

Örnek: Frame: 1101011011

Generator: 10011

Bölge 5 bit

5-1=4 bit CRC

Frame 4 sifir ekle ve böl.

X 1100001010	
10011	11010110110000
10011	10011
010011	00000
10011	00000
000001	00000
00000	00000
000010	00000
00000	00000
000101	00000
001011	00000
010110	00000
10011	00000
001010	00000
010100	00000
10011	00000
001110	00000
01110	00000
	CRC = 1110

Gönderilecek veri: 1101011011110

Örnek: Dataword : $x^3 + 1$

Üreteç fonksiyonu : $x^3 + x + 1$

Üreteç fonksiyonunun en yüksek derecesi ile dataword çarpılır.

Gönderici

$$x^3(x^3 + 1) = x^6 + x^3$$

$$\begin{array}{r} x^3 + x \\ \hline x^6 + x^3 + x^3 \\ x^6 + x^4 + x^3 \\ \hline x^4 \\ x^4 + x^2 + x \\ \hline x^2 + x \end{array}$$

$$\begin{array}{r} x^6 + x^5 + x^4 + x^3 \\ \hline x^3 + x + 1 \end{array}$$

$$CRC = x^2 + x$$

Gönderilecek dataword : $x^6 + x^3 + x^2 + x$

Alici

$$\begin{array}{r} x^3 + x \\ \hline x^6 + x^3 + x^2 + x \\ x^6 + x^4 + x^3 \\ \hline x^4 + x^2 + x \\ x^4 + x^2 + x \\ \hline 0 \end{array} \leftarrow \text{Sonuç} = 0 \text{ hata yok}$$

- Kontrol Toplami (Checksum)

TCP / IP yığınınındaki protokollerde başlık ya da başlıklar beraber verinin iletiminde bir hatanın olup olmadığını anlamada kullanılır.

Gönderilecek veri bitlerinin toplamı alınır ve sonra bulunan toplam, veri bitlerine eklenerek öyle gönderilir.

Toplamalar türmleyen aritmetigine göre yapılır.

Alici gelen sayıları toplar ve sonra türmleyenini alır, sonra o ise hata yoktur. Sıfırdan farklı ise yeniden iletim yapılır.

Gönderici

7
11
12
0
6
0

36, 4 bit ile ifade edilemediğinden

$$\begin{array}{r} 100100 \\ 10 \\ \hline \end{array}$$

11'e türmleyen $\rightarrow 0110 \rightarrow$ 6 ambalajlanmış toplam
 $\hline 1001 \rightarrow 9$ checksum

Toplam 36
Ambalaj Toplam 6
Checksum 9

Veri paketi : 7, 11, 12, 0, 6, 9 halinde gönderilir.

Alici

7
11
12
0
6
9

45, 4 bit ile ifade edilemediğinden

$$\begin{array}{r} 101101 \\ 10 \\ \hline \end{array}$$

11'e türmleyen $\rightarrow 1111 \rightarrow$ 15 ambalajlanmış toplam
 $\hline 0000 \rightarrow$ checksum

Toplam 45
Ambalaj Toplam 15
Checksum 0 \rightarrow Hata yok

Örnek: 4500 0095 0000 4000 3F11 BE40 CE3F EDO4 COA8 0133
 bit dizisi için kontrol toplamını hesaplayınız. Bu bit dizisi bir IP başlık
 örneğini göstermektedir.

IP başlığı 20 byte olur, 16 bitlik alanları ile

4500 0095 0000 4000 3F11 0000 CE3F EDO4 COA8 0133
 16 checksum BE40

16 bitlik çevir

Hex	Binary	1'e Tümlenilen
4500	0100 0101 0000 0000	1011 1010 1111 1111
0095	0000 0000 1001 0101	1111 1111 0110 1010
0000	0000 0000 0000 0000	1111 1111 1111 1111
4000	0100 0000 0000 0000	1011 1111 1111 1111
3F11	0011 1111 0001 0001	1100 0000 1110 1110
CE3F	1100 1110 0011 0111	0011 0001 1100 1000
ED04	1110 1101 0000 0100	0001 0010 1111 1011
COA8	1100 0000 1010 1000	0011 1111 0101 0111
0133	0000 0001 0011 0011	1111 1110 1100 1100

Sayıların toplamını bulup tümleneni
 almak yerine direkt tümlenelerin
 toplamını alırsak checksum buluruz.

$$\begin{array}{r}
 1011 1110 0100 0000 \\
 + \\
 1011 1110 0100 0000 \\
 \hline
 B \quad E \quad 4 \quad 0 \quad \leftarrow \text{checksum}
 \end{array}$$

?

Hata Düzeltme Tekniği

- Hamming Kodlama

Mesafe Özelliği ile beraber kullanılır. Örneğin; mesafe değeri 2 ise, alıcıda 1 bitlik hatalar sezilir ve düzelttilir. 2 bitlik hatalar sadece sezilir.

Bu literatürde HD,2 olarak isimlendirilir.

Bit Konumu	Konum Numarası	Test Biti	Veri Biti
12	<u>1100</u>	-	m_8
11	<u>1011</u>	-	m_7
10	<u>1010</u>	-	m_6
9	<u>1001</u>	-	m_5
8	<u>1000</u>	C_4	-
7	<u>0111</u>	-	m_4
6	<u>0110</u>	-	m_3
5	<u>0101</u>	-	m_2
4	<u>0100</u>	C_3	-
3	<u>0011</u>	-	m_1
2	<u>0010</u>	C_2	-
1	<u>0001</u>	C_1	-

Test Biti \rightarrow Sadece birtane 1 olanlara $C_1, C_2, C_3 \dots$ şeklinde yazıyoruz (C)

Veri Biti \rightarrow C olmayan yerlere m_1, m_2, m_3, \dots şeklinde yazıyoruz (M)

$$C_1 = m_1 \oplus m_2 \oplus m_4 \oplus m_5 \oplus m_7 \rightarrow 1. \text{biti 1 olsun}$$

$$C_2 = m_1 \oplus m_3 \oplus m_4 \oplus m_6 \oplus m_7 \rightarrow 2. \text{biti 1 olsun}$$

$$C_3 = m_2 \oplus m_3 \oplus m_4 \oplus m_8 \rightarrow 3. \text{biti 1 olsun}$$

$$C_4 = m_5 \oplus m_6 \oplus m_7 \oplus m_8 \rightarrow 4. \text{biti 1 olsun}$$

EXOR		
A	B	Cikis
0	0	0
0	1	1
1	0	1
1	1	0

0011
0101
0111
1001
10