Boston University
Electrical & Computer Engineering

EC463 Capstone Senior Design Project

Problem Definition and Requirements Review

CyberTap

Submitted to:
Cybereason
John Hancock Tower
200 Clarendon St, Boston, MA 02116
(855) 695-8200
by

Team 2
CyberTap
Team Members:
Felipe Dale Figeman fdale@bu.edu
Alex Fatyga afatyga@bu.edu
Evan Lang evanlang@bu.edu
Noah Malhi malhin@bu.edu
Justin Morgan justinfm@bu.edu

Submitted: October 5th, 2019

Client Sign-Off ______________________________________

CyberTap

Table of Contents

Project Summary

Our group is motivated to make a network tap to monitor and analyze Operational Technology (OT) networks handling greater throughput than other competing options. CyberTap is a device that suits the needs of today’s industrial control systems; it provides a way to analyze network traffic and enables detection of any presently occurring security threats without bottlenecking system networks. The device will be able to collect OT network packets, parse and generate metadata for all relevant network protocols of a system, and store the collected packets and generated metadata in storage. Our web application will allow users to observe and query network activity in an understandable and user-friendly manner. It will be implemented on a Field Programmable Gate Array (FPGA), utilizing their ability to quickly process large data loads. This hardware device combines all methods of today’s network monitoring into one hardware scalable device, providing lossless packet interception and backup.

1 Need for this Project

Until recently, cybersecurity was an afterthought for companies and their technological infrastructure. Much progress has been made since in enterprise environments with many companies realizing the importance of keeping their networks secure as the machines in these environments employ the typical internet protocols (TCP/IP) in their communications.

Industrial settings, on the other hand, are typically closed off from the internet. Few machines are able to connect outside of their Local Area Network (LAN). The rest, such as microcontrollers on the industrial equipment, communicate within the network with special Industrial Control System (ICS) protocols, some of which can be unique to a specific manufacturer.

Network sniffing and network protocol analysis in an Industrial Control System (ICS) network enables organizations to collect and analyze the traffic sent between devices, allowing them to identify and track down potentially malicious activities. Conventional network sniffing methods and protocol analyzers are slow as they are mostly software based, failing to offer sufficient analysis of the network protocols used in an ICS environment. In an industrial setting this results in limited visibility into potential cyber attacks. To achieve this without the latency associated with traditional methods, much of the workload will be offloaded to a FPGA.

By performing ICS network protocol analysis on hardware, the latency from traditional software methods is massively reduced without impacting the capabilities of the network sensor. Such a solution would allow for organizations to easily keep their systems secure without impacting the efficiency of their already established infrastructure.

2 Problem Statement and Deliverables

Modern software-based methods of network sniffing and protocol analysis result in limited visibility into potential cyber attacks or massive bottlenecking within ICS environments. This leaves some of the most vital infrastructure control systems vulnerable to outside attack. The goal is to create a system that can monitor and back up the network packets it intercepts so that they can be analyzed and the malicious packets identified, to enable reporting of network attacks in either real time, or post-facto.

The deliverable for this project is a hardware network monitoring device. It will be implemented on an FPGA and must combine both software and hardware components. CyberTap needs to have the ability to intercept network packets and parse all of the relevant network protocols such as TCP and pertinent OT protocols. The device must also analyze said traffic and use the parsings to be stored into transmission elements data. This data includes protocol information, source, destination, size, payload, and a few others. This data must be stored in another device such as a Solid State Drive (SSD), so it can later be retrieved for further processing and analysis. CyberTap must be capable of allowing network flow with analysis without creating a bottleneck. In general, latency overhead should be lower than using the standard Wireshark software solution. The system must also be scalable in a sense that as a network infrastructure grows and a companies needs change, the device adapts. The set-up and use of the device should be user-friendly and allow for quick maintenance.

For software deliverables, there needs to be a web application that will be used to monitor the data. This web application must be queryable according to client specifications and display all the necessary network metadata. The website itself must be secured to protect network data.

For testing, several Raspberry Pi Zeros (between 6 and 12) will send network traffic composed of the OT network protocols through the switch. The generated metadata and packet content will then be compared to the known contents sent by the devices. Packets will be sent at different network rates to ensure no loss of packets due to bandwidth restrictions. Since CyberTap is for ICS, it needs to be a long term solution. Testing the device with packet flow over time will be necessary to prove its longevity.

3 Visualization

The objective of this project is to offer a hardware based network monitor with a minimal impact on network performance. This has led to a configuration that interacts with the network as little as possible, while still capturing all network packets coming through the switch.

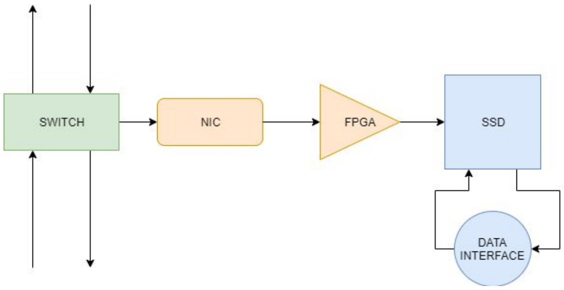


Figure 1 The Network Tap configuration would have the NIC connected to the network switch, using port mirroring to capture a copy of all the packets passing through the switch while not changing its functionality. The NIC would then communicate via PCIe with the FPGA, which will generate metadata on the packets, and write that metadata to the SSD via an M.2 connection.

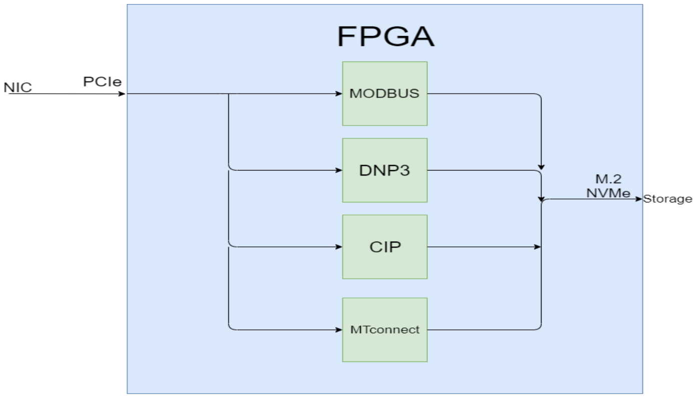
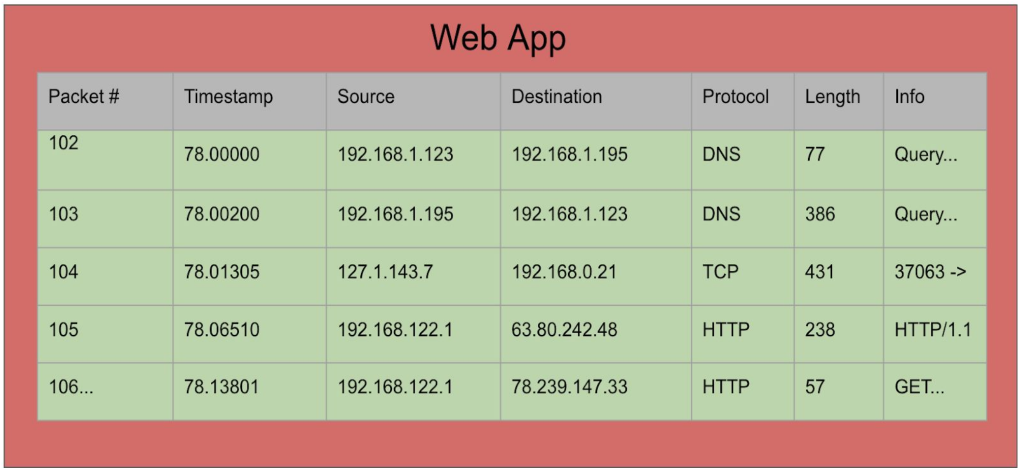


Figure 2 The FPGA will utilize a module system to allow it to be quickly modified for each clients individual needs. Each network protocol will have a corresponding black box that reads, parses, and outputs the relevant information. Different black boxes will be loaded onto the fpga depending on if that type of protocol is used within the current network. This modularity allows fast customization and easy scalability depending on client needs.



Web App

Packet #	Timestamp	Source	Destination	Protocol	Length	Info
102	78.00000	192.168.1.123	192.168.1.195	DNS	77	Query...
103	78.00200	192.168.1.195	192.168.1.123	DNS	386	Query...
104	78.01305	127.1.143.7	192.168.0.21	TCP	431	37063 ->
105	78.06510	192.168.122.1	63.80.242.48	HTTP	238	HTTP/1.1
106...	78.13801	192.168.122.1	78.239.147.33	HTTP	57	GET...

Figure 3 This figure displays the expected web app for users. They will be able to view the network activity. For each packet, there is a packet number, timestamp, source, destination, protocol, length and other info related to it. Users will, additionally, be able to query by source, destination and protocol.

Currently, off-the-shelf standard ports/adapters have been chosen to connect the various parts in the system in order to avoid overcomplications. A PCIe NIC and FPGA have been chosen in order to take advantage of the bare metal protocol. At this point in time, an M.2 SSD is planned to serve as storage in order to prevent packet loss due to insufficient IOPS.

In general, user input will be minimal; they will be able to query by source, destination or protocol in order to find the appropriate packet. The front end will allow users to query packets and monitor activity on the network, allowing easy access to the captured network data.

4 Competing Technologies

Network tapping and monitoring is not a new concept and has been done before. Several companies have similar products such as SharkTapBYP from midBit Technologies, ProfiShark from ProfiTap, and ETAP-2003 from DualComm. There has also been the Passive network tap device patent that has since been abandoned. These competing technologies do not use an FPGA and are more used as portable devices rather than on large scale networks. These all typically are used for small scale troubleshooting rather than for a growing company.

SharkTapBYP from midBit Technologies is an ethernet switch that allows you to 'tap into' an ethernet connection. The device attaches in-line on an ethernet link and delivers copies of packets to a standard ethernet port or a virtual ethernet port over USB. A Test Access Port (TAP) allows the engineer to view the data on an ethernet link. Packets flowing through the network ports are duplicated to the TAP port. The TAP port is typically connected to a PC running Wireshark or similar software. This means that the device itself is responsible for getting into the network and receiving the packets but a third party software like Wireshark is used to analyze and display information on the packets. Therefore, its requirements include an ethernet connection and PC connection so that it can receive and send packets; it does not use an FPGA like CyberTap and is an all in one small, portable device that just needs the ethernet and TAP cables plugged into it.

ProfiTap is a network visibility and quality device testing company. Their network tapping device is called ProfiShark and it utilizes Wireshark or any other software analyzer to capture traffic. The device is used primarily for troubleshooting as it is a small compact device that connects to other devices using USB 3.0. ProfiShark provides features such as timestamping, live capture, statistics and hardware filtering. The device is primarily used for short term testing via a laptop, but long-term traffic capture can be implemented by connecting a NAS to the device. With this long term mode, its features include a Ringbuffer or normal capture mode and the ability to split capture to different files based on size or time. The maximum network latency of their top device, the ProfiShark 10G+, is 10 Gbps (328ns). Their ProfiShark series devices range from ethernet testing to Fiber Network tapping with a total of five different tap devices to choose from.

ETAP-2003 from DualComm is a network tap that is USB powered to monitor and troubleshoot data traffic in a 10/100/1000 Base-T Ethernet network. ETAP-2003 is compatible with "Power over Ethernet" (PoE) so that deployment of the device will not block the end-to-end flow of PoE inline power, which is a useful feature for monitoring or capturing data traffic of a PoE Ethernet link. Like the previous competing technologies, it is a small, portable device that does not use an FPGA. Its captured data will also go through software like Wireshark.

For network taps there has been one patent filed in the US, US20040120259A1, the passive network tap. This patent was filed by Agilent Technologies Inc on 2002-12-20. The following is the abstract for the patent:

> "A tap device comprises a network entry connection, a network exit connection, and a network interconnection path there between. A high impedance tap circuit is electrically connected to the network connection path, and comprises an isolation transformer and active buffer element. Advantageously, a power interruption to said tap circuit does not affect network communications."

The patent claimed that a tap device is comprised of a network entry connection, an exit connection, and a network interconnection between the two. Some of its claims include a method for ensuring uninterrupted monitoring on a network as well as an apparatus for eavesdropping on a data network containing an input, output, and interconnection points. The patent was published on 2004-06-24 and assigned to Agilent Technologies Inc on 2004-10-18. Since then there has not been any activity on the patent and it has been abandoned.

These companies' primary clients are IT technicians that would bring one of these devices to people's homes to test their network. Primarily, they are troubleshooting and/or small scale devices that are not necessarily for large and scaling company use. These companies also use an outside source, such as Wireshark, to parse and generate metadata whereas this will be done on CyberTap's hardware. CyberTap is specifically targeting its use towards companies that need long term solutions for advanced network architectures.

5 Engineering Requirements

FPGA

1. Protocol recognition and metadata generating modules must be parameterizable within verilog design.
2. Back up network packets and generated metadata without loss of packets.
3. Network flow should not be affected by device failure.

Network

4. Be able to collect, recognize, and back up OT network packets.
5. Parse and generate metadata for all relevant network protocols which must also be stored.
6. Minimum 1,024kB/s bandwidth

Physical System

7. Must interface with switch through ethernet.

Price

8. Must be less than 1000 dollars
    a. Includes FPGA, necessary fans, connections, PCIe adapters

Web Application

9. Web App must be implemented to have a user friendly interface to all of the parsed metadata from the system
10. Ability to query by network protocol, destination and source

6 Appendix A References.

10/100/1000Base-T Network Tap. (n.d.). Retrieved from http://www.dualcomm.com/products/usb-powered-10-100-1000base-t-network-tap.

Destroy All Software. (n.d.). Retrieved from https://www.destroyallsoftware.com/compendium/network-protocols?share_key=97d3ba4c24d21147.

DNP3. (2019, September 29). Retrieved from https://en.wikipedia.org/wiki/DNP3#IEEE_Standardization.

DNP3 Tutorial Part 4: Understanding DNP3 Message Structure. (n.d.). Retrieved from https://www.dpstele.com/dnp3/tutorial-understanding-message-structure.php.

Home. (n.d.). Retrieved from http://www.midbittech.com/index.html.

IEEE Standard for Electric Power Systems Communications -- Distributed Network Protocol (DNP3). (2010, July 1). Retrieved October 4, 2019, from https://ieeexplore.ieee.org/document/5518537.

Modbus. (2019, September 23). Retrieved from https://en.wikipedia.org/wiki/Modbus.

Modbus Organization. (n.d.). Retrieved from http://www.modbus.org/specs.php.

ProfiShark Network TAPs. (n.d.). Retrieved from https://www.profitap.com/profishark-network-taps/.

Rouse, M., & Gerwig, K. (n.d.). What is TCP/IP and How Does It Work? Retrieved from https://searchnetworking.techtarget.com/definition/TCP-IP.

TCP Throughput Calculator. (n.d.). Retrieved from https://www.switch.ch/network/tools/tcp_throughput/.

US20040120259A1 - Passive network tap device. (n.d.). Retrieved from https://patents.google.com/patent/US20040120259A1/en.

7 Appendix B Supportive Research.

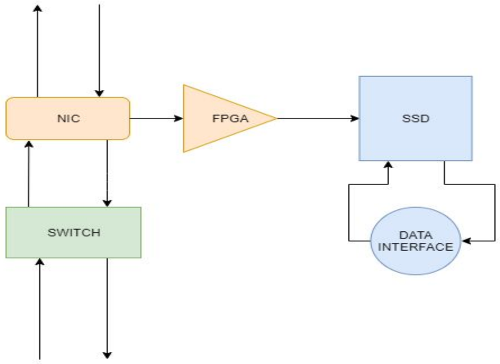


Figure 4 The first possible configuration of CyberTap is bump in the wire. The NIC is placed directly in the network's path, acting as a passthrough while also routing the data to the fpga. This configuration would allow the cybertap system to affect the network, however this is not a feasible design as it would affect the network flow on device failure (unless a fail open NIC was used). This design was considered but ultimately the Network Tap Configuration in Figure 1 was decided upon.

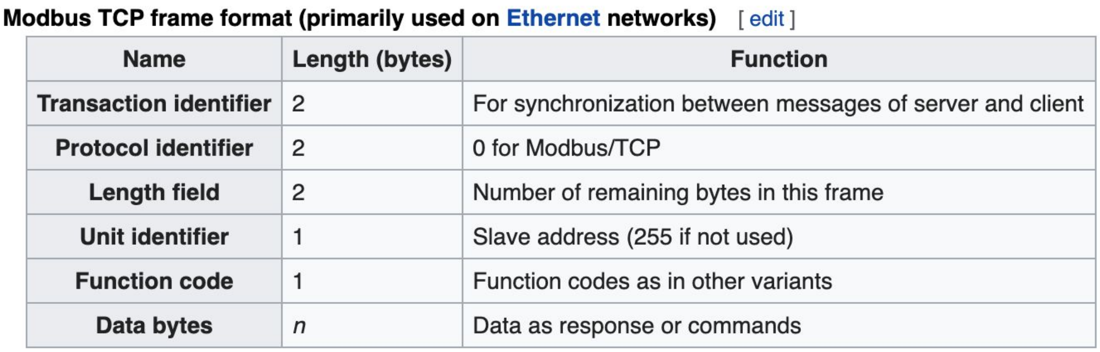
Industrial Control Systems (ICS):

Operational Technology (OT) – the hardware and software dedicated to detecting or causing changes in physical processes through direct monitoring and/or control of physical devices such as valves, pumps, etc.

Networking Protocols Research:

<u>Modbus TCP frame format (primarily used on Ethernet networks)</u>

Modbus is a de facto standard serial communications protocol used for connecting industrial electronic devices within industrial control systems. The main reasons for the use of Modbus in the industrial environment are the following: it was developed with industrial applications in mind, it is openly published and royalty-free, it is easy to deploy and maintain, and it allows for the movement of raw bits or words without placing many restrictions on vendors.

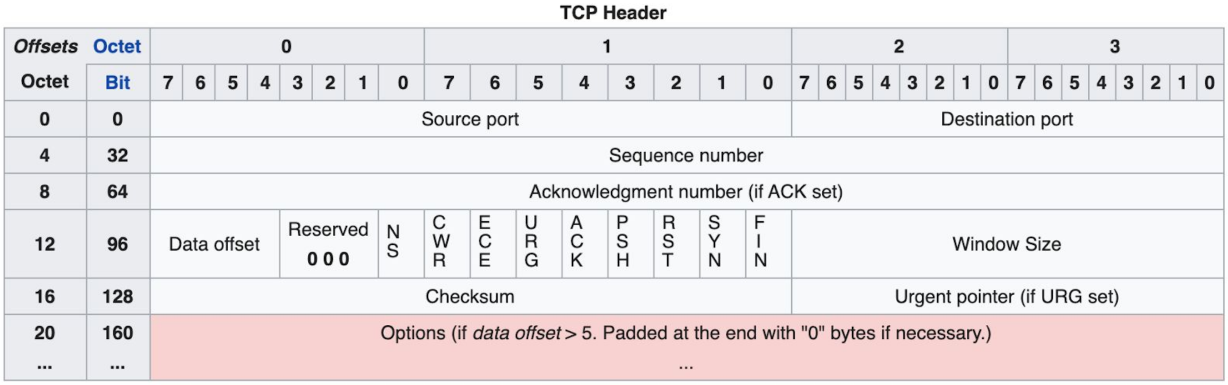


Modbus TCP frame format (primarily used on Ethernet networks) [edit]

Name	Length (bytes)	Function
Transaction identifier	2	For synchronization between messages of server and client
Protocol identifier	2	0 for Modbus/TCP
Length field	2	Number of remaining bytes in this frame
Unit identifier	1	Slave address (255 if not used)
Function code	1	Function codes as in other variants
Data bytes	*n*	Data as response or commands

Figure 5 CyberTap will be working with the Ethernet protocol (the simulated OT network is going to be a Local Area Network). Therefore, it will be using the Modbus TCP frame format shown above.

Figure 6 Above is a depiction of the structure of the total message, including the TCP header required in order to establish the connection between devices. This TCP header is where metadata for the source device, the destination device, and the protocol type will be generated from.



TCP Header

Offsets	Octet	0								1								2								3							
Octet	Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0	0	Source port																Destination port															
4	32	Sequence number																															
8	64	Acknowledgment number (if ACK set)																															
12	96	Data offset				Reserved 0 0 0			NS	CWR	ECE	URG	ACK	PSH	RST	SYN	FIN	Window Size															
16	128	Checksum																Urgent pointer (if URG set)															
20 ...	160 ...	Options (if *data offset* > 5. Padded at the end with "0" bytes if necessary.) ...																															

Figure 7 The Modbus information is embedded within the data portion of the TCP/IP packet. To go a level even lower, above is how a TCP header is composed. The diagram above matches that of the IETF (Internet Engineering Task Force) but is formatted and presented in a cleaner manner.

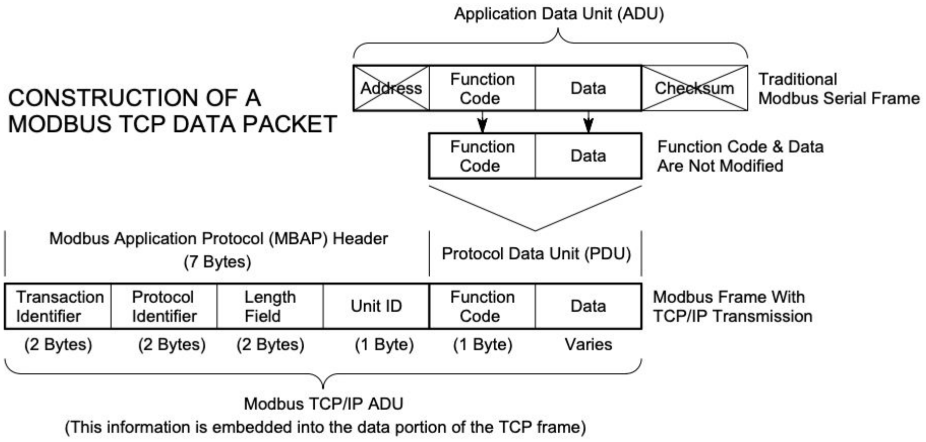
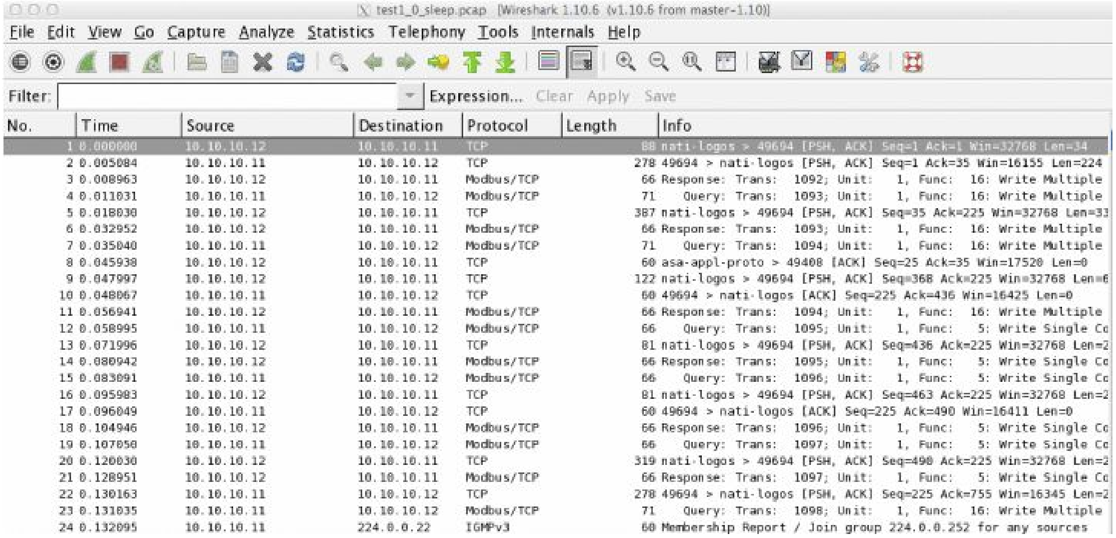

Application Data Unit (ADU)
CONSTRUCTION OF A MODBUS TCP DATA PACKET
Address
Function Code
Data
Checksum
Traditional Modbus Serial Frame
Function Code
Data
Function Code & Data Are Not Modified
Modbus Application Protocol (MBAP) Header (7 Bytes)
Protocol Data Unit (PDU)
Transaction Identifier
Protocol Identifier
Length Field
Unit ID
Function Code
Data
Modbus Frame With TCP/IP Transmission
(2 Bytes)
(2 Bytes)
(2 Bytes)
(1 Byte)
(1 Byte)
Varies
Modbus TCP/IP ADU
(This information is embedded into the data portion of the TCP frame)

Figure 8 This diagram fully describes how the two protocols fit together to form Modbus/TCP-IP.

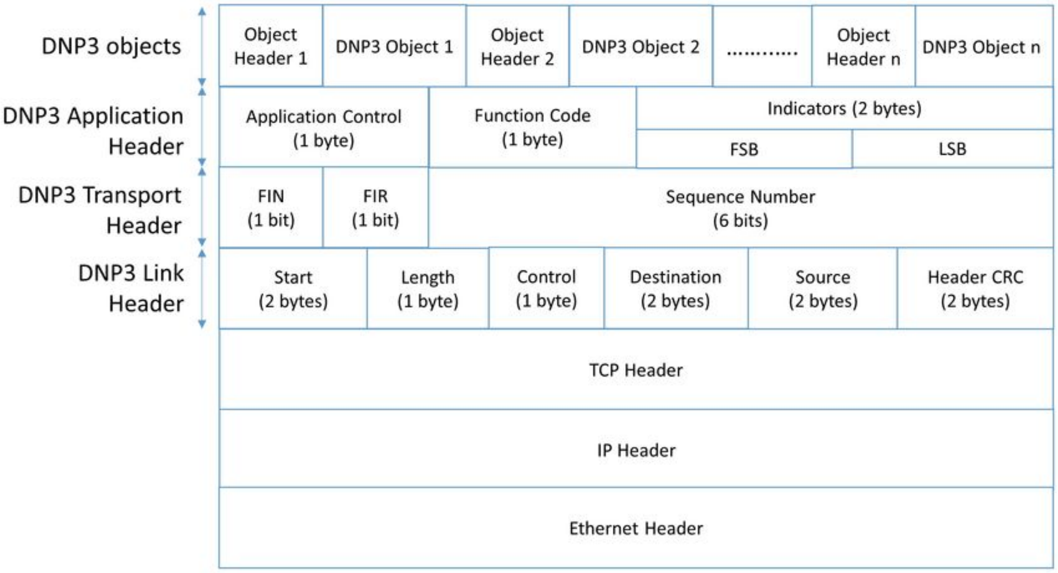

test1_0_sleep.pcap [Wireshark 1.10.6 (v1.10.6 from master-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save

```
No.   Time      Source       Destination  Protocol   Length Info
    1 0.000000  10.10.10.12  10.10.10.11  TCP           88 nati-logos > 49694 [PSH, ACK] Seq=1 Ack=1 Win=32768 Len=34
    2 0.005084  10.10.10.11  10.10.10.12  TCP          278 49694 > nati-logos [PSH, ACK] Seq=1 Ack=35 Win=16155 Len=224
    3 0.008963  10.10.10.12  10.10.10.11  Modbus/TCP    66 Response: Trans:  1092; Unit:   1, Func:  16: Write Multiple
    4 0.011031  10.10.10.11  10.10.10.12  Modbus/TCP    71    Query: Trans:  1093; Unit:   1, Func:  16: Write Multiple
    5 0.018030  10.10.10.12  10.10.10.11  TCP          387 nati-logos > 49694 [PSH, ACK] Seq=35 Ack=225 Win=32768 Len=33
    6 0.032952  10.10.10.12  10.10.10.11  Modbus/TCP    66 Response: Trans:  1093; Unit:   1, Func:  16: Write Multiple
    7 0.035040  10.10.10.11  10.10.10.12  Modbus/TCP    71    Query: Trans:  1094; Unit:   1, Func:  16: Write Multiple
    8 0.045938  10.10.10.12  10.10.10.11  TCP           60 asa-appl-proto > 49408 [ACK] Seq=25 Ack=35 Win=17520 Len=0
    9 0.047997  10.10.10.12  10.10.10.11  TCP          122 nati-logos > 49694 [PSH, ACK] Seq=368 Ack=225 Win=32768 Len=6
   10 0.048067  10.10.10.11  10.10.10.12  TCP           60 49694 > nati-logos [ACK] Seq=225 Ack=436 Win=16425 Len=0
   11 0.056941  10.10.10.12  10.10.10.11  Modbus/TCP    66 Response: Trans:  1094; Unit:   1, Func:  16: Write Multiple
   12 0.058995  10.10.10.11  10.10.10.12  Modbus/TCP    66    Query: Trans:  1095; Unit:   1, Func:   5: Write Single Co
   13 0.071996  10.10.10.12  10.10.10.11  TCP           81 nati-logos > 49694 [PSH, ACK] Seq=436 Ack=225 Win=32768 Len=2
   14 0.080942  10.10.10.12  10.10.10.11  Modbus/TCP    66 Response: Trans:  1095; Unit:   1, Func:   5: Write Single Co
   15 0.083091  10.10.10.11  10.10.10.12  Modbus/TCP    66    Query: Trans:  1096; Unit:   1, Func:   5: Write Single Co
   16 0.095983  10.10.10.12  10.10.10.11  TCP           81 nati-logos > 49694 [PSH, ACK] Seq=463 Ack=225 Win=32768 Len=2
   17 0.096049  10.10.10.11  10.10.10.12  TCP           60 49694 > nati-logos [ACK] Seq=225 Ack=490 Win=16411 Len=0
   18 0.104946  10.10.10.12  10.10.10.11  Modbus/TCP    66 Response: Trans:  1096; Unit:   1, Func:   5: Write Single Co
   19 0.107050  10.10.10.11  10.10.10.12  Modbus/TCP    66    Query: Trans:  1097; Unit:   1, Func:   5: Write Single Co
   20 0.120030  10.10.10.12  10.10.10.11  TCP          319 nati-logos > 49694 [PSH, ACK] Seq=490 Ack=225 Win=32768 Len=2
   21 0.128951  10.10.10.12  10.10.10.11  Modbus/TCP    66 Response: Trans:  1097; Unit:   1, Func:   5: Write Single Co
   22 0.130163  10.10.10.11  10.10.10.12  TCP          278 49694 > nati-logos [PSH, ACK] Seq=225 Ack=755 Win=16345 Len=2
   23 0.131035  10.10.10.11  10.10.10.12  Modbus/TCP    71    Query: Trans:  1098; Unit:   1, Func:  16: Write Multiple
   24 0.132095  10.10.10.11  224.0.0.22   IGMPv3        60 Membership Report / Join group 224.0.0.252 for any sources
```

Figure 9 A Wireshark example of taking in a modbus/TCP protocol.

<u>DNP3</u>

Another industry standard protocol for communication within OT networks is DNP3. DNP3 uses the following message architecture shown below.



DNP3 objects	Object Header 1	DNP3 Object 1	Object Header 2	DNP3 Object 2	....	Object Header n	DNP3 Object n
DNP3 Application Header	Application Control (1 byte)	Function Code (1 byte)	Indicators (2 bytes) FSB LSB				
DNP3 Transport Header	FIN (1 bit)	FIR (1 bit)	Sequence Number (6 bits)				
DNP3 Link Header	Start (2 bytes)	Length (1 byte)	Control (1 byte)	Destination (2 bytes)	Source (2 bytes)	Header CRC (2 bytes)	
	TCP Header						
	IP Header						
	Ethernet Header						

Figure 10 This diagram includes the surrounding TCP/IP header to show what a full message looks like. The metadata that will be generated will focus on the data provided by the transport headers, so the primary reason for outlining and understanding the DNP3 protocol structure will be for identifying and labeling the packet based on its message structure.

Bandwidth

Assuming there are approximately 40 devices connected to a switch per critical piece of infrastructure within a factory, and assuming an average 10ms scan time and 256 byte message size per device (average Modbus/TCP message size), this means a maximum of 1,024kB/s of bandwidth is required. Due to the ubiquity of Gigabit Ethernet and for the sake of scalability, the final bandwidth capabilities will likely be several Megabytes per second.

Operational Technology Network Research:

Kinds of Devices:

Programmable Logic Controllers (PLCs)

The main difference from most other computing devices is that PLCs are intended-for and therefore tolerant-of more severe conditions (such as dust, moisture, heat, cold), while offering extensive input/output (I/O) to connect the PLC to sensors and actuators. PLC input can include simple digital elements such as limit switches, analog variables from process sensors (such as temperature and pressure), and more complex data such as that from positioning or machine vision systems. PLC output can include elements such as indicator lamps, sirens, electric motors, pneumatic or hydraulic cylinders, magnetic relays, solenoids, or analog outputs. The input/output arrangements may be built into a simple PLC, or the PLC may have external I/O modules attached to a fieldbus or computer network that plugs into the PLC.

Scan time

Scan times of a few milliseconds may be encountered for small programs and fast processors, but for older processors and very large programs much longer scan times (on the order of 100 ms) may be encountered.

Process of a scan cycle

There are 5 main steps in a scan cycle:

1. Reading inputs
2. Executing the program
3. Processing communication requests
4. Executing CPU diagnostics
5. Writing outputs