Network Card Installation

Ensure that the power cord is removed from the computer before physically connecting the network card to the motherboard.

Windows:

1. Insert the network card into the motherboard and turn the computer back on.
2. Open the device manager and locate the menu option called Network adapters.
3. Inside the Network adapters folder, locate the name of the network card that you just installed.
4. Double click that network card and a properties window will open.
5. Inside of this properties window locate and click on the tab labeled Driver.
6. Inside of the Driver tab, click on the button labeled Update Driver.
7. On the window that pops up, click the icon labeled Search automatically for updated driver software.
8. Finish the installation process for any drivers that the search locates.
9. Once the best drivers for your device are already installed, you may exit the device manager.

Linux:

1. Insert the network card into the motherboard and turn the computer back on.
2. Locate the driver for the network adapter. This will be either on a CD that came with the adapter, or online from the manufacturer.
3. Follow the instructions in the README file that is included with the driver.
4. Once the driver has been installed, you can load the driver into the Linux kernel using #sudo modprobe <driver name>
5. To test that the driver and network adapter are operational use the following commands: #sudo lshw –class network and #sudo ifconfig
6. If your driver is listed after executing these commands and you don’t see anything like “unclaimed” or “down”, then your network card is ready to be used.

Identify Host IP Address

Windows:

1. Connect the Ethernet cable to the network card, and wait about a minute.
2. Open a Windows command prompt.
3. Type the command ipconfig in the command prompt.
4. Locate the name of your adapter—it will be called something like, “Ethernet adapter Ethernet 2”.
5. In the list under the name of your adapter, locate the line that reads “Autoconfiguration IPv4 Address”.

6. The IP address of your network card is located on this line. Example: 169.254.57.129.

Create Local Area Network

This step is only required for connecting Windows to a different operating system. Connecting Windows to Windows does not require this step.

Linux:

1. Follow the instructions in the section titled Identify Host IP Address and write down the IP address of your Windows host.
2. Open the Settings on your Linux system.
3. Once Settings is open, navigate to the tab labeled Network and click on it.
4. Inside of Network, locate the name of your adapter, and press the plus button that is on the same row.
5. Under the identity tab, provide a name for your new LAN in the text box labeled Name.
6. Next, navigate to the tab called IPv4.
7. Under the IPv4 tab, click the circle labeled Manual in the section IPv4 Method.
8. After clicking Manual, a new section titled Addresses will appear.
9. Click the text box titled Address and enter the first three numbers in the Windows IP address that you acquired in step one. For example, if the Windows IP address is 169.254.57.129, then enter 169.254.57. into the text box.
10. Now, for the fourth number that will be entered into the text box titled Address, choose any number that is NOT the same as the fourth number in the Windows IP address. Valid numbers are 0-255, but avoid using low values because these could be reserved. For example, a valid IP address would look like 169.254.57.128 or something similar.
11. Now, click the text box titled Netmask, and enter the value 255.255.0.0 inside that box.
12. Leave the third text box titled Gateway blank, and click the green button labeled Add in the top right corner of the window.
13. This will have created a new connection profile that can be selected for use.

Network Card Configuration

The additional resources section below contains conflicting information about which properties to configure on 10 Gb network cards. The property values listed are my attempt to find a combination of values that will allow for the network to operate as close to 10 Gbps as possible. I'm confident that these values will work for the network that we need, but there may be a more efficient combination of values.

To help determine which parts of your system are the bottleneck, I recommend installing Iperf. Iperf is already installed on all of the computers involved in network testing that I used. The application is controlled from the command line, and more information can be found on the man page. As a starting point, the only two commands that I ever used with

Iperf are: #iperf3 –c 169.254.215.45 -4 --interval 1 --parallel 1 –t 10 and #iperf3.exe –s. Iperf can be found on this website: https://iperf.fr/

Linux:

For Linux, I would recommend following the guides that are posted below. I will include all of the changes that I made here, but how to make them is another story entirely. The changes will be sectioned with headers that correspond to the commands required to make the changes.

ethtool:

1. Change the value of rx-usecs to zero.
2. Change the value of tx-frames-irq to one.
3. RX and TX ring parameters set to their maximum value. The maximum value is given by ethtool.

sysctl:

1. Change the following maximum socket buffer sizes:
   net.core.rmem_max = 268435456
   net.core.wmem_max = 268435456
2. Increase the Linux autotuning TCP buffer limits (min, default, and max number of bytes):
   net.ipv4.tcp_rmem = 4096 87380 134217728
   net.ipv4.tcp_wmem = 4096 65536 134217728
3. Don't cache ssthresh from the previous connection:
   net.ipv4.tcp_no_metrics_save = 1
4. Explicitly set htcp as the congestion control to Hamilton TCP:
   net.ipv4.tcp_congestion_control = htcp
5. If you are using Jumbo Frames, also set this:
   net.ipv4.tcp_mtu_probing = 1
6. Change the maximum number of packets queued on the input side:
   net.core.netdev_max_backlog = 300000
7. net.ipv4.tcp_moderate_rcvbuf =1

Windows:

1. Open the device manager and locate the menu option called Network adapters.
2. Inside the Network adapters folder, locate the name of the network card that you want to configure.
3. Double click that network card and a properties window will open.
4. Inside of this properties window locate and click on the tab labeled Advanced.
5. Inside of the Advanced tab, there is a list of network card properties on the left that can be selected.
6. Once a network card property is selected, its value can be changed in the value field on the right-hand side of the advanced tab menu.
7. Additionally, in the Power Management tab, the option, “Allow the computer to turn off this device to save power” should not be selected.

The following list is an overview of all of the properties that are available for configuration on an **Intel(R) Ethernet Controller X550**.

DMA coalescing: DMA (direct memory access) allows the network device to move packet data directly to the system's memory, reducing CPU utilization. DMA coalescing allows the NIC to collect packets before it initiates a DMA event. This may increase network latency but also increases the chances that the system will consume less energy. (DISABLED | DEFAULT: DISABLED)

Enable PME: PME is Power Management Event. When PME is enabled, you can enable Wake on Settings which explicitly allows wake up with a Magic Packet from shutdown under APM power management mode. (DISABLED | DEFAULT: DISABLED)

Flow Control: Enables adapters to generate or respond to flow control frames, which help regulate network traffic. Disabling this can result in dropped frames, but TCP ensures no data loss. There are mixed opinions on the performance of this property, and it is best to test its effects on a system to system basis. (DISABLED | DEFAULT: Rx & Tx ENABLED)

Header Data Split: Header data split enables the adapter to split the header information from an incoming packet so the CPU does not have to process the whole packet. (DISABLED | DEFAULT: DISABLED)

Interrupt Moderation: Interrupt moderation allows the adapter to moderate interrupts. Disabling this property can increase CPU utilization and decrease packet latency. (DISABLED | DEFAULT: ENABLED)

Interrupt Moderation Rate: Interrupt moderation rate sets the rate at which the controller moderates or delays the generation of interrupts. Moderation helps optimize network throughput and CPU utilization. This change isn't necessary if Interrupt Moderation is Disabled, but I included it because there is a chance that the system as a whole could perform better with Interrupt Moderation Enabled and Interrupt Moderation Rate set to LOW or MEDIUM. (OFF | DEFAULT: ADAPTIVE)

IPsec Offload: The TCP/IP transport can offload the calculation and validation of encrypted checksums for authentication headers (AH) and encapsulating security payloads (ESP) to the network adapter. The TCP/IP transport can also offload the encryption and decryption of ESP payloads to the network adapter. (DISABLED | DEFAULT: DISABLED)

IPv4 Checksum Offload: IPv4 checksum offload enables the adapter to verify the TCP/IP checksum on received packets (RX) and compute checksum on transmitted packets (TX). Enabling this feature can improve TCP/IP performance and reduce CPU utilization. (Rx & Tx ENABLED | DEFAULT: Rx & Tx ENABLED)

Jumbo Packet: Enables jumbo packet capability for TCP/IP packets. When large packets make up most traffic and more latency can be tolerated, jumbo packets can reduce CPU utilization and improve wire efficiency. It seems like enabling this would improve performance, but for some reason performance decreased when I tested it. (DISABLED | DEFAULT: DISABLED)

Large Send Offload V2 (IPv4): Enables the adapter to offload the task of segmenting TCP messages into valid Ethernet frames. Because the adapter hardware can complete data segmentation much faster than operating system software, this feature can improve transmission performance. The adapter also uses fewer CPU resources. (ENABLED | DEFAULT: ENABLED)

Large Send Offload V2 (IPv6): Enables the adapter to offload the task of segmenting TCP messages into valid Ethernet frames. (DISABLED | DEFAULT: ENABLED)

Locally Administered Address: Overrides the initial MAC address with a user-assigned MAC address. To enter a new network address, type a 12-digit hexadecimal number in this box. (NOT PRESENT | DEFAULT: NOT PRESENT)

Log Link State Event: Enables the logging of the following link state changes to the system event log:

- LINK_UP_CHANGE
  Indicates that the adapter has established link.
- LINK_DOWN_CHANGE
  Indicates that the adapter has lost link.
- LINK_DUPLEX_MISMATCH
  Indicates a mismatch in duplex between the adapter and the link partner

(ENABLED | DEFAULT: ENABLED)

Maximum Number of RSS Queues: Receive side scaling queues allocates queue space to buffer transactions between the network adapter and processors. Configuration of queues is not supported on all adapters that support RSS. The number of queues supported varies depending on the adapter and operating system. Only settings allowed for your adapter and operating system are displayed. (MAXIMUM VALUE | DEFAULT: 8 QUEUES)

Packet Priority & VLAN: Enables sending and receiving of IEEE 802.3ac tagged frames. When this feature is enabled, tagged packets use the queue settings defined by the operating system Priority Level Definition. Priority & VLAN is automatically enabled when you set up a VLAN on the VLAN tab. (PACKET PRIORITY & VLAN DISABLED | DEFAULT: PACKET PRIORITY & VLAN ENABLED)

Receive Buffers: Sets the number of buffers used by the driver when copying data to the protocol memory. Increasing this value can enhance the receive performance, but also consumes system memory. (MAXIMUM VALUE (4096) | DEFAULT: 512)

Receive Side Scaling (RSS): Enables processing for a TCP connection across multiple processors or processor cores. If your adapter does not support RSS, or if your operating system does not support it, the RSS setting does not display. Performance can decrease if this is enabled and other computers on the network do not have RSS enabled. (ENABLED | DEFAULT: ENABLED)

Speed & Duplex: It is very important that the network speed and duplex parameters are set to correspond to the settings on the switch to which they are connected. This will mitigate the occurrence of periodic "auto-synchronization" which may temporarily take connections off-line. (10 Gbps FULL DUPLEX | DEFAULT: AUTO NEGOTIATION)

TCP Checksum Offload (IPv4): Enables the adapter to compute (TX) or verify (RX) the TCP checksum of packets. This feature can improve performance and reduce CPU utilization. With Offloading enabled, the adapter computes or verifies the checksum for the operating system. (Rx & Tx ENABLED | DEFAULT: Rx & Tx ENABLED)

TCP Checksum Offload (IPv6): Enables the adapter to compute (TX) or verify (RX) the TCP checksum of packets. (DISABLED | DEFAULT: Rx & Tx ENABLED)

Transmit Buffers: Sets the number of buffers used by the driver when transmitting data. Increasing this value can enhance the transmit performance, but also consumes system memory. (MAXIMUM VALUE (16384) | DEFAULT: 512)

UDP Checksum Offload (IPv4): UDP checksum offload (IPv4) and UDP checksum offload (IPv6) enable the adapter to compute (TX) or verify (RX) the UDP checksum of packets. Enabling this may improve performance. (DISABLED | DEFAULT: Rx & Tx ENABLED)

UDP Checksum Offload (IPv6): UDP checksum offload (IPv4) and UDP checksum offload (IPv6) enable the adapter to compute (TX) or verify (RX) the UDP checksum of packets. Enabling this likely won't improve performance for our system. (DISABLED | DEFAULT: Rx & Tx ENABLED)

Wake on Link Settings: Wakes the computer if the network connection establishes link while the computer is in standby mode. (DISABLED; we don't need it | DEFAULT: DISABLED)

Wake on Magic Packet: Causes the network card to awaken the computer when it receives a magic packet. (DISABLED; we don't need it | DEFAULT: ENABLED)

Wake on Pattern Match: Is a superset of Wake on Magic Packet. It will cause the card to wake the machine when various things come in, including a magic packet, a NetBIOS name query, a TCP SYN packet (either TCPv4 or TCPv6), etc. (DISABLED; we don't need it | DEFAULT: ENABLED)

Additional resources with links:

(link followed by synopsis of the most relevant information)

https://docs.microsoft.com/en-us/biztalk/technical-guides/general-guidelines-for-improving-network-performance

- **Power Option** – Configure the network adapter driver to prevent power management functionality from turning off the network adapter to save power. This functionality may be useful for client computers but should seldom, if ever, be used on a BizTalk Server or SQL Server computer.
- **Fixed Speed/Duplex (do not use AUTO)** - It is very important that the network speed, duplex, and flow control parameters are set to correspond to the settings on the switch to which they are connected. This will mitigate the occurrence of periodic “auto-synchronization” which may temporarily take connections off-line.
- **Max Coalesce Buffers** - Map registers are system resources used to convert physical addresses to virtual addresses for network adapters that support bus mastering. Coalesce buffers are available to the network driver if the driver runs out of map registers. Set this value as high as possible for maximum performance. On servers with limited physical memory, this may have a negative impact as coalesce buffers consume system memory.

On most systems however, the maximum setting can be applied without significantly reducing available memory.

- **Max Transmit/Send Descriptors and Send Buffers** - This setting specifies how many transmit control buffers the driver allocates for use by the network interface. This directly reflects the number of outstanding packets the driver can have in its “send” queue. Set this value as high as possible for maximum performance. On servers with limited physical memory, this may have a negative impact as send buffers consume system memory. On most systems however, the maximum setting can be applied without significantly reducing available memory.
- **Max Receive Buffers** - This setting specifies the amount of memory buffer used by the network interface driver when copying data to the protocol memory. It is normally set by default to a relatively low value. Set this value as high as possible for maximum performance. On servers with limited physical memory, this may have a negative impact as receive buffers consume system memory. On most systems however, the maximum setting can be applied without significantly reducing available memory.
- **All offload options ON** - In almost all cases performance is improved when enabling network interface offload features. Some network adapters provide separate parameters to enable or disable offloading for send and receive traffic. Offloading tasks from the CPU to the network adapter can help lower CPU usage on the server which will improve overall system performance. The Microsoft TCP/IP transport can offload one or more of the following tasks to a network adapter that has the appropriate capabilities:
  - **Checksum tasks** - The TCP/IP transport can offload the calculation and validation of IP and TCP checksums for sends and receives to the network adapter; enable this option if the network adapter driver provides this capability.
  - **IP security tasks** - The TCP/IP transport can offload the calculation and validation of encrypted checksums for authentication headers (AH) and encapsulating security payloads (ESP) to the network adapter. The TCP/IP transport can also offload the encryption and decryption of ESP payloads to the network adapter. Enable these options if the network adapter driver provides this capability.
  - **Segmentation of large TCP packets** - The TCP/IP transport supports large send offload (LSO). With LSO, the TCP/IP transport can offload the segmentation of large TCP packets.
  - **Stack Offload** – The entire network stack can be offloaded to a network adapter that has the appropriate capabilities. Enable this option if the network adapter driver provides this capability.
- **Wake On LAN disabled (unless being used)** – Configure the network adapter driver to disable wake-on lan functionality. This functionality may be useful for client computers but should seldom if ever be used on a BizTalk Server or SQL Server computer.

https://support.microsoft.com/en-us/help/4494445/tcp-receive-auto-tuning-level-feature-in-windows

The TCP receive window auto-tuning settings can be modified in Windows. I did not find any specifics on how this would affect performance, but I think that it could improve performance when a TCP connection is first opened.

https://docs.microsoft.com/en-us/windows-server/networking/technologies/network-subsystem/net-sub-performance-tuning-nics

Performance Tuning for Low Latency Packet Processing

Many network adapters provide options to optimize operating system-induced latency. Latency is the elapsed time between the network driver processing an incoming packet and the network driver sending the packet back. This time is usually measured in microseconds. For comparison, the transmission time for packet transmissions over long distances is usually measured in milliseconds (an order of magnitude larger). This tuning will not reduce the time a packet spends in transit.

Following are some performance tuning suggestions for microsecond-sensitive networks.

- Set the computer BIOS to **High Performance**, with C-states disabled. However, note that this is system and BIOS dependent, and some systems will provide higher performance if the operating system controls power management. You can check and adjust your power management settings from **Settings** or by using the **powercfg** command. For more information, see Powercfg Command-Line Options
- Set the operating system power management profile to **High Performance System**. Note that this will not work properly if the system BIOS has been set to disable operating system control of power management.
- Enable Static Offloads, for example, UDP Checksums, TCP Checksums, and Send Large Offload (LSO).
- Enable RSS if the traffic is multi-streamed, such as high-volume multicast receive.
- Disable the **Interrupt Moderation** setting for network card drivers that require the lowest possible latency. Remember, this can use more CPU time and it represents a tradeoff.
- Handle network adapter interrupts and DPCs on a core processor that shares CPU cache with the core that is being used by the program (user thread) that is handling the packet. CPU affinity tuning can be used to direct a process to certain logical processors in conjunction with RSS configuration to accomplish this. Using the same core for the interrupt, DPC, and user mode thread exhibits worse performance as load increases because the ISR, DPC, and thread contend for the use of the core.

https://docs.microsoft.com/en-us/windows-server/networking/technologies/network-subsystem/net-sub-performance-top

Decent overview, not really helpful for technical decisions.

https://www.intel.com/content/www/us/en/support/articles/000005811/network-and-i-o/ethernet-products.html?productId=88208&localeCode=us_en

How should I optimize my driver settings?

- For Intel® Ethernet 10 Gigabit Converged Network Adapters, you can choose a role-based performance profile to automatically adjust driver configuration settings.
  - See Advanced driver settings for 10/25/40 Gigabit Ethernet Adapters for more information on configuring the individual driver settings listed below.
- Reduce Interrupt Moderation Rate to Low, Minimal, or Off:
  - Also known as Interrupt Throttle Rate (ITR).
  - The default is **Adaptive** for most roles.
  - The low latency profile sets the rate to **off**.
  - The storage profiles set the rate to **medium**.
  - Note: Decreasing Interrupt Moderation Rate increases CPU utilization.
- Enable Jumbo Frames to the largest size supported across the network (4KB, 9KB, or 16KB).
  - The default is Disabled.
  - Note: Enable Jumbo Frames only if devices across the network support them and are configured to use the same frame size.
- Disable Flow Control.
  - The default is Generate & Respond.
  - Note: Disabling Flow Control can result in dropped frames.
- Increase the Transmit Descriptors buffer size.
  - The default is 256. Maximum value is 2048.
  - Note: Increasing Transmit Descriptors increases system memory usage.
- Increase the Receive Descriptors buffer size.
  - The default is 256. Maximum value is 2048.
  - Note: Increasing Receive Descriptors increases system memory usage.

https://www.intel.com/content/www/us/en/support/articles/000005783/network-and-i-o/ethernet-products.html

https://www.intel.com/content/www/us/en/support/articles/000005593/network-and-i-o/ethernet-products.html?productId=88208&localeCode=us_en

Good overview of some NIC properties.

https://blog.packagecloud.io/eng/2017/02/06/monitoring-tuning-linux-networking-stack-sending-data/

https://blog.packagecloud.io/eng/2016/06/22/monitoring-tuning-linux-networking-stack-receiving-data/#tldr

Massive guides that have all of the information that you could ever want on Linux networking.

https://access.redhat.com/sites/default/files/attachments/20150325_network_performance_tuning.pdf

If you find yourself chest deep in the Linux kernel, I recommend starting with this link.

https://www.tradingtechnologies.com/xtrader-help/tt-system-requirements/appendix-setting-descriptions/nic-settings/

https://cromwell-intl.com/open-source/performance-tuning/ethernet.html

https://fasterdata.es.net/host-tuning/linux/test-measurement-host-tuning/

https://darksideclouds.wordpress.com/2016/10/10/tuning-10gb-nics-highway-to-hell/

Miscellaneous guides that provide network configuration information (These guides tend to conflict one another).