| | |
|---|---|
| Project | **IEEE 802.16 Broadband Wireless Access Working Group <http://ieee802.org/16>** |
| Title | CRC32 Clarifications |
| Date Submitted | **2005-07-22** |
| Source(s) | Ilan Zohar, Intel corp. — Ilan.zohar@intel.com, Voice: +972-3-9205842<br>Kyungjoo Suh, Jaehwan Chang, Jinwoo Roh, Jaehee Cho, Jeongheon Kim, Seungjoo Maeng, Panyuh Joo, Geunhwi Lim, Samsung Electronics — joo.suh@samsung.com, jaehwan.chang@samsung.com, Voice: +82-31-279-5123 |
| Re: | Sponsor Ballot on IEEE P802.16-2004/Cor1/D3 |
| Abstract | While in wide use, the definition of CRC32 has always required understanding of the context or the help of test vectors. This contribution offers a standalone definition of CRC32. |
| Purpose | This contribution clarifies the definition of CRC32 |
| Notice |  |
| Release |  |
| Patent Policy and Procedures |  |

# CRC32 clarifications

**Ilan Zohar – Intel Corporation**
**Kyungjoo Suh, Jaehwan Chang, Jaehee Cho – Samsung Electronics**

## 1. Motivation

The standard uses CRC32 to detect possible PDU transmission errors and refers to 802.3 for its definition. CRC32 is used in other contexts as well (e.g. 802.11). In both cases implementing CRC32 requires either relying on test vectors or understanding not only of the definition of CRC32 istelf but its context as well (such as the manner of transmission in 802.3).
This contribution presents a standalone definition of CRC32.

## 2. Proposed changes

*[Modify text in 6.3.3.5 as follows:]*

**6.3.3.5 CRC calculation**
A service flow may require that a CRC be added to each MAC PDU carrying data for that service flow (11.13.12). In this case, for each MAC PDU with HT=0 and CI=1, a CRC32, as defined in 6.3.3.5.1 for **SC, SCa, and** OFDM mode and 6.3.3.5.2 for OFDMA mode ~~IEEE 802.3,~~ shall be appended to the payload of the MAC PDU; i.e., request MAC PDUs are unprotected. The CRC shall cover the generic MAC header and the Payload of the MAC PDU. The CRC shall be calculated after encryption; i.e. the CRC protects the Generic Header and the ciphered Payload.

### 6.3.3.5.1 CRC32 calculation for SC, SCa, and OFDM mode

The data (input) bytes shall be flipped (for each byte exchange bit0 ←→bit7, bit1 ←→bit6, bit2 ←→bit5, and bit3 ←→bit4)

The CRC32 shall be calculated using the following standard generator polynomial of degree 32:

$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x + 1$
(where, the hexadecimal representation of truncated G(x) is "0x04c11db7")

The CRC32 is the 1's complement of the sum (modulo 2) of the following:

1. The remainder of $x^k$ $(x^{31} + x^{30} + x^{29} + \ldots + x^2 + x + 1)$ divided (modulo 2) by $G(x)$, where $k$ is the number of bits in the input data, and
2. The remainder after multiplication of the bit-flipped input data (treated as a polynomial) by $x^{32}$ and then division by $G(x)$.

The CRC32 field shall then be transmitted bit-flipped commencing with the most significant byte (the first transmitted byte will have in its bit7 the coefficient of $x^{24}$ and in bit0 the coefficient of $x^{31}$ .The fourth byte will have the coefficient of $x^0$ in bit 7 and the coefficient of $x^7$ in bit0).

As a typical implementation, at the transmitter, the initial remainder of the division is preset to all 1's and is then modified by division of the bit-flipped data by the generator polynomial $G(x)$. The 1's complement of this remainder is then bit flipped byte after byte when transmitted, with the most significant byte first.

At the receiver, the initial remainder is preset to all 1's and the input bytes shall be flipped first and then treated as coefficient of a polynomial. When divided by $G(x)$, this polynomial shall results in the absence of transmission errors, in a unique nonzero remainder value. The unique remainder value is the polynomial:

$x^{31} + x^{30} + x^{26} + x^{25} + x^{24} + x^{18} + x^{15} + x^{14} + x^{12} + x^{11} + x^{10} + x^{8} + x^{6} + x^{5} + x^{4} + x^{3} + x + 1$
(or as its hexadecimal representation 0xC704DD7B)

#### 6.3.3.5.1.1 CRC32 test vectors for SC, SCa, and OFDM mode

The following is an example of CRC calculation in SC, SCa and OFDM mode.

Generic MAC Header = 40 40 1A 06 C4 5A
Payload = BC F6 57 21 E7 55 36 C8 27 A8 D7 1B 43 2C A5 48
CRC32 for SC, SCa and OFDM mode = CB B6 5F 48

#### 6.3.3.5.2 CRC32 calculation for OFDMA mode

The data (input) bytes shall not be flipped as in OFDM mode.

The CRC32 shall be calculated using the following standard generator polynomial of degree 32:

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x + 1$$
(where, the hexadecimal representation of truncated G(x) is "0x04c11db7")

At the transmitter, the following procedure is applied;

1. First 32 bits are complemented, which is equivalent to setting the initial value of the CRC register as 0xFFFFFFFF.
2. The first bit of the first field (MSB of the first byte of the MAC header) corresponds to the $x^{n-1}$ term and the last bit of the last field corresponds to the $x^0$ term, where n is the number of bits in the input data sequence.
3. The resulting polynomial multiplied by $x^{32}$ is divided by G(x).
4. The remainder bit sequence is complemented.
5. The 32 bits of the CRC value are placed in the CRC field so that the $x^{31}$ term is the left-most bit of the first byte, and the $x^0$ term is the right most bit of the last byte.
6. The resulting CRC field is sent MSB first (6.3.3.1).

At the receiver, the initial remainder is preset to all 1's and the input bytes shall be fed into the CRC engine MSB first. When divided by $G(x)$, this polynomial shall result in the absence of transmission errors, in a unique nonzero remainder value. The unique remainder value is the polynomial:

$x^{31} + x^{30} + x^{26} + x^{25} + x^{24} + x^{18} + x^{15} + x^{14} + x^{12} + x^{11} + x^{10} + x^{8} + x^{6} + x^{5} + x^{4} + x^{3} + x + 1$
(or as its hexadecimal representation 0xC704DD7B)

#### 6.3.3.5.2.1 CRC32 test vectors for OFDMA mode

The following is an example of CRC calculation in OFDMA mode.

Generic MAC Header = 40 40 1A 06 C4 5A
Payload = BC F6 57 21 E7 55 36 C8 27 A8 D7 1B 43 2C A5 48
CRC32 for OFDMA mode = 1B D1 BA 21

### 7.5.1.2.5 AES-CCM Mode Example Encrypted MPDUs

*[Modify line 20 of page 70 as indicated below.]*

CRC32 for SC, SCa, and OFDM mode = CB B6 5F 48
CRC32 for OFDMA mode = 1B D1 BA 21

*[Modify line 43 of page 70 as indicated below.]*

CRC32 for SC, SCa, and OFDM mode = 92 1B 32 41
CRC32 for OFDMA mode = FD 03 7B 1D