Re: FIN_WAIT_2 and a socket connect that takes an age to close.

Source:
http://www.tech−archive.net/Archive/Development/microsoft.public.win32.programmer.networks/2004−06/0284.html

---

From: Alan J. McFarlane (*alanjmcf_at_yahoo.com.INVALID*)
Date: 06/14/04

```
Date: Mon, 14 Jun 2004 21:58:42 +0100
```

"Andy Coates" <nothanks@nowhere.com> wrote in message
news:eB2oJ37TEHA.3420@TK2MSFTNGP12.phx.gbl...
[...]
> This normally works fine, but I have one client application (for which I
> can't easily get source) that causes problems on shutdown. This
> particular
> socket takes ~minutes before the queued read operations complete. When I
> run netstat I see that the socket is in the FIN_WAIT_2 state, which I
> understand to mean that it has received the ACK from the client.
>
But not a FIN from it...

In FIN_WAIT_2, a connection is "half−closed". Remember that TCP is full−duplex and allows the connection to be closed in a single direction whilst the other direction continues to pass data. So in this case _your_ application has done SHUTDOWN(WRITE) (at least) and the peer has ack'd the resultant FIN.

But the _peer_ has not done similar: the application there has not done SHUTDOWN(WRITE) or anything similar in effect. The socket on the peer should be in CLOSE_WAIT−−that is, "waiting for a connection termination request from the local user", as RFC 793 puts it.

The pertinent bit of the sequence diagram is (for the server side):

```
ESTABLISHED
   |
   | CLOSE() / send FIN
  \/
FIN_WAIT_1
   |
   | rcv ACK of FIN / (nothing)
  \/
FIN_WAIT_2
   |
```

| rcv FIN / send ACK
\/
TIME_WAIT
|
| timeout=2MSL / delete TCB (TCP Control Block)
\/
TIME_WAIT

> *Why is this socket taking so long to shutdown?*
>
Well, as noted the connection is open and the connection /could feasibly/ remain in that state for a long time, being used for half−duplex comms until the peer application is finished and calls SHUTDOWN(). However many platforms have a timer that kills−off half−closed connections, and particularly so when the connection is idle (no data is flowing on the open half). I'm not sure what Windows does in this regard.

Other possibilities are that the peer application has a two minute timeout. I think you'd have to watch the packets to see what causes the local TCP to move from FIN_WAIT_2 to TIME_WAIT−−−if there's no incoming packet then the local stack has just timed it out.

Note, don't be confuse that putative timeout with the timeout in TIME_WAIT, it is quite separate. People often confuse timeouts in half−closed, with the normal TIME_WAIT timeout. It is 4 minutes in Windows (2000) btw.

−−
Alan J. McFarlane
http://homepage.ntlworld.com/alanjmcf/
Please follow−up in the newsgroup for the benefit of all.