

Bölüm 12: UDP ve TCP

Türkçe (İngilizce) karşılıklar

Bağlantısız (connectionless)
Connection-oriented (bağlantı temelli)
Veri dizisi (data stream)
Soket (socket)
Alındı (acknowledgment)
Üç yollu el sıkışma (three-way handshake)

Bölüm Hedefi

Bu bölümü bitirdiğinizde

- UDP nedir?
- TCP nedir?
- TCP bağlantısının kurulmasını
- TCP’de aktarımın gerçekleşmesini
- Ağdaki yoğun trafiğin TCP bağlantıları üzerindeki etkisini öğrenmiş olacaksınız.

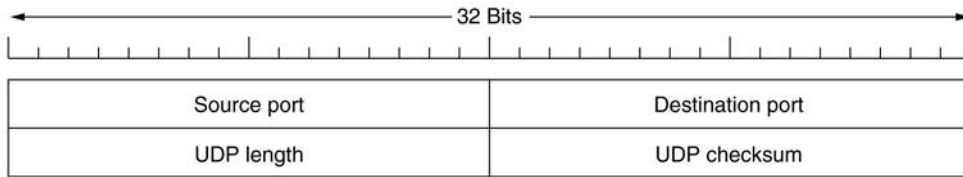
IP aynı bilgisayar üzerinde çalışan haberleşme programları arasında ayrım yapamaz. Çünkü paket yapısına baktığınızda sadece kaynak ve varış adreslerinin kullanıldığını görürsünüz. Ancak gerçekte, iki farklı bilgisayar üzerinde (A ve B diyelim) birbiri ile haberleşen pek çok program çalışabilir. Bu programlar ulaşım (transport) katmanında kullanılan protokoller yardımı ile birbirinden ayrılır.

IP, ulaşım (transport) katmanında iki protocol kullanır. Bu protokoller UDP (User Datagram Protocol) ve TCP (Transmission Control Protocol)’dir. Bunlardan UDP bağlantısız (connectionless) , TCP ise bağlantı temelli protokoldür.

12.1 UDP (User Datagram Protocol)

UDP, uygulamaların veri aktarımında kullandığı protokollerden biridir. Yaratılan UDP paketleri IP tarafından kapsülendir ve IP ortamında taşınır. UDP paketlerini taşımak için önceden bir bağlantı kurulmasına gerek duyulmaz.

UDP paket yapısı Şekil 12.1’de gösterildiği gibi basit bir yapıdır. Başlık 8 sekizliden oluşur ve bunu veri alanı takip eder. UDP paketlerine *bölüt (segment)* de denir. Kaynak ve varış kapıları IP datagramlarında adresleri verilen düğümlerin hangi kapılar üzerinden bu haberleşmeyi gerçekleştireceğini gösterir.



Kaynak kapısı (Source port)

Varış kapısı (destination port)

UDP uzunluğu (UDP length)

UDP hata sınaması (UDP checksum)

Şekil 12.1 UDP başlığı

UDP uzunluğu, başlık ve veri alanının uzunluğunu bildirir. UDP’de hata sınaması kullanılmayabilir. UDP oldukça basit bir protokoldür. Kaybolan verinin tekrar gönderilmesi, akış kontrolü gibi konularla ilgilenmez. IP’nin kullanımı için bir arayüz oluşturur. Basitliği nedeniyle gerçek zamanlı uygulamalarda tercih edilir.

Kısa istek ve cevapların taşındığı uygulamalarda da UDP kullanabilir. UDP bağlantılarının kurulması (*setup*) gerekmediği gibi sonlandırılması (*release*) da gerekmez. Kaybolan UDP bölütleri için birşey yapılmaz.

12.2 TCP (Transmission Control Protocol)

Bağlantı temelli bir protokol olan TCP’de kullanıcı verisi taşınmadan önce bağlantı kurulması gerekir. Protokol, ulaşım katmanında uçtan-uca güvenilir bir ortam yaratmayı amaçlar. Ortamın güvenilirliği, kaybolan paketlerin tekrar gönderilmesini temel alır.

TCP, kullanıcıdan gelen veri dizisini (data stream), 64 KB’yi geçmeyecek (genelde 1460 Byte uzunluğunda) veri bloklarına ayırır. TCP başlıklarını da ekledikten sonra IP katmanına gönderir. TCP paketleri IP tarafından kapsüllenerek taşınır. IP verinin güvenilir bir şekilde taşınacağı konusunda bir garanti vermez. Kayıp paketlerin farkedilmesi ve tekrar gönderilmesi TCP’nin görevidir.

TCP servisleri kaynak ve varış düğümleri üzerindeki soketler aracılığı ile verilir. Her soketin, bulunduğu düğümün IP numarasından ve o düğüme ait bir kapı (port) numarasından oluşan soket numarası vardır. Bu kapılar *TSAP* (*Transport Service Access Point*) olarak da adlandırılır. TCP bağlantıları, her iki uç düğüm üzerindeki soket numaraları ile tanımlanır. (*soket1*, *soket2*) gibi.

1024’ten küçük kapı numaralarının özel işlevleri vardır. Bunların bazıları aşağıda gösterilmiştir. TCP ve UDP kapı numaraları ve işlevleri hakkında daha uzun bir listeye www.iana.org/assignments/port-numbers sayfasından erişilebilir.

Kapı no	Protokol	İşlev
20	FTP	Dosya aktarımı-veri
21	FTP	Dosya aktarımı-kontrol
23	Telnet	Uzaktan bağlantı
25	SMTP	E-posta
80	HTTP	World Wide Web
110	POP-3	Uzaktan e-posta erişimi

Tüm TCP bağlantıları *çift yönlüdür* (*full duplex*).

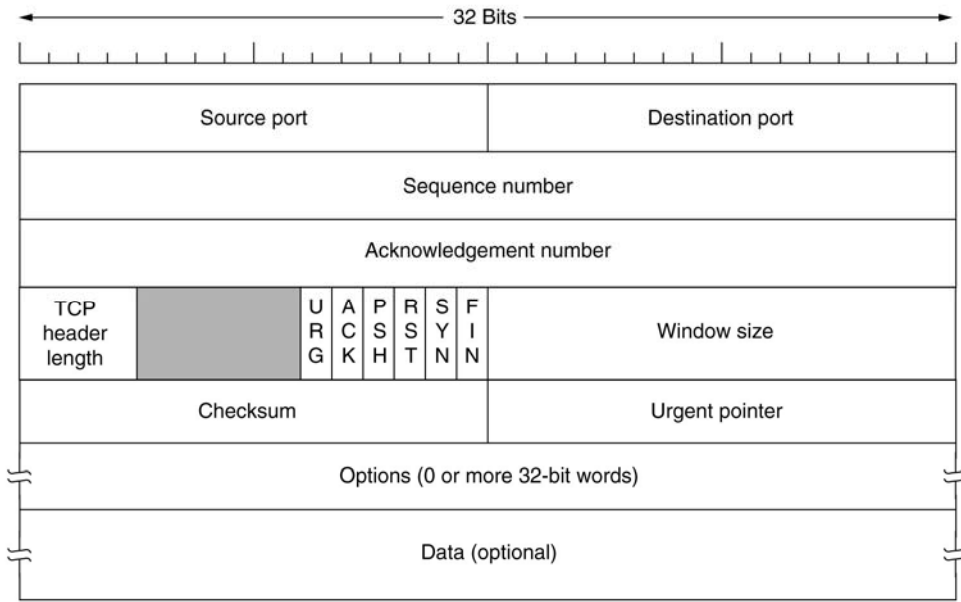
TCP birimleri tarafından yaratılan TCP paketlerine de *bölüt* (*segment*) denir. Bir TCP bölütü, Şekil 12.2’de gösterildiği gibi, 20 sekizliden oluşan sabit başlık, seçenek alanı ve veriden oluşur. Doğal olarak TCP bölütleri IP datagramlarına sığacak boyutta olmalıdır (en fazla 65 KB uzunluğunda). Ayrıca MTU da bölüt boyunun belirlenmesinde önem taşır. Aksi halde IP paketlerinin parçalanması gerekir.

12.2.1 TCP’de Güvenilir Aktarım

TCP’de aktarımın güvenilirliği aşağıdaki şekilde sağlanır:

- TCP bir bölüt gönderdiğinde bir sayaç kurar ve sayaç durmadan varış düğümünden gönderilen bölüt için bir *alındı* (*acknowledgment*) gelmesini bekler. Alındı zamanında gelmezse bölüt tekrar gönderilir.
- Varış ucundaki TCP birimi veri aldığı anda, bir alındı gönderir.
- TCP başlık ve veri üzerinde hata sınaması yapar. Hata sınamasından geçemeyen bir bölüt için alındı gönderilmez.
- TCP bölütleri IP tarafından kapsüllenerek taşındıkları için, parçalanma olasılıkları vardır. Parçalar varış noktasında birleştirilir.
- TCP varış noktasındaki bellek miktarı konusunda kaynak düğüme bilgi vererek, akış kontrolü de yapar. Burada amaç hızlı bir kaynağın, varış düğümünün kapasitesinin üzerinde veri göndermesini önlemektir.

12.2.2 TCP Başlık Yapısı



Kaynak kapısı (source port)
Varış kapısı (destination port)
Sıra numarası (sequence number)
Alındı numarası (acknowledgment number)
TCP başlık uzunluğu (TCP header length)
Pencere boyu (window size)
Hata sınama (checksum)
Acil işaretçi (Urgent pointer)
Seçenekler (options)
Veri (data)

Şekil 12.2 TCP başlığı

Kaynak ve varış kapılarının işlevi UDP'deki gibidir. TCP'de taşınan her sekizlinin bir sıra numarası vardır. Bağlantı kurulduğunda belirlenen sıra numarası bağlantı boyunca taşınan her sekizli için arttırılarak kullanılır. Varış düğümüne ulaşan veri için alındı (acknowledgment) mesajı gönderilir. Alındı mesajında en son alınan sekizlinin numarası (alındı numarası) yerine, beklenen (bir sonraki) sekizlinin numarası yazılır. Ayrıca bu numaranın bir alındıyı gösterdiğini bildirmek için ACK bitinin de 1 olması gerekir.

TCP başlık uzunluğu, başlığın uzunluğunu 32-bit cinsinden verir. Seçenekler nedeniyle başlık 20 sekizliden daha uzun olabilir.

URG alanının değeri 1 ise, bu durum acil işaretçisinin aktif olduğunu gösterir ve sıra numarasına eklenen acil işaretçisi değeri ile acil verinin (urgent data) yeri bulunabilir. Bu özellik kesmelerin (interrupts) yerinin bulunmasını sağlar. PSH biti verinin hemen bir üst katmana geçirilmesini bildirir.

RST bağlantının yeniden başlatılmasını sağlar. Düğümlerden birinde sorun çıkması durumunda gerekebilir.

SYN biti bir bağlantı kurulurken kullanılır. ACK bitinin 1 olması başlıktaki alındı numarasının geçerli olduğunu gösterir. SYN=1 and ACK=0 bir bağlantı kurulurken kullanılır ve bağlantı isteği (Connection Request) anlamına gelir. SYN=1 ve ACK=1, bağlantının kabul edildiğini (Connection Accepted) gösterir.

FIN biti bağlantının bittiğini gösterir. Hem SYN hem de FIN bölütlerinin sıra numarası vardır.

Pencere boyu, varış düğümüne gönderilebilecek sekizli (byte) sayısını bildirir. Gönderilmeye başlanacak sekizlinin numarası alındı numarası alanında belirtilmiştir. Eğer gelen bölütteki pencere boyu sıfır ise, bu alıcının (varışın) bellek alanının dolduğunu ve kaynağın beklemesi gerektiğini gösterir. Bu durumda, alıcı belleğinde yer açılınca aynı alındı numarası ve bu kez sıfır olmayan bir pencere boyu ile tekrar bir bölüt gönderir.

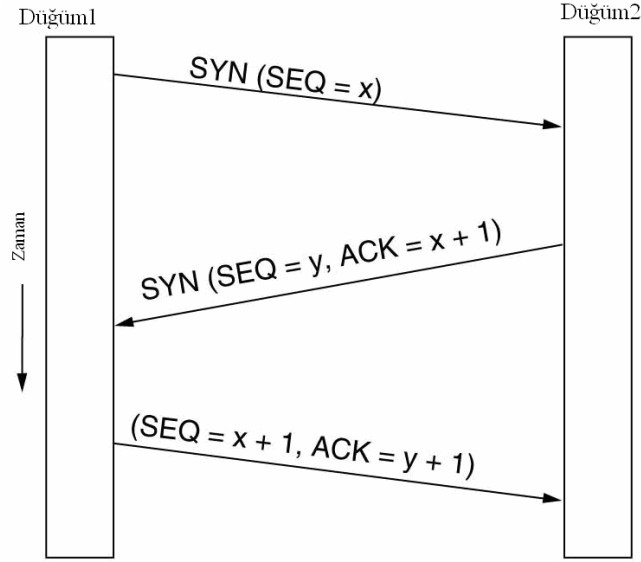
Hata sinama alanı veri ve başlık üzerinde basit bir hata kontrol mekanizması oluşturmak için kullanılır.

Seçenek alanları, ağ teknolojilerindeki gelişmeleri TCP'ye uyarlamamıza yardımcı olur. Örneğin, gelişen teknoloji ile varış noktalarında daha büyük pencereler kullanılabilir. Bu durumda, pencere boyutunu belirlemek için farklı ölçekler kullanılabilir. Bu da *pencere ölçek seçeneği (window scale option)* ile ifade edilir.

Klasik TCP protokolünde bir bölüt için alındı gelmezse ondan sonraki tüm bölütler gönderilir. Ancak yapılan geliştirmelerle seçenek alanı kullanarak sadece belli bir bölütün tekrar gönderilmesi istenebilir. Bu durum *negatif alındı (negative acknowledgment, NACK)* olarak adlandırılır.

12.2.3 Bir TCP Bağlantısının Kurulması

TCP'de bağlantılar *üç yollu el sıkışma (three-way handshake)* denen şekilde kurulur. Şekil 12.3'te bu yöntemin işleyişi gösterilmiştir. Bu durumda kaynak düğümü SYN bitini 1 yaparak bağlantı kurma isteğini iletir. Bu bölütün alındısını ve SYN değerini 1 olarak göndererek varış düğümü isteği onayladığını bildirir ve kaynaktan veri iletimi başlar. SYN bölütü sıra numaralarından birini kullanır.



Şekil 12.3 Üç yollu el sıkışma ile TCP bağlantısı kurulması

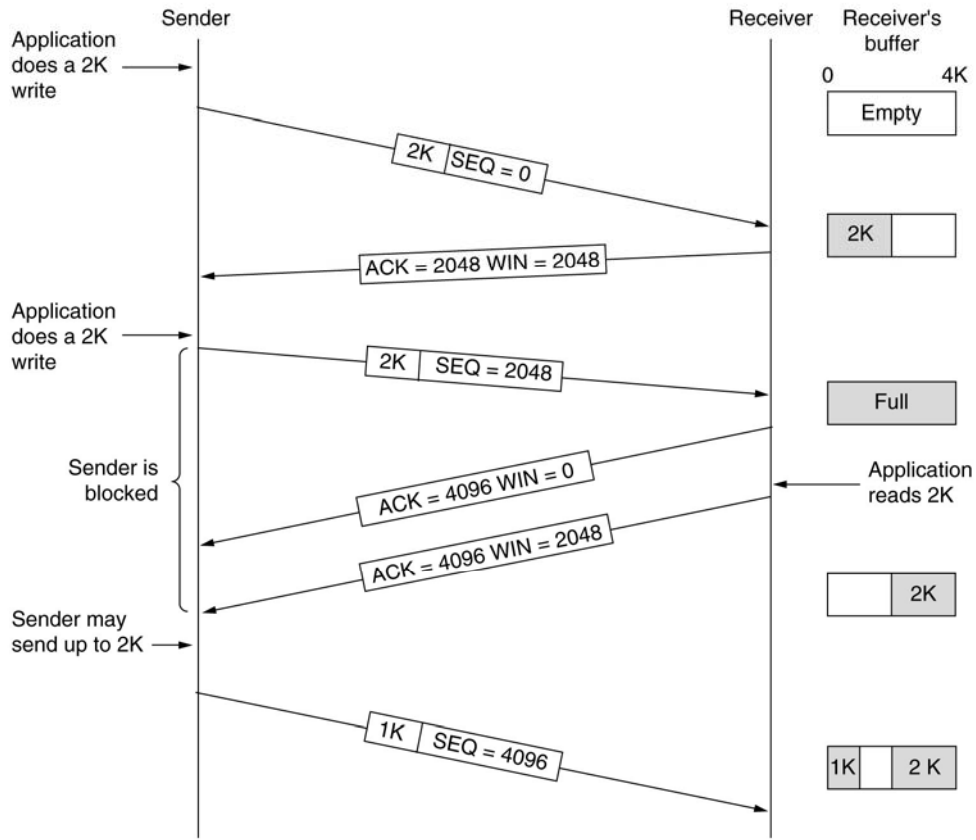
Bir bağlantının sonlandırılması için bölüm başlığındaki FIN bitine 1 değeri verilir. Bu bölüte gelen alındı bağlantının iki düğüm (kaynak ve varış) tarafından da sonlandırıldığını gösterir.

12.2.3 TCP’de Pencere Yönetimi

TCP’de veri aktarımı belli bir akış kontrolü içinde gerçekleşir Burada amaç: kaynak ve varış düğümlerini, hız açısından, uyum içinde çalıştırmaktır. Bu uyum, hızlı bir kaynak düğümünün varış düğümünün belleğinde yer olmaması durumunda veri aktarımına ara vermesi şeklinde düşünülebilir. Bu işlem, varış düğümünün boş bellek kapasitesini (standartta pencere boyu olarak geçmektedir) kaynak düğümüne bildirmesiyle gerçekleşir. Şekil 12.4’te bu duruma ait bir örnek verilmiştir.

Şekilde gönderilen her bölüm için gelen alındının varış düğümünde kalan kapasiteyi de gösterdiğini görmekteyiz. Varıştaki bellek alanı tamamen dolduğunda pencere boyu olarak 0 değeri gönderilir. Bu, kaynağın yeni bölüm göndermemesi anlamına gelir. Varış düğümünde tekrar yer açıldığında bir önceki bölümün alındısı tekrarlanır. Ancak bu kez bellekte açılan kapasite de kaynağa bildirilir.

Ağdaki yoğun trafiğe bağlı olarak bazen IP datagramları yönlendiricilerde yok edilebilir. Yok edilen bir bölüm için alındı gelmesi mümkün değildir. Kaynağın gereksiz yere beklemesini önlemek için, gönderilen bölüm sonrasında kaynak bir *zaman sayacı (retransmission timer)* kurar. Zaman sayacı geri sayma işlemini bitirdiğinde ilgili bölüm için alındı mesajı gelmediyse, o bölümün kaybolduğu düşünülür ve tekrar gönderilir. Üzerinde çalışılan konulardan biri, zaman sayacının alacağı değerin hesaplanmasıdır. Bu değer kaynak ve varış arasındaki *gidiş-dönüş süresine* yakın olmalıdır. Verilecek daha büyük değerler boş yere beklemeye neden olacaktır. Daha küçük değerler ise kaybolmamış bölümlerin bile kaybolmuş gibi algılanmasına neden olacaktır.



Şekil 12.4 TCP’de pencere yönetimi. [1]’den alınmıştır.

Kaybolmalar trafiğin yoğunluğu ile ilgili olduğu için tekrar gönderimin gerekli olduğu durumlarda gönderilen veri miktarını azaltma ihtiyacı duyulur. Bu amaçla, kaynakta, *tıkanıklık penceresi* (*congestion window*) adı verilen bir pencere daha kullanılır. Tıkanıklık penceresinin boyunun ayarlanması üzerinde çalışılan konulardan bir diğeridir.