**UDP (User Datagram Protocol) Nedir?**

* Türkçe anlamı “Kullanıcı Veri Bloğu İletişim Kuralları” olmaktadır.
* TCP/IP adı verilen protokol takımları bulunmaktadır ve UDP bu protokollerden bir tanesidir.
* UDP’nin temel işlevi verilerin gönderimini bağlantı kurulmaksızın gerçekleştirmektir.
* UDP protokolü yeni nesil bilgisayar ağlarında datagram modu oluşturabilmek için geliştirilmiştir. Böylelikle bilgisayarlarda **paket anahtarlı bilgisayar iletişimi** mümkün olabilmektedir.

(**Datagram** varış saati, düzeni ve teslimi garanti edilemeyen paket anahtarlamalı bir ağ ile ilişkilendirilmiş temel bir aktarım ünitesidir.)

* UDP kullanılmayan bilgisayarlarda sıklıkla TCP adı verilen başka bir protokol kullanılmaktadır.
* UDP, **ses ve video gönderiminde ve daha birçok alanda** kullanılır.
* Veri ismine **datagram** denilir.
* Diğer protokoller ile kıyaslandığında az sayıda mesaj alışverişine hizmet etmektedir.
* Dikkat edilmesi gereken bir nokta ise gönderilen verilerin sürecidir. UDP çoğu protokolün aksine veriyi gönderdikten sonra iletilip iletilmediği ile ilgilenmez ve bu konuda kullanıcıya bilgi vermez. Bu sebeple de UDP kullanıcılarına da tam bir güvenilirlik konusunda garanti vermemektedir.
* Süreci takip etmediği için daha hızlı bir şekilde aktarımı gerçekleştirir.
* **Verileri bağlantı kurmadan yollar.**

Buna ek olarak UDP 4 alandan oluşmaktadır.

Bu alanların her birinin uzunluğu ise 16 bittir. Toplam 64 bit uzunluğunda olan UDP ‘nin güvenilirlik konusundaki sorunları ileriki yıllarda halledildiğinde kullanıcı sayısı da artacaktır.

## **TCP’den Farkı Nelerdir?**

* Günümüzde en çok kullanılan protokoller UDP ve TCP’dir.
* Ses ve görüntüler eş zamanlı aktarımını gerçekleştirebilir.
* UDP diğer protokollerin aksine kurulum ve akış kontrolü gerektirmediği için veri iletim hızı yüksektir.
* Dikkat edilmesi gereken bir diğer nokta ise TCP ile UDP’nin **bant genişlikleridir**. TCP’nin bant genişliği daha fazla olduğu için daha sık tercih edilmektedir.
* TCP protokolünün tercih edilme sebeplerinden bir diğeri ise UDP protokolünden **daha güvenilir** olmasıdır. Örneğin TCP kullanılan bir cihazdan veri gönderimi gerçekleştirdiğinizde verinizin gidip gitmediğini kontrol edebilir ve bilgisine ulaşabilirsiniz.
* TCP protokolü ile aktarım yaptığınızda verileriniz sıralı bir şekilde **giderken UDP’de sıralı bir şekilde gitmez.**
* Buna ek olarak TCP ile kullanırsanız size kesintisiz bir bağlantı sunulur. Fakat UDP kullanımında ise **yalnızca veri gönderme işlemi sırasında bağlantı kurulur** ve gönderme işlemi tamamlandığında bağlantı kendiliğinden kapanır.
* UDP **işlemden işleme** iletişimi sağlarken, TCP **ana bilgisayardan ana bilgisayara** iletişimi destekler.
* TCP paket gönderir, UDP datagram.

Hem UDP hem de TCP, **IP üzerinde** çalışır.

## **TCP HANDSHAKING (TCP: Transmission Control Protocol)**

TCP el sıkışması (TCP handshake), bir bilgisayarın ağ üzerinden diğer bir bilgisayarla güvenilir bir iletişim kurmak için kullandığı bir protokol işlemidir. Bu işlem, TCP/IP (Transmission Control Protocol/Internet Protocol) protokolünün bir parçasıdır ve ağ üzerinde veri iletiminin güvenilirliğini sağlamak için kullanılır.

TCP el sıkışması, üç adımda gerçekleşir:

Açılış (SYN): İletişim kurmak isteyen bilgisayar (İstemci), hedef bilgisayara (Sunucu) bir iletişim başlatmak istediğini bildiren bir "bağlantı açma" isteği gönderir. Bu istek, bir "SYN" bayrağı (flag) ile işaretlenir. İstemci, aynı zamanda **rastgele bir başlangıç dizin numarası** (sequence number) da gönderir. İstemci bu adımı tamamladığında, beklemeye alır ve yanıt bekler.

Karşılama (SYN-ACK): Sunucu, İstemciye yanıt verir. Sunucu, İstemciye bir "bağlantı açma onayı" (ACK) ve kendisinin de iletişime hazır olduğunu bildiren bir "SYN" bayrağı gönderir. Ayrıca, **Sunucu da rastgele bir başlangıç dizin numarası oluşturur.** Bu iki bilgisayar artık iletişim kurmaya hazır hale gelir, ancak iletişim güvenliğini sağlamak için bir sonraki adım gerekir.

Onay (ACK): İstemci, Sunucunun yanıtını aldığında, iletişimin karşı tarafının da hazır olduğunu bilir. Bu nedenle, İstemci de bir "ACK" bayrağı ile yanıt verir. Bu yanıt, Sunucunun da İstemcinin iletişime hazır olduğunu bilmesini sağlar. İşte bu adımdan sonra, iki taraf arasındaki iletişim güvenli bir şekilde başlamış olur ve veri alışverişi başlayabilir.

TCP el sıkışması tamamlandığında, veri aktarımı başlar ve iletişim sırasında veriler güvenli bir şekilde paketlenir, iletilir ve doğrulanır. Bu el sıkışması, veri iletiminin güvenilirliğini ve doğruluğunu sağlamak için kullanılır ve iletişim hatlarının düzgün çalışıp çalışmadığını doğrular. Eğer herhangi bir aşamada bir sorun olursa, el sıkışması başarısız olur ve iletişim tekrar kurulmaya çalışılır. Bu, TCP'nin güvenilir veri iletimini sağlama yeteneğini temsil eder.

**Endpoint nedir, hangi amaçla kullanılır ?**

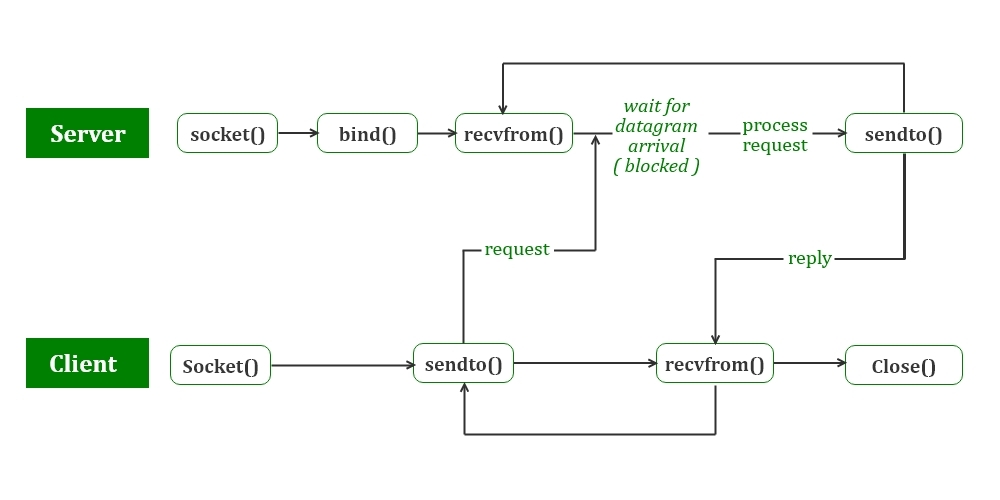
**UDP uç noktası, IP adresi ile kullanılan UDP bağlantı noktasının birleşimidir;** dolayısıyla aynı IP adresindeki farklı UDP bağlantı noktaları, farklı UDP uç noktalarıdır.

**Endpoint, verinin kaynağını veya hedefini temsil eden** bir cihaz veya programdır. UDP, endpointler arasında veri paketlerini iletmek için kullanılır ve bu endpointler, **belirli bir IP adresi ve port numarası** ile tanımlanır.

UDP endpointler birçok farklı uygulama ve senaryoda kullanılır: DNS suncuuları, VoIP (ses iletim uygulamaları), oyunlar, video akışı…

**UDP Server-Client With Socket (Sunucu-İstemci)**

UDP'de istemci, TCP'deki gibi sunucuyla bağlantı kurmaz ve bunun yerine bir datagram gönderir. Benzer şekilde, sunucunun bir bağlantıyı kabul etmesine gerek yoktur ve yalnızca datagramların gelmesini bekler. Datagramlar varışta, sunucunun verileri doğru istemciye göndermek için kullandığı gönderenin adresini içerir.



**Soket**, TCP/IP'de, veri iletişimi için gereken iki bilgi olan **IP adresi ve port numarasının yan yana yazılmasıyla oluşan iletişim kanalıdır.**

**Recvfrom** işlevi, tanımlayıcı soket **tarafından** adlandırılan bir soketteki verileri alır ve onu bir arabellekte saklar.

**UDP Sunucusu (Server)**

1. Bir UDP soketi oluşturun.
2. Soketi sunucu adresine bağlayın. //bind
3. Datagram paketi istemciden gelene kadar bekleyin. //listen
4. Datagram paketini işleyin ve istemciye bir yanıt gönderin.
5. 3. Adıma geri dönün.

**UDP İstemcisi (Client)**

1. Bir UDP soketi oluşturun.
2. Sunucuya bir mesaj gönderin.
3. Sunucudan yanıt alınana kadar bekleyin.
4. Yanıtı işleyin ve gerekirse 2. adıma geri dönün.
5. Soket tanımlayıcıyı kapatın ve çıkın.

**Client ile Multiclient Arasındaki Fark**

**Client (İstemci):** Client, bir sunucuya hizmet isteyen veya sunucu ile iletişim kuran bir bilgisayar veya cihazdır.

**Multipleclient (Çoklu İstemci):** "Multipleclient" ifadesi, aynı anda birden fazla istemcinin sunucuya bağlandığı bir senaryoyu ifade eder. Bu, bir sunucunun aynı anda birçok farklı istemci ile iletişim kurmasını sağlayan bir yapıdır.

**HTTP Protokolü**

HTTP (The Hypertext Transfer Protocol), internet üzerinde bilgi alışverişi yapmak için kullanılan bir iletişim protokolüdür.

HTTP protokolü client-server mimarisinde çalışır.

HTTP, ağa bağlı cihazlar arasında bilgi aktarmak için tasarlanmış bir uygulama katmanı protokolüdür ve bir alt katmanda **TCP üzerine kurulmuştur.**

HTTP isteği, web tarayıcıları gibi internet iletişim platformlarının bir web sitesini yüklemek için ihtiyaç duydukları bilgileri isteme şeklidir.

**Başlıca HTTP metotları şunlardır:**

**GET**: GET metodu, belirtilen bir URL'den veri almak için kullanılır.

**POST**: POST metodu, sunucuya veri göndermek için kullanılır.

**PUT**: PUT metodu, belirli bir URL'de bulunan veriyi güncellemek veya yeni bir veri kaynağı oluşturmak için kullanılır.

**DELETE**: DELETE metodu, belirli bir URL'de (sunucuda) bulunan veriyi silmek için kullanılır.

**HEAD**: HEAD metodu, GET metodu gibi çalışır, ancak yalnızca yanıt başlıklarını alır ve yanıtın gövdesini almaz. Bu, bir belgenin değişip değişmediğini veya var olup olmadığını kontrol etmek için kullanılır.

**PATCH**: PATCH metodu, belirli **bir kaynağın bir kısmını güncellemek** için kullanılır.

**OPTIONS**: OPTIONS metodu, bir sunucunun desteklediği HTTP metotlarını, başlık alanlarını ve diğer özellikleri bildirmek için kullanılır.

**CONNECT**: CONNECT metodu, sunucu ile güvenli bir şekilde iletişim kurmak için kullanılır.

**TRACE**: TRACE metodu, bir sunucuya bir isteği gönderir, ancak sunucunun isteği alırken nasıl işlediğini ve değiştirdiğini izlemek için kullanılır. Güvenlik nedenleriyle genellikle devre dışı bırakılır.

**HTTP durum kodu nedir?**

HTTP durum kodları, genellikle bir HTTP isteğinin başarıyla tamamlanıp tamamlanmadığını göstermek için kullanılan 3 basamaklı kodlardır. Durum kodları aşağıdaki 5 bloğa bölünmüştür:

1. 1xx Informational (bilgi amaçlı)
2. 2xx Success
3. 3xx Redirection (yönlendirme)
4. 4xx Client Error
5. 5xx Server Error