

Re: Howto reset and reuse a socket (not having a 7.80 sec delay)

Source: <http://linux.derkeiler.com/Newsgroups/comp.os.linux.development.apps/2005-06/0161.html>

From: Andrei Voropaev (*avorop_at_mail.ru*)

Date: 06/09/05

Date: 9 Jun 2005 08:19:34 GMT

On 2005-06-08, Troels Smit <troels_smit@hotmail.com> wrote:

- > *Hi,*
- >
- > *Im writing a client/server application using c++ socket on linux.*
- > *The clients must be able to reconnect to the server in an efficient*
- > *manner.*
- >
- > *The server is receiving data from the client.*
- > *The client is transmitting about 50% of the time in 200ms intervals.*
- >
- > *My problem occur when I reset the client.*
- >
- > *If the client is not transmitting when I reset it, the server will not*
- > *do anything until the client re-connects. When the client re-connects*
- > *the new connection is accepted and the old connection closed. This is*
- > *fine and progresses very fast.*
- >
- > *Now, if the client IS transmitting when I reset it, the server will*
- > *receive a "Connection reset by peer" and close the connection. When*
- > *the client then re-connects exactly 7.80 seconds!! will pass before*
- > *data starts to flow. The connections is accepted (so the new*
- > *connection is immidiately seen by the server) but the second packet*
- > *will not be received at the server, before 7.80 seconds has passed,*
- > *then everything again runs in a smooth manner.*
- >
- > *I have set the socket options "REUSE" and "LINGER". Where Linger is*
- > *set in aboard mode so the old tcp-connection should be immidiately*
- > *terminated.*
- >
- > *Any ideas ?*

No ideas. It's the problem of your server and not of the socket. Or maybe it is the problem of your client. Post the code, maybe someone will point it out for you. Usually you never should use SO_LINGER, unless you know exactly what you are doing (it is not your case).

comp.os.linux.development.apps: Re: Howto reset and reuse a socket (not having a 7.80 sec delay)

Also try to get the Stevens book "Unix