

# Multicast UDP

# Indirizzi IP di Multicast

- classe D: multicast

- binario 1110-----.-----.

- da 224.0.0.0 a 239.255.255.255

- Indirizzi Liberi

- da 225.0.0.0 a 238.255.255.255

- Indirizzi Esistenti

- 224.0.0.1 ALL-SYSTEMS.MCAST.NET

- 224.0.1.1 NTP.MCAST.NET

- . . . . .

# Time To Live (TTL)

Numero massimo di router che il pacchetto può attraversare

- 0 host locale
- 1 rete locale
- ...
- 48 nazione
- ...
- 255 internet

# Multicast Socket Ricezione

```
InetAddress mg = null;
int port = 4000;
MulticastSocket ms = null;

try {
    mg = InetAddress.getByName("224.0.0.1");

    ms = new MulticastSocket(port);

    ms.joinGroup(mg);

}
catch(UnknownHostException e) {
    System.out.println("Errore: gruppo mult sconosciuto");
}
catch(IOException e) {
    System.out.println("Errore");
}
```

Gruppo di multicast predefinito:  
ALL-SYSTEMS.MCAST.NET

Adesione al gruppo di multicast

# Multicast Socket Ricezione (2)

```
byte[] buffer = new byte[65507];
DatagramPacket dp =
    new DatagramPacket(buffer, buffer.length);

try {
    ms.receive(dp);

    System.out.println(
        new String(dp.getData(), 0, dp.getLength()));
}
catch(IOException e) {
    System.out.println("Errore");
}
```

# Multicast Socket Spedizione

```
InetAddress mg = null;  
MulticastSocket ms = null;
```

```
try {  
    mg = InetAddress.getByName("224.0.0.1");  
  
    ms = new MulticastSocket();  
  
    ms.setTimeToLive(1); //rete locale  
}  
catch(IOException e) {  
    System.out.println("Errore");  
}
```

Nel socket di spedizione non è necessario specificare la porta

# Multicast Socket Spedizione (2)

```
byte[] b = (new String("ciao a tutti")).getBytes();
int porta = 4000;
DatagramPacket dp =
    new DatagramPacket(b, b.length, mg, porta);

try {

    ms.send(dp);

}
catch(IOException e) {
    System.out.println("Errore");
}
```

# Ricezione Propri Pacchetti Multicast

.....

```
System.out.println("il loopback model era: " +  
                    ms.getLoopbackMode());  
  
ms.setLoopbackMode(true);
```

True = non riceve i pacchetti multicast spediti da se stesso

**Attenzione!!! non tutte le implementazioni della JVM supportano il setLoopbackMode**

# Chiusura Multicast Socket

```
try {  
    ms.leaveGroup(mg);  
    ms.close();  
}  
catch (IOException e) {  
    System.out.println("Errore");  
}
```