

Re: TCP Blocking sockets

Source: <http://coding.derkeiler.com/Archive/Delphi/alt.comp.lang.borland-delphi/2004-02/0491.html>

no_email_addy_at_no_email_addy.com

Date: 02/16/04

Date: Mon, 16 Feb 2004 11:55:08 GMT

>The TCP protocol ensures delivery. The TCP protocol has checksums to
>ensure correct data transfer, and will raise errors when incorrect
>data arrives.
>There is no handshaking needed when you are using TCP.

Aye I know that, but I'm streaming live data (GPS etc) so I need to make sure the tcp buffers don't fill up etc due to slow link speeds.

>> Either that, or just limit the amount of data sent per second say – but as
>> thru-put can vary at times (especially on GPRS), it's not an accurate measure of
>> how much data is in transit.
>
>How much data are you going to send here? It sounds like you are going
>to send a whole lot :)

It varies, but yes it can be a couple of MBytes (not a lot by today's standards), which is a lot when it comes to GPRS, and as GPRS can be unreliable, I just wanted to make sure all the streamed data isn't building up too much in the buffers.

>> The number of bytes that's in transit is what I'm really after, or rather how
>> much of what I've sent has been rx'ed at other end – to be able to judge how
>> fast I can push the data into the socket without flooding the thing.
>
>AFAIK this is not something you can get from standard TCP. You'll have
>to move it to the application layer and send some data back.

Yeah, was just wondering whether blocking mode in delphi tcp sockets was waiting till the data had actually been rx'ed at other end or not – as that's not the case, I'll just have to get the other end to tell me when it rx'ed the data (but add's to costs).

>Also, for your original question about blocking vs non-blocking
>sockets:
>The most important difference here is that Delphi generates socket
>events when using non-blocking (asynchronous) sockets. If you use
>blocking sockets, you often also want to run their read/write in a

alt.comp.lang.borland-delphi: Re: TCP Blocking sockets

>thread, as the synchronous read/write blocks execution of the rest of