**TC**

**KOCAELİ ÜNİVERSİTESİ**

**MÜHENDİSLİK FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**Ders Adı : Dağıtık Sistemler**

**Yrd.Doç.Dr. Ahmet Sayar**

**Proje : Chat Programı**

**Hazırlayanlar:**

**110201035 Burak UTKUCU**

**110201001 Ramazan GÜÇLÜ**

**110201003 Necmettin UZUN**

**110201063 Enes ALTUN**

**110201053 Feyyaz ÖZHAN**

Chat Projesi

Bu proje tcp udp kullanılarak iki bilgisayar arasında chat yapan programdır.

**Soket**

İnternet üzerinden süren iletişim **soketler** aracılığı ile sağlanmaktadır. **soketin** 2 temel bileşeni vardır. **IP** ve **port**. **IP**, ağdaki hangi bilgisayarın olduğunu belirtirken, **port** ise bilgisayarın içerisindeki bölümlenmiş numaralardır. İki soket tipi vardır.

1)**TCP**: Açılımı **Transmission Control Protocol**‘dür.Bu protokolde, istemciler birbirleri arasında bir bağlantı kurarlar ve bu bağlantı ile stream üzerinden veri akışını sağlarlar.Verileri doğru bir sıralamada iletirler.Kayıplara karşı veriler tekrar yollanır ve kontrol edilir.Bu yüzden verilerin akıbeti hakkında endişelenmenize gerek kalmaz.Fakat bu sebeplerden dolayı **UDP**‘ye göre daha yavaştırlar.

2)**UDP**: Açılımı **User Datagram Protocol**‘dür.Bu protokolde ise istemciler arasında bağlantı kurulmaz.Yollanacak veri paketlenir, adres bilgisi ve port bilgisi eklenip yollanır.Aynı mektuplar gibi.Fakat paketler internet üzerinde kaybolabilmekte ya da gecikebilmektedir.Paketler birbirinden bağımsız gittiği için hedefe ulaşan paketlerin sayısı azalabilir ya da doğru sıra ile ulaşmayabilir.Bunu kontrol etmek ve düzeltmek yazılımcıya kalır.Fakat bu dezavantajlarının yanı sıra, **TCP**‘ye göre daha hızlıdır.

Eğer dinamik olarak bir veri aktarımı oluyorsa ve hız önemliyse **UDP** kullanmak daha mantıklı olacaktır.Örneğin video aktarımı sırasında kayıplar çok önemli olmayabilir ama hızlı ve akıcı olması gerekir.

**Programın içeriği**

Chat programı server ve client ten oluşur. Programda tcp ve udp seçeneği vardır. Tcp de server kullanıcının girdiği bir portu dinlemeye alır. Client programı o port a bağlantı yaptığında chat başlar.

Bağlantı sağlamadan program bir alt satırdaki kod parçasına geçmez

clientSocket = serverSocket.accept();

Client'a veri gönderimi için kullandığımız PrintWriter nesnesi oluşturulur

PrintWriter out = new PrintWriter(clientSocket.getOutputStream(), true);

out.println(“”);

Client'dan gelen verileri tutan BufferedReader nesnesi oluşturulur

BufferedReader in = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));

clientGelen = in.readLine()

udp seçildiğinde server ile client arasında bağlantı sağlanmaz. Verinin doğru bir şekilde iletilmesi kontrol edilmez. Ilk olarak client server in portuna bir datagram paket yollar. Server bu paketten source port çıkarır ve oraya veri gönderebilir.

Server e veri alınırken

byte[] inBuf = new byte[256];

DatagramPacket inPacket = new DatagramPacket(inBuf, inBuf.length);

datagramSocket.receive(inPacket);

String data = new String(inPacket.getData(), 0, inPacket.getLength());

Bu gelen veri den source port ve address bulunur.

int source\_port = inPacket.getPort();

InetAddress source\_address = inPacket.getAddress();

Server cliente veri gönderirken bu source port ve source address i kullanarak cliente veri gönderir.

byte[] outBuf = msg.getBytes();

DatagramPacket outPacket = new DatagramPacket(outBuf, 0, outBuf.length, source\_address, source\_port);

datagramSocket.send(outPacket);

**Kaynak**

<http://phoenix.goucher.edu/>

<http://systembash.com/>

<http://lycog.com/>

<http://technet.microsoft.com/>