

DIJİTAL VERİ İLETİŞİM TEKNIKLERİ

Chapter 6 – Digital Data Communications Techniques

Eighth Edition
by William Stallings

Lecture slides by Lawrie Brown

Dijital Veri İletişim Teknikleri

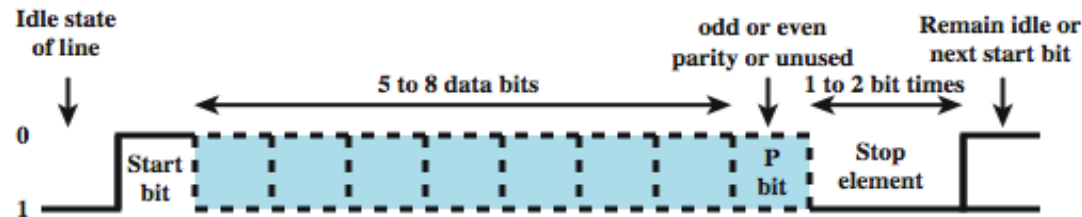
- *Bir konuşma iki yönlü iletişim hattı oluşturur;iki taraf arasında bir simetri oluşur ve mesajlar karşılıklı gönderilip alınır. Anlık sürekli cevaplar, tekrarlayan işlemler, birbirini izleyen yorumlarla iki birey birlikte planlanmış, paylaşımcı ve aynı hedefe odaklanmış bir davranış sergilerler. Bu gerçek bir iletişimdir.*

—*On Human Communication*, Colin Cherry

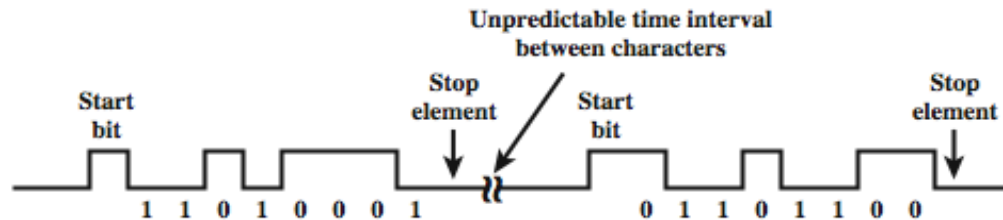
Asenkron ve Senkron İletim

- ⦿ Zamanlama probleminden dolayı gönderici ve alıcı arasında senkronizasyon mekanizmasına gerek duyulur
 - Alıcı, bir bit süresince veri katarından örnekleme yapar
 - Eğer zamanlama denetimi iki taraf arasında dengeli olmazsa örnekleme yanlış zamanda gerçekleşebilir ve belli sayıda bit sonrasında hatalı bit okunur
- ⦿ Zamanlama denetimi için iki çözüm önerilebilir
 - Asenkron iletim
 - Senkron iletim

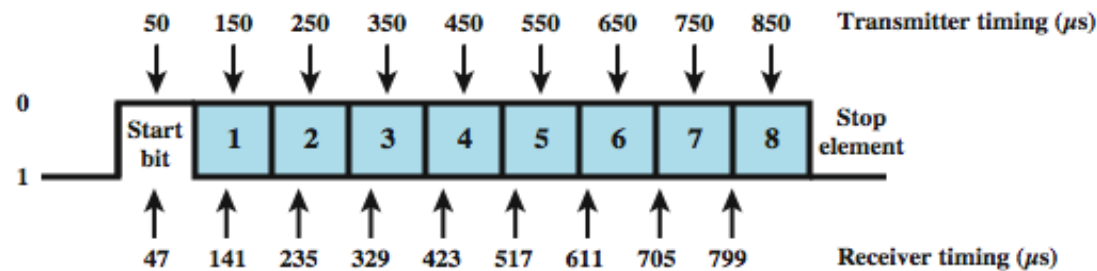
Asenkron İletim



(a) Character format



(b) 8-bit asynchronous character stream



(c) Effect of timing error

Asenkron İletim

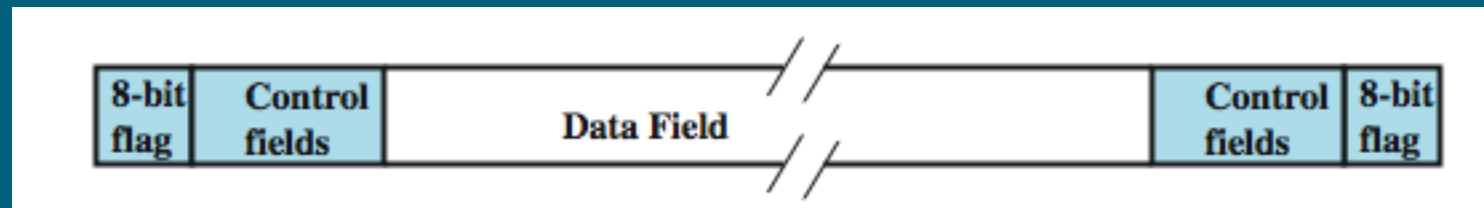
- Veri içindeki her bir karakter ayrı olarak işlem görür.
- Zamanla problemini önlemek için belirli süre içinde bir karakter gönderilir.
- Karakterin başında bir **başlangıç biti – start bit** (binary 0) gönderilerek alıcının karakterin geldiğini anlaması sağlanır.
- 5 ila 8 bit uzunluğundaki karakter en önemsiz bitten başlayarak gönderilir.
- Veri bitlerinin sonuna **çift (even)** ya da **tek (odd) eşlik (parity)** biti eklenir.
- En son eleman **bitiş (stop)** 1, 1,5 veya 2 bit uzunluğunda olup (binary 1) değerindedir.
- Alıcı karakter içindeki her bir biti örnekler ve sonraki karakteri bekler. **Boşta bekleme durumu - idle state** (binary 1)

Asenkron Özellikleri

- Basit
- Ucuz bir çözüm
- Senkronizasyon amacıyla her bir karakter için 2 veya 3 ilave bit ($\sim\%20$)
- Daha fazla bit blok olarak gönderilebilir. Ancak önceki şekilde gözüktüğü gibi zaman denetimindeki farktan kaynaklanan hata birikerek verilerde bozulmaya sebep olabilmekte.
- Uzun aralıklarla gönderilen veriler için uygun bir çözümdür. Klavyeden veri girişi gibi

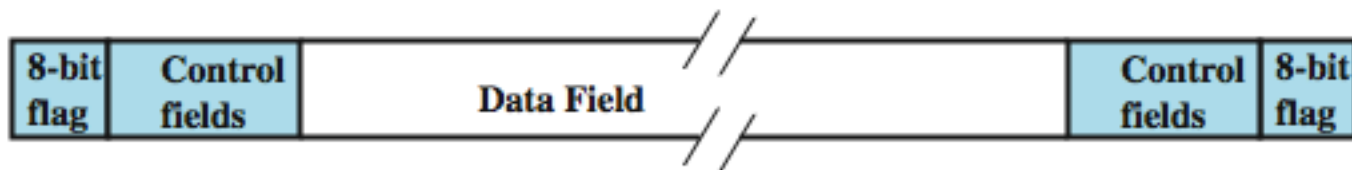
Senkron İletişim

- Veri bloğu bir çerçeve (frame) olarak iletilir.
- Gönderici ve alıcının zamanları senkronize olmalı
 - Ayrı bir zaman hattı kullanılabilir
 - Veya zaman sinyali veri içine yerleştirilir (Manchester e Diferansiyel Manchester kodlama)
 - Analog sinyallerde taşıyıcı frekansı senkronizasyon için kullanılır
- Çerçeve başı ve sonu bayrak (flag) ile belirlenir.
 - Başlangıç bayrağı (preamble), bitiş bayrağı (postamble)
- Veri bitleri önünde ve arkasında kontrol bitleri yer alır.
- Asenkron haberleşmeye göre daha verimli. Daha az ilave bit içerir. Kontrol bilgisi, başlangıç ve bitiş bayrakları 100 bitten azdır.



Senkron İletim

- Çerçeve: başlangıç, veri, bitiş ve kontrol alanları
- Başlangıç ve bitiş: 8 bits
- Kontrol alanı: Veri Bağlantı kontrol protokol bilgisi (Data link control protocol information)
- Veri alanı: Değişken büyüklükte
- İlave bitler 100 bitten az
- Asenkron iletim %20 ilave veri kullanır.



İletim Tiplerini Karşılaştırma

- Asenkron iletim %20 ilave veriye ihtiyaç duyar.
- HDLC (High level data link control)
 - Başlangıç, bitiş ve kontrol için: 48 bit
- Örnek: 1000 karakter için
 - HDCL
 - Toplam uzunluk: $1000 \times 8 = 8000$ bit
 - Her çerçevedeki fazlalık= 48 bit
 - İlave yüzdesi= $48/8000 = \%0,6$
 - Asenkron İletim
 - İlave yüzdesi = % 20
 - Toplam ilave veri = $\%20 \times 8000 = 1600$ bit

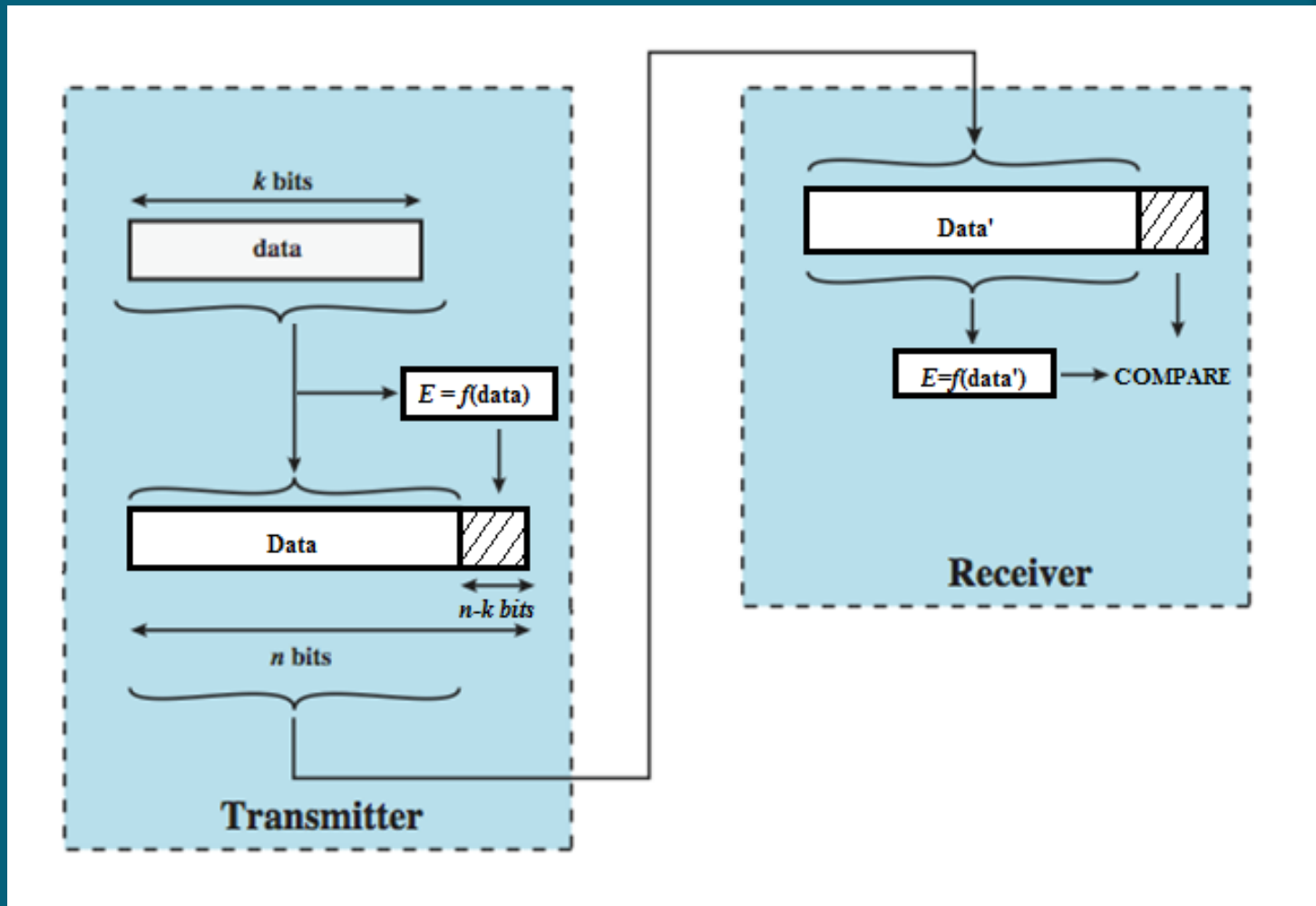
Hata Tipleri

- Gönderim ve alım sırasında bitlerde değişme meydana gelirse (1'den 0'a veya 0'dan 1'e) hata oluşur.
- İki tip hata oluşabilir
 - Tek bit hataları (rastgele hata)
 - Yalnızca bir bitte dönüşüm olup diğer bitler etkilenmez
 - İletim yolundaki gürültü (white noise) sebep olabilir
 - Patlama hatası (burst error)
 - Belli bir zaman diliminde arak arkaya bitlerin bozulması
 - Parazitler veya kablosuz iletişimde zayıflama ataya sebep olabilir
 - Yüksek veri iletim hızında hatalar da daha fazla olur.
 - 10 Mbps için, 1 μ s lik parazit -> 10 bitte hata
 - 100 Mbps için, 1 μ s lik parazit -> 100 bitte hata

Hata Sezme

- İletim sisteminin tasarımından bağımsız olarak iletilen çerçevede bir ya da birkaç bitin değişmesi şeklinde hatalar oluşabilir.
- Hatalar ilave kod vasıtasıyla sezilir.
- İletilen bitlere kod eklenir
- Alıcı gelen kodu çözümleyerek hata kontrolü yapar
- Koda rağmen sezilemeyen hatalar olabilir.

Hata Sezme İşlemi



Eşlik Sınaması

- En basit hata sezme tekniğidir
- Karakter iletiminde kullanılır
- Eklenecek eşlik biti 1 lerin toplam sayısını tek (odd) ya da çift (even) yapacak şekilde seçilir.
- Örnek
 - Gönderilen: Veri Bitleri: 1110101 Eşlik Biti: 1 (Tek eşlik)
 - Alınan: Veri Bitleri: 1110001 Eşlik Biti: 1 (Tek eşlik)
 - 1'lerin sayısı tek olmadığı için hata sezilmiş olur
- Tek sayıda bitteki değişim sezilebilir.
- Çift sayıda oluşacak hatanın algılanması mümkün olmamaktadır.

Çevrimli Fazlalık Sınaması

Cyclic Redundancy Check (CRC)

- En yaygın ve en güçlü bir hata sezme tekniğidir.
- Gönderici k bite $n-k$ kontrol biti (FCS) ilave ederek n bitlik çerçeve (frame) oluşturur.
- n bit tam olarak bazı önceden belirlenen sayılara bölünecek şekilde oluşturulur
- Alıcı, çerçevedeki verileri belirlenen sayıya böler
 - Bölme işlemi sonucu kalan olmazsa hata olmadığı anlaşılır
- Bu işlemde Modulo 2 aritmetiği, polinom ve dijital lojik ten yararlanır.

Çevrimli Fazlalık Sınaması

Cyclic Redundancy Check (CRC)

1. Veriye dizisi $P(x)$ polinomuna dönüştürülür; polinom katsayıları ilgili bitin değeri (1 veya 0)
 - Örnek:
 - Veri dizisi: 1 0 1 1 0 1 0 1 0 1
 - $P(x) = 1.x^9 + 0.x^8 + 1.x^7 + 1.x^6 + 0.x^5 + 1.x^4 + 0.x^3 + 1.x^2 + 0.x^1 + 1.x^0$
 $= x^9 + x^7 + x^6 + x^4 + x^2 + 1$
2. $P(x)$ polinomu x^{n-k} ile çarpılır
 - Yeni polinoma karşı düşen veri dizisi ilk dizinin sonuna $(n-k)$ adet 0 bitinin eklenmiş halidir
 - Veri dizisi: 1 0 1 1 0 1 0 1 0 1 0 0 0 0
3. $X^p.P(x)$ polinomu $(n-k)$. dereceden $G(x)$ polinomuna (üreteç – generating) bölünür
 - Bilinen bazı $G(x)$ polinomlar
 - $P(x) = x^{16} + x^{15} + x^2 + 1$
 - $P(x) = x^{16} + x^{12} + x^5 + 1$
 - $P(x) = x^{12} + x^{11} + x^3 + x^2 + x + 1$

Çevrimli Fazlalık Sınaması

Cyclic Redundancy Check (CRC)

4. Gönderici $X^{n-k}.P(x) / G(x)$ işlemini yaparak $Q(x)$ bölüm ve $R(x)$ kalanı bulur
 $X^{n-k}.P(x) = Q(x). G(x) + R(x)$

Yapılan işlemlerde modulo 2 aritmetiği kullanılırsa;

$$0 + 0 = 0$$

$$0 - 0 = 0$$

$$0 + 1 = 1$$

$$0 - 1 = 1$$

$$1 + 0 = 1$$

$$1 - 0 = 1$$

$$1 + 1 = 0$$

$$1 - 1 = 0$$

Buna göre;

$$X^{n-k}.P(x) - R(x) = Q(x). G(x)$$

$$X^{n-k}.P(x) + R(x) = Q(x). G(x)$$

Gönderici alıcıya $P(x)$ yerine $X^{n-k}.P(x) + R(x)$ polinomuna karşı düşen veri dizisini gönderir

Yeni dizi k bitlik veri dizisi ve onun sonuna eklenmiş $n-k$ bitlik ek diziden oluşur

Gönderilen dizi alıcı tarafından önceden bilinen $G(x)$ polinomunun tam katıdır.

Alıcı gelen dizinin karşılık düştüğü polinomu $G(x)$ 'e böldüğünde

Hata yoksa; Kalan – $Q(x) = 0$ olmalıdır

Hata varsa ; Kalan – $Q(x) \neq 0$ olacaktır.

Cyclic Redundancy Check (CRC)

Örnek

$$P(x) = x^9 + x^7 + x^4 + x^2 + x + 1 \text{ ve} \\ G(x) = x^4 + x^2 + x + 1 \text{ olarak seçilsin}$$

Veri Dizisi (1010010111)

Buna göre veri biti sayısı $k=10$, $n-k=4$ ve $n=14$ (çerçeve büyüklüğü) dir.

$$X^4.P(x) = x^{13} + x^{11} + x^8 + x^6 + x^5 + x^4 \text{ olur} \quad \text{Veri Dizisi (10100101110000) En sona 4 adet sıfır eklendi.}$$

$X^4.P(x)$ 4. dereceden $G(x)$ polinomuna bölünürse

$$x^{13} + x^{11} + x^8 + x^6 + x^5 + x^4 \quad x^4 + x^2 + x + 1$$

$$x^{13} + x^{11} + x^{10} + x^9 \quad x^9 + x^6 + x^5 + x^2$$

$$x^{10} + x^9 + x^8 + x^6 + x^5 + x^4$$

$$x^{10} + x^8 + x^7 + x^6$$

$$x^9 + x^7 + x^5 + x^4$$

$$x^9 + x^7 + x^6 + x^5$$

$$x^6 + x^4$$

$$x^6 + x^4 + x^3 + x^2$$

$$x^3 + x^2$$

$$X^4.P(x) = x^{13} + x^{11} + x^8 + x^6 + x^5 + x^4$$

$$G(x) = x^4 + x^2 + x + 1$$

$$Q(x) = x^9 + x^6 + x^5 + x^2$$

$$R(x) = x^3 + x^2$$

Bölünen
Bölen
Bölüm
Kalan

Gönderilecek Dizinin Karşı Düştüğü Polinom

$$Q(x).G(x) = X^4.P(x) + R(x) = x^{13} + x^{11} + x^8 + x^6 + x^5 + x^4 + x^3 + x^2$$

Veri Dizisi

1010010111100

Veri Bitleri CRC Bitleri

CRC - Örnek

Alınan Polinom

$$Q(x).G(x) = x^{13} + x^{11} + x^8 + x^6 + x^5 + x^4 + x^3 + x^2,$$

$$G(x) = x^4 + x^2 + x + 1$$

$Q(x).P(x)$, $G(x)$ e bölünürse

$$\begin{array}{r}
 x^{13} + x^{11} + x^8 + x^6 + x^5 + x^4 + x^3 + x^2 \\
 \underline{x^{13} + x^{11} + x^{10} + x^9} \\
 x^{10} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + x^2 \\
 \underline{x^{10} + x^8 + x^7 + x^6} \\
 x^9 + x^7 + x^5 + x^4 + x^3 + x^2 \\
 \underline{x^9 + x^7 + x^6 + x^5} \\
 x^6 + x^4 + x^3 + x^2 \\
 \underline{x^6 + x^4 + x^3 + x^2} \\
 0
 \end{array}
 \begin{array}{r}
 x^4 + x^2 + x + 1 \\
 x^9 + x^6 + x^5 + x^2
 \end{array}$$

$$Q(x).G(x) = x^{13} + x^{11} + x^8 + x^6 + x^5 + x^4$$

$$G(x) = x^4 + x^2 + x + 1$$

$$Q(x) = x^9 + x^6 + x^5 + x^2$$

$$R(x) = 0$$

Kalan 0 olduğu için Hata yok.

Alınan Veri

(10100101111100)

Bölünen

Bölen

Bölüm

Kalan

Hata Düzeltme

- ⦿ Bozulan veri bloğunun düzeltilmesi çoğunlukla verinin tekrar gönderilmesi ile sağlanır
- ⦿ Kablosuz haberleşmede bu teknik uygun bir çözüm oluşturmaz
 - Bit hata oranı (bit error rate – BER) yüksek olduğundan çok sayıda tekrardan gönderim gerekir.
 - Uydu haberleşmelerinde erişim süresi, iletim süresine göre çok uzun olduğundan tek çerçevede oluşan hata için ilgili çerçeveyi takip eden birden fazla çerçeve tekrar gönderilir.
- ⦿ Bunun yerine hatalı bitlerin düzeltilmesine çalışılır
- ⦿ Hata düzeltme teknikleri hata sezme tekniklerinde olduğu gibi bazı hatalı bitlerin düzeltilmesini sağlar.

CRC Yönteminin Özellikleri

- $Q(x)$. $G(x)$ polinomu yerine alıcıya $S(x) = Q(x) G(x) + E(x)$ polinomu hatalı olarak iletilirse;

Örnek:

- Gönderilen dizi: 1 0 1 0 0 1 0 1 1 1 1 1 0 0 yerine
- Alınan dizi: 1 0 1 1 0 0 1 1 1 0 1 0 0 alınır

- $S(x) = x^{13} + x^{11} + x^{10} + x^6 + x^5 + x^4 + x^2$

$$E(x) = S(x) - Q(x).G(x) = S(x) + Q(x).G(x)$$

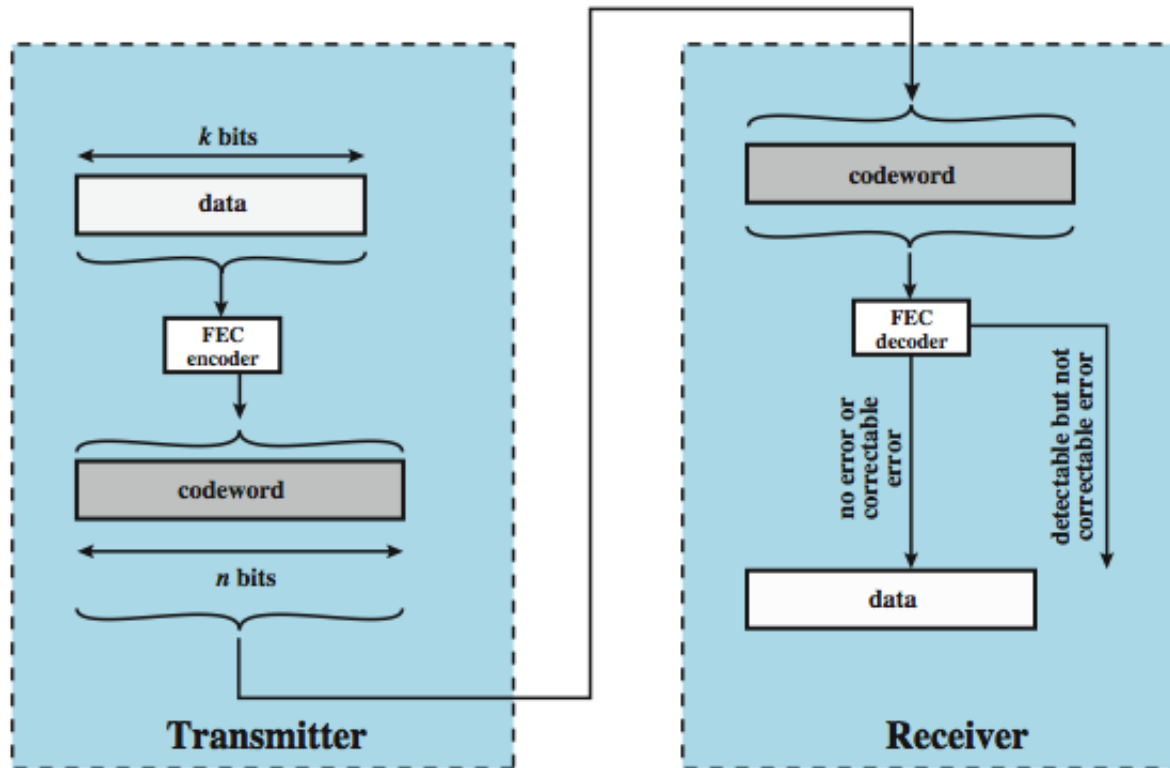
$$= (x^{13} + x^{11} + x^{10} + x^6 + x^5 + x^4 + x^2) - (x^{13} + x^{11} + x^8 + x^6 + x^5 + x^4 + x^3 + x^2)$$
$$= x^{10} + x^8 + x^3$$

- $E(x)$ in terim sayısı bozulan bit sayısına eşittir. Terimlerin dereceleri de bit pozisyonunu belirler.
- $E(x)$, $G(x)$ 'in tam katı değilse kalan alıcıda 0 olmayacağından hata fark edilecektir.
- $G(x)$ özel olarak seçildiğinden tek sayıdaki bit dönüşümü kalan 0 olacağından farkedilir
- Çift sayıda bit bozulduğunda bile hata sezme ihtimali yüksektir.
- Patlama bozulmasında bozulan bit sayısının $(n-k)$ dan az olması gerekir.

Hamming Kodlaması

- Hata düzeltmede kullanılan en yaygın yöntemdir
- En fazla j bitin ozulma ihtimali olduğu durumda oluşturulan koda j -bit hata bağıışıklığı olan kod denilir.
- Hamming (7,4) kodunda 4 adet bilgi ve 3 adet sına ma biti yer alarak 7 bitlik kod oluşturulur.
- Kod : $I_1 I_2 I_3 I_4 C_1 C_2 C_3$
 $C_1 = I_1 I_2 I_3$ o'e ilişkin çift eşlik biti
 $C_2 = I_1 I_3 I_4$ o'e ilişkin çift eşlik biti
 $C_3 = I_2 I_3 I_4$ o'e ilişkin çift eşlik biti
- Sın a ma bitlerinin hesaplanmasında XOR kllanılır
 $C_1 = I_1 \oplus I_2 \oplus I_4$
 $C_2 = I_1 \oplus I_3 \oplus I_4$
 $C_3 = I_2 \oplus I_3 \oplus I_4$
- Örnek
 - Kod : $I_1 I_2 I_3 I_4 C_1 C_2 C_3$
 $1 0 0 1 0 0 1$
 - Hatalı Kod: $1 1 0 1 0 0 1$
 $C_1 = 1$ (gönderilen 0)
 $C_2 = 0$ (gönderilen 0)
 $C_3 = 0$ (gönderilen 1)
 - I_2 deki hata C_1 ve C_3 'de ortaya çıkıyor.

Hata Düzeltme İşlemi



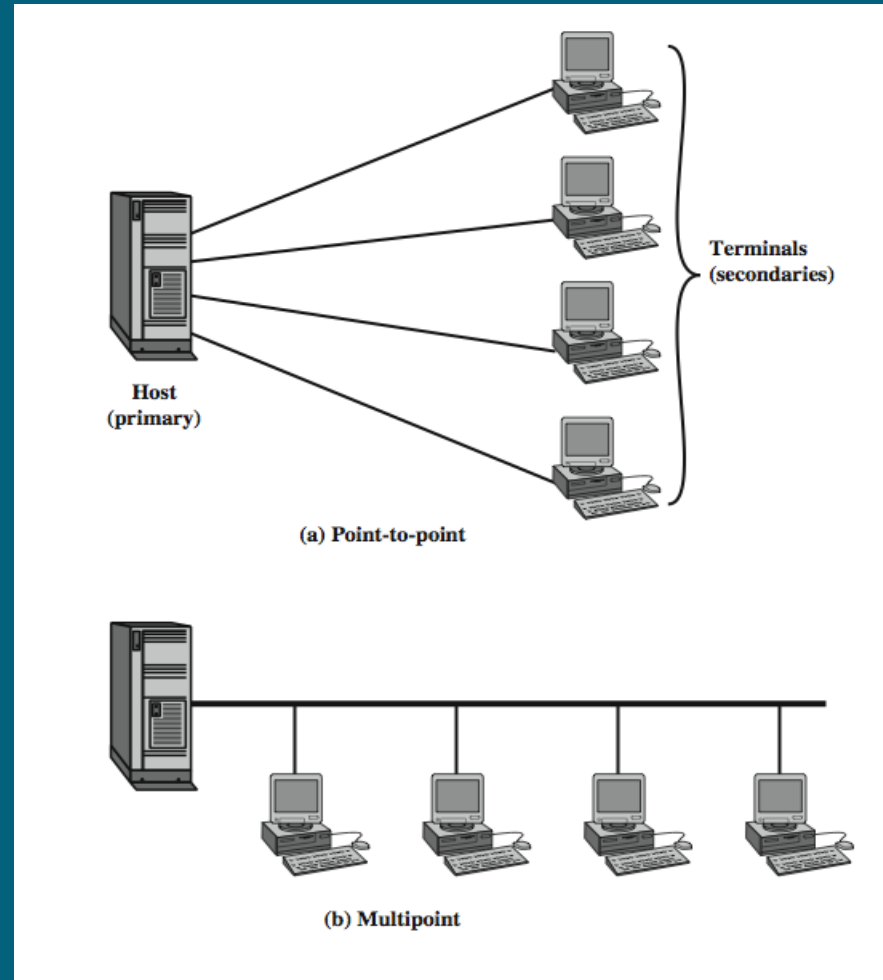
Hata Düzeltme Süreci

- Gönderilen mesaja fazlalık bitler eklenir.
- İlave bitler alıcının orijinal mesajın çıkarmasına yardımcı olur
- En yaygın teknik *blok hata düzeltme kodudur (block error correction code)*
 - k adet bit n adet bitten oluşan koda dönüştürülür
 - Her bir kod birbirinden farklıdır
 - Alıcıya hatalı kod ulaşırsa doğru kod tahmin edilmeye çalışılır
- Fazlalık bitlerin $(n-k)$ veri bitlerine (k) oranı $(n-k)/k$ kodun fazlalığı (redundancy of code) olarak ifade edilir.
- k/n ise kod oranı (code rate) olarak ifade edilir.
- Kod oranı $\frac{1}{2}$ için iki kat iletim kapasitesi gerekir.

Bağlantı Konfigürasyonu- Topoloji

- ⦿ İletim ortamındaki istasyonların fiziksel olarak yerleşimi
 - Uçtan uca (point to point)
 - İki istasyonun bağlantısı
 - İki router veya bilgisayar arasındaki haberleşme gibi
 - Çoklu uç (multi point)
 - İkiden fazla istasyonun bağlantısı
 - Mainframe ve terminal haberleşmesi bu yöntemi kullanır
 - Günümüzdeki yerel alan ağı (local area network - LAN) bu yapıyı kullanır

Bağlantı Konfigürasyonu - Topoloji



Bağlantı Konfigürasyonu - Tipleri

- Veri iletiminde iletim hattı tek yönlü (simplex), yarı çift yönlü (half duplex), tam çift yönlü (full duplex) olarak sınıflandırılabilir
- Tek yönlü (simplex)
 - Tek yönlü iletim yapılabilir
 - Tek hat üzerinden iletim yapılır
 - Televizyon haberleşmesi gibi
- Yarı çift yönlü (half duplex)
 - Bir anda sadece tek istasyon iletim yapılabilir
 - Tek hat üzerinden iletim yapılır
 - Terminal bilgisayar haberleşmesi gibi
- Tam çift yönlü (full duplex)
 - Aynı anda iki istasyonun haberleşmesi sağlanır
 - İki bilgisayar haberleşmesi bu yöntemi kullanır

Özet

- ⦿ Asenkron ve senkron iletim
- ⦿ Hata sezme teknikleri
- ⦿ Hata düzeltme
- ⦿ Bağlantı konfigürasyon konuları