

BİLGİSAYAR AĞLARI

LABORATUVAR

1. Önkoşullar

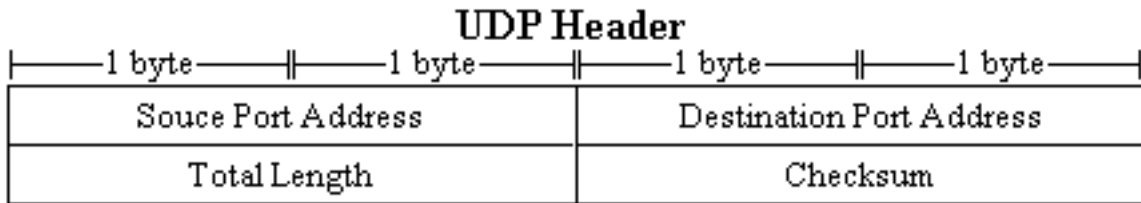
- 1.1. Protokol kavramını bilmeli
- 1.2. Client-Server terimlerinin anlamlarını bilmeli
- 1.3. Katmanlı mimariyi bilmeli
- 1.4. Java diline hakim olmalı
- 1.5. Veri yapılarını bilmeli
- 1.6. Thread ve Thread senkronizasyonu konularını bilmeli

2. UDP Nedir?

UDP (User Datagram Protocol) Kullanıcı Veri Bloğu Protokolü demektir.

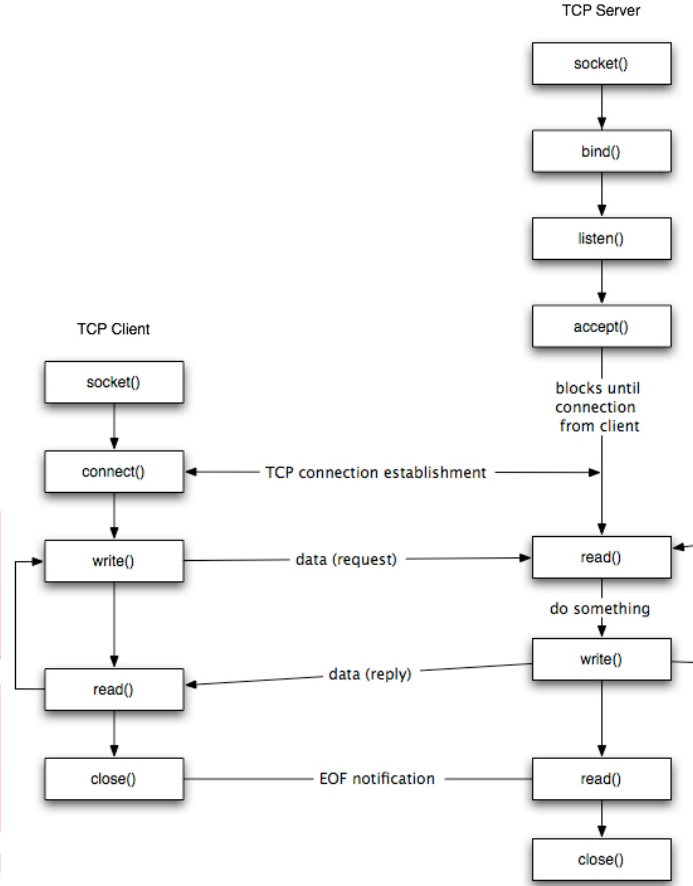
TCP/IP modelinde taşıma (transport) katmanında bulunan bir veri iletim protokolüdür. Bu katmanda veri gerekli büyüklükte paketlere (segment) bölünerek kapsülленir. Bu protokol verilerin hızlı ve hata denetimsiz olarak iletilmesini sağlar. Hata denetimi olmadığı için daha hızlı iletim olması beklenir.

Bir ağ üzerinde veriler iletilirken hatasız iletilmesine dikkat edilir. Fakat bu hata denetimi hem verinin büyümesine hem de internet trafiğinin artmasına neden olur (TCP). Bununla birlikte bazı verilerin hatasız gönderilmesine gerek yoktur. Örneğin ses ve görüntü gibi verilerde küçük hatalar pek algılanmaz, bundan dolayı verilerin hatasız gönderilmesinden önce hızlı gönderilmesi tercih edilir. Kısacası UDP protokolünde gönderilen bir paketin hata kontrolü yapılmaz, veriler bağlantı kurmadan yollar.



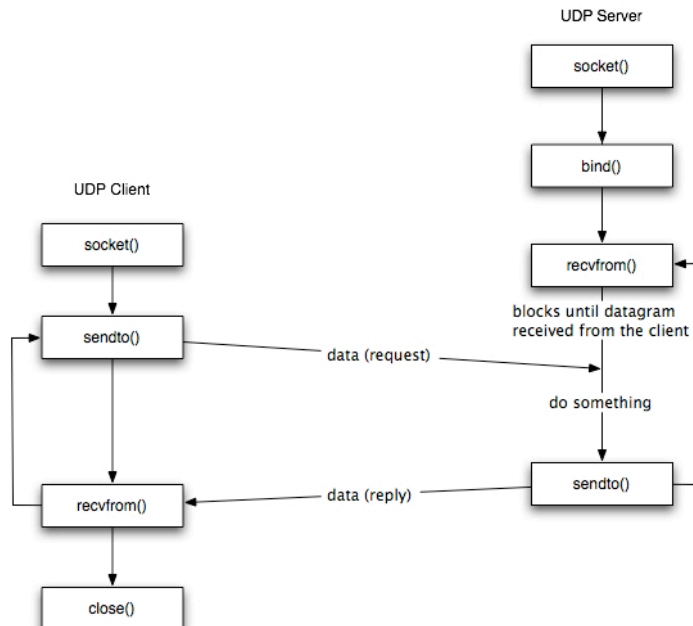
3. Server-Client İletişimi

Öncelikle TCP iletişiminin nasıl kurulduğunu hatırlayalım.



Resim 2: TCP İletişimi

UDP iletişimde ise, Resim 2’de verilmiş olan TCP iletişiminden farklı olarak, server ve client arasında herhangi bir bağlantı kurulmasına gerek duyulmadan (connectionless) veri gönderilip alınabilir.



Resim 3: UDP İletişimi

4. UDP Soket Programlama

Internet ağlarında soket programlama oldukça yaygın kullanılır. Soket ulaşılmak istenilen bilgisayarın IP değeri ve Port numarasından meydana gelir. Bu değerleri kullanarak iletişime geçmek istediğimiz bilgisayara erişmiş oluruz.

Bilgisayarımızda port numarası olarak bir tam sayı (16-bit unsigned integer, ranging from 0 to 65535) atarız. Bu tam sayı 1024'ten küçük olamaz çünkü bu portlar özel işlemler için ayrılmıştır. Bunların dışında 65535 dolaylarına kadar port numaralarını kullanabiliriz. Yine de eğer bir port numarası kullanılmak isteniyorsa internet üzerinden kullanım durumunun araştırılması iyi olur.

Soket programlamada server-client terimleri de oldukça sık kullanılır. Server kendine gelen istekleri cevaplamak için sürekli dinlemede kalan bilgisayardır, dinlemede kalmayı sağlayan programa da server programı denir. Client ise server bilgisayarına istekte bulunan yani hizmet isteyen bilgisayar veya türevleridir.

4.1. Java ile UDP Server Yazımı

Soket programlama tüm programlama dilleriyle yapılabilir. Fakat bunun kolaylığı zorluğu dilden dile değişir. Bu laboratuvarıda Java Programlama dili ile Server yazımını gerçekleştireceğiz.

Her Server yazılımı bilgisayarda bir port dinler ve bu porta bir mesaj geldiğinde server yazılımı gerekli işlemleri yapar. Java dilinde UDP server yazılımı için kullanılacak soket kütüphaneleri:

- java.net.DatagramPacket
- java.net.DatagramSocket

Bu kütüphanelerden bir soket tanımlaması **DatagramSocket** sınıfı kullanılarak yapılır. DatagramSocket sınıfı port bilgisi ister. Port bilgisi verildikten sonra bir thread yardımıyla oluşturulan bu soket **receive** metoduyla dinlenir. Soketin **receive** metodu "blocking call" olarak bilinir. Yani program bu metottan bir cevap gelmeden/almadan bir sonraki satıra geçmez. Eğer bir veri gelmişse program bir alt satırdan devam eder ve metoda gönderilmiş olan DatagramPacket nesnesi dolu olarak döner. Server kendi IP+Port'una gelen verileri böylece almış olur.

Servera bir veri gelmişse mutlaka bir ip adresi ve port numarası bulunur. Server programı dilerse gelen client'ları hafızasında saklar ve daha sonra bunlarla iletişime geçebilir.

4.2. Java ile UDP Client Yazımı

Serverda olduğu gibi, UDP client bir soket nesnesine ihtiyaç duyar. Client, serverdan farklı olarak "server ip" ve "server port" numaralarını bilmelidir. Bu bilgiler mesajın yollanacağı bilgisayar ve port numarası olacaktır. Client açık olduğu sürece kendi soketini sürekli dinler.

Client mesaj göndereceği zaman mesaj içeriğinden bir **DatagramPacket** oluşturulur ve soketin **send** metodu yardımıyla gönderilir.

Client, serverdan habersiz olarak veri gönderebilir. Bunun için server'ın açık olmasına veya açık olduğunun kontrol edilmesine ihtiyaç duymaz. Eğer veri gönderilmek istenen server bu veriyi alabilecek konumda değilse (dinlemede değilse) veri iletilmemiş olur. Bunun için geriye, client tarafına herhangi bir bilgi dönmez.

5. Lab Deęerlendirme

- 5.1. UDP Protokolü nerelerde ve neden kullanılır?
- 5.2. UDP Protokolünün IP adresini kapsamamasının nedeni nedir?
- 5.3. UDP checksum değeri nasıl hesaplanır? (Araştırıp cevaplayınız)
- 5.4. Serverlar gelen client'larla geri bağlantıyı nasıl sağlar?

6. Kaynaklar

- 6.1. Computer Networking: A Top-Down Approach, 6th ediditon. J.F. Kurose and K.W. Ross
- 6.2. <https://docs.oracle.com/javase/tutorial/networking/datagrams/>
- 6.3. <https://docs.oracle.com/javase/tutorial/networking/datagrams/clientServer.html>

