

## RE: reusing TcpClient object

**Source:** <http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework/2004-11/1573.html>

---

**From:** Kajal Sinha ([KajalSinha\\_at\\_discussions.microsoft.com](mailto:KajalSinha_at_discussions.microsoft.com))

**Date:** 11/25/04

Date: Wed, 24 Nov 2004 21:55:01 -0800

try to create a separate thread for connection which is using a System.Collections.Queue object to identify whether the queue is empty or not if not then Dequeue the topmost item and throw it to the relevant destination using that thread. you can also start a new thread for each message call if required.

"Junaid" wrote:

```
> I've got a situation where I need to send small strings across the
> network to a listening application. Events are fired randomly on the
> client and I need to send one message per event. I've got a separate
> thread running to handle network communication so I don't block the
> rest of the application.
>
> Currently, what I do is create a new TcpClient object and connect to
> the listening application whenever I need to send something. Here's
> the code which does that (managed C++):
>
> // where i need to send
> TcpClient *theSocket = new TcpClient(IPAddress, Port);
> NetworkStream *theStream = theSocket->GetStream();
> theStream->Write(str, 0, len);
> theSocket->Close();
>
> The TcpClient constructor I'm using makes a new object and connects in
> a single step. However, the connection process blocks the thread for a
> few seconds while it connects to the listener. While this does not
> significantly impact the spec, it would be nice to have the messages
> go over instantly rather than waiting a few seconds for the client to
> connect for every message.
>
> Here's what I tried doing to get around the problem:
>
> // created pseudo global variables
> __value struct dummy
> {
> static TcpClient *theSocket;
> static NetworkStream *theStream;
```

```
> /;  
>  
> // in my class' constructor:  
> dummy::theSocket = new TcpClient(IPAddress, Port);  
> dummy::theStream = dummy::theSocket->GetStream();  
>  
> // where i need to send  
> dummy::theStream->Write(str, 0, len);  
>  
> The problem is that this only works the first time around, and on  
> subsequent Write()s, I get the following exception: "Unable to write  
> data to the transport connection."  
>  
> Incidentally, the listening application is currently just a test app I  
> downloaded off some site (no source code unfortunately) which will sit  
> there and listen on a specified port. It simply displays anything it  
> receives. I'm not sure if this app is forcibly disconnecting my client  
> once it has received the data since I'm not actually disconnecting.  
>  
> I'm wondering if my approach is any good or if there's some other way  
> I can achieve what I want. Also, I'm new to socket/network programming  
> so it's entirely possible that what I'm attempting is not only not  
> viable, but also a bit silly to boot.  
>  
> Cheers  
>
```