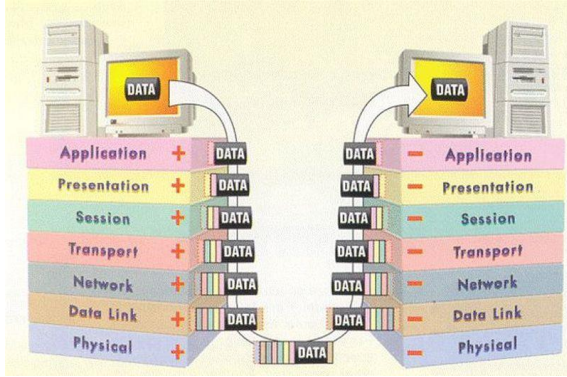


DATA LINK LAYER



Veri bağlantı katmanı OSI referans modelinin 2. katmanıdır.

Görevi gönderilecek sayısal verinin hatalara karşı kontrolünü sağlar.

Verinin ağ ortamında nasıl iletileceği, fiziksel

adreslemeyi, fiziksel kabloda çarpışma olmadan veri iletimini sağlar

Kısacası data link layer katmanında bir hata düzeltme/tespit etme sistemi vardır. Kaynak makinadaki ağ katmanından hedef makinadaki ağ katmanına veri aktarır.

CHECKSUM

İnternet üzerinde iletilen paketlerin bozulup bozulmadığının kontrolünü yapar. Bu yöntem paketin veri kısmındaki bilgileri toplar ve elde edilen bilgiyi ayrıca yollar. Alan kişi de veriyi toplayarak aynı kontrol toplamını (checksum) elde edip etmediğine bakar. Bu bozulup bozulmama kontrolünü bilindiği üzere TCP protokolü kullanmaktadır.

Örneğin:

Gönderilecek veri: **1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 1 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1** olarak verilmiş olsun bu veri 32 bit uzunluğundadır ve bizim paketlerimiz 16 bit uzunluğunda kabul edelim. Dolayısıyla veriyi 16 bitlik parçalara bölüp toplayacağız.

1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0
1 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
+

1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 (TOPLAM DEĞERİ VE 17 BİT UZUNLUĞUNDA DOLAYISIYLA 17. BİTTEKİ FAZLA OLAN VERİ İLE KENDİSİNİ TEKRAR TOPLUYORUZ)

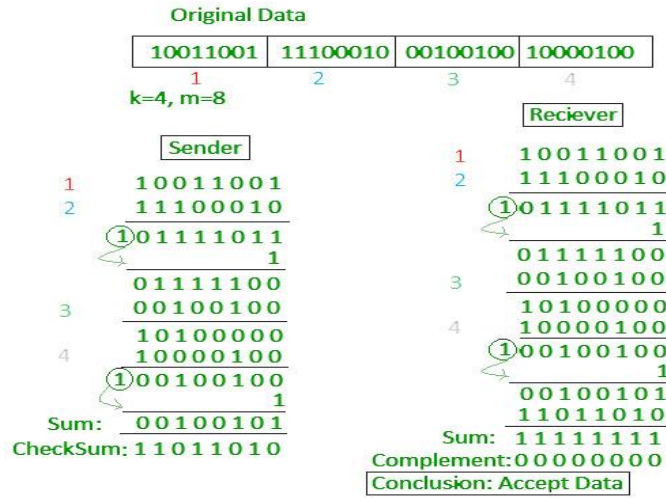
1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1
1

+

1 0 1 1 1 0 1 1 1 0 1 1 1 1 0 0 (VERİNİN 17. TAŞAN BİT İLE TOPLANMIŞ HALİ)

0 1 0 0 0 1 0 0 0 1 0 0 0 0 1 1 (SON OLARAK VERİNİN TERSİNİ ALIYORUZ, YANI 1 OLAN BİTLERİ 0 VE 0 OLAN BİTLERİ 1 YAPIYORUZ)

YUKARIDAKİ SON SATIRDA ELDE EDİLEN DEĞER 32 BİTLİK ORJİNAL PAKETİN TOPLAM KONTROL (CHECKSUM) DEĞERİDİR.



Bilgi yollanırken bu bilgi de ilave olarak pakete konulmakta ve karşı taraf aldığı zaman paketteki verilere aynı işlemleri uygulayarak sonucun doğru olup olmadığını kontrol etmektedir.

Alıcıya iletilen veri

1010001	00111001	00011101
---------	----------	----------

Data Checksum

ALICI KONTROLÜ

10101001 ALT BİRİM 1

00111001 ALT BİRİM 2

00011101 SAĞLAMA TOPLAMI

11111111 TOPLAM 00000000 TOPLAMIN TAMAMLAYICISI SONUÇ SIFIRDIR, HATA OLMADIĞI ANLAMINA GELİR.

Örnek 2 :

İletilecek veri birimi 10101001 00111001 ise gönderen makine ve alıcı makine aşağıdaki prosedür kullanılır.

GÖNDEREN MAKİNE

10101001 ALT BİRİM 1

00111001 ALT BİRİM 2

11100010 TOPLAM (1'LERİN TÜMLEYENİ KULLANILARAK)

00011101 SAĞLAMA TOPLAMI (TOPLAMIN TÜMLEYENİ)

FRAMİNG-ÇERÇEVELEME

Alınan bit sayısı iletilen bit sayısından daha az, eşit veya daha fazla olabilir ve farklı değerlere sahip olabilirler. Veri bağlantı katmanı bit akışını ayrı çerçevelere böler ve her çerçeve için checksum hesaplamasını yapar

Bir çerçeve hedefe ulaştığında, checksum yeniden hesaplanır.(yani hesaplanan checksum çerçevede bulunandan farklıysa, veri bağlantı katmanı bir hata oluştuğunu bilir ve bununla başa çıkmak için adımlar atar)

BİT AKIŞINI ÇERVELERE AYIRMA YÖNTEMLERİ:

- Character count
- Flag bytes with byte stuffing
- Starting and ending flags, with bit stuffing
- Physical layer coding violations

CHRACTER COUNT-KRAKTER SAYIMI

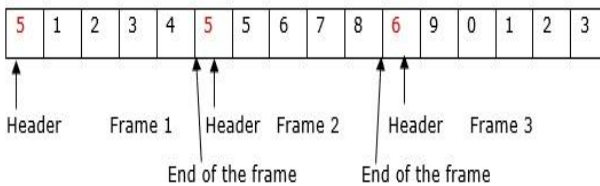
İlk çerçeveleme yöntemi, çerçevedeki karakter sayısını belirtmek için başlıktaki bir alanı kullanır. Hedefteki veri bağlantı katmanı karakter sayısını gördüğünde kaç karakterin takip ettiğini ve dolayısıyla çerçevenin sonunun nerede olduğunu bilir.

ÖRNEK VERİ : 1 2 3 4 5 6 7 8 9 0 1 2 3

Bu veriyi 3 çerçeveye bölelim: 1 2 3 4 5 6 7 8 9 0 1 2 3 ve bu çerçevelere chracter count atayalım.

Chracter count: çerçevedeki karakter sayısı + 1(chracter count dahil olduğu için)

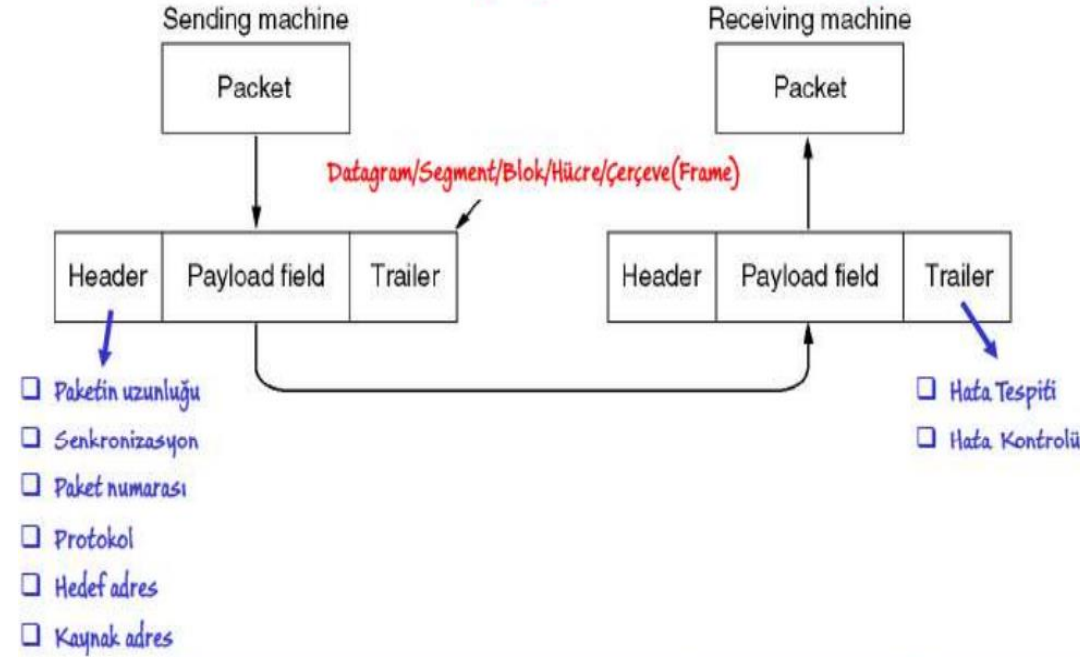
Now my compete data will transmit like:



Çerçevedeki başlangıç başlığı(header) karakter sayısını gösterir, dolayısıyla ilk çerçeve bu sayı dahil 5 veri biriminden oluşur.

Çerçeve kullanmanın avantajı, verilerin bozulma açısından kolayca kontrol edilebilecek kurtarılabılır parçalara bölünmesidir.

Paket Anahtarlama Ağı (Packet Switched Networks)



FLAG BYTES WITH BYTE STUFFING

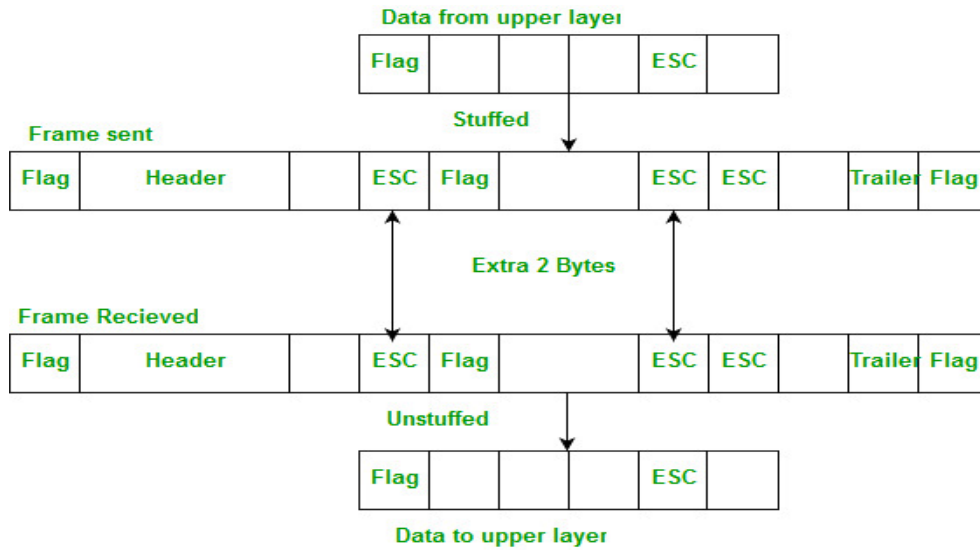
Flag byte : Bayrak biti, bir çerçevenin sonunu ve bir sonraki çerçevenin başlangıcını işaretlemek için bir sınırlayıcı olarak kullanılır.

Bununla birlikte, iletilmesi gereken orijinal mesajda flag byte içerirse byte ve bit doldurma işlemi yapılır.

BAYT DOLDURMA MEKANİZMASI

Bayrak baytının modeli mesaj baytında mevcutsa, alıcının bu modeli çerçevenin sonu olarak görmemesini sağlayacak bir strateji olmalıdır. Karakter odaklı protokolde benimsenen mekanizma bayt doldurmadır.

Bayt doldurmada, kaçış karakteri (ESC) adı verilen özel bir bayt, mesajdaki her bayttan önce, bayrak baytı ile aynı desene doldurulur. ESC dizisi mesaj baytında bulunursa, bundan önce başka bir ESC baytı doldurulur.



BİT DOLDURMA (STUFFED BITS) MEKANİZMASI

İletilecek mesaja bir veya daha fazla bilgi olmayan bit eklemek, senkronizasyon amacıyla mesaj dizisini kırmak için kullanılır. Aynı zamanda bit odaklı çerçeveleme olarak da bilinir. Burada 5 ardışık 1 bitten sonra 0 bit doldurulur. Yani ardı ardına gelen 5 bitten sonra ekstra bit eklenir

(a) 011011111111111111110010

(b) 011011111011111011111010010

Stuffed bits

(c) 0110111111111111111111110010

a- Orijinal veriler

b- Hat üzerinde göründükleri
şekliyle veriler

c - Veriler çözüldükten sonra
alıcının belleğinde depolandığı
şekliyle verilir.

ÖRNEK :

The following character encoding is used in a data link protocol:

A: 01000111; B: 11100011; FLAG: 01111110; ESC: 11100000

Show the bit sequence transmitted (in binary) for the four-character frame:

A B ESC FLAG when each of the following framing methods are used:

HAM VERİ : 01000111 11100011 11100000 01111110

(a) Character count

(b) Flag bytes with byte stuffing.

(c) Starting and ending flag bytes, with bit stuffing.

ANS:

a) 00000101 01000111 11100011 11100000 01111110

b) 01111110 01000111 11100011 11100000 11100000 11100000 01111110 01111110

c) 01111110 01000111 11100011 11100000 01111110 01111110

ÖRNEK :

GIVEN THE OUTPUT AFTER BYTE-STUFFING:

FLAG A B ESC ESC C ESC ESC ESC FLAG

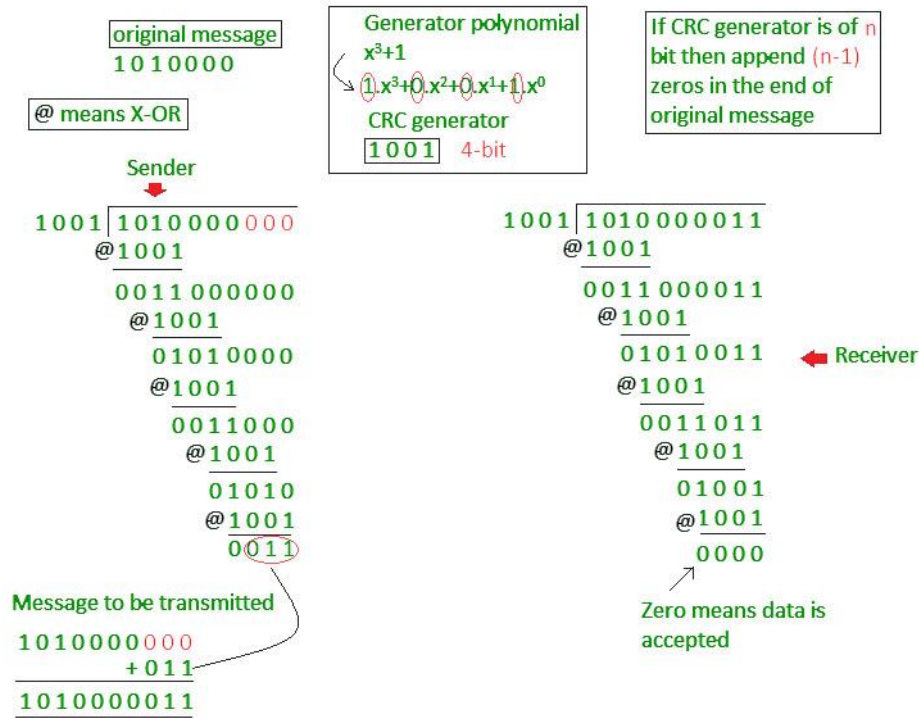
ESC FLAG D FLAG. WHAT IS THE ORIGINAL DATA?

ANS:

A B ESC C ESC FLAG FLAG D

DÖNGÜSEL ARTIKLIK KONTROLÜ (CRC)

Bir hata tespit çeşitidir. Bize bir data ve generator polinom verilir. Dataya generator polinomundaki en yüksek derece(r) kadar sağına 0 eklenir



GÖNDEREN CİHAZ İŞLEMLERİ

Data ve crc generator biti exorlanır en son elimizde kalan sonuç **remainder** olur

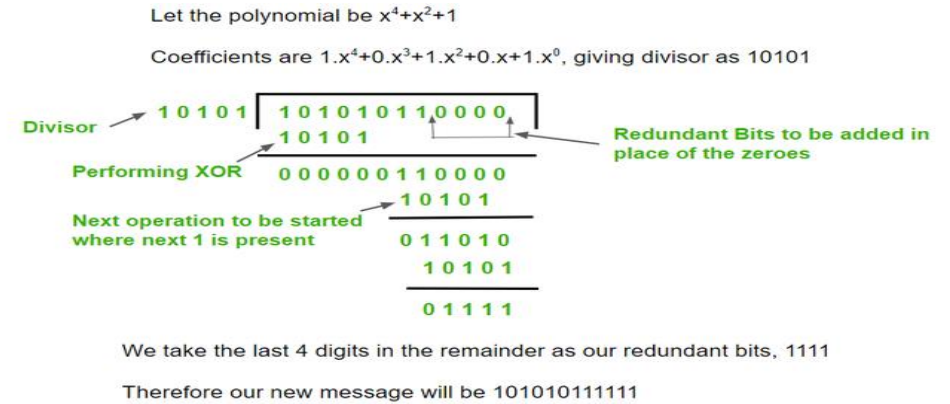
Codeword = data + remainder biti(r kadar sağdan itibaren)

ALICI CİHAZ İŞLEMLERİ

Codeword ve crc generator biti exorlanır en son elimizde kalan sonuç **0** olursa hatasız bit iletilmiş olur

ÖRNEK 2

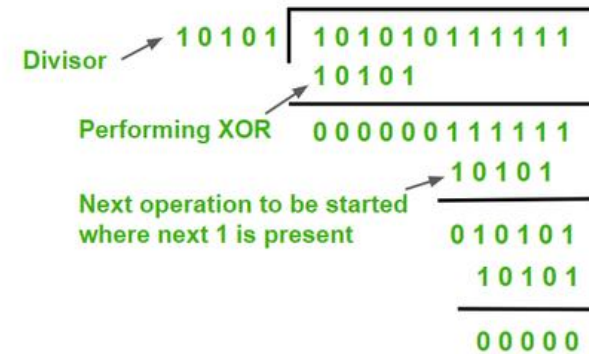
GÖNDEREN CİHAZ İŞLEMLERİ



ALICI CİHAZ İŞLEMLERİ

The polynomial at the receiver side will also be x^4+x^2+1

Coefficients are $1.x^4+0.x^3+1.x^2+0.x+1.x^0$, giving divisor as 10101



Remainder is 0 → No error in data → Accepted by receiver

HAMMING CODE

Tek Hata Düzeltme için Hamming Kodu :

- Adım 1 – Yedekli bit sayısının hesaplanması.
- Adım 2 – Yedekli bitlerin konumlandırılması.
- Adım 3 – Her bir yedek bitin değerinin hesaplanması.

Yedekli bit sayısının hesaplanması : $2^r \geq m + r + 1$

M: mesajdaki veri biti sayısı

R : yedek bit(eklenecek bit sayısı)sayı

Örnek : 11001010 mesajı için iletilen kalıp(odd parity kullanıldığını varsayalım)

1) YEDEKLİ BİT SAYISININ HESAPLANMASI

M:8 , r: ? $2^r \geq 8+r+1$ r :4 olmalıdır (eklenecek bit sayısı)

2) YEDEKLİ BİTLERİN KONUMLANDIRILMASI:

Yeni oluşturacağımız kalıp n veri uzunluğunda olmalıdır (n: m + r)

m : 8 , r : 4 için n : 12 dir ----- orijinal data : 11001010

$m+r+1 \leq 2^r$, m=8
 $8+r+1 \leq 2^r$, r=4
 $n=m+r+1=8+4+1=13$
1 2 3 4 5 6 7 8 9 10 11 12
 2^0 2^1 2^2 2^3
? ? 1 ? 1 0 0 ? 1 0 1 0

$3=2^0+2^1=1+2$
 $5=2^0+2^2=1+4$
 $6=2^1+2^2=2+4$
 $7=2^0+2^1+2^2=1+2+4$
 $9=2^0+2^3=1+8$
 $10=2^1+2^3=2+8$
 $11=2^0+2^1+2^3=1+2+8$
 $12=2^2+2^3=4+8$

HER BİR YEDEK BİTİN DEĞERİNİN HESAPLANMASI

C1 : 2^0 : 3, 5, 7, 9, 11 konumlarındaki veriler için kullanılır, dolayısıyla değeri 11011 1 exorladığımızda 1'dir.

C2 : 2^1 : 3, 6, 7, 10, 11 konumlarındaki veriler için kullanılır, dolayısıyla değeri 10001 1 nedeniyle 1'dir.

C3 : 2^2 : 5, 6, 7, 12 konumlarındaki veriler için kullanılır, dolayısıyla değeri 1000 0 olduğundan 0 olur

C4 : 2^3 : 9, 10, 11, 12 konumlarındaki veriler için kullanılır, dolayısıyla değeri 1010 1 nedeniyle 1'dir.

Yani 11001010 mesajı için iletilen bit modeli 1 1 1 0 1 0 0 1 1 0 1 0'dır

NYQUIST TEOREM

Maximum data rate = $2H \log_2 V$ birimi BPS dir.

H: bant genişliği

V:seviyeli sinyal sayısı

ÖRNEK :

TELEVİZYON KANALLARI 6 MHZ GENİŞLİĞİNDEDİR. DÖRT SEVİYELİ DİJİTAL SİNYALLER KULLANILIRSA KAÇ BİT/SN GÖNDERİLEBİLİR? GÜRÜLTÜSÜZ BİR KANAL VARSAYALIM.

Maksimum veri hızı = $2 * H * \log_2 V$ bit/sn = $2 * 6\text{MHz} * \log_2 4$ bit/sn = 24 Mbps

SİNYAL GÜRÜLTÜ ORANI

$\text{SNR}(\text{db}) = (S/N)_{\text{db}} = 10 \log_{10} (S/N)$

- Sinyalin gürültüye oranı yüksekse iyidir

ŞHANNON TEOREMİ

Kanal kapasitesi = $H \log_2 (1 + (S/N))$

S : ortalama alınan sinyal gücü

N : ortalama gürültü gücü

ÖRNEK :

150-KHz'lik bir hatta bir T1 taşıyıcı koymak için sinyal-gürültü oranı nedir?(T1'in veri hızı 1.544 Mbps'dir.)

- Shannon teoremini kullanarak: $H * \log_2 (1+S/N) = 1,544 * 10^6$
- $150.000 \log_2 (1+S/N) = 1.544 * 10^6$
- $1+S/N = 2^{10}$
- $S/N = 2^{10} - 1$
- $10 * \log S/N = 10 * \log_{10} (2^{10} - 1) = 10 * \log_{10} (1023) \approx 10 * \log_{10} (10^3) \approx 10 * 3 \approx 30 \text{ dB}$

İLETİM GECİKMESİ

Basitçe, tüm paketleri göndermek için geçen süreyi ifade eder veya tüm veri bitlerini iletişim ortamına emmek için geçen süreyi belirtir. Aktarma gecikme, paketin uzunluğuna ve ağın bant genişliğine bağlıdır.

İletim gecikmesi = veri boyutu / bant genişliği = (L / B) saniye

ÖRNEK :

AN IMAGE IS 1600 × 1200 PIXELS WITH 3 BYTES/PIXEL. ASSUME THE IMAGE IS UNCOMPRESSED. HOW LONG DOES IT TAKE TO TRANSMIT IT OVER A 56-KBPS MODEM CHANNEL? OVER A 1-MBPS CABLE MODEM? OVER A 10- MBPS ETHERNET? OVER 100-MBPS ETHERNET? OVER GIGABİT ETHERNET?

Görüntü $1600 \times 1200 \times 3$ bayt veya 5.760.000 bayttır. Bu 46.080.000 bittir.

56.000 bit/sn'de yaklaşık 822.857 saniye sürer.

1.000.000 bit/sn hızında, 46.080 saniye sürer.

10.000.000 bit/sn'de 4,608 saniye sürer.

100.000.000 bit/sn'de yaklaşık 0,461 saniye sürer.

1.000.000.000 bit/sn'de yaklaşık 46 saniye sürer

YAYILMA GECİKMESİ-DELAY İN RESPONS

Yayılma gecikmesi, bitlerin bağlantı üzerinden kaynaktan hedefe gitmesi için geçen süreyi ifade eder. Uzaklık ve yayılma hızı, yayılma gecikmesini etkileyen faktörlerdir.

Yayılma gecikmesi = mesafe / iletim hızı = d / s

ÖRNEK :

A CLIENT-SERVER SYSTEM USES A SATELLITE NETWORK, WITH THE SATELLITE AT A HEIGHT OF 40,000 KM. WHAT IS THE BEST-CASE DELAY IN RESPONSE TO A REQUEST?

İsteğin yukarı ve aşağı gitmesi, yanıtın da yukarı ve aşağı gitmesi gerekir.

Böylece kat edilen toplam yol uzunluğu 160.000 km'dir. Işığın havadaki hızı

ve vakum 300.000 km/sn olduğundan yayılma gecikmesi tek başına

$160.000/300.000$ saniye veya yaklaşık 533 msn

ÖRNEK :

A CLIENT-SERVER SYSTEM USES A SATELLITE NETWORK, WITH THE SATELLITE AT A HEIGHT OF 15,000 KM. WHAT IS THE BEST-CASE DELAY IN RESPONSE TO A REQUEST?

İsteğin yukarı ve aşağı gitmesi gerekir, yanıtın da yukarı ve aşağı gitmesi gerekir. Toplam

Böylece kat edilen yol uzunluğu 60.000 km'dir. Işığın havadaki ve boşluktaki hızı 300.000

km/sn, dolayısıyla tek başına yayılma gecikmesi $60.000/300.000=0,2$ saniye veya yaklaşık 200 msn'dir.