За сокетите накратко в примери:

Socket - комбинация на IP адрес с порт;

UDP - Useр Datagram Protocol - Транспортен протокол, който предоставя нетвърдена услуга - (изпращачът на пакет данни (наричан тук дейтаграма) не знае дали получателят е получил данните)

Реализация в Java:

import java.net.\*;

DatagramSocket - клас в Java за обмен на дейтаграми.

DatagramPacket - обектите от този клас са тези, чрез които става обменът на данните.

Използване:

//Дефиниране UDP Soket

DatagramSocket datagramSocket = new DatagramSocket(порт); //На мястото "порт" се записва число (порт), който да бъде резервиран на текущия компютър, за да може да поличават пакети, адресирани към програмата (порта). Възможно е да няма специфично зададен, тогава операционната система задава произволен порт.

//Получаване на данни

byte[] buffer = new byte[1024]; //Заделяне на място, където данните да бъдат записани

DatagramPacket datagramPacket = new DatagramPacket(buffer, 1024); //Създаване на дейтаграма-обект, където се задава къде да бъдат записани данните и колко е максималният размер на мястото за съхранение

datagramSocket.receive(datagramPacket); //Процесът/Нишката изпада в състояние на очакване за пристигането на входящите данни.

//Изпращане на данни

String textToSend = "Hello my network world!"; //Подготовка на данни те за изпращане (в примерът е посочен стринг, който трябва да бъде изпратен)

DatagramPacket sendDatagram = new DatagramPacket(textToSend.getBytes(), textToSend.length()); //Подготвяне на дейтаграма, носител на данните. В конструктура се изисква първия параметър да е масив от байтове, а втория да е броят байтове, който трябва да бъдат изпратени.

datagramSocket.send(sendDatagram);

//Свързване между компютри

//В мрежовото програмиране се налага едно от устройствата да започне да изпраща данни първо, за да се осъществи комуникацията, за това на поне едно устройство трябва да му се каже на къде да изпраща данни. Реализацията на това в Java става чрез метода на обект от клас DatagramSocket - connect(). Като параметри се подават двете необходими неща във всяка мрежова връзка - IP адрес или име на хост, както и номер на порт. Пример:

datagramSocket.connect(InetAddress.getByName("remoteServer"),отдалечен\_порт);

//Изпращане на данни към всички в дадена мрежа - Broadcast.

//Изпращането към всички се осъществява посредством специално разрешаване за наличието на такава връзка.

datagramSocket.setBroadcast(true);

datagramSocket.connect(InetAddress.getByName("255.255.255.255"),отдалечен\_порт);

datagramSocket.send(sendDatagram);

//Ако желаете да получавате данни от кой да е източник е нужно да не специфицирате отдалечения сокет (не трява да използвате метод connect). За да отговорите на този изпращач, данните му за сокета (IP адрес и порт) могат да се извлекат чрез методи на получената дейтаграма

//Пример за изпращане на данни

public class UDPExample {

public static void main(String[] args) {

DatagramSocket datagramSocket=null;

try {

datagramSocket = new DatagramSocket();

datagramSocket.connect(InetAddress.getByName("localhost"),5678);

String textToSend = "Hello my network world!";

DatagramPacket sendDatagram = new DatagramPacket(textToSend.getBytes(), textToSend.length());

datagramSocket.send(sendDatagram);

} catch (SocketException e) {

System.err.println("Greshka pri otvaryaneto na porta");

} catch (UnknownHostException e) {

System.err.println("Neizvesten host");

} catch (IOException e) {

System.err.println("Druga vhodno/izhodna greshka");

}

finally

{

if(datagramSocket!=null)

datagramSocket.close();

}

}

}

//Пример за приемане на данни

public class UDPExample {

public static void main(String[] args) {

DatagramSocket datagramSocket = null;

try {

datagramSocket = new DatagramSocket(8765);

byte[] buffer = new byte[1024];

DatagramPacket receiveDatagram = new DatagramPacket(buffer,

buffer.length);

datagramSocket.receive(receiveDatagram);

System.out.println("Polucheni danni ot "

+ receiveDatagram.getAddress().getHostAddress() + ":"

+ receiveDatagram.getPort() + " >> "

+ new String(receiveDatagram.getData()));

} catch (SocketException e) {

System.err.println("Greshka pri otvaryaneto na porta");

} catch (UnknownHostException e) {

System.err.println("Neizvesten host");

} catch (IOException e) {

System.err.println("Druga vhodno/izhodna greshka");

} finally {

if (datagramSocket != null)

datagramSocket.close();

}

}

}