

## Re: Socket weirdness

---

*Source:*

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework/2006-09/msg00528.html>

---

- *From:* "William Stacey [MVP]" <[william.stacey@xxxxxxxxxx](mailto:william.stacey@xxxxxxxxxx)>
  - *Date:* Wed, 20 Sep 2006 15:36:02 -0400
- 

| Your explanation seems to be accurate with respect to my test results and previous experience with sockets, however Alan's explanation led me to believe that a different mechanism was in place. I am referring specifically to the following statement of

| Alan's:

|

| >>> And at some point the first send now reaches the server, who responds  
| >>> with a packet with the RST flag set. That then makes its way back  
| >>> across the network. Again the client application can be doing sends,  
| >>> and maybe some packets are sent too.  
| >>> Then the RST reaches the client device, and works its way up through the stack.

|

| In your test case there was no concurrent sending, so I assumed that on a blocking send with no concurrent activity that the RST flag would "make its way back across the network" immediately from Alan's statements.

I think we are still talking about the same thing. RST makes it way back to the client on an ACK. However, you may not see that ACK until after 1 or 2 sends. I "think" tcp can send two packets without ACK, but then must wait (correct me here if wrong) for ACK before sending more. But eventually you will get the ACK with the RST set and the next send/receive will fail.

| Interesting that the RST made it back into the client stack but ConnectionReset is returned only on every subsequent request.

I did not follow you here.

| previous statements were referring to this particular statement of Alan's. Why not just fail the initial send if the RST flag has already been received, if in fact it is received?

It will fail if it was received. But in our case it not (and can not) be received until after the first send, because it is set in the ACK of that first send. But I may already have did another Send before first ACK is processed, so I may see it on second or third send. If we always do the "proper" shutdown both sides, then none of this matters and all should work

Re: Socket weirdness

fine:

- 1.. Finish sending data.
- 2.. Call shutdown() with the how parameter set to 1.
- 3.. Loop on recv() until it returns 0.
- 4.. Call closesocket().

Cheers.

--wjs

.