BSM 471-AĞ GÜVENLİĞİ

Hafta5: Katman 4 Protokolleri ve Çalışma Yapıları

Dr. Öğr. Üyesi Musa BALTA
Bilgisayar Mühendisliği Bölümü
Bilgisayar ve Bilişim Bilimleri Fakültesi

Konu İçeriği

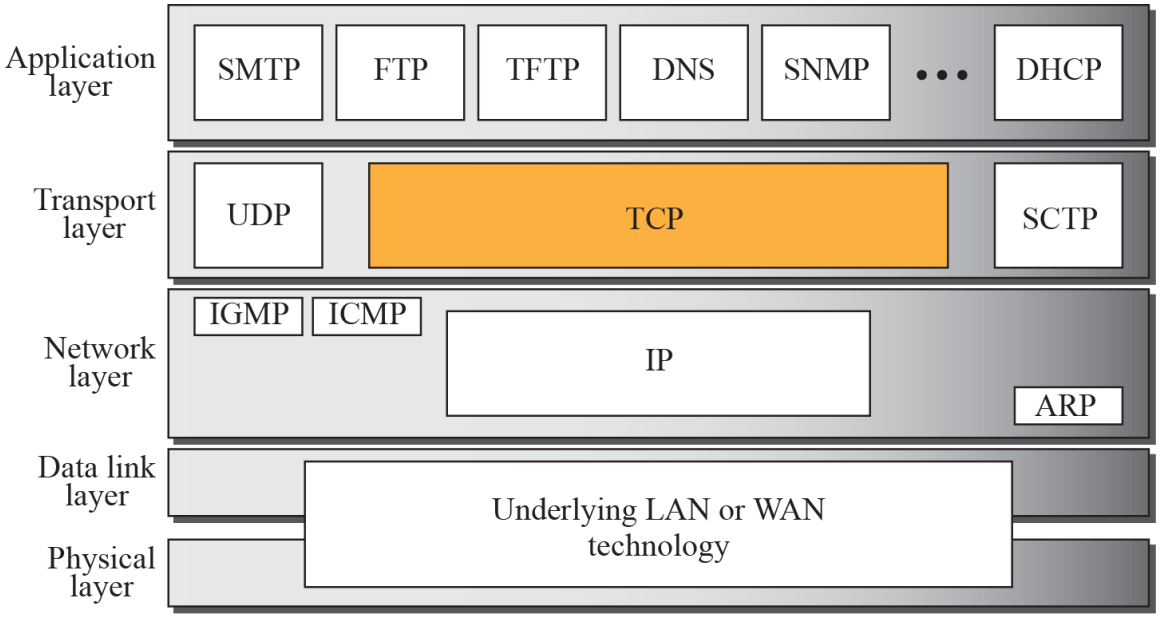
- TCP
  - Genel bilgiler
  - Başlık yapısı
  - Bağlantı kurulum, veri gönderme ve bağlantı sonlandırma
  - Akış kontrol
  - Hata kontrol
  - Tıkanıklık kontrol
- UDP
  - Genel bilgiler
  - Başlık yapısı

TCP Genel Bilgiler

- TCP bağlantı yönelimlidir ve güvenilirdir
- Kaynak ve hedef arasında sanal bir yol oluşturur. Bir mesaja ait tüm segmentler daha sonra bu sanal yol üzerinden gönderilir.
- Hata kontrol, akış kontrol ve tıkanıklık kontrol
- Her bağlantıda aktarılan veri baytları TCP tarafından numaralandırılır.
- Numaralandırma rastgele oluşturulmuş bir numara ile başlar.
- Bir segmentin sıra numarası alanındaki değer, o segmentte bulunan ilk veri baytına atanan sayıyı tanımlar.
- Bir segmentteki onay alanının değeri, bir tarafın almayı beklediği bir sonraki bayt sayısını tanımlar.
- Onay numarası kümilatif bir değerdir.

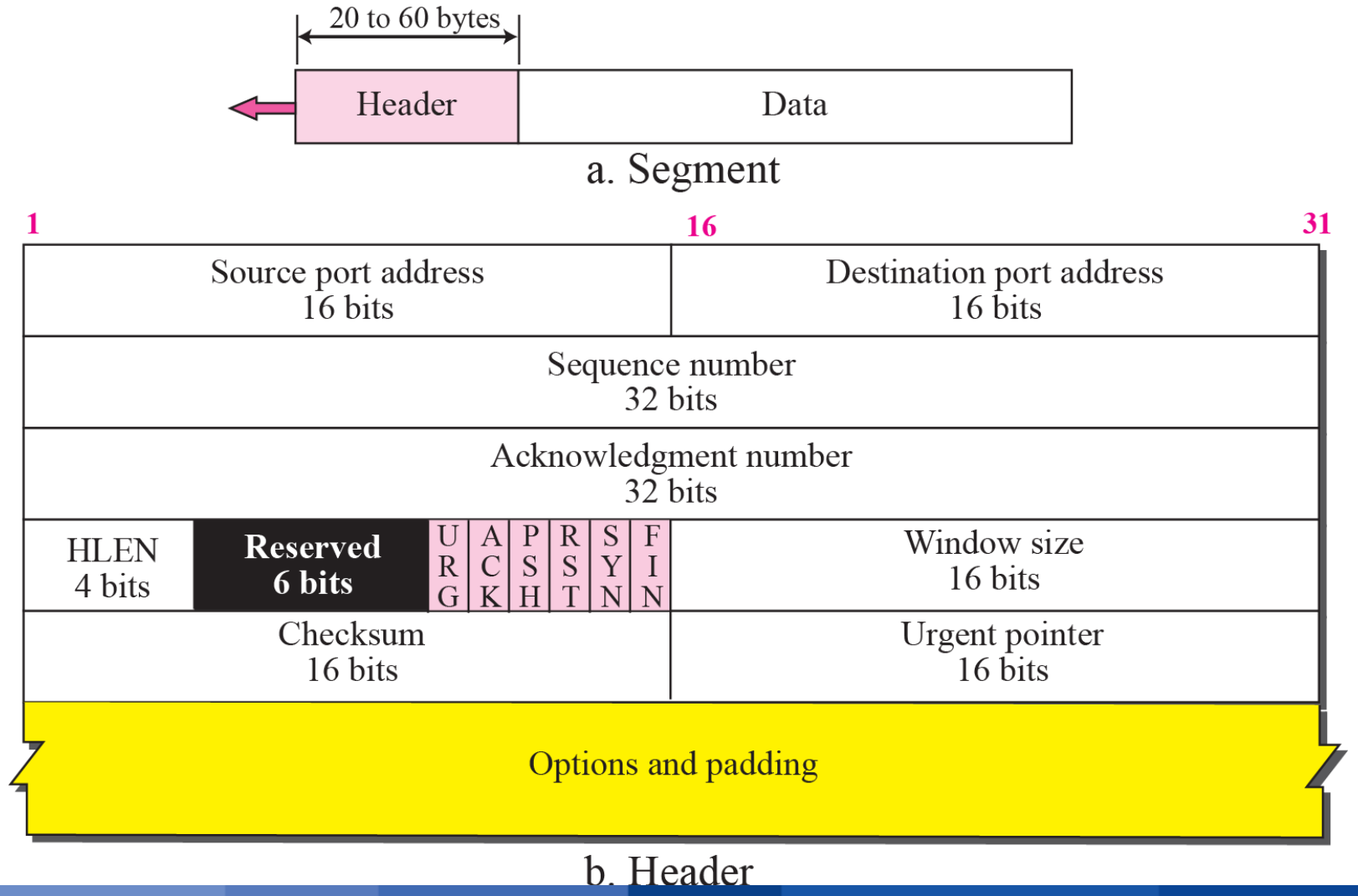
TCP (Transmission Control Protocol)

- TCP Protokolü taşıma katmanında TCP/IP protokolleri tarafından kullanılan bir iletim mekanizmasıdır. TCP segmenti olarak ifade edilir.

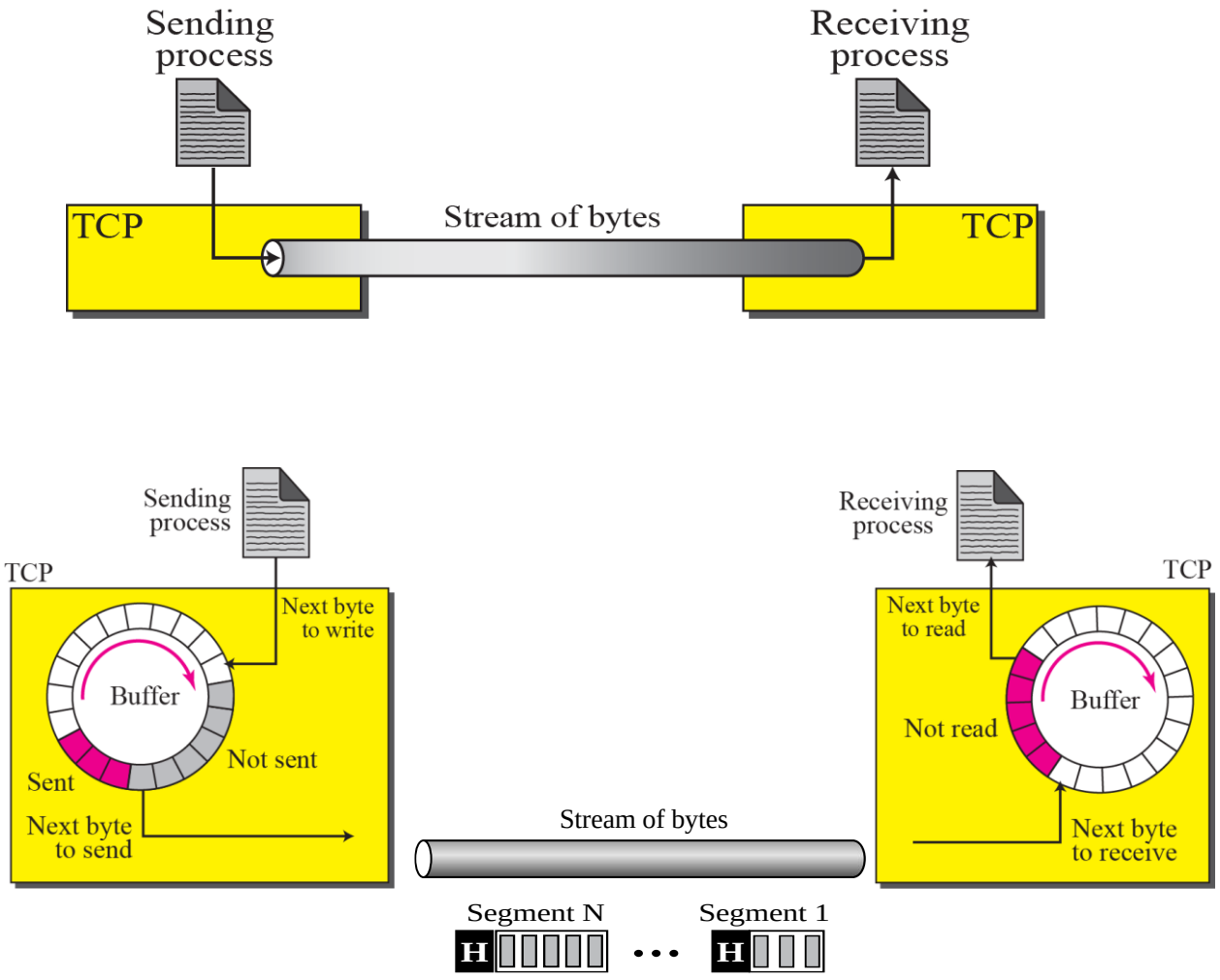


Port	*Protocol*	*Description*
7	Echo	Echoes a received datagram back to the sender
9	Discard	Discards any datagram that is received
11	Users	Active users
13	Daytime	Returns the date and the time
17	Quote	Returns a quote of the day
19	Chargen	Returns a string of characters
20 and 21	FTP	File Transfer Protocol (Data and Control)
23	TELNET	Terminal Network
25	SMTP	Simple Mail Transfer Protocol
53	DNS	Domain Name Server
67	BOOTP	Bootstrap Protocol
79	Finger	Finger
80	HTTP	Hypertext Transfer Protocol

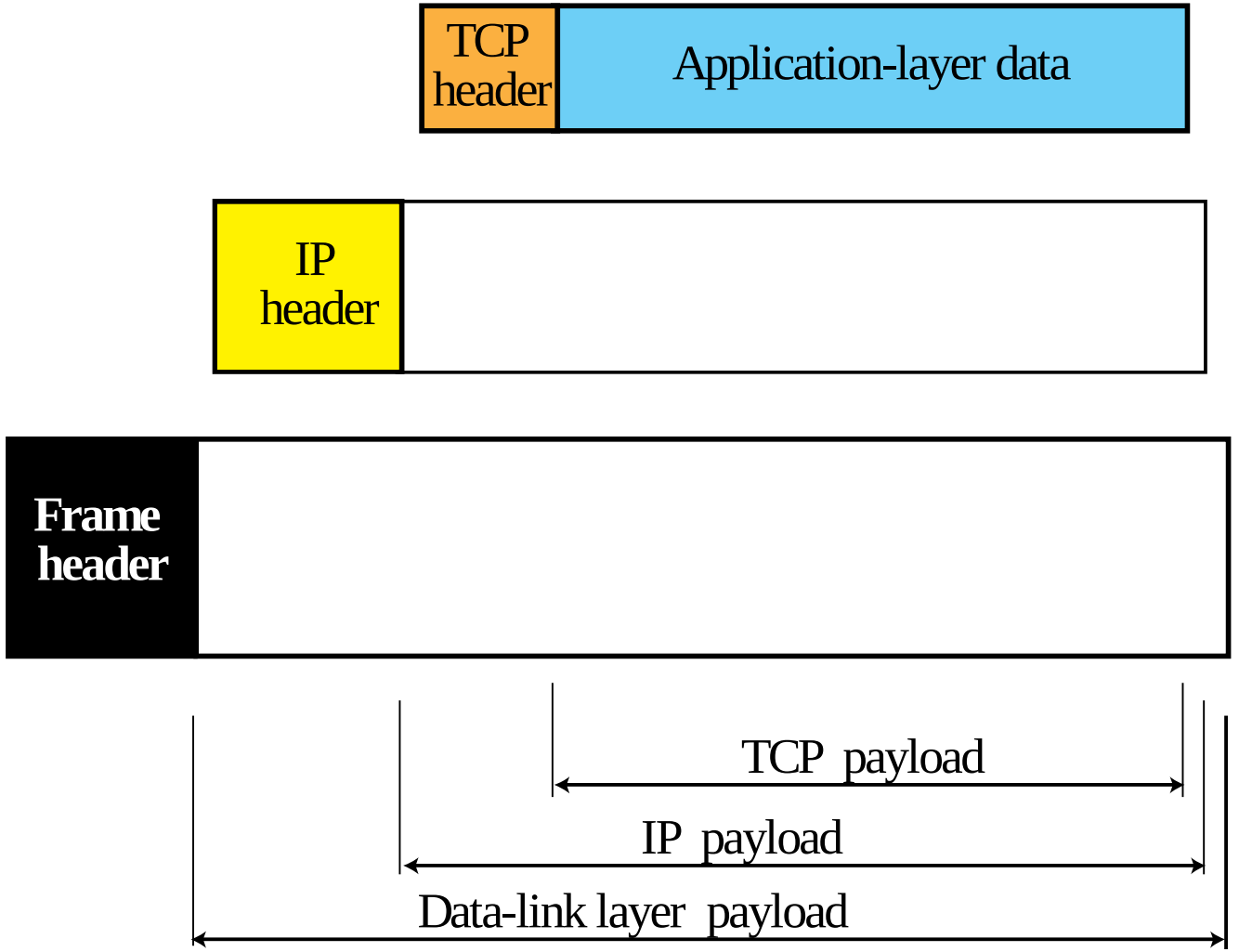
TCP Protokolü (Segment Yapısı)



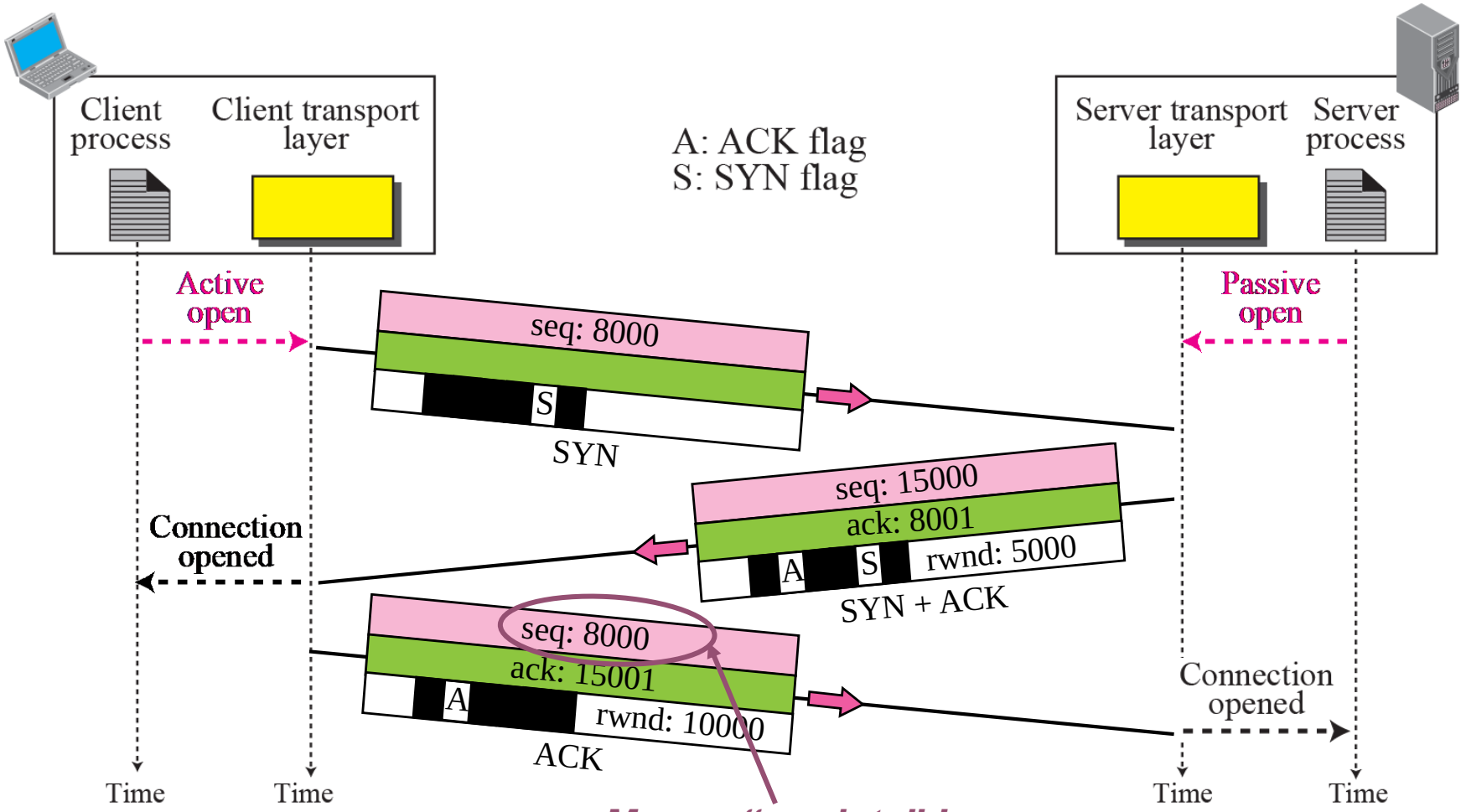
TCP Protokolü (Akış Gönderimi)



TCP Protokolü (Kapsülleme İşlemi)



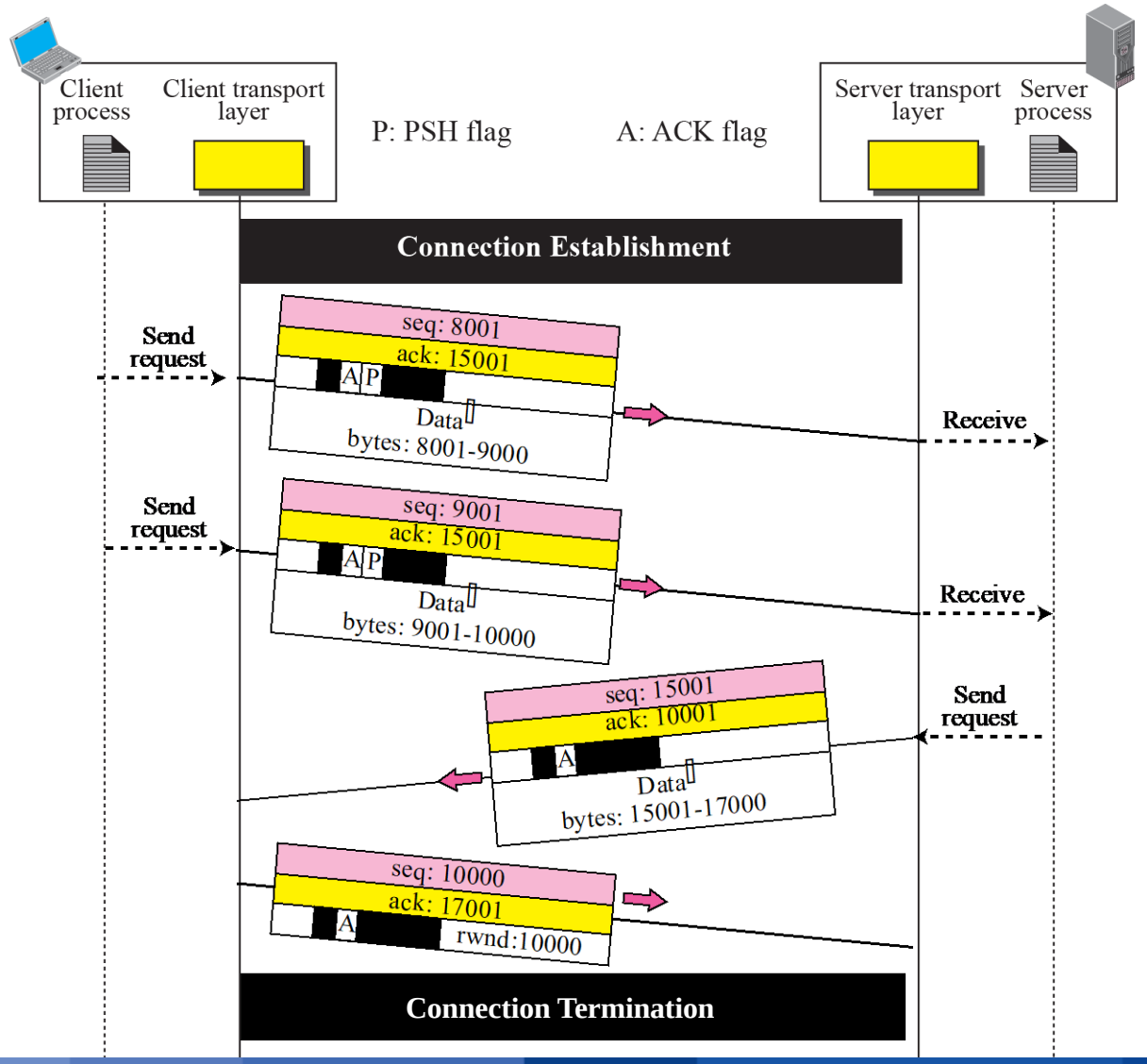
TCP Protokolü (3 Yollu El Sıkışma-Bağlantı Kurulumu)



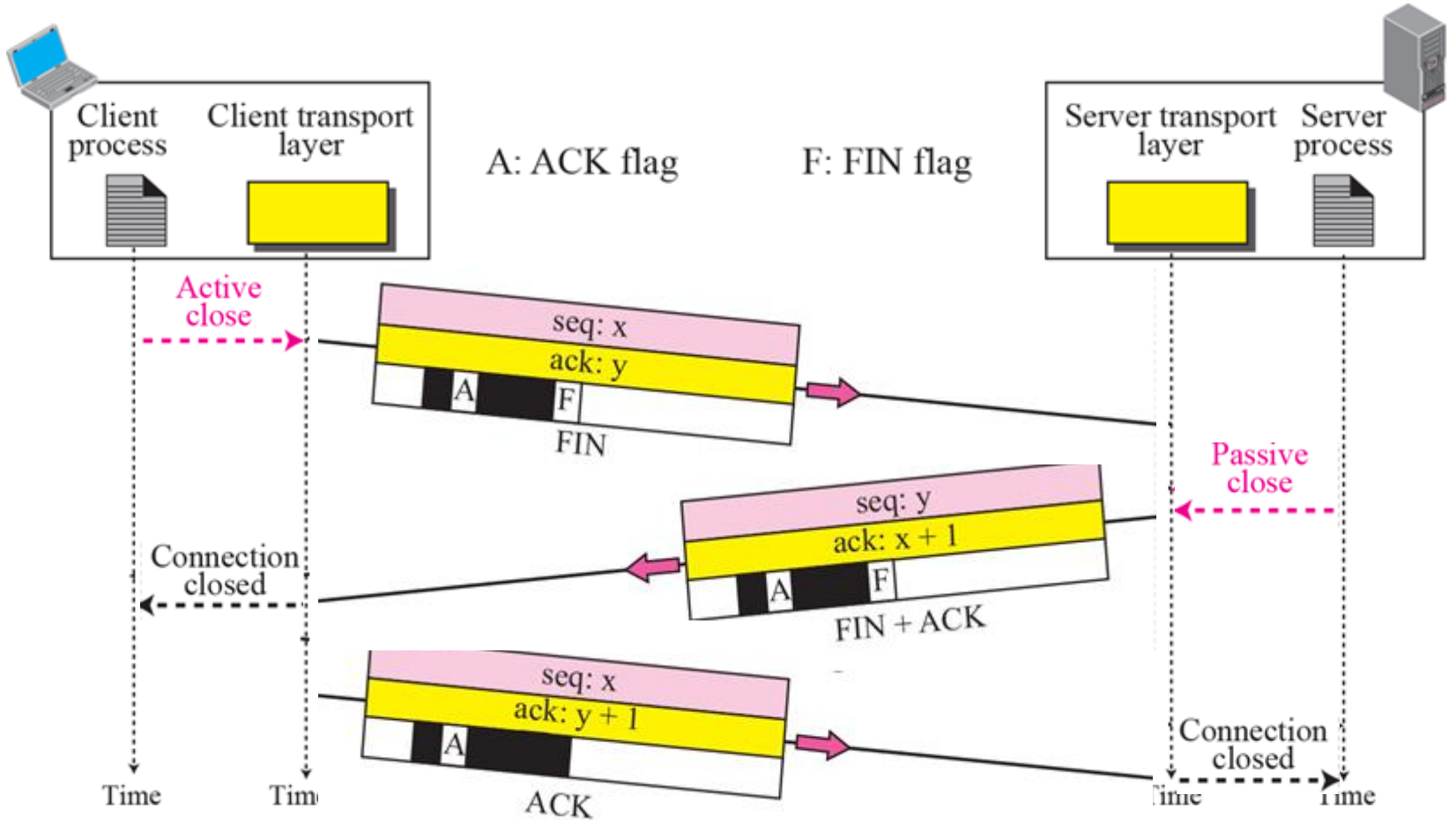
Means "no data" !

seq: 8001 if piggybacking

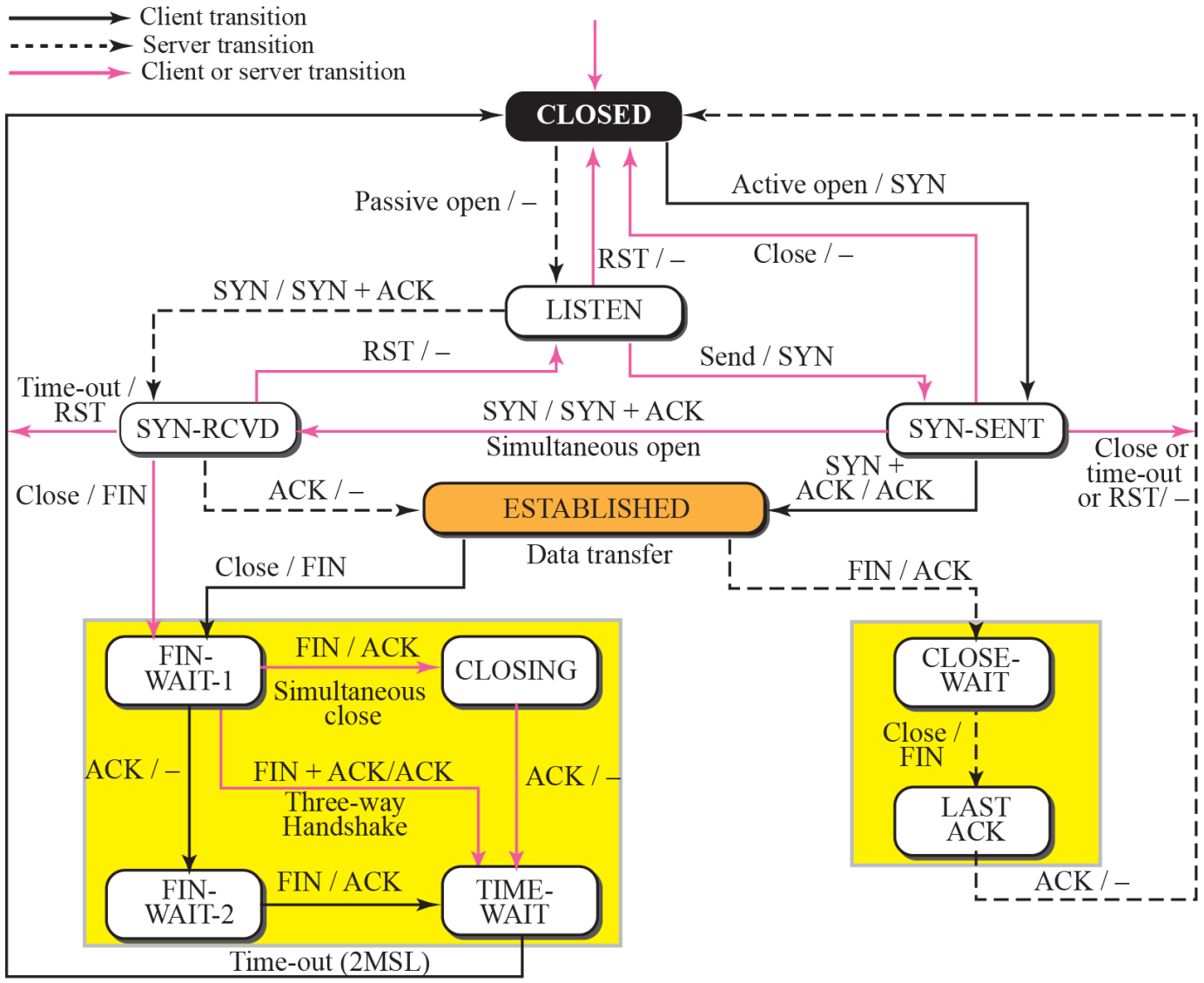
TCP Protokolü (3 Yollu El Sıkışma-Veri Transferi)



TCP Protokolü (3 Yollu El Sıkışma-Bağlantı Sonlandırma)

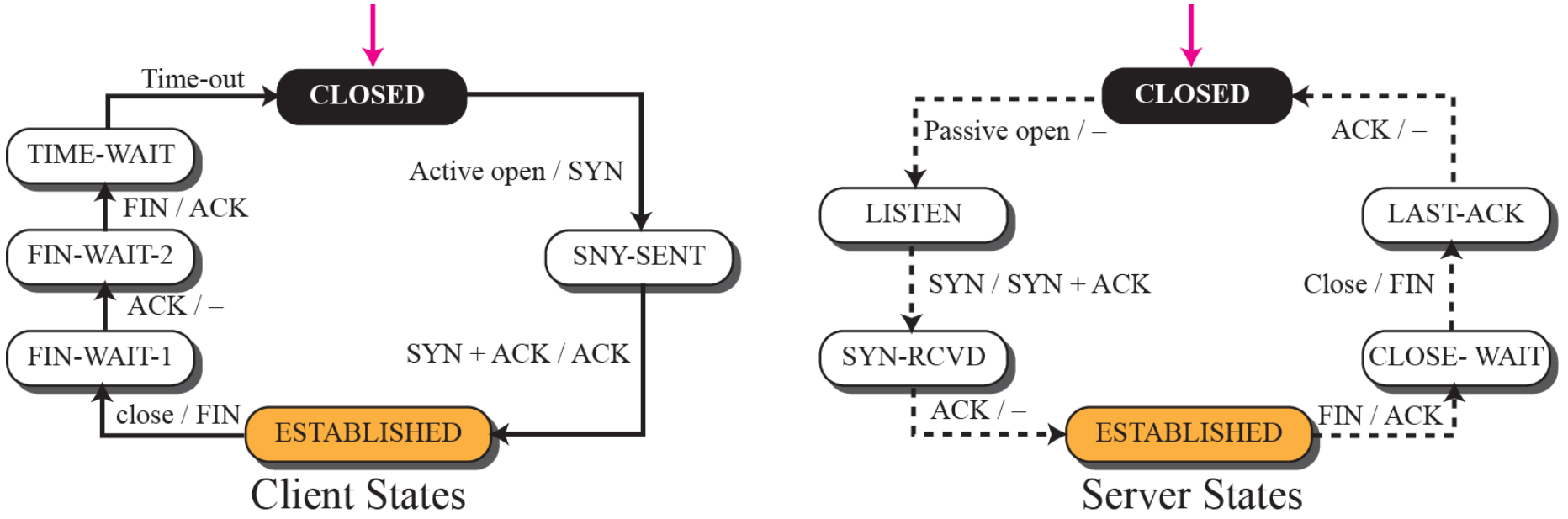


TCP Protokolü (Durum Geçiş Şeması)



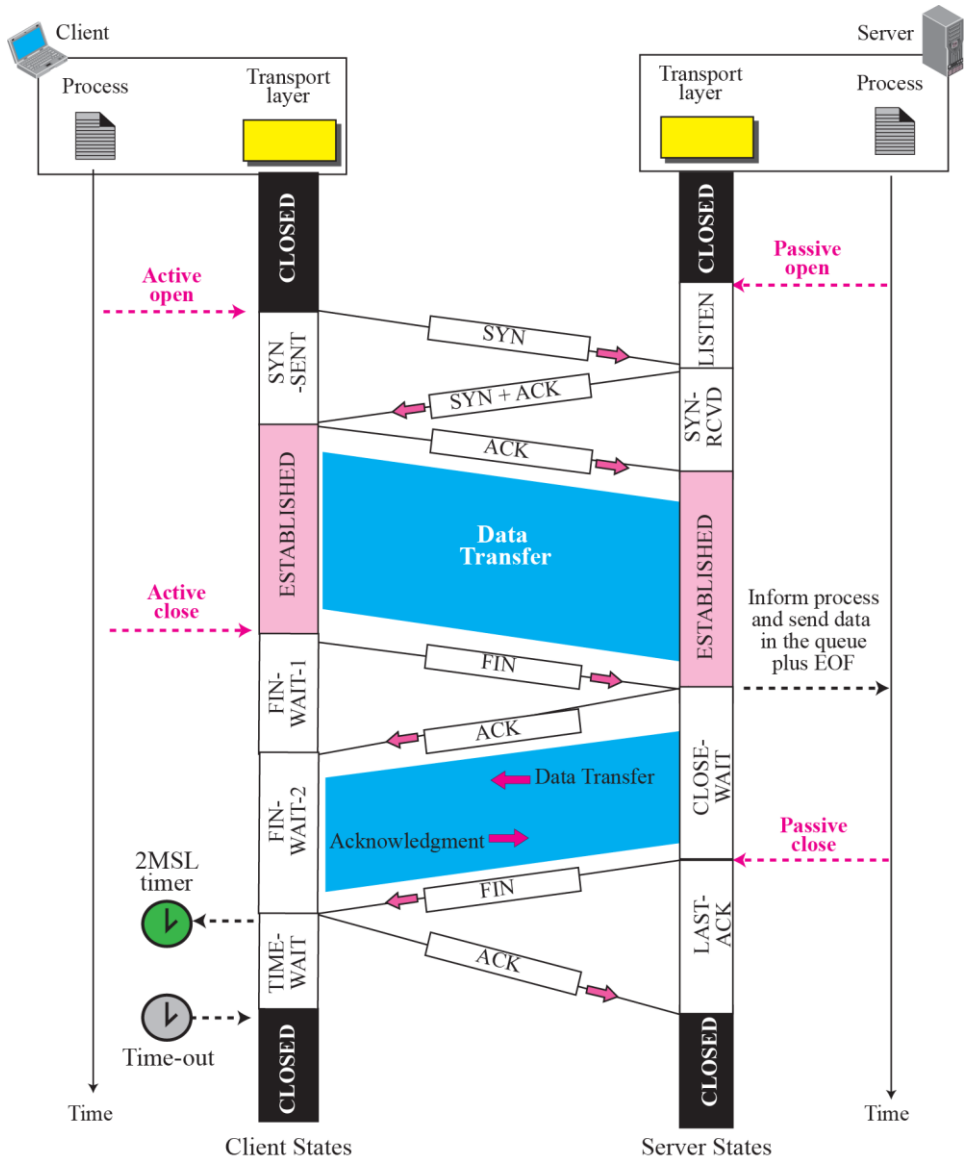
State	*Description*
`CLOSED`	No connection exists
`LISTEN`	Passive open received; waiting for SYN
`SYN-SENT`	SYN sent; waiting for ACK
`SYN-RCVD`	SYN+ACK sent; waiting for ACK
`ESTABLISHED`	Connection established; data transfer in progress
`FIN-WAIT-1`	First FIN sent; waiting for ACK
`FIN-WAIT-2`	ACK to first FIN received; waiting for second FIN
`CLOSE-WAIT`	First FIN received, ACK sent; waiting for application to close
`TIME-WAIT`	Second FIN received, ACK sent; waiting for 2MSL time-out
`LAST-ACK`	Second FIN sent; waiting for ACK
`CLOSING`	Both sides decided to close simultaneously

TCP Protokolü (Durum Geçiş Şeması-Diyagramı)



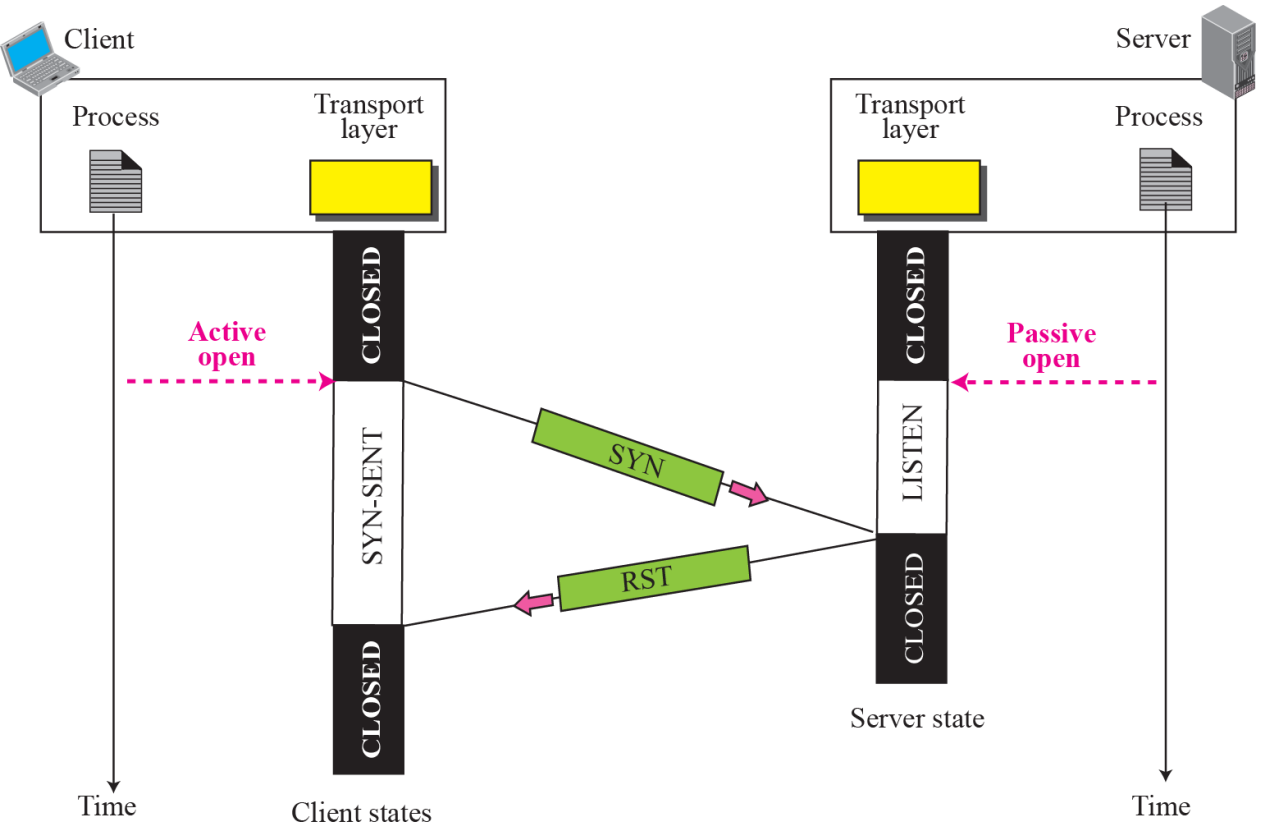
TCP Protokolü (Zaman Çizelgesi)

- Bir ACK'nın kaybolması ve yeni bir FIN'in gelmesi için yeterli zaman.
- Eğer TIME-WAIT durumu sırasında yeni bir FIN gelirse, istemci yeni bir ACK gönderir ve 2MSL zamanlayıcısını yeniden başlatır.
- Bir sonraki segmentte bir bağlantıdan yinelenen bir segmentin görünmesini önlemek için TCP, 2MSL süre geçmedikçe enkarnasyonun gerçekleşmesini gerektirmez.

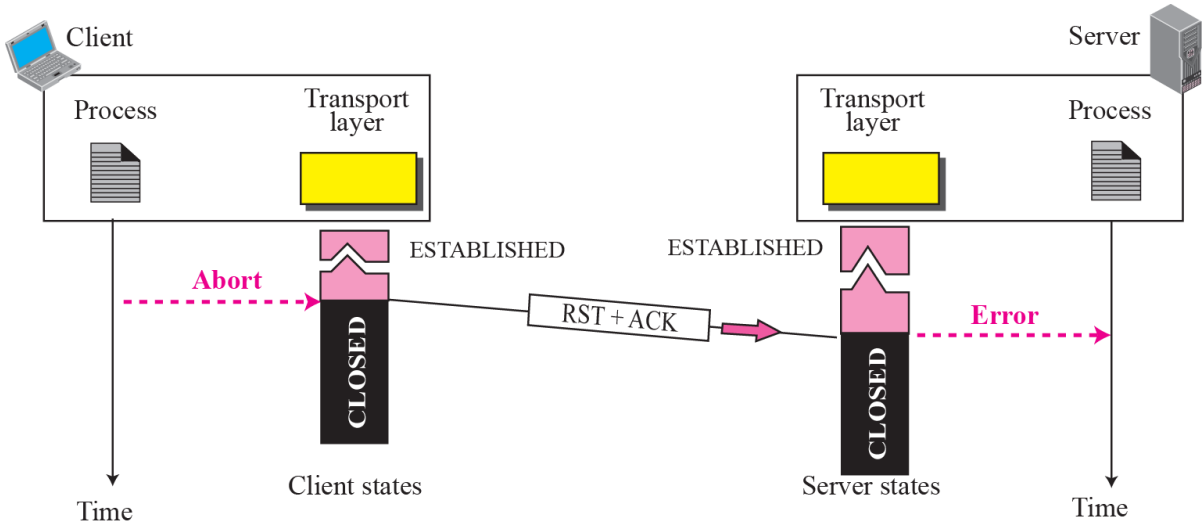


TCP Protokolü (Bağlantıları Red/İptal Etme)

Bağlantı Red Etme

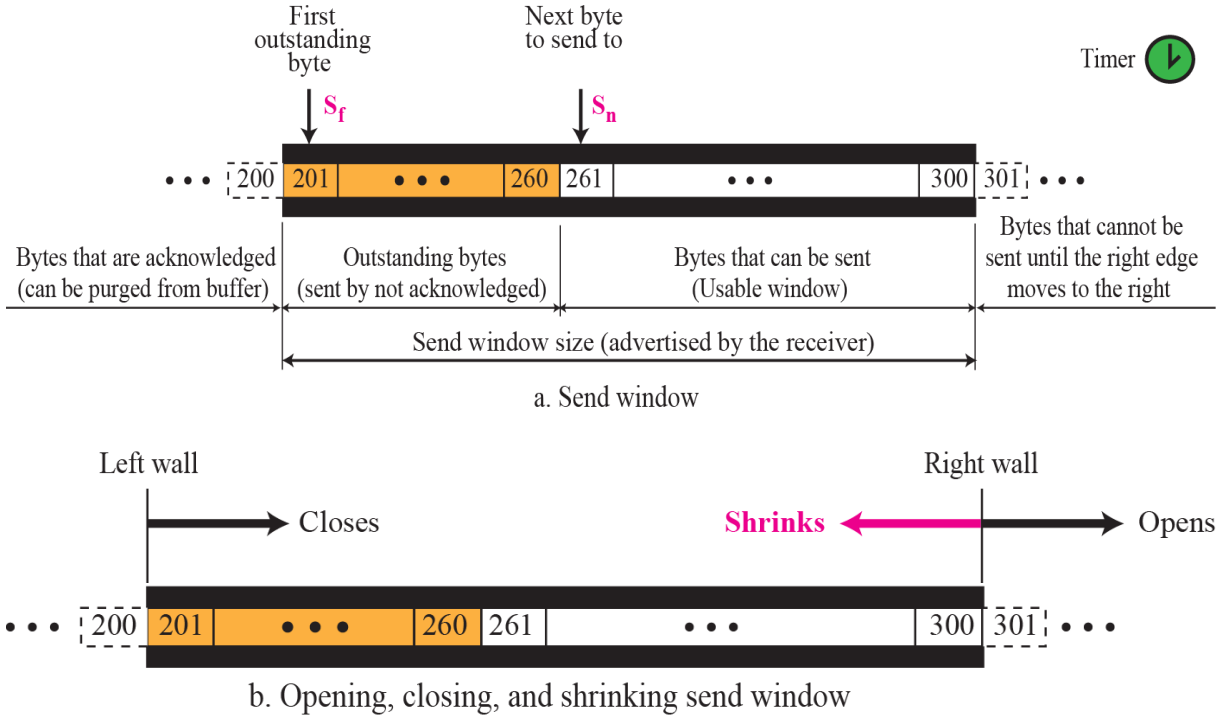


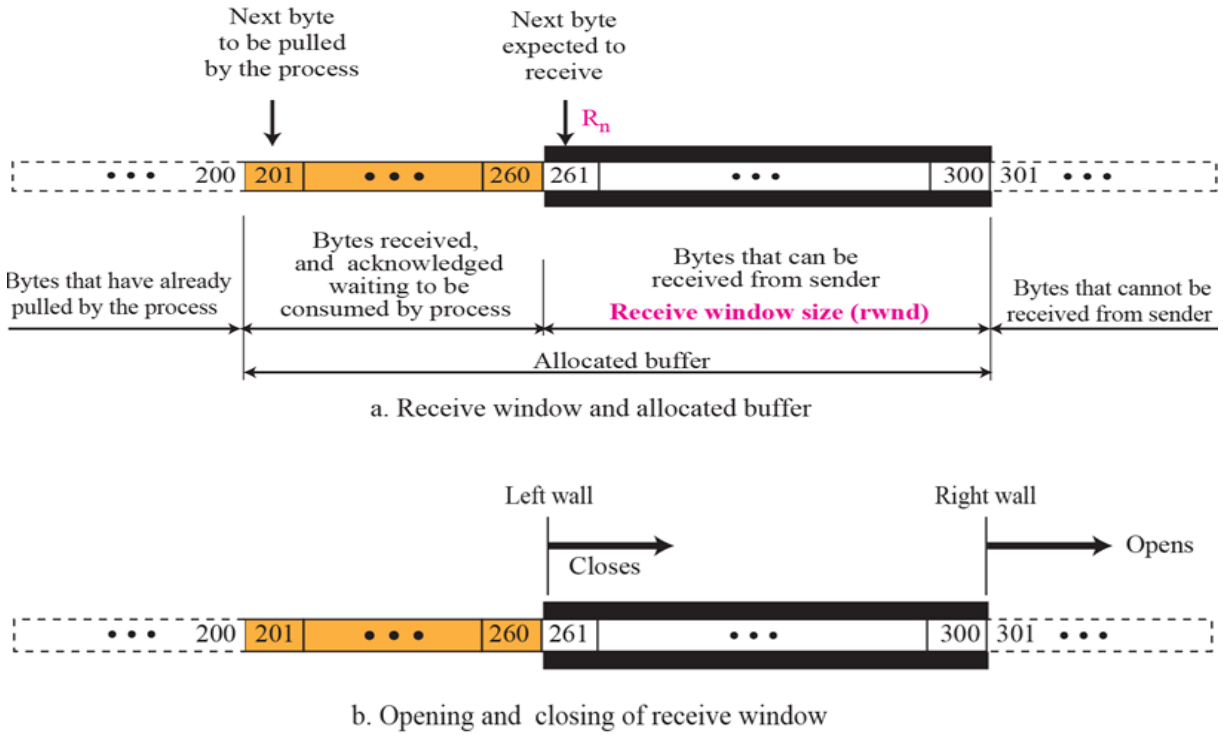
Bağlantı İptal Etme



TCP Protokolü (Windows/Pencere Kavramı)

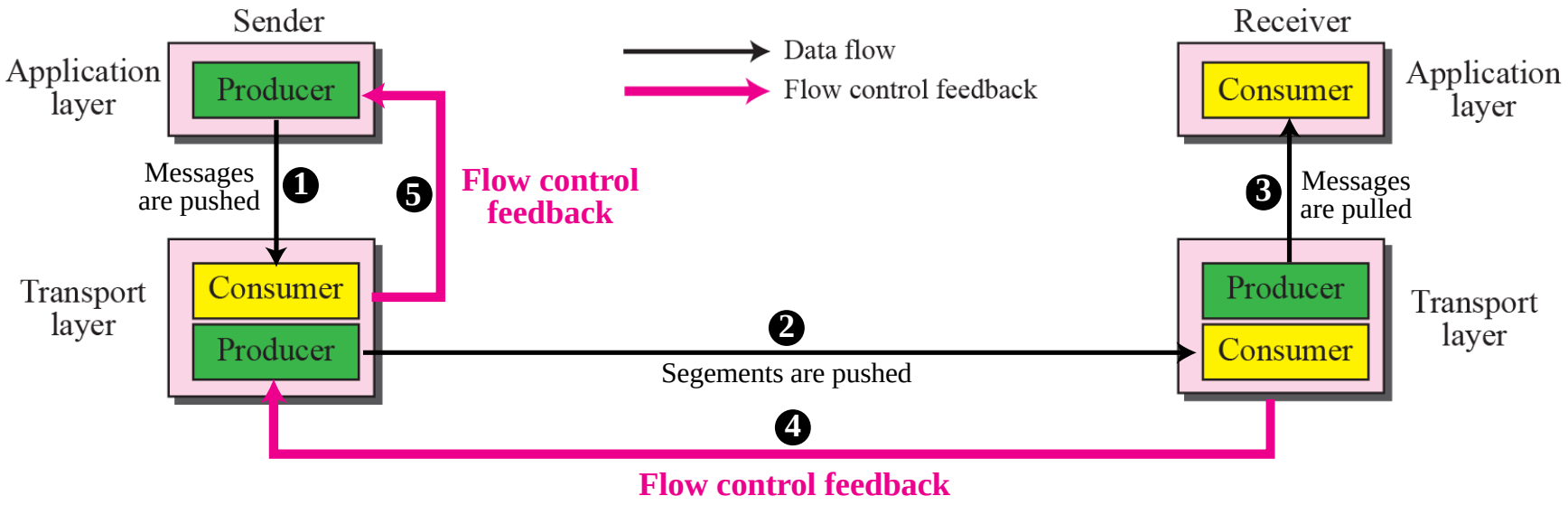
- TCP, veri aktarımının her yönü için iki pencere (gönderme penceresi ve alma penceresi) kullanır; bu, çift yönlü iletişim için dört pencere anlamına gelir.



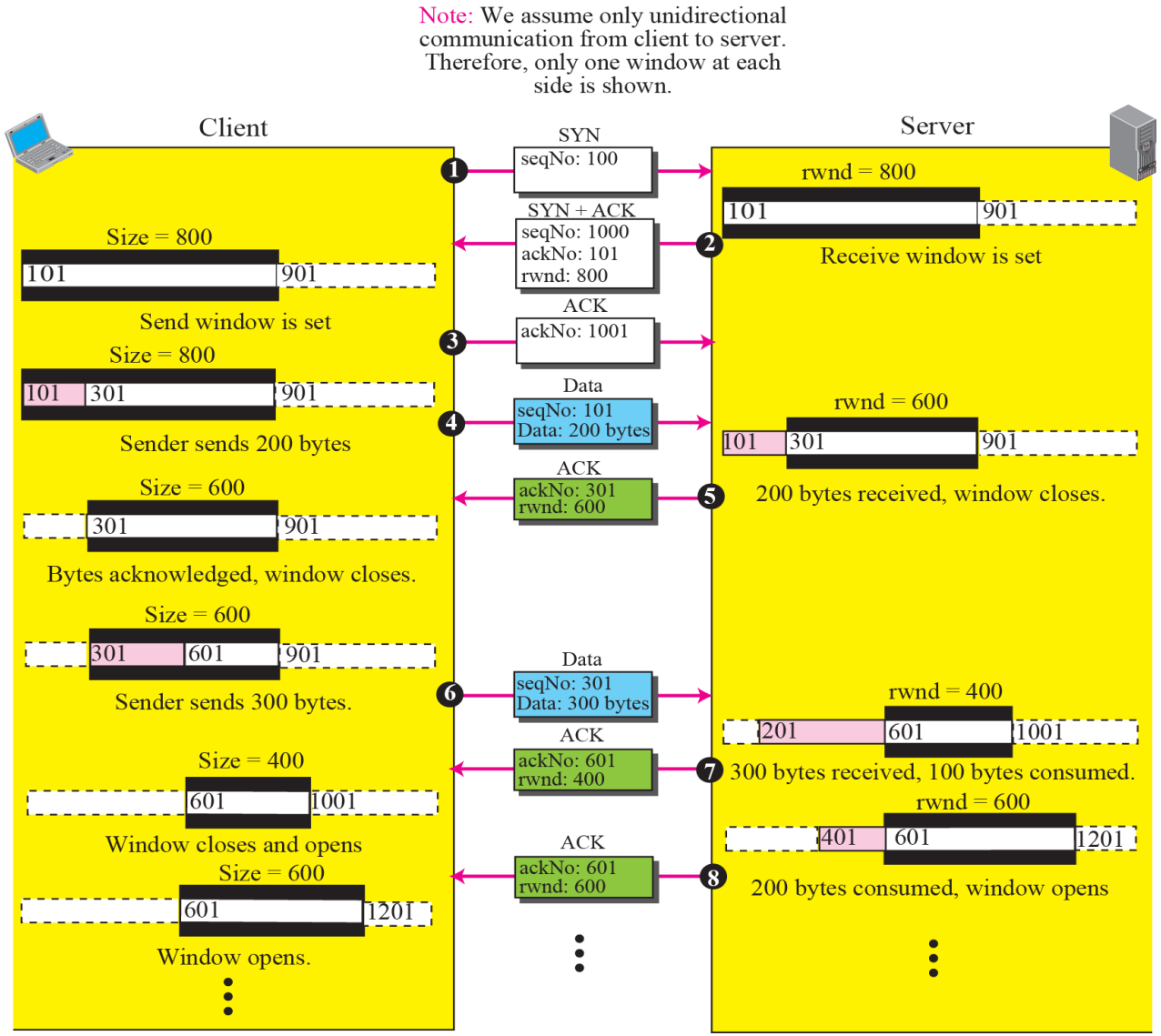


TCP Protokolü (Akış Kontrolü)

- Akış kontrolü, üreticinin veri yaratma hızını, tüketicinin verileri kullanma oranıyla dengeler. TCP akış kontrolünü hata kontrolünden ayırır.
- Aşağıda bir gönderen ile bir alıcı arasındaki tek yönlü veri aktarımını gösterilmektedir.



TCP Protokolü (Akış Kontrolü-Örnek Uygulama)



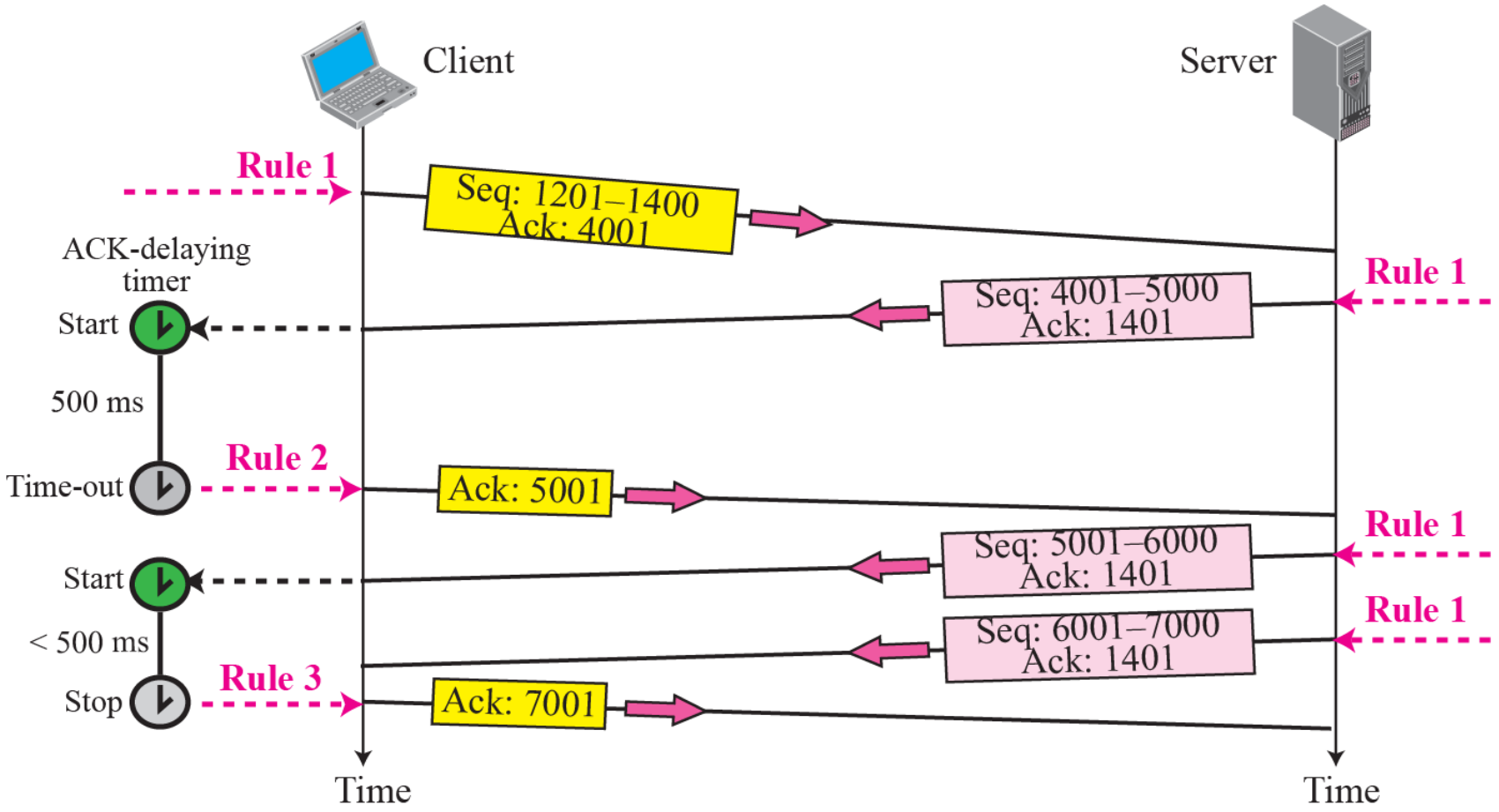
TCP Protokolü (Hata Kontrolü)

- TCP, güvenilir bir aktarım katmanı protokolüdür.
- Bu, TCP'ye bir veri akışı sağlayan bir uygulama programının, tüm akışı diğer programdaki uygulama programına ***hatasız*** ve herhangi bir ***parça kaybolmadan*** veya ***çoğaltılmadan*** iletmek için TCP'ye bağlı olduğu anlamına gelir.
- TCP'de hata denetimi üç araç kullanılarak sağlanır:
  - ***Checksum***
  - ***Onay***
  - ***Zaman aşımı***
- TCP en iyi bir Seçici Tekrarlama protokolü olarak modellenebilir.

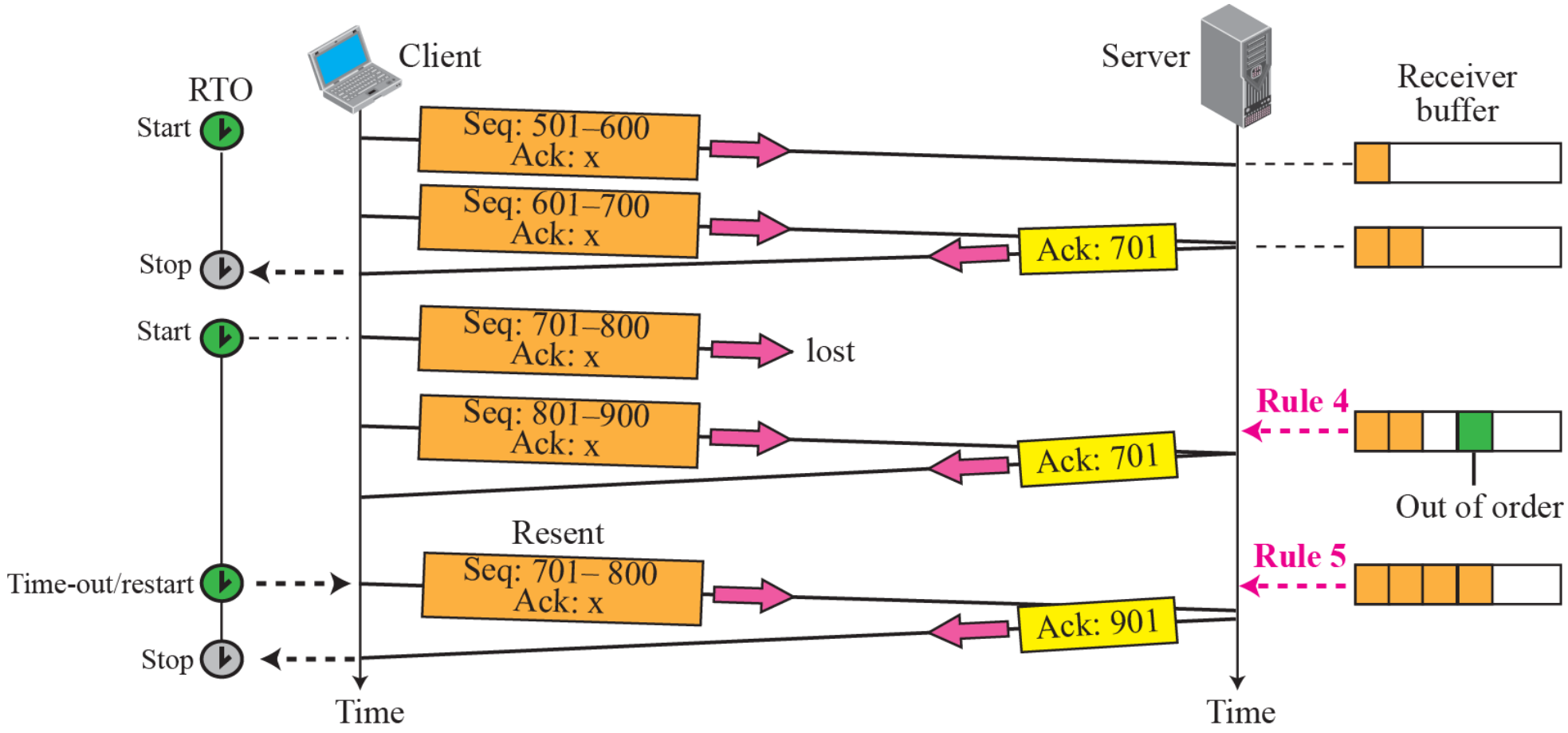
TCP Protokolü (Hata Kontrolü-Senaryolar)

- *Normal işlem*
- ***Segment Kaybı***
- *Hızlı Yeniden İletim*
- ***Kayıp Onay Bildirimi***
- *Kayıp Segmentin Yeniden Gönderimi*

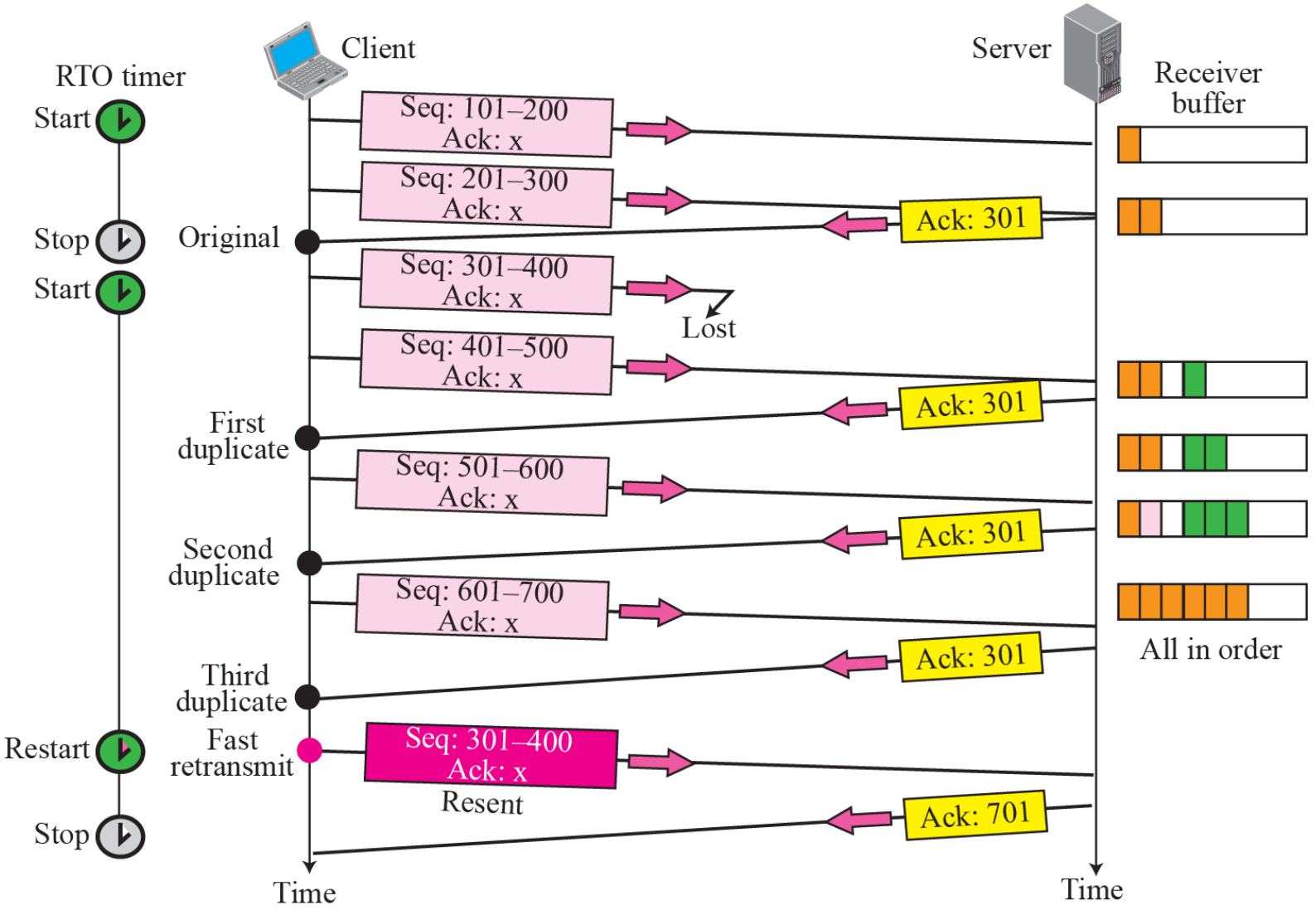
TCP Protokolü (Normal İşlem)



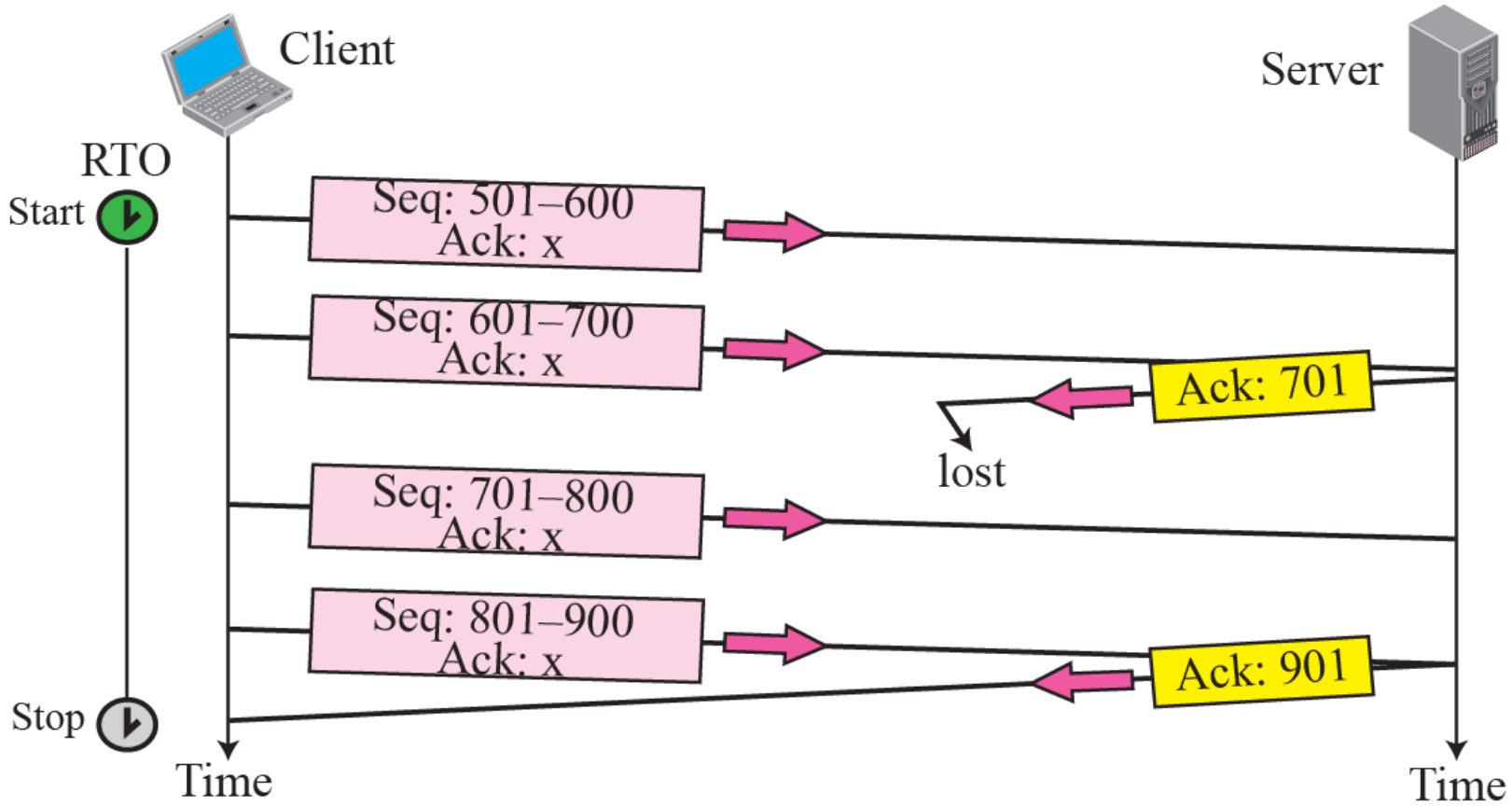
TCP Protokolü (Segment Kaybı)



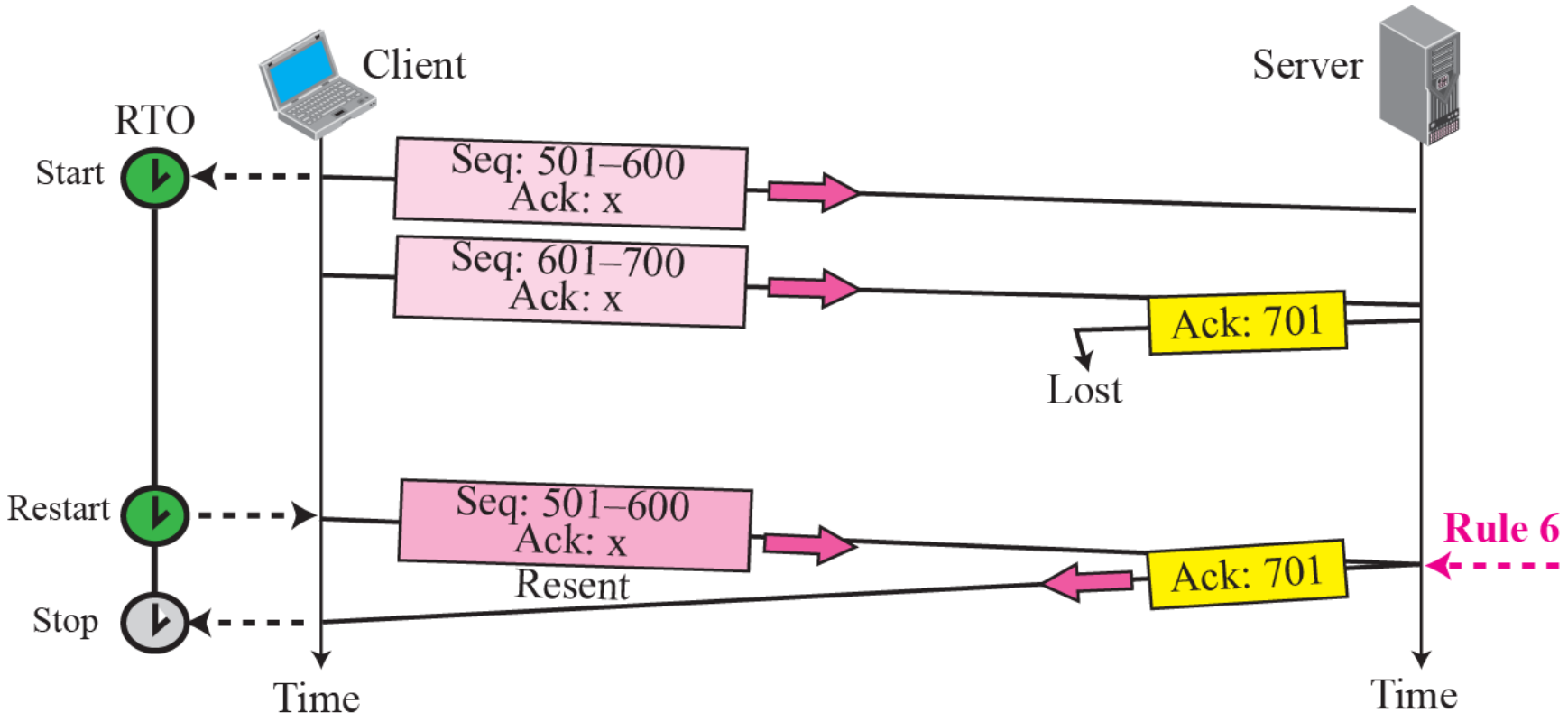
TCP Protokolü (Hızlı Yeniden İletim)



TCP Protokolü (Kayıp Onay Bildirimi)



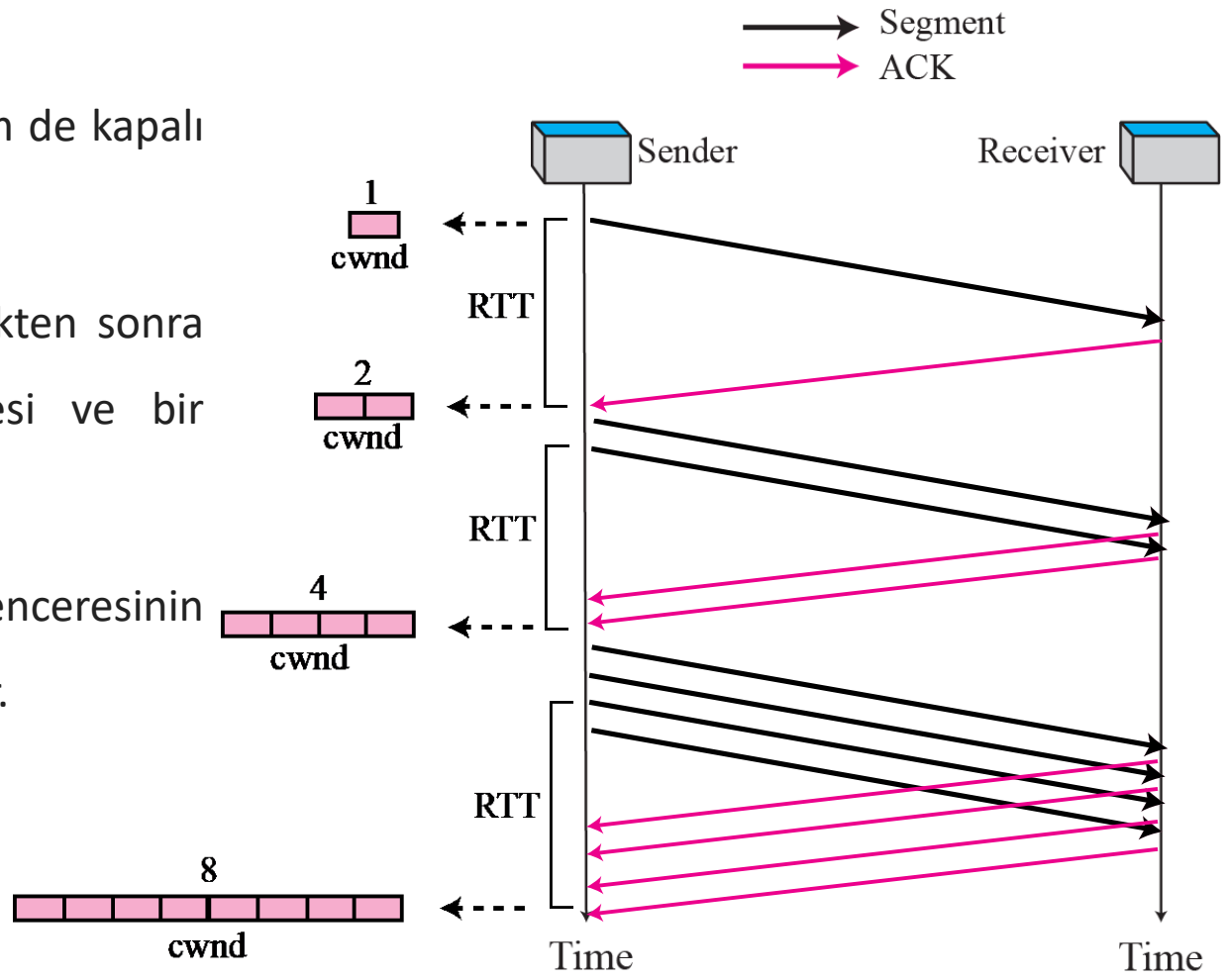
TCP Protokolü (Kayıp Segmentin Yeniden Gönderimi)



TCP Protokolü (Sıkışıklık Kontrolü)

Yavaş Başlatma, Üstel Artış

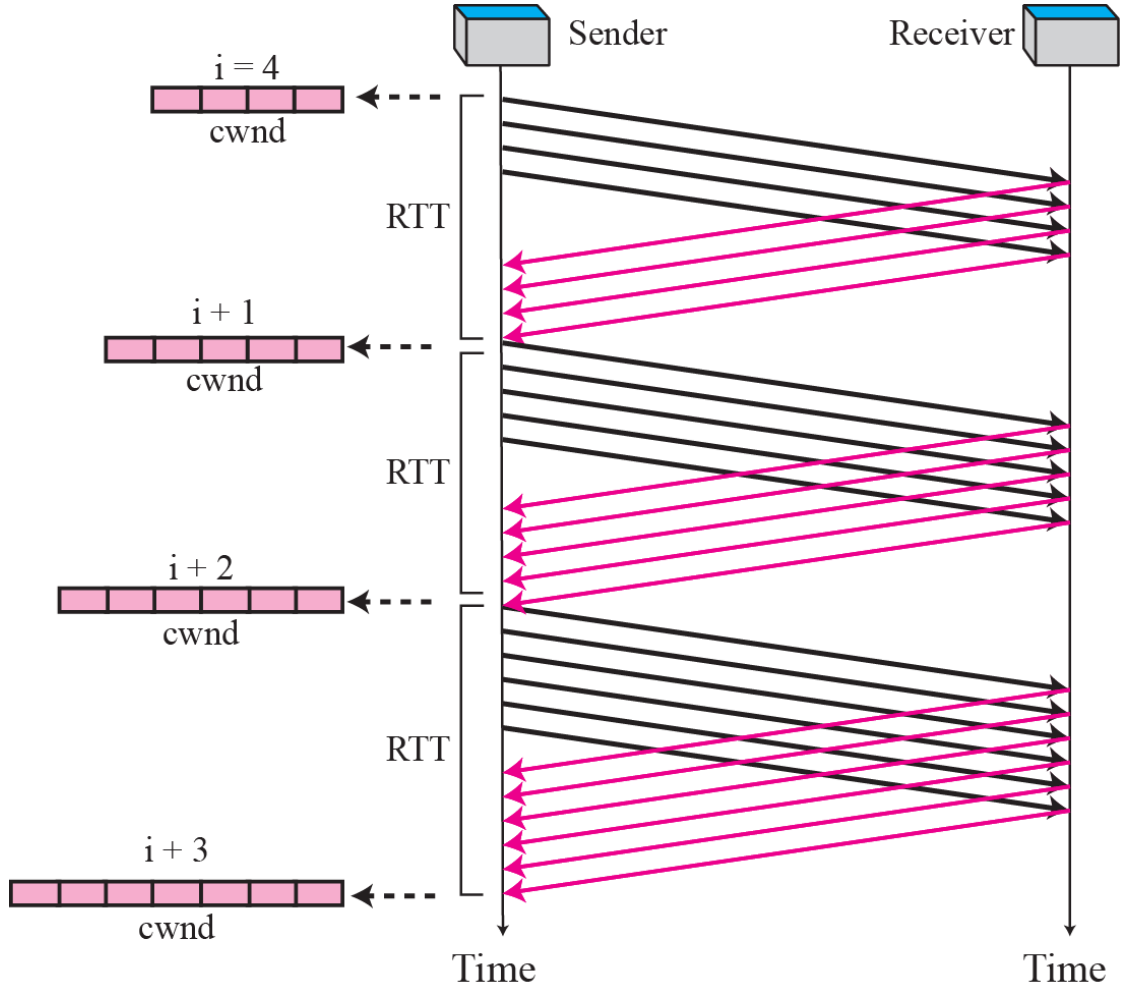
- TCP'deki tıkanıklık kontrolü, hem açık döngü hem de kapalı döngü mekanizmalarına dayanır.
- TCP, tıkanıklığı önleyen ve tıkanıklığı gerçekleştikten sonra algılayan ve hafifleten bir tıkanıklık penceresi ve bir tıkanıklık ilkesi kullanır.
- Yavaş başlatma algoritmasında, tıkanıklık penceresinin boyutu bir eşiğe ulaşıncaya kadar katlanarak artar.



TCP Protokolü (Sıkışıklık Kontrolü)

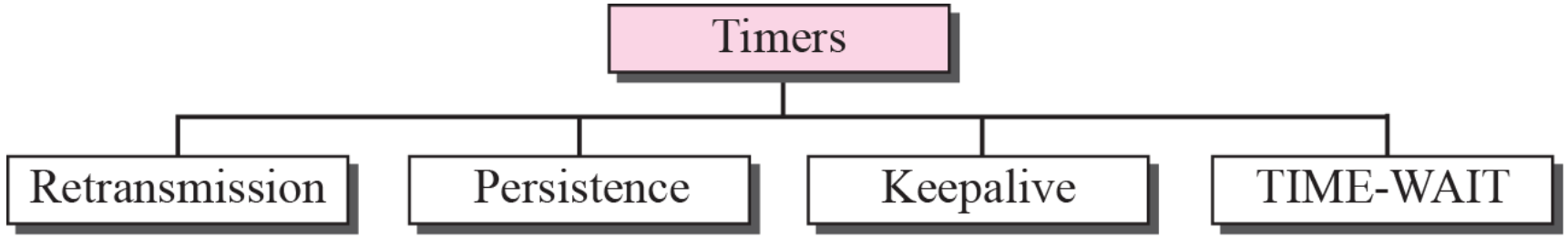
- Tıkanıklıktan kaçınma algoritmasında tıkanıklık penceresinin boyutu kadar artar tıkanıklık tespit edildi.

Tıkanıklıktan Kaçış, Katkı Artışı



TCP Protokolü (Zamanlayıcılar)

- TCP, iletişim sırasında aşırı gecikmelerle karşılaşmamak için birkaç zamanlayıcı kullanır.
- Bu zamanlayıcıların birçoğu hassastır, ilk analizde hemen görülmeyen problemlerle başa çıkmaktadır.
- TCP tarafından kullanılan zamanlayıcıların her biri, verilerin bir bağlantıdan diğerine düzgün bir şekilde gönderilmesini sağlamaktadır.



TCP Protokolü (Zamanlayıcılar-Retransmissio-RTT)

- Kayıp segmentleri yeniden iletmek için TCP yeniden iletim zaman aşımı (RTO) kullanır.
- TCP bir segment gönderdiğinde, zamanlama başlar ve onay alındığında durur.
- Zamanlayıcının süresi dolarsa zaman aşımı oluşur ve segment yeniden iletilir.
- RTO (yeniden iletim zaman aşımı 1 RTT içindir) yeniden iletim zaman aşımını hesaplamak için öncelikle RTT'yi (gidiş dönüş süresi) hesaplaması gerekir.
  - ***Measured RTT(RTTm)***
  - ***Smoothed RTT(RTTs)***
  - ***Deviated RTT(RTTd)***

TCP Protokolü (Zamanlayıcılar-Persistent)

- Sıfır pencere boyutu kilitlenme durumuyla başa çıkmak için TCP bir kalıcılık zamanlayıcısı kullanır.
- Gönderen TCP sıfır pencere boyutuyla bir onay aldığında bir kalıcılık zamanlayıcısı başlatır. Kalıcı zamanlayıcı söndüğünde, gönderen TCP prob adı verilen özel bir segment gönderir.
- Bu segment yalnızca 1 bayt yeni veri içeriyor. Bir sıra numarası vardır, ancak sıra numarası asla kabul edilmez; verilerin geri kalanı için sıra numarasının hesaplanmasında bile göz ardı edilir.
- Prob, alıcı TCP'nin kaybolan bildirimi yeniden göndermesine neden olur.

TCP Protokolü (Zamanlayıcılar-Keep Alive)

- İki TCP arasında uzun süre kullanılmayan bir bağlantıyı önlemek için bir tutma zamanlayıcısı kullanılır.
- İstemci bir sunucuya TCP bağlantısı açarsa bazı verileri aktarır ve sessiz hale gelirse istemci çökecektir. Bu durumda, bağlantı sonsuza kadar açık kalır. Yani bir zamanlayıcı kullanılır.
- Sunucu bir istemciden her haber aldığında, bu zamanlayıcıyı sıfırlar. Zaman aşımı genellikle 2 saattir.
- Sunucu 2 saat sonra istemciden haber alamazsa, bir prob segmenti gönderir.
- Her biri 75 s arayla 10 probdan sonra yanıt yoksa, istemcinin kapalı olduğunu varsayar ve bağlantıyı sonlandırır.

TCP Protokolü (Zamanlayıcılar-Time Wait)

- Bu zamanlayıcı tcp bağlantısının sonlandırılması sırasında kullanılır.
- Zamanlayıcı, 2. FIN için son Ack gönderildikten ve bağlantı kapatıldıktan sonra başlar.
- Bir TCP bağlantısı kapatıldıktan sonra, ağ üzerinden yoluna devam eden datagramların kapalı bağlantı noktasına erişmeye çalışması mümkündür.
- Sessiz zamanlayıcı, yeni kapatılan bağlantı noktasının hızlı bir şekilde tekrar açılmasını ve bu son datagramları almasını önlemeyi amaçlamaktadır.

TCP Protokolü (Zamanlayıcılar-Örnek)

1. SYN segmenti gönderildiğinde, RTTM, RTTS veya RTTD için bir değer yoktur. RTO değeri 6.00 saniyeye ayarlanmıştır. Yanda bu değişkenin değeri gösterilmektedir:

$$RTO = 6$$

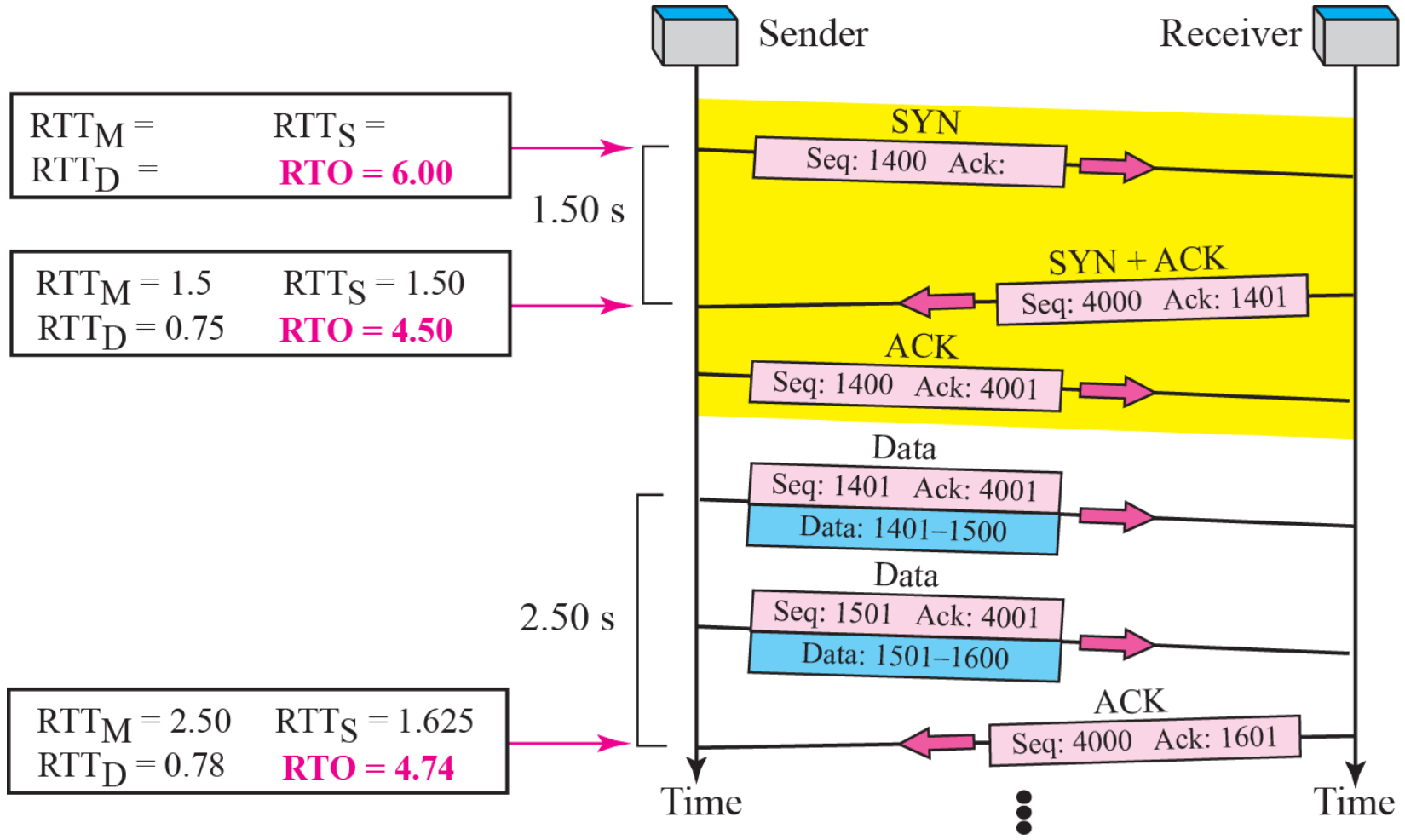
2. SYN + ACK segmenti geldiğinde, RTTM ölçülür ve 1,5 saniyeye eşittir.

$$RTT_M = 1.5$$
$$RTT_S = 1.5$$
$$RTT_D = (1.5) / 2 = 0.75$$
$$RTO = 1.5 + 4 \times 0.75 = 4.5$$

3. İlk veri segmenti gönderildiğinde, yeni bir RTT ölçümü başlar. Bir ölçüm zaten devam ettiği için ikinci veri segmenti için RTT ölçümü başlatılmaz. Son ACK segmentinin gelişi, bir sonraki RTTM değerini hesaplamak için kullanılır. Son ACK segmenti her iki veri segmentini de (kümülatif) kabul etse de, varış noktası ilk segment için RTTM değerini kesinleştirir. Bu değişkenlerin değerleri artık aşağıda gösterildiği gibidir.

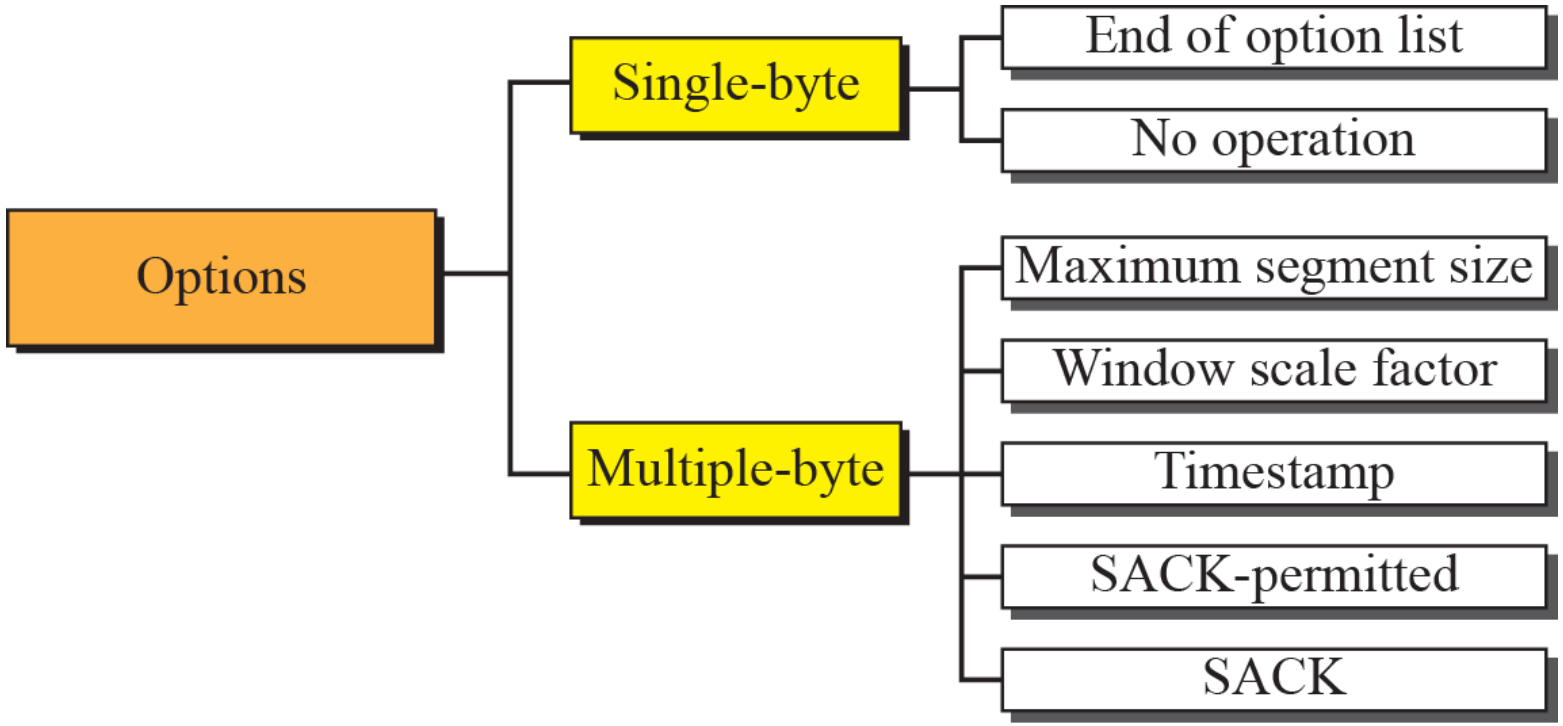
$$RTT_M = 2.5$$
$$RTT_S = 7/8 \times 1.5 + (1/8) \times 2.5 = 1.625$$
$$RTT_D = 3/4\ (7.5) + (1/4) \times |1.625 - 2.5|$$
$$RTO = 1.625 + 4 \times 0.78 = 4.74$$

TCP Protokolü (Zamanlayıcılar-Örnek)

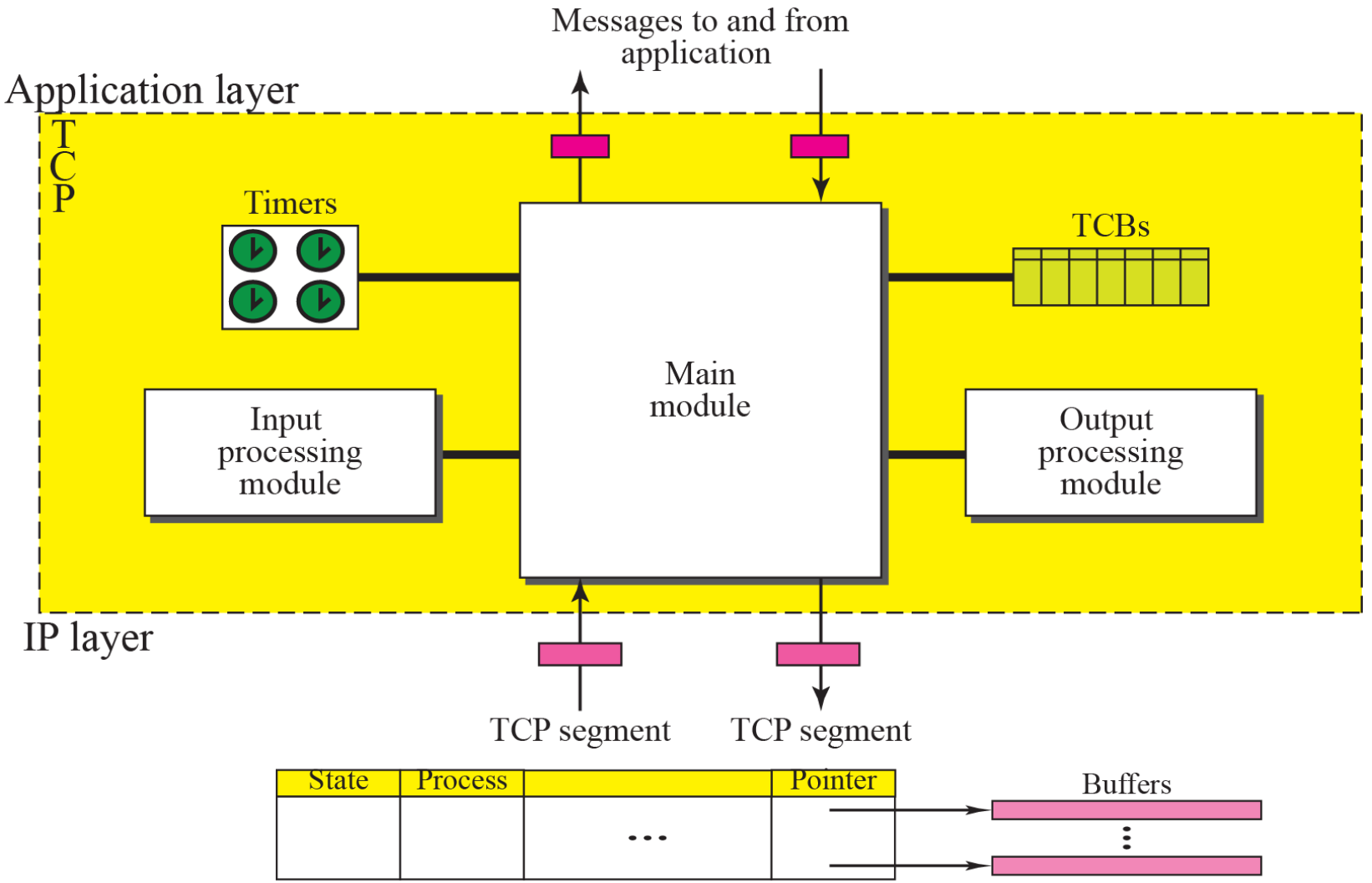


TCP Protokolü (Opsiyonlar)

- TCP üstbilgisinde 40 bayta kadar isteğe bağlı bilgi bulunabilir.
- Seçenekler hedefe ek bilgi aktarır veya diğer seçenekleri hizalar.



TCP Protokolü (İletim Kontrol Bloğu)



UDP (User Datagram Protocol)

- UDP Protokolü taşıma katmanında TCP/IP protokolleri tarafından kullanılan bir iletim mekanizmasıdır. Datagram olarak ifade edilir.
- Kullanıcı Datagram Protokolü (UDP) **bağlantısız, güvenilir olmayan** bir aktarım protokolüdür.
- UDP'de bir akış kontrol mekanizması ve alınan paketleri için bir **onay/geri bildirim mekanizması** yoktur.
- Kısıtlı olarak hata kontrol mekanizması sağlamaktadır. Eğer hata tespit ederse, paketi drop eder.

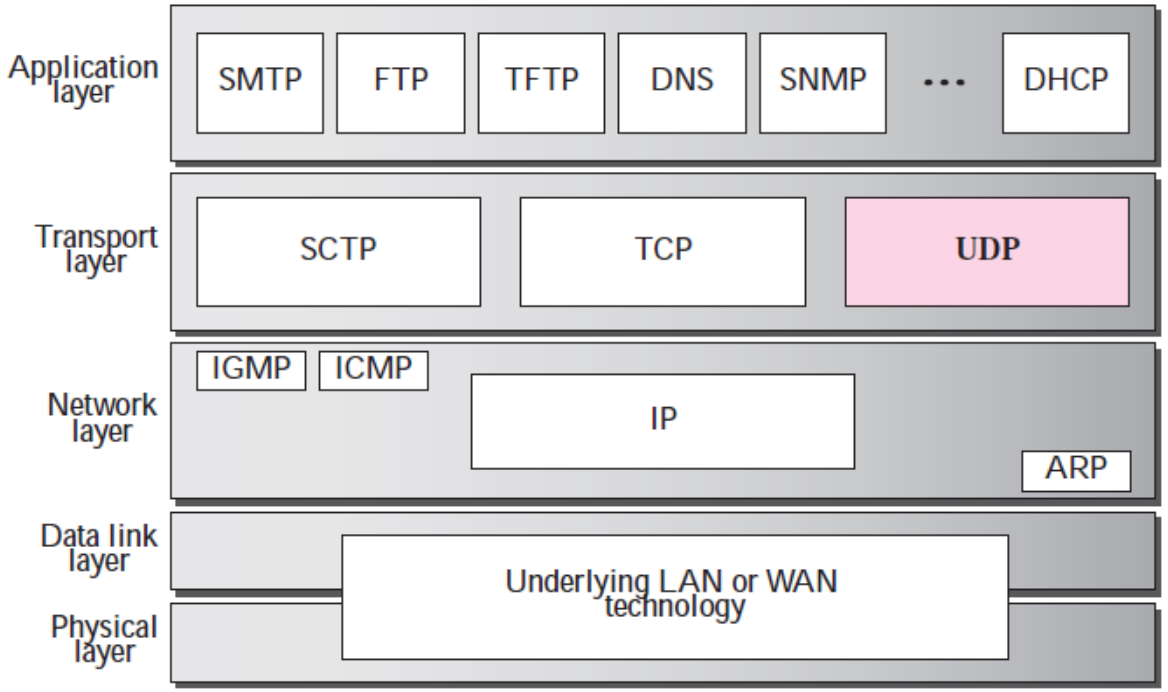
➢ Encapsulation-Decapsulation

➢ Kuyruklama

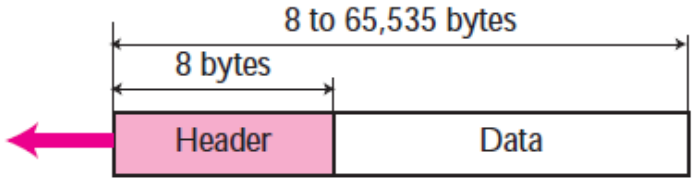
➢ Multiplexing-Demultiplexing

- Genellikle alınan bir mesajın kısımları arasında en ufak bir gecikmenin dahi tolere edilemeyeceği gerçek zamanlı uygulamalar (**özellikle multimedia streaming**) için uygundur.

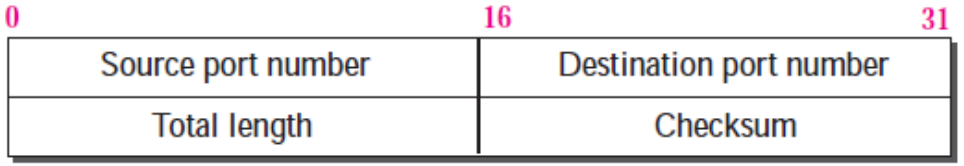
UDP Protokolü



Port	*Protocol*	*Description*
7	Echo	Echoes a received datagram back to the sender
9	Discard	Discards any datagram that is received
11	Users	Active users
13	Daytime	Returns the date and the time
17	Quote	Returns a quote of the day
19	Chargen	Returns a string of characters
53	Domain	Domain Name Service (DNS)
67	Bootps	Server port to download bootstrap information
68	Bootpc	Client port to download bootstrap information
69	TFTP	Trivial File Transfer Protocol
111	RPC	Remote Procedure Call
123	NTP	Network Time Protocol
161	SNMP	Simple Network Management Protocol
162	SNMP	Simple Network Management Protocol (trap)

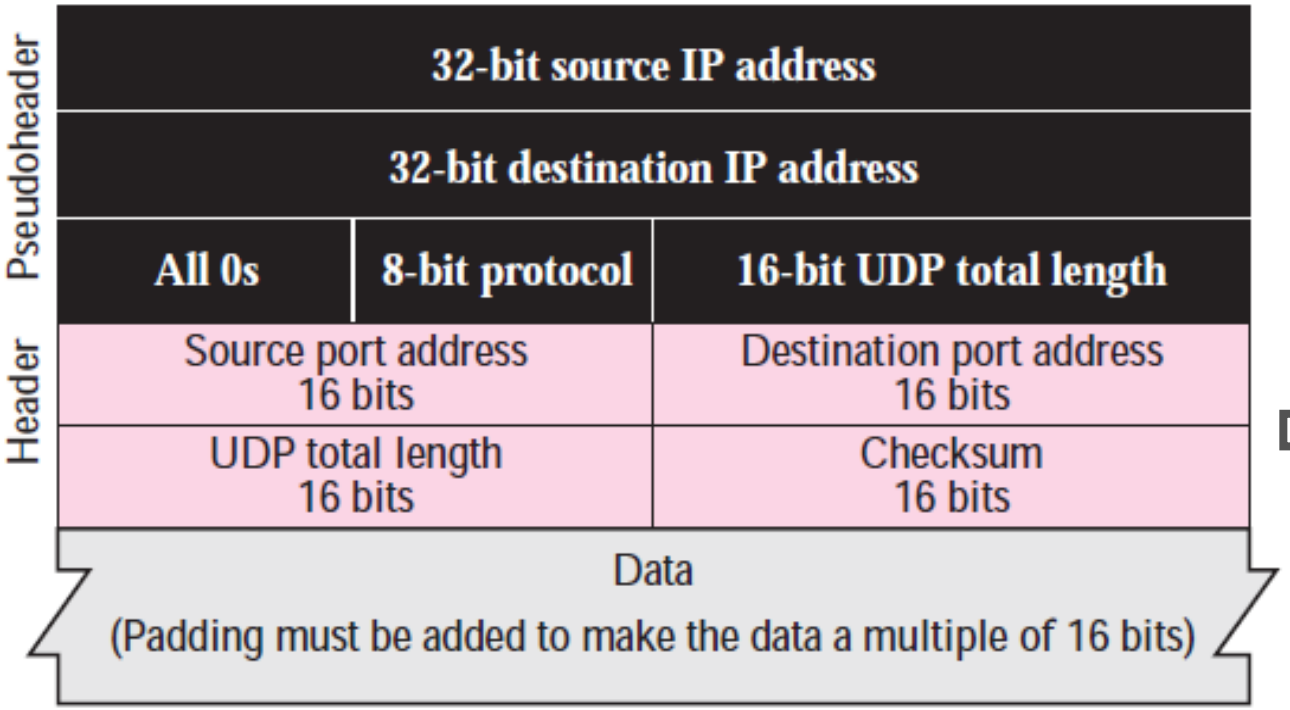


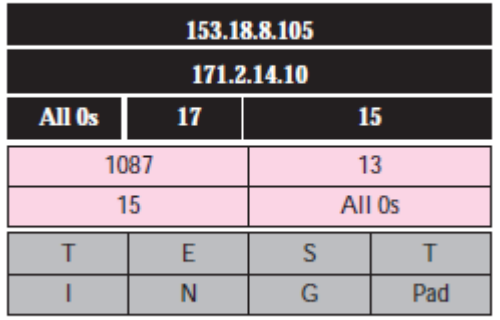
a. UDP user datagram

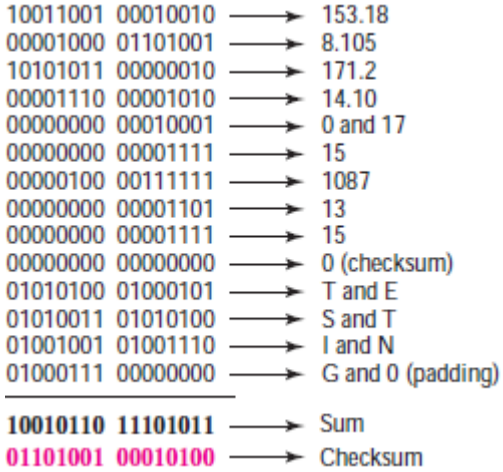


b. Header format

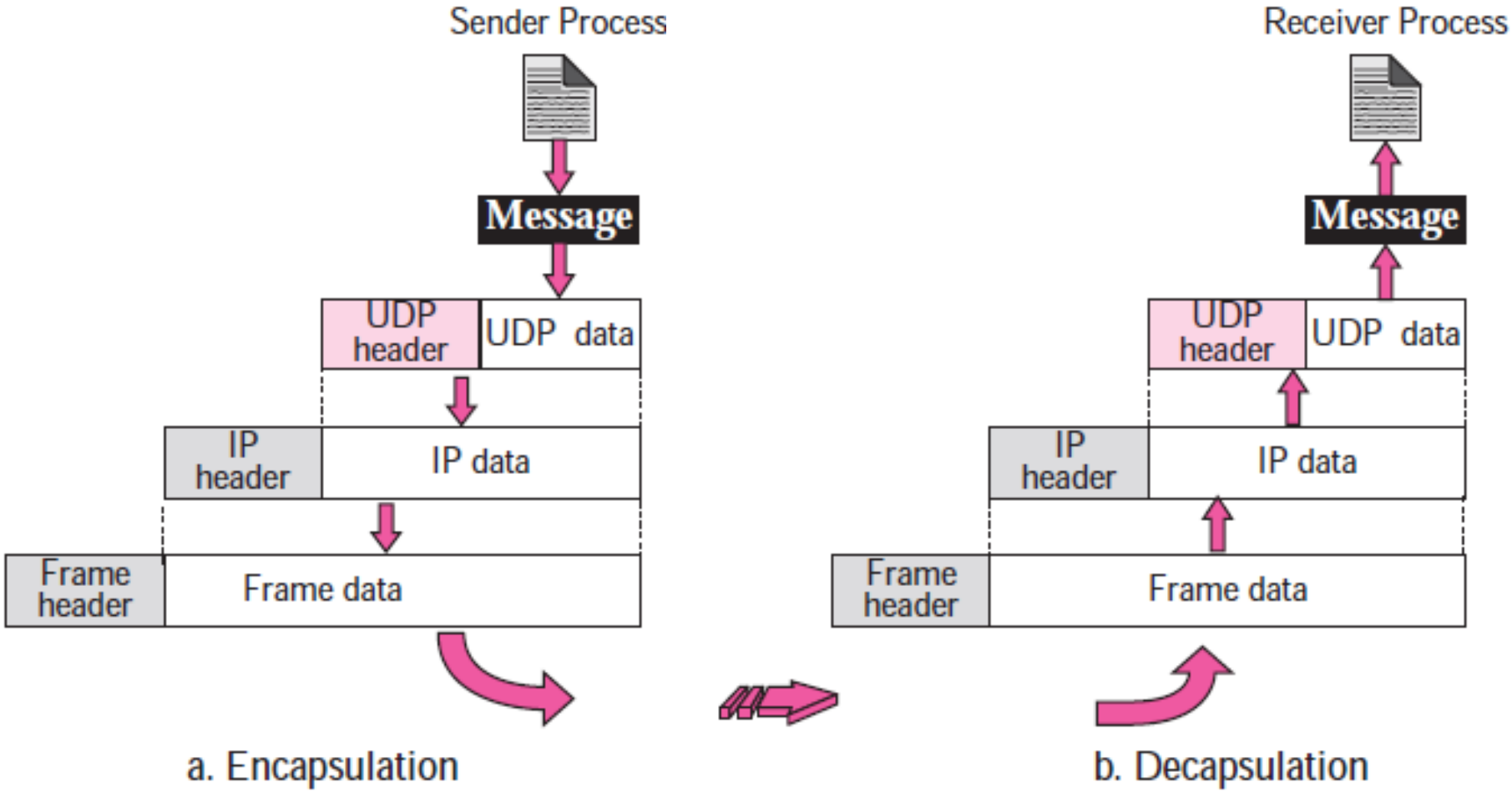
UDP Protokolü (Hata Kontrolü)



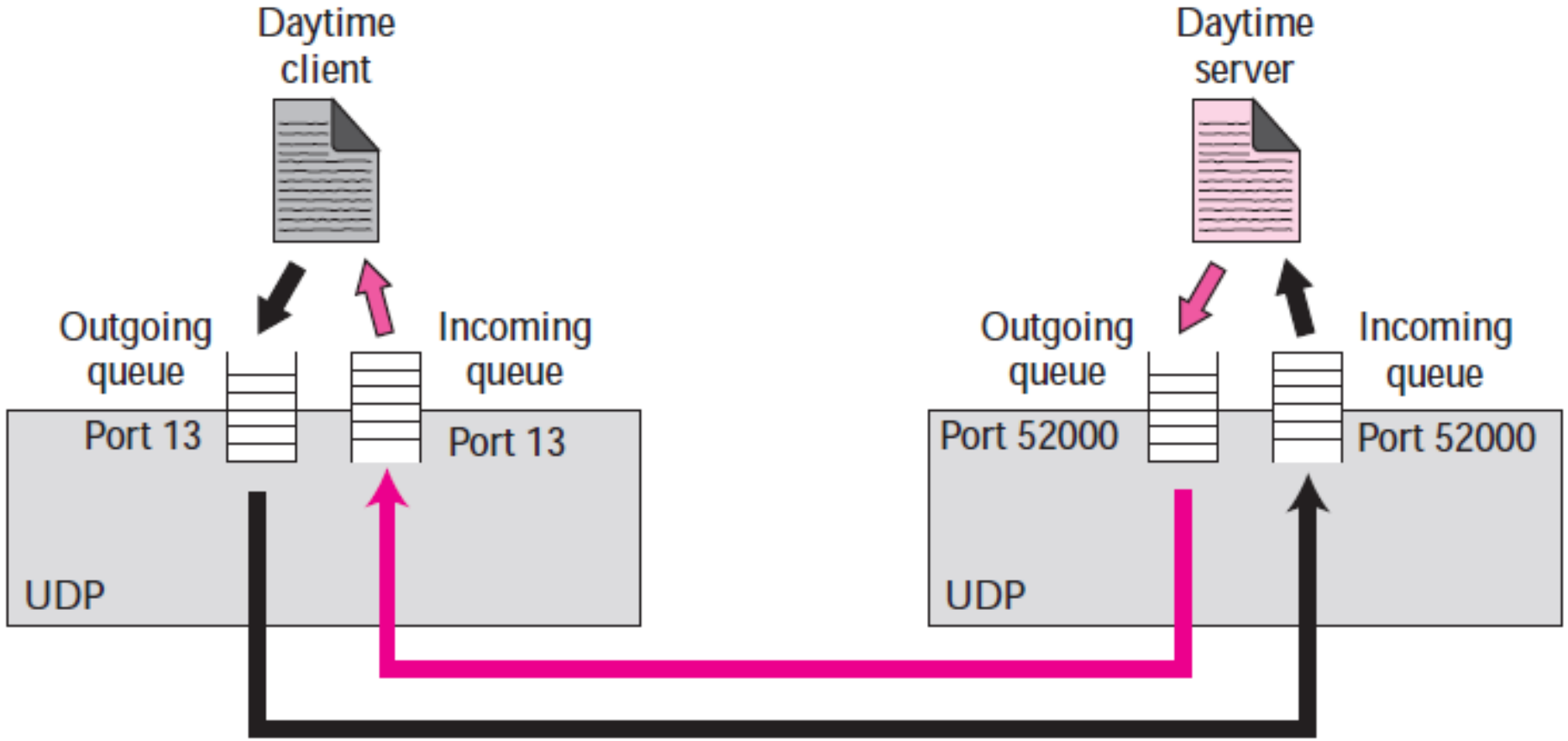




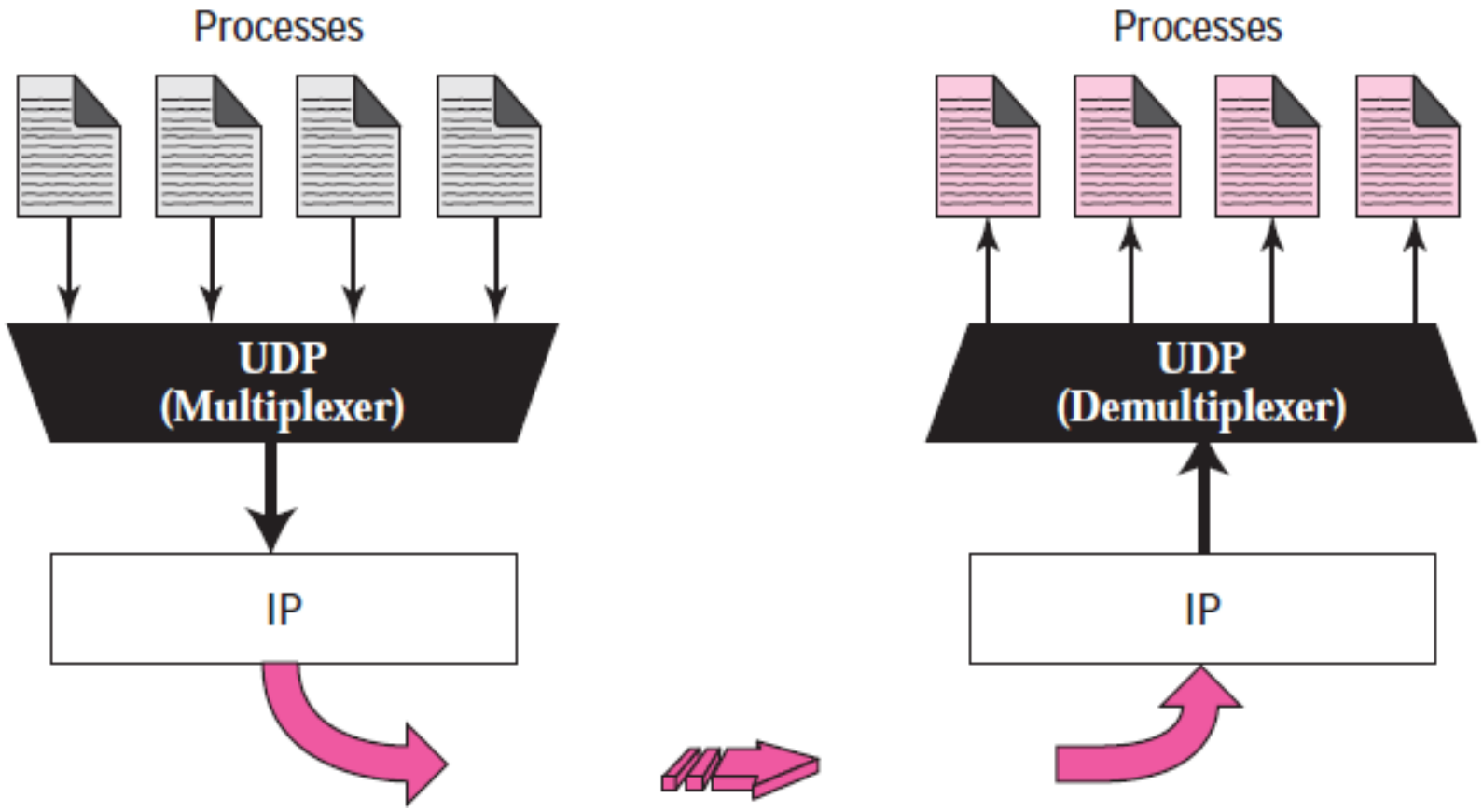
UDP Protokolü (Encapsulation-Decapsulation)



UDP Protokolü (Kuyruklama)



UDP Protokolü (Multiplexing-Demultiplexing)



UDP Protokolü (Çalışma Yapısı)