

LẬP TRÌNH MẠNG



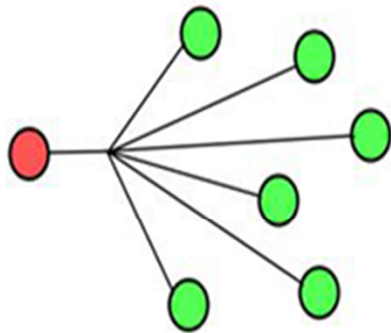
CHƯƠNG V LẬP TRÌNH MULTICAST



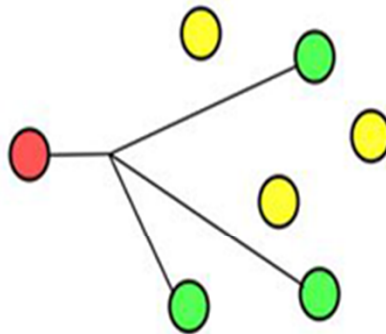
Chương VI: Lập trình Multicast

1. Khái niệm
2. Địa chỉ Multicast
3. Lập trình Multicast với Java

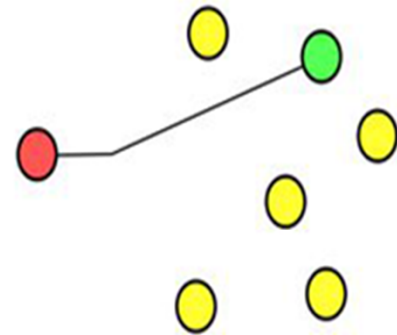
Chương V: Lập trình Multicast



(a) Broadcast

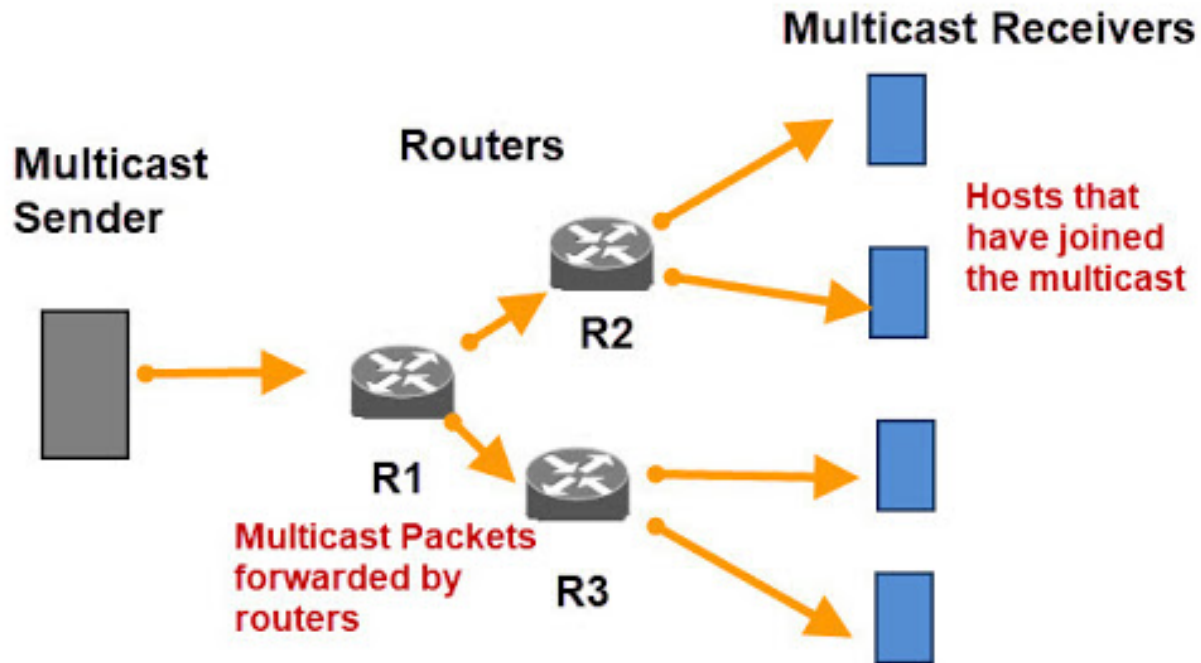


(b) Multicast



(c) Unicast

Chương V: Lập trình Multicast



Chương V: Lập trình Multicast

- Unicast: 1 máy tính gửi và chỉ 1 máy tính nhận trong mạng
- Multicast: liên lạc theo nhóm
- Broadcast: gửi đến tất cả các điểm trong một mạng, không dùng để đặt riêng cho một host

Khái niệm

- Một máy có thể gửi message cho nhiều máy theo một địa chỉ.
- Chỉ cần gửi 1 packet, nhiều địa chỉ có thể nhận được
- Lợi ích:
 - Xử lý thông tin nhanh
 - Giảm lưu lượng dữ liệu trên đường truyền
 - Giải quyết bài toán quan hệ 1-nhiều và nhiều nhiều trên mạng.

Khái niệm Multicast Group

- Multicast group:
 - Là một địa chỉ, tương tự như địa chỉ của một host
 - Nhận diện như địa chỉ máy đích một cách logic hoặc một nhóm.
- Các hosts có thể gửi dữ liệu cho nhóm, tham gia hay rời bỏ nhóm.

Ứng dụng Multicast

- Truyền hình trực tiếp
- Game nhiều người chơi
- Phân phối phần mềm, email
- Giải thuật vạch đường (Routing Protocol) khi các router cập nhật thông tin
- ...

Địa chỉ Multicast

- Sử dụng IGMP (Internet Group Multicasting Protocol)
- Sử dụng lớp D của địa chỉ IP: từ 224.0.0.0 đến 239.255.255.255
- Mỗi địa chỉ IP trong khoảng này biểu diễn cho một nhóm multicast
- Một địa chỉ IP trong nhóm multicast được sử dụng chung cho tất cả các thành viên của nhóm để gửi và nhận dữ liệu

Lập trình Multicast

- Sử dụng giao thức UDP
- Các bước lập trình multicast:
 - Tạo một UDP socket
 - Tham gia một nhóm multicast chỉ ra bởi một địa chỉ IP lớp D
 - Nhận các packet gửi đến cho nhóm đó
 - Gửi các packet đến các máy trong nhóm
 - Rời bỏ khỏi nhóm multicast
 - Đóng socket

Các phương thức Multicast

- Lớp `java.net.MulticastSocket`
- `public MulticastSocket(int port)`: tạo socket với port
- `public void joinGroup(InetAddress group)`: tham gia nhóm multicast tại địa chỉ xác định
- `public void leaveGroup(InetAddress group)`: rời nhóm multicast
- `public void send(DatagramPacket dp)`: gửi datagramPacket đi
- `public synchronized void receive(Datagram Packet dp)`: nhận một datagramPacket

Các phương thức Multicast(tt)

VD: gia nhập nhóm multicast

- **INET_ADDR** = “224.0.0.3”

- **PORT** =8888

```
InetAddress address =InetAddress.getByName(INET_ADDR);
```

```
MulticastSocket clientSocket = new MulticastSocket(PORT);
```

```
clientSocket.joinGroup(address);
```

Các phương thức Multicast(tt)

VD: gửi datagramPacket đến địa chỉ multicast, port

- String msg = "Sent message no " + i
- DatagramPacket msgPacket = new DatagramPacket(msg.getBytes(),
msg.getBytes().length, addr, PORT);
- serverSocket.send(msgPacket);

Các phương thức Multicast(tt)

VD: nhận DatagramPacket trên MulticastSocket

- MulticastSocket **clientSocket** = new MulticastSocket(**PORT**);
- DatagramPacket **msgPacket** = new **DatagramPacket**(buf, buf.length);
- clientSocket.**receive**(**msgPacket**);