Configure the max limit for concurrent TCP connections

March 6, 2004 by Snakefoot | 14 Comments

To keep the TCP/IP stack from taking all resources on the computer, there are different parameters that control how many connections it can handle. If running applications that are constantly opening and closing connections (P2P), or are providing a service which many tries to connect to at the same time (Web-server like IIS), then one can improve the performance of these applications by changing the restriction limits.

There is a parameter that limits the maximum number of connections that TCP may have open simultaneously.

> [HKEY_LOCAL_MACHINE \System \CurrentControlSet \Services \Tcpip \Parameters]
> TcpNumConnections = 0x00fffffe (Default = 16,777,214)
>
> Note a 16 Million connection limit sounds very promising, but there are other parameters (See below), which keeps us from ever reaching this limit.

When a client makes a connect() call to make a connection to a server, then the client invisible/implicit bind the socket to a local dynamic (anonymous, ephemeral, short-lived) port number. The default range for dynamic ports in Windows is 1024 to 5000, thus giving 3977 outbound concurrent connections for each IP Address. It is possible to change the upper limit with this DWORD registry key:

> [HKEY_LOCAL_MACHINE \System \CurrentControlSet \Services \Tcpip \Parameters]
> MaxUserPort = 5000 (Default = 5000, Max = 65534)
>
> Note it is possible to reserve port numbers so they aren't used as dynamic ports in case one have a certain application that needs them. This is done by using the ReservedPorts (MS KB812873) setting.
>
> Note Vista changes the default range from 1024-5000 to 49152-65535, which can be controlled with the dynamicport setting using netsh. More Info MS KB929851.
>
> More Info The Cable Guy - Ephemeral, Reserved, and Blocked Port Behavior

> More Info MS KB196271
> More Info MS KB319502
> More Info MS KB319504
> More Info MS KB328476
> More Info MS KB836429

For each connection a TCP Control Block (TCB - Data structure using 0.5 KB pagepool and 0.5 KB non-pagepool) is maintained. The TCBs are pre-allocated and stored in a table, to avoid spending time on allocating/deallocating the TCBs every time connections are created/closed. The TCB Table enables reuse/caching of TCBs and improves memory management, but the static size limits how many connections TCP can support simultaneously (Active + TIME_WAIT). Configure the size of the TCB Table with this DWORD registry key:

> [HKEY_LOCAL_MACHINE \System \CurrentControlSet \Services \Tcpip \Parameters]
> MaxFreeTcbs = 2000 (Default = RAM dependent, but usual Pro = 1000, Srv=2000)

To make lookups in the TCB table faster a hash table has been made, which is optimized for finding a certain active connection. If the hash table is too small compared to the total amount of active connections, then extra CPU time is required to find a connection. Configure the size of the hash table with this DWORD registry key (Is allocated from pagepool memory):

> [HKEY_LOCAL_MACHINE \System \CurrentControlSet \services \Tcpip \Parameters]
> MaxHashTableSize = 512 (Default = 512, Range = 64-65536)
>
> Note Microsoft recommends for a multiprocessor environment, that the value should not be higher than the maximum amount of concurrent connections (MaxFreeTcbs), also if multiprocessor then it might be interesting to look at the registry-key NumTcbTablePartitions (Recommended value CPU-count multiplied by 4).
>
> More Info MS KB151418
> More Info MS KB224585

If having allocated a 1000 TCBs then it doesn't mean that one will be able to have a 1000 active connections. Especially if the application is quickly opening and closing connections, because after a connection is "closed" it enters the state TIME_WAIT, and will continue to occupy the port number for 4 minutes (2*Maximum Segment Live, MSL) before it is actually removed. This behavior is specified in RFC 793, and prevents attempts to reconnect to the same party, before the old socket is recognized as closed at both sides. It is possible to change

how long a socket should be in TIME_WAIT state before it can be re-used freely:

> [HKEY_LOCAL_MACHINE \System \CurrentControlSet \services \Tcpip \Parameters]
> TcpTimedWaitDelay = 120 (Default = 240 secs, Range = 30-300)
>
> More Info MS KB137984
> More Info MS KB149532
> More Info MS KB832954
>
> Note with Win2k the reuse of sockets have been changed, so when reaching the limit of more than 1000 connections in TIME-WAIT state, then it starts to mark sockets that have been in TIME_WAIT state for more than 60 secs as free. It is possible to configure this limit:
>
> [HKEY_LOCAL_MACHINE \System \CurrentControlSet \services \Tcpip \Parameters]
> MaxFreeTWTcbs = 1000 (Default = 1000 sockets)
>
> Note with Win2k3 SP1 the reuse of sockets have been changed, so when it has to re-use sockets in TIME_WAIT state, then it checks whether the other party is different from the old socket. Eliminating the need to fiddle with (TcpTimedWaitDelay) and (MaxFreeTWTcbs) any more.

If using an application protocol that doesn't implement timeout checking, but relies on the TCPIP timeout checking without specifying how often it should be done, then it is possible to get connections that "never" closes, if the remote host disconnects without closing the connection properly. The TCPIP timeout checking is by default done every 2 hour, by sending a keep alive packet. It is possible to change how often TCPIP should check the connections (Affects all TCPIP connections):

> [HKEY_LOCAL_MACHINE \System \CurrentControlSet \services \Tcpip \Parameters]
> KeepAliveTime = 1800000 (Default = 7,200,000 milisecs)
>
> More Info MS KB140325

When data is sent/received the data is copied back and forth to non-paged pool memory for buffering. If there are many connections receiving/sending data, then it is possible to exhaust the non-paged pool memory. The max size of the non-paged pool buffer allocated for each connection is controlled by MaxBufferredReceiveBytes or TCPIP Receive Window depending on which is smallest. More Info MS KB296265

Note if using the Professional/Home edition of Windows then it is very

likely that it is crippled (By Microsoft) not to handle many concurrent TCP connections. Ex. Microsoft have officially stated that the backlog limit is 5 (200 when Server), so the Professional edition is not able to accept() more than 5 new connections concurrently. More Info MS KB127144

Note even if having optimized Windows to handle many concurrent connections, then connections might still be refused when reaching a certain limit, in case a NAT-Router/Firewall is placed infront of it, which is unable to handle so many concurrent connections.

Note if having activated SYN-Attack-Protection (Enabled by default in Win2k3 SP1) or installed WinXP SP2, a limit is introduced on how many connection attempts (half-open) one can make simultaneously (XP SP2 & Vista = 10; Vista SP2 = no limit). This will keep worms like blaster and sasser from spreading too fast, but it will also limit other applications that creates many new connections simultaneously (Like P2P).

> *EventID 4226: TCP/IP has reached the security limit imposed on the number of concurrent TCP connect attempts*
>
> More Info www.LvlLord.de
>
> Windows Vista SP2 removes the limit again, but it can be enabled with the following DWORD registry setting:
>
> [HKEY_LOCAL_MACHINE \SYSTEM \CurrentControlSet \Services \Tcpip \Parameters]
> EnableConnectionRateLimiting = 1
>
> More Info MS KB969710

Related No more than 10 connections to a remote computer

Updated: 22 May 2009

Tags:

crippled , ephemeral , tcpip

Category:

Windows NT4 / WinNT4 / Network
Windows XP / WinXP / Network
Windows 2000 / Win2k / Network
Windows 2003 / Win2k3 / Network

Comments:

twl
7 March 2009 at 1:32

Quite some article there. Hell you lost me couple times on definitions, but considering I did understand the point of article and use of it this is very god information to have.
As it seems ever since tcpip limitations the boxes has been suffering huge lags in connecting/ons while now it seems that these methods compined and bottom level (auto detections) disabled windows actually could work as some sort of decent server.

Reply

Talgat-_-
5 January 2010 at 17:00

hi guys! im from kz... my provider is megaline.kz i have problems with my connection speed also high ping in online games! I tried to call to operator but in my town cant help me! speed is very low until morning to evening. but sometimes in the night speed goes up.. i think that they connect to one internet port many peoples because of this speed is low :(how can i protect my connection or my port? plz help me! sorry for my english... imho i can explain my situation write to my mailbox taxa_91_virtual@inbox.ru thx

Reply

James Watt
27 February 2010 at 4:51

I know this information is a bit dated, but I have to give you credit for such an in depth analysis of the Windows TCP/IP stack. Any changes with Windows 7? Also, do 64bit Windows operating systems have different values? Thanks.

Reply

Snakefoot
2 March 2010 at 0:32

> James Watt wrote:
> Any changes with Windows 7? Also, do 64bit Windows operating systems have different values? Thanks.

Vista/Win7 introduces the Next Generation TCP/IP stack, that requires less fine tuning. Registry settings like TcpWindowSize, MaxFreeTcbs, MaxHashTableSize, NumTcbTablePartitions are now ignored. But for outbound connections there are still "only" 65534 port numbers available.
More Info Performance Tuning Guidelines for Windows Server 2008
More Info TCP/IP Registry Values for Vista and Win2k8

Reply

Emrah
2 August 2010 at 17:52

Hi,
My company hasgot a 50 terminal server licence but only 5 clients log on at the same time. I check the terminal server configuration->RDP-TCP->properties->network adapter->Maximum connections is 5 I can't change it. Could you please help me?

Reply

Yehuda
26 May 2011 at 18:27

Very nicely written
ill try to apply it on my production servers
Thanks
i

Reply

Tasneem
30 May 2012 at 10:49

Brilliant information. Thinking of making these changes on our servers for our clients. Thank You!

Reply

SUNDAR
13 September 2013 at 8:51

Hi,
My company hasgot a 50 terminal server licence but only 10 clients log on at the same time. I check the terminal server configuration->RDP-TCP->properties->network adapter->Maximum connections is 10 I can’t change it. Could you please help me

Reply

AndrewK
7 November 2013 at 4:07

Hi,
Very nicely written article! Recently, all our computers were upgraded to Windows 7. One particular application that runs makes multiple tcp connections to a another application when the connection is made! Then application has to be closed and opened again and sometimes it still makes more than one connection. Any suggestion?

Reply

Duder
21 January 2014 at 18:59

This article is totally useless and doesn't give any good info on what most people are trying to figure out: the max limit of connections from other computers. All these numbers in the thousands are irrelevant. Windows XP limits it to 10, and Win7 to 20, and it seems some people don't like to admit they don't know how to do anything about it, or they confuse it with the half-open connection garbage.

Reply

Lame
18 May 2015 at 4:36

Duder you moron, you are replying to an article that was written 10 years ago, obviously its outdated and you're mentioning about Win7?
And I'm also a moron for replying to your comment a year after.
But I just had to.

ho909
4 December 2015 at 1:25

MaxFreeTWTcbs = 1000 - ????
MaxFreeTWTcbs = 65000 !

Reply

Paul
28 February 2016 at 16:22

Hello- I am really desperate as we built a chat application that uses websocket and stomp protocol, after about 1500 connections, users cannot connect anymore.
I believe it has to do with some limit set to the number of concurrent open to connections by the OS, but I am not sure how to increase this limit or to determine if this really is the cause.
We are running our application on Windows 2008 server, the app is written in Java and hosted in Tomcat.
Any help would be very much appreciated.
Thanks in advance

Reply

Snakefoot
5 March 2016 at 10:56

TomCat/Apache can work in two modes. BIO - One thread per connection. NIO - multiple connections for each thread.
Setup your connector in server.xml to use NIO. Http11Nio2Protocol (Tomcat 8+) or Http11NioProtocol (Tomcat 6+) and configure maxConnections.
For a chat-application consider to use small connector-buffers (socket.appWriteBufSize="1024" and socket.appReadBufSize="1024") to lower footprint for each chat-connection.

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website

Post Comment

Search

Related Posts

Improve latency for TCP by not waiting for Push flag

Configure host name resolution order in Windows NT

Description of host name resolution order

Configure the priority of protocols bound to the network

Diagnose performance issues with network hardware

Recent Posts

Disable IPv6 imaginary tunnel network interfaces

Encrypted backup to OneDrive or DropBox

Description of soft and hard page faults

Windows 10 Upgrade with black screen

WordPress 4.2 Upgrade

Meta

Log in

Entries RSS

Comments RSS