

Re: Multicast or not

Source: <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.mfc/2004-12/0950.html>

From: Relvinian (*r_at_msn.com*)

Date: 12/14/04

Date: Mon, 13 Dec 2004 19:57:11 -0700

AliR wrote:

- > *Hi everyone,*
- >
- > *I am trying to make a design decision when it comes to the client/server*
- > *communication of our program, but I am not knowledgeable enough to make an*
- > *informed decision. I am trying to do some research before I start writing*
- > *test applications.*
- >
- > *Here is the problem, I have one server application which needs to talked to*
- > *multiple clients, (normally 40 to 50 clients) with in a LAN, or WAN. The*
- > *server application needs to send the exact same packets to all the clients,*
- > *and the packets must reach all the clients at almost the same instant, (+/-*
- > *200ms).*
- >
- > *The way things have been done in the past was to have a separate thread for*
- > *each client in the server application which managed a streaming socket*
- > *connect. When a packet had to be sent, the main thread would loop through*
- > *the threads and post a message to each one to send the packet. The problem*
- > *with this was that sometimes the time difference between the first client*
- > *who received message to the last client who received the message would be 2*
- > *or more seconds, specially if the clients were handheld devices with 802.11*
- > *connections.*
- >
- > *One idea was to keep the multiple threads instead of using*
- > *PostThreadMessage, we would use events, and the threads would be set to wait*
- > *for an event WaitForSingleObject, and the main thread would signal the*
- > *events and the threads would send the packets. I am not really sure if that*
- > *is going speed things up or not. And as far as I remember you can't have*
- > *multiple threads waiting on the same event!?*
- >
- > *Now the final idea is to use Multicast, but since Multicast uses UDP I am*
- > *not sure if it is reliable enough for what I am trying to do, specially when*
- > *it comes to the timing issue, and that packet delivery is required.*
- >
- > *Any input into this would be greatly appreciated.*
- > *Thanks*
- > *AliR*
- >

>

AliR,

There is no way to guarantee a packet will reach the destination computer within a specific timeframe (this more--so applies to WAN then LAN but still possible on LAN).

UDP packets by nature are "unsafe" meaning that the IP layer doesn't do any type of error checking to make sure the packet arrived at the destination machine in tact.

UDP packets have their uses and could be the "right" choice in this (especially with multi-casting). For the most part, as long as your network isn't bogged down in traffic, a single multi-cast packet should reach your LAN machines reliably -- not guaranteed though.

Multiple threads on a server sending out packets causes two problems in my opinion. 1) you generate more packets then necessary for the same information for all clients. 2) the Server machine may be slower in sending the packets because of extra threads.

It is a hard design to make in regards what is the best way to send packets to the machine. If you want reliability, go TCP and flood the machines (without waiting for responses). Let the IP protocol handle the errors it can (unless you get critical errors because of bad network, machine down, etc).

Relvinian