

Bilgisayar Ağları Computer Networks

Hazırlayan: M. Ali Akcayol
Gazi Üniversitesi
Bilgisayar Mühendisliği Bölümü

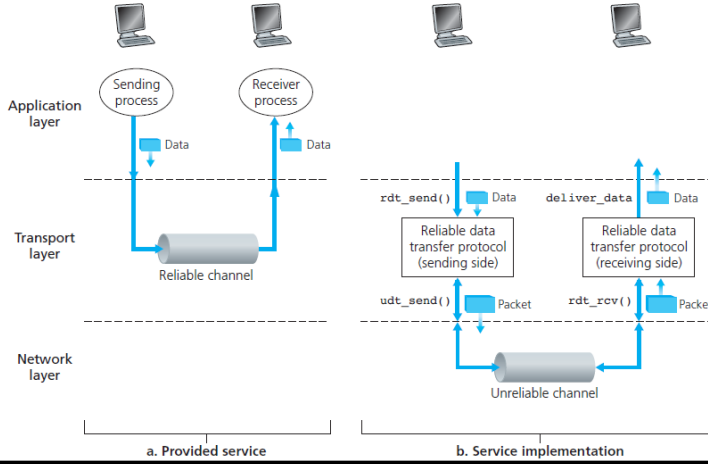
Bu dersin sunumları, "James Kurose, Keith Ross, Computer Networking: A Top-Down Approach 6/e, Pearson, 2013." kitabı kullanılarak hazırlanmıştır.

İçerik

- ▶ Reliable data transfer prensipleri
- ▶ Reliable data transfer protokolü
- ▶ Pipeline ile reliable data transfer protokolü
- ▶ Go-Back-N
- ▶ Selective repeat

Reliable data transfer prensipleri

- ▶ Güvenilir veri aktarımı **transport layer'da, network layer'da** veya **link layer'da oluşturulabilir.**
- ▶ Ağlarda en önemli 10 problem içerisinde yer almaktadır.
- ▶ **Her bit eksiksiz, doğru ve sıralı iletilir.**



3

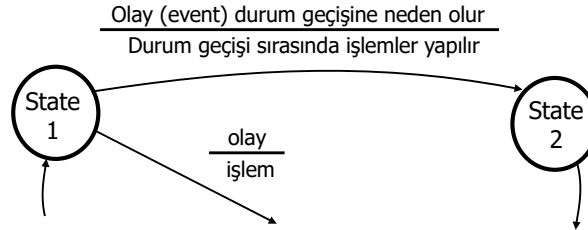
İçerik

- ▶ Reliable data transfer prensipleri
- ▶ **Reliable data transfer protokolü**
- ▶ Pipeline ile reliable data transfer protokolü
- ▶ Go-Back-N
- ▶ Selective repeat

4

Reliable data transfer protokolü

- ▶ Alt katmandaki protokol güvenilir değilse **reliable veri aktarımı oldukça zordur.**
- ▶ Alt katman güvenilir değilse bit bozulur, paket kaybolabilir.
- ▶ Üst katmanda **güvenilir veri aktarımı için çok sayıda mekanizma gerekir** (feedback, timer, sequence number, ...).
- ▶ Protokol **sonlu durum makinesiyle** (finite state machine) **modellenebilir.**

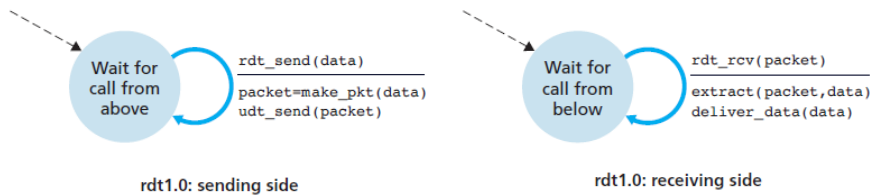


5

Reliable data transfer protokolü

rdt 1.0

- ▶ Alt katmandaki kanal tamamen güvenilirdir.
 - ▶ **Bit hatası yoktur.**
 - ▶ **Kayıp paket yoktur.**
- ▶ Gönderici ve alıcı FSM ile aşağıdaki gibi modellenebilir.



- ▶ Gönderici üst katmandan gelen veriyi pakete dönüştürüp alt katmandaki kanala gönderir.
- ▶ Alıcı alt katmandan aldığı paketten veriyi çıkarıp üst katmana gönderir.

6

Reliable data transfer protokolü

rdt 2.0

- ▶ Alt katmandaki kanalda **paket içindeki bitler bozulabilir.**
- ▶ **Kayıp paket yoktur** alıcıda paketler sıralı alınır.
- ▶ Bozulan paketler alıcı tarafından tekrar istenir (**Automatic Repeat reQuest (ARQ)** protokolleri).
- ▶ ARQ protokolleri için **üç bileşen gerekir:**
 - ▶ Error detection (checksum, CRC)
 - ▶ Receiver feedback (ACK, NAK)
 - ▶ Retransmission
- ▶ **Acknowledgment (ACK)**, paketin hatasız alındığını gösterir.
- ▶ **Negative ACK (NAK)**, paketin bozulduğunu gösterir.

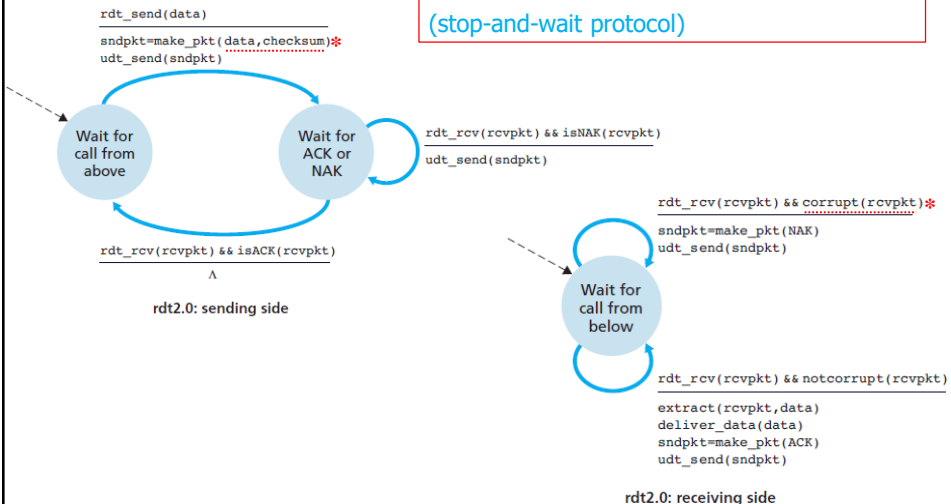
7

Reliable data transfer protokolü

rdt 2.0

- ▶ Hata algılama eklendi.

Bir paket doğrulanmadan sonrakine geçilmez.
(stop-and-wait protocol)



Reliable data transfer protokolü

rdt 2.0

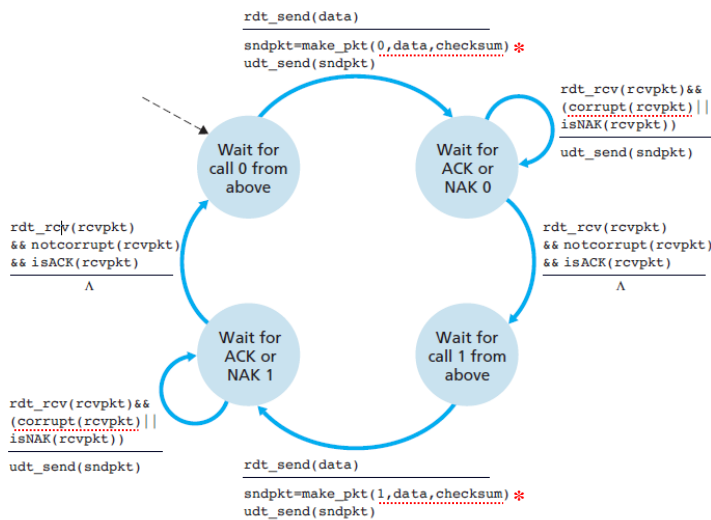
- ▶ rdt 2.0'da **ACK/NAK bozulursa?**
 - ▶ Gönderici, alıcı tarafından doğru alınıp alınmadığını bilemez.
 - ▶ **Retransmit yapılırsa duplicate olabilir.**
- ▶ Duplicate olan paketlerin belirlenmesi için:
 - ▶ Göndericide **her pakete sıra numarası eklenir.**
 - ▶ Pakete yeni bir alan eklenir (sequence number).
 - ▶ Gönderici **ACK/NAK bozulursa retransmit yapar.**
 - ▶ Alıcıda aynı sıra numarasına sahip tekrarlı **gelen paketler atılır.**

9

Reliable data transfer protokolü

rdt 2.1

- ▶ Sıra numarası alanı eklendi.

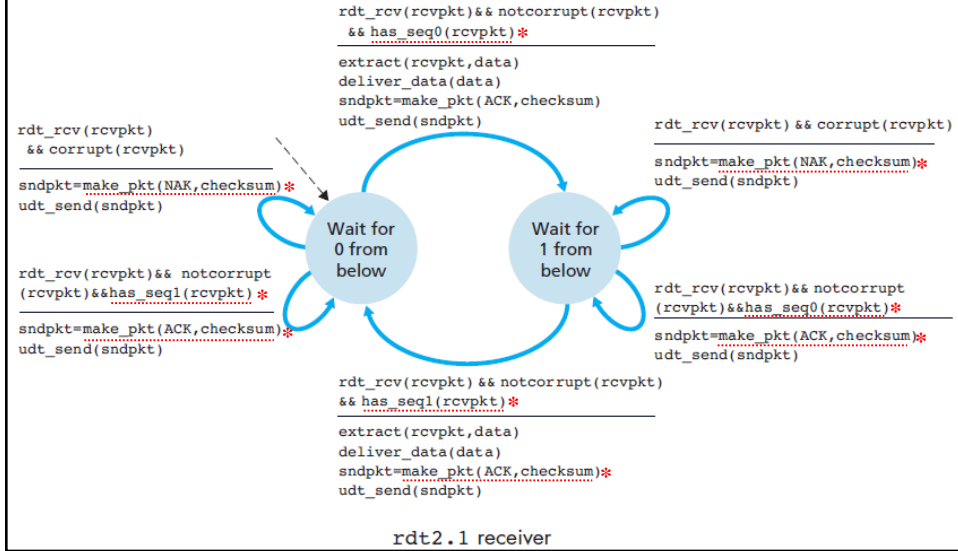


10

Reliable data transfer protokolü

rdt 2.1

- Sıra numarası alanı eklendi.



Reliable data transfer protokolü

rdt 2.1

- Gönderici:
 - Her pakete sıra numarası eklenir.
 - Minimum bir bit ile (0, 1) sıra numarası verilebilir ([alternating-bit protocol](#)).
 - ACK/NAK bozuk gelirse **retransmit yapılır**.
- Alıcı:
 - Gelen **paket çift mi kontrol edilir**.
 - Bulunulan durum beklenen sıra numarasını gösterir.
 - **Çift gelen paketler atılır**.
 - **Beklenmeyen** sıra numarasına sahip **paket gelirse ACK gönderilir**.

Reliable data transfer protokolü

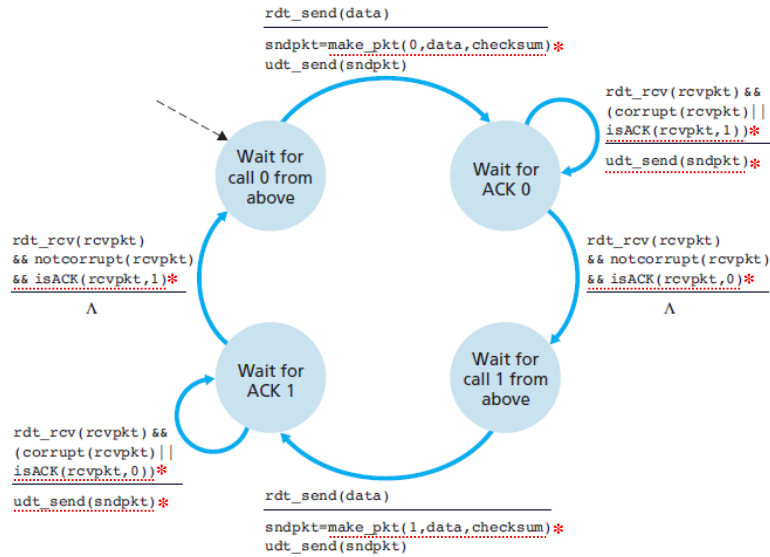
rdt 2.2

- ▶ NAK paketi yerine **ACK paketi ile negatif geri bildirim yapılır.**
- ▶ Alıcı:
 - ▶ ACK paketine sıra numarası eklenir.
 - ▶ **Bozuk alınan paket için NAK yerine, en son doğru alınan için tekrar ACK gönderilir.**
- ▶ Gönderici:
 - ▶ Aynı sıra numarası için **ACK tekrar gelirse retransmit yapar.**
 - ▶ **Beklediği** sıra numarası için **ACK gelirse** sonraki sıra numarasına geçer.

13

Reliable data transfer protokolü

rdt 2.2

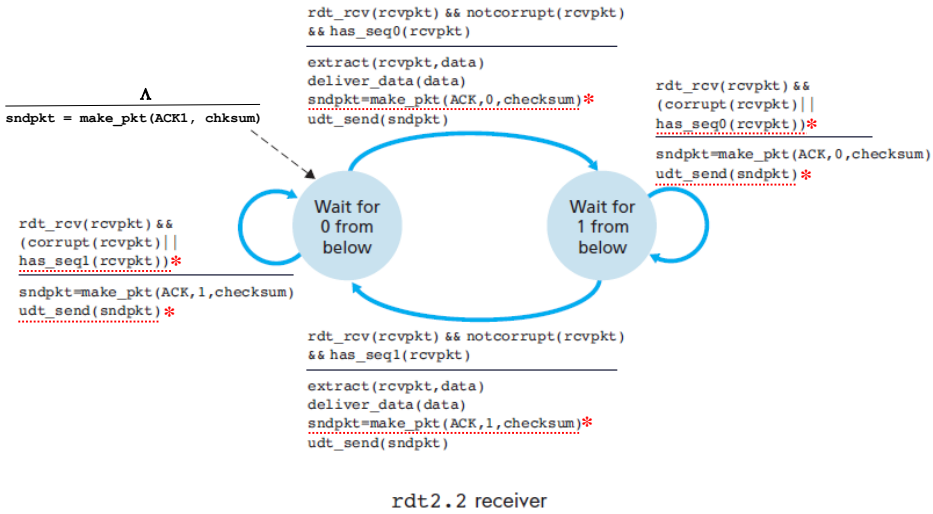


rdt2.2 sender

14

Reliable data transfer protokolü

rdt 2.2



15

Reliable data transfer protokolü

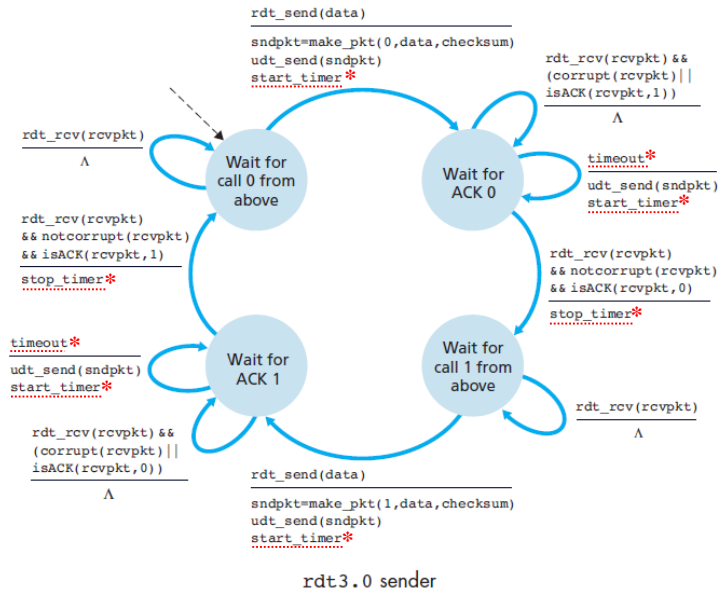
rdt 3.0

- ▶ **Alt kanalda paketler kaybolabilir** (İnternet'te karşılaşılan durum).
- ▶ **İki konunun belirlenmesi gereklidir:**
 - ▶ Kayıp paket nasıl anlaşılacak?
 - ▶ Kayıp olduğu anlaşılınca ne yapılacak?
- ▶ Paketin kaybolduğunun anlaşılması için **gönderici tarafa timer eklenir.**
- ▶ Bekleme süresi **aşağı doğru sayan timer** ile tutulur.
- ▶ **Gönderici** bir süre bekler **ACK gelmezse retransmit yapar.**
- ▶ ACK kaybolmamış da gecikmişse **retransmit edilen paket sıra numarası ile algılanır.**

16

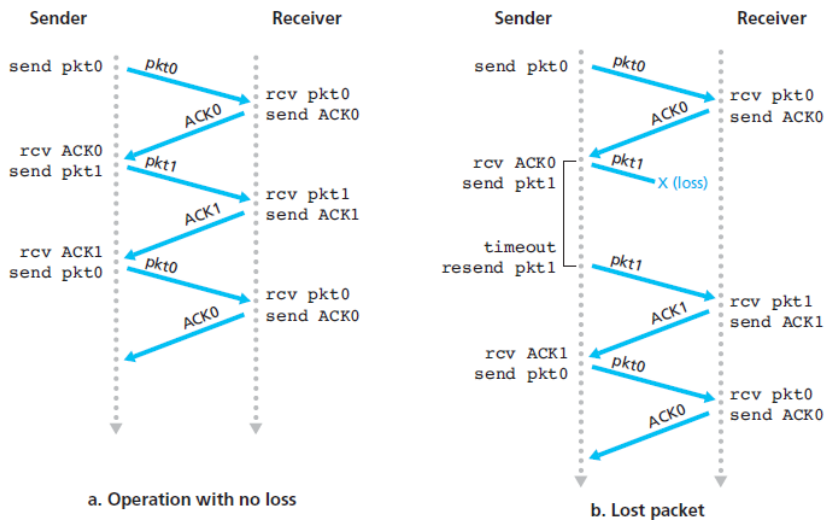
Reliable data transfer protokolü

rdt 3.0



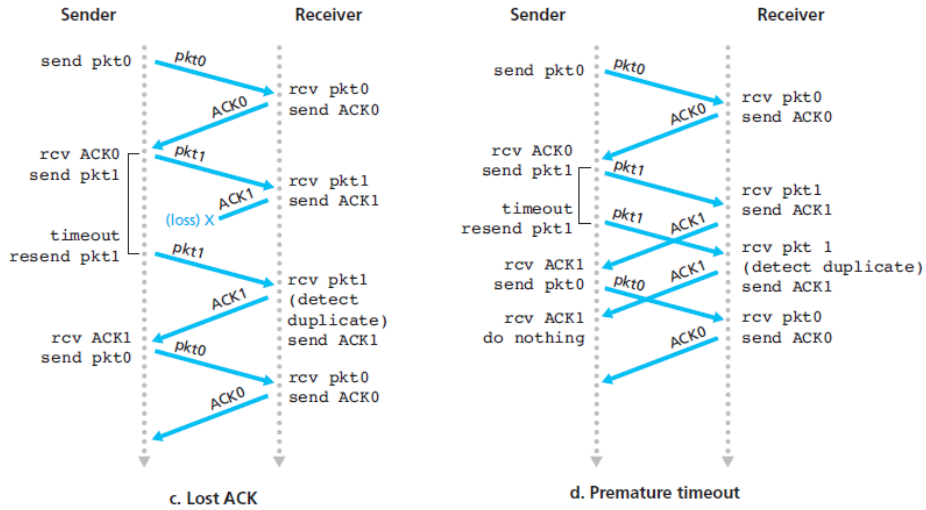
Reliable data transfer protokolü

rdt 3.0



Reliable data transfer protokolü

rdt 3.0



19

İçerik

- ▶ Reliable data transfer prensipleri
- ▶ Reliable data transfer protokolü
- ▶ Pipeline ile reliable data transfer protokolü
- ▶ Go-Back-N
- ▶ Selective repeat

20

Pipeline ile reliable data transfer protokolü

- ▶ rdt 3.0 protokolü **stop-and-wait çalışır** ve **performansı çok düşüktür**.
- ▶ İki uç nokta arasında RTT süresi 30ms, transmission rate 1Gbps ve paket boyutu 1kB olsun.
- ▶ **Paket iletim ortamına 8µs'de verilir.**

$$d_{trans} = \frac{L}{R} = \frac{8000 \text{ bits/packet}}{10^9 \text{ bits/sec}} = 8 \text{ microseconds}$$

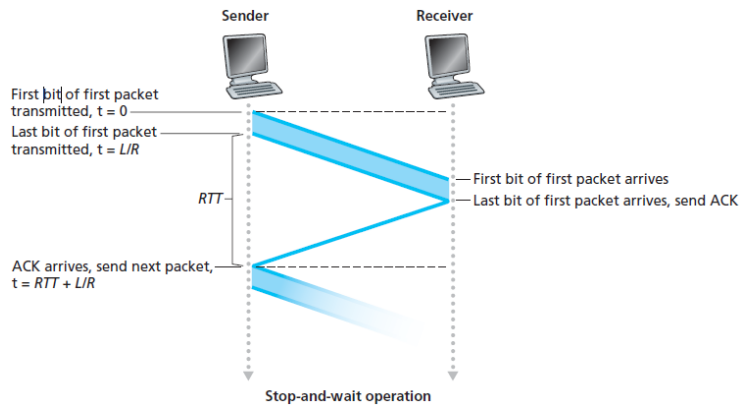
- ▶ Göndericinin kanalı **kullanım oranı %0,027 olur.**

$$U_{sender} = \frac{L/R}{RTT + L/R} = \frac{.008}{30.008} = 0.00027$$

- ▶ **Stop-and-wait yerine pipelining** yapılarak kullanım oranı artırılır.

21

Pipeline ile reliable data transfer protokolü

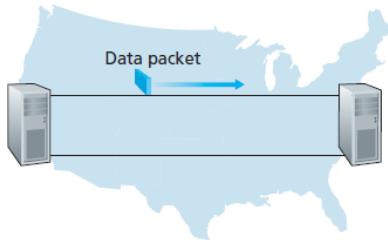


$$U_{sender} = \frac{L/R}{RTT + L/R} = \frac{.008}{30.008} = 0.00027$$

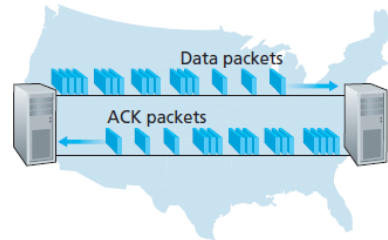
22

Pipeline ile reliable data transfer protokolü

- ▶ Pipelining ile **birden fazla paket ACK alınmadan art arda gönderilir.**
- ▶ Sıra numarası aralığı artırılmak zorundadır.
- ▶ Gönderici ve alıcıda **sirasız gelen paketler için buffer gereklidir.**
- ▶ İki temel pipelining protokolü vardır: [Go-Back-N](#) ve [Selective Repeat](#)



a. A stop-and-wait protocol in operation

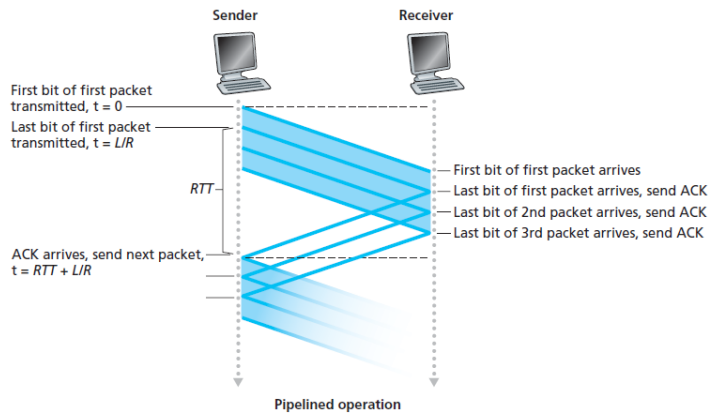


b. A pipelined protocol in operation

23

Pipeline ile reliable data transfer protokolü

- ▶ Göndericinin kanalı **kullanım oranı %0,08 olur** (3 kat arttı).



$$U_{\text{sender}} = \frac{3 \cdot L/R}{RTT + L/R} = 0.0008$$

24

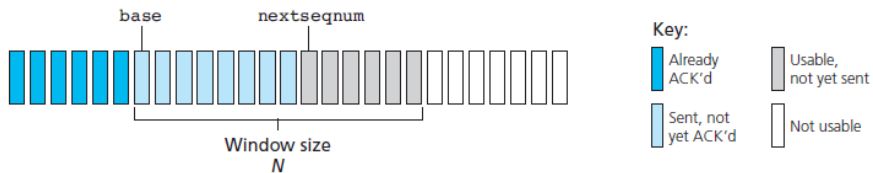
İçerik

- ▶ Reliable data transfer prensipleri
- ▶ Reliable data transfer protokolü
- ▶ Pipeline ile reliable data transfer protokolü
- ▶ Go-Back-N
- ▶ Selective repeat

25

Go-Back-N

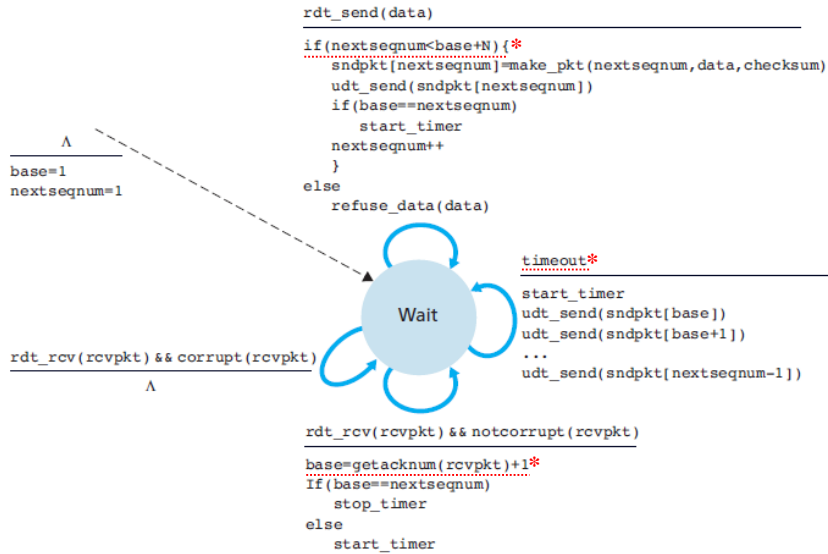
- ▶ ACK alınmadan N paket (pencere boyutu) pipeline'a gönderilir.
- ▶ Paket başlığına **k-bit sıra numarası** eklenir.
- ▶ **ACK(n)**: n.paket ve öncekilerin tamamı içindir (**cumulative acknowledgement**).
- ▶ Her **ACK geldiğinde pencere ileri kaydırılır** (**sliding window protocol**).
- ▶ Bir paket için **ACK beklenmeye başlayınca** timer başlatılır.
- ▶ **timeout(n)**: n.paket için timeout olursa, n ile ardından **gönderilenlerin tamamı retransmit yapılır** (Go-Back-N).



Sender's view of sequence numbers in Go-Back-N

26

Go-Back-N

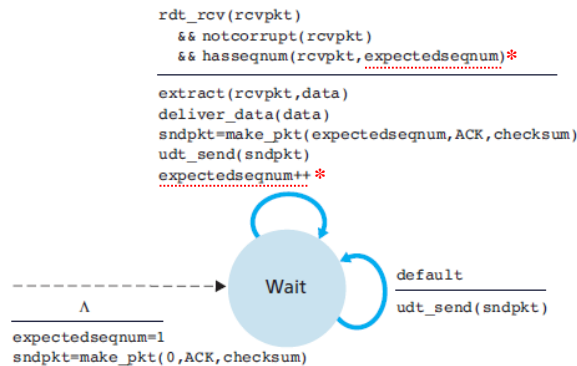


Extended FSM description of GBN sender

27

Go-Back-N

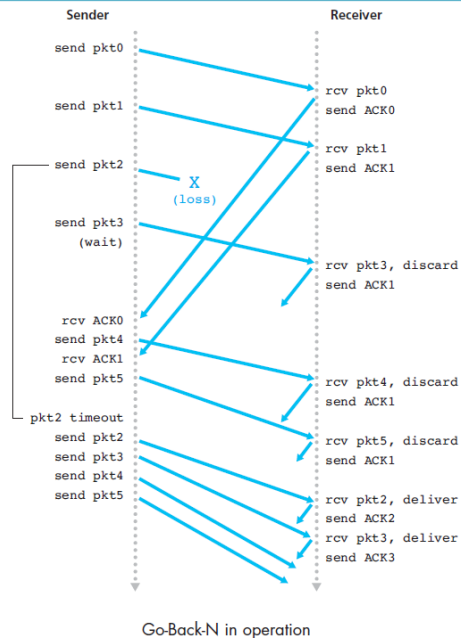
- ▶ **Beklenen sıra numarasına sahip paket gelirse ACK gönderilir.**
- ▶ **expectedseqnum** beklenen sıra numarasını tutar.
- ▶ **Sirasız gelen paketler atılır** ve en son doğru alınan paket için ACK retransmit edilir.



Extended FSM description of GBN receiver

28

Go-Back-N



Go-Back-N in operation

29

İçerik

- ▶ Reliable data transfer prensipleri
- ▶ Reliable data transfer protokolü
- ▶ Pipeline ile reliable data transfer protokolü
- ▶ Go-Back-N
- ▶ Selective repeat

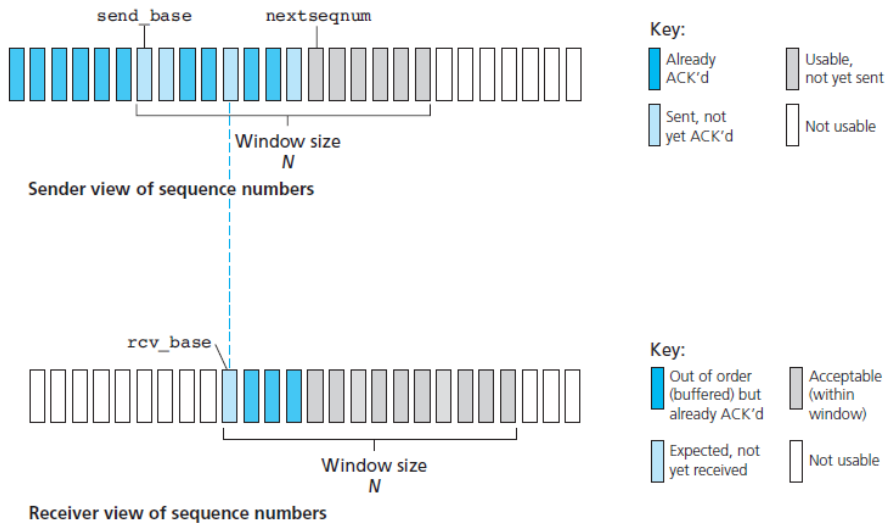
30

Selective repeat

- ▶ Go-Back-N protokolünde **bir paket hatası çok sayıda paketin retransmit edilmesine yol açar.**
- ▶ Selective-repeat protokolünde **gönderici sadece hatalı paketleri retransmit eder.**
- ▶ **Alıcı doğru aldığı her paket için ayrı ACK gönderir.**
- ▶ Alıcıda **sırasız gelen paketler buffer'a alınır.**
- ▶ Üst katmana sıralı bloklar halinde gönderilir.
- ▶ **Gönderici timer'ı her paket için gönderildiği anda başlatılır.**
- ▶ Gönderici penceresi N adet paketi ACK almadan göndermeye izin verir.

31

Selective repeat



32

Selective repeat

Gönderici olayları ve yapılan işlemler:

- ▶ Üst katmandan veri alındı
 - ▶ Gönderici sonraki sıra numarası kullanılabilir mi diye kontrol eder.
 - ▶ Sonraki sıra numarası kullanılabilir ise paket gönderilir, değilse buffer'a alınır veya üst katmana iade edilir.
- ▶ Timeout
 - ▶ Her paket kendi timer'ına sahiptir.
 - ▶ Timeout olan paket retransmit edilir.
- ▶ ACK alındı
 - ▶ ACK alınan paket işaretlenir.
 - ▶ ACK sıra numarası == **send_base** ise, pencere ACK alınmayan ilk pakete kadar kaydırılır.
 - ▶ Pencere kaydırılırsa yeni numaralar kullanılabilir hale gelir.

33

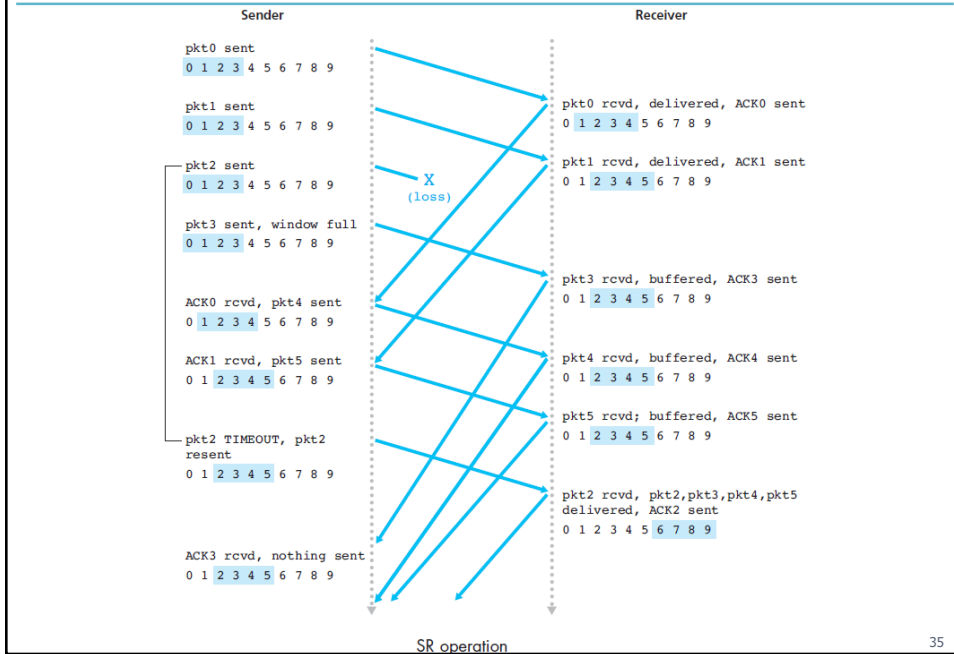
Selective repeat

Alıcı olayları ve yapılan işlemler:

- ▶ [**rcv_base**, **rcv_base+N-1**] arasında bir paket doğru alındı
 - ▶ Gelen paket için ACK gönderilir.
 - ▶ Gelen paket numarası == **rcv_base** ise ilk gelmeyen sıra numarasına kadar ardışık numaralı paketler üst katmana gönderilir ve pencere ilk beklenen sıra numarasına kadar kaydırılır
- ▶ [**rcv_base-N**, **rcv_base-1**] arasında bir paket doğru alındı
 - ▶ Gelen paket için ACK gönderilir.
- ▶ Diğer durumlar
 - ▶ Paket atılır.

34

Selective repeat



Selective repeat

