AN4839

LAN780x Connectivity and Throughput Issue Triage

Author: *Andrew Rogers*
Microchip Technology, Inc.

1.0 INTRODUCTION

This document helps end-system integrators follow a methodical debug flow to most efficiently uncover the source of problems and help collect the necessary information required to find suitable solutions. The information contained within the document can be used to determine root causes and resolve issues. In the event that a root cause or solution cannot be identified, it serves as a guide for preparing information needed when submitting a Microchip support case request.

2.0 SECTIONS

This application note covers the following topics:

- Section 4.0, "Problem Statement Questionnaire"
- Section 5.0, "Data Logging"
- Section 6.0, "Debug Flow for Problem Isolation and Resolution"

This document also includes the following supplemental information:

- Section Appendix A:, "How to Verify a USB2 Connection"
- Section Appendix B:, "How to Verify a USB3 Connection"
- Section Appendix C:, "How to Verify an Ethernet Connection"
- Section Appendix D:, "How to Perform Ethernet Performance Measurements and Interpret Results"
- Section Appendix E:, "How to Dump LAN780x EEPROM/OTP Configuration Data and Verify Output File"
- Section Appendix F:, "How to Collect a USB Protocol Analyzer Trace with LAN780x".
- Section Appendix G:, "How to Collect an Ethernet Wireshark Trace with LAN780x"

3.0 REFERENCES

Refer to the following documents when using this application note:

- *LAN7800 Data Sheet*
- *LAN7801 Data Sheet*
- *LAN7800 Schematic Checklist*
- *LAN7801 Schematic Checklist*
- *LAN7800 Evaluation Board*
- *LAN7801 Evaluation Board*
- *LAN7800/LAN7800i Software User's Guide*
- *AN1120 – Ethernet Theory of Operation*
- *AN2686 – Ethernet Compliance Testing for 10BASE-T, 100BASE-TX, and 1000BASE-T*
- *AN8.13 – Suggested Magnetics*
- *AN18.15 PCB Design Guidelines for QFN and DQFN Packages*
- *AN4754 – Using Microchip Bridge Controllers with External Ethernet PHYs*
- Microchip University – *Introduction to USB 2.0: https://mu.microchip.com/introduction-to-usb-20*
- Microchip University – *USB 3 Fundamentals: https://mu.microchip.com/usb-3-fundamentals*
- Microchip University – *Ethernet Fundamentals: https://mu.microchip.com/ethernet-fundamentals*

4.0 PROBLEM STATEMENT QUESTIONNAIRE

It is critical to clearly understand and describe the behavior, characteristics, and patterns of an encountered problem. To ensure that the problem is clearly understood and that support engineers have the necessary information to provide fast assistance, clearly answer the questions in Table 1.

Note: When submitting a support case through Microchip's support portal, include the questions and answers indicated in Table 1 in the support case details. Clearly framing the issue upfront is the best way to speed up problem resolution.

TABLE 1: PROBLEM STATEMENT QUESTIONNAIRE

Question	Answer		
1. Describe your problem and how it is observed.			
2. When does this issue present itself?			
How frequent?			
3. Does the issue happen on all of the systems or just an isolated few?			
4. A. Do you observe any failure patterns (e.g: "Failure always happens when I do X.")?			
B. If so, what particular pattern is observed?			
5. Does this result in a hard failure, or is the system able to recover and resume operation without user action?			
6. If user action is required to recover the system after the issue happens, which of the following can recover it?		a) Power cycle	
		b) Plugging/unplugging the Ethernet	
		c) Plugging/unplugging the USB	
		d) Link-down/ link-up command	
		e) Others (Please specify.)	
7. What are the details of the host system(s) that this issue is observed on?		a) Operating system version	
		b) Processor	
		c) Board support package (if used)	
		d) Other relevant information	

TABLE 1: PROBLEM STATEMENT QUESTIONNAIRE (CONTINUED)

Question	Answer
8. What are the details of the Bridge Controller configuration (i.e. EEPROM/OTP settings, driver configuration settings)?	
9. Does the issue occur on other PCs? If yes, how many different PCs have been tried and how many show the same behavior?	
10. Does the issue occur with Microchip's EVB of the same or similar product?	
11. If using Linux®: Try another LTS kernel version. Does the failure persist? (Also, record which kernel(s) have you tried.)	

5.0 DATA LOGGING

Obtaining clear data for review is helpful. Even if the issue is not well understood, there are common logs that are always useful to have available.

Note: If submitting a support case through Microchip's support portal, include the logs in Table 2 in the support case attachments.

Collect the information in Table 2 during the debug process. This information will be critical in diagnosing and debugging any issue. If filing a support case through Microchip Support, collect this data ahead of time to speed up the response time. Assume that the Microchip engineer will request all these items during initial correspondence.

TABLE 2: RECOMMENDED DATA LOGS FOR COLLECTION

Item	Notes	How-to
Collect OTP/EEPROM dump of part.	Summarize in writing what settings are intended to be modified/set.	Section Appendix E:, "How to Dump LAN780x EEPROM/OTP Configuration Data and Verify Output File"
The schematic of LAN780x and all schematic pages which the LAN780x signals route to	A searchable `.pdf` file is the most useful format. Image files are not recommended.	N/A
Layout of LAN780x and all portions of the layout which the LAN780x signals route to	Native design program files are most useful. Native Gerber files are also useful but less preferred. Images and `.pdf` files of Gerbers are difficult to review and are not desirable.	N/A
USB Protocol Trace	Save the protocol trace file in native format, not a compressed or simplified form. (i.e.: Many protocol analyzers can save to a `.csv` file or similar which masks or discards critical information.)	Section Appendix F:, "How to Collect a USB Protocol Analyzer Trace with LAN780x".
Ethernet Wireshark Trace (Use SharkTap if you cannot run Wireshark on your system.)	Save the protocol trace file in native format.	Section Appendix G:, "How to Collect an Ethernet Wireshark Trace with LAN780x"
Performance/Throughput Logs	iPerf/iPerf3 is strongly preferred.	Section Appendix D:, "How to Perform Ethernet Performance Measurements and Interpret Results"
Linux Specific: Output of 'lsusb -t -v'	This displays the entire USB tree in detail.	N/A
Linux Specific: Output of 'ifconfig -a'	This displays all network interfaces.	N/A
Linux Specific: Output of 'ethtool [interface]'	This displays more details on the LAN780x device. The name of the interface as assigned by the Linux® host must first be obtained through ifconfig.	N/A
Linux Specific: Output of dmesg | grep LAN780x	This displays kernel messages which relate to LAN780x.	N/A

6.0 DEBUG FLOW FOR PROBLEM ISOLATION AND RESOLUTION

Utilize the flow charts in Figure 1 and Figure 2 to aid in isolating the nature of the issue and provide suggestions for common resolutions to these issues. It is recommended to always start with the hardware issue debug flow chart (in Figure 1) even if you believe your issue is software or performance related. The link partner should also be set for auto negotiation.

FIGURE 1: HARDWARE ISSUE DEBUG FLOW CHART

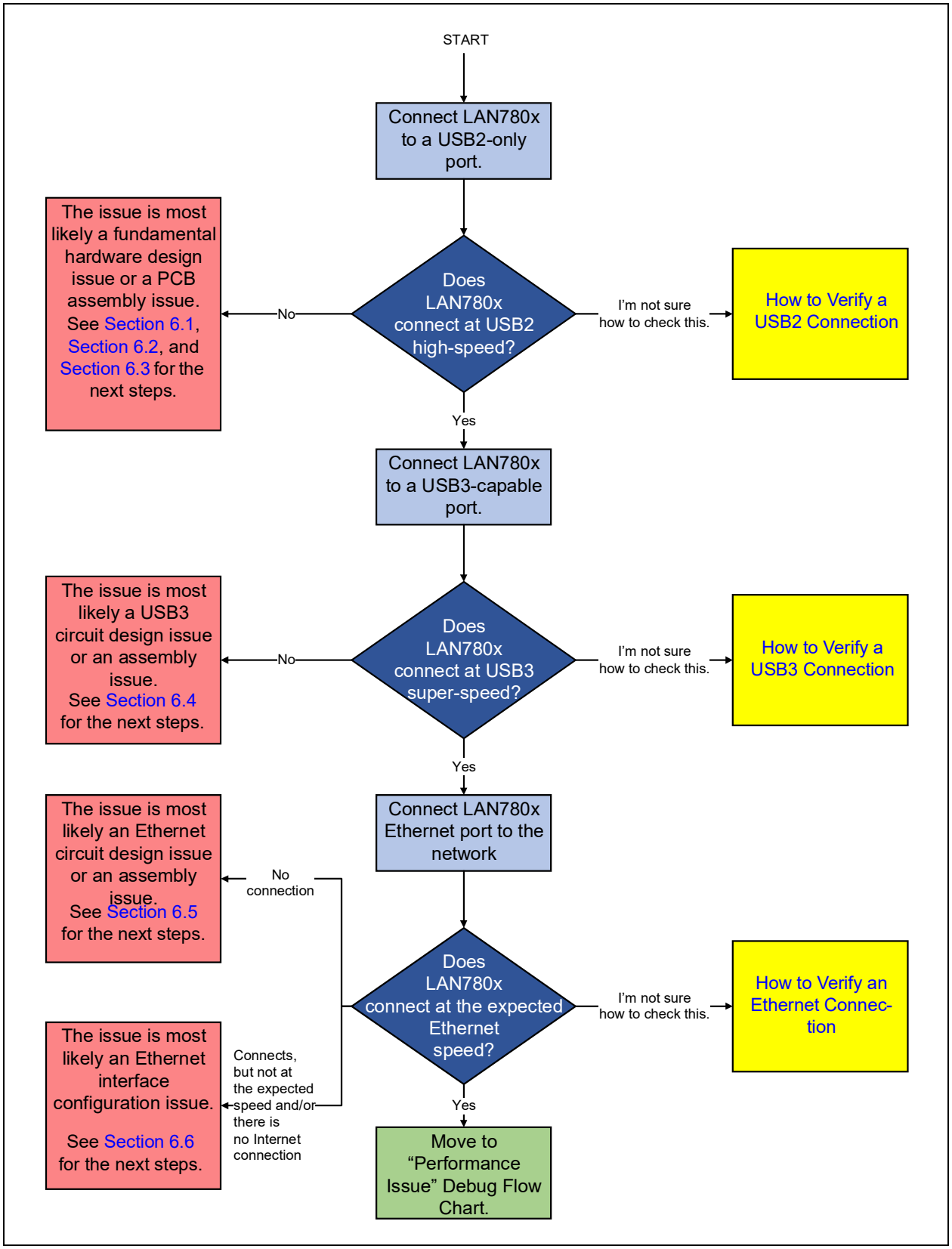
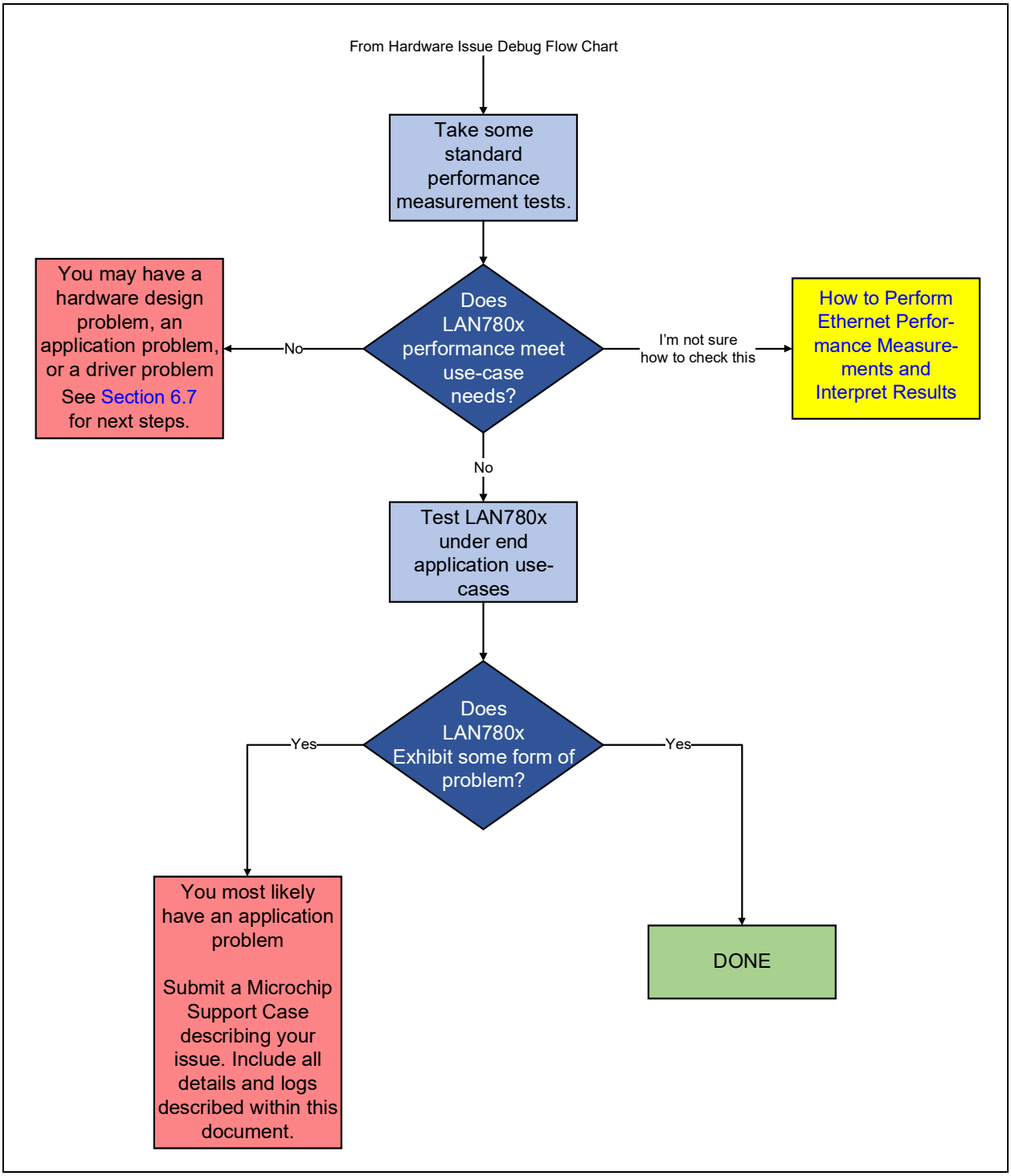


FIGURE 2: PERFORMANCE ISSUE DEBUG FLOW CHART



6.1 Hardware Issue – Fundamental PCB Design

Fundamental PCB Design issues are discovered by systematically checking the following core aspects:

6.1.1 POWER

Probe all PCB power rails on an oscilloscope and take a noise measurement. Ensure ripple and noise are at least less than +/– 100 mV from the nominal voltage. Be sure to probe power while LAN780x is in use and connected (or while you are attempting to connect it to a host) to see if any unexpected noise, droop, or transient behavior may be causing operational issues.

6.1.2 GROUND

A low-impedance ground path to a noise-free ground domain is essential for operation. Ensure the PCB layout design sufficiently connects the LAN780x ePad to a large ground plane. Measure the ground near the LAN780x device to ensure it is noise-free. The ground shift across the PCB is as close to zero as possible.

The LAN780x ePad is also the primary thermal relief pathway. It is essential to fill the ground pad with many vias to a large ground plane to aid in dissipation of thermal energy originating from LAN780x. Refer to *AN18.15 PCB Design Guidelines for QFN and DQFN Packages.*

6.1.3 CLOCKS

The LAN780x uses a clock source to perform USB and Ethernet communication. If the clock is not operating per device requirements (i.e. incorrect frequency, amplitude, or jitter exceeds limits), then it is expected that LAN780x connectivity will be impacted. Probe the clock and verify the following:

- Frequency (including variation)
- Amplitude
- Duty cycle

Note:	The LAN780x also allows for a standard crystal or single-ended clock source. If using a crystal, ensure the loading capacitors meet the crystal's own specifications. If using a single-ended clock source, ensure that the XO pin on LAN7800/LAN7801 is left floating.

6.1.4 SCHEMATIC DESIGN

Microchip offers several resources for ensuring proper schematic design:

1. **LAN7800 Design Documentation:** Visit the LAN7800/LAN7801 product page for an assortment of schematic design, layout design, and placement guidelines.
2. **Evaluation Boards:** The following designs can be used:
   - EVB-LAN7800
   - EVB-LAN7801
3. **Microchip LANCheck Design Review Service:** Visit https://www.microchip.com/en-us/support to submit a schematic and/or layout review service request. Microchip engineers will review your design and provide feedback.

6.2 Hardware Issue – PCB Assembly or IC Damage

Assembly issues can be difficult to identify. The best methods are:

1. **Visual Inspection:** Perform a detailed visual inspection of the entire PCB. Check for missing components, incorrect component placement, and poor solder work. This step takes time but is absolutely essential, especially for new designs.

Note:	For devices which cannot be visually inspected, an X-ray may be required to verify solder connections.

2. **BOM Verification:** Verify that all components placed on the PCB match what is specified in the BOM.
3. **Cleaning:** Perform a thorough cleaning of the entire PCB to remove excess manufacturing residue. Certain solder flux residues can be conductive and create weak shorts which can cause unexpected operation.

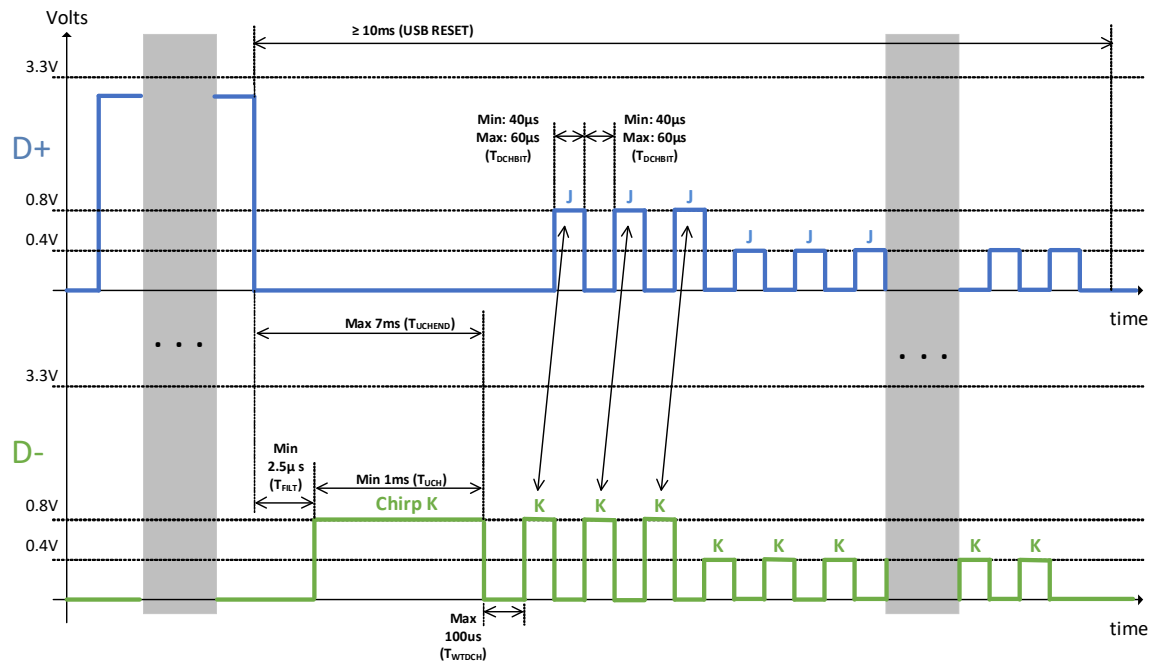
4. **Solder Reflow:** Try to reflow the solder on critical components manually or the entire PCB (if possible) to see if operation is improved or changes. If operation changes after reflow, a PCB soldering issue (i.e.: too much solder, too little solder, or incorrect solder baking profile) may be responsible for the observed issues.
5. **Q&A:** Answer the following questions and think about the answers. If you are experiencing problems where only certain PCBs exhibit certain issues, that is strong evidence of a PCB assembly issue.
   a) Is the observed issue the same on each board produced, or different from board-to-board?
   b) Do some boards work and others do not?
   c) Is the issue localized to an individual build of PCBs (i.e.: previous manufacturing runs had no issues, and no changes in manufacturing have been made).
6. **Failure Analysis:** If the issue is only occurring on one board or a very small number of boards, consider submitting the LAN780x device IC to Microchip for Failure Analysis.

6.3 Hardware Issue – USB2 Connectivity

If both Hardware Issue – Fundamental PCB Design and Hardware Issue – PCB Assembly or IC Damage are ruled out as root causes, focus on the following things:

1. **Try different cables:** Ensure the cable being used is a high-quality USB2 capable cable. If in doubt, purchase new cables from a reputable vendor. Try both long and short cables to see if a connection can be established.
2. **Check VBUS_DET:** Ensure that **VBUS_DET** is around 3.0V-3.3V when the system is connected to a USB host. Typically, **VBUS_DET** is connected to the USB connector's 5V VBUS pin through a resistor divider. For embedded connections, **VBUS_DET** may be supplied from a GPIO on the SoC/CPU or simply tied high to 3.3V permanently. If **VBUS_DET** is not supplied, the LAN780x will never connect via USB2 or USB3.
3. **Check for voltage on D+:** Try the following experiment to see if the USB2 D+ pull-up resistor is being applied to **VBUS_DET** of LAN780x:
   - ***Step 1:*** Apply power to LAN780x.
   - ***Step 2:*** Apply a forced 3.3V to the **VBUS_DET** pin without making a connection to an upstream host. If the host is embedded, hold the host in Reset during this experiment.
   - ***Step 3:*** Probe the USB2_DP signal from the LAN780x. If the LAN780x hardware design is correct, a voltage of approximately +3.0V will be observed. If a proper +3.0V voltage is detected, jump to item #4 below and measure the high-speed chirp sequence. If no voltage is detected on **USB2_DP**, jump to item #5 and proceed through the remaining items to check. If an intermediate voltage is detected, then this is evidence of a hardware design or assembly problem. Jump back to Section 6.1, "Hardware Issue – Fundamental PCB Design" and Section 6.2, "Hardware Issue – PCB Assembly or IC Damage" and review all guidance again.
4. **Look for the high-speed chirp:** Measure the high-speed chirp (only if a ~3.0V is measured on **USB2_DP** of LAN780x in item #2 above). (See Figure 3.) This can be observed on an oscilloscope by doing the following:
   - ***Step 1:*** Connect an oscilloscope to both **USB2_DP** and **USB2_DM** of the LAN780x.
   - ***Step 2:*** Set up a trigger on the oscilloscope. Trigger should be set up on D+. The trigger should be configured as 'falling edge' with a voltage threshold of about 2.0V.
   - ***Step 3:*** Power on the LAN780x.
   - ***Step 4:*** Connect LAN780x to a USB2.0 Host.
   - ***Step 5:*** Capture the high-speed chirp. Make sure to scale the oscilloscope to optimize visibility of the high-speed chirp. Use the following figure as guidance for what to look for and how to properly scale the oscilloscope voltage ranges and time scales. If the High-Speed chirp looks irregular, you can submit a support case to Microchip support for comments on what may be going wrong based on what is observed.

FIGURE 3: HIGH-SPEED CHIRP



5. **Check USB connector schematic pinout:** Double check the pinout of your selected USB connector against the connector's data sheet. Ensure that the **USB2_DP** from LAN780x is routed directly to the connector's **D**+ pin, and **USB2_DM** is routed directly to the connector's **D**– pin. If the polarity of the USB2 interface is reversed, a USB2 connection cannot be made.
6. **Check USB connector PCB layout footprint:** It is a very common mistake to swap D+ and D– signals in the connector PCB footprint—this should be very closely checked.

Note 1: If you performed the experiment described in item #3, step #3, but observed a voltage on **D**– instead of **D**+, then the polarity of USB2 signals is swapped at some point in the design.

2: The polarity of the USB2 signals **D**+ and **D**– can be swapped through configuration parameter PORT_SWAP in the USB_CFG0 register. However, this setting must be applied only after programming an EEPROM of the LAN780x internal OTP configuration memory.

7. **Remove the EEPROM:** If an EEPROM is placed on the design, try physically removing it to see if the connection issue is resolved. The LAN780x does not need EEPROM or OTP configuration in order to connect its USB interface. If the connection issue is resolved, the configuration content of the EEPROM must have some issues. Use the MPLAB Connect Configuration software to generate a new configuration binary.
8. **Collect a USB Protocol Trace:** Connect a USB2 protocol analyzer on the upstream port of the LAN780x and begin collecting the trace before powering on the LAN780x system. Open the analyzer file and look for activity or errors.
9. **Submit a Support Case to Microchip:** If The USB2 connectivity issue is still not resolved or understood after following these debug items, jump back to Section 6.1, "Hardware Issue – Fundamental PCB Design" and Section 6.2, "Hardware Issue – PCB Assembly or IC Damage" and review all guidance again. If the issue is still not resolved or understood, submit a support case to Microchip support with all of your findings clearly documented. Be sure to include the answers to the questionnaire in Section 4.0, "Problem Statement Questionnaire" and all data logs described in Section 5.0, "Data Logging" in the support case submission to accelerate the time to submission. (If you cannot collect any of the requested logs for any reason, state this upfront in the case submission.)

6.4 Hardware Issue – USB3 Connectivity

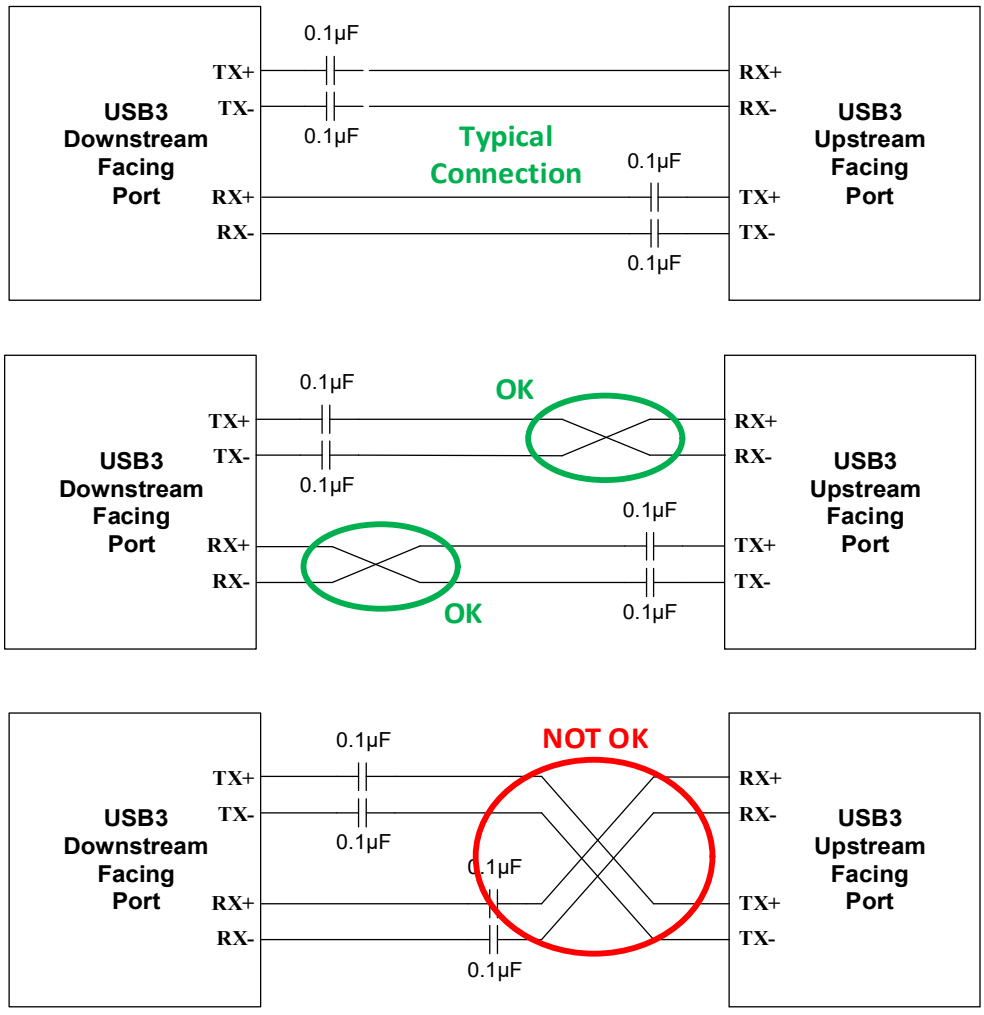
If USB2.0 connects, but USB3.0 does not, perform the following:

1. **Try different cables:** Ensure the cable being used is a high-quality USB3 capable cable. If in doubt, purchase new cables from a reputable vendor. Try both long and short cables to see if a connection can be established.
2. **Check TX caps:** Ensure USB3.0 TX signals have DC blocking capacitors valued at 0.1 µF.
3. **Check RX caps:** If USB3.0 RX traces have DC blocking capacitors (they are optional per USB specification), they should be 0.3 µF. It is not common for designers to place RX capacitors, so do not be concerned if they are not implemented.
4. **Check USB connector schematic pinout:** Double check the pinout of your selected USB connector against the connector's data sheet. The USB3.0 TX signals originating from LAN780x (named USB3_TXDP/USB3_TXDM) should be connected to the pins named **TX**+ and **TX**– or similar on the connector, and the USB3.0 RX signals originating from LAN780x (named USB3_RXDP/USB3_RXDM) should be connected to the pins named **RX**+ and **RX**– or similar on the connector.

Note:	USB3 allows for polarity swapping on the TX differential pair and the RX differential pair. Polarity can be swapped as needed to optimize the PCB layout.

5. **Check USB connector PCB layout footprint:** It is a very common mistake to swap TX+/– and RX+/– differential pairs in the connector PCB footprint. While polarity of the + and – signals within a differential pair is allowed, swapping the entire transmit differential pair with the receive differential pair is not. This should be very closely checked.
6. **Check for TX/RX Crossover in embedded connections to a host:** The TX+/– differential pair of the LAN780x should connect to the RX+/– differential pair pins of an embedded host controller or hub downstream port. Similarly, the RX+/– differential pair of the LAN780x should connect to the TX+/– differential pair pins of an embedded host controller or hub downstream port. The DC blocking caps must be present on both differential pairs as well. See Figure 4.

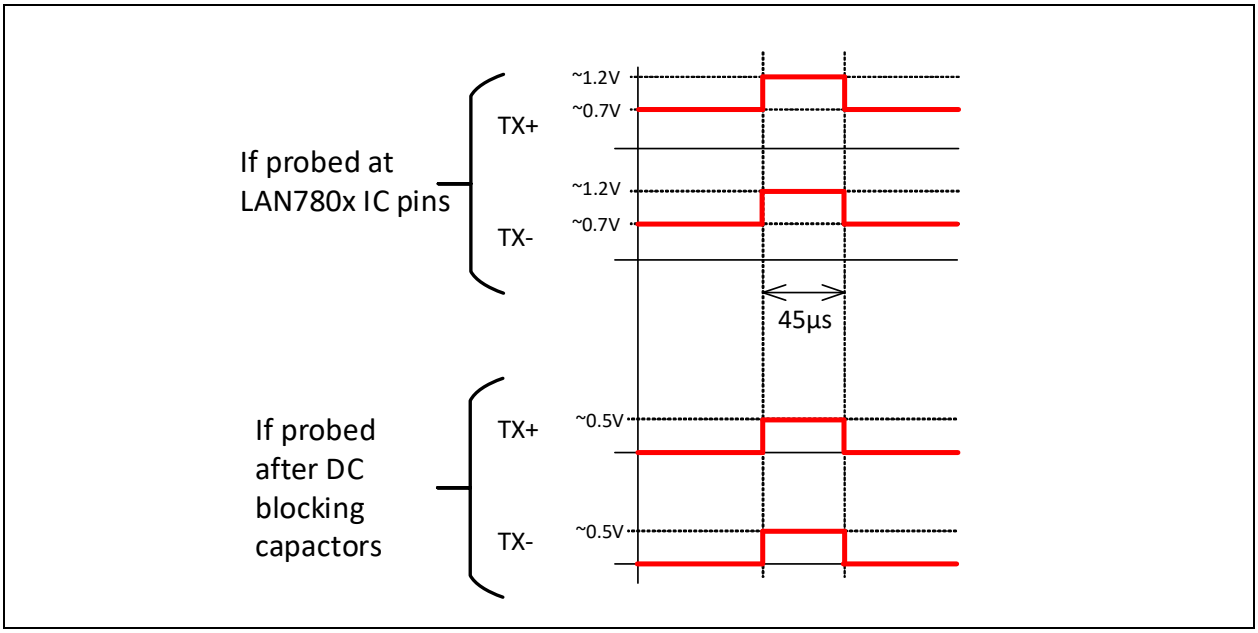
FIGURE 4: TX/RX CONNECTIONS IN AN EMBEDDED CONNECTION



7. **Check the Type-C multiplexer:** If the USB3 connector is a USB Type-C connector, then an external USB3 multiplexing device must be used. Conventions and requirements for multiplexing devices are always specific to the selected multiplexing device. Consult documentation from the supplier of the multiplexer for guidance.
8. **Check for a DC voltage on TX pins of the IC:** The USB3_TXDP/USB3_TXDM signals should have a DC voltage present (if probing on the pins of the IC) of around 0.7V when the USB3 interface is enabled.
9. **Verify RX terminations on pins of IC:** The impedance to ground on USB3_RXDP/USB3_RXDM will measure approximately 50Ω when the USB3 interface is powered on but not connected to a host. To measure, apply power to the LAN780x, force the upstream port VBUS_DET signal to 3.3V, and probe the pin of the IC.
10. **Monitor for RX.Detect:** The USB3_TXDP/USB3_TXDM will continuously emit very short 'RX.Detect' pulses while searching for a connection. The presence of these pulses is good evidence that the USB3 interface is functional. To measure the pulses, apply power to the LAN780x, force the upstream port VBUS_DET signal to 3.3V, and probe the **TXDP** or **TXDM** pin of the IC. Do not connect the LAN780x to a host while probing for RX.Detect.

Note: The USB specifications do not define clear parameters and requirements for the RX.Detect waveform and frequency. That is left to the individual silicon designer. For a description of the RX.Detect waveform properties of the LAN780x (and other Microchip USB3 devices), see Figure 5.

FIGURE 5: RX.DETECT PULSE



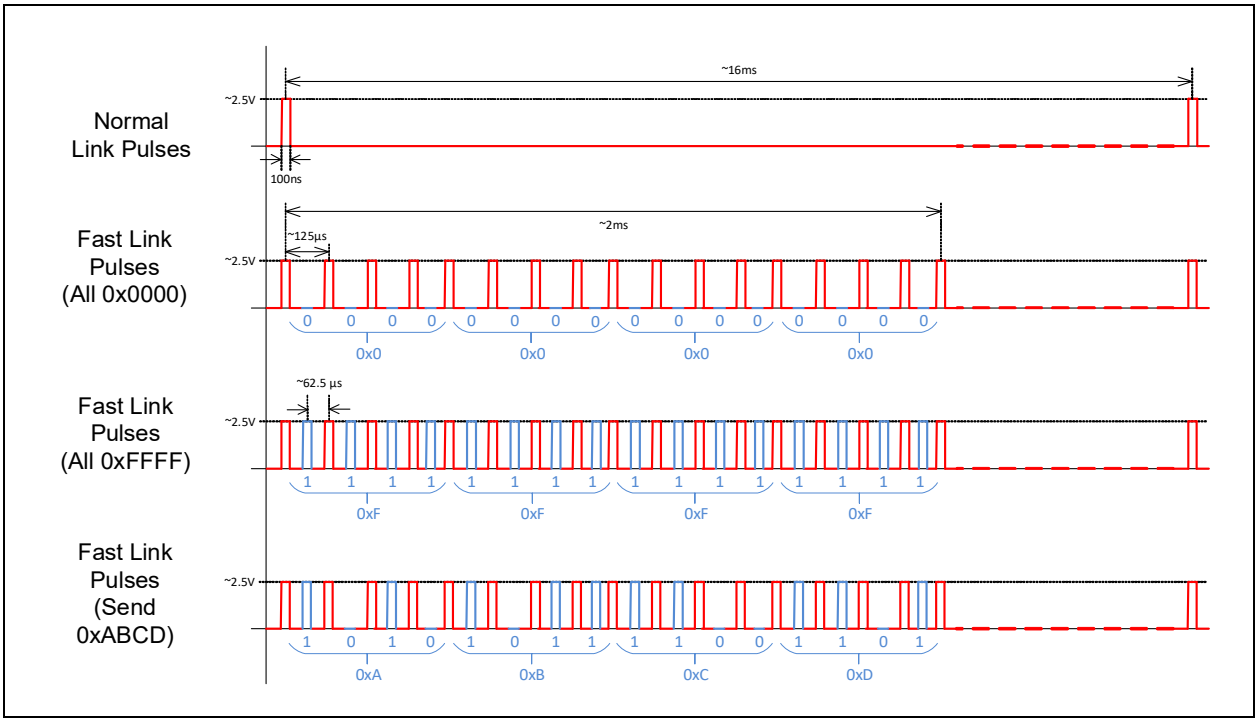
11. **Collect a USB protocol trace:** Connect a USB3 protocol analyzer on the upstream port of the LAN780x and begin collecting the trace before powering on the LAN780x system. Open the analyzer file and look for activity or errors.
12. **Submit a support case to Microchip:** If The USB2 connectivity issue is still not resolved or understood after following these debugging methods, jump back to Section 6.1, "Hardware Issue – Fundamental PCB Design" and Section 6.2, "Hardware Issue – PCB Assembly or IC Damage" and review all guidance. If the issue is still not resolved or understood, submit a support case to Microchip support with all of your findings clearly documented. Be sure to include the answers to the questionnaire in Section 4.0, "Problem Statement Questionnaire" and all data logs described in Section 5.0, "Data Logging" in the support case submission to accelerate the time to submission. (If you cannot collect any of the request logs for any reason, state this upfront in the case submission as well.)

6.5 Hardware Issue – Ethernet Connectivity

1. **Verify Ethernet device on PC:** The LAN780x will appear as a network interface on the host PC. Verify that it is detected and that a driver is loaded for the interface. See Section Appendix C:, "How to Verify an Ethernet Connection" for detailed instructions.
2. **Try different cables:** Ensure the cable being used are Cat5e or a higher quality alternative. Cat5 and below are not suitable for Gigabit Ethernet. It is very common for Ethernet cables to be unlabeled, and it is often difficult to verify their rating. When in doubt, purchase new cables from a reputable vendor. Try both long and short cables to see if a connection can be established.
3. **Try a direct connection:** Manually set the Static IP address of the LAN780x through the operating system network configuration options. Manually set the static IP of another verified working computer. Make a direct connection from the LAN780x under test to a verified working computer and try a ping test to the computer's static IP address.
4. **Probe for NLP/FLP:** While searching for a connection, the Ethernet transmitter will issue Normal Link Pulses (NLP) or Fast Link Pulses (FLP) on the wire. (See Figure 6.) If the Ethernet interface is configured for 10 Mbit/s speeds only, a NLP 100 nanosecond pulse is sent every 16 milliseconds. If Auto-negotiation is enabled, a group of 33 FLP pulses are sent, one after another, every 16 milliseconds. An oscilloscope can be used to observe these pulses. If the pulses are present, then the Ethernet interface is enabled and highly functional.

Note: The FLP pulses can be manually decoded to discover the capabilities that the Ethernet interface is advertising. For example, this can be used to verify/correlate the configuration settings of the LAN780x. For more information, consult Microchip's *AN1120 "Ethernet Theory of Operation"* section on AUTO-NEGOTIATION.

FIGURE 6: NLP/FLP PULSES



5. **Check magnetics:** Selecting proper magnetics for the LAN7800 is important for ensuring proper connectivity. Refer to *AN8.13 – Suggested Magnetics* for suggestions. If using LAN7801 with an external PHY, follow the guidance of the PHY supplier for selection of magnetics.
6. **Check Connector Schematic Pinout:** Double check the pinout of your selected Ethernet connector against the connector's data sheet.
7. **Check USB connector PCB layout footprint:** Verify that the PCB footprint pinout matches the description of the connector as described in the connector's data sheet.
8. **[LAN7801 Only] Verify MDIO operation:** LAN7801 requires connection to an external Ethernet PHY which is controlled via MDIO. Most PHYs don't require special drivers for basic Ethernet connectivity and the generic MDIO driver included in the operating system is sufficient for configuring the PHY to basic operation. Probe the MDIO lines to make sure communication is present, and an MDIO analyzer can be used to decode communication if needed. See *AN4754 - Using Microchip Bridge Controllers with External Ethernet PHYs* for more details.
9. **[LAN7801 only] Verify TXC and RXC clock delay settings:** The TXC and RXC clock signals must be delayed relative to the RGMII data signals for proper operation. This delay can be enabled in either the LAN780x settings, or the PHY settings. Ensure that the clock delay is enabled in either the MAC or the PHY. See *AN4754 - Using Microchip Bridge Controllers with External Ethernet PHYs* for more details.
10. **[LAN7801 Only] Verify external 125 MHz clock source:** Verify that an external 125 MHz clock is being supplied to the LAN7801. If an external clock is not supplied, enable the internal 125 MHz clock source via driver, EEPROM, or OTP configuration settings on the LAN7801. See *AN4754 - Using Microchip Bridge Controllers with External Ethernet PHYs* for more details.
11. **Collect an Ethernet protocol trace:** Use Wireshark to collect an Ethernet protocol trace to see if any communication is present. If Wireshark cannot be run on the host of LAN780x, then a discrete Ethernet protocol analyzer or Ethernet protocol tap device (such as a SharkTap) can be used.
12. **Submit a Support Case to Microchip:** If the Ethernet connectivity issue is still not resolved or understood after

following these debug methods, jump back to Section 6.1, "Hardware Issue – Fundamental PCB Design" and Section 6.2, "Hardware Issue – PCB Assembly or IC Damage" and review all guidance. If the issue is still not resolved or understood, submit a support case to Microchip support with all of your findings clearly documented. Be sure to include the answers to the questionnaire in Section 4.0, "Problem Statement Questionnaire" and all data logs described in Section 5.0, "Data Logging" in the support case submission to accelerate the time to submission. (If you cannot collect any of the request logs for any reason, state this upfront in the case submission as well).

6.6 Hardware Issue – Ethernet Interface Configuration Issue

1. **MAC Address:** Every Ethernet component on a network must have a unique MAC Address. If two components in the network share the same MAC Address, communication issues will occur. Typically, MAC Addresses are obtained through the IEEE Registration Authority and are programmed to each Ethernet interface during production. Verify the MAC Address of your equipment and ensure a unique value is being programmed before using in a network.

Note:	It is possible to program a random value to the MAC Address register of LAN780x via EEPROM/OTP, or manually override the MAC address through the LAN780x driver options to be able use the device during development and test. If a random address is selected, the probability that another device on the network has the same address is quite low. This is not an acceptable approach, however, for typical end products in production.

2. **IP Address:** Ensure that the IP address is not being manually set to a static value through the network options in the operating system. In most cases, the IP Address should be assigned automatically by the network at the time of connection.
3. **Auto-negotiation and Forced Speed Selection:** It is highly recommended to always enable all connection speeds and leave auto-negotiation enabled on LAN780x. Manually disabling auto-negotiation and forcing specific connection speeds is acceptable but should be done with great care and only if there is a specific reason to do so.
4. **Submit a Support Case to Microchip:** If The Ethernet connectivity issue is still not resolved or understood after following these debug items, jump back to Section 6.1, "Hardware Issue – Fundamental PCB Design" and Section 6.2, "Hardware Issue – PCB Assembly or IC Damage" sections and review all guidance. If the issue is still not resolved or understood, submit a support case to Microchip support with all of your findings clearly documented. Be sure to include the answers to the questionnaire in Section 4.0, "Problem Statement Questionnaire" and all data logs described in Section 5.0, "Data Logging" in the support case submission to accelerate the time to submission. (If you cannot collect any of the request logs for any reason, state this upfront in the case submission as well).

6.7 Performance Issue – Ethernet/USB Throughput Issue

Run the iPerf/iPerf3 tests described. Do not skip any steps. Once results are collected, compare them directly. In a well-functioning end application, the results for Tests #1-3 should be approximately equivalent. If there are differences, note where the differences are.

Test #1: Direct-connect LAN780x system to a known good computer with known good cables. Remember to set static IP addresses to both systems before running the same iPerf/iPerf3 tests.

Terminate all non-critical programs and processes on the system connected to LAN780x to minimize the chances that CPU loading will impact the performance measurement.

Test #2: Connect LAN780x system to the local network and rerun same throughput tests. Compare the results to those collected in Test #2 (including terminating non-system-critical programs and processes). Record the results.

Test #3: Repeat the same measurements as in Steps 2/3, but with all end-application software running within the system. Record the results.

6.7.1 IF PERFORMANCE FOR TEST #1 IS UNACCEPTABLE

If performance for Test #1 is unacceptable, a fundamental issue is present. The issue could be one of root causes specified in Table 3.

TABLE 3: POSSIBLE PERFORMANCE ISSUE ROOT CAUSES IF TEST #1 RESULTS ARE UNACCEPTABLE

Possible Root Cause	How to Verify
Hardware problem on LAN780x design	Jump back to Section 6.1, "Hardware Issue – Fundamental PCB Design" and Section 6.2, "Hardware Issue – PCB Assembly or IC Damage" and review all guidance again.
Bad Ethernet cable	Purchase Cat5e or higher-quality cables of various lengths from a reputable brand and retest.
Poor signal integrity from LAN780x system. How to verify?	Perform USB2, USB3, and Ethernet electrical compliance testing on the LAN780x system.
The processor connected to LAN780x does not have sufficient computing power to achieve Gigabit Ethernet line speed.	Acquire an EVB-LAN7800 and other known-good USB to Ethernet products and repeat tests. If the tests show similar results, this suggests that the processor performance cannot keep up with Ethernet gigabit speed line rate.
Interoperability problem between LAN780x and a known-good PC.	Try performance measurements with several other known-good computers to see if the issue is localized to a certain computer or type of computers.
LAN780x driver has a problem.	First, ensure the driver is updated to the latest version. LAN780x drivers are continually updated and improved. LAN780x driver releases are quality tested for performance, but new performance issues and incompatibilities may still exist under certain use-case conditions or within certain systems. If the issue performance persists after performing an update to the latest release, and all other possibilities from this list are eliminated, there may be a driver-incompatibility issue with your specific system. Submit a support case to Microchip support with all of your findings clearly documented. Be sure to include the answers to the questionnaire in Section 4.0, "Problem Statement Questionnaire" and all data logs described in Section 5.0, "Data Logging" in the support case submission to accelerate the time to submission. (If you cannot collect any of the request logs for any reason, state this upfront in the case submission as well.)

6.7.2 IF PERFORMANCE FOR TEST #1 IS GOOD BUT TEST #2 IS UNACCEPTABLE

Something within the network itself is likely causing the slowdown. Consider the following questions in Table 4 and try to find a pattern or dependency.

TABLE 4: POSSIBLE PERFORMANCE ISSUE ROOT CAUSES IF TEST #2 RESULTS ARE UNACCEPTABLE

Possible Root Cause	How to Verify
Network does not have enough bandwidth or performance.	Run iPerf/iPerf3 measurements between several other computers on the network. Do they perform at full line speed with the LAN780x connected to the network? Do the measurements between computers in the network change when the LAN780x is removed from the network?
Interoperability problem with network components	Record what the LAN780x is connected to (i.e.:an unmanaged switch, managed switch, router, etc.) Try connecting to the network through other ports or through other switches. Do the results change?

TABLE 4: POSSIBLE PERFORMANCE ISSUE ROOT CAUSES IF TEST #2 RESULTS ARE UNACCEPTABLE

Possible Root Cause	How to Verify
Configuration issue with LAN780x	Revert LAN780x to default settings (except for MAC address). Ensure that auto-negotiation is enabled and all connection speeds are enabled. Verify the connection speed of LAN780x while connected to the network.
LAN780x driver has a problem.	First, ensure the driver is updated to the latest version. LAN780x drivers are continually updated and improved. LAN780x driver releases are quality tested for performance, but new performance issues and incompatibilities may still exist under certain use-case conditions or within certain systems. If the issue performance persists after performing an update to the latest release, and all other possibilities from this list are eliminated, there may be a driver incompatibility issue with your specific system. Submit a support case to Microchip support with all of your findings clearly documented. Be sure to include the answers to the questionnaire in Section 4.0, "Problem Statement Questionnaire" and all data logs described in Section 5.0, "Data Logging" in the support case submission to accelerate the time to submission. (If you cannot collect any of the request logs for any reason, state this upfront in the case submission as well.)

6.7.3 IF PERFORMANCE FOR TESTS #1 AND #2 ARE GOOD, BUT TEST #3 IS UNACCEPTABLE

Something in the end application software is causing the slowdown. See Section 6.8, "Performance Issue – Application Level Issue" for recommendations on the next steps.

6.8 Performance Issue – Application Level Issue

If LAN780x performance suffers only within the end application, with the end application software running, then the issue is mostly one of the following:

- The processor does not have enough power to handle the end application software.
- The CPU cores are not being utilized properly.
- The end application software has an optimization problem.

Strategies for dealing with end application issues vary widely depending on the operating system, processor, and software development tools. Solutions for these types of issues are outside the scope of this document.

APPENDIX A: HOW TO VERIFY A USB2 CONNECTION

Even if the design is primarily curated around USB3 connectivity, verification of USB2 connectivity is important. This ensures the application can still function at a basic level even if the USB3 interface becomes non-functional or there is a cabling issue.

A.1 Basic Verification

In order to verify the USB2 connectivity, do the following:

- In a cabled application:
  - Connect the device to the USB host using a USB2-only cable.

Note:	If using a USB Type-C cable, take extra care to ensure it is a USB2-only capable. Type-C cables do come in USB2.0-only varieties, but it is far more common for Type-C cables to be fully USB3 capable, and Type-C cables are not usually well-labeled with their capabilities.

- In an embedded application:
  - Option A: Disable the USB3 operation of the host SOC/CPU.
  - Option B: Physically remove the DC blocking capacitors on the USB3 traces to force USB2 roll-back.

Next, perform the steps in the succeeding sections to verify connectivity:

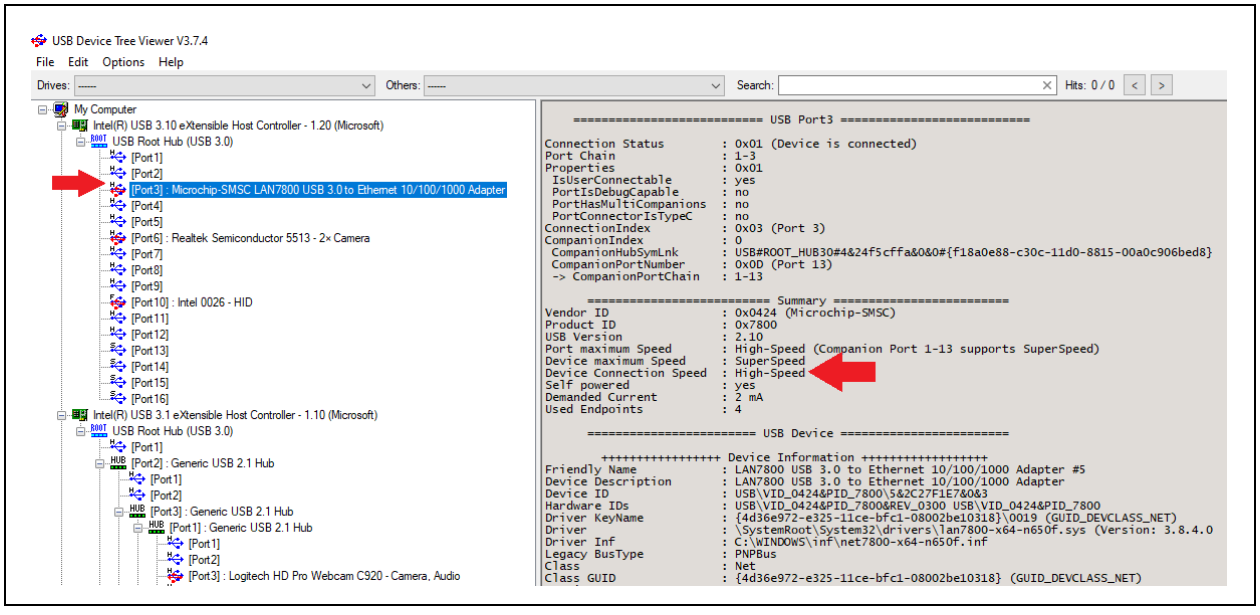
A.1.1 WINDOWS®

1. Obtain a USB Tree Viewing application such as the one available here: https://www.uwe-sieber.de/usb-treeview_e.html

Note:	Perform an Internet search for “USB tree viewer” if the above URL becomes dead.

2. Open the Tree Viewer application and search for the LAN780x device. (See Figure 7.) If you cannot find it, try disconnecting the LAN780x and then reconnecting and watching for any changes in the screen.

FIGURE 7: VERIFYING LAN780X USB2 CONNECTION IN WINDOWS®



3. Confirm connection speed. This can be done by looking at two places:

a) In the main tree representation, the USB icon to the left of the LAN780x device text will have a small “H” next to it, indicating High-Speed (USB2).

b) After selecting the LAN780x device in the tree view, the device details will be displayed. The Connection Speed (as well as maximum supported speed) are shown. A connection speed of “High-Speed” means the LAN780x is connected at USB2 speeds.

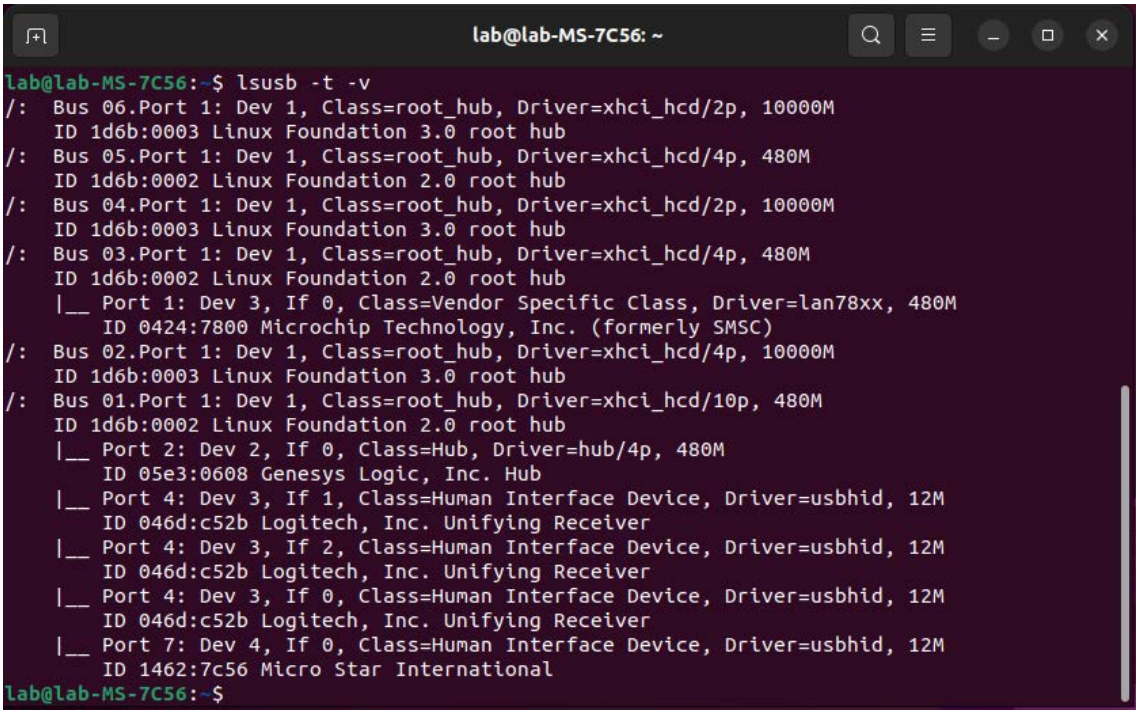
A.1.2 LINUX®

1. Linux® includes a basic command for viewing details about all connected devices. This command is called ‘lsusb’. (See Figure 8.) To verify the connection speeds of all connected devices, enter the following command into the Linux command terminal:

```
lsusb -t -v
```

2. The ‘-t’ argument will show the USB bus in tree form.
3. The ‘-v’ argument shows additional details beyond what a standard ‘lsusb’ command will display (including the connection speed).

FIGURE 8: VERIFYING LAN780X USB2 CONNECTION IN LINUX®



4. Search through the output for a device with ID ‘0424:7800’ or ‘0424:7801’. The text 480M indicates 480 Mbit/s connection speed, which means the LAN780x is connected at USB2 speeds.

Note:	0x0424 is Microchip’s USB Vendor ID.

A.1.3 MACOS®

1. The USB Device Tree can be seen in macOS® through the following:

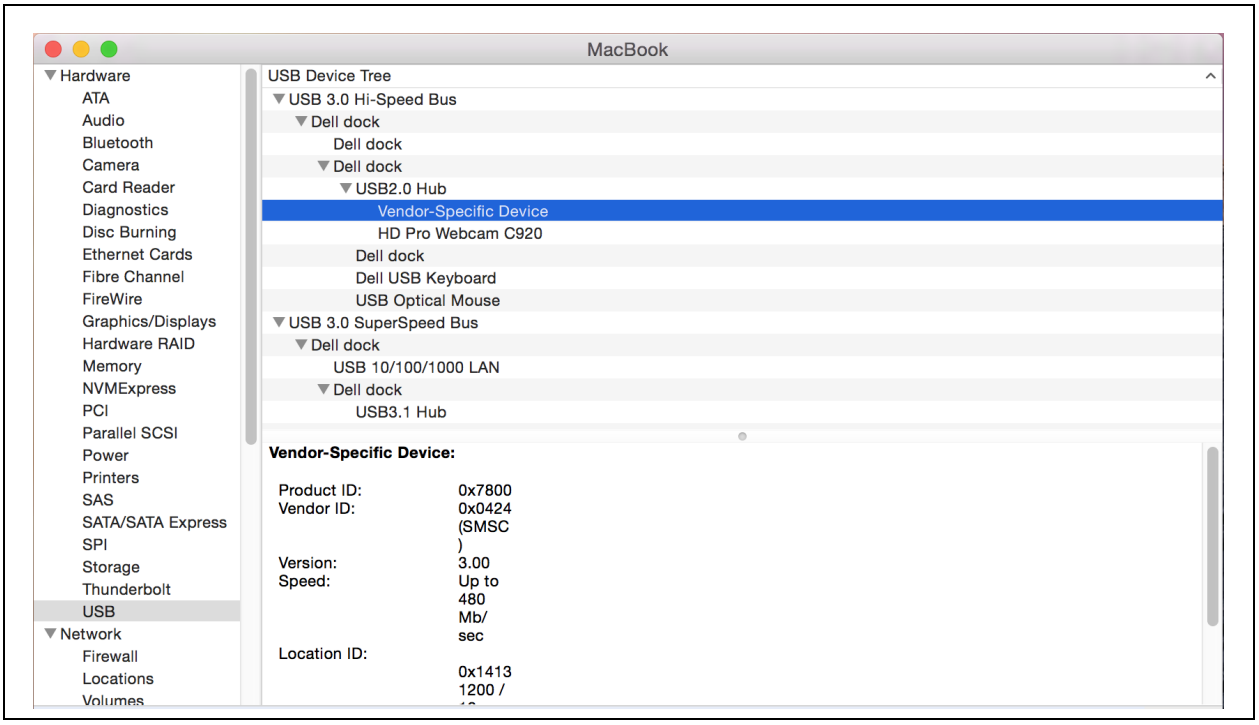
Apple dropdown menu> About this Mac > System Report >Hardware >USB

Note: In the event that this pathway becomes non-existent in the future versions of macOS, do an Internet search for “How to see the USB Device Tree for macOS” to find the correct menu navigation pathway.

2. USB2 devices will be shown under “USB 3.0 Hi-Speed Bus” or similarly named option. See Figure 9.

Note: The USB2.0 bus in macOS Device Tree may be shown as “USB 3.0 Hi-Speed Bus,” which may be easily misinterpreted as a USB3 speed bus.

FIGURE 9: VERIFYING LAN780X USB2 CONNECTION IN MACOS®



A.2 Mid-Confidence Verification

Connect a USB2 protocol analyzer in between the host and the LAN780x. Ensure the buffer size of the protocol analyzer is collecting a sufficient amount of data (i.e.: more than 1 GB). Open the protocol analyzer trace and run an error report. (Each analysis software varies, so consult the documentation of your selected protocol analyzer for details.) Errors are expected to be occasionally present in a USB connection and the presence of an error does not necessarily mean a major problem exists, but in most cases, you should expect to see zero errors. Specifically, look for the following error types which could indicate poor signal quality or a marginally functional host/hub/device (e.g.: inaccurate clock, bad ground, etc.).

- CRC errors – This means data was corrupted at some point in the USB signal chain during transmission.
- Turnaround errors – This kind of error means the addressed device did not detect the packet and hence did not respond to it.

A.3 High-Confidence Verification

For applications with high quality and robustness requirements, it is strongly recommended to complete USB2 product certification testing, or at minimum, USB2 electrical signal quality testing (USB2 eye diagrams). This can be done at an external USB-IF approved test lab if the equipment and expertise are not available internally. Visit www.usb.org and navigate to the “Independent Test Labs” page for a list of approved labs located around the globe.

APPENDIX B: HOW TO VERIFY A USB3 CONNECTION

When a USB3 connection is present, the LAN780x will always connect at USB3 speeds automatically. The LAN780x does not need special configuration or special handling to connect at USB3 speeds.

B.1 Basic Verification

To verify the USB3 connectivity, do the following:

- In a cabled application:
  - Connect the device to the USB host using a USB3-capable cable.

Note:	If using a USB Type-C cable, take extra care to ensure it is USB3 capable. Type-C cables do come in USB2.0-only varieties and Type-C cables are not usually well-labeled with their capabilities.

- In an embedded application:
  - Ensure the USB3 interface of the SOC is enabled and functional.

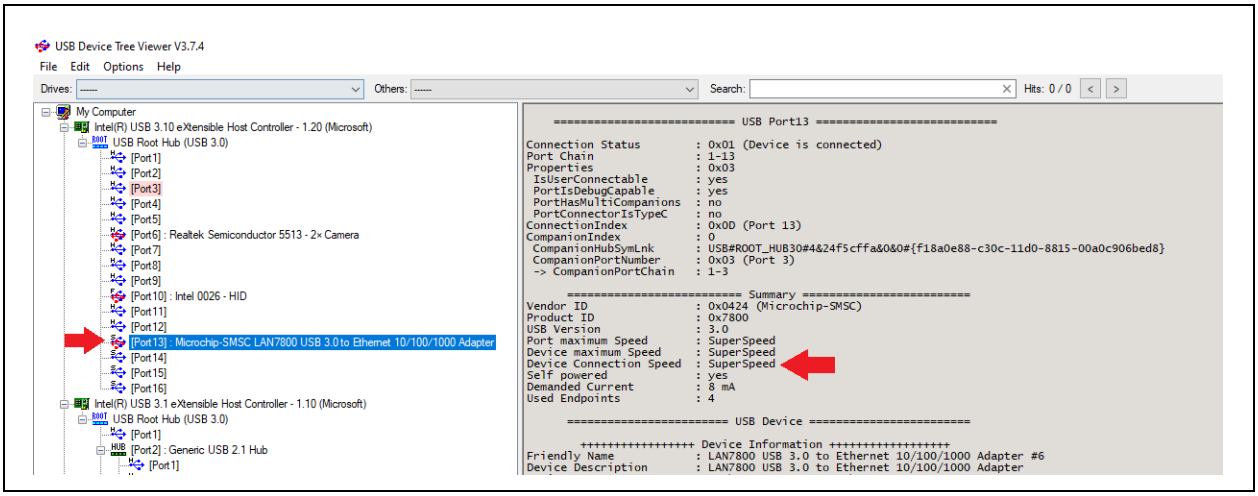
B.1.1 WINDOWS

1. Obtain a USB Tree Viewing application, such as the one available here: https://www.uwe-sieber.de/usb-treeview_e.html.

Note:	Perform an Internet search for “USB tree viewer” if the above link becomes dead.

2. Open the Tree Viewer application and search for the LAN780x device. (See Figure 10.) If you cannot find it, try disconnecting the LAN780x and then reconnecting. Watch out for any changes in the screen.

FIGURE 10: VERIFYING LAN780X USB3 CONNECTION IN WINDOWS®



3. Confirm connection speed by looking at the following places:
   a) In the main tree representation, the USB icon to the left of the LAN780x device text will have a small “S” next to it, indicating Super-Speed (USB3).
   b) After selecting the LAN780x device in the tree view, the device details will be displayed. The connection speed (as well as maximum supported speed) are shown. A connection speed of “SuperSpeed” means the LAN780x is connected at USB3 speeds.

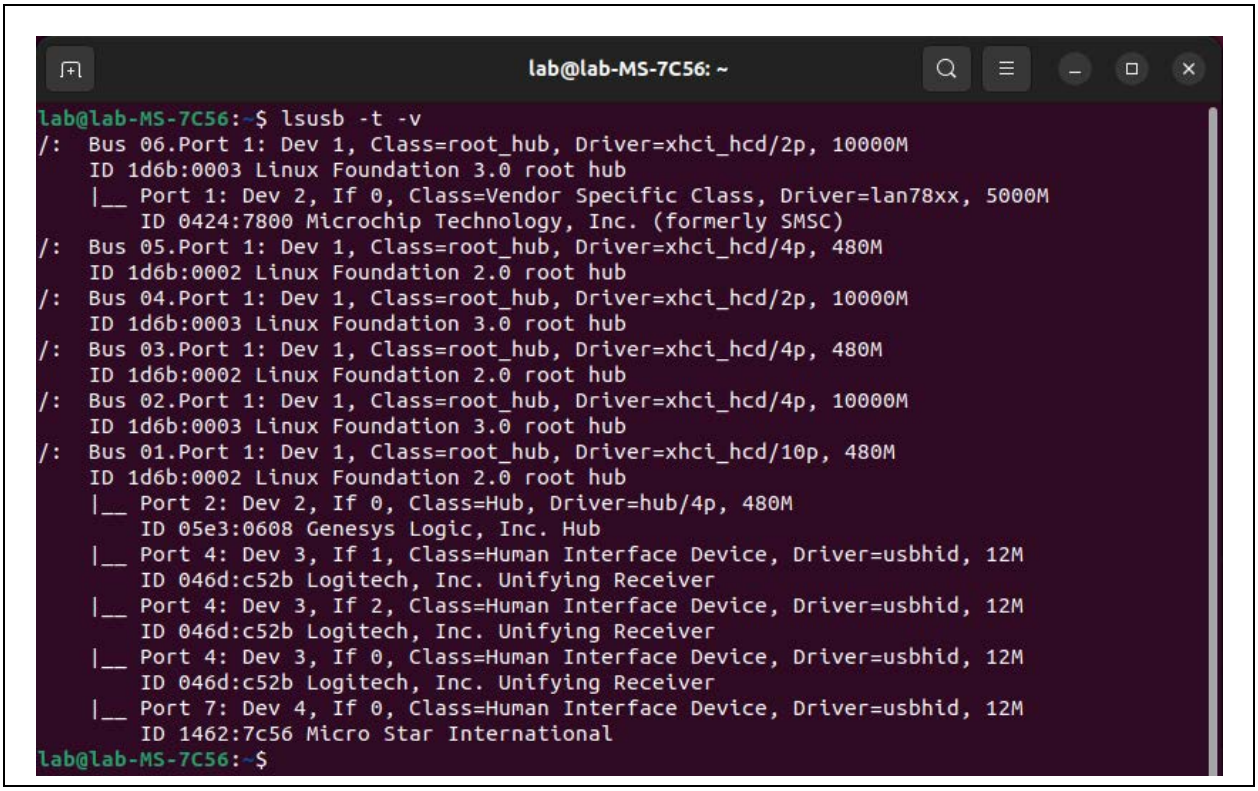
B.1.2 LINUX

1. Linux includes a basic command for viewing details about all connected devices. This command is called 'lsusb'. (See Figure 11.) To verify the connection speeds of all connected devices, enter the following command into the Linux command terminal:

```
lsusb -t -v
```

2. The '-t' argument will show the USB bus in tree form.
3. The '-v' argument shows additional details beyond what a standard 'lsusb' command will display (including the connection speed).

FIGURE 11: VERIFYING LAN780X USB3 CONNECTION IN LINUX®



4. Search through the output for a device with ID 0424:7800 or 0424:7801. The text, "5000M," indicates 5 Gbit/s connection speed, which means LAN780x is connected at USB3 speeds.

Note:	0x0424 is Microchip's USB Vendor ID.

B.1.3 MACOS

The USB Device Tree can be seen in macOS through the following:

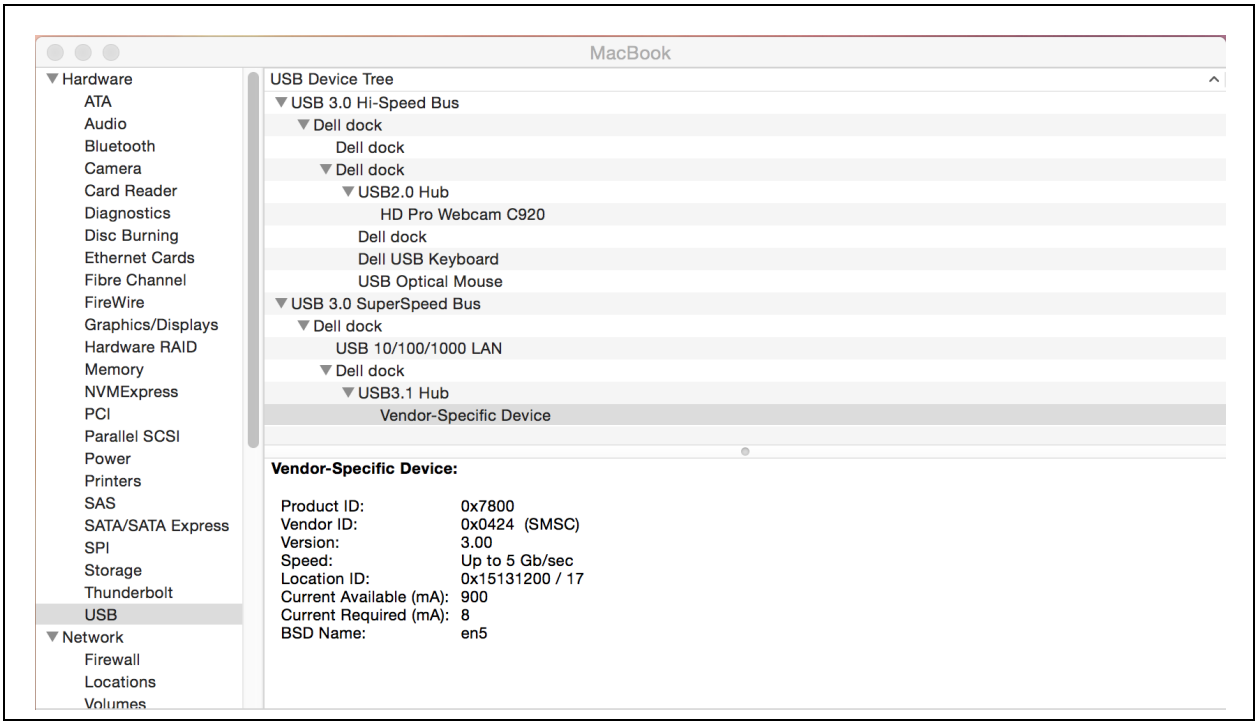
Apple dropdown menu>About this Mac>System Report>Hardware>USB

Note:	In the event that this pathway becomes non-existent in the future versions of macOS, do an Internet search for "How to see the USB Device Tree for macOS" to find the correct menu navigation pathway.

USB3 devices will be shown under "USB 3.0 SuperSpeed Bus" or similarly named option.

Note: The USB2.0 bus shown in macOS Device Tree may be displayed as “USB 3.0 Hi-Speed Bus,” which may be easily misinterpreted as a USB3 speed bus. Refer to Figure 12.

FIGURE 12: VERIFYING LAN780X USB3 CONNECTION IN MACOS®



B.2 Mid-Confidence Verification

Connect a USB3 protocol analyzer between the host and the LAN780x. Ensure the buffer size of the protocol analyzer is collecting a sufficient amount of data (i.e.: more than 1 GB). Open the protocol analyzer trace and run an error report. (Each analysis software varies, so consult the documentation of your selected protocol analyzer for details.) Errors are expected to be occasionally present in a USB connection and the presence of an error does not necessarily mean a major problem exists, but in most cases, you should expect to see zero errors. Specifically, look for the following error types which could indicate poor signal quality or a marginally functional host/hub/device (e.g. inaccurate clock, bad ground, etc.).

- CRC errors –This means data was corrupted at some point in the USB signal chain during transmission.
- Turnaround errors – This kind of error means the addressed device did not detect the packet and hence did not respond to it.
- Check the LTSSM state machine diagram and make sure entry into recovery is not happening frequently (a sign that communication is marginal).

B.3 High-Confidence Verification

For applications with high quality and robustness requirements, it is strongly recommended to complete USB3 product certification testing, or at minimum, USB3 electrical signal quality testing, including both transmitter and receiver jitter tolerance testing. This can be done at an external USB-IF approved test laboratory if the equipment and expertise are not available internally. Visit www.usb.org and navigate to the “Independent Test Labs” page for a list of approved labs located around the globe.

APPENDIX C: HOW TO VERIFY AN ETHERNET CONNECTION

C.1 Basic Verification

The method to verify an Ethernet connection varies depending on the operating system. Follow the specific guidance below.

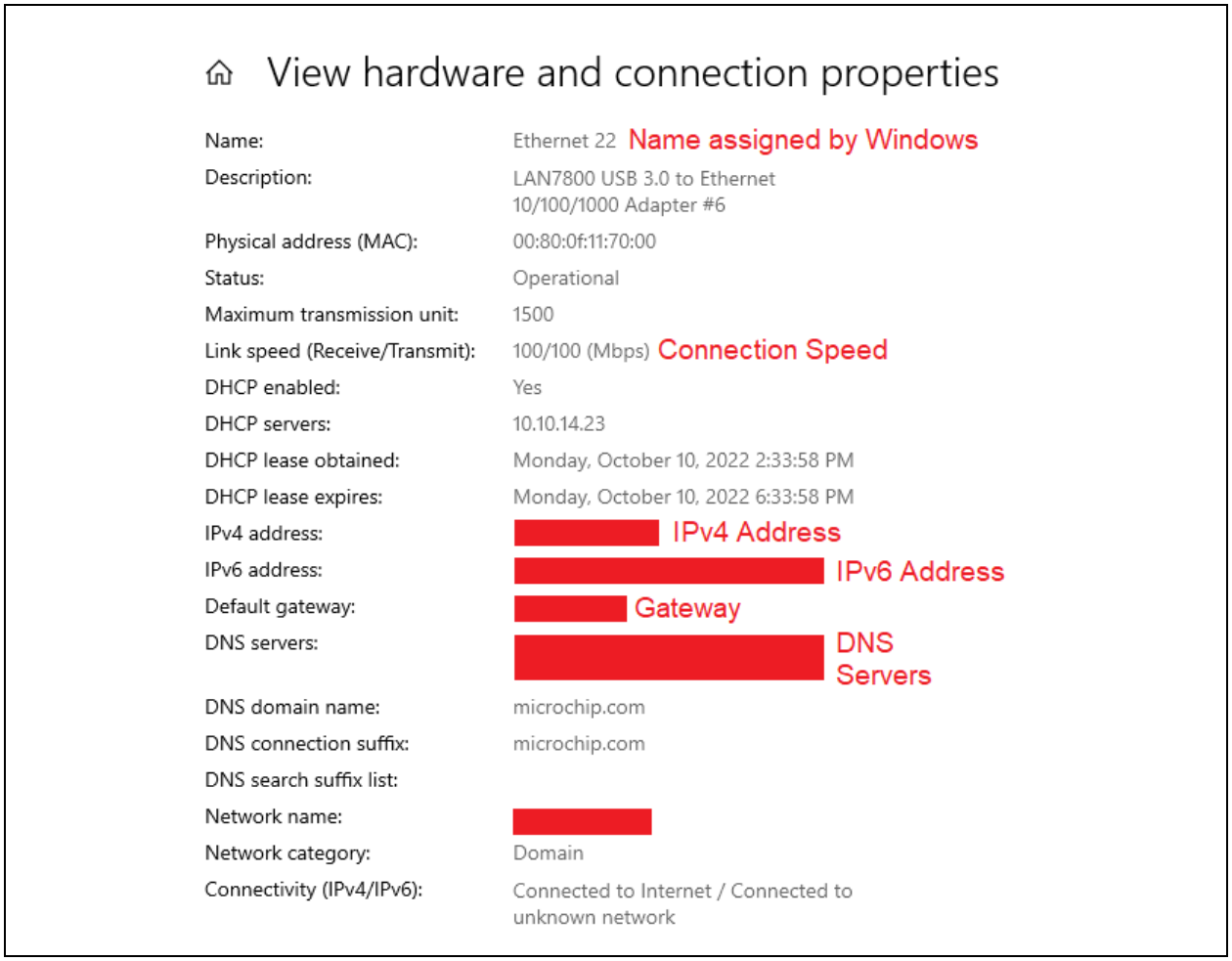
C.1.1 WINDOWS®

These instructions are for Windows®10. Certain sequences or steps may be different in other versions of Windows.

C.1.1.1 Connection Status

1. From the Windows menu, select Settings and choose Network and Internet. Select the View hardware and connection properties option.
2. From this screen, scroll down until the LAN780x device is found. The network interface name assigned by Windows is shown in Figure 13, as well as connection speeds, IP addresses, etc.

FIGURE 13: CONNECTION STATUS INFORMATION IN WINDOWS®



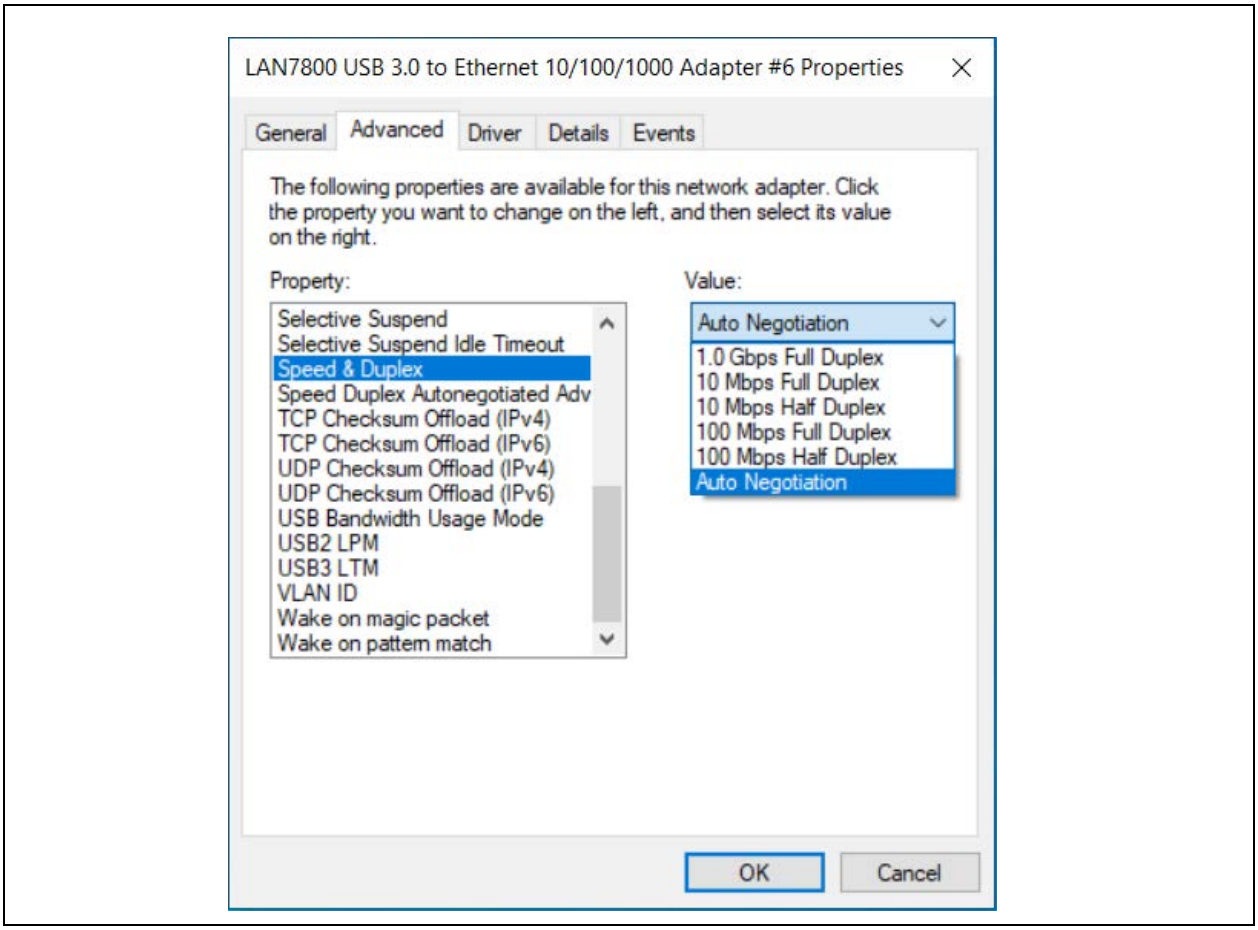
C.1.1.2 Driver Settings

The driver settings should also be verified to ensure all key settings are configured as expected. Perform the following to access the driver settings:

1. Navigate to *Control Panel >Network and Internet>Network Connections*.

2. Select the LAN780x device – Windows will give this a unique name such as Ethernet “N” where N is some number.
3. Once the device is selected and interface and connection details are displayed, select the “Properties” button. (You must have administrative rights to access).
4. Select the “Configure...” button.
5. Move to the “Advanced” tab. See Figure 14.
6. Open each option and ensure the selected value matches expectations for the device.

FIGURE 14: LAN780X DRIVER ADVANCED SETTINGS IN WINDOWS®



C.1.2 LINUX

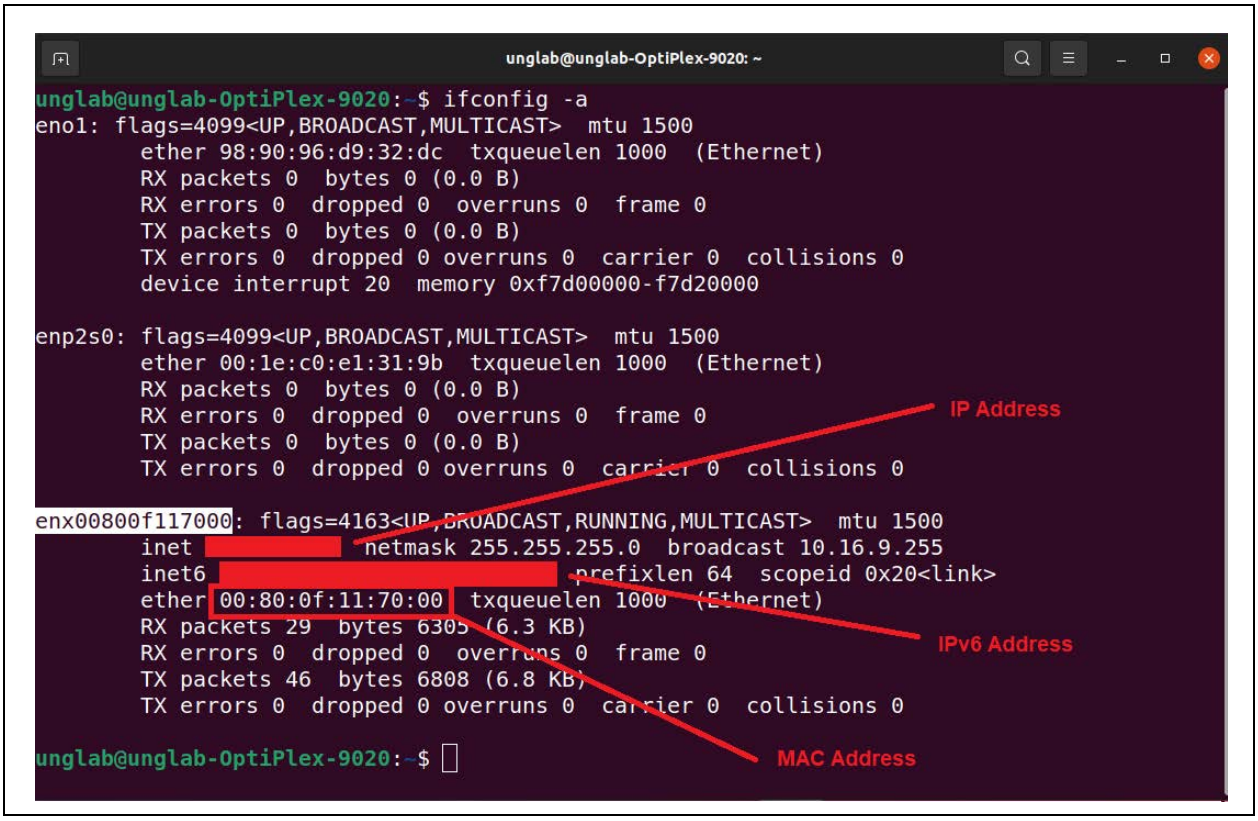
In a Linux environment, two utilities are commonly used to conduct basic Ethernet connection verification:

1. ifconfig – A networking tool that allows the user to display network configuration information, setup IP addresses/ netmasks, setting hardware addresses and enabling/disabling interfaces.
2. ethtool – Ethernet specific tool for getting advanced details, and performing advanced tasks such as programming EEPROM configuration.
3. Display all network interfaces and find the name assigned to LAN7800.
4. Issue the following command:

```
ifconfig -a
```

5. All network interfaces will be displayed to the user along with IP addresses, MAC addresses, and some basic traffic statistics. A sample printout of an ethtool examination of a LAN7800 device is shown in Figure 15.

FIGURE 15: LAN780X IFCONFIG PRINTOUT

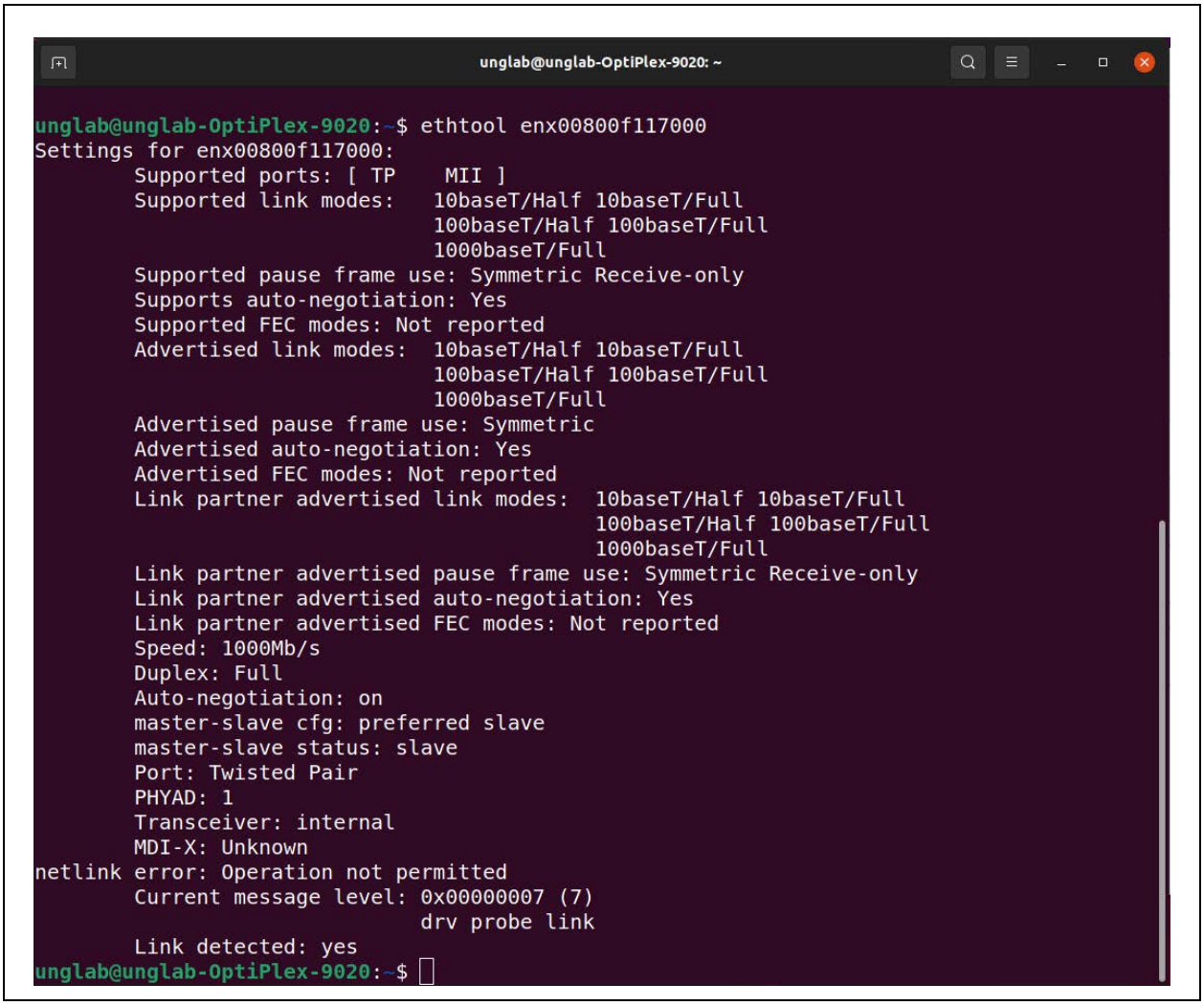


6. Next, examine the interface using ethtool to extract additional information:

```
ethtool <interface>
```

7. Ethtool will display key parameters about the interface configuration, including supported speeds, auto-negotiation support, details about the Link partner, the current connection speed, the current connection duplex, and more. An example printout of an ethtool examination of a LAN7800 device is illustrated in Figure 16.

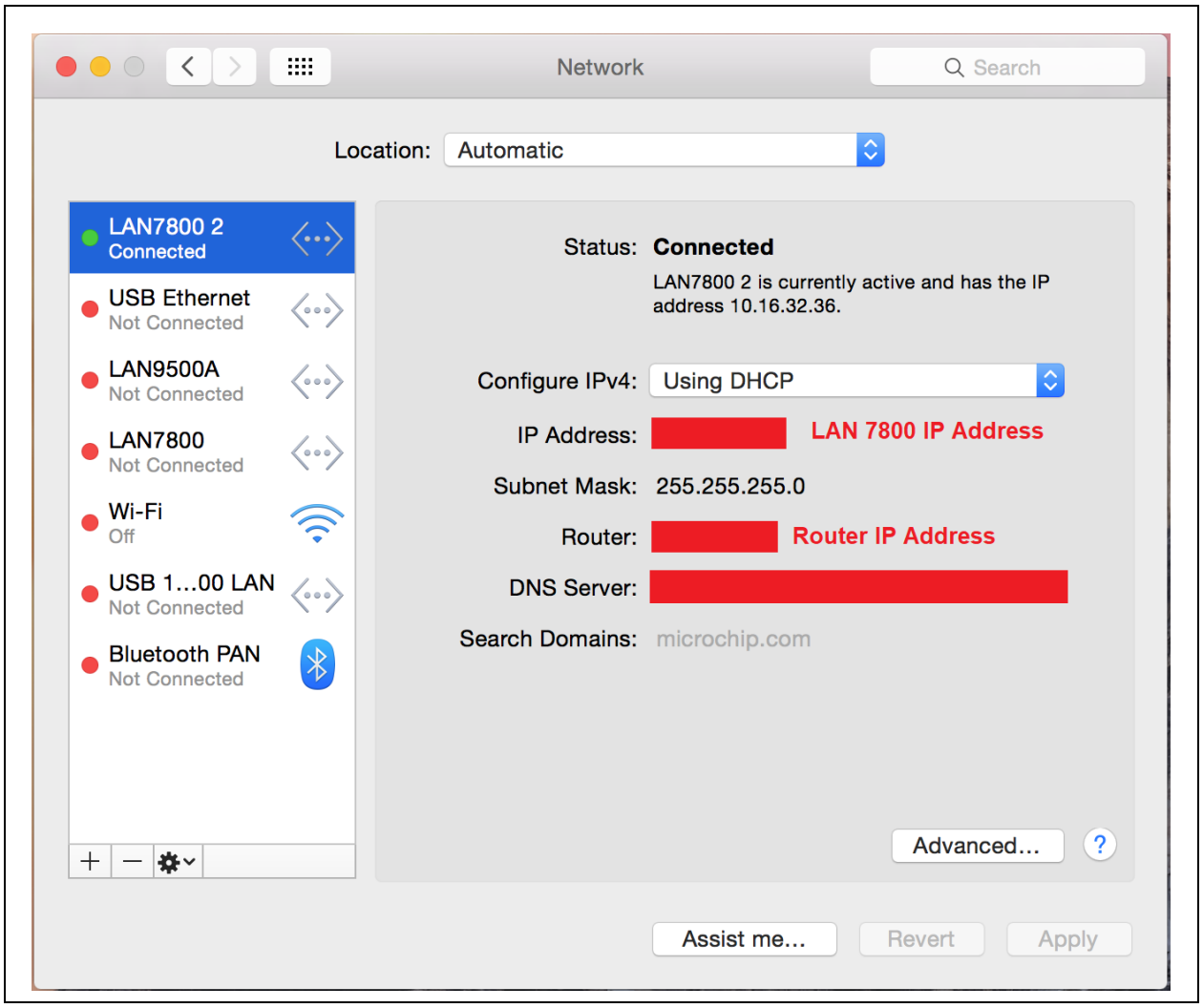
FIGURE 16: LAN780X ETHTOOL PRINTOUT



C.1.3 MACOS

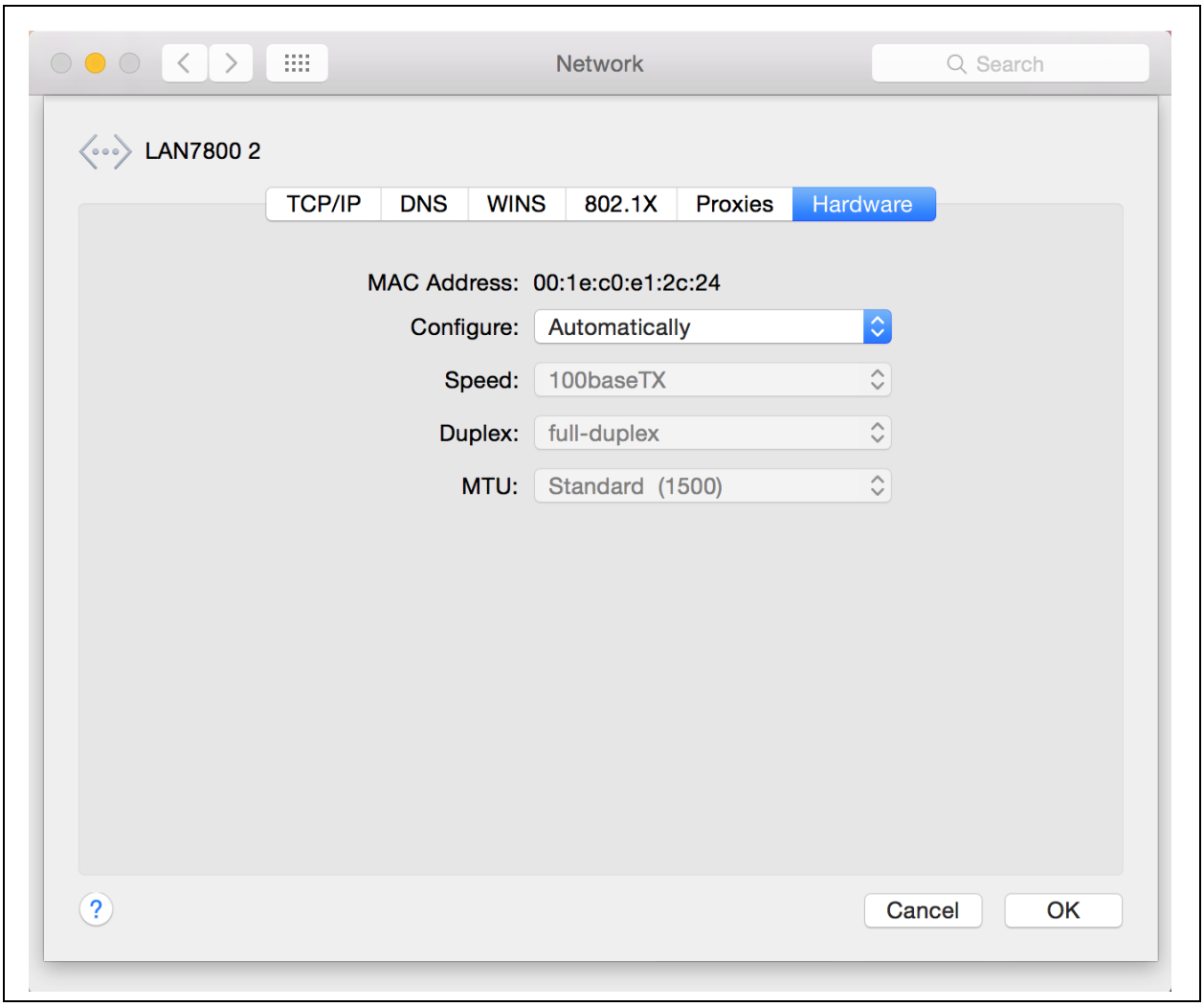
On your Mac, choose *Apple Menu >System Settings (or System Preferences)*, then click on Network. Select the network interface of interest and view the network connection status on the right. IP Addresses and DNS servers will be displayed, as well as basic connection status. Refer to Figure 17.

FIGURE 17: MACOS® NETWORK STATUS OF LAN7800



To view hardware configuration settings, click on the **Advanced** button, and navigate to the **Hardware** tab. The connection speed and options for manual selection of settings are available as shown in Figure 18.

FIGURE 18: MACOS® ADVANCED HARDWARE OPTIONS FOR LAN7800



C.2 Mid-Confidence Verification

Performing network speed tests is a good way to verify the Ethernet connection beyond simply checking for connection status and speed. iPerf/iPerf3 is the most common utility for this. These tests are further discussed in Appendix D: "How to Perform Ethernet Performance Measurements and Interpret Results".

C.3 High-Confidence Verification

An independent compliance and interoperability testing lab is the best way to ensure a robust solution. The University of New Hampshire InterOperability Laboratory (UNH-IOL) is the best option for this kind of testing. Visit their website for details: https://www.iol.unh.edu/testing/ethernet.

APPENDIX D: HOW TO PERFORM ETHERNET PERFORMANCE MEASUREMENTS AND INTERPRET RESULTS

It is critical to use a consistent and methodical approach to measuring Ethernet performance. The following steps are recommended:

1. Identify two high-performing computers and some Cat5e Ethernet cables or others of similar quality. Always verify the computers and cables first by running independent performance tests on the test computer without connecting to the LAN780x system in question. A computer capable of achieving full Gigabit Ethernet throughput at line speed is needed. Gigabit Ethernet line speed is approximately 950 Mbps on both TX and RX bidirectionally.
2. Install iPerf3 on both computers. iPerf3 is the industry standard tool and the most common tool to use for Ethernet throughput measurement. iPerf3 is a newer, actively maintained version of iPerf. Nevertheless, iPerf is still widely used. The same version of iPerf3 (or iPerf) should be installed on both computers. If different versions are used, the results may be unreliable. Visit https://iperf.fr/ for installers or instructions. Both iPerf3 and iPerf can be installed on the same computer if desired.

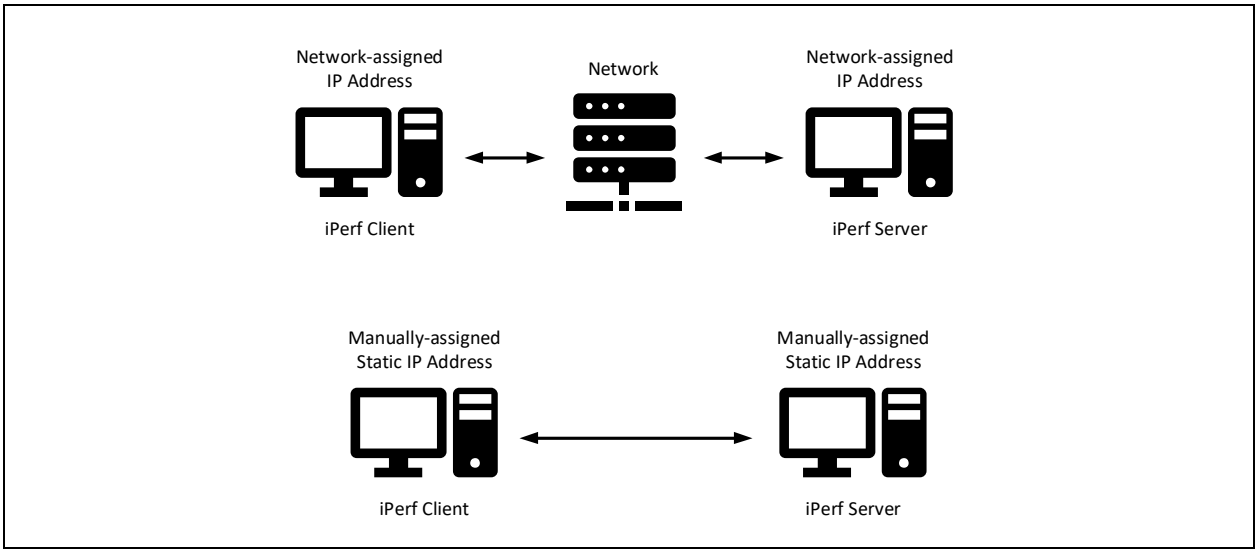
Note:	On Linux distributions, iPerf3/iPerf can be installed using a simple 'apt-get' type command. Do an Internet search for the exact command prompt needed to install iPerf3/iPerf on your exact Linux distribution.

D.1 Setup

One computer will need to act as the iPerf server, while the other operates as the client. In theory, it should not make any difference which computer is set up for which role as transfer direction is controlled via the test command prompts.

The server and client can be connected to a typical network, or connected directly together, as shown in Figure 19.

FIGURE 19: LAN780X DRIVER ADVANCED SETTINGS IN WINDOWS®



When connected directly together, the Ethernet interface must be configured for a manually selected Static IP. The selected IP address is not important so long as the two computers are not assigned the same address.

D.2 Basic Commands

D.2.1 SERVER

Run the following command on the computer selected to be the server to enable the TCP server in iPerf3 to be used for TCP client tests.:

```
sudo iperf3 -s
```

Note: iPerf3 listens on port 5201 by default. You may have to open port 5201 on your firewall if connected through a network. Alternatively, you can specify a different port through the '-p' iPerf argument.

While in Server mode, no further interaction with the computer is required. All of the test options can be selected through commands on the client. The following will be displayed on the screen after the Server mode is enabled.

```
-----------------------------------------------------------
Server listening on port 5201
-----------------------------------------------------------
```

Additional test results will be printed on the server terminal once tests are run on the client side.

Note: If using iPerf (not iPerf3), separate commands are necessary for opening a TCP server and a UDP server. These are 'iperf -s' for TCP server and 'iperf -s -u' for UDP server. The default port for iPerf is 5001 (as opposed to 5201 on iPerf3).

D.2.2 CLIENT TEST COMMANDS

On the client computer, issue all of the following commands to get a good survey of performance. Additional tests can be run by varying TCP window sizes (–w), segment sizes (–M), buffer lengths (–l), test duration (–t), and bandwidth (–a for TCP, –b for UDP) to see the effect on performance.

1. Test 1: TCP Send

```
sudo iperf3 -c XXX.XX.XX.XX -V
```

2. Test 2: TCP Receive

```
sudo iperf3 -c XXX.XX.XX.XX -R -V
```

Note: In this example, –R is used to enable Reverse mode where the server sends, client receives.

3. Test 3: TCP Bi-direction Send/Receive

```
sudo iperf3 -c XXX.XX.XX.XX -bidir -V
```

Note: In this example, –bidir is used to enable simultaneous bidirectional mode where two connections with the server are made: one for send and one for receive.

4. Test 4: UDP Send

```
sudo iperf3 -c XXX.XX.XX.XX -u -b 1000M -V
```

Note: In this example, –u is used to select UDP mode, and –b is used to set bandwidth to a maximum theoretical value.

5. Test 5: UDP Receive

```
sudo iperf3 -c XXX.XX.XX.XX -u –b 1000M –R –V
```

Note: In this example, –u is used to select UDP mode, and –b is used to set bandwidth to a maximum theoretical value.

6. Test 6: UDP Bi-direction Send/Receive

```
sudo iperf3 -c XXX.XX.XX.XX -u –b 1000M –bidir –V
```

D.2.3 EXAMPLE RESULTS

The succeeding results were collected by connecting two LAN7800 evaluation boards to two x86-based Linux computers.

1. Test 1: TCP Send

```
iperf 3.9
Linux phxlab-pcie-L1SS 5.15.0-48-generic #54-Ubuntu SMP Fri Aug 26 13:26:29 UTC 2022
x86_64
Control connection MSS 1448
Time: Fri, 14 Oct 2022 20:41:15 GMT
Connecting to host 192.168.0.101, port 5201
      Cookie: 5ykor6y6pt5g7dbczu675wuqnmritesoxqr5
      TCP MSS: 1448 (default)
[  5] local 192.168.0.100 port 34650 connected to 192.168.0.101 port 5201
Starting Test: protocol: TCP, 1 streams, 131072 byte blocks, omitting 0 seconds, 10
second test, tos 0
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-1.00   sec   113 MBytes   948 Mbits/sec    0    225 KBytes
[  5]   1.00-2.00   sec   112 MBytes   942 Mbits/sec    0    250 KBytes
[  5]   2.00-3.00   sec   112 MBytes   942 Mbits/sec    0    250 KBytes
[  5]   3.00-4.00   sec   112 MBytes   943 Mbits/sec    0    263 KBytes
[  5]   4.00-5.00   sec   112 MBytes   941 Mbits/sec    0    263 KBytes
[  5]   5.00-6.00   sec   112 MBytes   942 Mbits/sec    0    263 KBytes
[  5]   6.00-7.00   sec   112 MBytes   941 Mbits/sec    0    263 KBytes
[  5]   7.00-8.00   sec   112 MBytes   941 Mbits/sec    0    263 KBytes
[  5]   8.00-9.00   sec   112 MBytes   941 Mbits/sec    0    263 KBytes
[  5]   9.00-10.00  sec   112 MBytes   941 Mbits/sec    0    263 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
Test Complete. Summary Results:
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-10.00  sec  1.10 GBytes   942 Mbits/sec    0             sender
[  5]   0.00-10.00  sec  1.10 GBytes   941 Mbits/sec                  receiver
CPU Utilization: local/sender 7.2% (0.2%u/7.1%s), remote/receiver 12.2% (1.6%u/10.6%s)
snd_tcp_congestion cubic
rcv_tcp_congestion cubic

iperf Done.
```

2. Test 2: TCP Receive

```
iperf 3.9
Linux phxlab-pcie-L1SS 5.15.0-48-generic #54-Ubuntu SMP Fri Aug 26 13:26:29 UTC 2022
x86_64
Control connection MSS 1448
Time: Fri, 14 Oct 2022 20:41:49 GMT
Connecting to host 192.168.0.101, port 5201
Reverse mode, remote host 192.168.0.101 is sending
      Cookie: l7mqzoyh4wjmu37vosyd7odord7xt4xhu3c4
      TCP MSS: 1448 (default)
[  5] local 192.168.0.100 port 39372 connected to 192.168.0.101 port 5201
```

```
Starting Test: protocol: TCP, 1 streams, 131072 byte blocks, omitting 0 seconds, 10
second test, tos 0
[ ID] Interval           Transfer     Bitrate
[  5]   0.00-1.00   sec   112 MBytes   941 Mbits/sec
[  5]   1.00-2.00   sec   112 MBytes   941 Mbits/sec
[  5]   2.00-3.00   sec   112 MBytes   941 Mbits/sec
[  5]   3.00-4.00   sec   112 MBytes   941 Mbits/sec
[  5]   4.00-5.00   sec   112 MBytes   941 Mbits/sec
[  5]   5.00-6.00   sec   112 MBytes   941 Mbits/sec
[  5]   6.00-7.00   sec   112 MBytes   941 Mbits/sec
[  5]   7.00-8.00   sec   112 MBytes   942 Mbits/sec
[  5]   8.00-9.00   sec   112 MBytes   941 Mbits/sec
[  5]   9.00-10.00  sec   112 MBytes   941 Mbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
Test Complete. Summary Results:
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-10.00  sec  1.10 GBytes   942 Mbits/sec    0             sender
[  5]   0.00-10.00  sec  1.10 GBytes   941 Mbits/sec                  receiver
snd_tcp_congestion cubic
rcv_tcp_congestion cubic
```

3. Test 3: TCP Bi-direction Send/Receive

```
iperf 3.9
Linux phxlab-pcie-L1SS 5.15.0-48-generic #54-Ubuntu SMP Fri Aug 26 13:26:29 UTC 2022
x86_64
Control connection MSS 1448
Time: Fri, 14 Oct 2022 20:42:04 GMT
Connecting to host 192.168.0.101, port 5201
      Cookie: j6fn2stwe34s3bxkb7fjbkmeg53yftjnv3ge
      TCP MSS: 1448 (default)
[  5] local 192.168.0.100 port 37228 connected to 192.168.0.101 port 5201
Starting Test: protocol: TCP, 1 streams, 131072 byte blocks, omitting 0 seconds, 10
second test, tos 0
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-1.00   sec   113 MBytes   947 Mbits/sec    0    236 KBytes
[  5]   1.00-2.00   sec   113 MBytes   944 Mbits/sec    0    236 KBytes
[  5]   2.00-3.00   sec   112 MBytes   941 Mbits/sec    0    247 KBytes
[  5]   3.00-4.00   sec   112 MBytes   941 Mbits/sec    0    247 KBytes
[  5]   4.00-5.00   sec   112 MBytes   941 Mbits/sec    0    260 KBytes
[  5]   5.00-6.00   sec   112 MBytes   942 Mbits/sec    0    274 KBytes
[  5]   6.00-7.00   sec   112 MBytes   944 Mbits/sec    0    321 KBytes
[  5]   7.00-8.00   sec   112 MBytes   943 Mbits/sec    0    321 KBytes
[  5]   8.00-9.00   sec   112 MBytes   938 Mbits/sec    0    321 KBytes
[  5]   9.00-10.00  sec   112 MBytes   942 Mbits/sec    0    321 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
Test Complete. Summary Results:
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-10.00  sec  1.10 GBytes   942 Mbits/sec    0             sender
[  5]   0.00-10.00  sec  1.10 GBytes   941 Mbits/sec                  receiver
CPU Utilization: local/sender 6.7% (0.4%u/6.4%s), remote/receiver 39.7% (5.5%u/34.2%s)
snd_tcp_congestion cubic
rcv_tcp_congestion cubic

iperf Done.
```

4. Test 4: UDP Send

```
iperf 3.9
Linux phxlab-pcie-L1SS 5.15.0-48-generic #54-Ubuntu SMP Fri Aug 26 13:26:29 UTC 2022
x86_64
Control connection MSS 1448
Setting UDP block size to 1448
Time: Fri, 14 Oct 2022 20:45:39 GMT
Connecting to host 192.168.0.101, port 5201
      Cookie: qtnqrhijzzyuz2uakscr63w75pss3tly7xnv
      Target Bitrate: 1000000000
[  5] local 192.168.0.100 port 57677 connected to 192.168.0.101 port 5201
Starting Test: protocol: UDP, 1 streams, 1448 byte blocks, omitting 0 seconds, 10
second test, tos 0
[ ID] Interval           Transfer     Bitrate         Total Datagrams
```

```
[  5]   0.00-1.00   sec   114 MBytes   956 Mbits/sec  82569
[  5]   1.00-2.00   sec   114 MBytes   956 Mbits/sec  82562
[  5]   2.00-3.00   sec   114 MBytes   956 Mbits/sec  82548
[  5]   3.00-4.00   sec   114 MBytes   957 Mbits/sec  82570
[  5]   4.00-5.00   sec   114 MBytes   956 Mbits/sec  82566
[  5]   5.00-6.00   sec   114 MBytes   956 Mbits/sec  82553
[  5]   6.00-7.00   sec   114 MBytes   956 Mbits/sec  82567
[  5]   7.00-8.00   sec   114 MBytes   956 Mbits/sec  82567
[  5]   8.00-9.00   sec   114 MBytes   956 Mbits/sec  82559
[  5]   9.00-10.00  sec   114 MBytes   956 Mbits/sec  82563
- - - - - - - - - - - - - - - - - - - - - - - - -
Test Complete. Summary Results:
[ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total Datagrams
[  5]   0.00-10.00  sec  1.11 GBytes   956 Mbits/sec  0.000 ms  0/825624 (0%)  sender
[  5]   0.00-10.00  sec  1.11 GBytes   956 Mbits/sec  0.020 ms  445/825606 (0.054%)
receiver
CPU Utilization: local/sender 35.9% (5.9%u/30.0%s), remote/receiver 32.5% (10.7%u/
21.8%s)

iperf Done.
```

5. Test 5: UDP Receive

```
iperf 3.9
Linux phxlab-pcie-L1SS 5.15.0-48-generic #54-Ubuntu SMP Fri Aug 26 13:26:29 UTC 2022
x86_64
Control connection MSS 1448
Setting UDP block size to 1448
Time: Fri, 14 Oct 2022 20:45:56 GMT
Connecting to host 192.168.0.101, port 5201
Reverse mode, remote host 192.168.0.101 is sending
      Cookie: oz4gyx36uvygdnu5p2fptdjcpva474rid5te
      Target Bitrate: 1000000000
[  5] local 192.168.0.100 port 48720 connected to 192.168.0.101 port 5201
Starting Test: protocol: UDP, 1 streams, 1448 byte blocks, omitting 0 seconds, 10
second test, tos 0
[ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total Datagrams
[  5]   0.00-1.00   sec   111 MBytes   934 Mbits/sec  0.015 ms  0/80664 (0%)
[  5]   1.00-2.00   sec   114 MBytes   956 Mbits/sec  0.011 ms  0/82563 (0%)
[  5]   2.00-3.00   sec   114 MBytes   956 Mbits/sec  0.017 ms  0/82563 (0%)
[  5]   3.00-4.00   sec   114 MBytes   956 Mbits/sec  0.017 ms  0/82562 (0%)
[  5]   4.00-5.00   sec   114 MBytes   956 Mbits/sec  0.018 ms  0/82559 (0%)
[  5]   5.00-6.00   sec   114 MBytes   956 Mbits/sec  0.021 ms  0/82565 (0%)
[  5]   6.00-7.00   sec   114 MBytes   956 Mbits/sec  0.015 ms  0/82563 (0%)
[  5]   7.00-8.00   sec   114 MBytes   956 Mbits/sec  0.022 ms  0/82565 (0%)
[  5]   8.00-9.00   sec   114 MBytes   956 Mbits/sec  0.019 ms  0/82563 (0%)
[  5]   9.00-10.00  sec   114 MBytes   956 Mbits/sec  0.017 ms  0/82560 (0%)
- - - - - - - - - - - - - - - - - - - - - - - - -
Test Complete. Summary Results:
[ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total Datagrams
[  5]   0.00-10.00  sec  1.11 GBytes   954 Mbits/sec  0.000 ms  0/823828 (0%)  sender
[  5]   0.00-10.00  sec  1.11 GBytes   954 Mbits/sec  0.017 ms  0/823727 (0%)
receiver

iperf Done.
```

6. Test 6: UDP Bi-direction Send/Receive

```
iperf 3.9
Linux phxlab-pcie-L1SS 5.15.0-48-generic #54-Ubuntu SMP Fri Aug 26 13:26:29 UTC 2022
x86_64
Control connection MSS 1448
Setting UDP block size to 1448
Time: Fri, 14 Oct 2022 20:46:13 GMT
Connecting to host 192.168.0.101, port 5201
      Cookie: to4fxqk6dd5ykzl3eqdtykotytjuzulsjetr
[  5] local 192.168.0.100 port 57327 connected to 192.168.0.101 port 5201
Starting Test: protocol: UDP, 1 streams, 1448 byte blocks, omitting 0 seconds, 10
second test, tos 0
[ ID] Interval           Transfer     Bitrate         Total Datagrams
[  5]   0.00-1.00   sec   114 MBytes   957 Mbits/sec  82620
[  5]   1.00-2.00   sec   114 MBytes   956 Mbits/sec  82560
```

```
[  5]   2.00-3.00   sec   114 MBytes   957 Mbits/sec  82580
[  5]   3.00-4.00   sec   114 MBytes   956 Mbits/sec  82550
[  5]   4.00-5.00   sec   114 MBytes   956 Mbits/sec  82560
[  5]   5.00-6.00   sec   114 MBytes   956 Mbits/sec  82560
[  5]   6.00-7.00   sec   114 MBytes   957 Mbits/sec  82570
[  5]   7.00-8.00   sec   114 MBytes   956 Mbits/sec  82560
[  5]   8.00-9.00   sec   114 MBytes   956 Mbits/sec  82570
[  5]   9.00-10.00  sec   114 MBytes   956 Mbits/sec  82550
- - - - - - - - - - - - - - - - - - - - - - - - -
Test Complete. Summary Results:
[ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total Datagrams
[  5]   0.00-10.00  sec  1.11 GBytes   956 Mbits/sec  0.000 ms  0/825680 (0%)  sender
[  5]   0.00-10.00  sec  1.11 GBytes   956 Mbits/sec  0.027 ms  0/825672 (0%)
receiver
CPU Utilization: local/sender 22.5% (2.7%u/19.8%s), remote/receiver 35.0% (12.2%u/
22.8%s)

iperf Done.
```

D.3 Interpreting Results and Performance Tuning

D.3.1 TCP PERFORMANCE

Different connections may need different TCP settings to optimize performance. Collecting iPerf traces using default settings, as previously described, is a good starting point, but additional consideration may be required to find the optimized results.

D.3.1.1 Performance Tuning via TCP Window Size

The TCP window size is the most critical parameter. TCP window size is an advertisement of how much data (in bytes) the receiving device is able to receive. The receiving device should adjust its TCP window size to control the flow of data.

The TCP Window size is set through the –w argument, with the selected value to follow. If the TCP window size is too small, the sender will become idle and performance is not maximized. If the TCP window size is too large, the receiver will not be able to actually receive the data the sender is trying to transmit and will have to issue a TCP-zero window size to stop the flow of traffic while the filled buffers are processed. The start-stop nature causes inefficiencies and is overall slower than a just-right TCP window size.

The starting point for TCP window size is called the bandwidth*delay product, which is simply the bottleneck bandwidth multiplied by the roundtrip time (ping).

For a LAN7800 connection, the bottleneck should be determined first. Refer to Table 5 to determine the bottleneck.

TABLE 5: LAN7800 CONNECTION BOTTLENECKS

USB Connection Speed	Ethernet Connection Speed	Bottleneck (real performance considering overhead)
USB2.0 Full Speed (12 Mbit/s)	10 Mbit/s	~9.5 Mbit/s (See **Note 1**.)
	100 Mbit/s	~10 Mbit/s (See **Note 1**.)
	1000 Mbit/s	~10 Mbit/s (See **Note 1**.)
USB2.0 High Speed (480 Mbit/s)	10 Mbit/s	~9.5 Mbit/s (See **Note 1**.)
	100 Mbit/s	~95 Mbit/s (See **Note 1**.)
	1000 Mbit/s	~350 Mbit/s (See **Note 1**.)
USB3.0 Super Speed (5 Gbit/s)	10 Mbit/s	~9.5 Mbit/s (See **Note 1**.)
	100 Mbit/s	~95 Mbit/s (See **Note 1**.)
	1000 Mbit/s	~950 Mbit/s (See **Note 1**.)

Note 1: These values only describe the bottleneck of the protocols used. The network performance should also be considered in case the network performance is slower than the physical interface.

EXAMPLE 1: USB3 HOST CONNECTION, 1000 MBIT/S ETHERNET CONNECTION, 10 MS PING

bandwidth*delay

= 950 Mbit/sec * 10 ms

= 9,500,000 bits

= 1.1875 MB

EXAMPLE 2: USB2 HOST CONNECTION, 1000 MBIT/S ETHERNET CONNECTION, 2 MS PING

bandwidth*delay

= 35 Mbit/sec * 2 ms

= 70,000 bits

= 8.75 kB

These calculations can serve as the starting point for experiments.

Experiment #1: Sweep the TCP window higher and lower to chart out performance trends and find the optimal window size for the specific connection under test.

Note:	Operating systems have varying limits to TCP window sizes. iPerf will usually generate a warning when the requested window size does not align with system capabilities.

Experiment #2: Run multiple parallel TCP streams. If TCP settings are correct and the system is working normally, then the throughput sum of multiple parallel TCP streams should be equivalent to a single stream test. If not, then the TCP window size may not be correct (or there may be some other system or network issue outside of the influence of iPerf settings.)

D.3.1.2 Performance Tuning via Maximum Transmission Unit (MTU)

Maximum transmission unit (MTU) represents the maximum data packet size that a device can accept. MTU is measured in Bytes and has a maximum setting of 1,500. It is a setting of the network interface and is configured by the driver but can be modified through driver settings. Most devices connected to a typical modern Ethernet connection will support what is called Path MTU Discovery (PMTUD). Through this, the MTU of all devices, switches, and routers within a network can be discovered.

Data payloads which would create packets that exceed MTU are broken up into smaller, separate packets so that the MTU of any component on the network path is not violated. This is called fragmentation. The fragmented packets are reassembled upon receipt at the final packet destination.

In iPerf, you can use the –m argument to display what the MTU setting is utilized for the connection. You can compare this value with the MTU settings on the computers under test to see if there is a mismatch between what is expected and what is being used. If you observe a setting that is below what is expected and what is being indicated in the network device settings, you can examine components within the network and try different computers to isolate the source of the issue.

Note:	If the –m argument shows a value of 536, this may indicate that PMTUD is not supported on one of the computers.

The –M argument can also be used to manually modify the TCP maximum segment size. Keep in mind that the MTU should not be exceeded, and an IP header will include 40 Bytes, so –M should not exceed the MTU –40.

Note:	**Jumbo Frames:** Jumbo frames can be used on Gigabit Ethernet local area networks. Jumbo frames are not defined in the IEEE 802.3 specifications for Ethernet, so support and maximum size (up to 9,000 Bytes) varies by vendor.

D.3.2 UDP PERFORMANCE

User Datagram Protocol (UDP) is a communication protocol used to achieve low-latency data transfer. UDP protocol sends messages called datagrams. A UDP message can be split up into several IP packets. If any of the IP packets within a message is lost or corrupted, the entire message is lost. UDP protocol is considered best-effort. It does not have retry mechanisms for lost or corrupted messages, so a certain level loss is expected. As such, applications must be tolerant to loss.

iPerf creates a constant bit rate UDP stream which simulates a basic use-case-like voice communication. Since TCP does not report a packet loss to the user, UDP tests can be helpful for determining the real amount of packets lost in a connection.

D.3.2.1 Performance Tuning via Datagram Size

To make a measurement representative of the use case, the datagram size should be adjusted to match the datagram size implemented in the end application.

If it is desired to use UDP to measure packet loss instead of datagram loss, then the UDP message must be forced to be small enough to fit within a single packet. This is achieved using the –l command. The default size for UDP messages is 1470 bytes, which is sufficient for Ethernet communication since a UDP header is 8 bytes, and a maximum Ethernet Frame size is 1518 bytes.

D.3.2.2 Jitter

The server makes continuous jitter calculations. The client records a timestamp in every packet. The server tracks the transmit time by calculating the receive time minus the send time, and the jitter is reported by tracking the variation in time stamp differences. No clock synchronization is done in this process like what is done in 1588 PTP applications.

APPENDIX E: HOW TO DUMP LAN780X EEPROM/OTP CONFIGURATION DATA AND VERIFY OUTPUT FILE

E.1 Step 1: Dump EEPROM/OTP Raw Binary Data

E.1.1 WINDOWS

Windows users can use one of two utilities available on Windows:

1. MPLAB Connect Configurator:
   - Includes a GUI-based utility for creating configuration binaries which can be programmed to EEPROM/OTP and a command line utility for production line programming.
   - **Link:** https://www.microchip.com/en-us/tools-resources/configure/mplab-connect-configurator
2. 7800WinCmd.exe:
   - A command-line only utility with various status and control functions, including the ability to program/read EEPROM/OTP memory of a LAN780x device.
   - **Link:** Visit the LAN7800 or LAN7801 product page and click the link titled “LAN780x Windows Command Line Utilities”.

E.1.1.1 MPLAB Connect

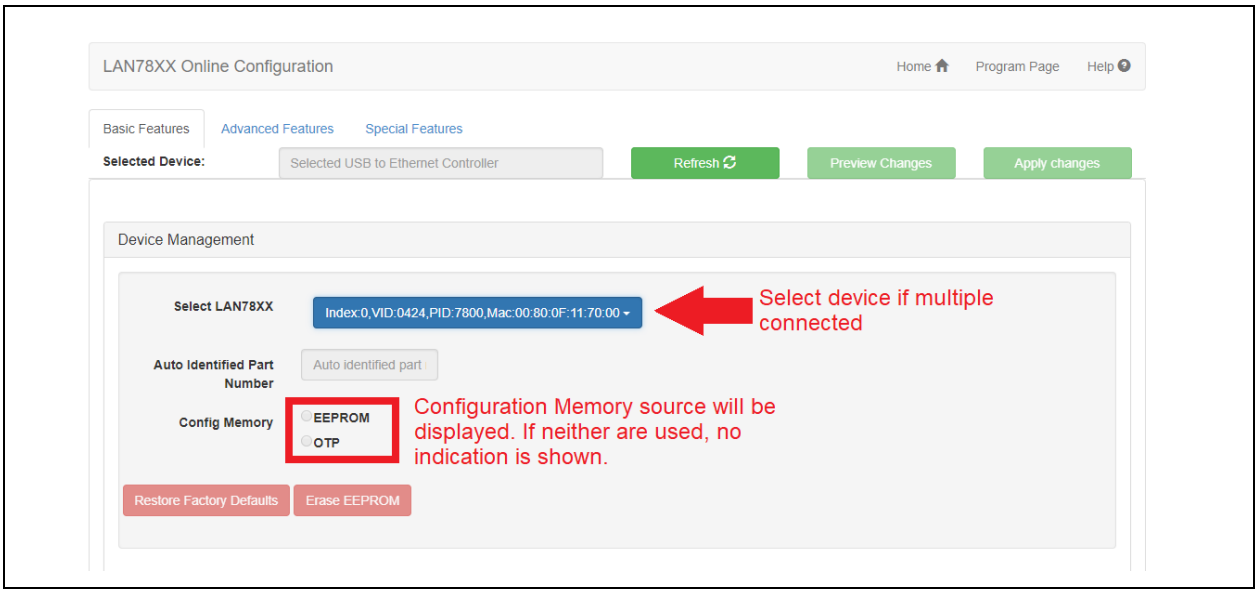
1. Download the latest version: https://www.microchip.com/en-us/tools-resources/configure/mplab-connect-configurator
2. Connect LAN7800 to your system.
3. Navigate to the GUI application and open the application.
4. Find the “USB Ethernet Bridge Controllers” and select the ‘Online’ radio button on the LAN780x Ethernet Bridging devices row. See Figure 20.

FIGURE 20: OPENING THE LAN7800 IN THE MPLAB CONNECT GUI

5. The tool will now attempt to communicate with the device. If multiple LAN780x devices are connected, you may manually select which one to communicate within the Device Management dropdown menu. (See Figure 21.) The source of configuration for the device (EEPROM or OTP) will also be shown here.

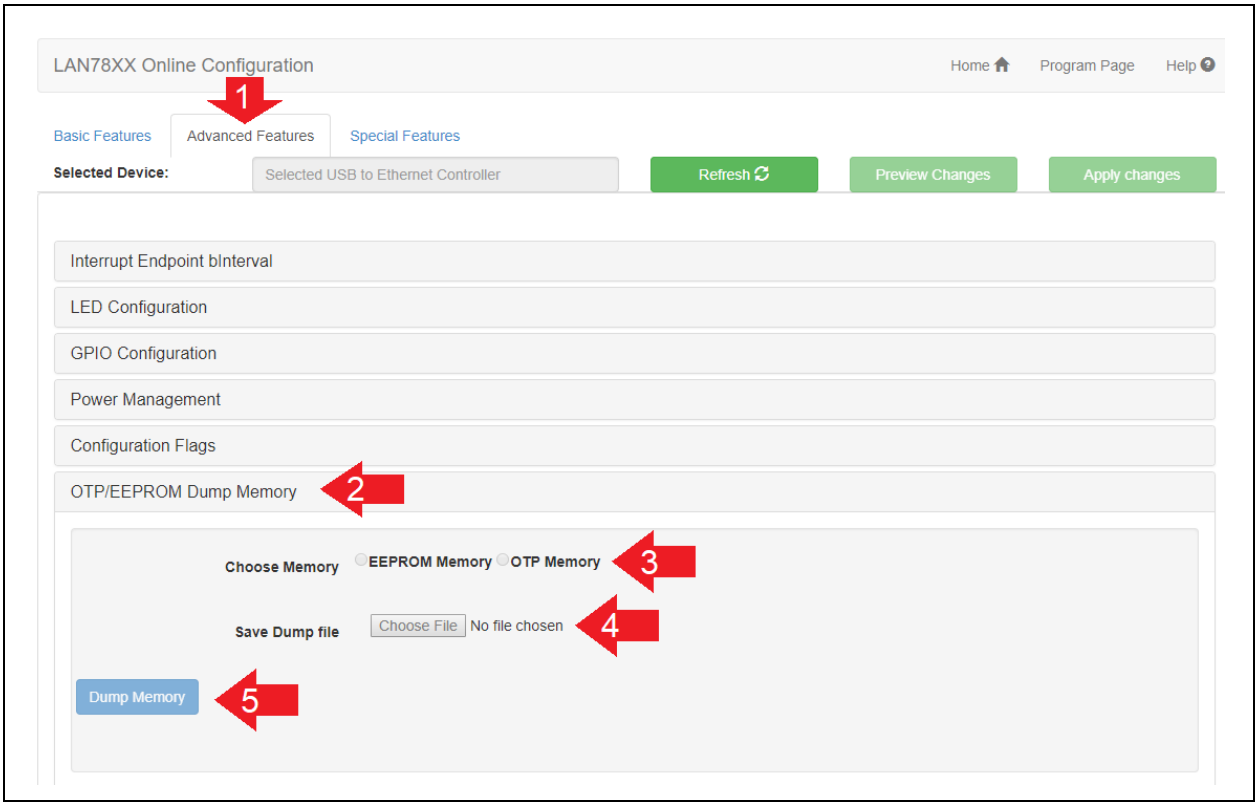
Note: If the host PC suspends the LAN780x device, then the tool will not be able to access it. Typically, if the LAN780x is not being actively used, a modern host PC will place it into USB Suspend to save power. Disabling USB Suspend function on the USB host prevents this issue. Different versions of Windows have different ways to navigate to this configuration option. Users are advised to do an Internet search for their specific OS version to obtain instructions.

FIGURE 21: DEVICE MANAGEMENT TAB IN MPLAB® CONNECT CONFIGURATOR



6. Select the **Advanced Features** tab at the top of the tool (1), open the 'EEPROM/OTP Dump Memory' dropdown menu (2), select either EEPROM or OTP memory target (3), select a destination file path and filename (4), and click on the **Dump Memory** button (5). The tool will save the raw binary data to an external file which can be viewed using a binary/hex editing tool. See Figure 22.

FIGURE 22: DUMPING AN EEPROM IMAGE IN MPLAB CONNECT



E.1.1.2 7800WinCmd.exe

1. Download the latest version and unzip the package to a directory of your choice.
2. Open a command prompt terminal on Windows (See Figure 23.) and navigate to the directory where 7800WinCmd was unzipped:

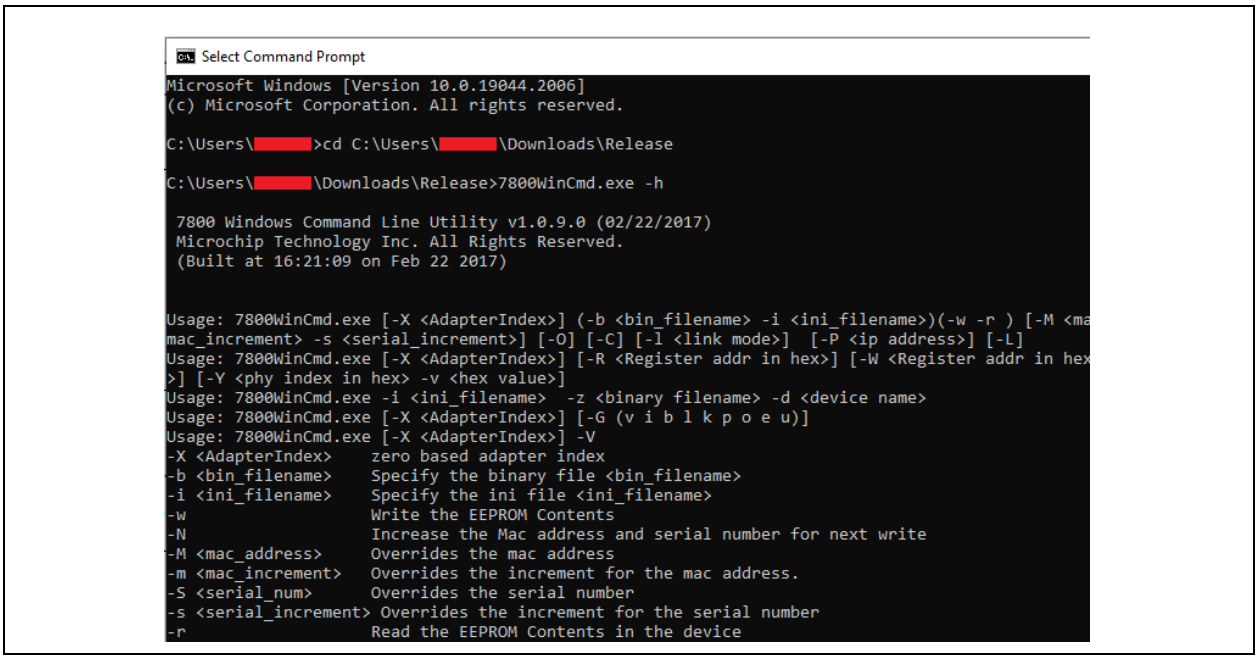
```
cd <filepath here>
```

FIGURE 23: CHANGING DIRECTORIES IN WINDOWS® COMMAND PROMPT

3. Issue the following command to get the help prompt and ensure the application is functional. (See Figure 24.):

```
7800WinCmd.exe -h
```

FIGURE 24: 7800WINCMD.EXE -HELP COMMAND



4. Issue the following command to dump the EEPROM contents of the connected LAN780x device (See Figure 25.).

```
7800WinCmd.exe -r
```

The file will be saved in raw binary format to the working directory.

Also, issue the following command to dump the OTP contents of the connected LAN780x device (See Figure 26.):

```
7800WinCmd.exe -O -r
```

FIGURE 25: 7800WINCMD.EXE DUMP EEPROM MEMORY

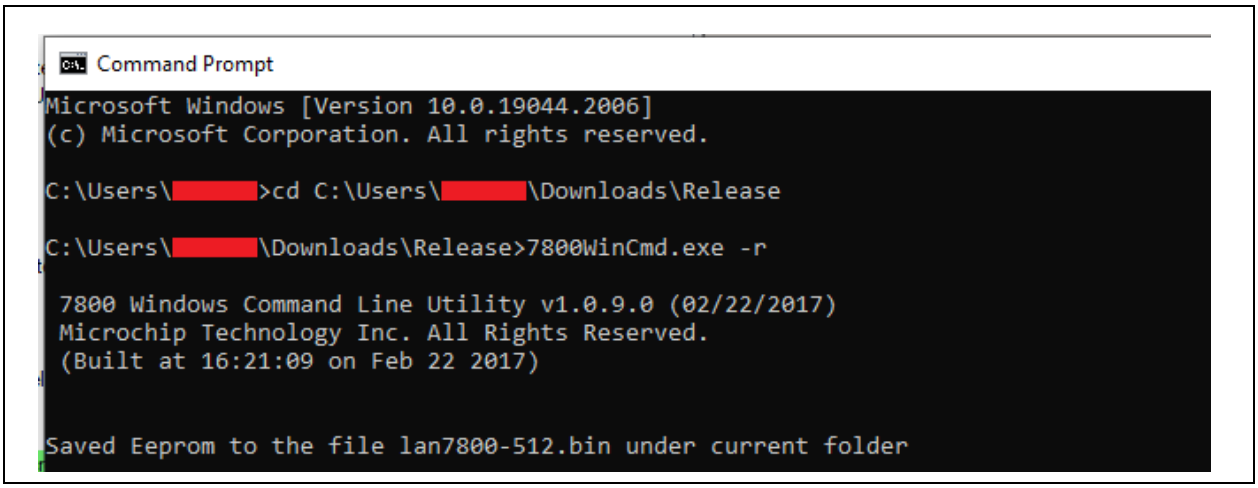
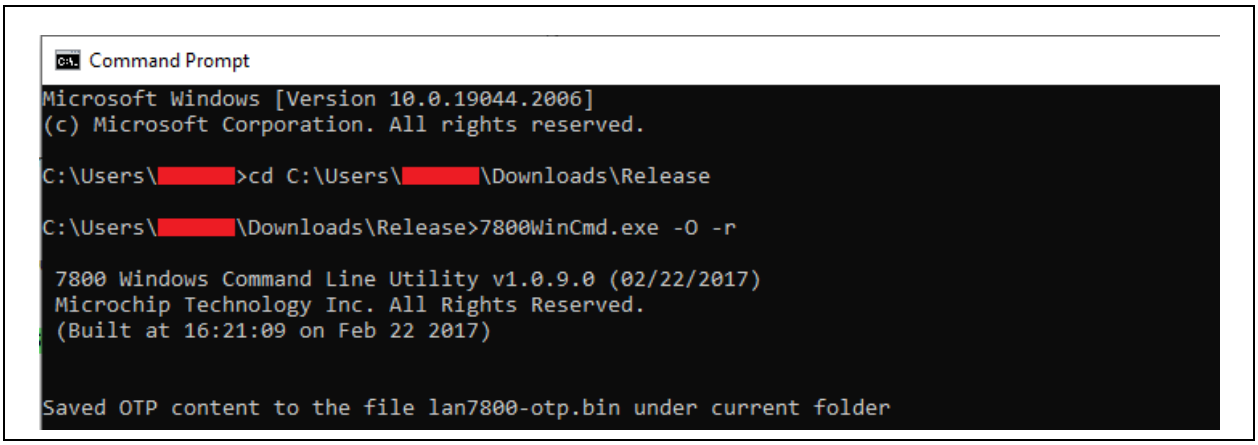


FIGURE 26: 7800WINCMD.EXE DUMP OTP MEMORY



The file will be saved in raw binary format to the working directory.

Note:	If the host PC suspends the LAN780x device, then the tool will not be able to access it. Typically, if the LAN780x is not being actively used, a modern host PC will place it into USB Suspend to save power. Disabling USB Suspend function on the USB host will prevent this issue. Different versions of Windows have different ways to navigate to this configuration option, so do an Internet search for your specific OS version to obtain instructions.

E.1.2 LINUX

The ethtool utility is included in most Linux distributions. If it is not installed, it can be installed using the distribution-specific command for installing programs (i.e.: Ubuntu uses ‘sudo apt-get install ethtool’).

To dump the EEPROM, a single command to the command prompt is needed. The DEVNAME of the LAN780x must be known. It can be obtained by listing all network interfaces with ‘ifconfig -a’ and finding the name assigned to the LAN7800.

```
ethtool <DEVNAME> -e
```

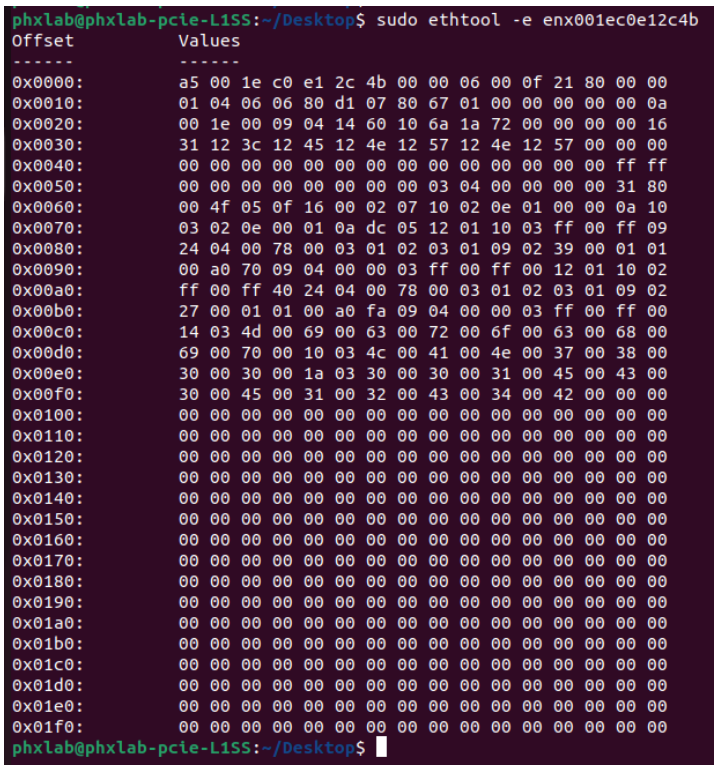
See Figure 27 for an example of the output of this command.

To dump the OTP, OTP access must first be enabled through the following command:

```
ethtool --set-priv-flags <DEVNAME> OTP_ACCESS on
```

Then, use the same command previously mentioned for EEPROM access to dump the OTP memory (`ethtool <DEVNAME> -e`).

FIGURE 27: 7800WINCMD.EXE DUMP OTP MEMORY



E.1.3 MACOS

There is currently no utility for EEPROM/OTP memory read-back on macOS.

E.2 Step 2: Open Configuration File and Verify Contents

The EEPROM and OTP dump files should be quickly examined to verify if the contents are valid. Download a hex editor tool for Windows if one is not already installed.

E.2.1 EEPROM FILE

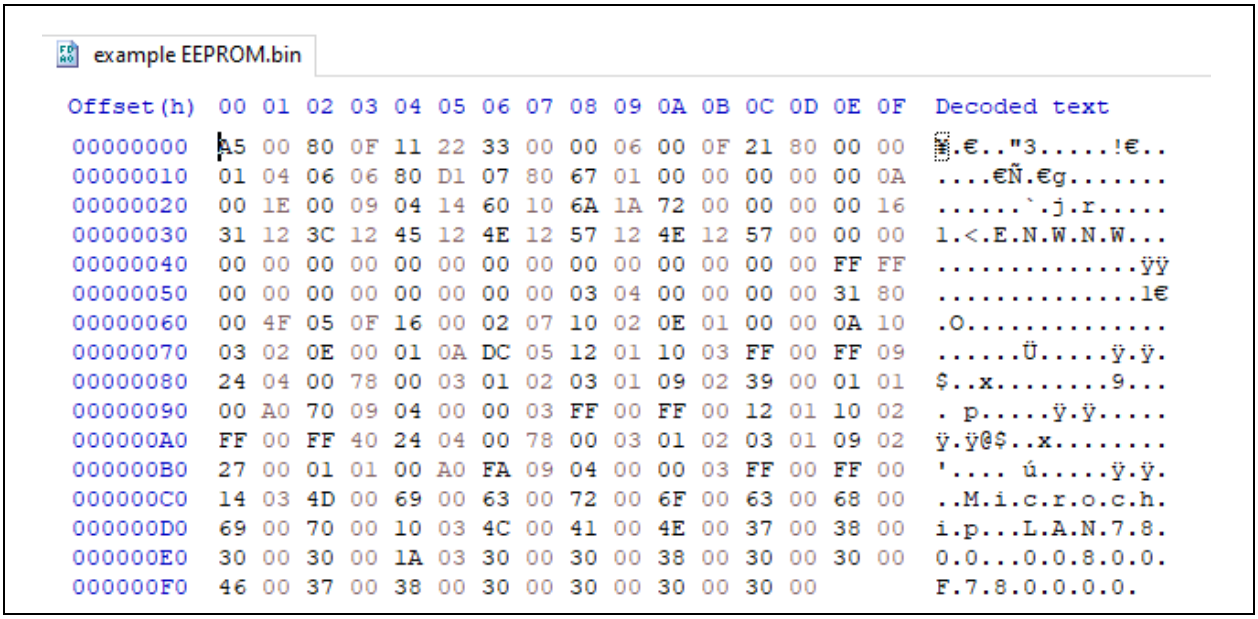
If the EEPROM file is blank, all bytes will be 0x00 or 0xFF throughout the entire file (depending on the default content of the selected EEPROM device).

In a normal system if EEPROM is used as the primary configuration method, a MAC address will be programmed into the EEPROM. Additional optional configuration data may be programmed as well.

Open the file using the hex editor tool and look specifically for the following byte offsets:

- Byte 0: This is referred to as the EEPROM Magic Byte. The value of the Magic Byte in the EEPROM programming case is always '0xA5'. (See Figure 28.)
- Bytes 1-6: Contain the MAC Address.
- Bytes 7+: Refer to LAN7800 documentation for details.

FIGURE 28: EXAMPLE LAN7800 EEPROM CONFIGURATION FILE



E.2.2 OTP FILE

If the OTP file is blank, all bytes will be 0x00 throughout the entire file. In a normal system, if OTP is used as the primary configuration method, a MAC address will be programmed into the OTP. Additional optional configuration data may be programmed as well.

- Byte 0: This is referred to as the OTP "Magic Byte". The value of the Magic Byte can be different depending on the desired use case:

"0xF3": Program OTP to first bank of OTP Data

"0xF7 ": Program OTP to second bank of OTP Data

- Bytes 1-6: Contain the MAC Address.
- Bytes 7+: Refer to LAN7800 documentation for details.

Figure 29 shows a sample EEPROM Configuration file with the proper "Magic Byte" and a dummy MAC address of 00:80:0F:44:55:66 programmed.

FIGURE 29: EXAMPLE LAN7800 OTP CONFIGURATION FILE

example OTP.bin

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	F3	00	80	0F	44	55	66	00	00	06	00	0F	21	80	00	00	ó.€.DUf.....!€..
00000010	01	04	06	06	80	D1	07	80	67	01	00	00	00	00	00	0A	....€Ñ.€g.......
00000020	00	1E	00	09	04	14	60	10	6A	1A	72	00	00	00	00	16	......`.j.r.....
00000030	31	12	3C	12	45	12	4E	12	57	12	4E	12	57	00	00	00	1.<.E.N.W.N.W...
00000040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	FF	FF	..............ÿÿ
00000050	00	00	00	00	00	00	00	00	03	04	00	00	00	00	31	80	..............1€
00000060	00	4F	05	0F	16	00	02	07	10	02	0E	01	00	00	0A	10	.O..............
00000070	03	02	0E	00	01	0A	DC	05	12	01	10	03	FF	00	FF	09	......Ü.....ÿ.ÿ.
00000080	24	04	00	78	00	03	01	02	03	01	09	02	39	00	01	01	$..x........9...
00000090	00	A0	70	09	04	00	00	03	FF	00	FF	00	12	01	10	02	. p.....ÿ.ÿ.....
000000A0	FF	00	FF	40	24	04	00	78	00	03	01	02	03	01	09	02	ÿ.ÿ@$..x........
000000B0	27	00	01	01	00	A0	FA	09	04	00	00	03	FF	00	FF	00	'.... ú.....ÿ.ÿ.
000000C0	14	03	4D	00	69	00	63	00	72	00	6F	00	63	00	68	00	..M.i.c.r.o.c.h.
000000D0	69	00	70	00	10	03	4C	00	41	00	4E	00	37	00	38	00	i.p...L.A.N.7.8.
000000E0	30	00	30	00	1A	03	30	00	30	00	38	00	30	00	30	00	0.0...0.0.8.0.0.
000000F0	46	00	37	00	38	00	30	00	30	00	30	00	30	00			F.7.8.0.0.0.0.

APPENDIX F: HOW TO COLLECT A USB PROTOCOL ANALYZER TRACE WITH LAN780X

Note: The screenshots collected for this document represent the tools as a particular snapshot in time. These tools are continuously updated and GUIs may change over time. Consult the latest documentation for the protocol analyzer software being used if these images are no longer accurate or relevant at the time of reading this document.

A USB Protocol analyzer is a valuable tool for end-system integrators working with devices that utilize USB communication. USB protocol traces are useful to debugging USB issues such as connection issues, connection-loss issues, low performance, or interoperability problems. Generally speaking, the solutions provided by the major equipment vendors have comparable core features. The level of accuracy and detail as well of analyzer software feature set will usually scale along with the price (in other words, higher cost solutions generally offer better accuracy and software tools).

The examples in this document are collected using the two most popular options: (1) USB Protocol Suite from Teledyne LeCroy and (2) Data Center from Total Phase.

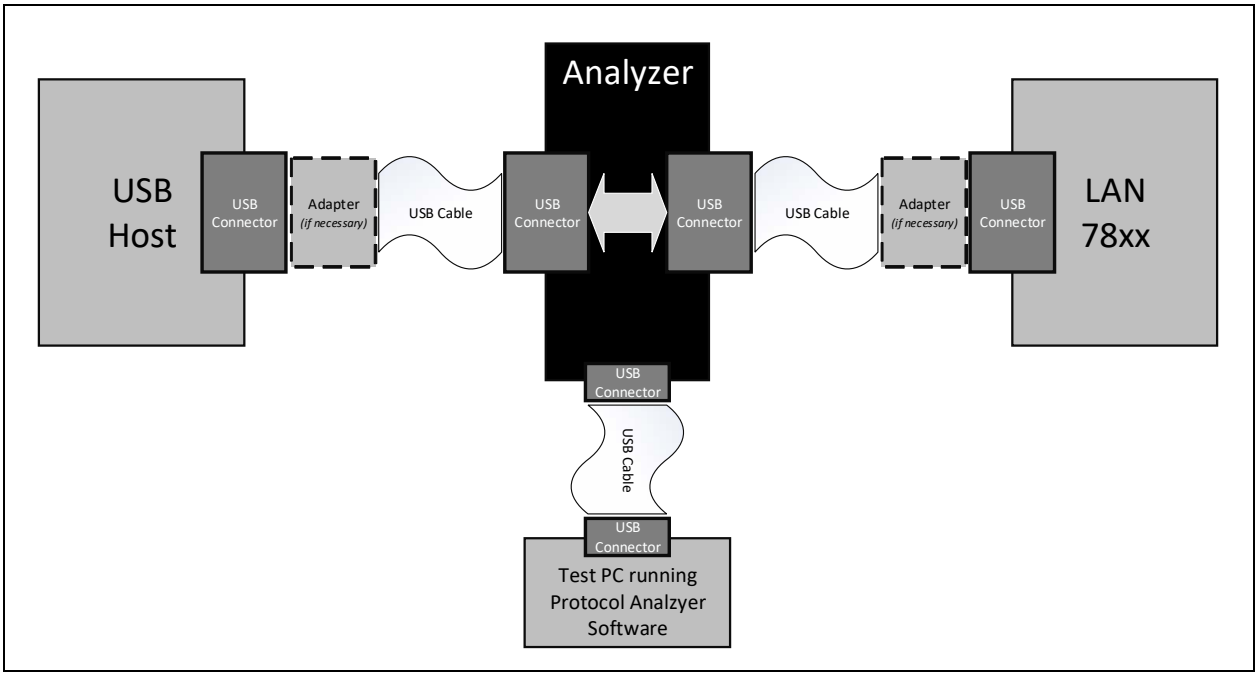
F.1 Hardware Setup

A USB protocol analyzer is a tool which must be inserted in between the USB device and the host port, so it is generally practical to use with cabled applications.

A separate USB host executing the USB protocol analyzer software is required. Refer to Figure 30.

Note: It is possible to use the same USB host that connects to the USB device in analysis to run the USB protocol analyzer software. However, this is generally not recommended as it may change the behavior that needs to be observed.

FIGURE 30: USB PROTOCOL ANALYZER



Note: It is possible to insert an analyzer in an embedded case through precision PCB rework by cutting PCB traces and soldering striped wire leads of a USB cable (which is cut in half) to the exposed traces. This type of rework requires precision PCB rework and soldering skills to achieve and may result in irreversible damage to the PCB. One method to enable this type of capability on the design is to implement zero Ohm resistors directly in series on USB traces which can be removed when a USB analyzer is required.

F.2 Analyzer Software Setup

Different USB protocol analysis solutions are available from major test equipment vendors, including Teledyne LeCroy, Total Phase, and Ellisys. The software for each is unique, so documentation for the specific analyzer used should be consulted.

Software is generally available from the test equipment vendor websites free of charge. They may require you to sign up for an account to access the download link. An analyzer is not required just to run the software and open or examine a previously captured trace file.

F.2.1 BUFFER SIZE

Buffers must be configured in most analyzers to set the size of the protocol trace capture. (See Figure 31 and Figure 32.) This is an important consideration as some analyzers leave very small default buffer sizes which make it difficult to capture the problem at hand. Generally speaking, the analyzer should be configured to maximize the buffers as long as the computer used for file analysis has enough computing power to work with a very large file. This can make navigating the trace file difficult due to the sheer size of the file, but this can be overcome by efficiently using the search functions in the analyzer software. Also, a protocol trace file can be trimmed down later to remove unnecessary data to make saving and transmission of the file easy to observe.

FIGURE 31: EXAMPLE – ADJUSTING ANALYZER BUFFER SIZE IN LECROY ANALYZER SOFTWARE

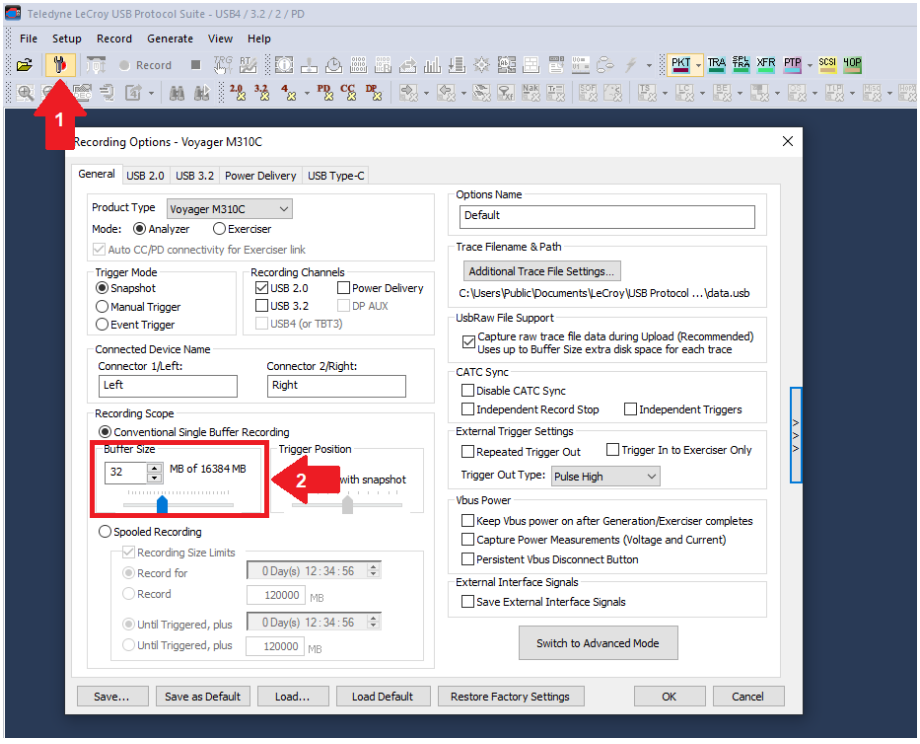
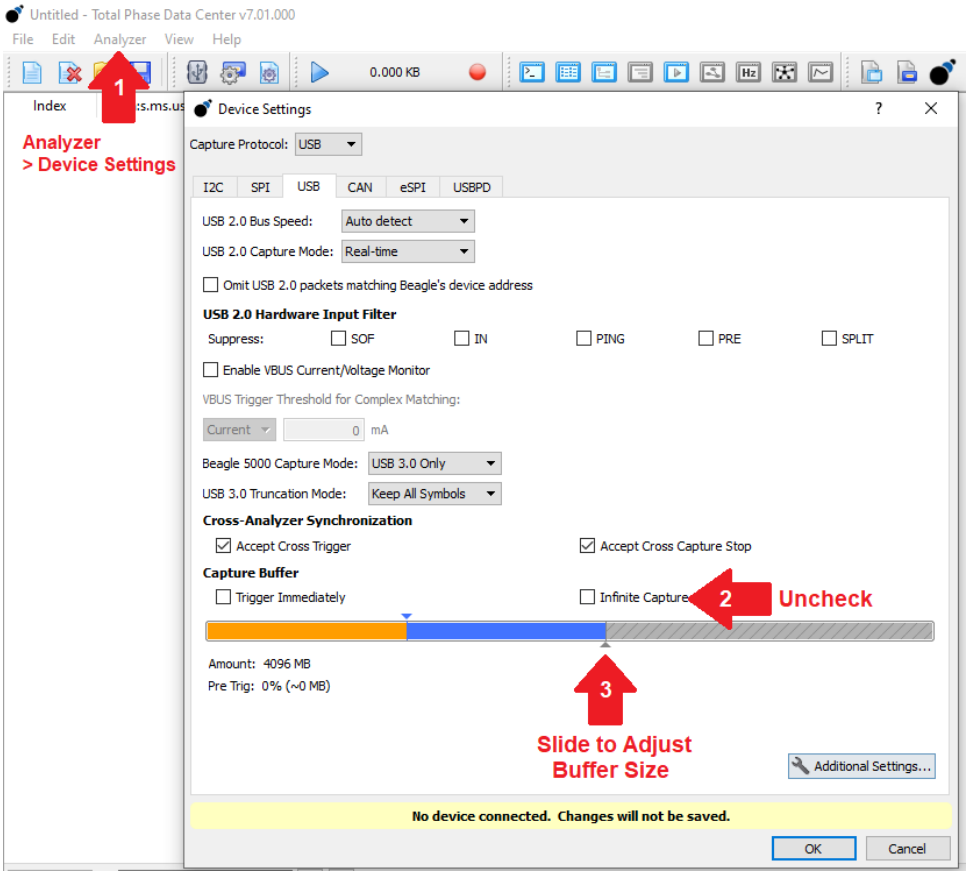


FIGURE 32: EXAMPLE – ADJUSTING ANALYZER BUFFER SIZE IN TOTAL PHASE ANALYZER SOFTWARE



F.2.2 TRIGGER OPTIONS

Generally, USB protocol traces can be captured using one of three mechanisms (The names given to these mechanisms may vary from one solution to another. Refer to Figure 33 and Figure 34.):

- **Software Activated:** USB protocol trace is initiated by clicking a button in the software analyzer tool. The protocol trace will begin capture immediately until the buffer is full.
- **Hardware GPIO Activated:** If the issue which must be observed on the protocol analyzer can be triggered by some physical hardware action, then a trigger connected to a GPIO on the analyzer can be used to start the collection of the protocol trace.
- **Event Triggered:** In this mode, the analyzer is constantly decoding the USB traffic, waiting for an event to trigger protocol capture. This could include triggering on a particular data payload (or portion of a data payload), some kind of error event, some kind of bus event (USB suspend, wake-up, etc.), or virtually any other event that can be observed in a protocol trace.

FIGURE 33: EXAMPLE – SELECTING CAPTURE MODE IN LECROY ANALYZER SOFTWARE

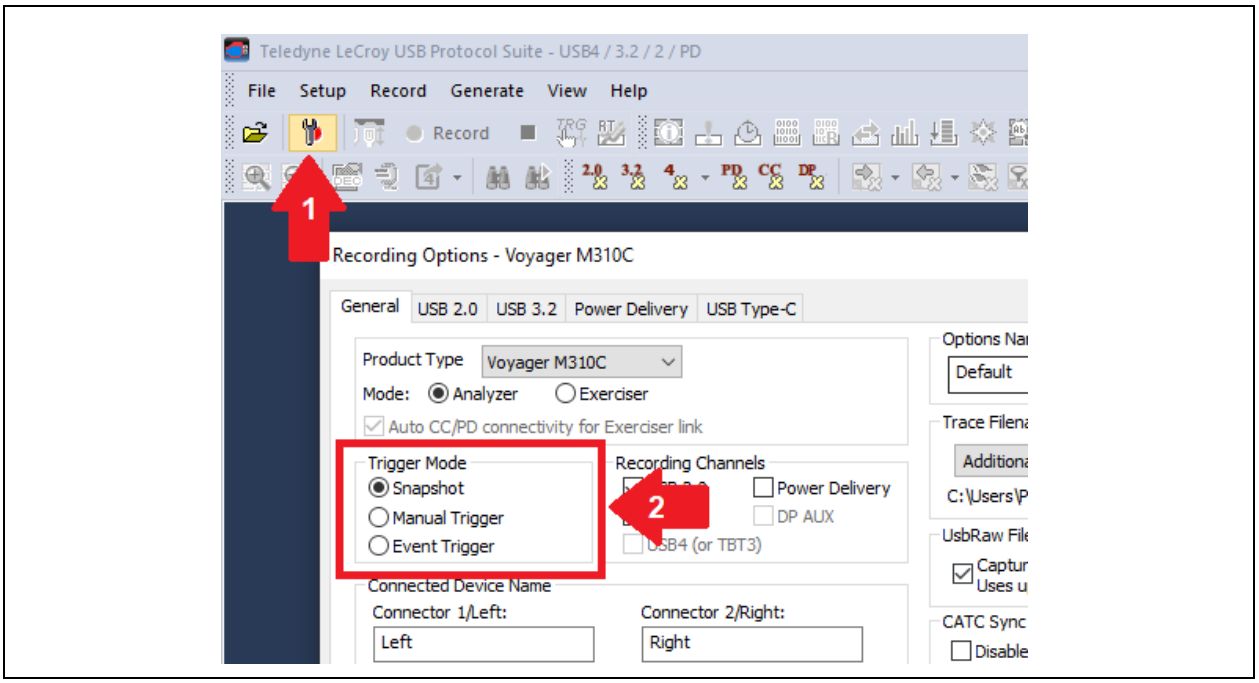
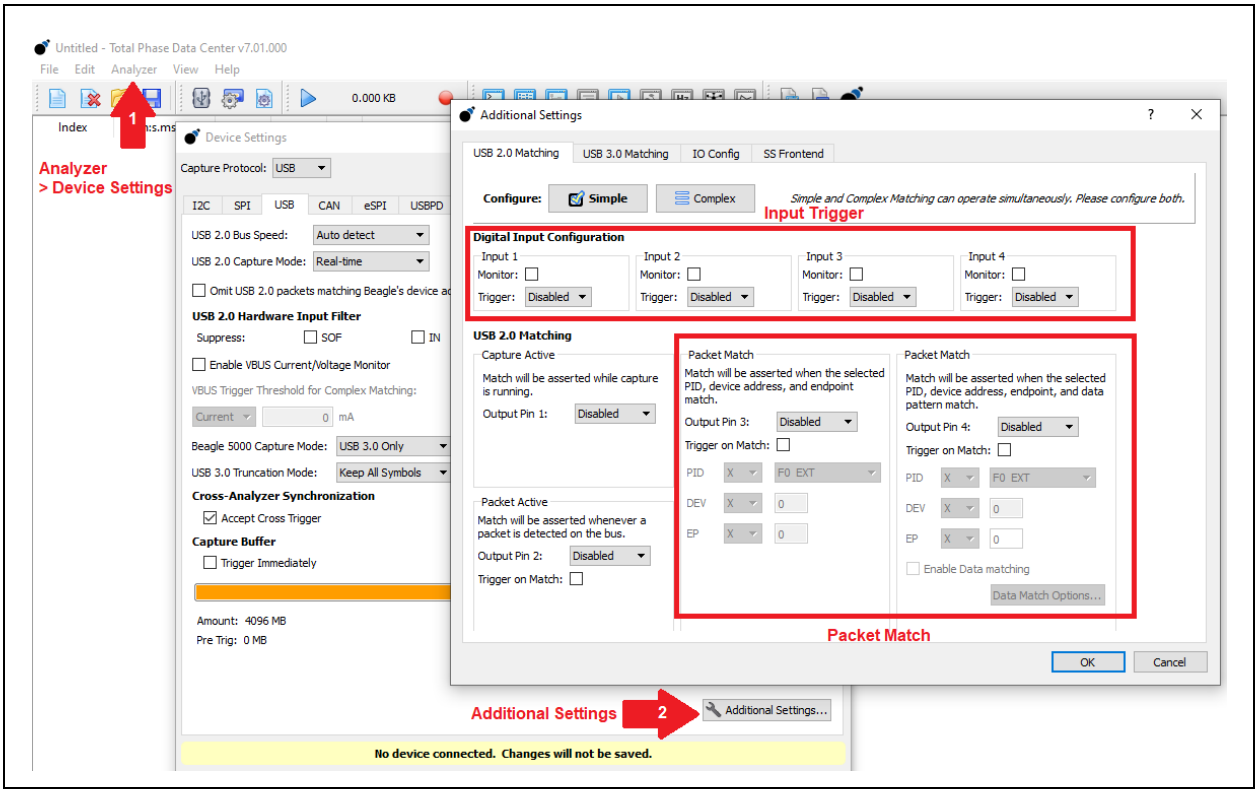


FIGURE 34: EXAMPLE – SELECTING CAPTURE MODE IN TOTAL PHASE ANALYZER SOFTWARE



F.2.3 SPEED SELECTION

Ensure that the protocol analyzer software is enabling the speeds that need to be captured. Most protocol analyzer software will have a speed auto-recognition capability, but this may need to be enabled before it works. Refer to Figure 35 and Figure 36.

FIGURE 35: EXAMPLE – SELECTING CAPTURE SPEED(S) IN LECROY ANALYZER SOFTWARE

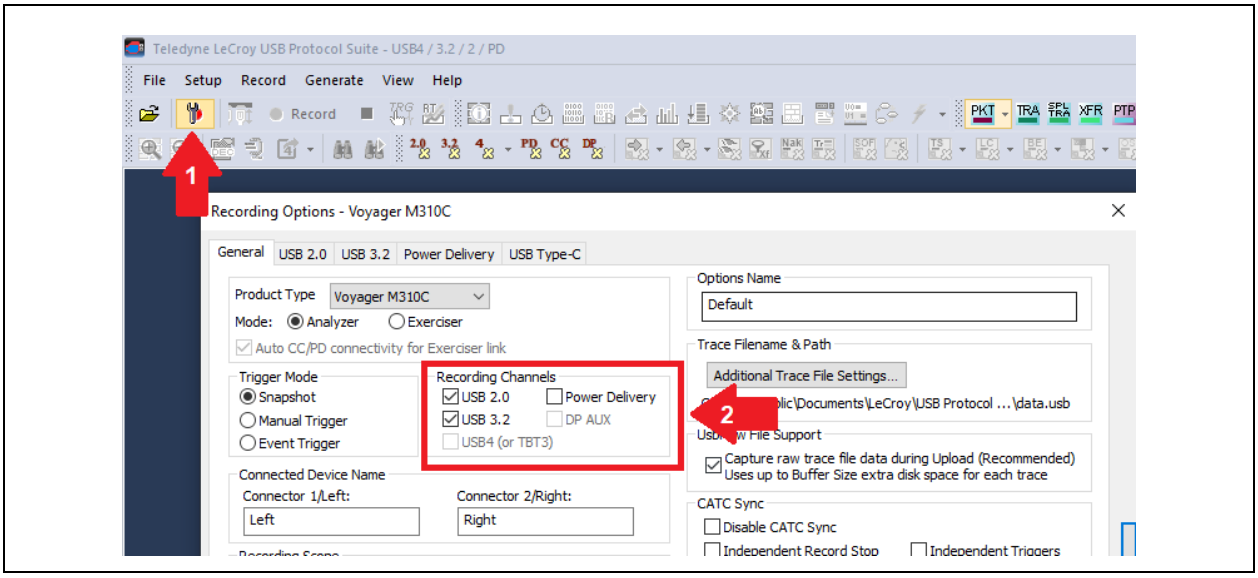
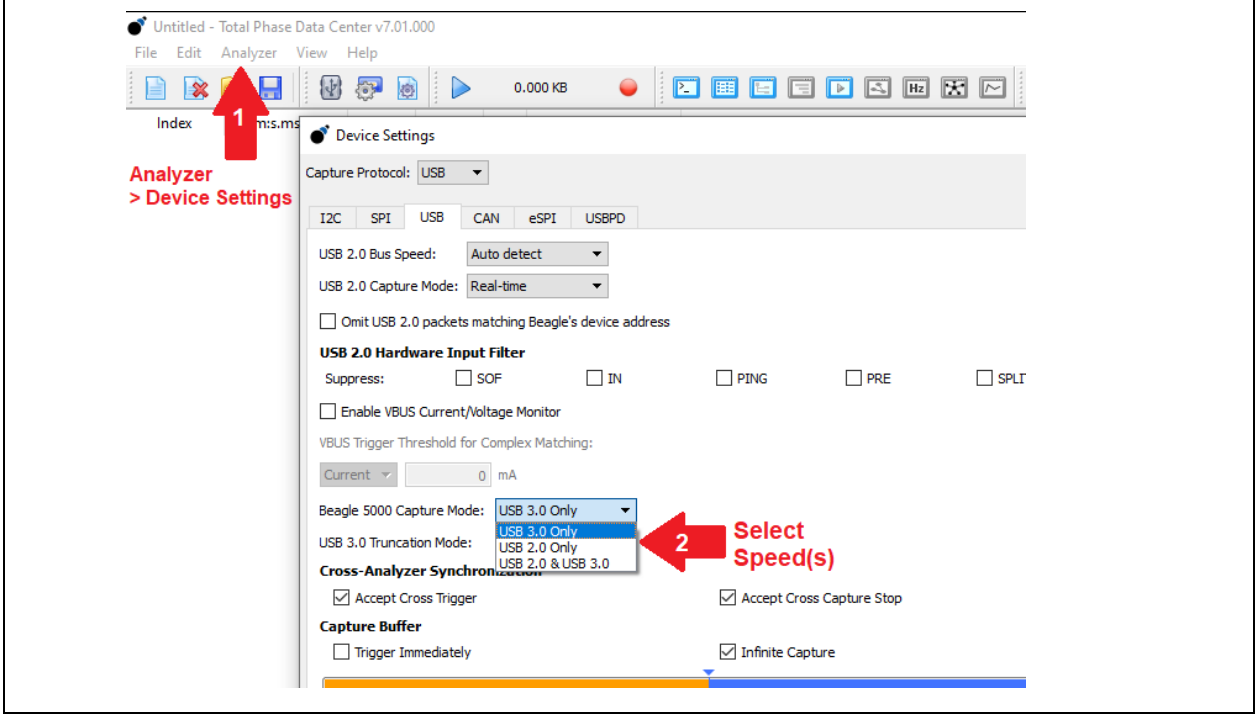


FIGURE 36: EXAMPLE – SELECTING CAPTURE SPEED(S) IN TOTAL PHASE ANALYZER SOFTWARE



F.3 Trace Collection

It is always the best practice to include the full USB tree enumeration at the beginning of a USB protocol trace by following the these steps:

1. Connect the USB Analyzer hardware to the system.
2. Setup protocol trace options, such as buffer size, trigger mechanism, and speed.
3. Enable the protocol trace in the analyzer software.
4. Power on the USB system or plug in the USB cable to the host port.
5. Allow the system to run until the USB buffers are filled.

This sequence is useful because the USB Protocol analyzer can use the device tree enumeration data to discover information about the devices. This information allows the analyzer to apply device-class specific decoding to the traffic, making the trace much more human-readable.

However, if the issue you are trying to debug is occurring long after enumeration occurs and/or the triggering mechanism is GPIO-based or event-based, it is not possible to include the full device tree enumeration as the amount of memory buffer in the analyzer will not be able to capture full USB tree enumeration and all traffic leading up to the trigger event. In this case, the trace will be very difficult to read as all traffic will be presented in raw form without any context. This is because the analyzer does not know anything about the devices in the trace since that information can only be obtained while devices communicate their descriptors during enumeration. The simplest solution to this problem is to collect two separate USB traces:

1. One trace at time of power on or connection, solely to capture the enumeration data.
2. One trace which captures the problem occurrence.

Using the enumeration trace, it is possible to assign the device-class decoding manually to the trace which captures the issue at hand, as long as all USB device address assignments are the same in both traces. It is important to collect the two traces at the same time, since USB address assignment from hosts can be at random and will differ from one connection to another.

F.4 Trace File Verification

Even if users are unfamiliar with the protocol analyzer software and plan to send it to others for review, it is highly recommended to do some minimal checks to ensure that the trace file contains usable information. The following should be done as a minimal verification:

1. Open the trace file.
2. Scroll through the file and verify that some form of USB traffic is visible. See Figure 37 and Figure 38.

FIGURE 37: EXAMPLE – VIEWING BASIC TRAFFIC IN LECROY ANALYZER SOFTWARE

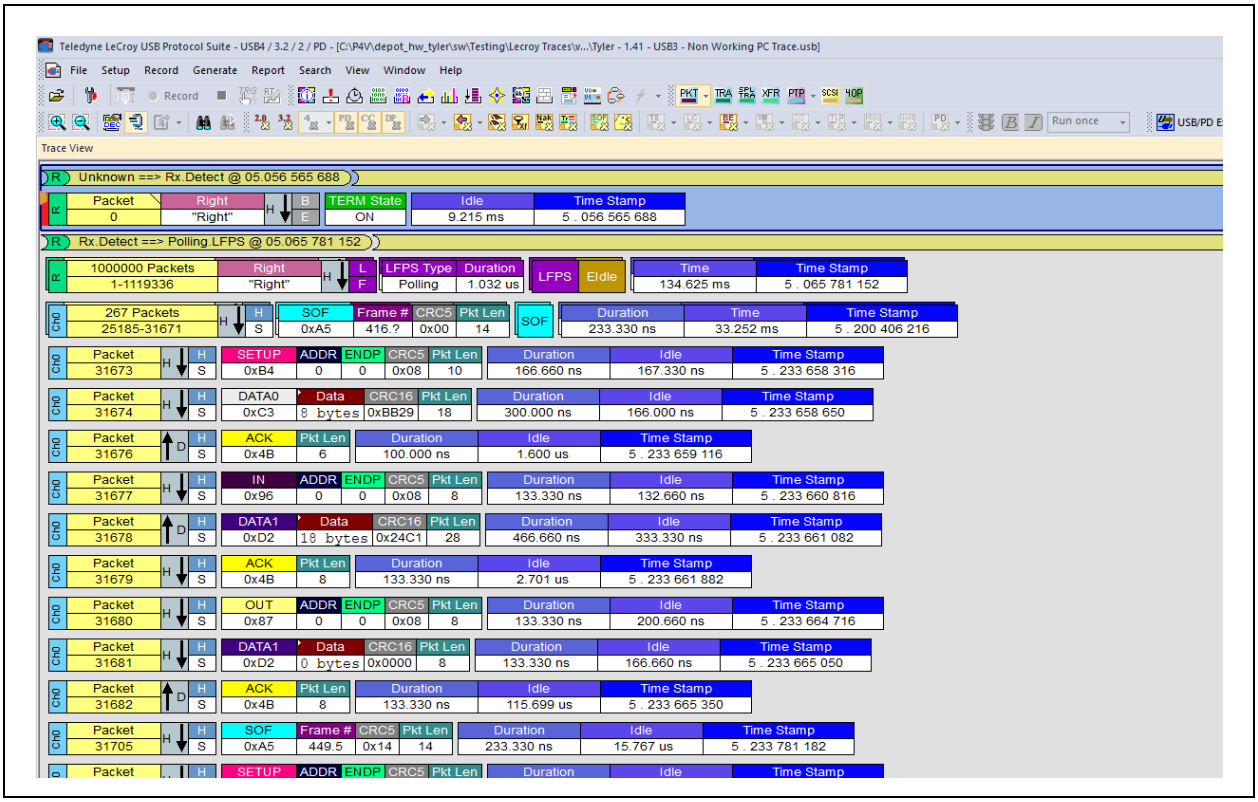
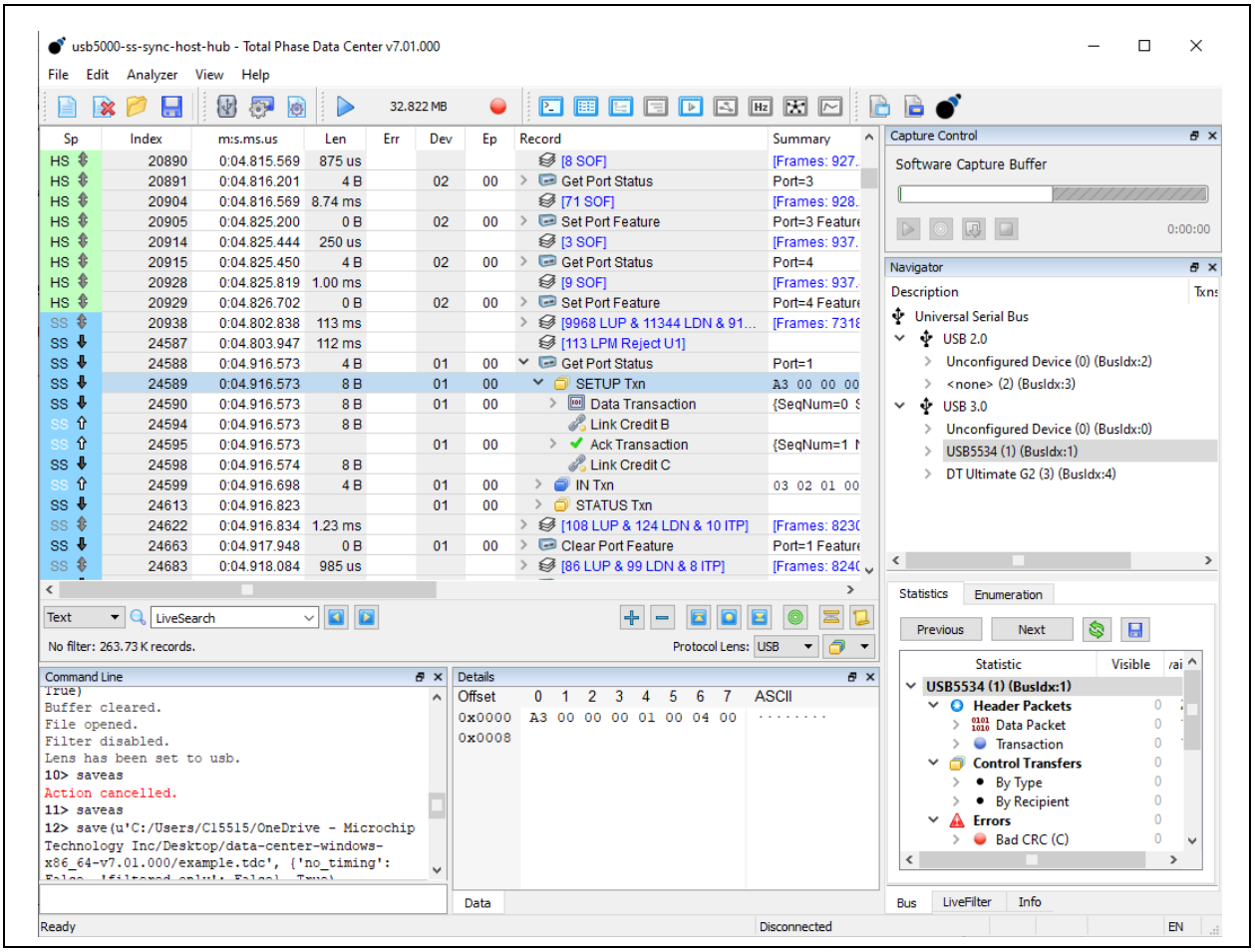


FIGURE 38: EXAMPLE – VIEWING BASIC TRAFFIC IN TOTAL PHASE ANALYZER SOFTWARE



3. Open the device tree to show a list of devices detected by the analyzer. Check if all the devices expected to appear are displayed. See Figure 39 and Figure 40.

FIGURE 39: EXAMPLE – SHOWING DETECTED DEVICES IN LECROY ANALYZER SOFTWARE

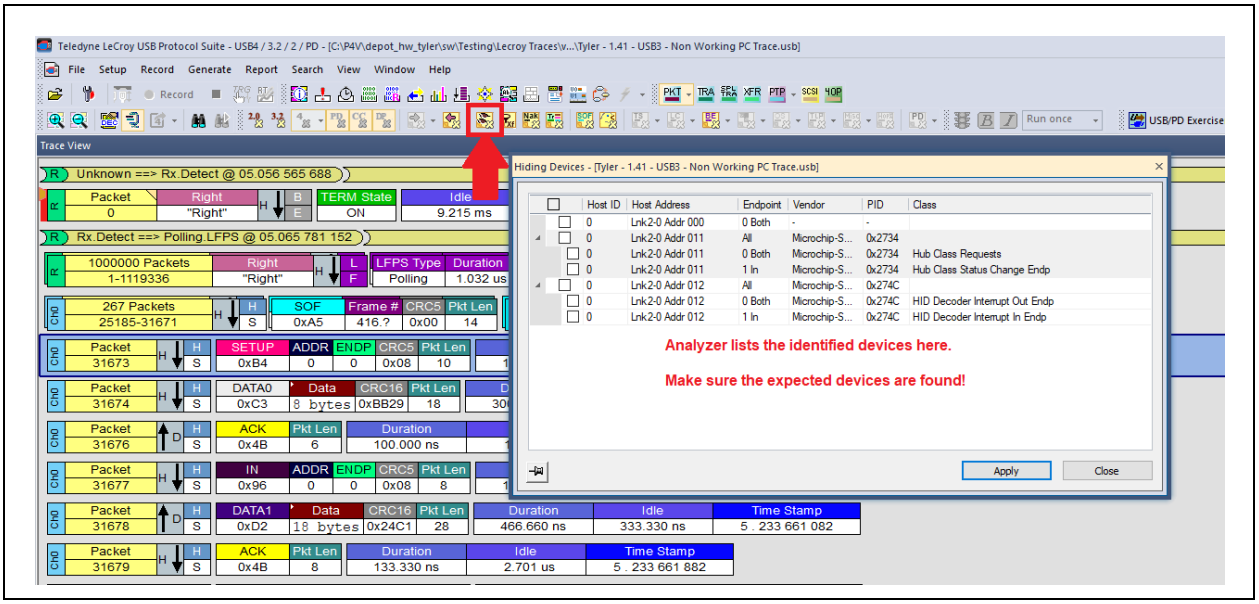
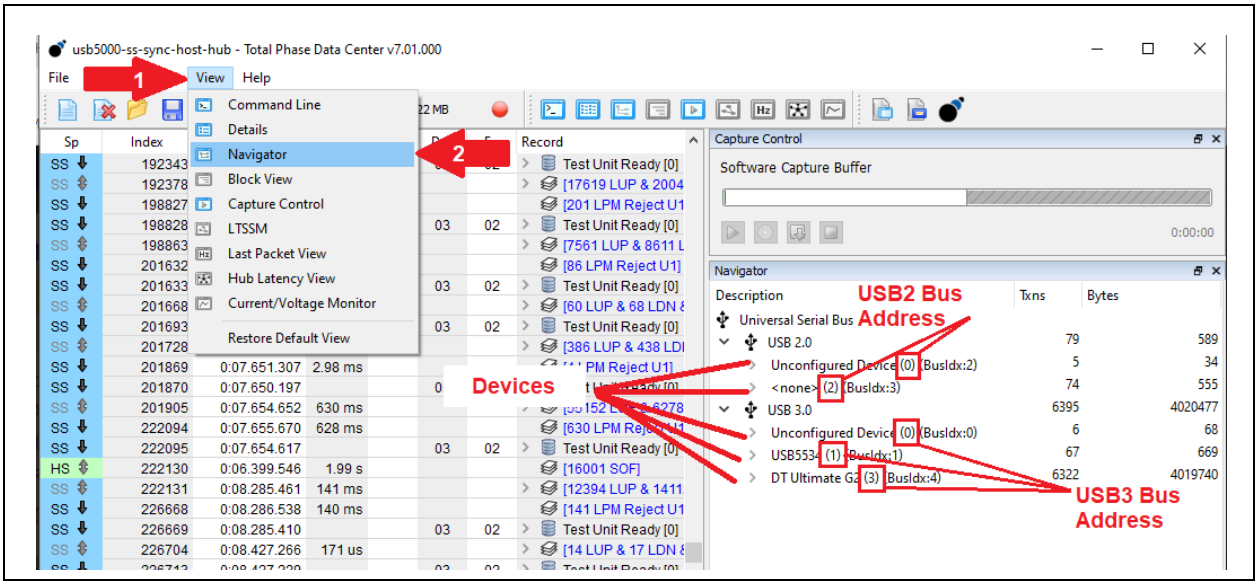


FIGURE 40: EXAMPLE – SHOWING DETECTED DEVICES IN TOTAL PHASE ANALYZER SOFTWARE



4. Check the error summary for any errors found. See Figure 41 and Figure 42.

FIGURE 41: EXAMPLE – ERROR SUMMARY IN LECROY ANALYZER SOFTWARE

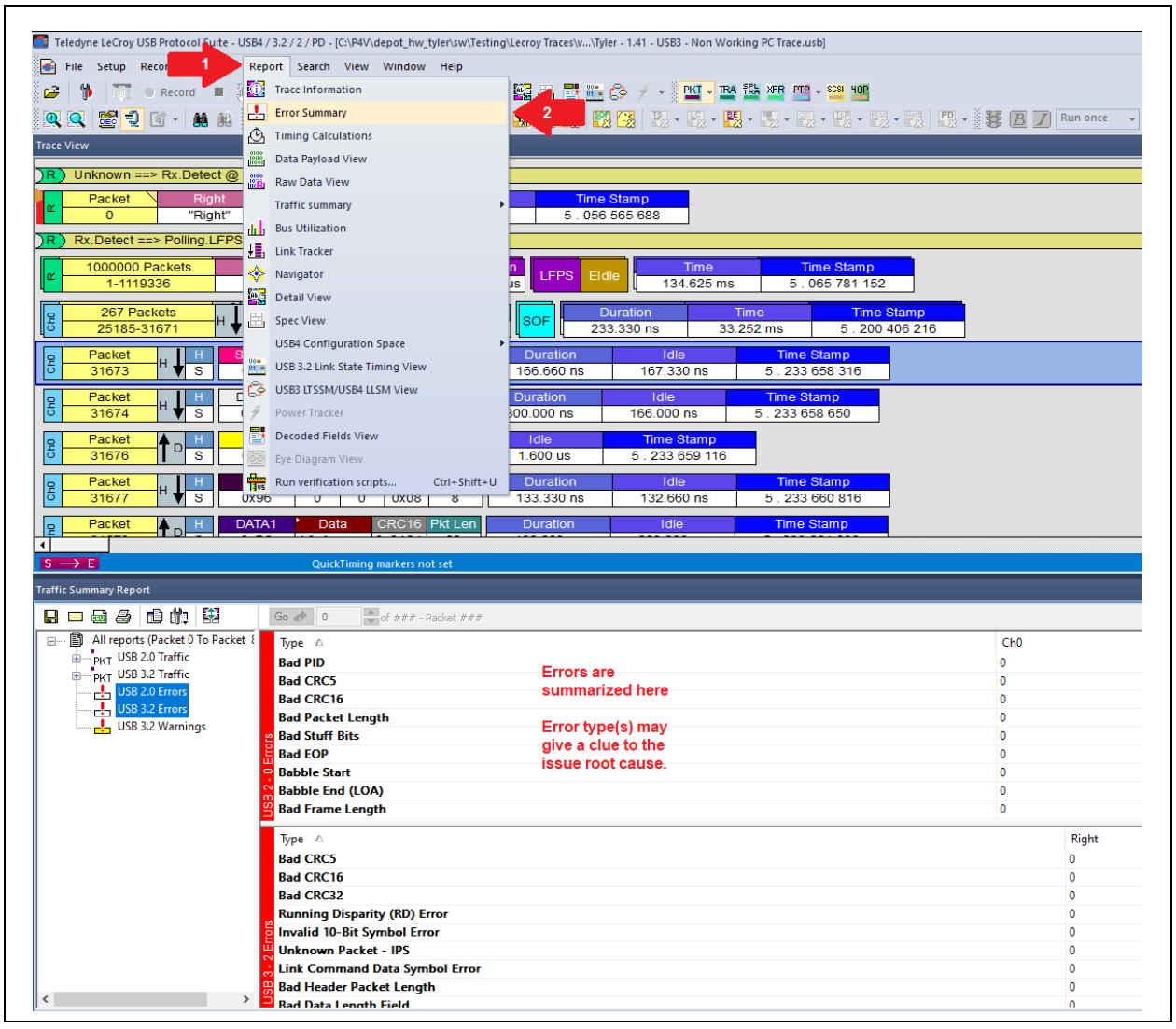
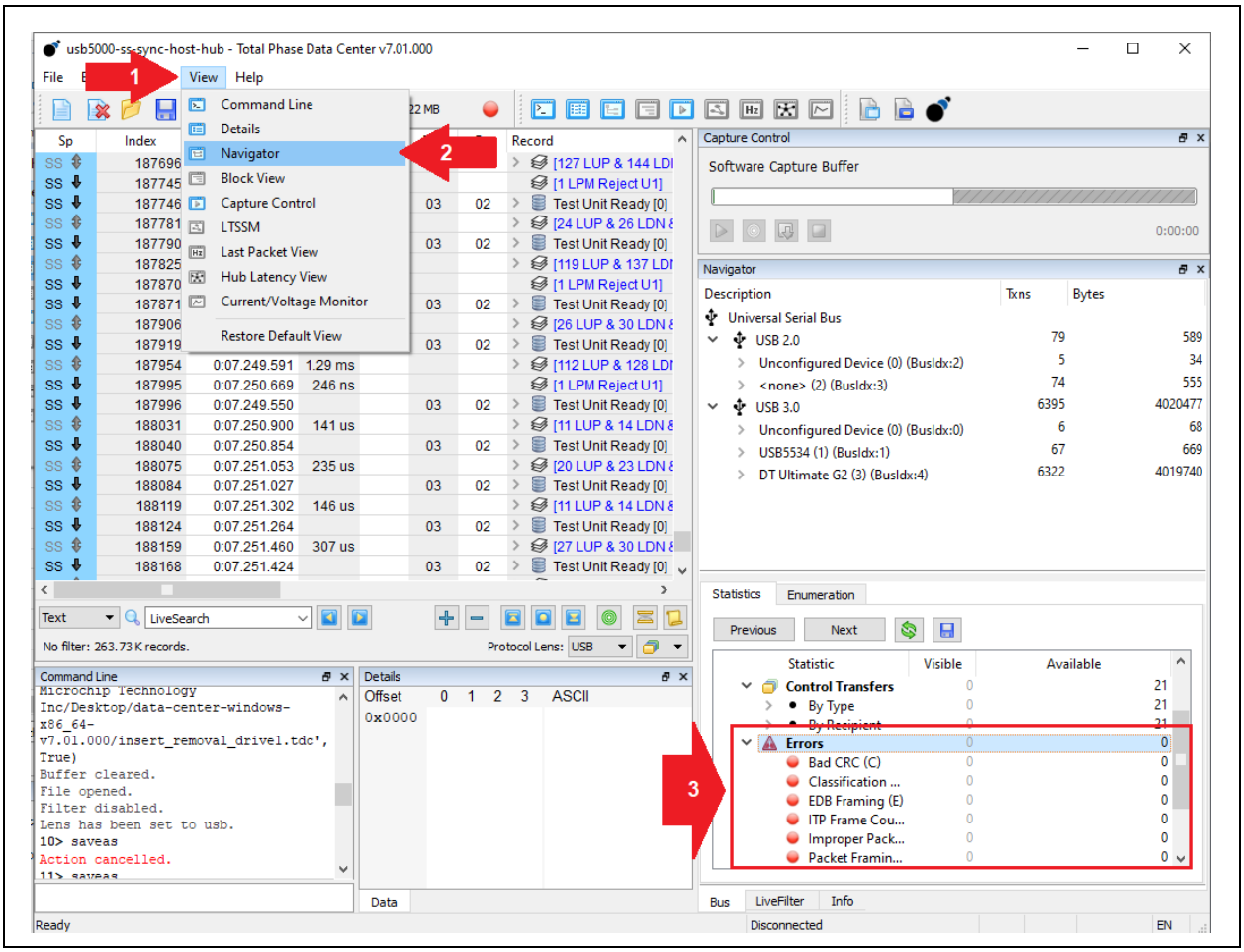


FIGURE 42: EXAMPLE – ERROR SUMMARY IN TOTAL PHASE ANALYZER SOFTWARE

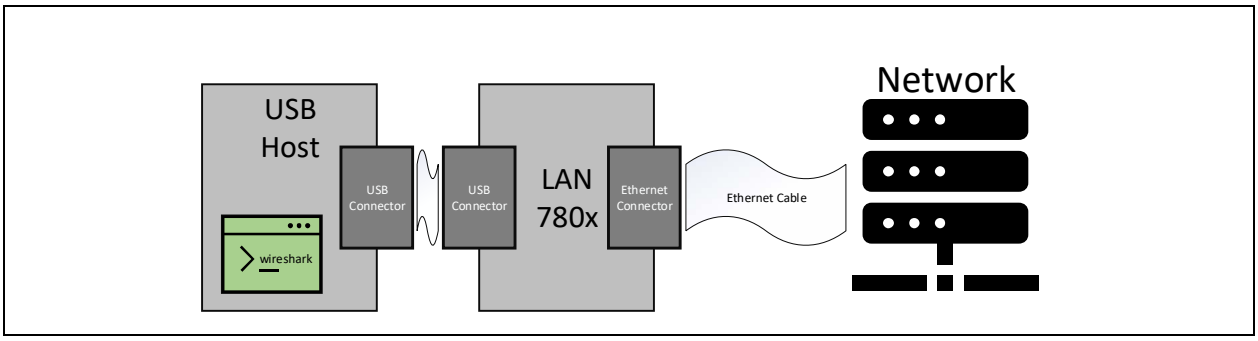


APPENDIX G: HOW TO COLLECT AN ETHERNET WIRESHARK TRACE WITH LAN780X

Wireshark is an application that allows for a wide range of protocols to be traced through software without requiring protocol analyzer software. The support protocols include Ethernet, USB, Bluetooth, and more.

The application just needs to be installed on the host which requires protocol sniffing. See Figure 43.

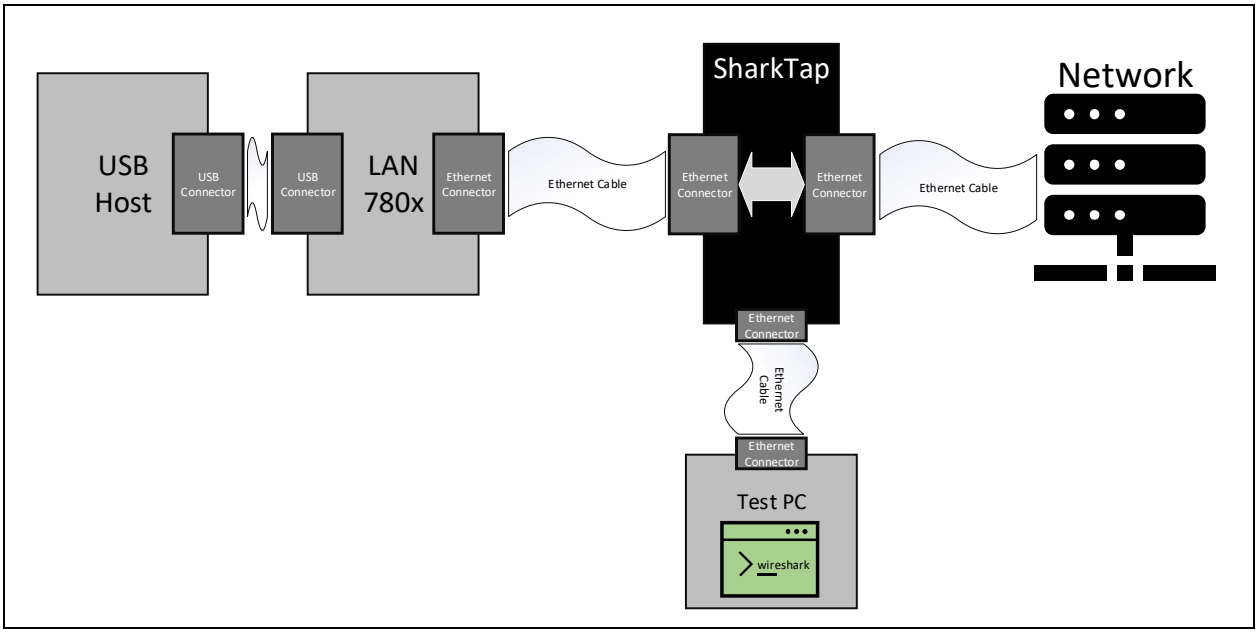
FIGURE 43: WIRESHARK ETHERNET PROTOCOL ANALYZER



If Wireshark cannot be installed for some reason (i.e.: an embedded host with no administrative rights), then a protocol analyzer called a SharkTap (manufactured and sold by midBit Technologies, LLC) can be used.

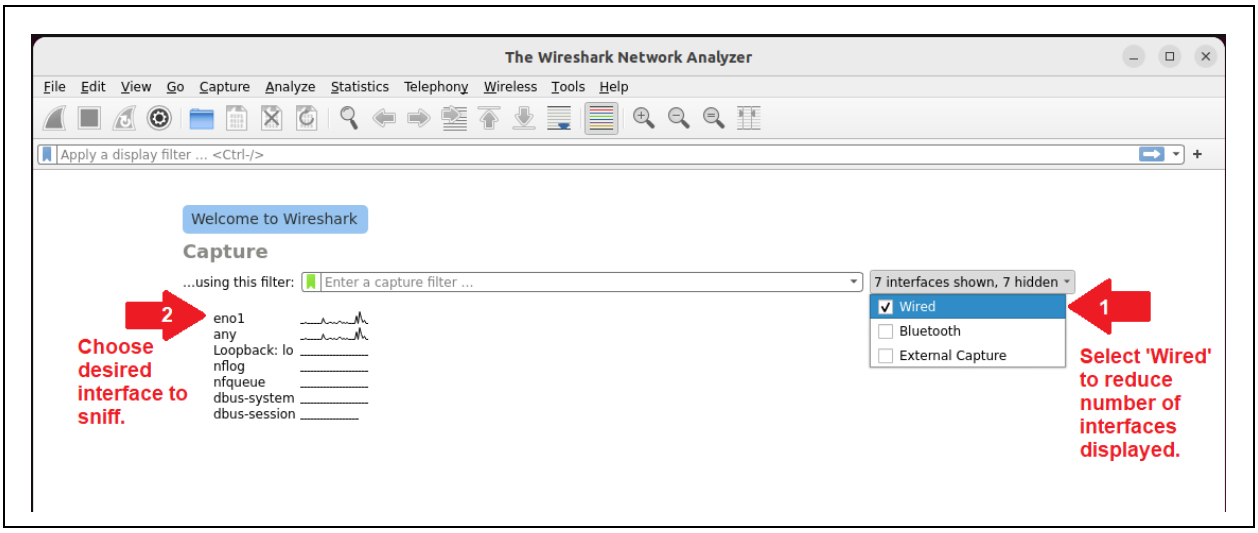
A SharkTap is simply a 2-port Ethernet switch which mirrors all traffic through the pass-through ports to the analyzer port. A separate test PC can be used to run the Wireshark software which collects the trace. See Figure 44.

FIGURE 44: WIRESHARK WITH SHARKTAP PROTOCOL ANALYZER



Download and install the latest version of Wireshark. Wireshark should typically be installed and run using administrator privileges. Once installed, open the program and select the interface to be sniffed. The name of the interface must be known and can be discovered through Device Manager in Windows, or by issuing the 'ifconfig -a' in Linux. See Figure 45.

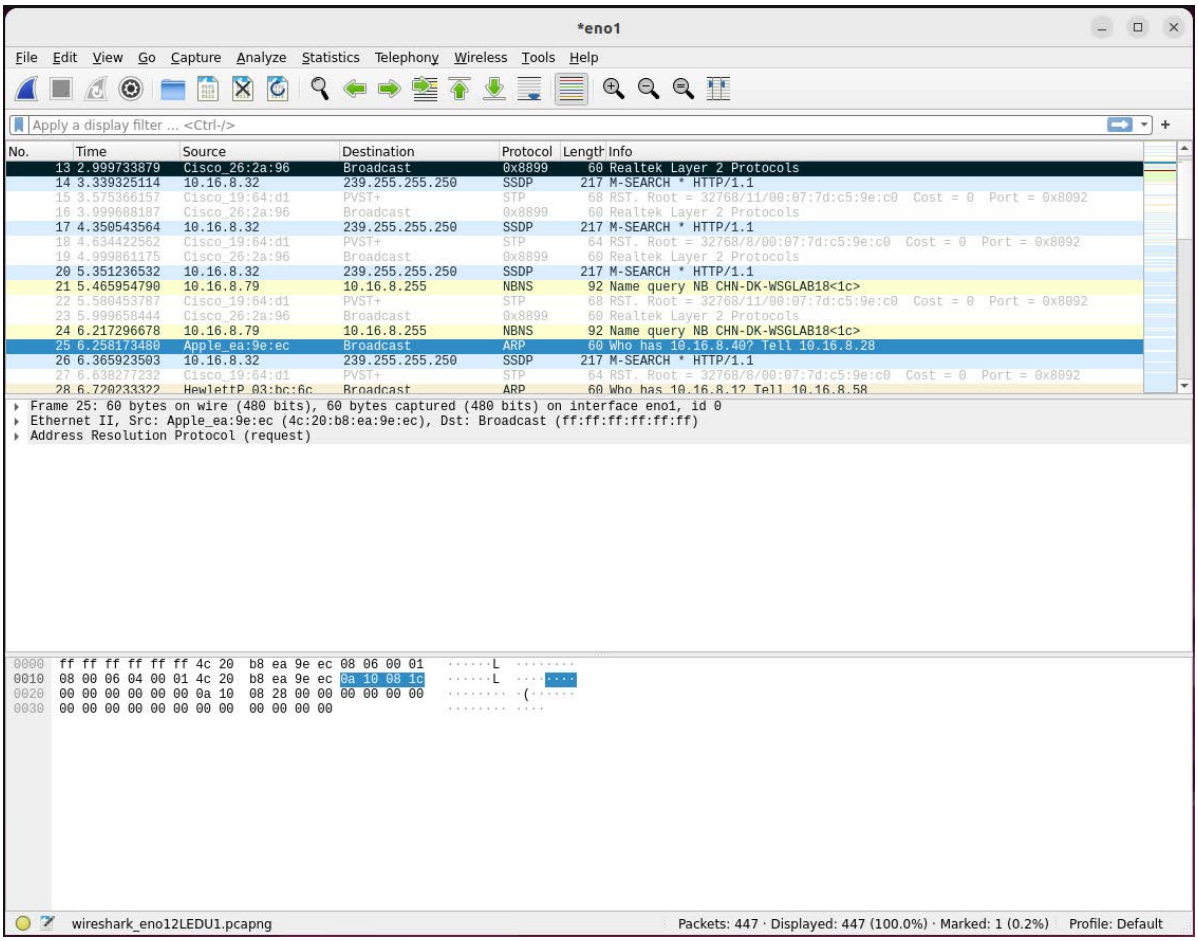
FIGURE 45: WIRESHARK MAIN MENU



Note:	If the desired interface is not shown, ensure Wireshark is being run with administrator privileges.

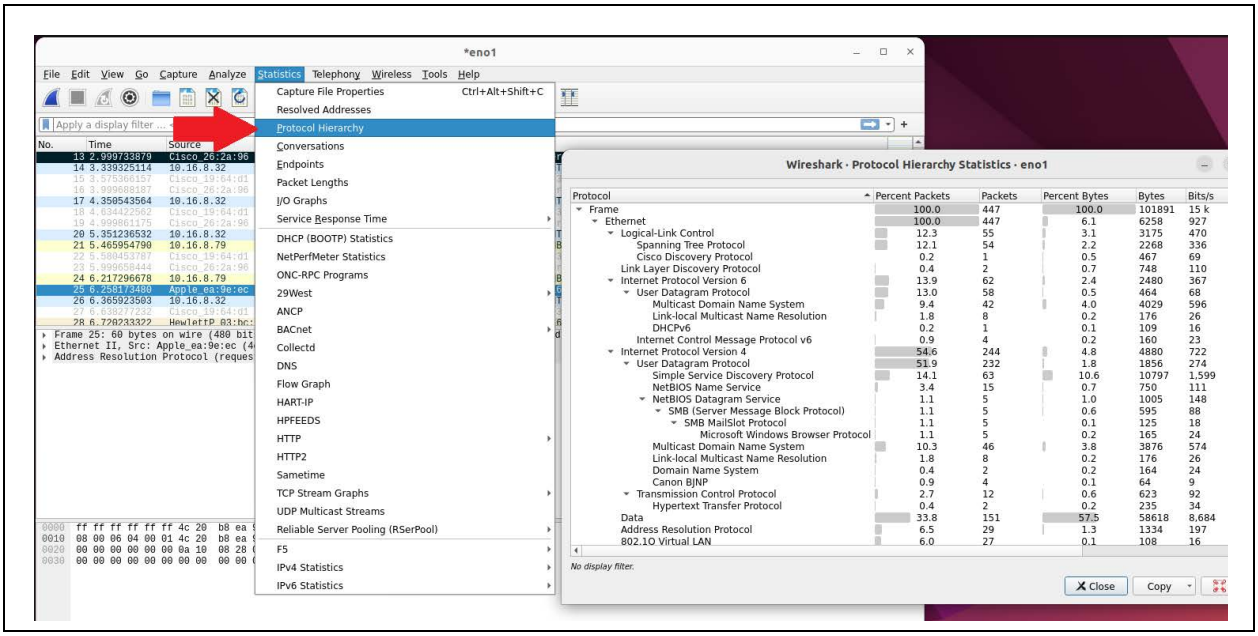
Once an interface is selected, Wireshark will immediately start capturing protocol. Click on the **Stop** button to stop data collection. (See Figure 46.) The software automatically decodes data into an easily readable format. Individual packets can be selected to display the raw binary data.

FIGURE 46: WIRESHARK PROTOCOL CAPTURE



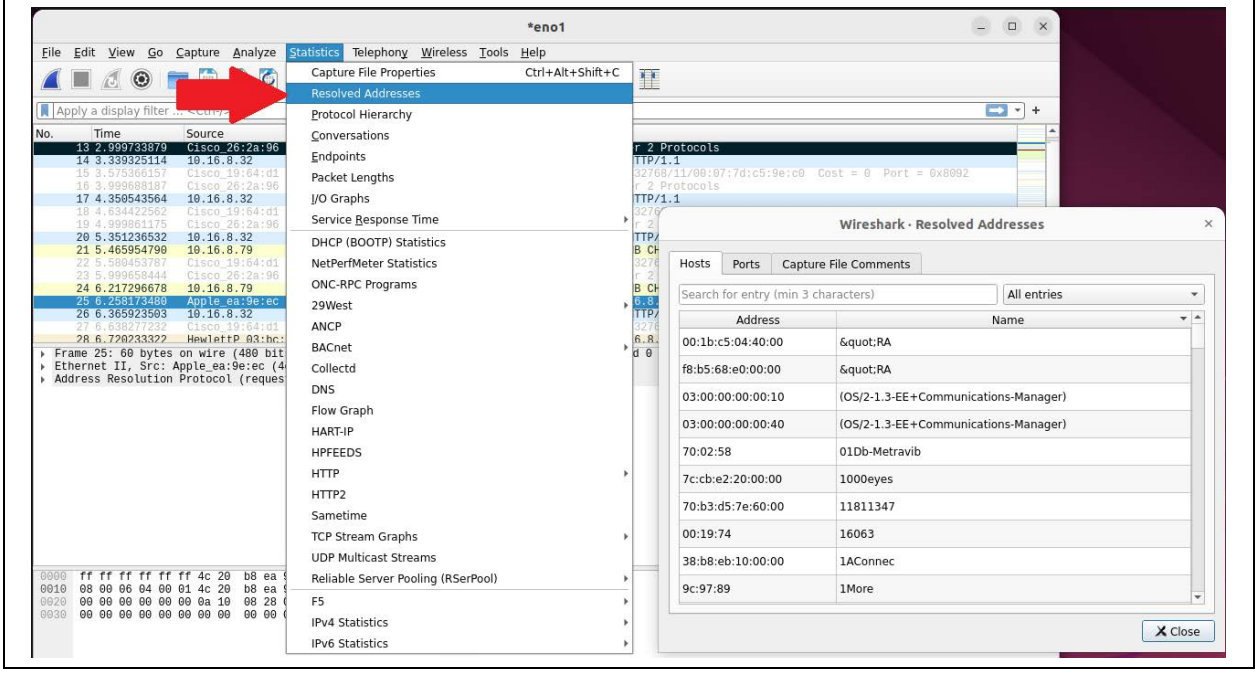
To see a summary of all the protocols collected in the trace, select *Statistics>Protocol Hierarchy*. This will open a window with statistics on what kinds of protocols are present (i.e.: IPv4 or IPv6, UDP, or TCP, etc.). Refer to Figure 47.

FIGURE 47: WIRESHARK PROTOCOL HIERARCHY STATISTICS



To see a summary of all of the devices/addresses detected in the protocol trace, select *Statistics > Resolved Addresses*. Refer to Figure 48.

FIGURE 48: WIRESHARK RESOLVED ADDRESSES



APPENDIX H: APPLICATION NOTE REVISION HISTORY

TABLE H-1: REVISION HISTORY

Revision Level & Date	Section/Figure/Entry	Correction
DS00004839A (12-20-22)	Initial release.	

NOTES:

THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at www.microchip.com. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://microchip.com/support

Note the following details of the code protection feature on Microchip products:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip product is strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is "unbreakable" Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.

ISBN: 978-1-6683-1758-7

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Austin, TX
Tel: 512-257-3370

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453
Tel: 317-536-2380

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608
Tel: 951-273-7800

Raleigh, NC
Tel: 919-844-7510

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110
Tel: 408-436-4270

Canada - Toronto
Tel: 905-695-1980
Fax: 905-695-2078

ASIA/PACIFIC

Australia - Sydney
Tel: 61-2-9868-6733

China - Beijing
Tel: 86-10-8569-7000

China - Chengdu
Tel: 86-28-8665-5511

China - Chongqing
Tel: 86-23-8980-9588

China - Dongguan
Tel: 86-769-8702-9880

China - Guangzhou
Tel: 86-20-8755-8029

China - Hangzhou
Tel: 86-571-8792-8115

China - Hong Kong SAR
Tel: 852-2943-5100

China - Nanjing
Tel: 86-25-8473-2460

China - Qingdao
Tel: 86-532-8502-7355

China - Shanghai
Tel: 86-21-3326-8000

China - Shenyang
Tel: 86-24-2334-2829

China - Shenzhen
Tel: 86-755-8864-2200

China - Suzhou
Tel: 86-186-6233-1526

China - Wuhan
Tel: 86-27-5980-5300

China - Xian
Tel: 86-29-8833-7252

China - Xiamen
Tel: 86-592-2388138

China - Zhuhai
Tel: 86-756-3210040

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444

India - New Delhi
Tel: 91-11-4160-8631

India - Pune
Tel: 91-20-4121-0141

Japan - Osaka
Tel: 81-6-6152-7160

Japan - Tokyo
Tel: 81-3-6880- 3770

Korea - Daegu
Tel: 82-53-744-4301

Korea - Seoul
Tel: 82-2-554-7200

Malaysia - Kuala Lumpur
Tel: 60-3-7651-7906

Malaysia - Penang
Tel: 60-4-227-8870

Philippines - Manila
Tel: 63-2-634-9065

Singapore
Tel: 65-6334-8870

Taiwan - Hsin Chu
Tel: 886-3-577-8366

Taiwan - Kaohsiung
Tel: 886-7-213-7830

Taiwan - Taipei
Tel: 886-2-2508-8600

Thailand - Bangkok
Tel: 66-2-694-1351

Vietnam - Ho Chi Minh
Tel: 84-28-5448-2100

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4485-5910
Fax: 45-4485-2829

Finland - Espoo
Tel: 358-9-4520-820

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Garching
Tel: 49-8931-9700

Germany - Haan
Tel: 49-2129-3766400

Germany - Heilbronn
Tel: 49-7131-72400

Germany - Karlsruhe
Tel: 49-721-625370

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Germany - Rosenheim
Tel: 49-8031-354-560

Israel - Ra'anana
Tel: 972-9-744-7705

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Padova
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Norway - Trondheim
Tel: 47-7288-4388

Poland - Warsaw
Tel: 48-22-3325737

Romania - Bucharest
Tel: 40-21-407-87-50

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Gothenberg
Tel: 46-31-704-60-40

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820