is piggybacked at the end of the test packet. It resides in the last 2 octets of the UDP payload.

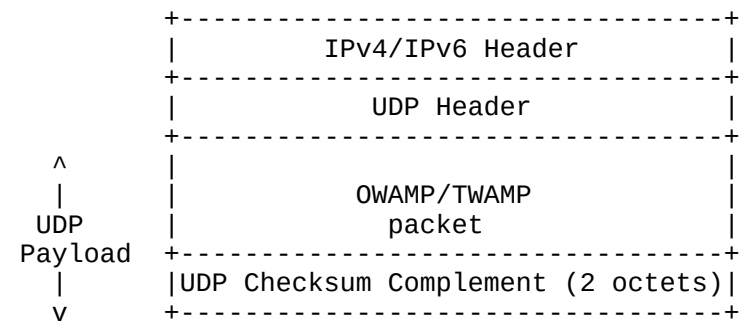


Figure 2: Checksum Complement in OWAMP/TWAMP Test Packets

The Checksum Complement is used to compensate for changes performed in the packet by intermediate entities, as described in the Introduction (Section 1). An example of the usage of the Checksum Complement is provided in Appendix A.

3.2. OWAMP/TWAMP Test Packets with Checksum Complement

The One-Way Active Measurement Protocol [OWAMP] and the Two-Way Active Measurement Protocol [TWAMP] both make use of timestamped test packets. A Checksum Complement MAY be used in the following cases:

- In OWAMP test packets sent by the sender to the receiver.
- In TWAMP test packets sent by the sender to the reflector.
- In TWAMP test packets sent by the reflector to the sender.

OWAMP/TWAMP test packets are transported over UDP, either over IPv4 or over IPv6. This document applies to both OWAMP and TWAMP over IPv4 and over IPv6.

OWAMP/TWAMP test packets contain a Packet Padding field. This document proposes to use the last 2 octets of the Packet Padding field as the Checksum Complement. In this case, the Checksum Complement is always the last 2 octets of the UDP payload, and thus the field is located at (UDP Length - 2 octets) after the beginning of the UDP header.

Figure 3 illustrates the OWAMP test packet format, including the UDP Checksum Complement.

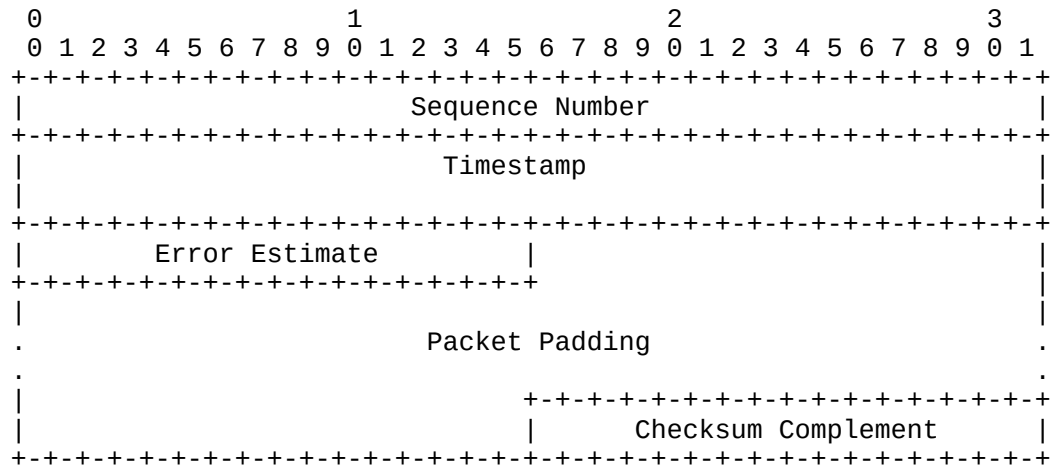


Figure 3: Checksum Complement in OWAMP Test Packets

Figure 4 illustrates the TWAMP test packet format, including the UDP Checksum Complement. ("TTL" means "Time to Live", and "MBZ" refers to the "MUST be zero" field [IPPMIPsec].)

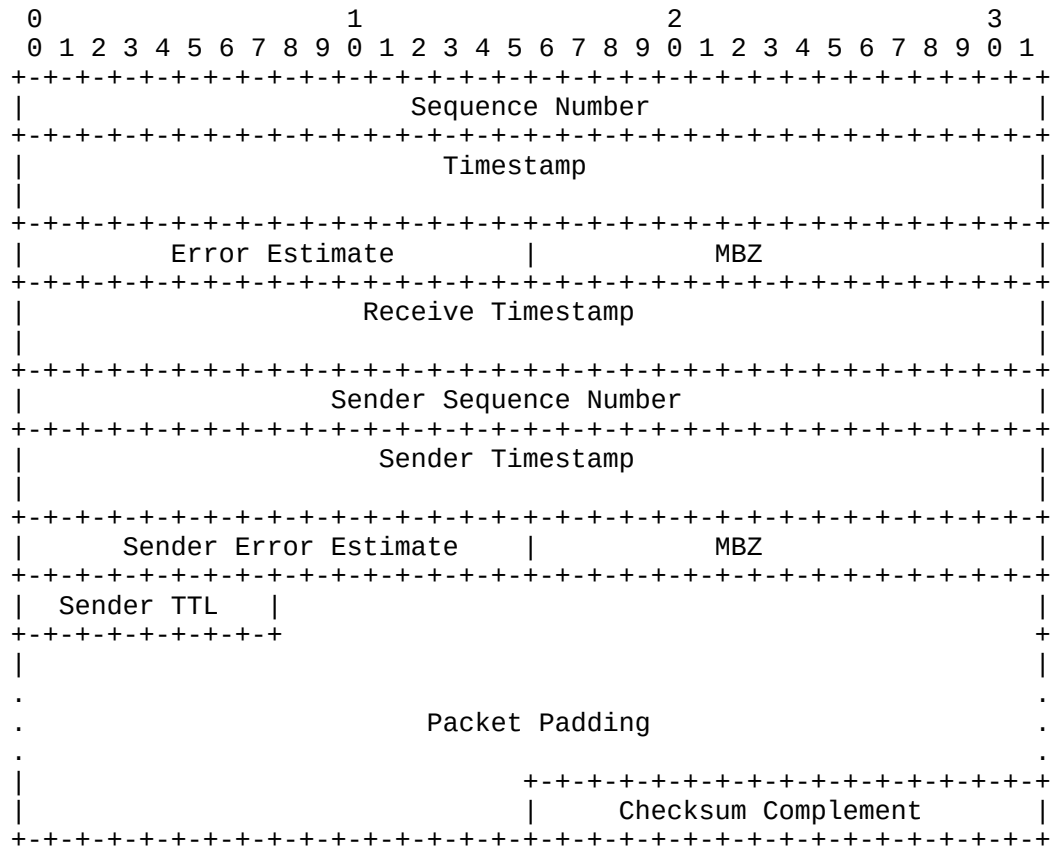


Figure 4: Checksum Complement in TWAMP Test Packets

The length of the Packet Padding field in test packets is announced during the session initiation through the <Padding Length> field in the Request-Session message [OWAMP] or in the Request-TW-Session message [TWAMP].

When a Checksum Complement is included, the padding length MUST be sufficiently long to include the Checksum Complement:

- In OWAMP, the padding length is at least 2 octets, allowing the sender to incorporate the Checksum Complement in the last 2 octets of the padding.

- In TWAMP, the padding length is at least 29 octets in unauthenticated mode and at least 58 octets in authenticated mode. The additional padding is required, since the header of reflector test packets is longer than the header of sender test packets. The difference between the sender packet and the reflector packet is 27 octets in unauthenticated mode and 56 octets in authenticated mode. Thus, the padding in reflector test packets is shorter than the padding in sender packets. Using at least 29 octets of padding (58 in authenticated mode) in sender test packets allows both the sender and the reflector to use a 2-octet Checksum Complement. Note: If the minimal length requirement is not met, the reflector cannot use a Checksum Complement in the reflected test packets, but the sender can use a Checksum Complement in the test packets it transmits.

- Two optional TWAMP features are defined in [TWAMP-Reflect]: octet reflection and symmetrical size. When at least one of these features is enabled, the Request-TW-Session message includes the <Padding Length> field, as well as a <Length of padding to reflect> field. In this case, both fields must be sufficiently long to allow at least 2 octets of padding in both sender test packets and reflector test packets. Specifically, when octet reflection is enabled, the two Length fields must be defined such that the padding expands at least 2 octets beyond the end of the reflected octets.

As described in Section 1, the extensions described in this document are implemented by two logical layers -- a protocol layer and a timestamping layer. It is assumed that the two layers are synchronized regarding whether the usage of the Checksum Complement is enabled or not; since both logical layers reside in the same network device, it is assumed that there is no need for a protocol that synchronizes this information between the two layers. When Checksum Complement usage is enabled, the protocol layer must take care to verify that test packets include the necessary padding, thereby avoiding the need for the timestamping layer to verify that en-route test packets include the necessary padding.

3.2.1. Transmission of OWAMP/TWAMP with Checksum Complement

The transmitter of an OWAMP/TWAMP test packet MAY include a Checksum Complement field, incorporated in the last 2 octets of the padding.

A transmitter that includes a Checksum Complement in its outgoing test packets MUST include a Packet Padding field in these packets, the length of which MUST be sufficient to include the Checksum Complement. The length of the Packet Padding field is negotiated during session initiation, as described in Section 3.2.

3.2.2. Intermediate Updates of OWAMP/TWAMP with Checksum Complement

An intermediate entity that receives and alters an OWAMP/TWAMP test packet can alter either the UDP Checksum field or the Checksum Complement field in order to maintain the correctness of the UDP Checksum value.

3.2.3. Reception of OWAMP/TWAMP with Checksum Complement

This document does not impose new requirements on the receiving end of an OWAMP/TWAMP test packet.

The UDP layer at the receiving end verifies the UDP Checksum of received test packets, and the OWAMP/TWAMP layer should treat the Checksum Complement as part of the padding.

3.3. Interoperability with Existing Implementations

The behavior defined in this document does not impose new requirements on the reception behavior of OWAMP/TWAMP test packets. The protocol stack of the receiving host performs the conventional UDP Checksum verification; thus, from the perspective of the receiving host, the existence of the Checksum Complement is transparent. Therefore, the functionality described in this document allows interoperability with existing implementations that comply with [OWAMP] or [TWAMP].

3.4. Using the Checksum Complement with or without Authentication

Both OWAMP and TWAMP may use authentication or encryption, as defined in [OWAMP] and [TWAMP].

3.4.1. Checksum Complement in Authenticated Mode

OWAMP and TWAMP test packets can be authenticated using an HMAC (Hashed Message Authentication Code). The HMAC covers some of the fields in the test packet header. The HMAC does not cover the Timestamp field and the Packet Padding field.

A Checksum Complement MAY be used when authentication is enabled. In this case, an intermediate entity can timestamp test packets and update their Checksum Complement field without modifying the HMAC.

3.4.2. Checksum Complement in Encrypted Mode

When OWAMP and TWAMP are used in encrypted mode, the Timestamp field is encrypted.

A Checksum Complement SHOULD NOT be used in encrypted mode. The Checksum Complement is effective in both unauthenticated mode and authenticated mode, allowing the intermediate entity to perform serial processing of the packet without storing and forwarding it.

On the other hand, in encrypted mode, an intermediate entity that timestamps a test packet must also re-encrypt the packet accordingly. Re-encryption typically requires the intermediate entity to store the packet, re-encrypt it, and then forward it. Thus, from an implementer’s perspective, the Checksum Complement has very little value in encrypted mode, as it does not necessarily simplify the implementation.

Note: While [OWAMP] and [TWAMP] include an inherent security mechanism, these protocols can be secured by other measures, e.g., [IPPMIPsec]. For reasons similar to those described above, a Checksum Complement SHOULD NOT be used in this case.

4. Security Considerations

This document describes how a Checksum Complement extension can be used for maintaining the correctness of the UDP Checksum.

The purpose of this extension is to ease the implementation of accurate timestamping engines, as illustrated in Figure 1. The extension is intended to be used internally in an OWAMP/TWAMP-enabled node, and not intended to be used by intermediate switches and routers that reside between the sender and the receiver/reflector. Any modification of a test packet by intermediate switches or routers should be considered a malicious man-in-the-middle (MITM) attack.

It is important to emphasize that the scheme described in this document does not increase the protocol's vulnerability to MITM attacks; a MITM attacker who maliciously modifies a packet and its Checksum Complement is logically equivalent to a MITM attacker who modifies a packet and its UDP Checksum field.

The concept described in this document is intended to be used only in unauthenticated mode or authenticated mode. As described in Section 3.4.2, using the Checksum Complement in encrypted mode does not simplify the implementation compared to using the conventional checksum, and therefore the Checksum Complement should not be used.

5. References

5.1. Normative References

[Checksum] Rijsinghani, A., Ed., "Computation of the Internet Checksum via Incremental Update", RFC 1624, DOI 10.17487/RFC1624, May 1994, <http://www.rfc-editor.org/info/rfc1624>.

[IPv6] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <http://www.rfc-editor.org/info/rfc2460>.

[KEYWORDS] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[OWAMP] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, DOI 10.17487/RFC4656, September 2006, <http://www.rfc-editor.org/info/rfc4656>.

[TWAMP] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, DOI 10.17487/RFC5357, October 2008, <http://www.rfc-editor.org/info/rfc5357>.

[TWAMP-Reflect] Morton, A. and L. Ciavattone, "Two-Way Active Measurement Protocol (TWAMP) Reflect Octets and Symmetrical Size Features", RFC 6038, DOI 10.17487/RFC6038, October 2010, <http://www.rfc-editor.org/info/rfc6038>.

[UDP] Postel, J., "User Datagram Protocol", STD 6, RFC 768, DOI 10.17487/RFC768, August 1980, <http://www.rfc-editor.org/info/rfc768>.

5.2. Informative References

[IEEE1588] IEEE, "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", IEEE Std 1588-2008, DOI 10.1109/IEEESTD.2008.4579760, July 2008.

[IPPMIPsec] Pentikousis, K., Ed., Zhang, E., and Y. Cui, "IKEv2-Derived Shared Secret Key for the One-Way Active Measurement Protocol (OWAMP) and Two-Way Active Measurement Protocol (TWAMP)", RFC 7717, DOI 10.17487/RFC7717, December 2015, <http://www.rfc-editor.org/info/rfc7717>.

[RFC7821] Mizrahi, T., "UDP Checksum Complement in the Network Time Protocol (NTP)", RFC 7821, DOI 10.17487/RFC7821, March 2016, <http://www.rfc-editor.org/info/rfc7821>.

[ZeroChecksum] Fairhurst, G. and M. Westerlund, "Applicability Statement for the Use of IPv6 UDP Datagrams with Zero Checksums", RFC 6936, DOI 10.17487/RFC6936, April 2013, <http://www.rfc-editor.org/info/rfc6936>.

Appendix A. Checksum Complement Usage Example

Consider a session between an OWAMP sender and an OWAMP receiver, in which the sender transmits test packets to the receiver.

The sender's software layer generates an OWAMP test packet with a timestamp T and a UDP Checksum value U. The value of U is the checksum of the UDP header, UDP payload, and pseudo-header. Thus, U is equal to:

$$U = Const + checksum(T) \quad (1)$$

Where "Const" is the checksum of all the fields that are covered by the checksum, except the timestamp T.

Recall that the sender's software emits the test packet with a Checksum Complement field, which is simply the last 2 octets of the padding. In this example, it is assumed that the sender initially assigns zero to these 2 octets.

The sender's timestamping engine updates the Timestamp field to the accurate time, changing its value from T to T'. The sender also updates the Checksum Complement field from zero to a new value C, such that:

$$checksum(C) = checksum(T) - checksum(T') \quad (2)$$

When the test packet is transmitted by the sender's timestamping engine, the value of the checksum remains U as before:

$$U = Const + checksum(T) = Const + checksum(T) + checksum(T') - checksum(T') = Const + checksum(T') + checksum(C) \quad (3)$$

Thus, after the timestamping engine has updated the timestamp, U remains the correct checksum of the packet.

When the test packet reaches the receiver, the receiver performs a conventional UDP Checksum computation, and the computed value is U. Since the Checksum Complement is part of the padding, the value of checksum(C) is transparently included in the computation, as per Equation (3), without requiring special treatment by the receiver.

Acknowledgments

The author gratefully acknowledges Al Morton, Greg Mirsky, Steve Baillargeon, Brian Haberman, and Spencer Dawkins for their helpful comments.

Author's Address

Tal Mizrahi
Marvell
6 Hamada St.
Yokneam, 20692
Israel

Email: talmi@marvell.com