Subject: Socket: SYN_SENT stalled when ipscan connections in CoWork (MT)
Posted by kohait00 on Thu, 26 Aug 2010 14:36:41 GMT
View Forum Message <> Reply to Message

hey guys,

i am having trouble with Socket (btw: there is hardly any API info for it).

to make a long story short: my app has got some kind of ip scan behaviour to find arbitrary devices spread on the network. i use CoWork for it to do this. but as the scan proceeds (intervall of some 30-40 ip's) the 'netstat -a' lists some of them as SYN_SENT, until it piles up to 10 of those. (i know of XP limit of 10 half open connections). but what surprises me the most: i am correctly closing the Socket actually, when it cant find no connection (ref code borrowed from http server in upp).

this problem only arises when using CoWork and firing a lot of those.. when using my WorkQueue (a single thread thread pool) everything is alright, no SYN_SENT pile. even with CoWork, when i fire only 10 of em, wait till all complete, then fire again, it works fine..

now is this a Upp problem or a windows problem? how can it be that SYN_SENT connections pile up while i normally close the sockets?

a testcase is provided. it is my WorkQueueTest modified. to fire some connecitons leftclick multiple times. ip base is set hardcode 192.168.10.170+X, incrementing each click..

EDIT: since the CoWork has CPU_Cores+2 threads, i have 4 of em. so 4 connections are searched for at once. if one thread finishes, it grabs the next connection intent. so there are almost always 4 pending scans, which is less then the 10 limit. so i am kinda lost here..

as soon as the program is quited though, the stalled SYN_SENT disappear..why does the system maintain those half oopened conns when i am not interested in them (closed it).

this is basicly the code that i am using

```
 if(!socket.IsOpen()) {
  if(!ClientSocket(socket, sock_host, sock_port, true, NULL, 0, false)) {
   error = Socket::GetErrorText();
   socket.Close();
   socket.Clear();
   aborted = true;
  }
  else
   socket.Linger(0);
 }
 if(!aborted)
 while(!socket.PeekWrite(1000)) {
  int time = msecs();
  if(time >= end_time) {
```

```
  error = NFormat(t_("%s:%d: connecting to host timed out"), sock_host, sock_port);
  socket.Close();
  socket.Clear();
  break;
 }
 if(progress(time - start_time, end_time - start_time)) {
  aborted = true;
  error = "Aborted";
  socket.Close();
  socket.Clear();
  break;
 }
}
```

...
//some where here the connection can be used, but is open if not aborted, but this is not very important

EDIT: WRONG TESTCASE, down there the right one.

Subject: Re: Socket: SYN_SENT stalled when ipscan connections in CoWork (MT)
Posted by kohait00 on Thu, 26 Aug 2010 16:09:31 GMT
View Forum Message <> Reply to Message

ofcorse the stalled SYN_SENT are not existing devices just to be sure..

try it with the test case scanning wit CoWork (already selected) or using the WorkQueue.. (choose in .h)..

then rapidly firing 20-30 scans by left clicking

Subject: Re: Socket: SYN_SENT stalled when ipscan connections in CoWork (MT)
Posted by kohait00 on Fri, 27 Aug 2010 11:59:32 GMT
View Forum Message <> Reply to Message

WRONG TEST CASE POSTED SOORY, here comes the right one..

File Attachments

1) WorkQueueTest.rar, downloaded 457 times

Subject: Re: Socket: SYN_SENT stalled when ipscan connections in CoWork (MT)
Posted by kohait00 on Fri, 27 Aug 2010 16:00:46 GMT
View Forum Message <> Reply to Message

by the way, this problem is occuring when runnning on windows.
i know of the outgoing connection limitation of windows (10 connections)..but what surprises me the most is that those SYN_SENT remain even though i closesocket normally after time out and Linger(0) is set. so a hard socket close should occure.

i even did a handbreak to only have run a max of 4 concurrent connect attempts by wating for completion of all threads. (if(++q%4 = 0) wq.Finish())

i did some searching but cant find the heart of the problem, there is a lot of crap out there, but nothing that directly adresses this one..

http://docs.softinventive.com/En/Total_Network_Inventory/FAQ
Quote:
Q: Where do I install the program on - a server or a workstation?

A: Either server or workstation can run Total Network Inventory. It is just a matter of usage convenience, because it's not a client-server application and you need to have access to the graphical console of the computer you install it on, either directly or using some remote desktop utility. Besides, if you run it under domain admin account, you will be able to scan all computers "as current user", otherwise you would need to specify domain admin credentials explicitly.

However take note that if you install the program on Windows XP (starting with SP2), Windows Vista or Windows 7, and if there are many scan threads launched simultaneously, there may be issues with connections to remote computers. This is due to a restriction on the maximum number of TCP half-open connections (connection attempts, SYN_SENT socket state) existing in the mentioned Windows versions, which doesn't allow more than 10 outbound connections to be in this state at a time. After reaching this limit, all other connections in the system (including those executed by this program) are queued and may reach their timeout, thus producing inconsistent results. This issue is also known as "Event 4226 issue", because reaching the limitation produces a record in the System Event Log with EventID 4226. Windows XP SP0/SP1, Windows 2000 Professional and all Windows Server systems don't have such limitation. So in general case we suggest installing the program on a server operating system.

http://www.microsoft.com/products/ee/transform.aspx?ProdName
=Windows%20Operating%20System&ProdVer=5.1.2600.2180&
EvtID=4226&EvtSrc=Tcpip&LCID
Quote:
Explanation

The TCP/IP stack in Windows XP with Service Pack 2 (SP2) installed limits the number of concurrent, incomplete outbound TCP connection attempts. When the limit is reached, subsequent connection attempts are put in a queue and resolved at a fixed rate so that there are only a limited number of connections in the incomplete state. During normal operation, when programs are connecting to available hosts at valid IP addresses, no limit is imposed on the

number of connections in the incomplete state. When the number of incomplete connections exceeds the limit, for example, as a result of programs connecting to IP addresses that are not valid, connection-rate limitations are invoked, and this event is logged.

such as viruses and worms, spread to uninfected computers. Malicious programs often attempt to reach uninfected computers by opening simultaneous connections to random IP addresses. Most of these random addresses result in failed connections, so a burst of such activity on a computer is a signal that it may have been infected by a malicious program.

Connection-rate limitations may cause certain security tools, such as port scanners, to run more slowly.

but why the hell is an actively closed connection considered half-opened? or does the system impose another timeout in this, which has nothing to do with the select timeout??

Subject: Re: Socket: SYN_SENT stalled when ipscan connections in CoWork (MT)
Posted by kohait00 on Tue, 31 Aug 2010 08:05:53 GMT
View Forum Message <> Reply to Message

anyone had the same problem?

Subject: Re: Socket: SYN_SENT stalled when ipscan connections in CoWork (MT)
Posted by kohait00 on Fri, 10 Sep 2010 09:09:01 GMT
View Forum Message <> Reply to Message

i'm really struggeling with this one, any one help?
