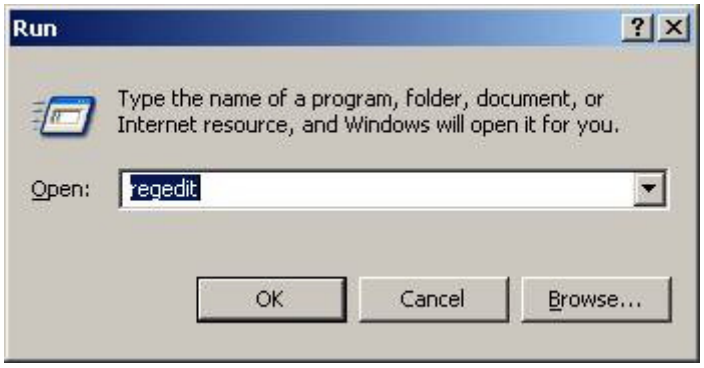
Step-by-step instructions for tuning TCP under Windows XP

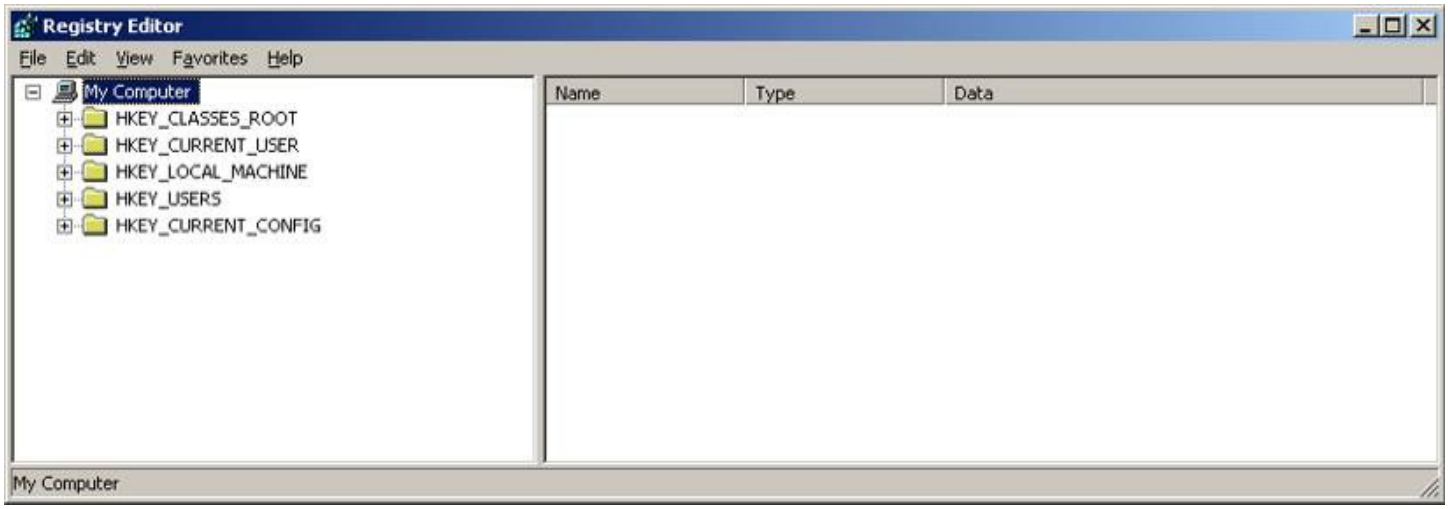
Editing the registry

Be extremely careful, because it can be very difficult to restore anything you didn't mean to change. Consider backing up the registry before you start and keeping detailed notes of everything you change, especially prior values.

1.) Open regedit by clicking on "Start -> Run", typing in regedit and clicking "OK"



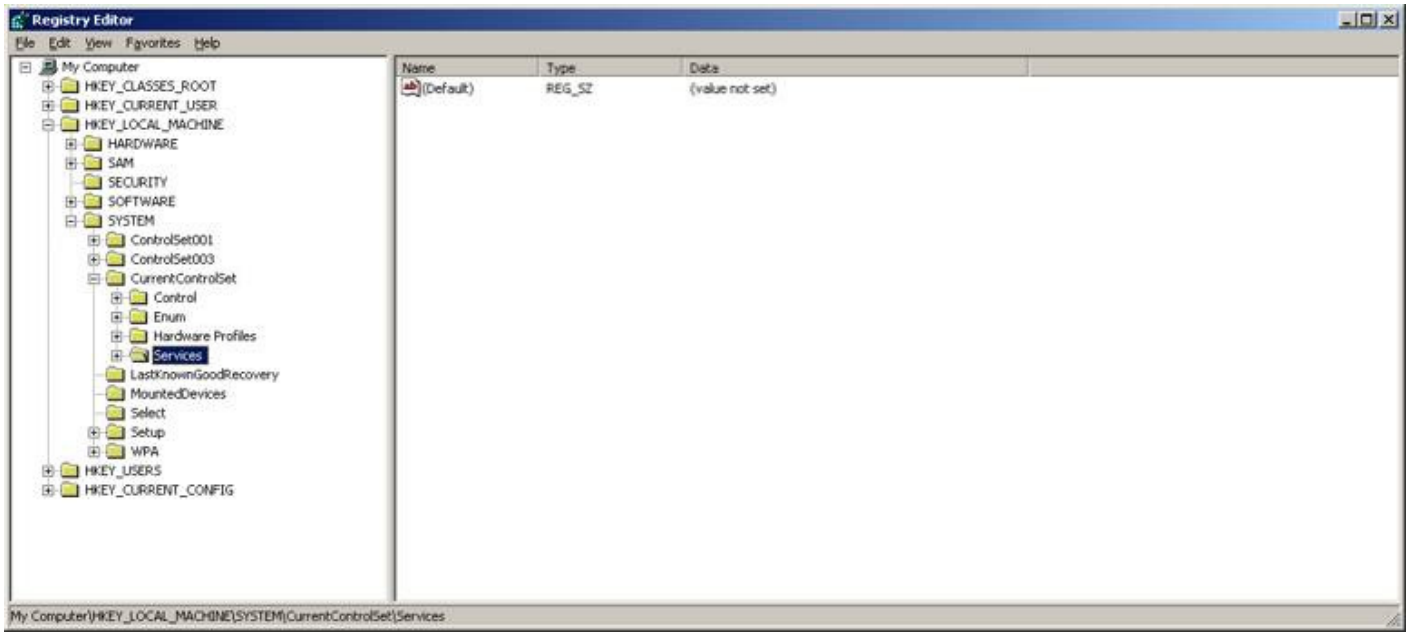
The registry editor will open a window:



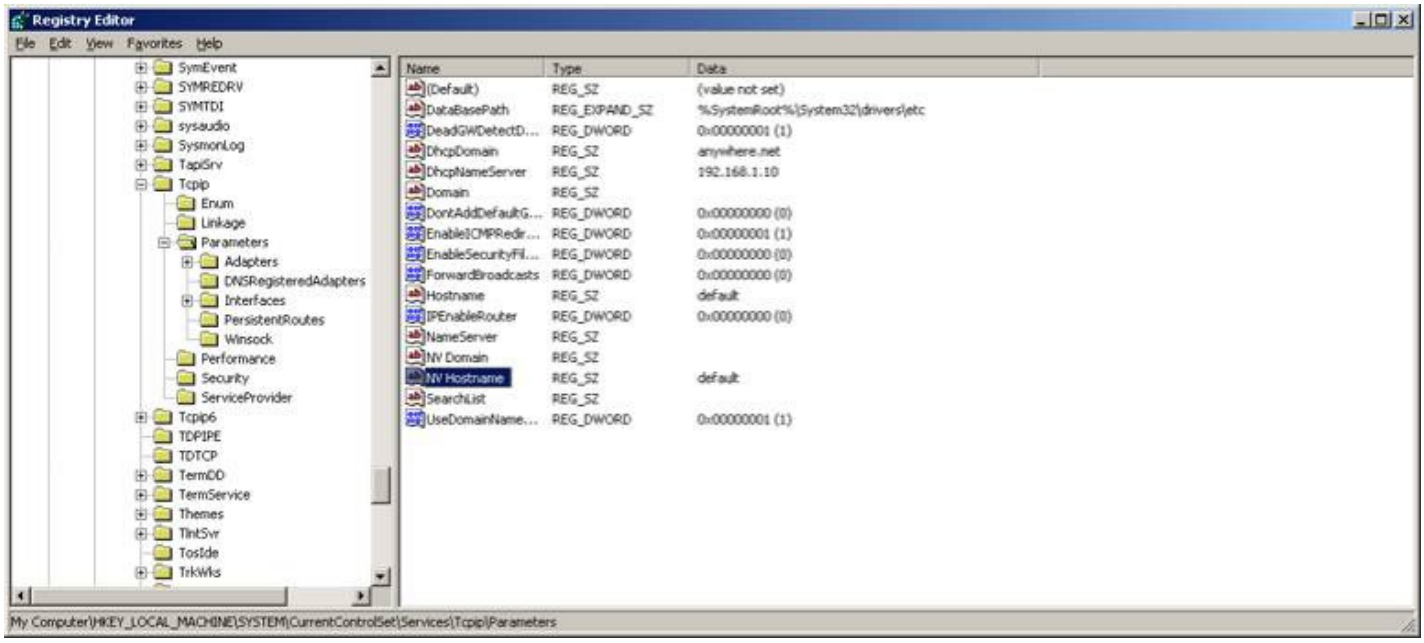
2.) Browse to the TCP/IP registry keys under
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters

by expanding successive folders on the left panel and scrolling down as needed.

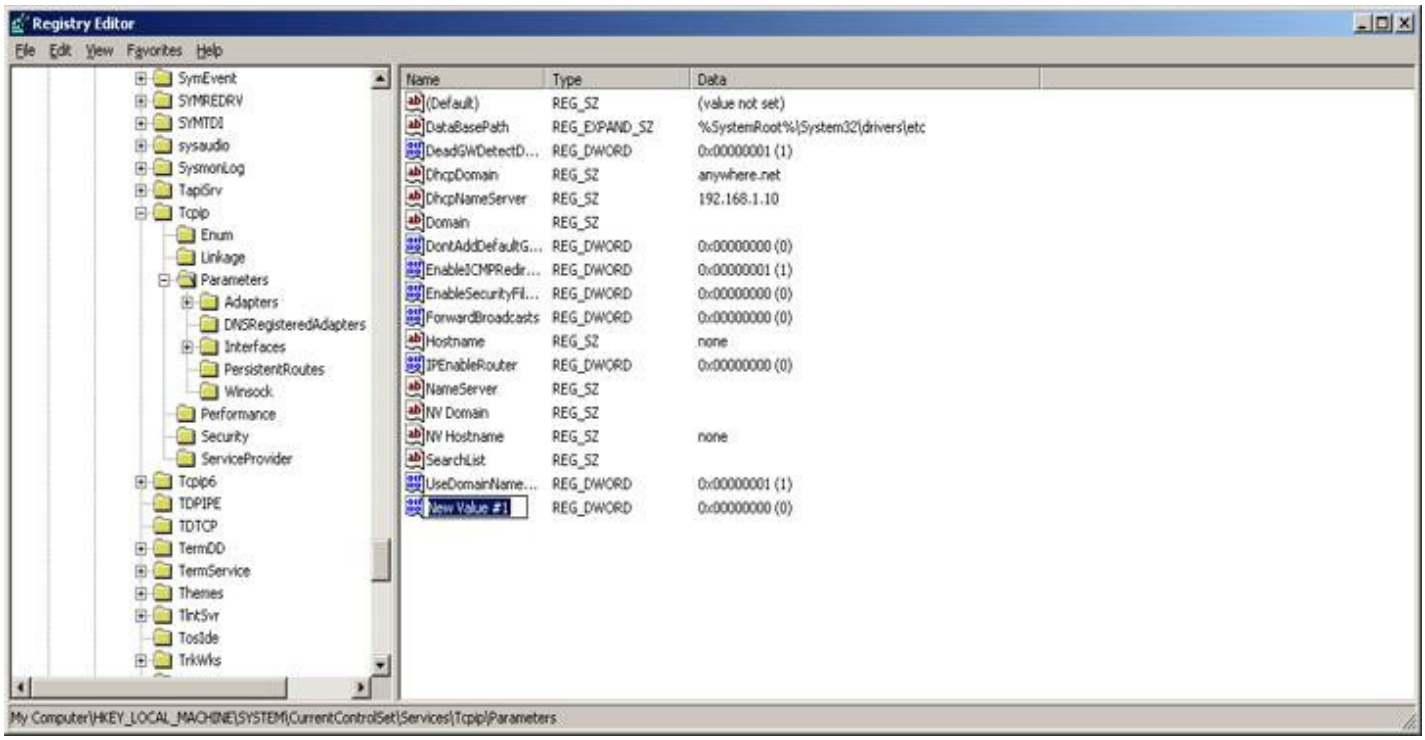
Here is the view after browsing to HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services:



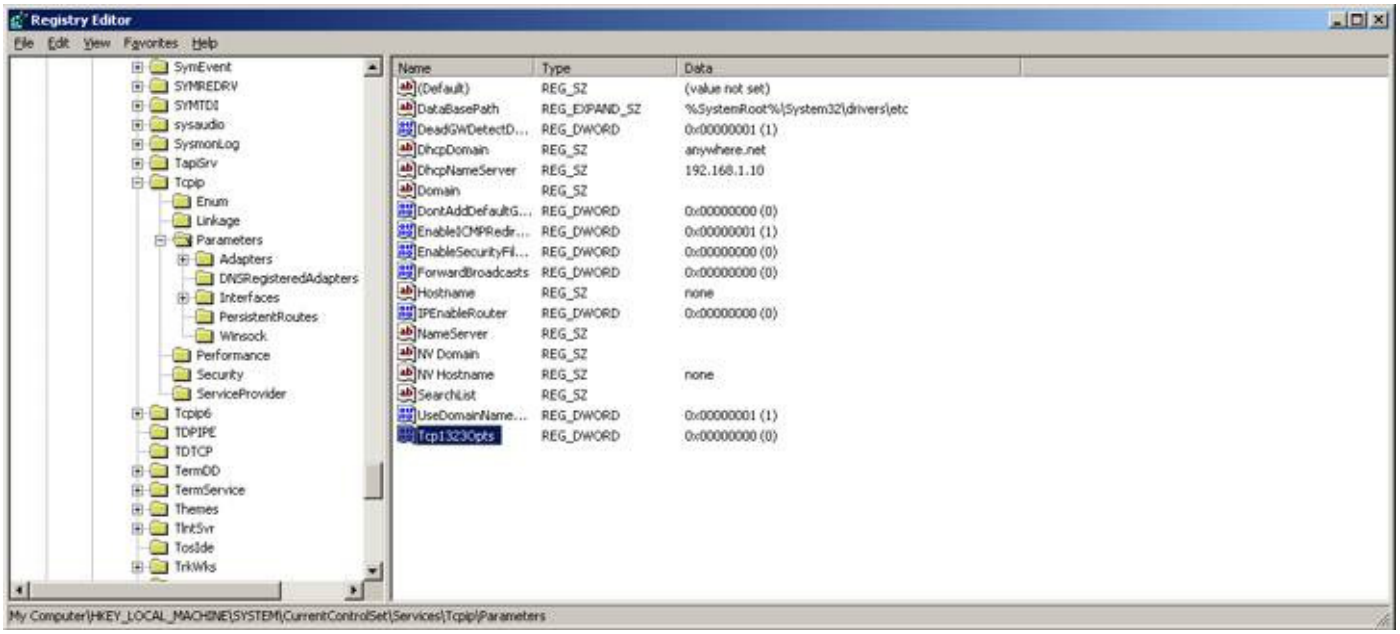
Here is the view after browsing to HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters:



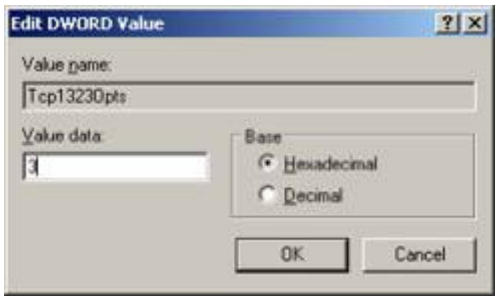
3.) Adjust the registry keys to tweak your network performance. To tune TCP/IP you will need to add specific DWORD values, unless your system has been tuned previously. To add a value, right click in the right window showing the key values and select "New -> DWORD value".



4). Rename this "New Value" to the option you wish to set. The first and most important is to enable RFC 1323 features by editing the new option to "Tcp1323Opts" and hitting Enter.

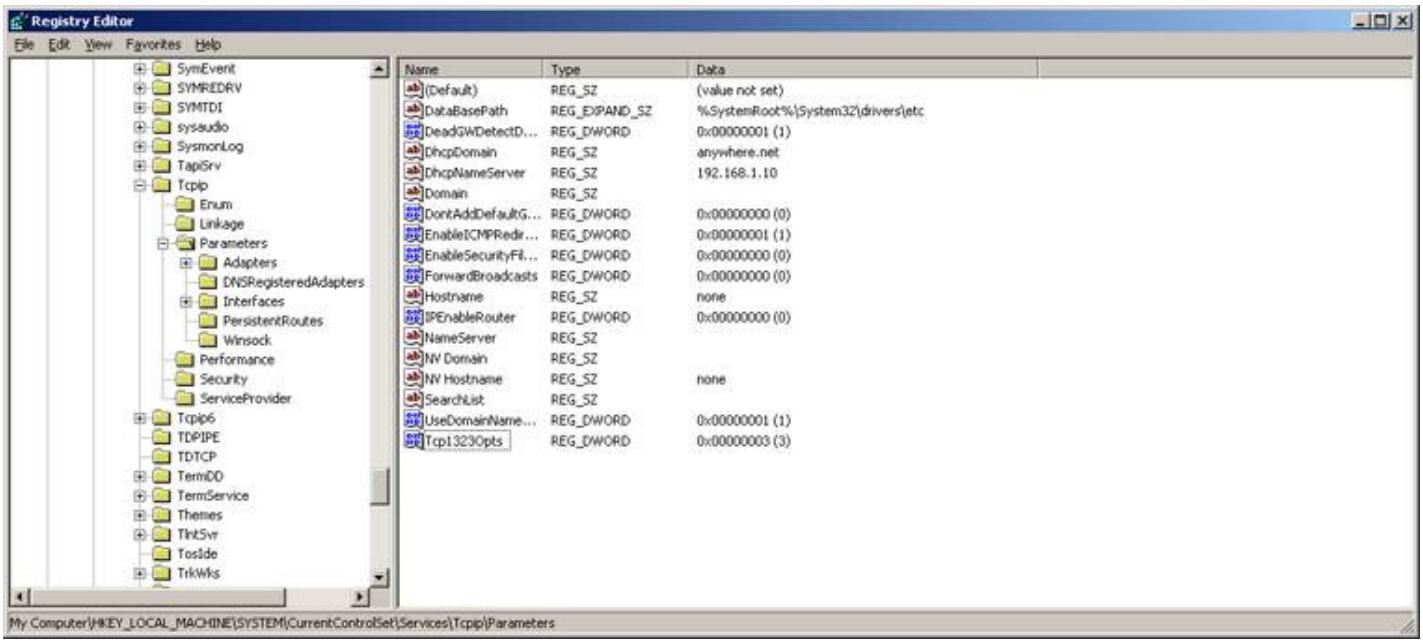


5). Double-click your new value to open a dialog box and enter a valid value range and select "OK". For "Tcp1323Opts" you want to enter "3", which turns on all features (see the list below if you are interested in the details).



You should now see your new REG_DWORD value (In this case option "TCP1323" is REG_DWORD with value 3.

(Note that values are displayed in hexadecimal and decimal).



6) Repeat steps 3, 4 and 5 to add DWORD registry parameters "GlobalMaxTcpWindowSize" and "TcpWindowSize",

both of which are set to the computed DBP of the path. (e.g. 400000 might be a good starting value).

7) Restart your system for the changes to take effect.

Detailed descriptions of the registry parameters

Tcp1323Opts
Key: Tcpip\Parameters

Value Type: REG_DWORD—number (flags)

Valid Range: 0, 1, 2, 3

0 (disable RFC 1323 options)

1 (window scaling enabled only)

2 (timestamps enabled only)

3 (both options enabled)

Default: No value. The default behavior is as follows: do not use the Timestamp and Window Scale options when initiating TCP connections but use them if the TCP peer that is initiating communication includes them in the SYN segment.

Description: This parameter controls the use of RFC 1323 Timestamp and Window Scale TCP options. Explicit settings for timestamps and window scaling are manipulated with flag bits. Bit 0 controls window scaling, and bit 1 controls timestamps.

GlobalMaxTcpWindowSize
Key: Tcpip\Parameters

Value Type: REG_DWORD—Number of bytes

Valid Range: 0–0x3FFFFFFF (1073741823 decimal; however, values greater than 64 KB can only be achieved when connecting to other systems that support RFC 1323 window scaling, which is discussed in the TCP section of this article.)

Default: This parameter does not exist by default.

Description: The *TcpWindowSize* parameter can be used to set the receive window on a per-interface basis. This parameter can be used to set a global limit for the TCP window size on a system-wide basis.

TcpWindowSize
Key: Tcpip\Parameters, Tcpip\Parameters\Interface*interfaceGUID*

Value Type: REG_DWORD—number of bytes

ValidRange: 0–0x3FFFFFFF (1073741823 decimal). In practice the TCP/IP stack will round the number set to the nearest multiple of maximum segment size (MSS). Values greater than 64 KB can be achieved only when connecting to other systems that support RFC 1323 Window Scaling, which is discussed in the "Transmission Control Protocol (TCP)" section of this article.
Default: The smaller of the following values:

- 0xFFFF
- *GlobalMaxTcpWindowSize* (another registry parameter)
- The larger of four times the MSS
- 16384 rounded up to an even multiple of the MSS

The stack also tunes itself based on the media speed:

- Below 1 Mbps: 8 KB
- 1 Mbps – 100 Mbps: 17 KB
- Greater than 100 Mbps: 64 KB

The default can start at 17520 for Ethernet, but may shrink slightly when the connection is established to another computer that supports extended TCP header options, such as Selective Acknowledgements (SACK) and TCP Timestamps, because these options increase the size of the TCP header beyond the usual 20 bytes, leaving slightly less room for data.
Description: This parameter determines the maximum TCP receive window size offered. The receive window specifies the number of bytes that a sender can transmit without receiving an acknowledgment. In general, larger receive windows improve performance over high-delay, high-bandwidth networks. For greatest efficiency, the receive window should be an even multiple of the TCP Maximum Segment Size (MSS). This parameter is both a per-interface parameter and a global parameter, depending upon where the registry key is located. If there is a value for a specific interface, that value overrides the system-wide value. See also *GobalMaxTcpWindowSize*.

EnablePMTUDiscovery
Key: Tcpip\Parameters
Value Type: REG_DWORD—Boolean
ValidRange: 0, 1 (false, true)
Default: 1 (true)
Description: When this parameter is set to 1 (true) TCP attempts to discover the Maximum Transmission Unit (MTU), or largest packet size, over the path to a remote host. By discovering the Path MTU (PMTU) and limiting TCP segments to this size, TCP can eliminate fragmentation at routers along the path that connect networks with different MTUs. Fragmentation adversely affects TCP throughput and network congestion. Setting this parameter to 0 (not recommended) causes an MTU of 576 bytes to be used for all connections that are not to destinations on a locally attached subnet.

MTU
Key: Tcpip\Parameters\Interfaces*interfaceGUID*
Value Type: REG_DWORD—number
ValidRange: 88–*the MTU of the underlying network*
Default: 0xFFFFFFFF
Description: This parameter overrides the default Maximum Transmission Unit (MTU) for a network interface. The MTU is the maximum IP packet size, in bytes, that can be transmitted over the underlying network. For values larger than the default for the underlying network, the network default MTU is used. For values smaller than 88, the MTU of 88 is used.

Note:
Windows Server 2003 TCP/IP uses PMTU detection by default and queries the network interface card driver to find out what local MTU is supported. Altering the MTU parameter is generally not necessary and may result in reduced performance. See the "Path Maximum Transmission Unit (PMTU) Discovery" section of this article for more details.

AFD Registry Parameters

Note: Under Windows XP SP2, the "DefaultReceiveWindow" value in the AFD branch of the Registry takes precedence over the RWIN values in the TCP branch. Under SP1, it is the other way around, the values in the TCP branch override the AFD value and go out in TCP packet headers.

Afd.sys is the kernel-mode driver that is used to support Windows Sockets applications. When there are three default values, the default is calculated based on the amount of memory detected in the system:

- The first value is the default for smaller computers (less than 19 MB).
- The second value is the default for medium computers (<32 MB on Windows XP Professional, <64 MB on Windows Server 2003).
- The third value is the default for large computers (>32 MB on Windows XP Professional, >64 MB on Windows Server 2003).

For example, if the default is given as 0/2/10, a system containing 12.5 to 20 MB of RAM would default to 2.
The following values can be set under:

HKEY_LOCAL_MACHINE
\SYSTEM
\CurrentControlSet
\Services
\Afd
\Parameters

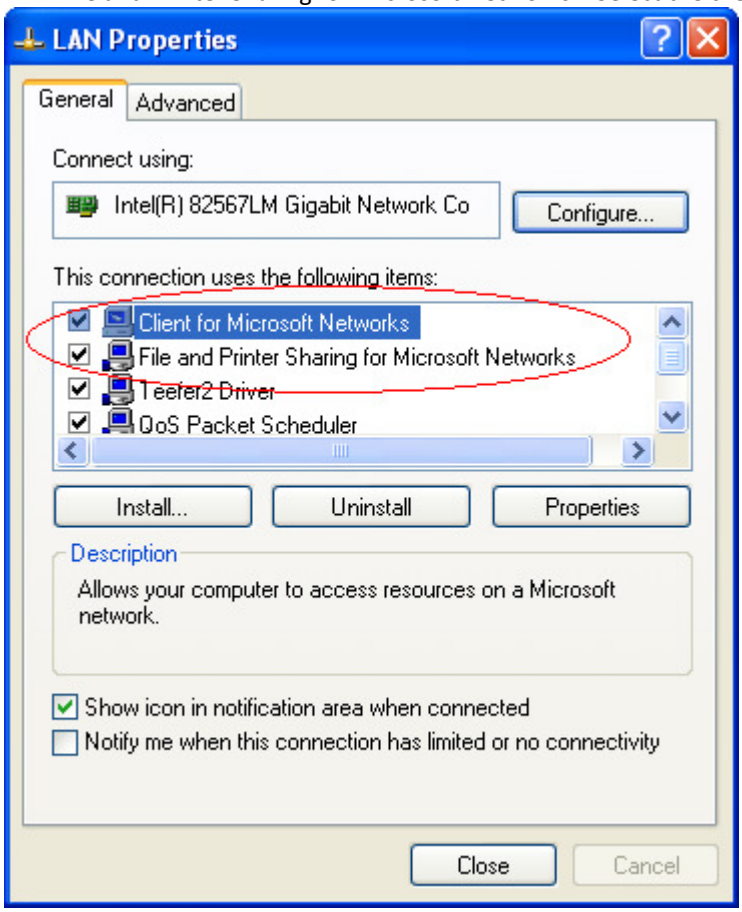
DefaultReceiveWindow
Value Type: REG_DWORD
Default: 4096/8192/8192
Description: The number of receive bytes that AFD buffers on a connection before imposing flow control. For some applications, a larger value here gives slightly better performance at the expense of increased resource utilization. Applications can modify this value on a per-socket basis with the SO_RCVBUF socket option.

DefaultSendWindow
Value Type: REG_DWORD
Default: 4096/8192/8192
Description: This is similar to *DefaultReceiveWindow*, but for the send side of connections.

Lastly and most, important remove or disable Microsoft services from network interfaces.
There are a couple ways to minimize you network footprint. The best way is to remove the client and service altogether, but some of us have local networks for file sharing or print sharing so it's not possible.
These clients and services are the single easiest way worms like code-red or other propagate themselves. The clients and services open UDP and TCP ports that advertise on the network. If the PC is standalone (not part of a domain) and used for gaming and internet these service are overhead and make your PC a target for hackers.

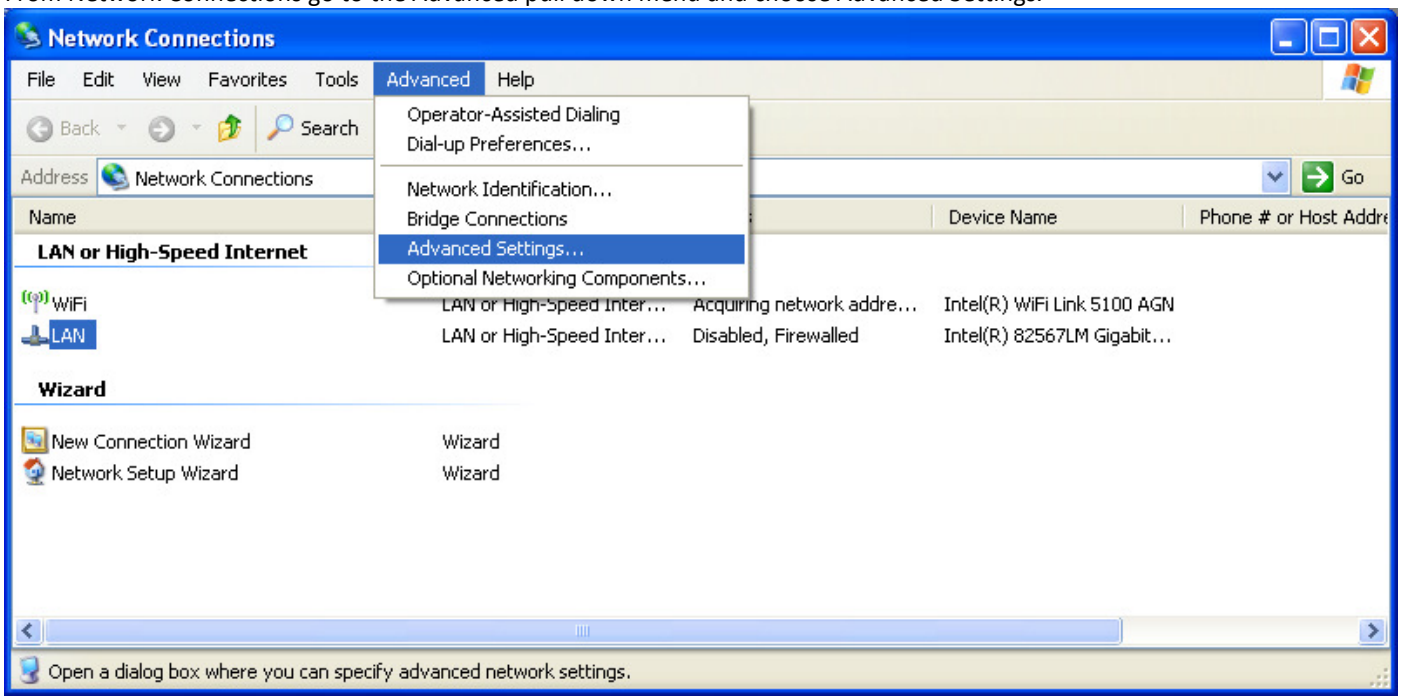
The first way is to remove the Microsoft client and File and Printer Sharing for Microsoft Networks.

From Network Connections go to properties of the network used for internet access. Uninstall both Client for Microsoft Networks AND File and Printer Sharing for Microsoft Networks. Select the client or service and select Uninstall.

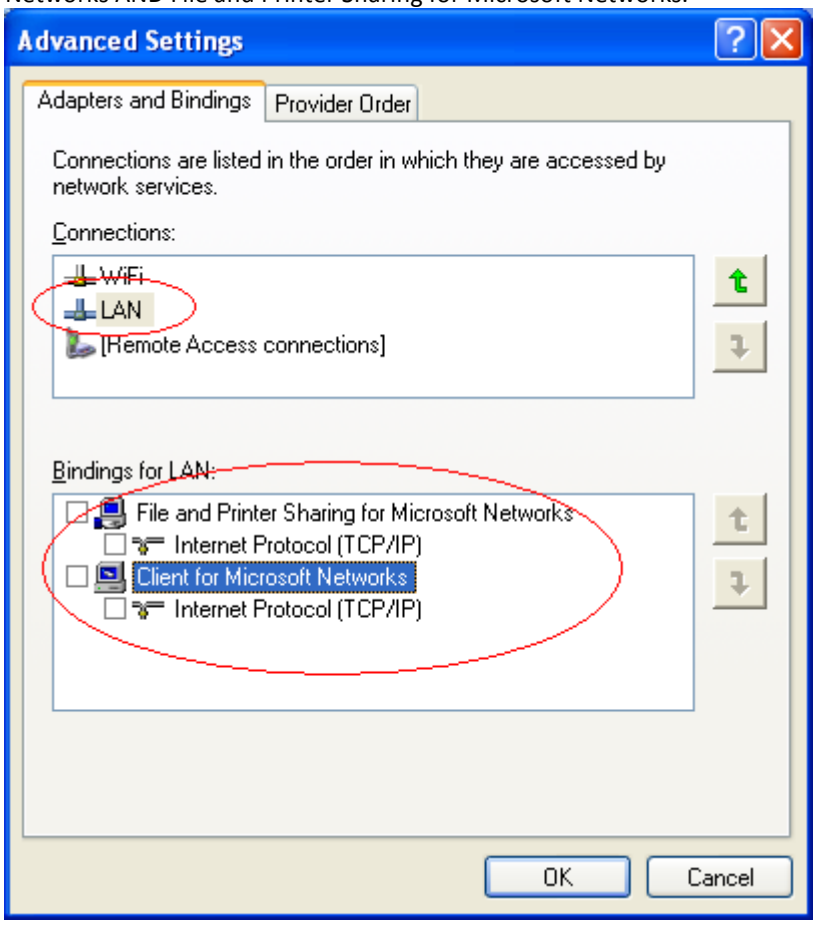


The second way if the services are required for say the wired connection but not the wireless connection is to unbind Microsoft client and File and Printer Sharing for Microsoft Networks.

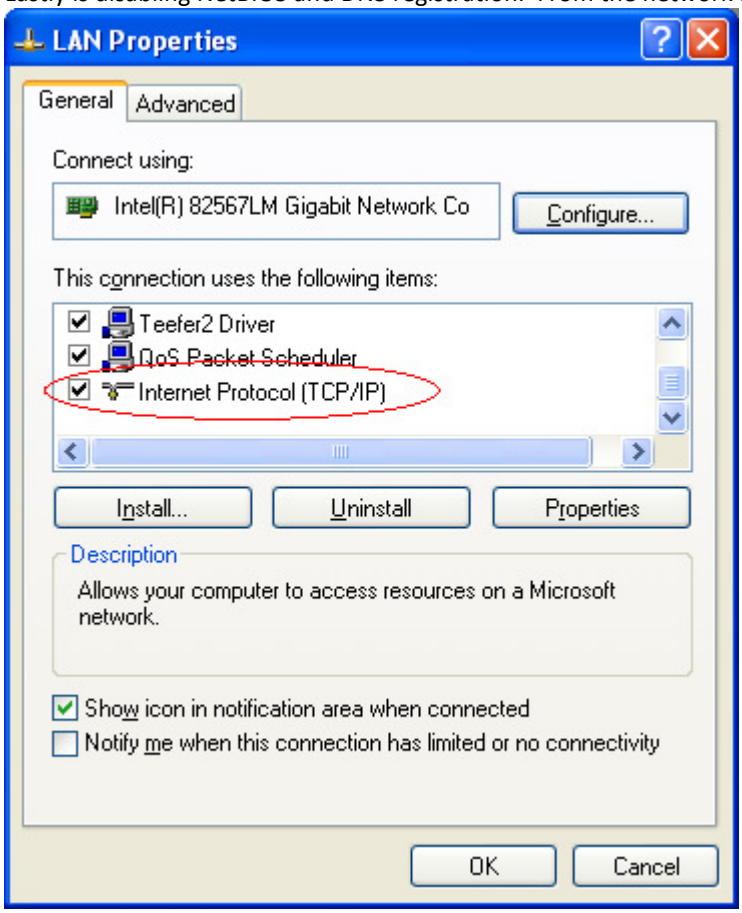
From Network Connections go to the Advanced pull down menu and choose Advanced Settings.



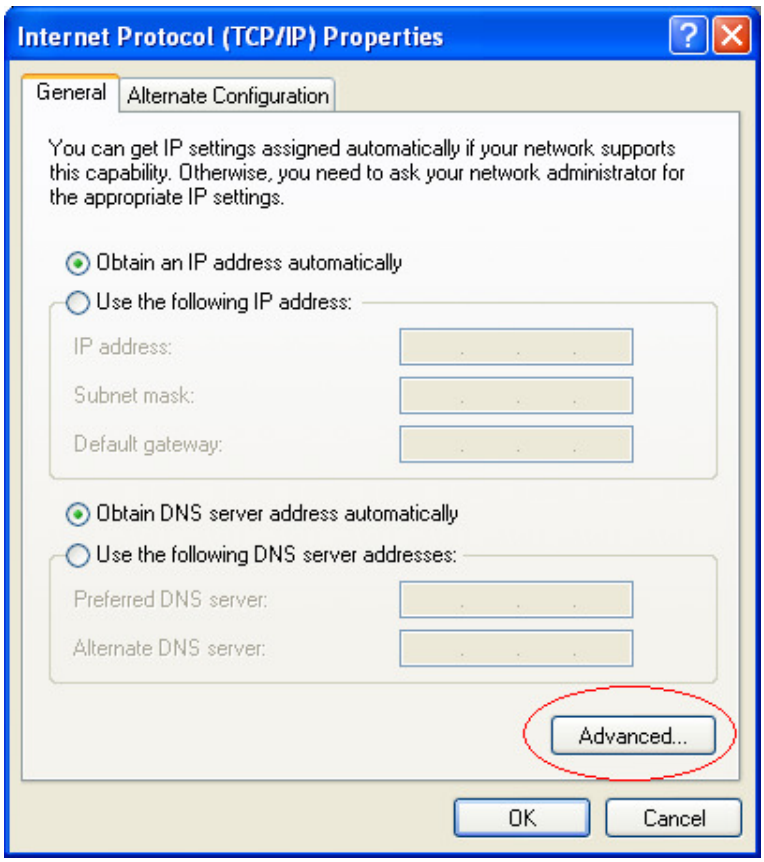
Next select the network adapter in the Connections dialog then below in the Binds for xxx uncheck the both Client for Microsoft Networks AND File and Printer Sharing for Microsoft Networks.



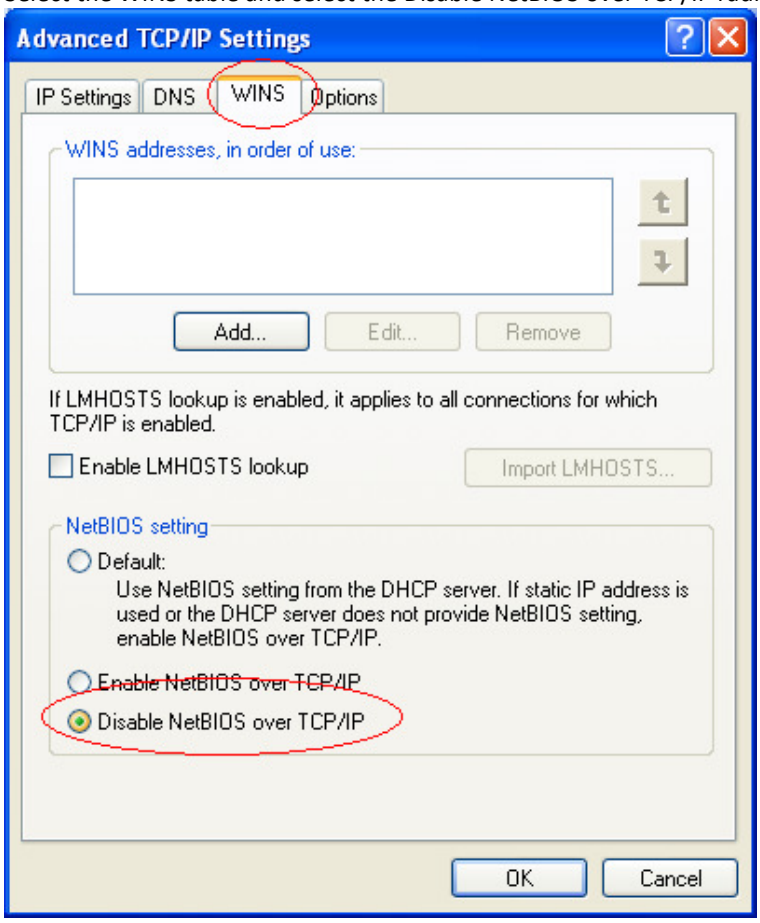
Lastly is disabling NetBIOS and DNS registration. From the network adapter properties double click the TCP/IP protocol



Select Advanced



Select the WINS table and select the Disable NetBIOS over TCP/IP radio button.



Now select the DNS tab and uncheck the Register this connection's addresses in DNS.