

## Ethernet Auto-negotiation

In today's world of computer networks auto-negotiation is an important plug-and-play technology. Auto-negotiation as an algorithm was defined by Section 28 of the IEEE 802.3 standard and first introduced in 1997 as part of the IEEE 802.3u standard on Fast Ethernet. Auto-negotiation was designed to be backward compatible with original Ethernet networking standards as well. Auto-negotiation was further enhanced in 1999 by the IEEE standard 802.3ab with the introduction of Gigabit Ethernet. Auto-negotiation is best defined as the mutual agreement by two network devices sharing a wire on the speed, duplex, and controls to govern the use of that wire. As a protocol auto-negotiation exists strictly at the PHY (physical) layer of the OSI (Open System Interconnection Reference Model) and is implemented by software, hardware, or a mixture of both. Specifically this white paper will detail how the protocol negotiates speed, duplex, Auto-MDIX (cable termination), and flow control.

As you will see in the technical discussions that follow, auto-negotiation is an extremely important setting on today's wired Ethernet networks. For a link to function properly the devices on either side of the wire must be configured in the same manner; either both set to auto-negotiation or both set to the same hard-coded speed and duplex settings. In an environment where one device is set to auto-negotiate and the other device is set to a hard-coded speed and duplex the auto-negotiate algorithm can detect speed and set that appropriately. The duplex setting of the remote device is indeterminable by the auto-negotiating device. Following the IEEE standard, the auto-negotiating device falls back to half-duplex. This presents an issue if the remote device is set to full-duplex. Typically in such a scenario, users complain of slow network connectivity and application timeouts. These symptoms will be explained in detail in the section discussing Duplex.

Lastly, it should be noted that, according to the IEEE specification the use of gigabit Ethernet requires the use of auto-negotiation therefore 1000Mb/s is not a valid hard-coded option in a true IEEE compliant networking device.

### Speed

IEEE 802.3u introduced 100Mb/s to what was previously only a 10 Mb/s Ethernet networking world. Now that computers had a choice of what speed to communicate a procedure needed to be introduced to govern this decision. With the introduction of a third speed, 1000 Mb/s or Gigabit Ethernet, this procedure became even more important. Thus the auto-negotiation protocol was created and the NWay algorithm adapted to provide a plug and play solution to this decision making process while still maintaining complete backwards compatibility with the 10 Mb/s protocol.

The 10 Mb/s standard detects an active link with another network device through the transmission and reception of Link Integrity Test (LIT) pulses whenever the device is not actively sending or receiving data. These LIT pulses or Normal Link Pulses (NLP), as the name was later changed to, consist of a single uni-polar positive-only pulse for the duration of 100ns at an interval of 16ms with a +/-8ms window.

The auto-negotiation protocol introduced with the 100 Mb/s standard transmits a Fast Link Pulse (FLP) instead of an NLP. A single FLP burst consists of a series of 33 pulses. Each burst of 33 pulses is 2ms long in total and fall into the same transmission interval of 16ms +/- 8ms. The individual pulses are 125  $\mu$ s with 62.5 $\mu$ s +/- 7 $\mu$ s between pulses. Diagram 1 illustrates this

timing. The individual pulses alternate between clock pulses and data pulses with the first and all successive odd numbered pulses being clock pulses. Each of the 16 data pulses (with each pulse or lack of pulse representing a 1 or a 0, respectively) consist of a single bit of data and collectively add up to 16 bits of 2 bytes of data. These 2 bytes make up the link code word (LCW) which contains the information needed for auto-negotiation.

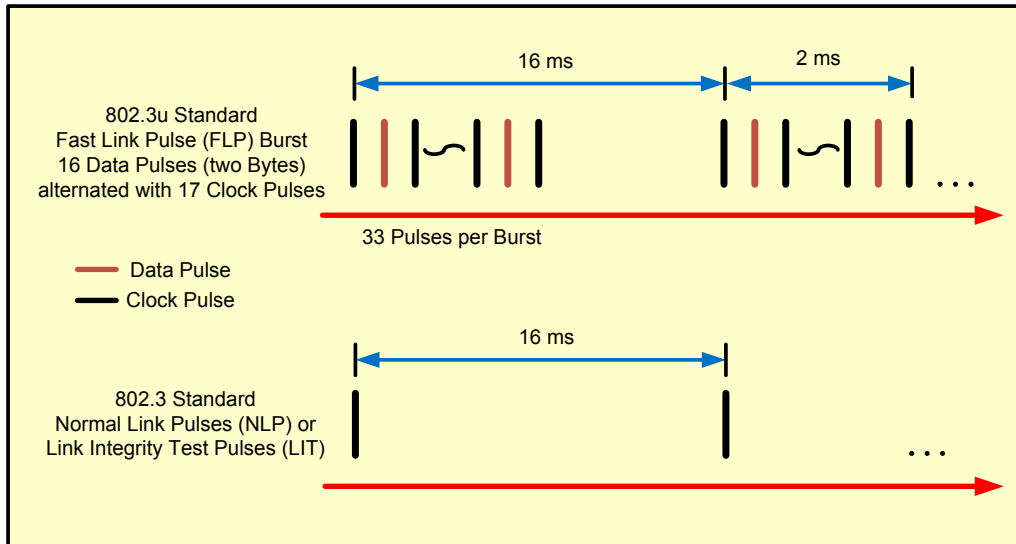


Diagram 1. NLP and FLP Timing.

There are multiple LCW formats but the most important LCW is the base page. This base page is the transmission stating the capabilities of that device. The first 5 bits only have two valid values. They state either to use IEEE 802.3 (Ethernet) or IEEE 802.9 (IsoEthernet over Cat3 twisted pair). The next 5 bits state what speed and duplex combinations that a device can communicate. Bits A5 and A6 are used for Flow Control and D14 is used to acknowledge a negotiation. The last bit, D15 is used to denote the need to use Next Page, a more advanced LCW used to negotiate Gigabit speeds and controls. Diagram 2 illustrates the Base Page.

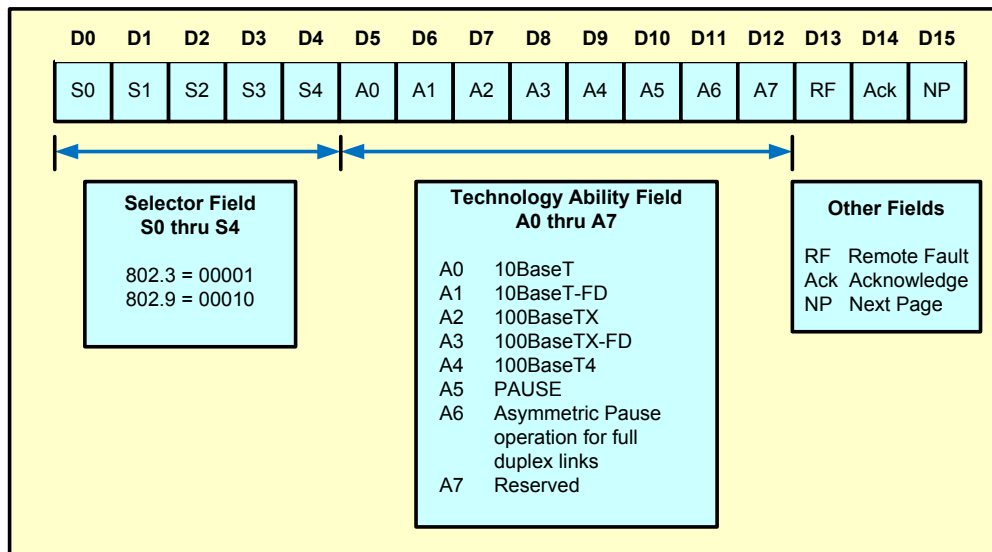


Diagram 2. LCW Base Page.

In order for two devices to agree on a speed for transmission over the wire six identical LCW's must be transmitted and received over the wire, three from each side. Once a device has received three identical LCW's contained within an FLP from the remote end the local device will transmit an FLP with an Acknowledgement bit (ACK) set. It should be noted that each device is only stating what its own capabilities are therefore the two devices need to use the same order of priority to agree upon a speed. This order of priority is part of the IEEE standard and can be found in diagram 3. Once both sides have received a response ACK the speed is agreed upon.

Priority		Technology Type
<b>1</b>	<b>Highest</b>	<b>1000BaseT – Full Duplex</b>
<b>2</b>		<b>1000BaseT – Half Duplex</b>
<b>3</b>		<b>100BaseT2 – Full Duplex</b>
<b>4</b>		<b>100BaseTX– Full Duplex</b>
<b>5</b>		<b>100BaseT2 – Half Duplex</b>
<b>6</b>		<b>100BaseT4</b>
<b>7</b>		<b>100BaseTX – Half Duplex</b>
<b>8</b>		<b>10BaseT – Full Duplex</b>
<b>9</b>	<b>Lowest</b>	<b>10BaseT – Half Duplex</b>

Diagram 3. Priority Resolution table

FLP's were designed to line up with the NLP so that a 10Mb/s device will detect signal on the line at the usual interval and be able to communicate. An auto-negotiation capable device will detect the existence of NLP's and, due to the backward compatibility standards, be able to fall back and communicate via NLP's to work at 10Mb/s.

## Duplex

With the introduction of the IEEE 802.3u Fast Ethernet standard the possibility of simultaneous bi-directional communication became available. Again a protocol and method needed to be introduced to govern this decision. As discussed in the previous section, duplex negotiations are handled in the Base Code Word for 100Mb/s networks and are a part of the Next Page and Message Page LCW's in a 1000Mb/s network.

Duplex mismatch is the most common cause for network link problems outside of physical cabling or hardware failure. Duplex mismatches are caused by the inability of an auto-negotiation device to predict the settings of a hard-coded device. This is because the transmission of FLP's is disabled when a device is hard-coded in accordance with the IEEE specification. Also in accordance with the IEEE specification the auto-negotiation device will connect with the Half-Duplex setting when the duplex setting of the other device cannot be determined.

Duplex mismatches can be difficult to identify because they do not cause a complete link drop. Often the link will perform sufficiently to prevent any alarms initially. This is especially the case when the link is used sparsely. Problems will crop up once link activity increases however. The specific problem is that a half-duplex device believes that only one device can talk at a time so it will not talk as long as the other (full-duplex) device is talking. The full-duplex device is not under these restrictions and believes that both devices can transmit simultaneously. If, while the half-duplex device is transmitting, it detects a transmission from the other device it will immediately stop transmitting, throw away all incoming transmissions as invalid, and initiate a stand-off timer to quiesce the medium. In the meantime the full-duplex device completes transmission and assumes reception. The full-duplex device will also receive the truncated packet from the half-duplex device, determine that it is bad, and flag the Cyclical Redundancy Check (CRC) error counter. The half-duplex device will attempt a retransmission of its packet once the stand-off timer is done but the full-duplex device feels no need to retransmit (remember it doesn't know that the other device threw away its packet) and so the half-duplex device will never receive that packet unless a higher level of the OSI layer requires an acknowledgement and causes a retransmission.

The symptoms of this situation will most often show up as a sluggish network link or an application or applications with excessive timeouts. On a correctly configured connection CRC errors should be negligible so an excessive CRC count is often considered symptomatic of a duplex mismatch.

Duplex mismatches can be a particularly difficult problem with unmanaged switches. By definition, an unmanaged switch does not have the ability to hard-code a port to a particular speed or duplex setting and is always in auto-negotiate mode. If a device is hard-coded to a particular speed or duplex the unmanaged switch will not be able to create a fully functioning link with this device and problems will eventually occur. Please refer to diagram 4 for an illustrated example.

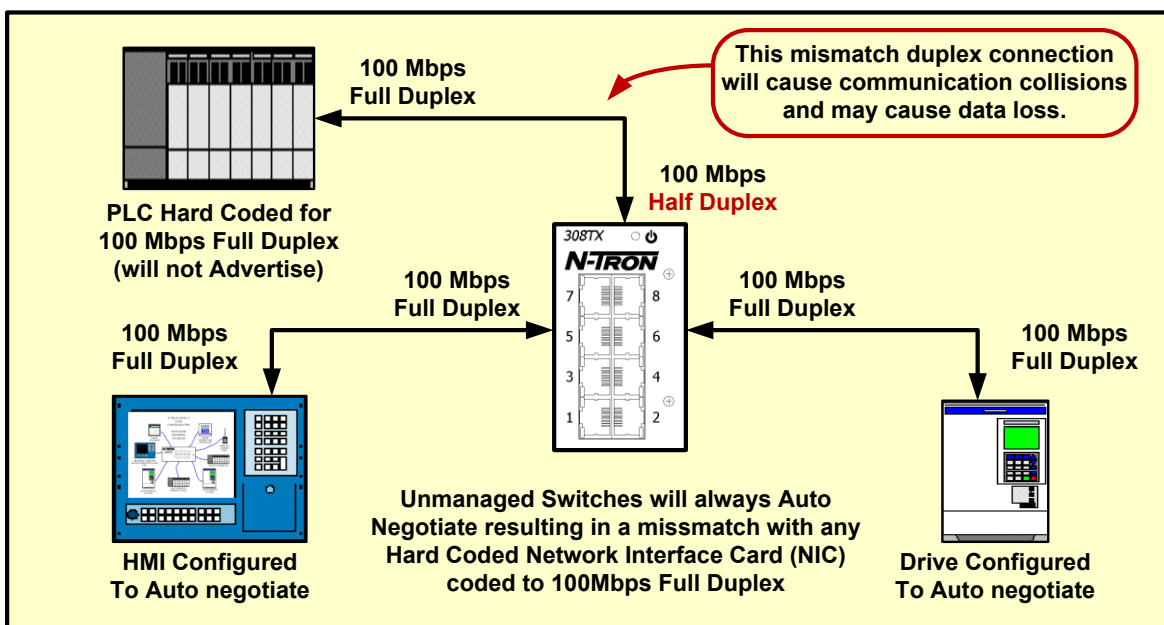


Diagram 4. Duplex mismatch scenario

## Auto-MDIX (Media Dependent Interface with Crossover support)

The introduction of twisted pair cabling also allowed for the possibility of multiple ways to plug the cable in. A category-5e twisted pair cable contains 8 distinct copper wires but there are two industry standard ways to wire the RJ-45 connectors. The goal of these standards is to provide the same wire on each of the pins 1 to 8 on either side of the cable. In order to wire a crossover cable the cable maker installs one end upside down thereby presenting pins 1 to 8 on one side and pins 8 to 1 on the other side.

In order for one device to connect to another device the transmit (TX) on one must be connected to the receive (RX) on the other device and vice versa. This system is required in order for the two devices to communicate. Since most cables are usually wired to be a straight-through cable it was decided to fix this problem at the device level. Traditionally, network devices and computer network cards are wired opposite of each other. Media Device Interface (MDI) is the orientation that a computer card is traditionally cabled to and Media Device Interface-Crossover (MDIX) is the orientation used for a switch or other network device. This was sufficient in the old days but required the use of a special cross-over cable for computer-to-computer and switch-to-switch communications.

Auto-MDIX is a procedure developed and patented by two engineers at HP and is included in the Gigabit Ethernet standard from the IEEE, IEEE 802.3ab. The Auto-MDIX protocol removes the need of a specific cross-over or straight-through cable by attaching both the receiver and the transmitter to both wires in the pair. Per the Gigabit Ethernet standard the receiver knows what the transmitter is sending. It electrically subtracts that signal from what it senses on the wire and using echo cancellation the receiver is able to compute what is being transmitted from the remote end.

## Flow Control

With the ever increasing speed at which devices can transmit data it is important that backplanes, buffers, and switch to switch ports keep up with this speed escalation. If a switch's backplane speed is greater than the sum of the cumulative speed of all of that switch's ports we often call the switch a wire speed switch. This is often unattainable with higher density switches. A link will become saturated when the connection between two devices has more data to transmit than bandwidth to transfer that data. This is an easy scenario to create on a link between two switches if the uplink port is the same speed as the user ports. This introduces the need for flow control, a process that allows one device to ask the other to pause in order to let it catch up. This pause procedure could include a timer until a restart, require a unpauses notification, or simply be a stall tactic with dummy data to delay communications.

In 10 Mb/s networks devices that require a pause in the network simply fill the media with a dummy packet after each packet is received to prevent data from coming through. This technique is called backpressure. Backpressure is also the pause process in 100 Mb/s half-duplex networks. In 100 Mb/s full-duplex and 1000 Mb/s full-duplex networks the auto-negotiation protocol's pause control is implemented. A device that needs a pause sends an FLP with the appropriate pause bit set (either A5 or A6).

## Summary

In conclusion, the auto-negotiation standard allows for a plug and play environment to exist in a networking world with multiple speeds, duplexes, cabling standards, and flow controls. N-TRON recommends leaving all devices in a network set to auto-negotiate to allow for ease of deployment and to minimize the possibility of introducing configurations into your network now or at a later date. If hard-coding is necessary then N-TRON recommends hard-coding both sides and documenting these hard-coded settings to ensure against future problems if network changes are made.

Globally recognized as a market leader in the Industrial Ethernet marketplace, N-TRON's products are used throughout the world in a wide variety of applications including maritime, process control, wind farms, wastewater treatment plants, nuclear power plants, solar energy, and security and surveillance where reliability is an absolute necessity.

N-TRON is headquartered in Mobile, Alabama, with operations located throughout the United States, Canada, EMEA, India and the Pacific Rim. N-TRON products are distributed in over 75 countries worldwide.