"Ethernet And Implementation In Automation"

Robert Rawlyk
Beckhoff Automation, LLC
North American Headquarters
Burnsville, Minnesota USA

ABSTRACT – *There has been a significant movement in recent years to bring Ethernet to the factory floor. Many were predicting the replacement of common industrial fieldbus systems with Ethernet. At present, there are several different Ethernet protocols for industrial automation and communication. The intention of this paper is to explain the differences between Ethernet and the various protocols that are transmitted over Ethernet. In understanding the differences between various protocols and the demands of an application, an automation specialist can determine what protocol(s) would be appropriate for their application. In order to bring Ethernet to the factory floor, it is also important to understand what hardware and knowledge for configuring that hardware is required.*

I. INTRODUCTION

There are many misconceptions about what Ethernet is and what Ethernet isn't. Technically, the IEEE defines Ethernet to be the IEEE 802.3 standard and in particular, Carrier Sense Multiple Access Collision Detection (CSMA/CD). This specification covers the physical hardware and voltage levels used as well as the makeup of an Ethernet Frame. A fairly common misconception from the office world is that the TCP/IP protocol is Ethernet – after all, you plug an Ethernet cable into your PC to get access to the corporate network and the Internet. Technically, Ethernet is just the medium to transmit encapsulated HTTP, TCP/IP, and many other protocols data messages.

II. BACKGROUND NETWORK COMMUNICATION

There are two common models of network communication. One is the more theoretical or top-down ISO/OSI 7 Layer communication model; the other is the more bottom-up and widely-installed, TCP/IP model. Since the TCP/IP model was in place before the ISO/OSI standard was made, it is very common to try to fit TCP/IP into the ISO/OSI model. Regardless, the desire for both models is such that as long as the rules are followed, any layer can be replaced and the surrounding layers are unaffected.

<table>
<tr><th>ISO/OSI Model</th><th>TCP/IP Model</th></tr>
<tr><td>7. Application Layer</td><td rowspan="3">4. Application Layer</td></tr>
<tr><td>6. Presentation Layer</td></tr>
<tr><td>5. Session Layer</td></tr>
<tr><td>4. Transport Layer</td><td>3. Transport Layer</td></tr>
<tr><td>3. Network Layer</td><td>2. Network Layer (IP Layer)</td></tr>
<tr><td rowspan="2">2. Data Link Layer</td><td>1. Logical Link Layer (IEEE 802.2)</td></tr>
<tr><td rowspan="2">Medium Access Control (MAC)
IEEE 802.3, 802.4, 802.5, 802.11</td></tr>
<tr><td>1. Physical Layer</td></tr>
</table>

Table I – Communication Layer Models[1]

The upper layers are important for communication as a whole; the "Ethernet" side of things only covers the "lower" portion of the Data Link Layer and Physical Layer according to ISO/OSI and the only media access control for the TCP/IP model.

If we treat a conversation between multiple parties as a layered communication, the desired end result is that any person can communicate with others and all participants will understand what has been said. We can add and exchange layers and still get the messages across. For example, if the participants are in the same room and speak the same language, the physical layer is air and each person corresponds to layer 7 – the application layer. If we separate the participants we may need to add a layer, such as a

phone line. At first it may be a land line that carries the message. The land line knows nothing about the message nor the language that the message is in, it merely converts the audible noise to electricity and transmits it. We can very easily exchange the physical layer of a land line for a cell phone or walkie-talkie and the conversation will be transmitted properly. If we switch to a written message and fax machine we have added a layer. The fax machine converts the scanned information, transmits it over the same physical layer – be it wireless or landline – and the fax machine on the other end decodes it and prints out the message.

Computer networking uses these layers to exchange information from different types of computers, different applications and over different networks. A web page does not need to be created for the target viewing device. The page can be viewed from a web-enabled cell phone or from a Mac with a modem, a UNIX station with a shared network connection or a MS-Windows PC with a cable modem.

Controls applications utilizing industrial networks such as Profibus or DeviceNet use the same layering principles. With flexible control software it is possible to readily interchange networks and exchange the same information with sensors. A photo eye can be just as easily wired into a Profibus network as a DeviceNet, Modbus, SERCOS or Ethernet derivative network. Layers below the application are interchanged for different hardware and protocols. An email can be sent over DeviceNet just as it can be exchanged over Ethernet - it's not as common, but nothing is preventing it.

III. ETHERNET COMMUNICATION

Ethernet Communication follows the following specifications: Each Ethernet device must have a MAC ID (Medium Access Control) Identifier or MAC Address, be it your PC or your Xbox™. The MAC address is a 6 byte hexadecimal number, usually in the form of 00-50-F2-C3-2F-44. The addresses are globally unique. Each manufacturer of Ethernet devices is assigned a block of MAC ID's and each manufacturer cannot request more ID's until over 80% of the addresses are used. With over 281 trillion addresses available, it will be some time before they are all used. Ethernet communication at a high level is simple and elegant. Every device on the network listens for the network to be idle, if the network is idle and there is something to be transmitted, the device starts transmitting according to a specific message format.

Figure 1 – Ethernet Frame Structure

Component	Function	Number of Bytes
Preamble	Synchronization for Receive frame	7 Bytes (1010101010101….)
SFD Start Frame Delimiter	Indicates the start of the frame	1 Byte (10101011)
DA Destination Address	MAC ID to receive the message	6 bytes (MAC ID format)
SA Sender Address	MAC ID of the sending device	6 bytes (MAC ID format)
LEN Length/Ethertype	Length of message or if LEN > 0x600 it designates a type of Ethernet message	2 Bytes (EtherCAT, Profibus, Powerlink have registered Ethertypes)
Data	Data package to be sent	<1500 bytes
Pad	Pad the Data Frame to 46 Bytes if necessary	If the data is less than 46 bytes then padding bytes are added to bring the data up to 46 bytes
FCS Frame Check Sequence	CRC Data	Filled in with data according to a math formula for error detection

Table II – Ethernet Frame Structure[2]

Since any device has equal access, an opportunity to communicate on the network and some delay in sending the data and responding, a single frame must remain on the network for 25.6 microseconds in order for a collision to be detected, creating the minimum frame length.

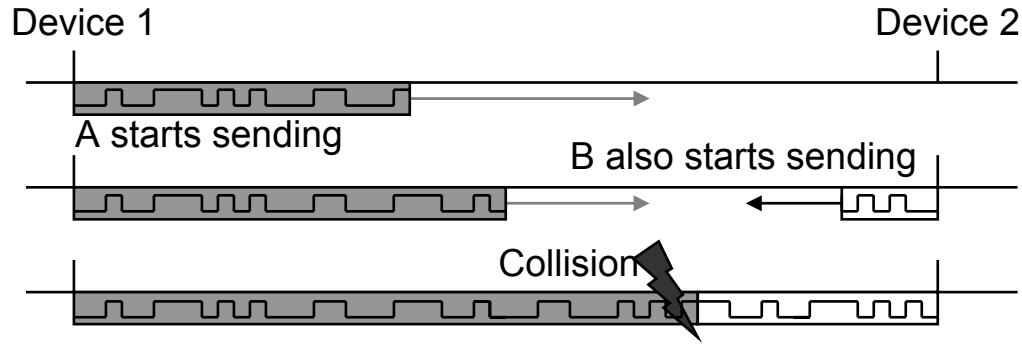

Figure 2 – Ethernet Collision Detection

After a collision occurs, each device will back off for an arbitrary amount of time and try again. If a second collision occurs, the sending device will back off for a longer period of time. There is a limit for the number of re-tries after which a frame will be dropped. This brings us to network topology and when a collision can occur.

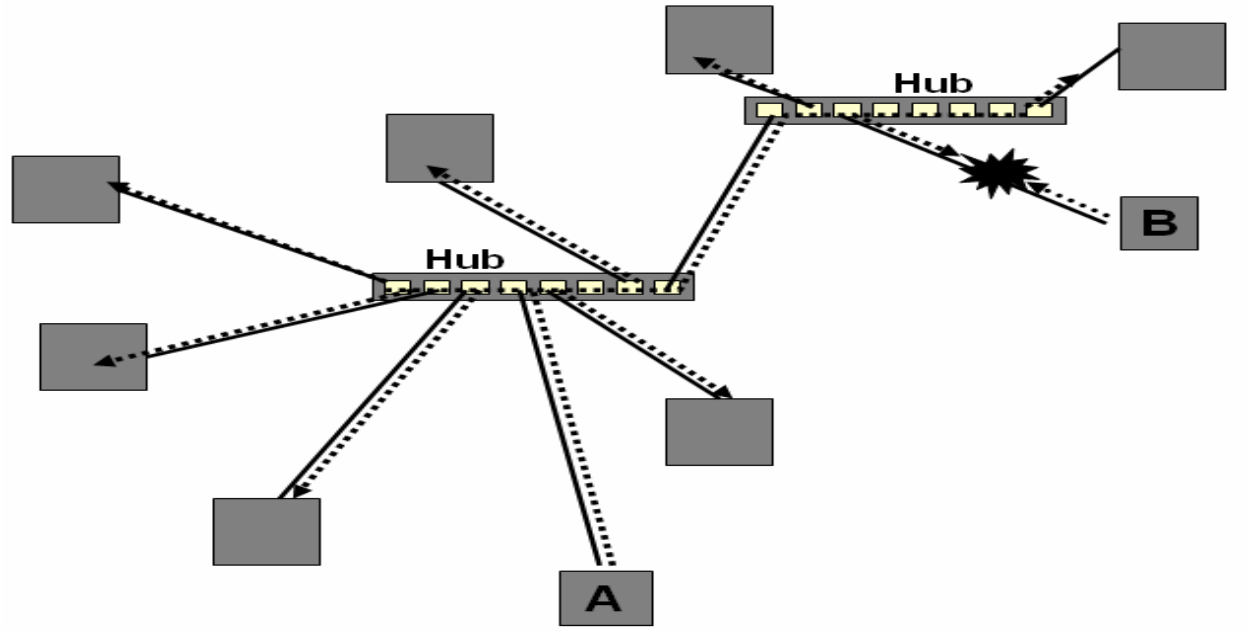

Figure 3 – Ethernet Collision Domain Half Duplex Star Topology

Some applications use hubs. A hub is a very simple physical layer only device - it takes any message coming in on any port and repeats it out all ports. Many networks with hubs are half duplex, which means the same twisted pair is used for sending and receiving. In networks with hubs, collisions are commonplace unless other restrictions are imposed to limit traffic.

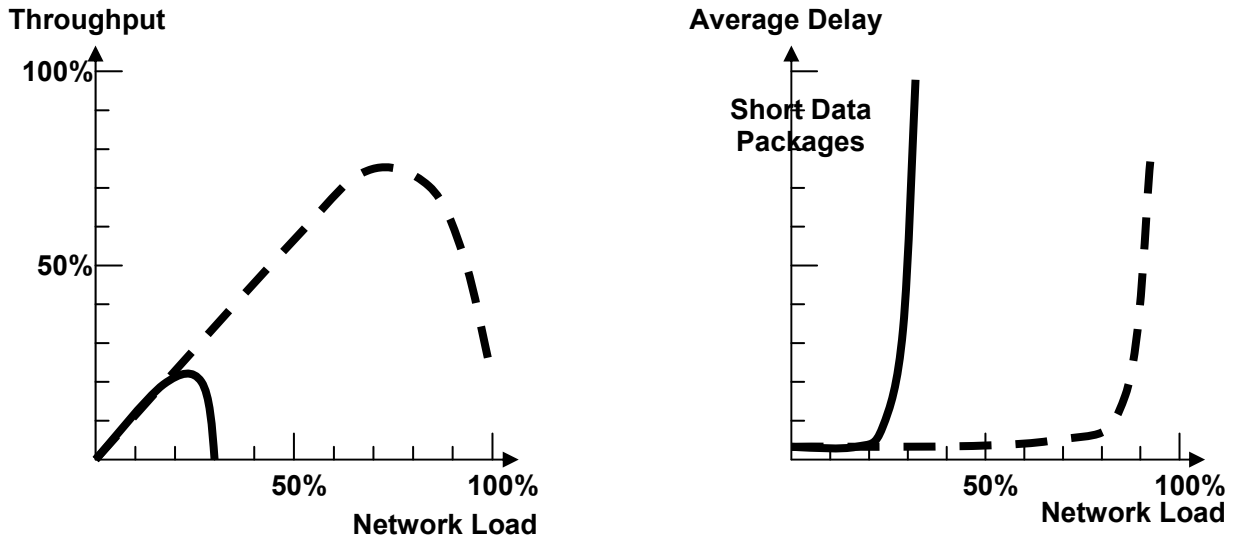

Figure 4 – Throughput in a Collision Domain

If we compare a half duplex Ethernet system (solid) with a Token Ring System IEE802.4 (dashed) we see that it doesn't take a lot of traffic to bring a network to a halt. The network fills with retransmissions and less new messages. To alleviate this situation in networks with a significant amount of traffic, full duplex (one pair of wires for sending one pair for receiving) is used. Switches can also be introduced.

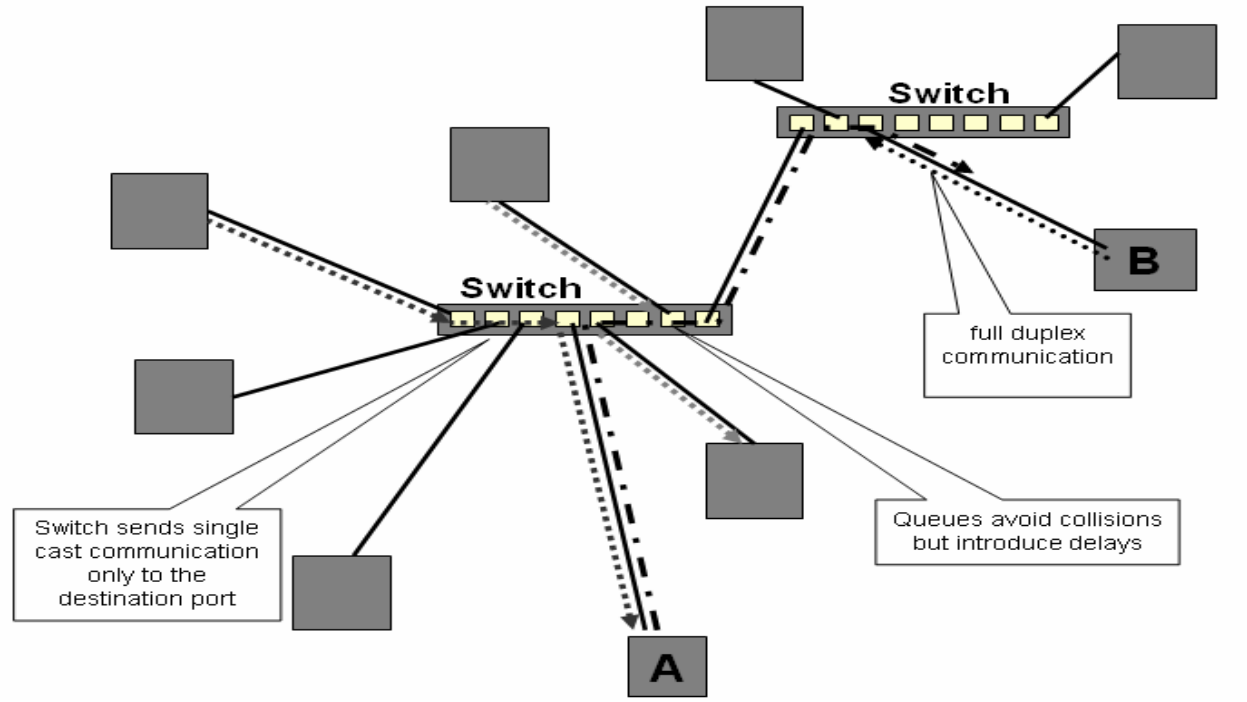

Figure 5 – Switched Full Duplex Ethernet Network

A switch is a layer 2 device as it not only passes the frame along, but actually reads the destination address of the frame and then transmits it only on the port where that MAC address exists. This requires that either the switch "learn" where each device is or the switch must be programmed. A switch also must be able to queue messages for multiple messages to a single port; this can introduce delays in the order of several microseconds. If an Ethernet broadcast message is sent (address FF-FF-FF-FF-FF-FF, then a switch will repeat the message on all ports.

IV. TCP/IP COMMUNICATION

TCIP/IP contains the 2 layers above Ethernet – the Network layer and the Transport Layer. The network layer is the IP or internet protocol layer. This layer is concerned with getting the information all the way from the source to the destination, even over multiple networks. The IP protocol contains the familiar IP address of format www.xxx.yyy.zzz – this allows routing messages over different networks. The IP message adds another 20 bytes of header information:

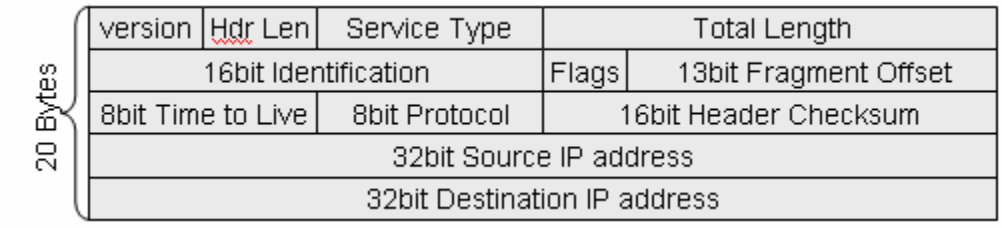

Figure 6 – IP Header

The IP message will then be encapsulated by the layers below it. The major point of the IP protocol is to have the message remain intact and be transmitted over any lower network, such as Ethernet, Token ring, or wireless, etc.

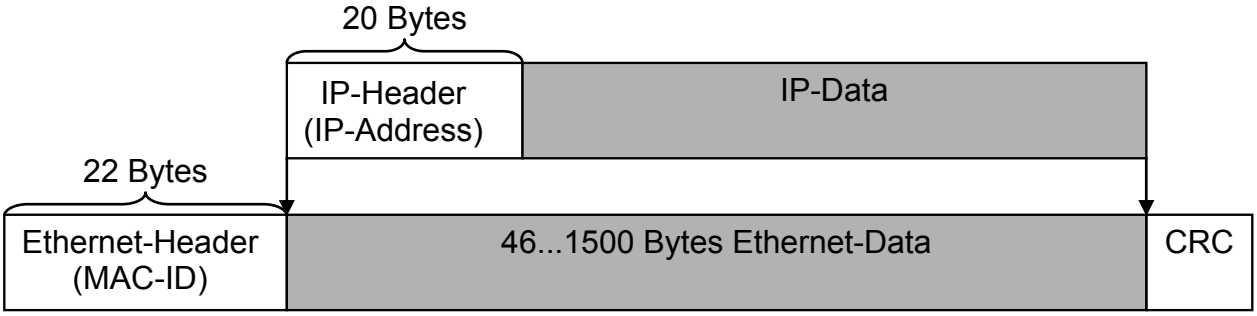

Figure 7 – IP Encapsulation

When an IP message reaches a router in order to be transmitted over another network, the router will unpack the IP message out of the Ethernet frame and, based on the destination IP address, determine which network the message is destined for and pack the message into a new Ethernet message. If the router is also acting as a bridge, it will pack the message into a frame for transmission over a different physical layer, for example, Token Ring. Routers are layer 3 devices as they operate on the data contained in the IP Layer (or Network layer) and they have their own IP addresses. Messages destined for a remote network will be sent through a router. Routers are typically managed, which means they must be configured. Routers will not re-transmit Broadcast IP messages so if Broadcast or Multicast messages are to be used, the routers must be configured correctly.

The TCP layer sits on top of the Internet Protocol and provides another header.

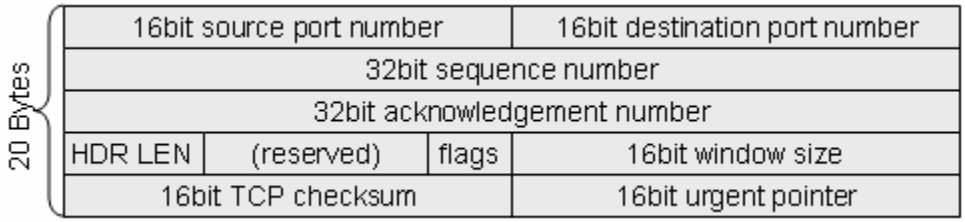

Figure 8 – TCP Header

The TCP system sets up a connection and confirms that messages have made it to their destination. Since a message or file might be very large, each message carries a sequence number so that the

message can be properly reconstructed at the other end. A document would make no sense whatsoever if the words arrived in random order and if certain messages never show up – those messages must be retransmitted. TCP must handle all the re-assembly and message retransmissions. This is an important task. To set up a connection, a three-way handshake of messages is required. The first sends a request (Syn); the second device acknowledges the request and includes a request back to the first device (Syn). The first device acknowledges and then transmission can take place. To terminate the connection, a four way hand shake is used. One device requests a disconnect (fin); the second acknowledges this; the second device then sends a second message requesting a disconnect (fin); the first device then acknowledges and the connection is terminated. Establishing and terminating connections takes some time.

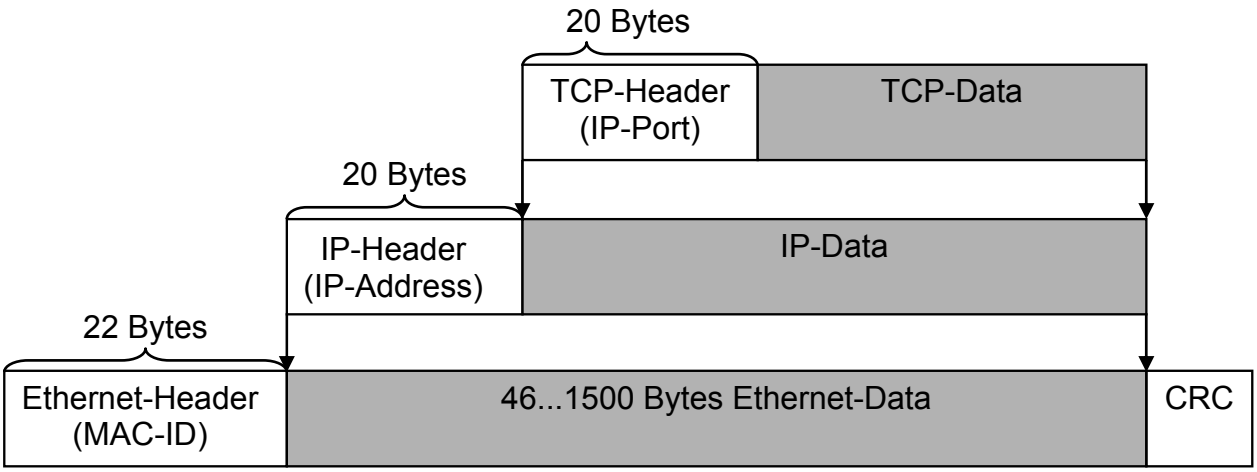

Figure 9– TCP Encapsulation

The TCP messages get encapsulated into the IP messages, which in turn are encapsulated into Ethernet messages. At first glance, this is a very inefficient way to transmit small amounts of data. However, it's not quite as bad as it seems. For example, if a 6 byte message is to be transmitted, 6 bytes get put into a 26 byte TCP message, which is then placed inside a 46 byte IP message and those 46 bytes are put into an Ethernet message. An Ethernet message requires a minimum data length of 46 bytes. Either the message will contain 40 bytes of headers and 6 bytes of data or the Ethernet frame will contain 6 bytes of data and 40 bytes of meaningless padding. It will still take three messages to establish communication and this is pure overhead for TCP/IP.

V. ETHERNET AND INDUSTRIAL AUTOMATION

Industrial communication can be significantly different than normal computer communication, which Ethernet and TCP/IP communication were designed for. If we examine a typical node of remotely located I/O or drive information, we might have as much at 20 bytes of input data and 20 bytes of output data. Very rarely would there be 120 bytes of input data and 120 bytes of output data, which would be 60 analog points all wired back to one location. If we look at a typical drive, it may have 4 bytes of output data for commands and 4 bytes of input data to feedback status and actual position. To send a 72 byte message for 4 bytes of data is not a very efficient use of bandwidth. If we want to talk to more than one address, then we need to send multiple frames and add the 12 byte inter-packet gap.

$$\frac{4 bytes Ouput Data}{72 Bytes Header + 12 Bytes IPG} = 4.76\% Efficiency \quad (1)$$

If the drive can instantaneously respond with its 4 bytes of input data, then we will keep our 4.76% efficiency. More likely, it will take at least 10us for the device to respond and if it has to traverse the entire TCP/IP stack up and down, it will take longer than that. Then we have the following.

$$\left(\frac{100E6Bit/s*10us}{8bits/byte}\right)=125BytesIdleTimeToExecuteTCPIPStack \quad (2)$$

$$\frac{8BytesData}{84Bytes+125BytesIdle+84Bytes}=2.73\%Efficiency \quad (3)$$

When compared to Profibus, which is considered to have a large overhead with 12 bytes, it's obvious that Ethernet is really not suited for small amounts of data. Ethernet is fast at 100MBit, but if only 4.76% of the data is transmitted, then it's really only a 4.76Mbit/s network. Some protocols allow for multiple devices to be addressed with one message, thus increasing the efficiency of the network. In the case of EtherCAT, that efficiency can be as high as 80%.

A very good reason to use Ethernet is cost. Thousands of miles of Ethernet cable are already installed and 1000 ft of cable can be bought online for less than $50 and 50 connectors goes for about $16 if unassembled. Profibus connectors can be $30 per connector or more and the cable can be as high as $1.20 per foot. So obviously, hardware costs are a significant factor. A DeviceNet scanner card will run anywhere from a few hundred to over a thousand dollars, whereas an industrialized Ethernet card will go for about $70. Not every Ethernet protocol is the same, some require special master cards and special switches, increasing the price to levels comparable to standard fieldbuses.

Before choosing a protocol, it is very important to consider the required response time and determinism of the application. Is the desire to use Ethernet as a replacement for a system that is running a standard fieldbus like CanOpen, DeviceNet, Profibus, Interbus-S, ControlNet, and SERCOS, or is the desire to add a higher level of communication between areas of automation? If the requirements of the application are not determined correctly and the wrong implementation is chosen, a great deal of time will be spent trying to get a network to behave in a way it was not built for. The various published values for the different Ethernet protocols range from ~50ms to 50us with jitter that can be ~1ms to ~100ns. To achieve the very low cycle times and high level of determinism, most systems eliminate the TCP/IP stack and directly put frames onto the Ethernet. This means that you basically need a segregated Ethernet network for control. The networks that base their communication on TCP/IP either add directly into the existing stack or replace the standard TCP/IP stack with one that is dedicated to that particular protocol. These are typically slower, but the implementations can co-exist with other network traffic.

VI. Modbus TCP

Modbus TCP is essentially Modbus messaging over the TCP/IP protocol; it's fairly well known and easy to implement. The initial connection time can be on the order of a few seconds, but once established, the responses are in the order of milliseconds. The drawback is that the entire TCP stack must be processed on every message. This requires a slave device to have full TCP/IP capabilities, not just Ethernet, but requires no special hardware.[3]

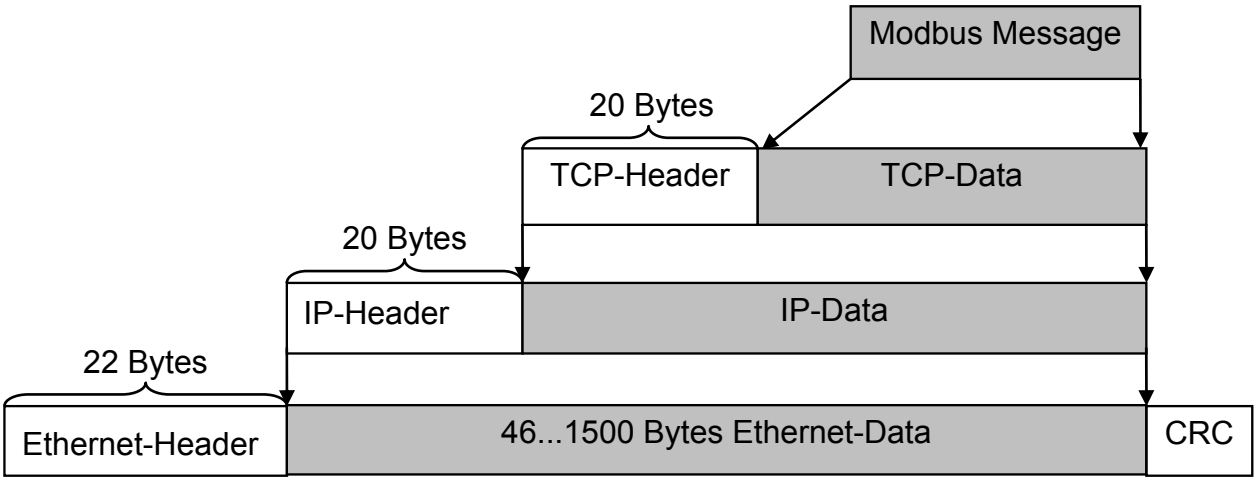

Figure 10– Modbus TCP Encapsulation

VII. PROFINET

PROFINET comes in a few varieties. Standard PROFINET uses all standard hardware and the full TCP/IP Stack. There is PROFINET Real-Time, which provides cycle times on the order of 5 to 10ms. In this case, the TCP/IP stack is removed and PROFINET RT or Real-Time (registered Ethertype 0x8892) frames are put directly onto Ethernet. The third level is PROFINET IRT or Isochronous Real Time. Cycle times range from 250us to 1ms and a jitter of less than 1us. In this case, special hardware is used to time slice the Ethernet network and provide synchronization messages.[4]

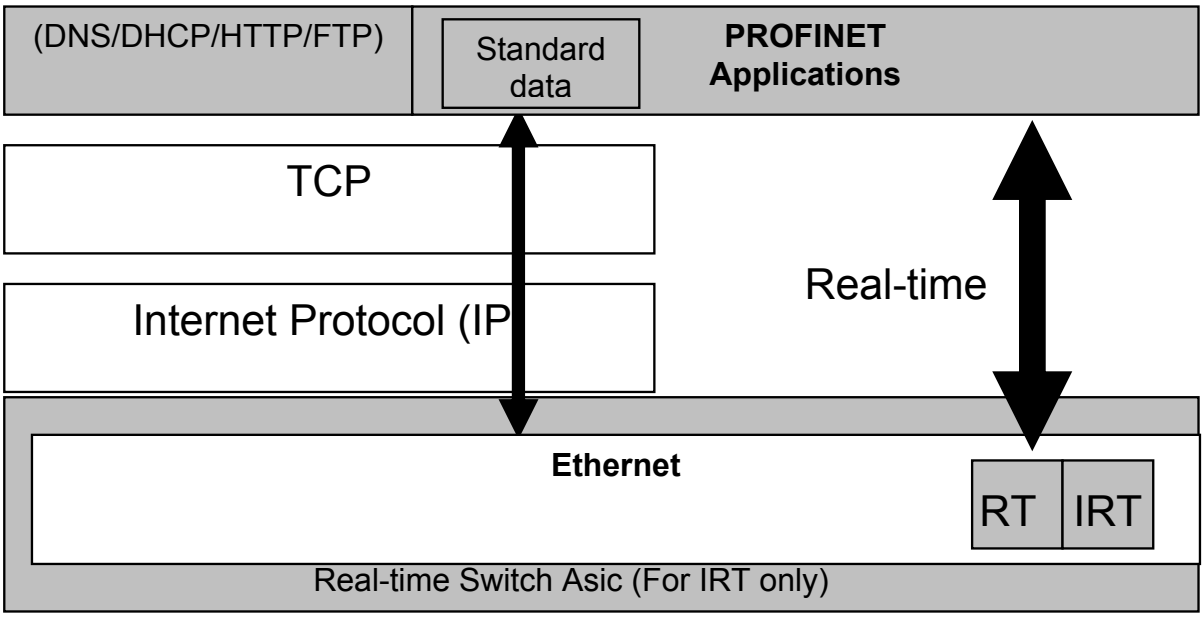

Figure 11– PROFINET Protocol Stack

VII. ETHERNET Powerlink

ETHERNET Powerlink comes in two varieties – open and protected. In open mode, any ETHERNET Powerlink device can be connected to any network, however, cycle times are in the millisecond range and jitter is around 10us. In protected mode, a Gateway is placed between the isochronous ETHERNET

Powerlink and the rest of the network and hubs must be used in the protected area. In protected mode, the Ethernet traffic is time sliced and up to 8 nodes can be accessed in 200us. Longer cycle times can achieve more nodes and data. ETHERNET Powerlink allows for different cycle times for different devices on the same network.[5]

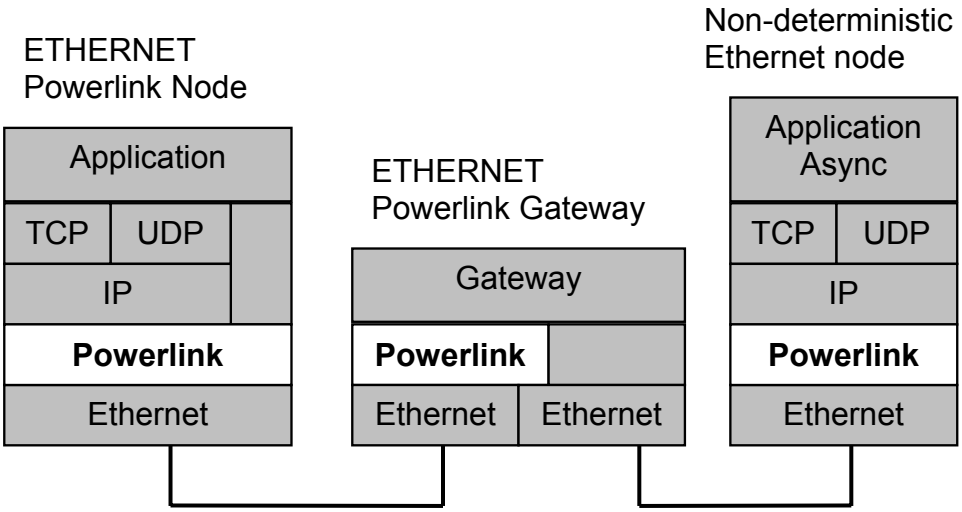

Figure 12– ETHERNET Powerlink Protocol Stack

VIII. Ethernet/IP

Ethernet/IP is designed to operate on standard, off-the-shelf hardware. It is designed for controller-to-controller communication with larger amounts of data in slower time frames. It is designed to encapsulate DeviceNet messages. Ethernet/IP can be synchronized, but it is limited by the worst case communication scenario, so updates can be rather long - on the order of 50ms - non-synchronized updates are much quicker. Ethernet/IP does support distributed clocks for synchronous functions and utilizes Multicast messages. When using Multicast messages, Layer 3 switches or routers must be used and managed carefully. [6]

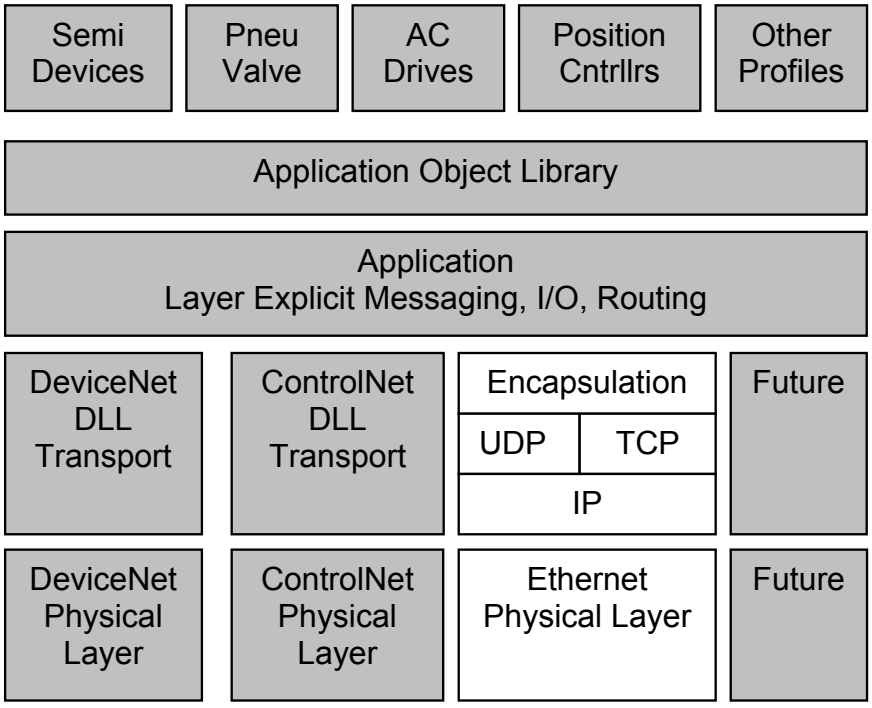

Figure 13– ETHERNET Powerlink Protocol Stack

IX. EtherCAT

EtherCAT is designed to operate in a fairly simple, high-speed manner. It is a solely master-slave system with the allowances for up to 65,535 slave devices. It operates on a principle different to other systems, whereby a message is sent from the master and, as it passes through each device, that same message is manipulated and returned back through all devices to the master. Inputs inject their data into the EtherCAT message as the message passes through the device and outputs read their settings out of the message. After the message has passed through the device and the device has verified the CRC the device, it will set its output accordingly. Since multiple devices can be addressed in one Ethernet frame, the efficiency of each Ethernet frame is greatly increased. Typical cycle times for EtherCAT include 100us for 100 devices, each with 6 bytes of data in and 6 bytes of data out. Through distributed clocks, outputs can be synchronized within 20nanoseconds of one another.[7]

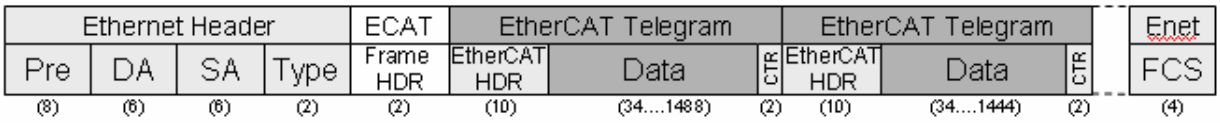

Figure 14– EtherCAT Ethernet Frame Powerlink Protocol Stack

IX. Conclusions

This paper is a brief overview of some of the common varieties of Ethernet protocols, what Ethernet is and what it isn't. Ethernet is not the magic solution many had anticipated. There are many protocols that don't readily talk to each other. The most important consideration before choosing an Ethernet protocol is to fully understand the needs of the application and from there, select a protocol that meets those needs. Forcing a protocol upon an application it is not suited for can result in a significant amount of unnecessary effort to make it work as well as lower than expected performance.

X. References

[1] Computer Networks Third Edition Andrew S. Tanenbaum 1996 Pretice Hall

[2] 802.3-2002 IEEE Standard for Information technology Telecommunications and information exchange between systems Local and metropolitan area networks Specific requirements Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications

[3] Modbus Application Protocol V1.1a – June 2004

[4] Profibus International PROFINET Real-Time Communication Slide Set RT_IRT Communication engl 041233.pdf

[5] ETHERNET Powerlink Standardization Group Presentation Feb 8 2005

[6] Introducing Ethernet/IP by Nick Jones and William H. Moss 6th Annual Meeting

[7] Introduction To EtherCAT presentation by Martin Rostan