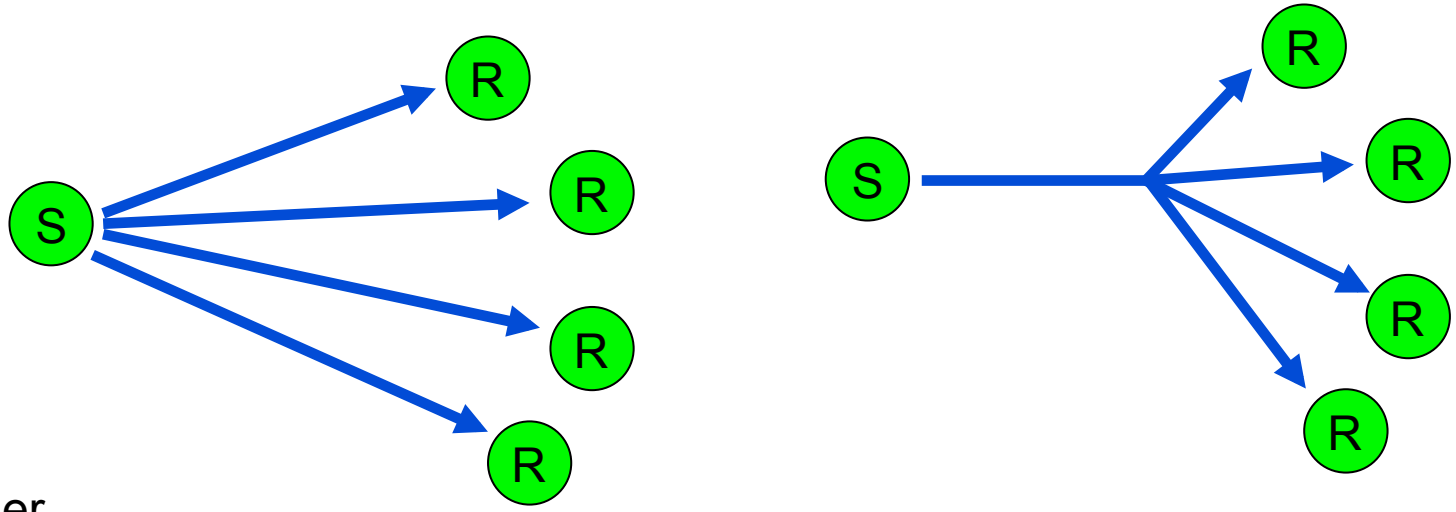
Multicast: Comunicação em Grupo

Sistemas Distribuídos, 2011

SLIDES_7

Multicast

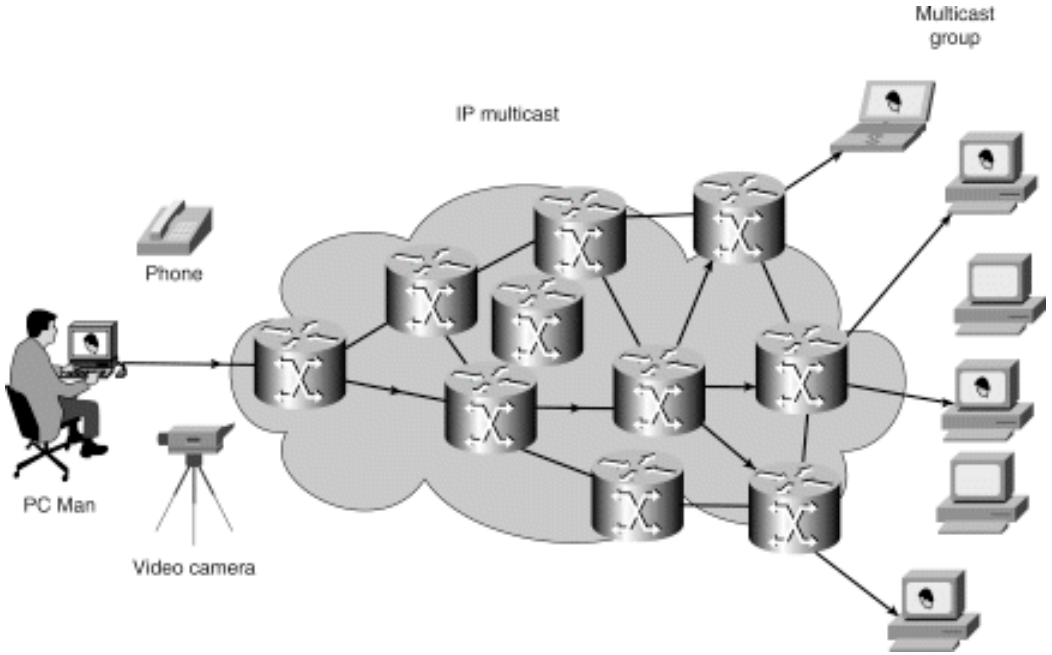
- Ideia: enviar a mesma informação para diversos receptores simultaneamente, utilizando uma única mensagem
  - Unicast
  - Multicast
  - Broadcast



S – Sender
R – Receiver

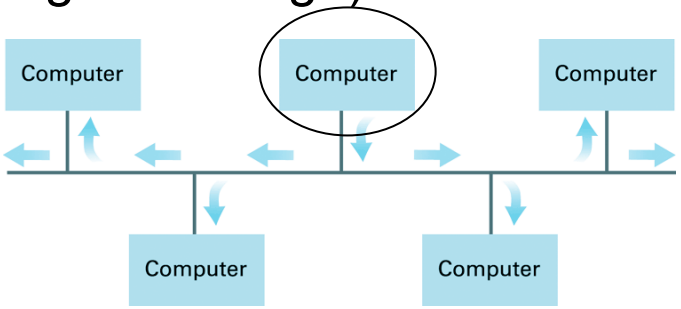
Multicast

- No entanto, pretende-se que não seja necessário utilizar uma ligação por cada receptor.
  - Nas redes locais → uso das características da Ethernet
  - Nas redes globais → uso de routers multicast

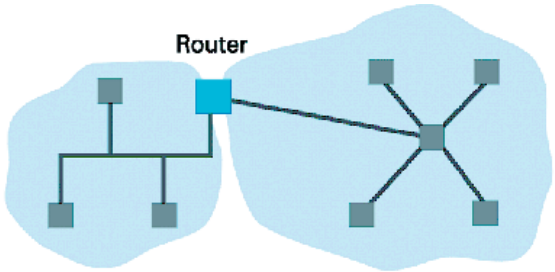


Multicast Routers

- IP packets can be multicasted on a local network and on the wider Internet.
- Local multicasts make use of the hardware multicast feature of Ethernet (single message).



- Internet multicasts make use of multicast routers, which forward single datagrams to routers on other networks with members, where they are again hardware multicasted to local members.



Onde é útil usar Multicasting?

- Aplicações multimédia
  - Difusão de video, som, etc.
  - Jogos multi-utilizador
- Servidores redundantes (fault-tolerance)
- Serviços de descoberta (spontaneous networking)
- Melhorar o desempenho com dados replicados
- Propagação de notificação de eventos

IP Multicast

- Routers interpret any datagram sent to an IP address in the range of: **224.0.0.0 to 239.255.255.255** as ***multicast*.**
- Any IP application with a UDP socket can send to a multicast address (with some limitations).
- Applications that ***join*** a **multicast group** can receive multicast datagrams sent to that group address.

Receiving/Sending Multicasts

- A multicast receiver application must:
  1. Get a UDP socket.
  2. Bind to the application's port number (i.e. name the socket).
  3. Join the application's multicast address group.
  4. Receive.
  5. Close the socket when complete.
- A multicast sender must:
  1. Get a UDP socket.
  2. Set the IP Time-To-Live appropriately.
  3. Send to the application's multicast address and port number.
  4. Close the socket when complete.
- The sender does not have to join a multicast group.

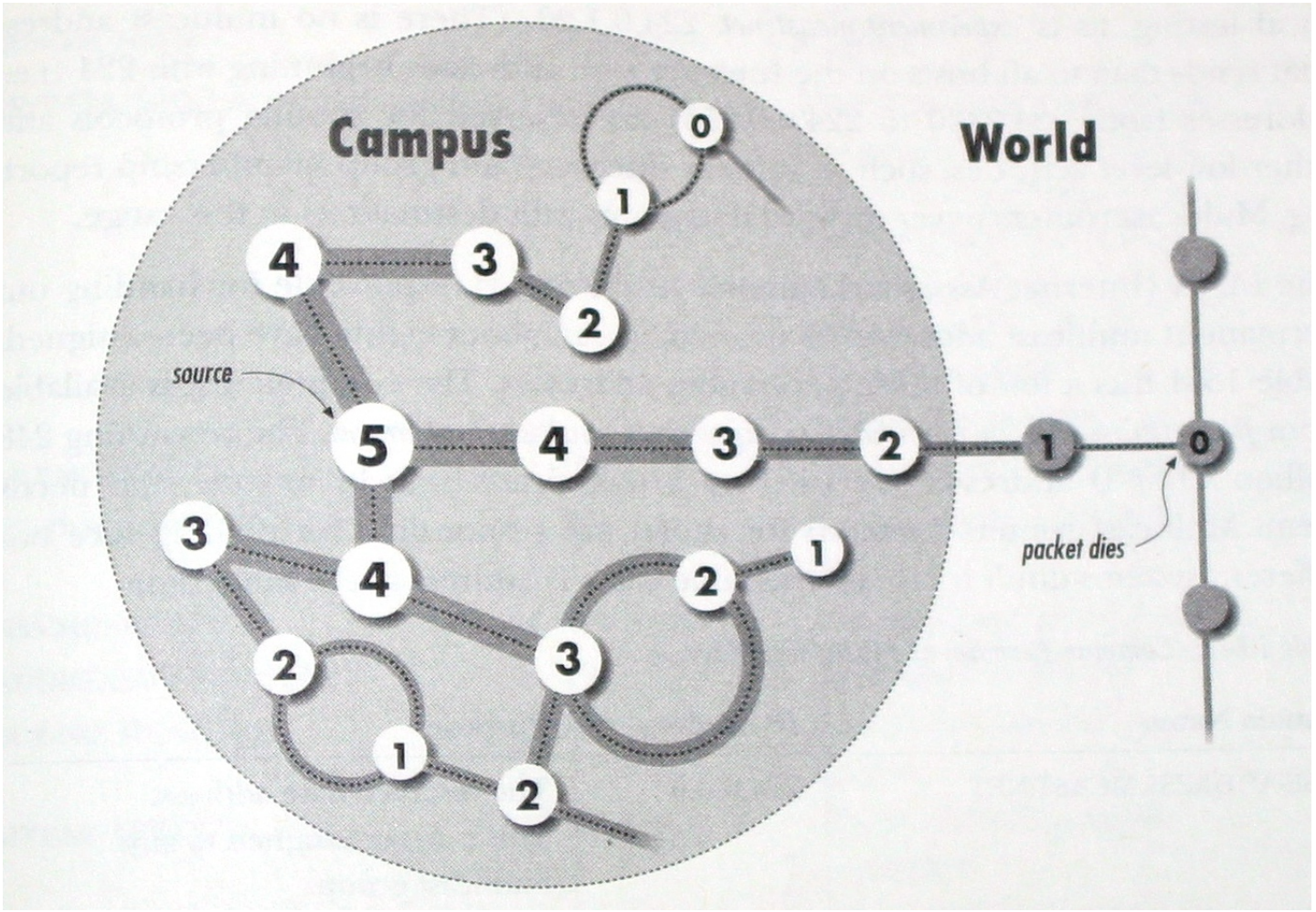
IP Multicast

- When a multicast message arrives at a computer, copies are forwarded to all of the local sockets that have joined the specific multicast address and the specified port number.
- The membership of multicast groups is dynamic: members can join and leave whenever they want.
- IP multicast is available only via UDP.
  - Unreliable: same semantics than UDP

TTL: Time-to-Live

- To limit the distance of propagation of a multicast datagram, the sender specifies the number of routers it is allowed to pass:
  - Time-to-Live: TTL
- **The default value is 1.**
- **This allows the multicast propagate only on the local network.**
- **MulticastSocket**
  - **public void setTimeToLive(int ttl)**

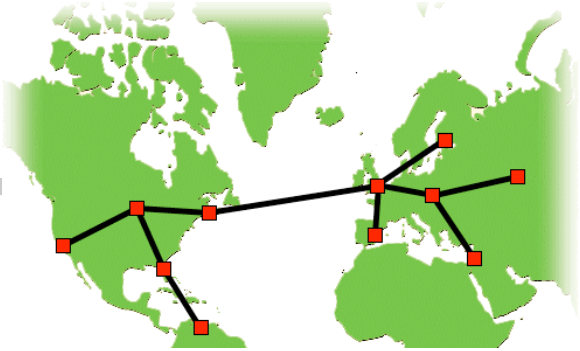
Pack dies when TTL reaches 0

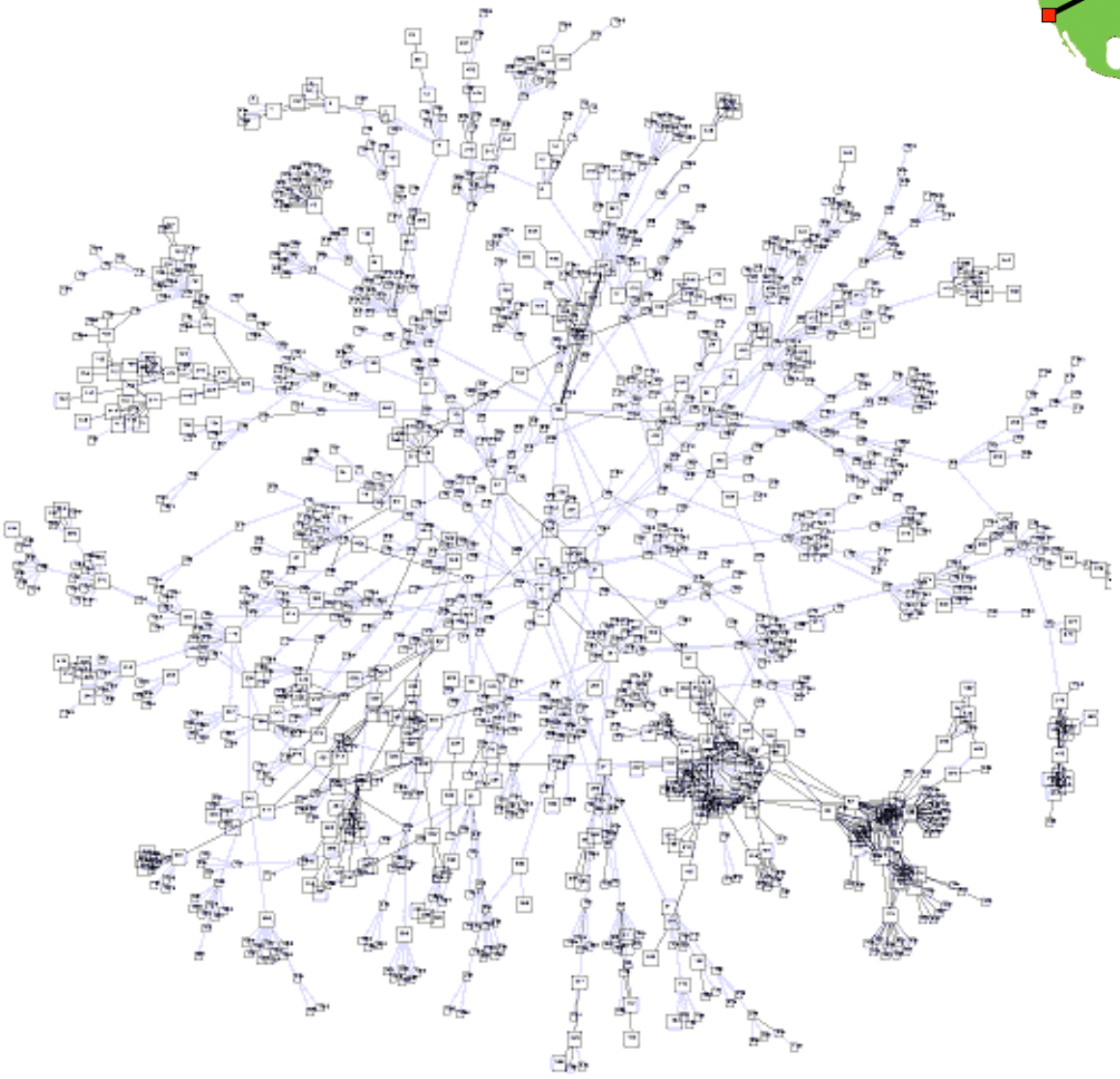


Internet Multicasting

- Most providers of push media currently use ***unicast*** to deliver content to their customers. This works, but does not scale well.
- ***Multicast*** allows a content provider to send a single datastream to a single address, a datastream that network routers subsequently distribute to as many receivers as desired.
- Multicast requires no additional effort on the part of the sender to add new receivers, since the network handles the distribution from the single datastream.

The MBone!





Multicast Address Allocation

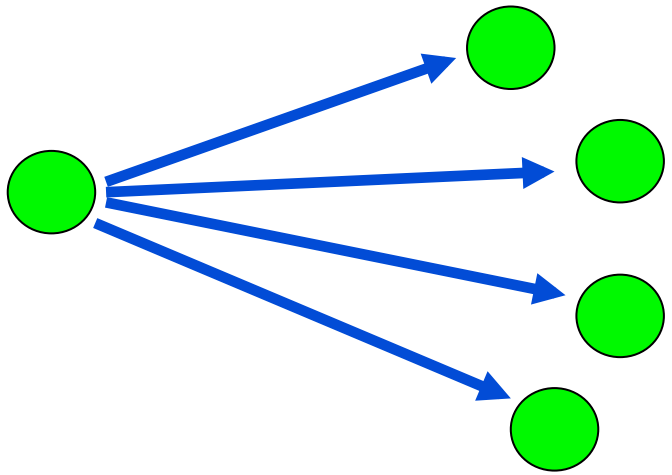
- Permanent Multicast Groups:
  - 224.0.0.1 ..... 224.0.0.255
- Temporary Groups:
  - 225.0.0.1 ..... 239.255.255.255

Temporary Multicast Group

- When a mcast group is created it requires a free multicast address to avoid accidental participation in an existing group.
- The IP multicast protocol does not address this issue.
- When the users only need to communicate locally they set the TTL to a small value.
- However, programs using IP multicasting over the Internet require a solution to the problem (directory of multicast sessions).

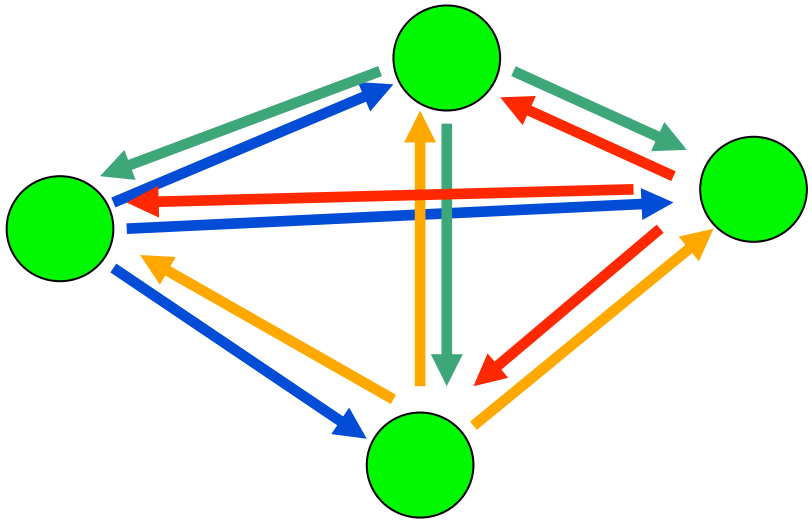
Types of multicast: One-to-Many Multicasting

- Real-time data distribution
  - Weather, stocks, telemetry, and remote sensing
- File distribution
  - Software updates, database mirrors, and web caching
- Cryptographic key distribution
- Network management
- System configuration



Types of multicast: Many-to-Many Multicasting

- Conferencing
  - Video, audio, and whiteboard sharing
- Collaborative document sharing
- Interactive distance learning
- Virtual reality
- Gaming



Java: MulticastSocket

- The multicast datagram socket class is useful for sending and receiving IP multicast packets.
- java.net.MulticastSocket ***extends*** java.net.DatagramSocket
  - A MulticastSocket is a (UDP) DatagramSocket, with additional capabilities for joining "groups" of other multicast hosts on the Internet.
- A multicast group is specified by a class D IP address and by a standard UDP port number.
- Class D IP addresses are in the range: 224.0.0.0 to 239.255.255.255.

Multicast peer joins a group and sends and receives datagrams

```
MulticastSocket socket = null;

try {
  InetAddress group = InetAddress.getByName("225.0.0.0");
  socket = new MulticastSocket(6789);
  socket.joinGroup(group);
  socket.setTimeToLive(1);

  byte[] buf = "Hello".getBytes();
  DatagramPacket mesgOut =
  new DatagramPacket(buf, buf.length, group, port);
  socket.send(msgOut);

  ////

  byte[] inBuf = new byte[8*1024];
  DatagramPacket msgIn =
  new DatagramPacket(inBuf, inBuf.length);
  socket.receive(msgIn);

  System.out.println("Received:" + new String(inBuf, 0,
                                    msgIn.getLength()));

  socket.leaveGroup(group);
} catch (Exception e) { e.printStackTrace(); }
...
```

Example 2: Receiver

```
import java.net.*;
int port = 5000;
group = "225.4.5.6";

// Create the socket and bind it to port 'port'.
MulticastSocket s = new MulticastSocket(port);

// join the multicast group
s.joinGroup(InetAddress.getByName(group));

// Now the socket is set up and we are ready to receive packets
byte buf[] = byte[1024];
DatagramPacket pack = new DatagramPacket(buf, buf.length);
s.receive(pack);

// Do something useful with the data we just received,
System.out.println("Received data from: " + pack.getAddress().toString() + ":" + pack.getPort() + "
      with length: " + pack.getLength()); System.out.write(pack.getData(),0,pack.getLength());

// And when we have finished receiving data leave the multicast group and close the socket
s.leaveGroup(InetAddress.getByName(group);
s.close();
```

When the socket is created the `DatagramSocket.setReuseAddress(boolean)` method is called to enable the SO_REUSEADDR socket option.

Example 2: Simple Sender (does not join group)

```
import java.net.*;

int port = 5000;
String group = "225.4.5.6";
int ttl = 1;

// Create the socket but we don't bind it as we are only going to send data
MulticastSocket s = new MulticastSocket();
// We don't have to join the multicast group if we are only sending data

// Fill the buffer with some data
byte buf[] = byte[10];
for (int i=0; i<buf.length; i++)
        buf[i] = (byte)i;
// Create a DatagramPacket
DatagramPacket pack = new DatagramPacket(buf, buf.length,
                                        InetAddress.getByName(group), port);
// Do a send. Note that send takes a byte for the ttl and not an int.
s.send(pack,(byte)ttl);
// And when we have finished sending data close the socket
s.close();
```

Important methods of **MulticastSocket**

// Create a multicast socket

public MulticastSocket();

// Create a multicast socket in a specific port

public MulticastSocket(int port);

- // Get and set the current time-to-live

public int getTimeToLive();

public void setTimeToLive(int ttl);

- // Join and leave a multicast group

public void joinGroup(InetAddress group);

public void leaveGroup(InetAddress group);

Joining/Leaving Mcast Groups

- When one sends a message to a multicast group, all subscribing recipients to that host and port receive the message (within the time-to-live range of the packet).
- The socket needn't be a member of the multicast group to send messages to it.
- A socket relinquishes membership in a group by the leaveGroup(InetAddress addr) method.
- Multiple MulticastSocket's may subscribe to a multicast group and port concurrently, and they will all receive group datagrams.

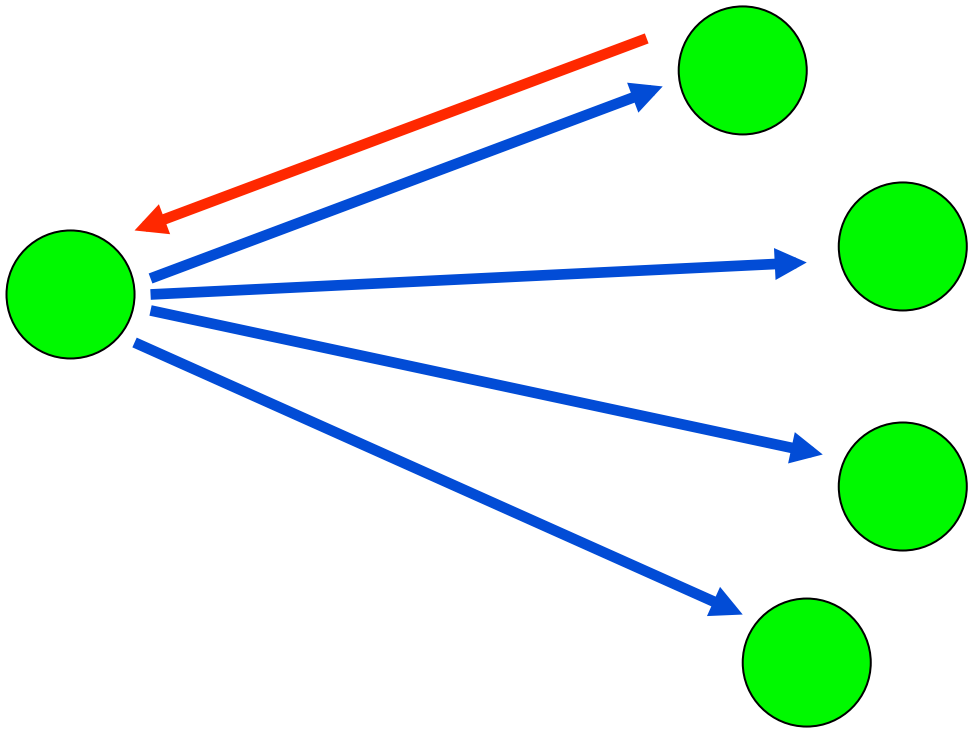
Failure Model

- Datagram multicasts have the same failure characteristics as UDP datagrams:
  - Messages can be lost
  - Messages can arrive out-of-order
  - Messages can be duplicated
- Thereby, with IP mcast we have the following semantics:
  - Unreliable Multicast
- There are other (more powerful) multicast semantics…
  - Check [Coulouris2001] section 11.4

Reliability Semantics of Multicast

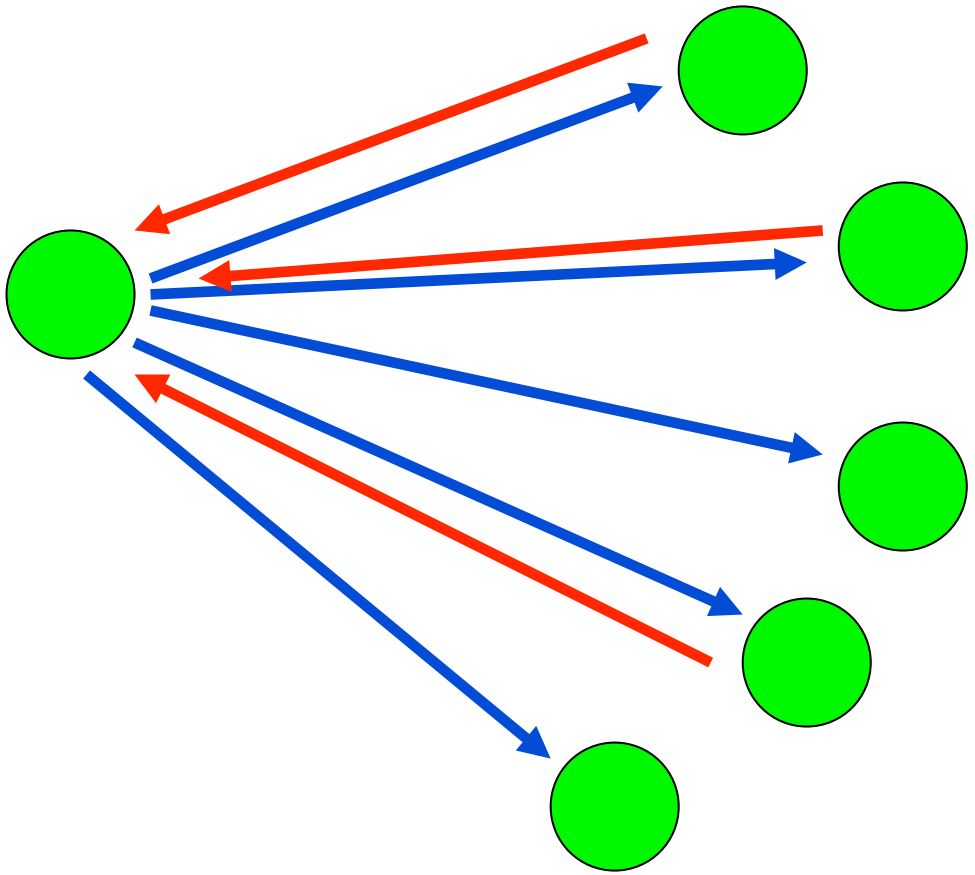
- 0-reliable
- 1-reliable
- m-out-n reliable
- All-reliable

How to achieve 1-Reliable?



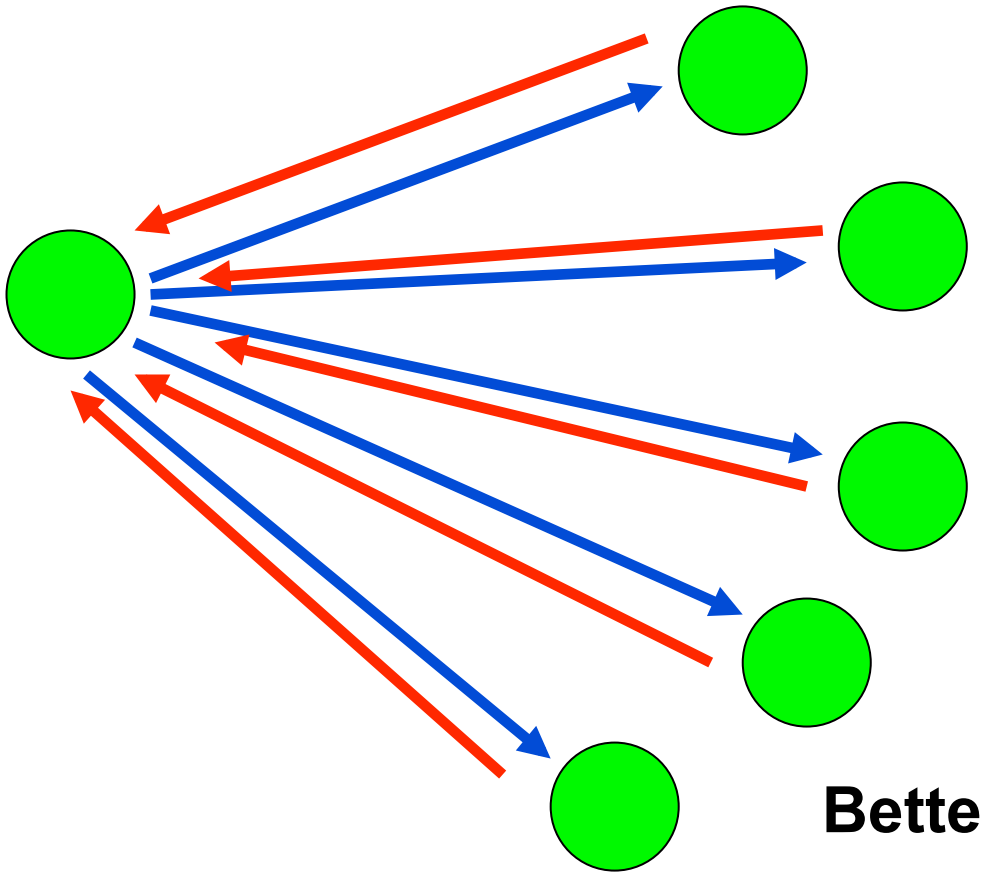
Example: read data from replicated servers

How to achieve m-out-n Reliable?



Example: voting system

How to achieve All-Reliable?



Better Algorithm?...

Example: updating replicated data...

...To be continued...

- Mais à frente será leccionado um tópico sobre comunicação em grupo, onde esta questão da fiabilidade será explicada.