

Veri Bağlantı Kontrol Protokolleri

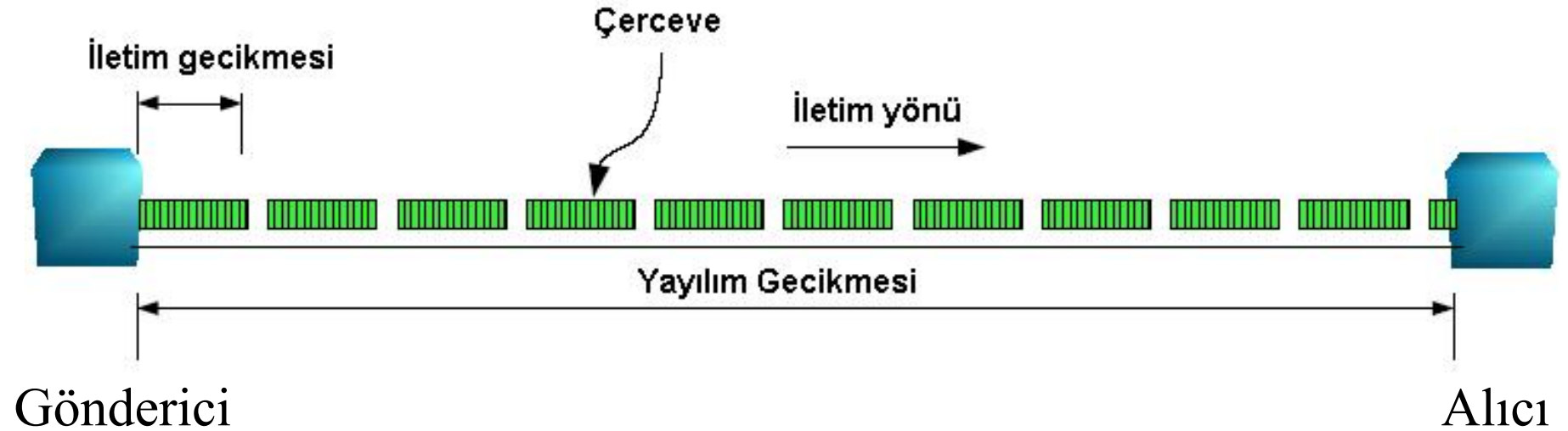
Şimdiye kadar işaretlerin iletim hatları üzerinde gönderilmesi ile ilgilendik. Şimdi ise verilerin bir veri haberleşme bağlantısı üzerinden iletilmesi ile ilgileneceğiz

Veri Bağlantı Katmanı – Temel Fonksiyonları

- Çerçeve eşzamanlaması
- Akış kontrolü – vericiden alıcıya veri akışını düzenleme
- Hata kontrolü – hataları bulup düzeltme
- Adresleme –çok noktalı bir bağlantıda, iletişimdeki iki aygıtın tanımlanması
- Kontrol ve Verinin aynı bağlantıda olması
- Bağlantı yönetimi – bağlantının başlatılması, sürdürülmesi ve sonlandırılması gibi, veri değişim yönetimi işlemleri

Akış Kontrolü

- Veri gönderen sistemin, alan sistemin alma hızına ayak uydurması gereklidir.
 - tampon taşmasını engelleme
- Terminoloji
 - Çerçeve
 - veri bloğunu oluşturan bitlerin toplamı
 - İletim zamanı
 - Göndericinin bir çerçevenin tamamını iletim ortamına göndermesi için gerekli süre (çerçeve uzunluğu ile değişir)
 - Yayılım zamanı
 - Bir bitin iletim ortamında göndericiden alıcıya ulaşma süresi
 - Bit uzunluğu
 - Bir bağlantı hattının tamamını kaplayacak bit sayısı $B = R d / V$
 - R: veri hızı, d: hattın boyu m, V :yayılım hızı m/s



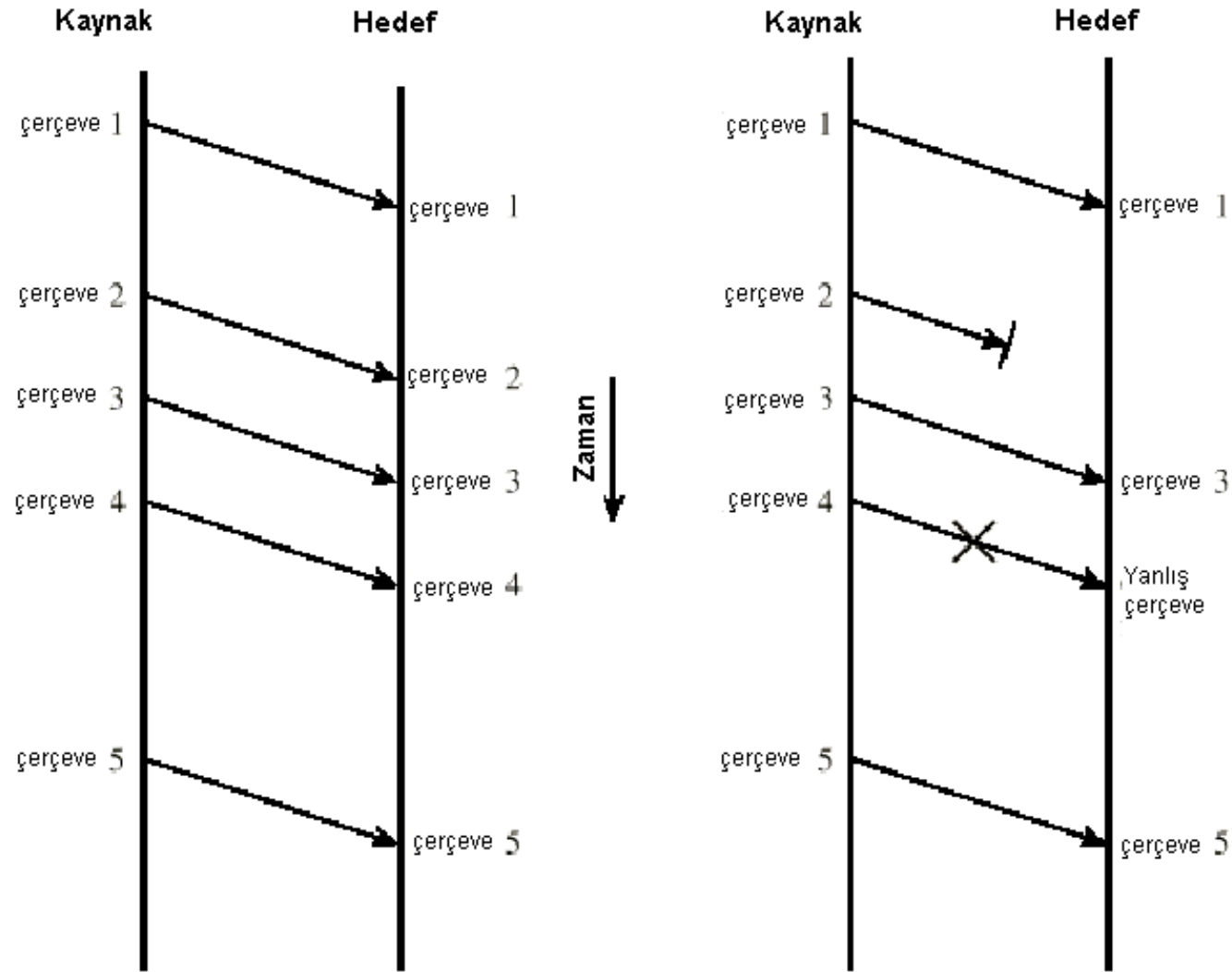
Transmisyon hattı boyunca, yayılım zamanının , iletim zamanına oranı

$a = (d/V)/(L/R) = B/L$ dır. (herhangi bir anda , gönderilebilecek maksimum çerçeve sayısı olarak da yorumlanabilir) Yukarıdaki örnekte 10.2 dir

B: bit uzunluğu, L:çerçevedeki bit sayısı

- Eğer R sabit ise a, R ile artacaktır
- Eğer R sabit ise a d ile artacaktır

Çerçeve İletim Modeli

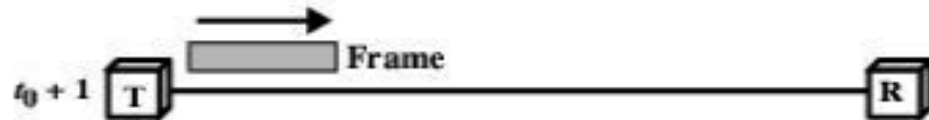


(a) Hatasız iletim

(b) Kayıplı ve hatalı iletim

Dur ve Bekle akış kontrolü (Stop & Wait)

- En basit akış kontrol protokolüdür
- Temel davranış
 - Kaynak çerçeveyi gönderir
 - Hedef çerçeveyi alır ve bir bilgilendirme ve onaylama mesajı gönderir. (Acknowledge)
 - Kaynak diğer çerçeveyi göndermeden önce ACK bekler
 - Kaynak ACK'yı göndermeyerek akışı durdurabilir
 - Sadece geniş çerçeveler ve/veya düşük veri hızları için verimlidir.



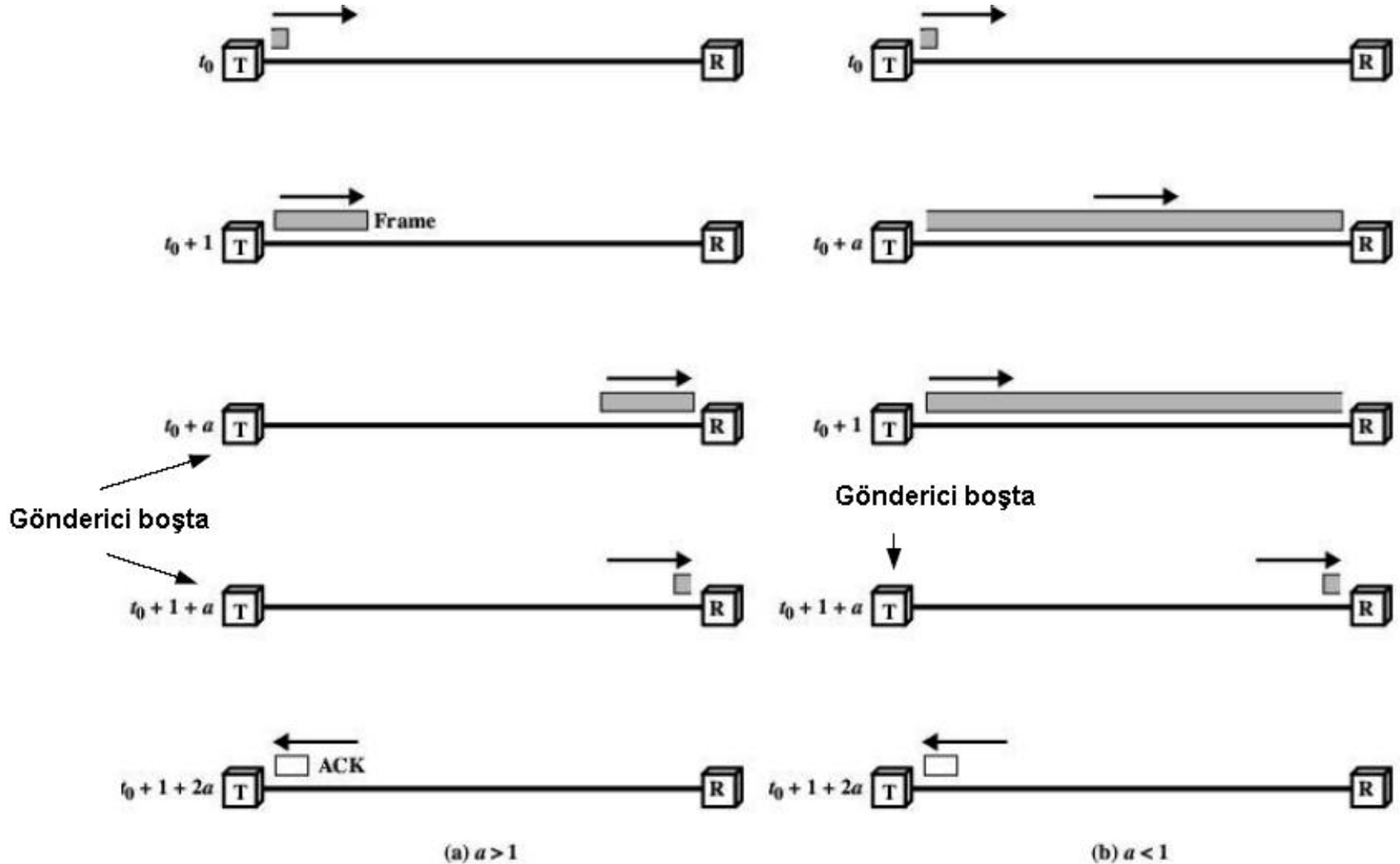
İletim süresi
Normalize edilerek
1 olarak alınsın

a : yayılım süresi
(iletim süresi 1
iken)

Veri Yapısını Küçük Parçalara Ayırma (Fragmentation)

- Geniş veri blokları küçük çerçevelere bölünebilir
 - Sınırlı arabellek dolayısıyla buna ihtiyaç duyulabilir
 - Hatalar daha hızlı algılanır
 - Hata varsa, küçük veri çerçevelerinin yeniden gönderilmesi daha hızlı ve verimli olacaktır
 - Bir istasyonun uzun süreler boyunca ortamı meşgul etmesi engellenir
- Dur ve bekle akış protokolu bu durumda hattı verimli kullanmaz.

Dur ve Bekle, Hat kullanımı



İletim süresi = 1, yayılım süresi = a

Örnek: İki yer istasyonu uydu aracılığıyla haberleşsin.

Jeostasyon bir uydu için mesafe 36000 km dir. Bu durumda iki yer istasyonu arasındaki mesafe yaklaşık 72000 km olacaktır

Hız $R = 1$ Mbps olsun . Buradan,

$B = (10^6 \times 72,000,000)/3 \times 10^8 = 240,000$ yada 240 k bit alıcı ve verici arasındaki mesafe boyunca ortamı dolduracaktır.

Çerçeve uzunluğu 8000 bit ise

$$a = 240000/8000 = 30.$$

İlk bitin ulaşması için geçen süre $72000000/(3 \times 10^8) = 240$ ms olur.

Son bit 8 ms daha gerektirir. Göndericiye gelecek ACK ise $240 + 8 + 240 = 488$ ms. olacaktır

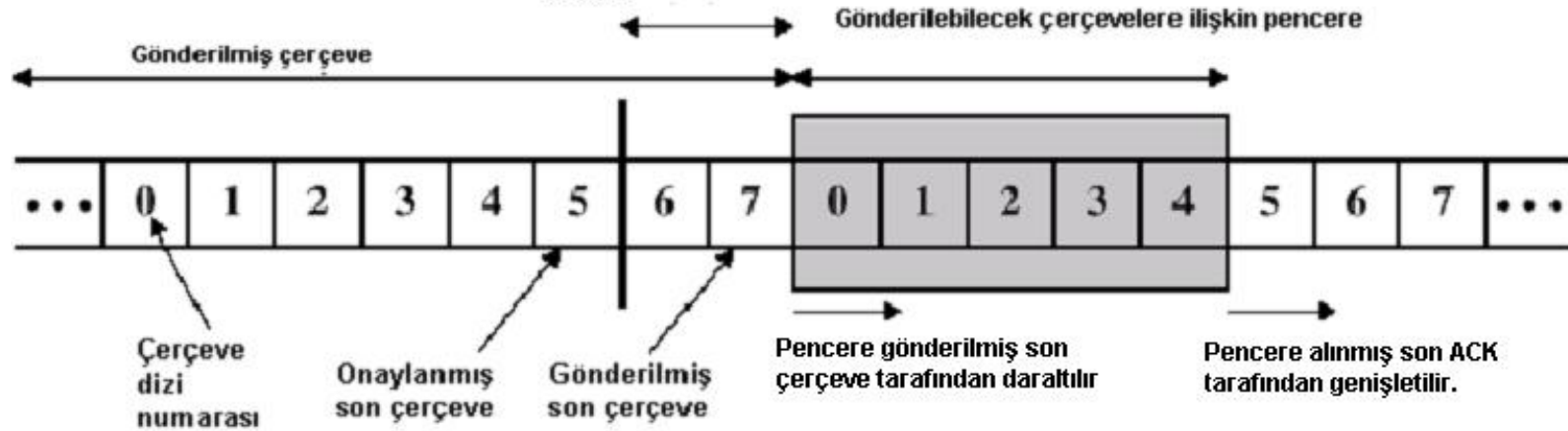
Gerçek iletim süresi 8 ms dir, Ancak, veri gönderip ACK almak için gerekli toplam zaman 488 ms olacaktır. Bu da verimsizlik anlamına gelir

Kayan Pencere Akış Kontrolü (Sliding Windows)

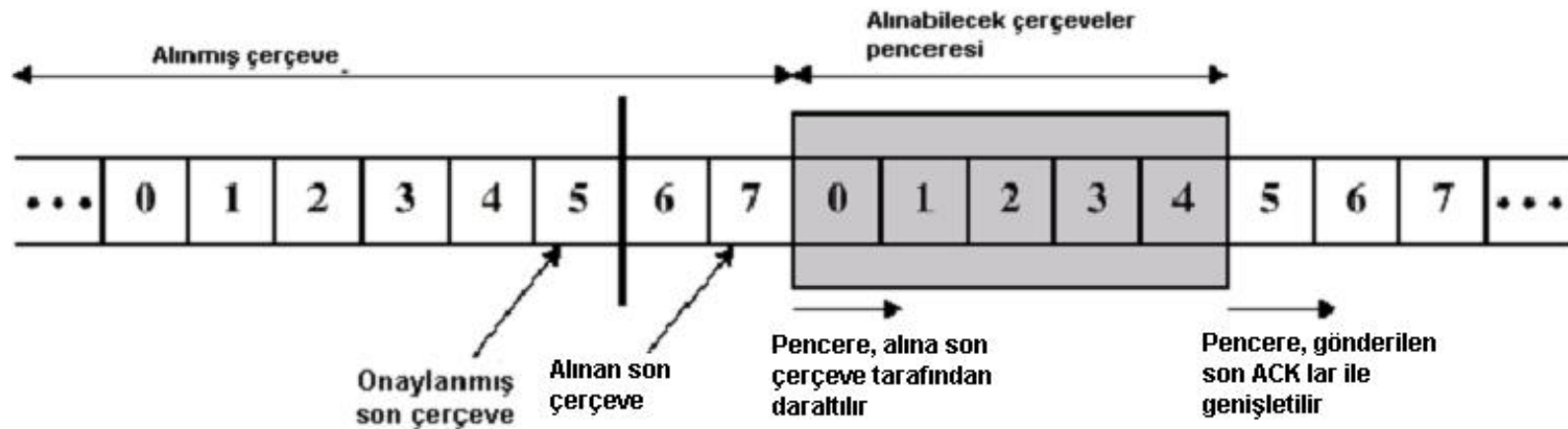
- Birden fazla çerçevenin iletimine izin verir
 - Alıcı W çerçeve uzunluklu tampona sahiptir
 - Verici ACK beklemeksizin W adet çerçeve iletebilir
- Herbir çerçeve bir dizi numarası ile numaralandırılır
 - ACK bir sonraki beklenen çerçevenin numarasını içerir
 - Dizi numaraları, çerçeve içinde (k) uzunluklu bir alan kaplar
 - Çerçeveler $\text{mod } 2^k$ ya göre numaralandırılır.
- Gönderici ve alıcı, gönderecekleri ve alacakları paketlerin listesini tutarlar
- Herbir liste bir *pencere* olarak düşünülebilir.
- Alıcıda ve vericideki maksimum pencere boyutu W dir.

Kayan Pencere yapıları

Ack alınana kadar bellekte tutulan çerçeve



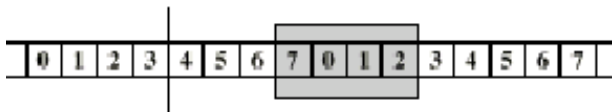
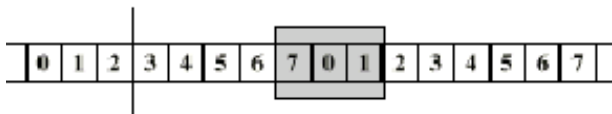
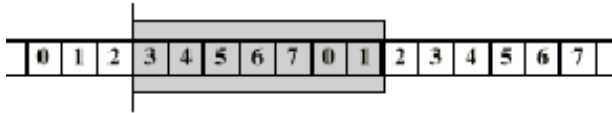
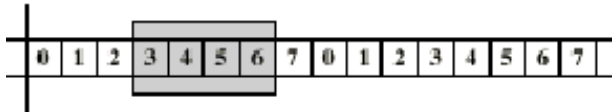
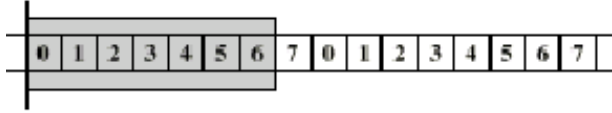
(a) Verici perspektifi



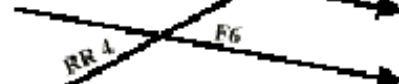
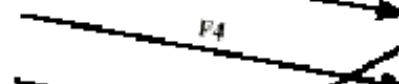
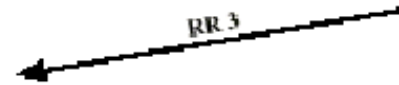
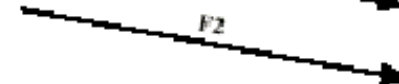
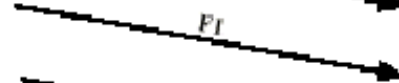
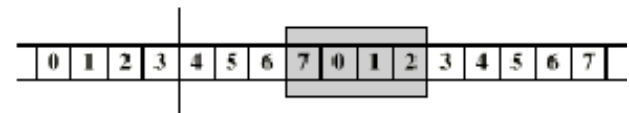
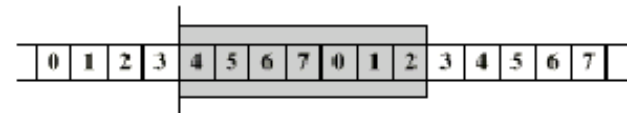
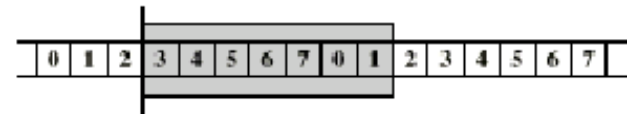
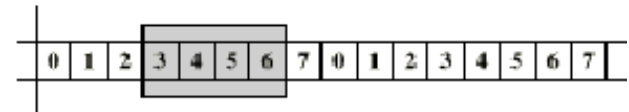
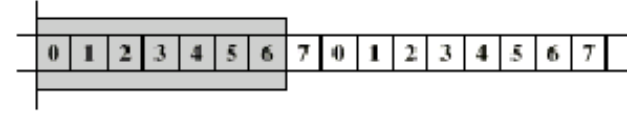
(b) Alıcı perspektifi

Kayan Pencere Örneği

Kaynak Sistem A



Hedef Sistem B



Burada Dizi numarası 3 bir $W=7$ dir

Kayan Pencerelerin getirdiği diğer farklar

- Alıcı, daha fazla veri gelmesini engellemekle birlikte gelen çerçeveleri onaylayabilir. Bunun için RNR kullanılır
- Alıcı işleme devam etmek için normal Ack (RR) kullanır.
- Eğer çift yönlü bir haberleşme söz konusu ise daha etkin bir yöntem kullanılır (Piggyback)
 - Gönderilecek veri ve ACK için, aynı çerçeve içersisinde ayrı dizi numarası alanları tanımlanır. Veri gönderilirken ACK da gönderilir
 - İstasyon hem veri hem ACK gönderecekse, her ikisi birden aynı çerçevede gönderilir.
 - Eğer gönderilecek veri yoksa, bir ACK paketi gönderilir. (RR yada RNR gibi)
 - Eğer veri var fakat ACK yoksa, gönderici son ack numarasını tekrarlar yada ACK geçerli bayrağı (ACK valid flag) kullanır (TCP).
- Kayan pencereler yöntemi, ortamı verilerle doldurulmuş bir boru hattı gibi kullandığından daha etkindir.

Hata Kontrolü

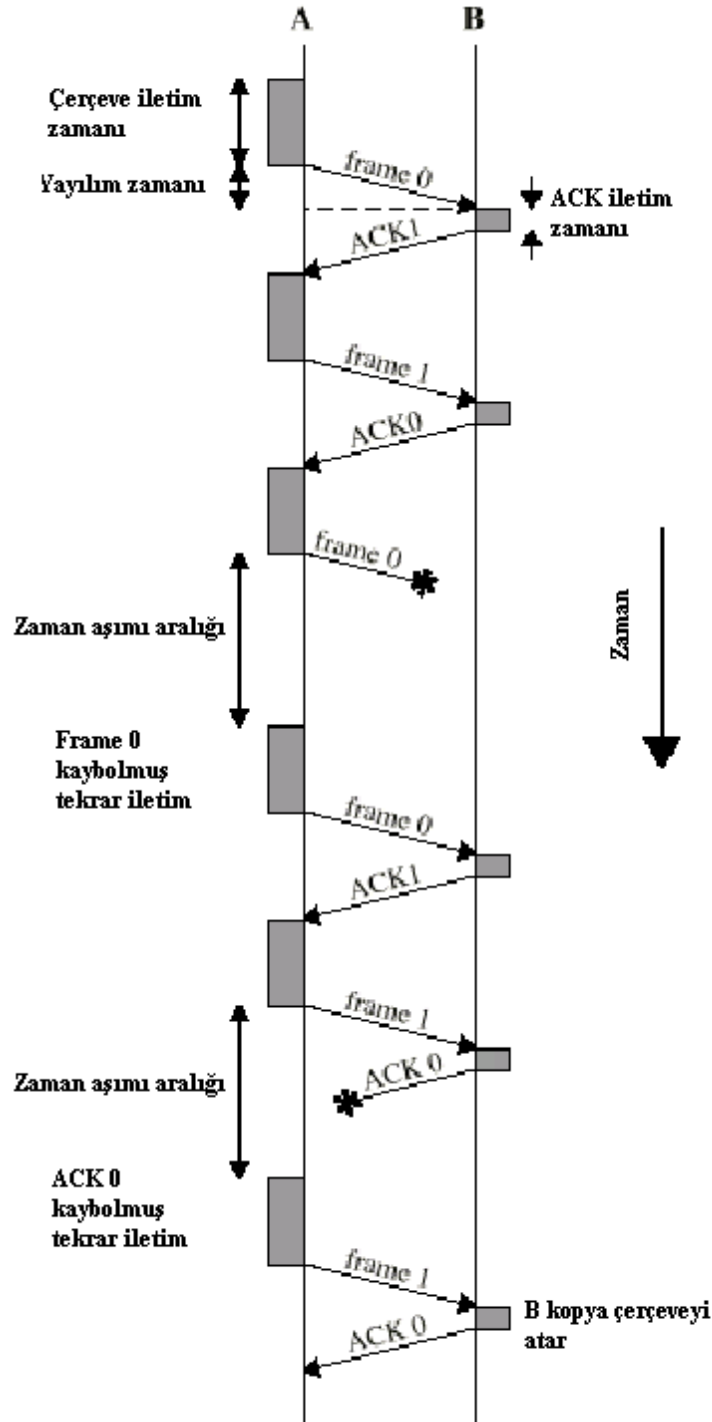
- Veri iletimi esnasında hataların algılanması ve düzeltilmesi (güvenilir bir bağlantı sağlanır)
- Daha önce anlatılan hata algılama ve düzeltme teknikleri bit seviyesindedir ve çok büyük hataları düzeltemezler. Veri link katmanındaki hata yönetimi tüm çerçevenin düzeltilmesini sağlar.
- Neler önemlidir ?
 - Kayıp çerçeveler
 - Bozuk çerçeveler
- Hata kontrol bileşenleri :Otomatik yenileme isteği -Automatic repeat request (ARQ)
 - Hata algılama
 - Kayıp çerçeveler, zamanaşırımları ile (timeout)
 - Bozuk çerçeveler, parite, kontrol toplamları, CRC vs.. ile
 - Pozitif ack: Doğru çerçeveler için gönderilir.
 - Zaman aşımı sonrasında yeniden gönderme.
 - Gönderici, belirli bir sürede ACK algılamaz ise, paketi yeniden gönderir.
 - Negatif ack ve yeniden gönderme

Otomatik Tekrar İsteği (ARQ)

- ARQ üç değişik şekilde sınıflandırılır
 - Dur ve bekle ARQ (Stop and Wait ARQ)
 - Dur bekle protokolü tabanlıdır
 - N'ye geri dön ARQ (Go- Back –N ARQ)
 - Kayan pencereler protokolü tabanlıdır
 - Seçmeli-reddetme ARQ (Selective – Reject ARQ)
(tekrar gönderileni seçme)
 - daha büyük arabellek gerektirir

Dur ve Bekle ARQ

- Kaynak tek çerçeve gönderir
- ACK için bekler
- Alınan çerçeve bozulmuşsa onu atar
 - Verici bir zaman aşımı süresi içerir
 - ACK zamanında gelmezse, veriyi tekrar gönderir
- ACK bozulmuşsa verici onu tanımlayamaz
 - Verici, veriyi tekrar gönderir
 - Alıcı iki kopya alırsa tekrarlama işlemi nasıl çözülür?
 - Çerçeveler gönderici tarafından alternatif olarak 0 ve 1 olarak numaralandırılır.
 - Alıcı pozitif ACK lar için ACK0 ve ACK1 kullanır.
 - Alıcı tekrarlananı tanır ve siler.



Dur ve Bekle ARQ

Buradaki numaralandırma tıpkı kayan pencerelerdeki gibidir.

ACK0 1,i doğru aldım bana 0 'ı gönder anlamındadır.

- Basittir
- Verimli değildir.

Go-Back-N ARQ

- Kayan pencereler protokolüne dayanır.
- Eğer hata yoksa, ACK (RR) beklenen bir sonraki pencereyi işaret eder.
- Giden çerçevelerin numaralarını kontrol etmek için pencere kullanır.
- Eğer hata varsa, reddetme komutu gönderir (REJ)
 - Bu çerçeveyi ve daha sonraki tüm çerçeveleri, çerçeve doru alınıncaya kadar iptal eder.
 - Verici geri dönmeli, bozuk çerçeveyi ve daha sonraki tüm çerçeveleri yeniden göndermelidir

Go-Back-N ARQ Örneği – Bozuk/kayıp çerçeve

- İki alt durum:
 - Durum 1
 - Gönderici i çerçevesini gönderir.
 - Çerçeve iletim esnasında bozulur yada kaybol
 - Alıcı bozuk alınan çerçeveyi siler.
 - Gönderici i+1. Çerçeveyi gönderir.
 - Alıcı i+1. Çerçeveyi alınca i olmadığı için REJ i gönderir.
 - Gönderici i. Ve daha sonraki tüm çerçeveleri yeniden gönderir.
 - Durum 2
 - i. Çerçeve kaybolur yada bozulur, gönderici daha fazla çerçeve göndermez.
 - Alıcı hiçbirşey almaz. ACK (RR) yada REJ göndermez.
 - Göndericide zaman aşımı oluştuğunda Özel bir P bitini (POLL bit) 1 yaparak RR gönderir
 - P=1 in anlamı alıcıya bunun ack' lanması gerekli olan bir komut olduğudur.
 - Alıcı RR i ile cevaplar ve gönderici çerçeveyi yeniden gönderir.

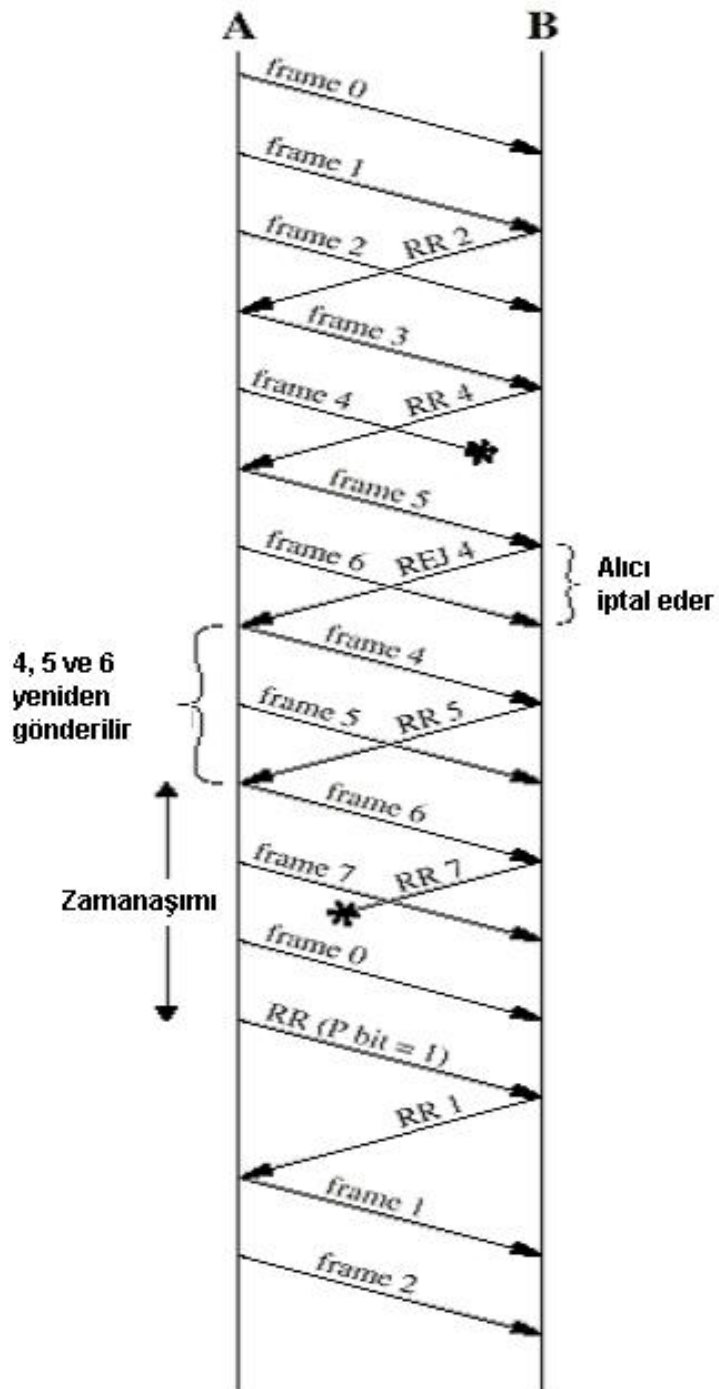
Hatalı/Kayıp RR

- Durum 1
 - Alıcı I . Çerçeveyi alır ve RR ($i + 1$) gönderir.
 - RR iletim sırasında kaybolur
 - Ack lar kümülatif olduğu için , daha sonraki çerçeve için gönderilecek RR alındığında, gönderici zamanaşımına uğramamış ise herhangi bir hata oluşmaz
 - Hata kendiliğinden çözülmüş olur.
- Durum 2
 - Eğer gönderici zaman aşımına uğrarsa, Alıcıya $P=1$ yaparak RR gönderir.
 - Gönderici diğer bir zamanlayıcıyı ayarlar (P-bit zamanlayıcısı)
 - Eğer alıcı cevap veremez yada cevap bozulur yada kaybolursa p bit zaman aşımı oluşur.
 - Bu durumda, gönderici, Alıcıya $P=1$ yaparak tekrar RR gönderir ve p bit zamanlayıcısını sıfırlar.
 - Bu işlem belirli sayıda tekrarlanır. Daha sonra reset işlemi gerçekleştirilir.

Hatalı/kayıp REJ

- Kayıp veri çerçevesi ile aynı işlemler yapılır
 - Eğer gönderici daha fazla çerçeve gönderirse , REJ- i tekrarlanır
 - Eğer gönderici bir şey göndermez ise zaman aşımı olur
 - $P=1$ yaparak RR gönderir
 - Alıcı RR- i ile yanıtlar
 - Gönderici i . ve daha sonraki çerçeveleri gönderir.

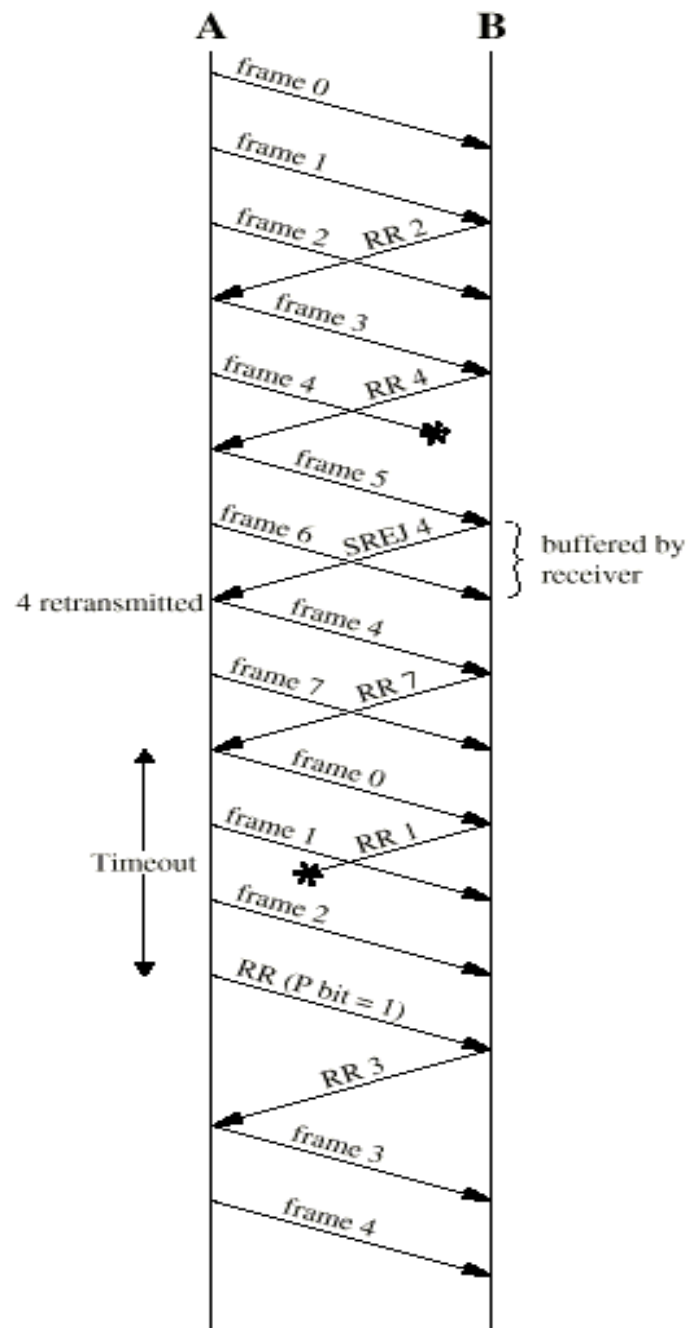
Go-Back-N ARQ örneği



Seçmeli Reddetme (Selective Reject)

- Seçmeli yeniden iletim olarak da isimlendirilir.
- Yanlızca reddedilen çerçeveler gönderilir. Bunun için SREJ kullanılır.
- Daha sonraki çerçeveler alıcı tarafından kabul edilir ve tamponlanır
- Yeninden göndermeyi azaltır (daha verimlidir.)
- dezavantajları
 - Alıcı, çerçeveyi beklemek için daha büyük tamponlar kullanır.
 - Alıcı ve vericide, ardışıl çerçeveleri takip edebilmek için daha karmaşık yapılara ihtiyaç vardır

Seçmeli reddetme örneği



Yüksek Seviyeli Veri Bağlantı Kontrol Protokolü (High Level Data Link Control- HDLC)

- En önemli veri bağlantı kontrol protokolüdür
 - Geniş kullanım alanına sahiptir
 - Diğer bazı veri bağlantı kontrol protokollerine temel oluşturmuştur (ITU.T LAPB, LAPD, LAPF)
 - Açık bir protokoldür
- ISO tarafından standartlaştırılmıştır.
 - ISO 3009 ve ISO 4335
- IBM in, BISYNC, SDLC protokolleri daha önce tanımlanmasına rağmen, özel yapılar olduklarından yaygınlaşamadılar.
- Temel Özellikler
 - Üç istasyon türü
 - İki hat yapılandırması
 - Üç veri transfer modu

HDLC İstasyon türleri

- Birincil (Master) istasyon
 - Bağlantının işletilmesini kontrol eder
 - Gönderilen çerçeveler komut olarak isimlendirilir
 - Herbir ikincil istasyon için ayrı bir mantıksal bağlantı oluşturur.
- İkincil (slave) istasyon
 - Birincil istasyonun kontrolü altındadır
 - Gönderilen çerçeveler yanıt olarak isimlendirilir.
- Birleşik istasyon
 - Hem komut hem de yanıt gönderebilir.

HDLC Bağlantı yapıları

- Dengesiz
 - Bir adet birincil, bir yada daha fazla ikincil istasyon
 - Full duplex yada half duplex iletim desteği
- Dengeli
 - İki birleşik istasyon
 - Full duplex yada half duplex iletim desteği

HDLC Aktarım Modları (1)

- Normal yanıt Modu (NRM)
 - Dengesiz yapılandırma
 - İkincile aktarımı, birincil istasyon başlatır.
 - İkincil istasyon yalnızca birincil istasyondan gelen komutlara yanıt verebilir.
 - Çok noktalı bağlantılarda kullanılır
 - Ana bilgisayar genellikle birincil istasyondur.
 - Terminaller ikincil istasyonlardır.

HDLC Transfer Modes (2)

- Asenkron Dengeli Mode (ABM)
 - Dengeli bir yapıdır.
 - İstasyonlardan herhangi biri , izin almaksızın iletişimi başlatabilir.
 - En çok kullanılan tekniktir
 - Tarama yükü yoktur.

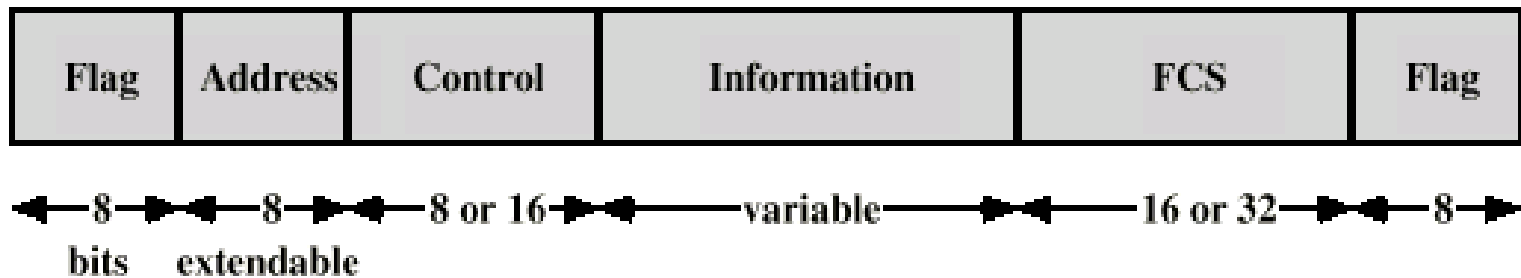
HDLC Transfer Modes (3)

- Asenkron Yanıt Modu (ARM)
 - Dengesiz yapılandırmadır
 - İkincil istasyon, birincilden izin almadan, iletimi başlatabilir.
 - Hattın sorumlu olan birincidir.
 - Nadiren Kullanılır

HDLC Çerçeve Yapısı

- Senkron iletişim kullanır
- Tüm iletim çerçeveler halinde yapılır
- Tüm veri ve kontrol bilgisi değişimleri için tek bir çerçeve yapısı kullanılır
 - Kontrol alanlarının yapıları, veri ve komut çerçevelerinde farklı fonksiyonlar ve anlamlar taşır

HDLC çerçeve yapısı



Bayrak (Flag) alanları

- Tek bir bit dizisi kullanarak çerçeveyi her iki uçtan sınırlandırır (01111110)
 - Bir çerçeveyi bitirip diğerini açabilir.
 - Kullanıcı- ağ arayüzünde alıcılar, senkronizasyon için bayrak dizisini gözlemlerler
- 01111110 bit dizisini içeren bir veri olması durumunda, hatayı engellemek için araya bit sokma (bit stuffing) yöntemi kullanılır. Bu yöntemle herhangi bir bit dizisi veri olarak gönderilebilir (veri şeffaflığı).
 - Her 5 adet 1 den sonra 0 araya sokulur
 - Eğer alıcı 5 tane peşi sıra 1 tespit ederse, bir sonraki biti kontrol eder
 - Eğer 0 ise silinir
 - Eğer 1 ise ve 7. Bit 0 ise bayrak olarak kabul edilir.
 - Eğer 6. Ve 7. Bitler 1 ise gönderici, iptal istemi belirtmektedir.

Araya bit ekleme örneği

Orjinal bit dizisi

111111111111011111101111110

Araya bir eklendikten sonra

1111101111101101111101011111010



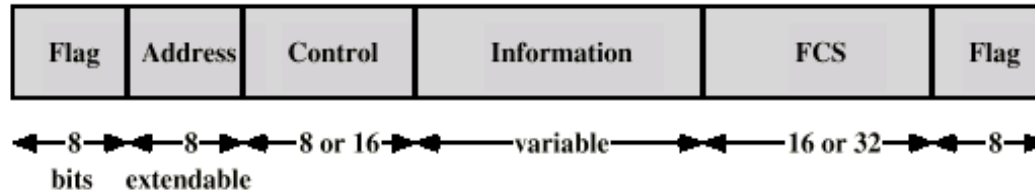
Evrilen bit iki çerçeveyi birleştirebilir.



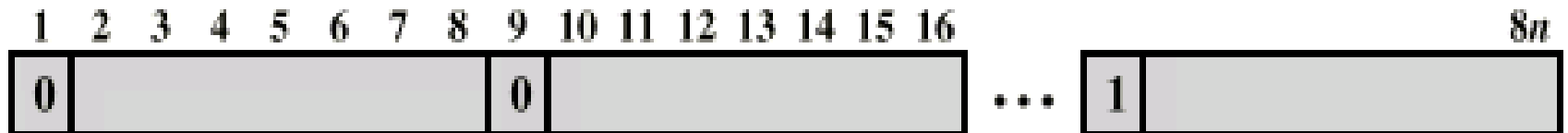
Evrilen bir bit çerçeveyi iki parçaya ayırabilir.

- Hata oluşturabilecek örnekler

HDLC Adres alanı



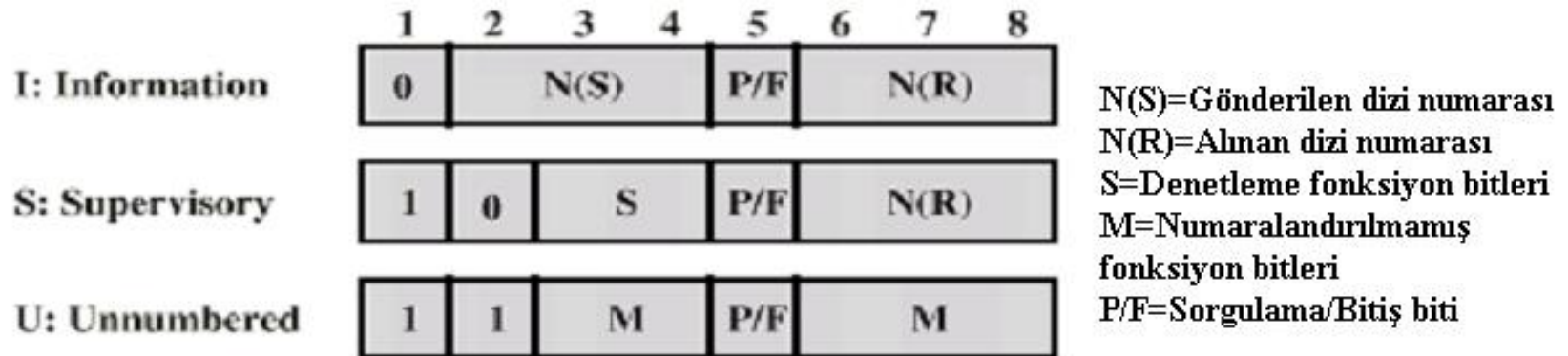
- Çerçeveyi alacak yada gönderecek ikincil istasyonu tanımlar
- Genellikle 8 bit tir
- Noktadan noktaya hatlar için gerekli değildir. Ancak uyumluluk için kullanılır
- 7 bitin katları biçiminde genişletilebilir
 - Her bir oktetin en düşük ağırlıklı biti, onon son oktet (1) olup olmadığını belirtir.
- Tamamiyle 1111111 olan tek oktet lik adres genişyayım adresidir.



HDLC Kontrol Alanı

- Çerçeve tiplerine göre değişir
 - Bilgi (Information) – Kullanıcıya gönderilecek bilgidir (bir üst seviyeye gider)
 - Bilgi çerçevelerinde, hata ve akış kontrolü çift yönlüdür
 - Denetleme (Supervisory) - piggyback kullanılmadığında ARQ kullanılır
 - Numaralandırılmamış (Unnumbered) – Ek bağlantı kontrol fonksiyonları
- Kontrol alanının ilk bir yada iki biti çerçeve tipini belirler.

HDLC Kontrol Alanı Detayları



8 bit kontrol alanı yapısı



16 bit kontrol alanı yapısı

Sorgula/Bitiş biti

- Anlamı içeriğe göre değişir
- Komut Çerçevesi
 - P biti anlamına gelir
 - 1 yapılırsa alıcıdan yanıt istenir
- Yanıt çerçevesi
 - F biti anlamına gelir
 - 1 yapılırsa istekte bulunan komutun yanıtıdır

Bilgi alanı

- Yalnızca bilgi (I) ve bazı numaralandırılmamış (U) çerçevelerde mevcuttur
- Uzunluğu 8 bitin (oktet) katları biçiminde olmalıdır.
- Değişken uzunlukludur. Örneğin LAPF de 15384 byte'lara kadar çıkabilir.

Çerçeve Kontrol dizisi alanı (FCS)

- FCS byraklar hariç çerçevenin geri kalan bitleri üzerinden hesaplanır
- Bağlantı için hata kontrolünü oluşturur
- 16 bit CRC (CRC-CCITT) FCS de standarttır
- Bağlantı kurulumu sırasında, opsiyonel olarak 32 bit CRC (CRC-32) kullanılabilir

| CRC | $C(x)$ |
|-----------|---|
| CRC-8 | $x^8+x^2+x^1+1$ =100000111 |
| CRC-10 | $x^{10}+x^9+x^5+x^4+x^1+1$ |
| CRC-12 | $x^{12}+x^{11}+x^3+x^2+x^1+1$ =1100000001111 |
| CRC-16 | $x^{16}+x^{15}+x^2+1$ =11000000000000101 |
| CRC-CCITT | $x^{16}+x^{12}+x^5+1$ =10001000000100001 |
| CRC-32 | $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$ |

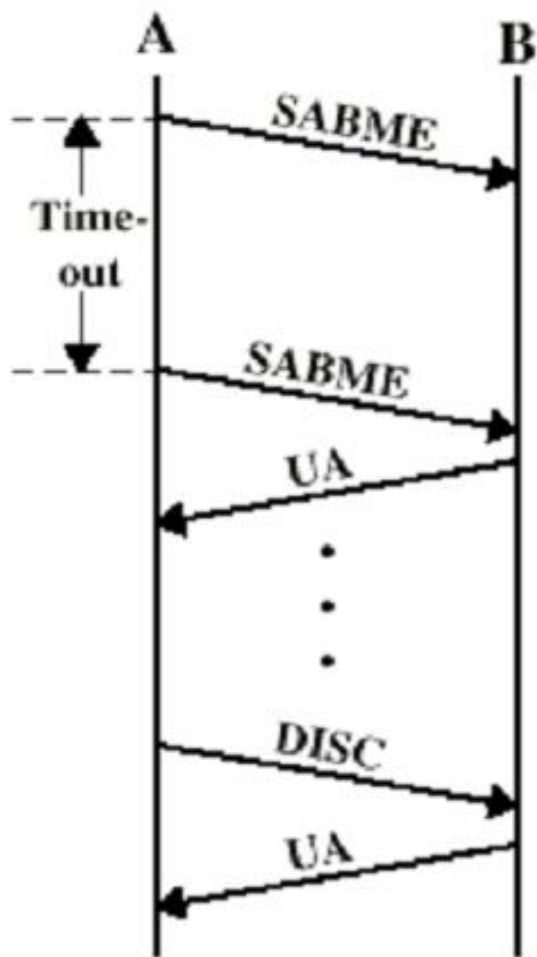
Tipik HDLC Çalışması

- İki istasyon arasında, Bilgi, denetleme ve Numaralandırılmamış çerçevelerin değişimi
- HDLC çalışması üç evereden oluşur
 - İklendirme
 - Bağlantı tipi (NRM, ABM, yada ARM) ve dizi numaralarının uzunlukları belirlenir.
 - Veri aktarımı
 - I ve S çerçevelerinin değişimidir
 - Bağlantı kesme
 - Bağlantıdaki herhangi istasyon tarafından DISC çerçevesi ile sağlanır

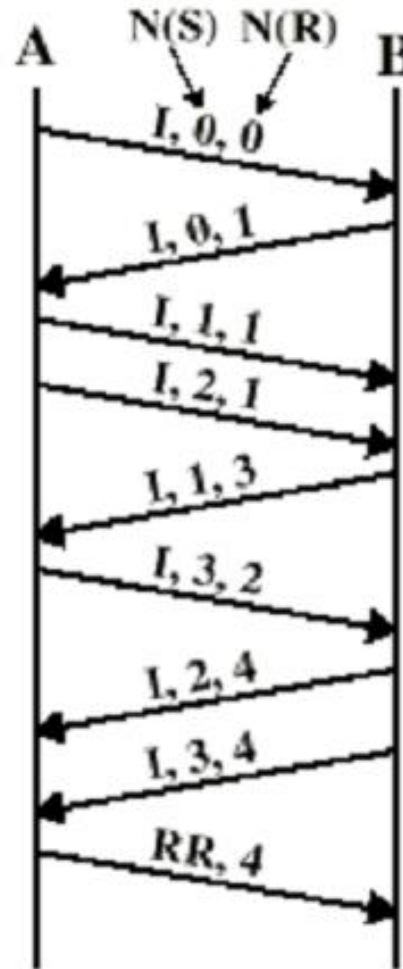
| Name | Command/ Response | Description |
|---|----------------------|--|
| Information (I) | C/R | Exchange user data |
| Supervisory (S) | | |
| Receive ready (RR) | C/R | Positive acknowledgment; ready to receive I-frame |
| Receive not ready (RNR) | C/R | Positive acknowledgment; not ready to receive |
| Reject (REJ) | C/R | Negative acknowledgment; go back N |
| Selective reject (SREJ) | C/R | Negative acknowledgment; selective reject |
| Unnumbered (U) | | |
| Set normal response/extended mode (SNRM/SNRME) | C | Set mode; extended = 7-bit sequence numbers |
| Set asynchronous response/extended mode (SARM/SARME) | C | Set mode; extended = 7-bit sequence numbers |
| Set asynchronous balanced/extended mode (SABM, SABME) | C | Set mode; extended = 7-bit sequence numbers |
| Set initialization mode (SIM) | C | Initialize link control functions in addressed station |
| Disconnect (DISC) | C | Terminate logical link connection |
| Unnumbered Acknowledgment (UA) | R | Acknowledge acceptance of one of the set-mode commands |
| Disconnected mode (DM) | R | Responder is in disconnected mode |
| Request disconnect (RD) | R | Request for DISC command |
| Request initialization mode (RIM) | R | Initialization needed; request for SIM command |
| Unnumbered information (UI) | C/R | Used to exchange control information |
| Unnumbered poll (UP) | C | Used to solicit control information |
| Reset (RSET) | C | Used for recovery; resets N(R), N(S) |
| Exchange identification (XID) | C/R | Used to request/report status |
| Test (TEST) | C/R | Exchange identical information fields for testing |
| Frame reject (FRMR) | R | Report receipt of unacceptable frame |

HDLC Komut ve yanıtları

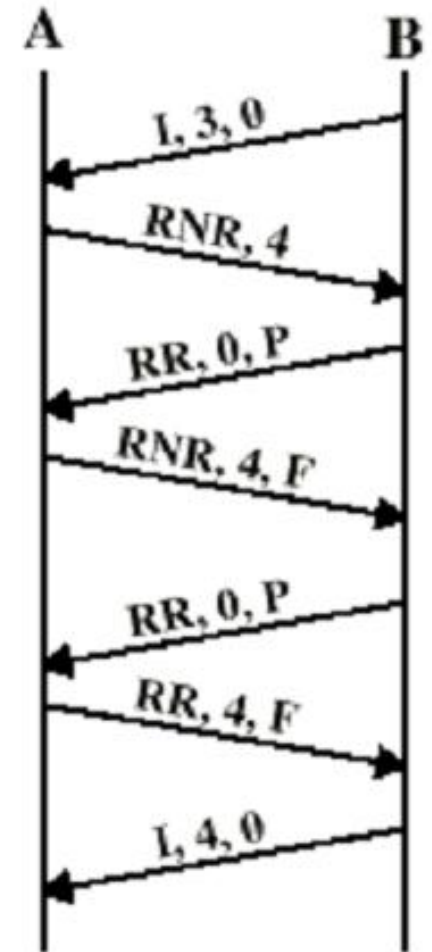
Çalışma Örneği (1)



a) Bağlantı kurma ve sonlandırma

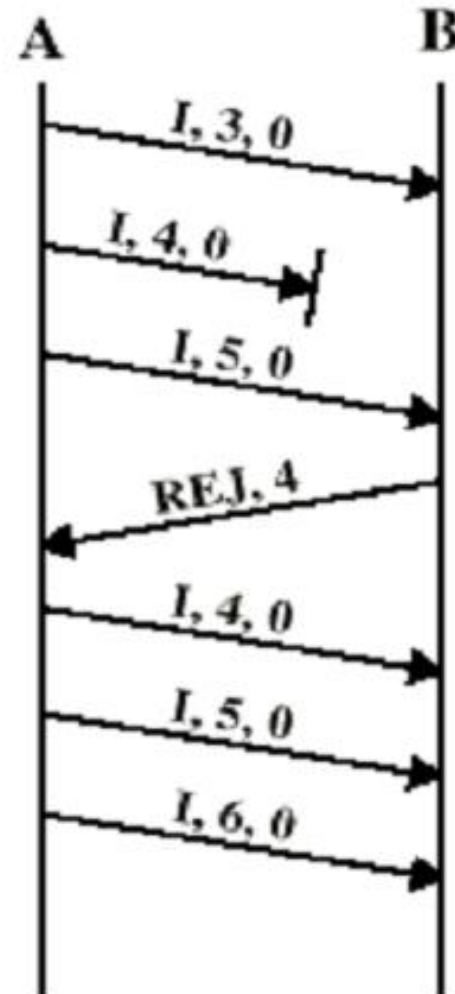


b) iki yönlü veri değişimi

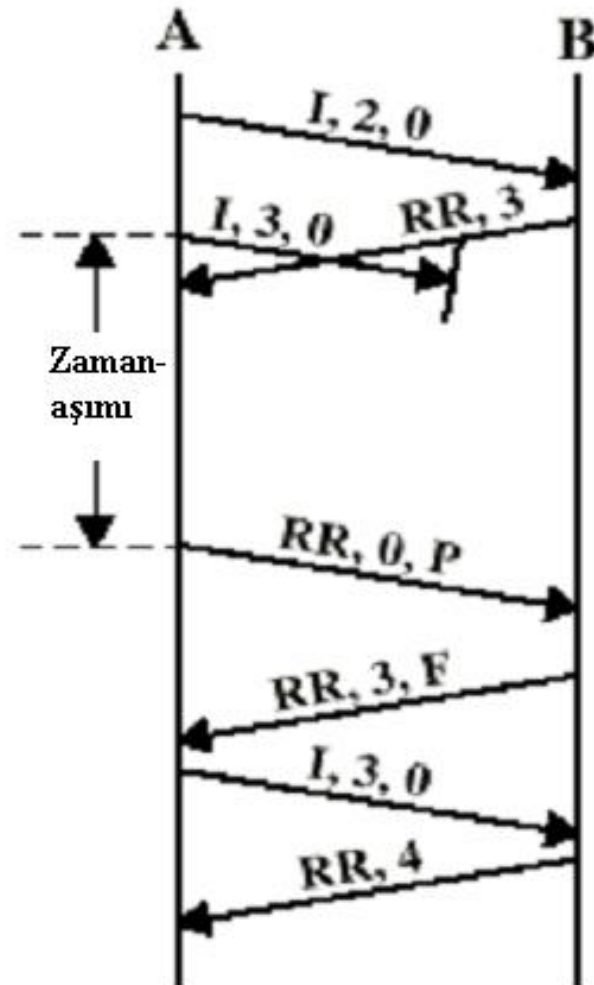


c) meşgul durumu

Çalışma Örneği (2)



d) Reddetme ile kurtama

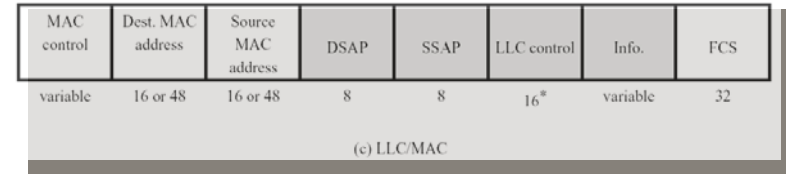


e) Zaman aşımını kurtama

Diğer veri bağlantı protokolleri (LAP-B ve LAP-D)

- Link Access Procedure, Balanced (LAP-B)
 - X.25 (ITU-T) paket anahtarlama standartında tanımlanmıştır
 - HDLC nin alt kümesidir – yalnızca ABM kullanılır
 - Sistem ve paket anahtarlama ağ noktası arasında noktadan noktaya haberleşme sağlar
- Link Access Procedure, D-Channel (LAP-D)
 - ISDN de kullanılır. ITU-T Q.921 de komut ve yanıtlar özetlenmiştir
 - HDLC nin alt kümesidir – yalnızca ABM kullanılır
 - Daima 7-bitlik dizi numaraları kullanılır (3-bit yoktur)
 - 16 bitlik adres alanı iki alt adres içerir
 - Biri aygıt (kullanıcı tarafında birden fazla aygıt olabilir), diğeri kullanıcı (arayüzün kullanıcı tarafındaki olası birde fazla kullanıcı) içindir (bir üst seviye)

Diğer veri bağlantı protokolleri (LLC)



- Logical Link Control (LLC)
 - IEEE 802 de kullanılır; HDLC den farklı bir çerçeve yapısına sahiptir.
 - Veri bağlantısı iki katmana ayrılmıştır (MAC ve LLC)
 - Birincil ve ikincil istasyon yoktur – tüm istasyonlar eşdeğerdir
 - Medium Access Control (MAC) katmanı
 - İki adres içerir: Gönderici ve alıcı
 - Hata Kontrolü (CRC-32)
 - Ortamla ilgili kontrol fonksiyonları içerir
 - Logical Link Control (LLC) layer
 - Hedef ve kaynak hizmet erişim noktaları (DSAP, SSAP)
 - Üç LLC hizmeti: Bağlantı -modu, Bilgilendirmeli bağlantısız mod ve bağlantısız mod

Diğer veri bağlantı protokolleri

Frame Relay

- Yüksek hızlı paket anahtarlama için geliştirilmiştir
- X.25 den türemiş onun yerine kullanılmak üzere geliştirilmiştir
- LAP-F protokolünü kullanır
- LAP-F içinde gerçekte iki protokol vardır
 - kontrol – HDLC ye benzer ancak farklı adresleme kullanır
 - çekirdek - LAP-F kontrol çerçevesi gibidir. Ancak kontrol alanı içermez

- yalnızca ABM kullanılır
- 7-bitlik dizi numaraları
- 16 bit CRC
- 2, 3 yada 4 oktet adres alanı
 - Veri Hattı bağlantı tanıtıcısı DLCI olarak adlandırılır
 - Yerel bazda mantıksal bir bağlantı (virtual circuit) tanımlar

Diğer veri bağlantı protokolleri (ATM)

- Asynchronous Transfer Mode
- Yüksek hızlı ağlarda sürekli kesintisiz veri transferi
- HDLC tabanlıdır değildir. Tamamen yeni bir format tanımlar . "cell"
- Sabit uzunlukludur. 53 oktet (424 bit). Buda donanımsal anahtarlamaı kolaylaştırır.