



**INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO**



APLICACIONES PARA COMUNICACIONES EN RED

EQUIPO 5

TEMAS

3.3 MULTIDIFUSIÓN.

3.3.1 DATAGRAMAS DE MULTIDIFUSIÓN.

**3.3.2 RESOLUCIÓN DE DIRECCIONES DE MULTIDIFUSIÓN
LÓGICAS A FÍSICAS.**

- **AMARO ROMERO ROBERTO FERNANDO**
- **LOPEZ VELAZQUEZ JESUS MICHELLE**
- **NAVARRETE PIÑA ARATH ANTONIO**
- **PALACIOS ALVARADO CARLA**

INTRODUCCIÓN

El multicasting emerge como una poderosa herramienta que equilibra eficiencia y alcance en la transmisión de datos. A diferencia de los modelos de comunicación unicast, que se limitan a interacciones punto a punto, y broadcasting, que puede generar congestión de red, el multicast ofrece una solución única al permitir que un solo host envíe datos a múltiples receptores interesados. Este enfoque optimiza el uso del ancho de banda al duplicar los datos solo cuando es necesario, minimizando así el tráfico innecesario en la red.



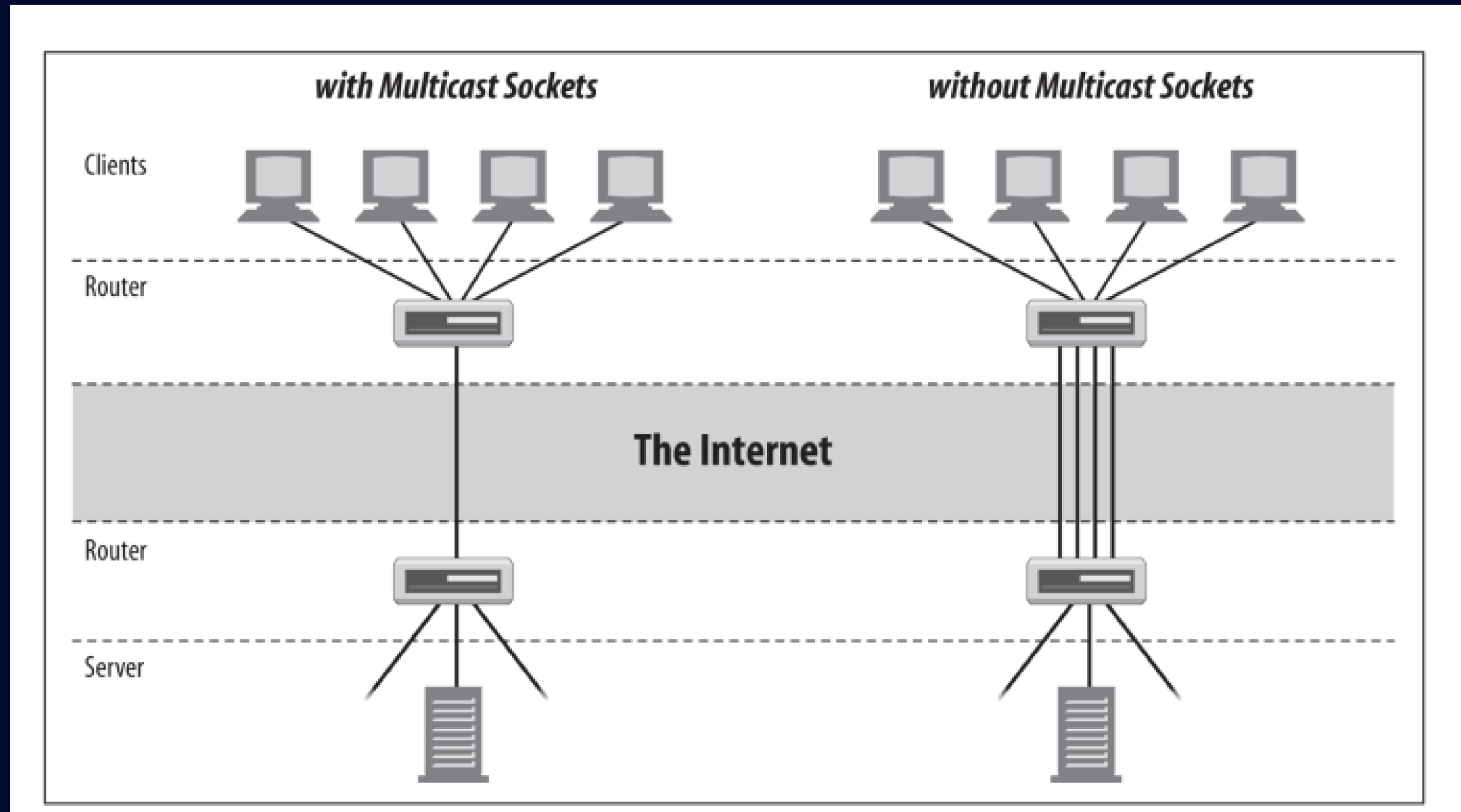
DATAGRAMA

Se utilizan para enviar datos a grupos de multidifusión a través de UDP.

La principal diferencia con un Datagrama normal es la importancia del TTL, un único byte en el encabezado IP

Destinations	TTL value
The local host	0
The local subnet	1
The local campus—that is, the same side of the nearest Internet router—but on possibly different LANs	16
High-bandwidth sites in the same country, generally those fairly close to the backbone	32
All sites in the same country	48
All sites on the same continent	64
High-bandwidth sites worldwide	128
All sites worldwide	255

EJEMPLOS



```
public class MulticastSocket extends DatagramSocket  
    implements Closeable, AutoCloseable
```

```
MulticastSocket ms = new MulticastSocket(2300);
```

Next, join a multicast group using the MulticastSocket's `joinGroup()` method:

```
InetAddress group = InetAddress.getByName("224.2.2.2");  
ms.joinGroup(group);
```

```
byte[] buffer = new byte[8192];  
DatagramPacket dp = new DatagramPacket(buffer, buffer.length);  
ms.receive(dp);
```

```
ms.leaveGroup(group);  
ms.close();
```

```
InetAddress ia = InetAddress.getByName("experiment.mcast.net");  
byte[] data = "Here's some multicast data\r\n".getBytes("UTF-8");  
int port = 4000;  
DatagramPacket dp = new DatagramPacket(data, data.length, ia, port);  
MulticastSocket ms = new MulticastSocket();  
ms.send(dp);
```

NOISISTULCONCON

La multidifusión demuestra ser una herramienta esencial en situaciones donde la velocidad y la eficiencia de la difusión de información son críticas.

Al permitir que los datos se envíen simultáneamente a múltiples destinatarios, facilita conexiones más flexibles y optimiza el uso de los recursos informáticos y físicos.

La capacidad de dirigirse a una gran audiencia con tráfico similar ayuda a gestionar la red de forma más eficiente, garantizando un rendimiento óptimo y maximizando el uso de los recursos disponibles.

REFERENCIAS
REFERENCIAS
REFERENCIAS
Harold, E. R. (2013). Java Network Programming
(4a ed.). O'Reilly Media, Inc.
REFERENCIAS
REFERENCIAS
REFERENCIAS