

Forum:[Hack Forums](#)

Thread: [Make your Internet bandwidth Rapid \[No Download\]](#)

Beilinder - January 11, 2013, 11:42 am

First of all, this won't work on Windows XP, as far as I know.

Anything you do is YOUR responsibility!

This is basically for Windows Vista and Windows Server 2008 but most of the method can be applied to Windows 7.

TCP/IP tweaks, patches & Manual registry hacks

NOTE: For people who wants to skip the technical part and wants the optimal Internet setting download the below patch which does everything except for "gaming tweak" (mentioned at the end of the post)

Quote:https://#.com/#!download|457p1|257722757|Vista_TCPIP_Patch.zip|1|0|0

Windows Vista / 2008 Tweaks

Windows Vista introduces a number of new features to the TCP/IP stack, including CTCP, and TCP Window Auto-Tuning. This new implementation works much better by default than previous Windows versions with broadband internet connections, and is able to adjust the RWIN value on the fly, depending on the BDP (bandwidth-delay product). This, however, introduces some problems with older routers and restricts the user from tweaking some of the TCP/IP parameters. Still, there is always some room for improvement, and this article explains known the tweakable TCP/IP parameters.

To enter some of the commands below, you will need to run "elevated" command prompt. To do so, click the Start icon > Run > type: cmd , then click CTRL+SHIFT+ENTER. Alternatively, you can navigate to Start > All Programs > Accessories > right-click Command Prompt and choose "Run as Administrator"

Check the TCP/IP state

To check the current status of the Vista TCP/IP tweakable parameters, in elevated command prompt type the following command:

```
netsh int tcp show global
```

The settings, as well as their default and recommended state are explained below. The two most important tweakable parameters are "Auto-Tuning Level" and "Congestion Control Provider".

TCP Auto-Tuning

To turn off the default RWIN auto tuning behavior, (in elevated command prompt) type:

```
netsh int tcp set global autotuninglevel=disabled
```

The default auto-tuning level is "normal", and the possible settings for the above command are:

disabled: uses a fixed value for the tcp receive window. Limits it to 64KB (limited at 65535).
highlyrestricted: allows the receive window to grow beyond its default value, very conservatively
restricted: somewhat restricted growth of the tcp receive window beyond its default value
normal: default value, allows the receive window to grow to accommodate most conditions
experimental: allows the receive window to grow to accommodate extreme scenarios (not recommended, it can degrade performance in common scenarios, only intended for research purposes. It enables RWIN values of over 16 MB)

Our recommendation: normal (unless you're experiencing problems).

If you're experiencing problems with your NAT router or SPI firewall, try the "restricted", "highlyrestricted", or even "disabled" state.

Notes:

- Reportedly, many home NAT routers with a SPI firewall may have problems with enabled tcp auto-tuning in it's "normal" state, resulting in slow speeds, packet loss, and general reduced network performance.
- auto-tuning also causes problems with older routers that do not support TCP Windows scaling.
- netsh set commands take effect immediately after executing, there is no need to reboot.
- sometimes when using "normal" mode and long lasting connections (p2p software / torrents), tcp windows can get very large and consume too much resources, if you're experiencing problems try a more conservative setting.

If you're experiencing problems with Auto-Tuning, see also:

MS KB 835400 - email issues

MS KB 934430 - network connectivity behind firewall problems

MS KB 940646 - 3G WWAN throughput issues

MS KB 929868 - web browsing issues

MS KB 932170 - slow network file transfer

The above are the M\$ Knowledge based articles. To view them input the following in your browser
Quote:<http://support.microsoft.com/kb/<insert number here>>

Example to view MS KB 83540 type in:

Quote:<http://support.microsoft.com/kb/935400>

Compound TCP - Improve throughput

The traditional slow-start and congestion avoidance algorithms in TCP help avoid network congestion by gradually increasing the TCP window at the beginning of transfers until the TCP Receive Window boundary is reached, or packet loss occurs. For broadband internet connections that combine high TCP Window with higher latency (high BDP), these algorithms do not increase the TCP windows fast enough to fully utilize the bandwidth of the connection.

Compound TCP (CTCP) is a newer method, available in Vista and Server 2008 (there is also a hotfix available for XP/2003). CTCP increases the TCP send window more aggressively for broadband connections (with large RWIN and BDP). CTCP attempts to maximize throughput by monitoring delay variations and packet loss. It also ensures that its behavior does not impact other TCP connections negatively.

By default, Vista has CTCP turned off, and Server 2008 turned on. Turning this option on can significantly increase throughput.

To enable CTCP, in elevated command prompt type:

```
netsh int tcp set global congestionprovider=ctcp
```

To disable CTCP:

```
netsh int tcp set global congestionprovider=none
```

Possible options are: ctcp, none, default (restores the system default value).

Recommended setting: ctcp

It is better to use this newer generation CTCP congestion control algorithm for most broadband connections, I recommend it being turned on.

ECN Capability

ECN (Explicit Congestion Notification) is a mechanism that provides routers with an alternate method of communicating network congestion. It is aimed to decrease retransmissions. In essence, ECN assumes that the cause of any packet loss is router congestion. It allows routers experiencing congestion to mark packets and allow clients to automatically lower their transfer rate to prevent further packet loss. ECN is disabled by default in Vista, as it is possible that it may cause problems with some older routers that do not support this feature.

To check whether your router supports ECN, you can use the Microsoft Internet Connectivity Evaluation Tool.

Quote: <http://www.microsoft.com/windows/using/tools/igd/default.mspx>

The results will be displayed under "Traffic Congestion Test".

To enable ECN, in elevated command prompt type:

```
netsh int tcp set global ecncapability=enabled
```

Possible settings are: enabled, disabled, default (restores the state to the system default).

The default state is: disabled

Our recommendation: disabled

RSS - Receive-side Scaling

The receive-side scaling setting enables parallelized processing of received packets on multiple processors, while avoiding packet reordering. It avoids packet reordering by separating packets into "flows", and using a single processor for processing all the packets for a given flow. Packets are separated into flows by computing a hash value based on specific fields in each packet, and the resulting hash values are used to select a processor for processing the flow. This approach ensures that all packets belonging to a given TCP connection will be queued to the same processor, in the same order that they were received by the network adapter.

To set RSS:

```
netsh int tcp set global rss=enabled
```

Possible rss settings are: disabled, enabled, default (restores rss state to the system default).

Default state is: enabled

Recommended: enabled (if you have 2 or more processor cores and a NIC that can handle RSS)

Manually tuning Registry Parameters

Many of the registry keys tuning TCP/IP parameters from previous Windows versions no longer work in Vista and Server 2008. Below is a list of the few we've confirmed to still work. Note that for changes to these settings to take effect the computer needs to be rebooted. As always, a registry backup is recommended if making any changes, and some proficiency in using regedit is required.

In regedit (Start icon > Run > type: regedit while logged in as administrator), you can navigate and edit the following keys.

MTU (Maximum Transmission Unit) - the maximum packet size.

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{...}\
MTU=1500 (DWORD, entry does not exist by default)

The {...} part of the above path is the unique identifier of your network adapter. You can recognize the correct adapter by looking at it's IP address, if obtaining IP automatically labeled by: DhcpIPAddress=192.168.x.x text value, for example.

We recommend leaving this at default, unless you want to lower it. Vista uses the largest possible packet size for the underlying network by default.

Note: In some test environments, the correct MTU entry may be offset by 8. The 8 offset seems to coincide with the size of the PPPoE overhead. Check the result with the TCP Analyzer.

TCP 1323 Options

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\
Tcp1323Opts=1 (DWORD, entry created automatically by Windows when you run the "netsh int tcp set global autotuninglvl=..." command, set to 0 by default).

Setting this seems to have no effect, since auto-tuning uses the TCP 1323 scale factor and changes it on the fly, disregarding this setting. Additional testing may be required to determine it's effect if auto-tuning is turned off. Setting it to 1 is best for broadband connections.

NetDMA

NetDMA enables support for advanced direct memory access. In essence, it provides the ability to more efficiently move network data by minimizing CPU usage. NetDMA frees the CPU from handling memory data transfers between network card data buffers and application buffers by using a DMA engine.

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters
EnableTCPA=1 (DWORD, 1 to enable, 0 to disable NetDMA. Value not present by default in Vista)

Recommended setting is 1, a new DWORD value may need to be created if not already present in the registry.

DefaultTTL

TTL can be safely left alone in many cases. It is a limit to the time and number of hops/routers a

packet will travel before being discarded. A number that's too small risks packets being discarded before reaching their destination. A number that's too large (over 128) will cause delay in when lost IP packets are discarded.

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters
DefaultTTL=64 (DWORD, set to a decimal value between 32 and 128. Recommended: 64)

TcpMaxDataRetransmissions

Determines how many times unacknowledged data (non-connect segment) is retransmitted before TCP aborts the connection. The retransmission timeout is doubled with each successive retransmission on a connection. It is reset when responses resume.

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters
TCPMaxDataRetransmissions=7 (DWORD, recommended: between 3 and 10, default registry value 255, default 5 in documentation)

SynAttackProtect

This undocumented setting provides protection against SYN denial of service (DoS) attacks. When enabled, connections timeout sooner if SYN attack is detected. When set at 1, TCPMaxDataRetransmissions can be lowered further.

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters
SynAttackProtect=1 (DWORD, recommended: 1, not present in registry by default)

TcpTimedWaitDelay (port allocation)

Short lived (ephemeral) TCP/IP ports above 1024 are allocated as needed by the OS. The default Vista values have improved from previous Windows versions, and are usually sufficient under normal load. However, in some instances under heavy load it may be necessary to adjust the settings below to tweak the availability of user ports requested by an application.

If the default limits are exceeded under heavy loads, the following error may be observed: "address in use: connect exception". By default under Vista (when the values are not present in the registry), the OS can allocate up to 16384 ephemeral ports above port 1024, and the OS waits for 120 seconds before reclaiming ports after an application closes the TCP connection. This is a considerable improvement over older Windows versions. However, if necessary, the following registry values can be added/edited:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters
MaxUserPort=65535 (DWORD, not in the registry by default. Recommended: leave at default, or use a number above 16384 up to 65535 decimal as necessary) - maximum number of ports to use. 1024 is automatically subtracted from entered value to allow for reserved ports under 1024.
TcpTimedWaitDelay=30 (DWORD, 0xffffffff in registry by default. Recommended: 30 decimal, denoting 30 seconds) - time to wait before reclaiming ports, in seconds. Default time before reclaiming ports, if value is at 0xffffffff or not present in the registry is 120 seconds. Just reducing the delay is often sufficient without changing MaxUserPort, as it allows for reusing ports more efficiently.

Ephemeral ports can be checked and changed using netsh as well.

To query the current values, in command prompt, type:

netsh int ipv4 show dynamicportrange tcp (for UDP, use the same command, replacing only "tcp" with "udp" at the end)

To set both the starting, and max user port using netsh, in elevated command prompt run:
netsh int ipv4 set dynamicportrange protocol=tcp start=1025 num=64511 (start=NNN denoting the starting port, and num=NNN denoting the number of ports)

Notes:

By default, dynamic ports are allocated between ports 49152 and 65535 (for a total of 16384 ephemeral ports).

Using netsh allows to set both the starting port and port range. Editing the Registry allows for setting the port range, and the starting port is fixed at 1025. Deleting the MaxUserPort registry entry (or setting it to a value outside the allowed range) causes the OS to revert to using the default values.

Some system processes can install port filters to block certain port ranges. If ephemeral ports run into these filtered port ranges, TCP/IP applications will be unable to bind to any ports.

QoS Reserved Bandwidth

As with Windows XP, network adapters have a "QoS Packet Scheduler" enabled by default, which reserves 20% of bandwidth by default for QoS applications that request priority traffic. Note this only has effect in the presence of running QoS applications that request priority traffic. Registry value is undocumented for the Vista version of Windows. To customize this setting, in the Windows Registry:

HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\Psched

NonBestEffortLimit=0 (DWORD, not present in the registry by default. Recommended: 0, possible values between 0 and 100) - indicates the percentage value of reserved bandwidth for QoS applications. Set to 0 to disable.

Notes: This tweak applies only to Windows versions that have QoS Packet Scheduler enabled. It will ONLY have effect in the presence of running QoS applications.

Gaming Tweak in next post.

Gaming Tweak - Disable Nagle's algorithm

The tweak below allows for tweaking or disabling Nagle's algorithm. Disabling "nagling" allows for very small packets to be transferred immediately without delay. Note that disabling Nagle's algorithm is only recommended for some games, and it may have negative impact on file transfers/throughput. The default state (Nagling enabled) improves performance by allowing several small packets to be combined together into a single, larger packet for more efficient transmission. While this improves overall performance and reduces TCP/IP overhead, it may briefly delay transmission of smaller packets. Keep in mind that disabling Nagle's algorithm may have some negative effect on file transfers, and can only help reduce delay in some games. To implement this tweak, in the registry editor (Start>Run>regedit) find:

This setting configures the maximum number of outstanding ACKs in Windows XP/2003/Vista/2008:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{NIC-id}

There will be multiple NIC interfaces listed there, for example:
{1660430C-B14A-4AC2-8F83-B653E83E8297}. Find the correct one with your IP address listed. Under this {NIC-id} key, create a new DWORD value:

TcpAckFrequency=1 (DWORD value, 1=disable, 2=default, 2-n=send ACKs if outstanding ACKs before timed interval. Setting not present by default).

For gaming performance, recommended is 1 (disable). For pure throughput and data streaming, you can experiment with values over 2. If you try larger values, just make sure $\text{TcpAckFrequency} \times \text{MTU}$ is less than RWIN, since the sender may stop sending data if RWIN fills without acknowledgement.

Also, find the following key (if present):

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\Parameters

Add a new DWORD value:

TCPNoDelay=1 (DWORD value, 0 to enable Nagle's algorithm, 1 to disable, not present by default)

To configure the ACK interval timeout (only has effect if nagling is enabled), find the following key:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{NIC-id}

TcpDelAckTicks=0 (DWORD value, default=2, 0=disable nagling, 1-6=100-600 ms). Note you can also set this to 1 to reduce the nagle effect from the default of 200ms without disabling it.

Notes:

Reportedly, the above gaming tweak (disabling nagle's algorithm) can reduce WoW (World of Warcraft) latency by almost half!

XP/2003 needs hotfix or SP2 for it to work (MS KB 815230)

Vista needs hotfix or SP1 for it to work (MS KB 935458)

Hope you found this tutorial useful :)

letsPlayUp - January 12, 2013, 1:32 pm

Will try this.

Bookmarking for the moment.

I'll comment back when this totally works.

bastiandageek - January 15, 2013, 12:10 am

crazy thanks for taking the time to make this

Beilinder - January 23, 2013, 10:39 am

Glad you guys liked it :) Good luck with this!