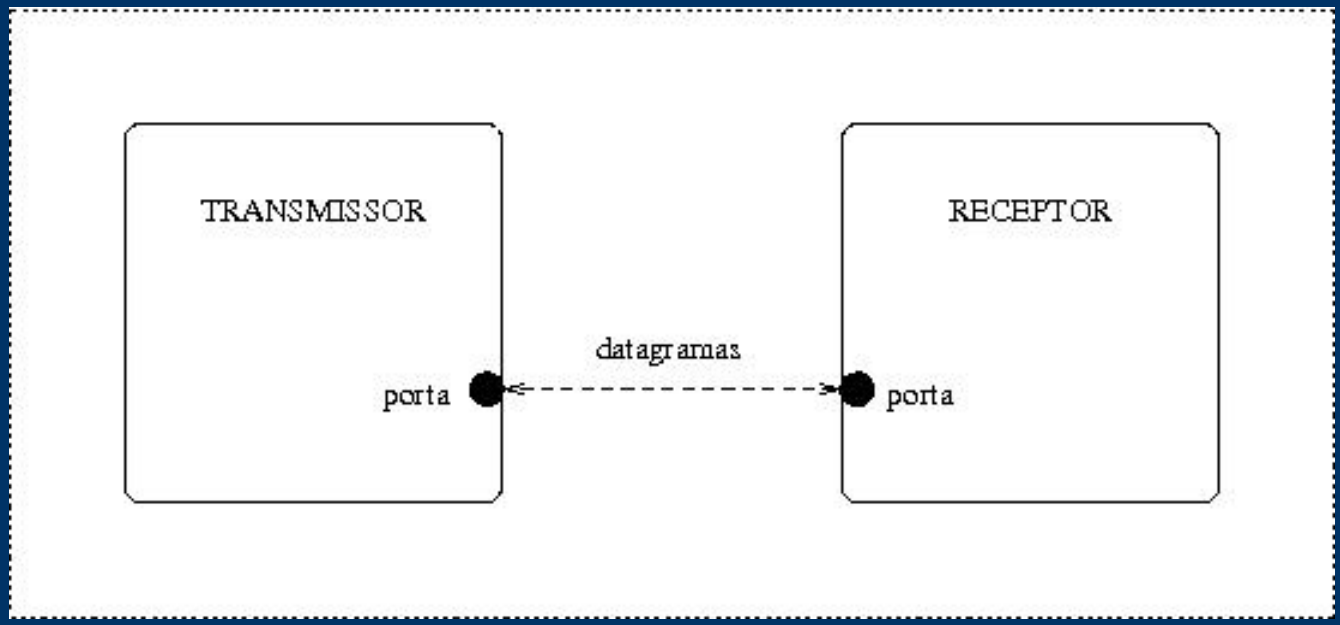
Programação de Aplicações Distribuídas em Java usando UDP/IP

Alcides Calsavara

Java Network Programming, 2nd Ed.
Elliote Rusty Harold
O'Reilly, 2000

Comunicação entre dois processos



Receptor

- Fixa uma porta de comunicação para enviar e receber mensagens
  - int port = 4545;
- Cria um *socket* associado à porta de comunicação
  - DatagramSocket socket = new DatagramSocket(port);
- Prepara um *buffer* para receber um datagrama
  - byte[ ] buffer = new byte[512]; *// 512 bytes de dados*
  - DatagramPacket datagrama = new DatagramPacket(buffer, buffer.length);
- Recebe um datagrama
  - socket.receive(datagrama); *// bloqueante*

Receptor

- Descobre o IP e a porta do transmissor
  - InetAdress RemoteIP = datagrama.getAddress( );
  - int RemotePort = datagrama.getPort( );
- Extrai os dados do datagrama recebido
  - String dados = new String(datagrama.getData( ), datagrama.getOffset( ), datagrama.getLength( ));

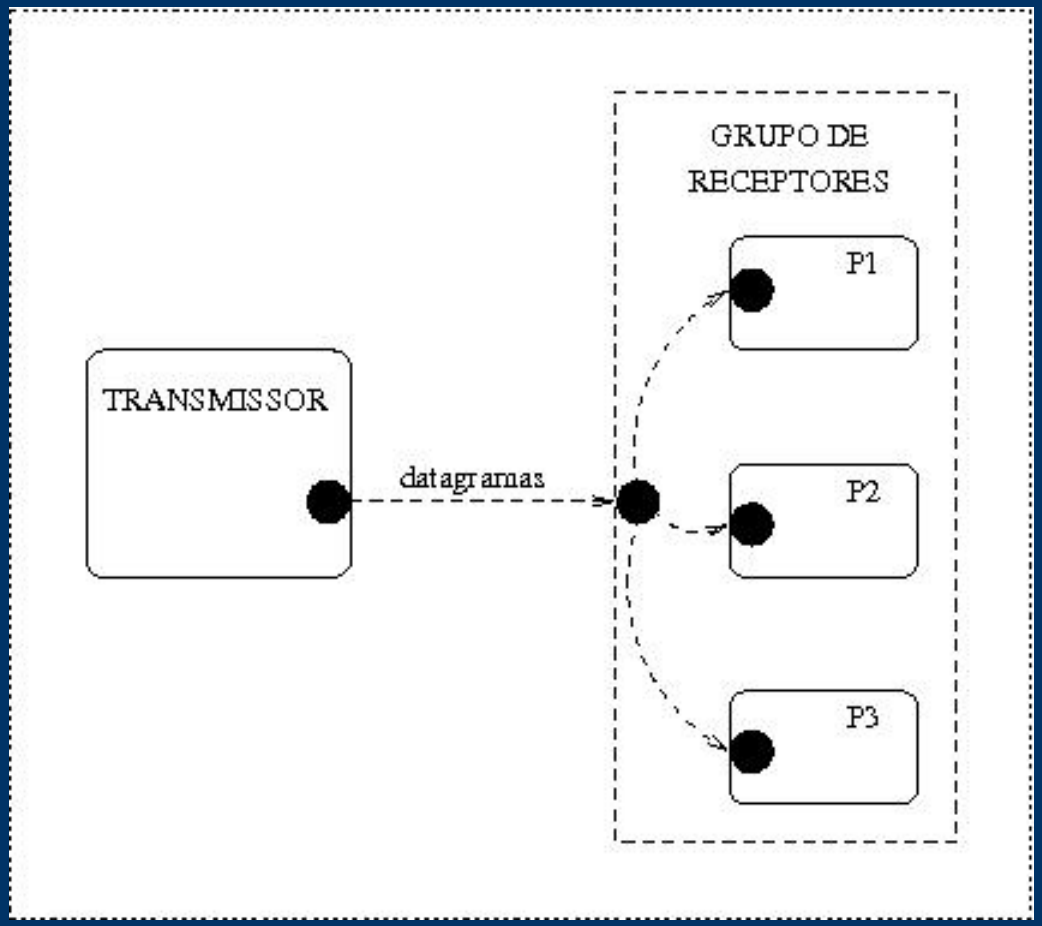
Transmissor

- Define o IP e a porta do receptor
  - InetAddress RemoteIP = InetAddress.getByName(“www.terra.com.br”);
  - int RemotePort = 4545;
- Cria um *socket* associado a uma porta qualquer
  - DatagramSocket socket = new DatagramSocket( );

Transmissor

- Prepara os dados para envio
  - String dados = “Quem venceu a corrida?”;
  - byte[ ] buffer = dados.getBytes( );
- Cria um datagrama para envio
  - DatagramPacket datagrama = new DatagramPacket(buffer, buffer.length, RemoteIP, RemotePort);
- Envia o datagrama para o receptor
  - socket.send(datagrama);

Comunicação em grupo



Endereçamento de grupo

- Cada grupo possui um IP
- Intervalo de IP reservado para grupos
  - De 224.0.0.0
  - A 239.255.255.255

Receptor

- Prepara o endereço do grupo de receptores
  - InetAddress GroupIP = InetAddress.getByName(“224.0.0.1”);
- Fixa uma porta de comunicação do grupo
  - int GroupPort = 2000;
- Cria um socket para comunicação em grupo
  - MulticastSocket socket = new MulticastSocket(GroupPort);
- Entra no grupo de receptores
  - socket.joinGroup(GroupIP);

Receptor

- Prepara um *buffer* para receber um datagrama
  - byte[ ] buffer = new byte[512]; *// 512 bytes de dados*
  - DatagramPacket datagrama = new DatagramPacket(buffer, buffer.length);
- Recebe um datagrama
  - socket.receive(datagrama); *// bloqueante*
- Extrai os dados do datagrama recebido
  - String dados = new String(datagrama.getData( ), datagrama.getOffset( ), datagrama.getLength( ));
- Sai do grupo
  - socket.leaveGroup(GroupIP);

Transmissor

- Prepara o endereço do grupo de receptores
  - InetAddress GroupIP = InetAddress.getByName(“224.0.0.1”);
- Define a porta de comunicação do grupo
  - int GroupPort = 2000;
- Cria um socket para comunicação em grupo
  - MulticastSocket socket = new MulticastSocket( );

Transmissor

- Prepara os dados para envio
  - String dados = “Quem venceu a corrida?”;
  - byte[ ] buffer = dados.getBytes( );
- Cria um datagrama para envio
  - DatagramPacket datagrama = new DatagramPacket(buffer, buffer.length, GroupIP, GroupPort);
- Envia o datagrama para o grupo de receptores
  - socket.send(datagrama);