IoT Tethered Protocols

Abstract:

The advancement of IIoT 4.0 for smart factories, cities and buildings ushers in many exciting possibilities for improved automation and capabilities. IoT devices are unlocking the great potential for improved efficiency and improved user experiences. However, there are many different IoT protocols, network topologies and frequency bands, making IoT an intranet of things and not an internet of things. Therefore, in order to determine which IoT technology to use in solving your use case and future proofing your investment, an understanding of the IoT ecosystem is needed. This is the second in a series of papers describing the different protocols, topologies and frequency bands used in IoT deployments focusing on IoT Tethered Protocols.

Clint Smith, P.E.

csmith@nextgconnect.com

1.Overview:

The internet of things (IoT) used in industrial, military, commercial, enterprise or consumer devices is anything but a simple topic. The vastness of the types of IoT devices, their operating systems, capabilities, methods of communication, as well as initial and recurring cost in selecting the proper device that meets the use case requirements makes it challenging.

The advancement of Industrial IoT (IIoT) 4.0 for smart factories, cities and buildings ushers in many exciting possibilities for improved automation. IoT facilitates the exchange of data between the physical world and a user or computer application. IoT devices collect and create massive amounts of data as well as exchange that data with other devices to enable actions to take place based on the data and policy rule sets defined.

IoT devices are found in many places. Some are currently deployed in industrial, transportation, health, smart cities, smart buildings, energy utilities, security and consumer products. However, the IoT industry is fragmented based on the plethora of devices and protocols being utilized. The fragmentation of the IoT industry has created an Intranet of things and not an Internet of things.

Ideally everything should utilize the internet protocol (IP), be open source and use REST commands with a common API. Unfortunately, reality is quite different and there is no single answer and sometimes your decision was made based on legacy platforms already in place.

There are numerous sources of information available regarding IoT devices from the internet and vendors all pushing a particular solution. In fact, the information is so vast and dispersed that making a detailed informed decision is beyond a challenging task.

Additionally, more and more protocols are being added to the IoT ecosystem. It seems that every three-letter acronym is getting an additional letter added to it reflecting its entrance into the IoT world.

Therefore, if you are embarking or have embarked on an IoT path you need to be aware of the various options to pick from. Although the IoT decision process involves many steps or decision points the first thing that you need to do is determine your objective and use case. The objective determination is more of a business decision than a technical one because the business decision should be driving the technical decision. Specifically, defining what your use case or cases are that you need to solve is critical in the IoT selection process.

Choosing a particular protocol also impacts the efficiency and performance of the IoT solution and with numerous diverse protocols out there for IoT, it is hard for one to decide which ones to use. To help in the decision process this paper gives an overview of the protocols available in IoT world. `

2.Index

Contents

1. Overview: 1
2. Index 2
3. IoT Introduction 3
3.1 Protocol types 3
3.2 IoT Device Protocols 4
3.3 IoT Network Topologies 6
3.4 Connectivity 8
3.5 US IoT Spectrum 9
3.6 IoT Decision 10
4. Tethered (wired) 11
4.1 Ethernet 11
4.2 Internet Protocol (IP) 13
4.3 Barcodes 15
4.4 RS-232 17
4.5 RS422 17
4.6 RS-485 18
4.7 4-20mA 19
4.8 SPI 22
4.9 I2C 23
4.10 HART 25
4.11 Modbus 25
4.12 LonTalk 27
4.13 Fieldbus 28
4.14 ARCNET 29
4.15 PROFINet 29
4.16 Foundation Fieldbus 31
4.17 Profibus 33
4.18 INTERBUS 34
4.19 BACnet 35

4.20 LonWorks 38
4.21 CEBus 39
4.22 X10 39
4.23 PLC 41
4.24 G.hn 42
4.25 Tethered Device Protocols 44

3.IoT Introduction

IoT whether it is for industrial process control, building automation, smart cities, utilities, smart homes or other things have some common items that need to be understood.

The IoT decisions regarding technology can be boiled down into three main categories. The first category is the protocol that is used. The protocol selected directly determines what functions the IoT device can perform. The second category is the network topology that you want or need to utilize to realize the IoT solution. There are numerous types of topologies used and often it is a hybrid topology that is implemented. The third category is the transport or connectivity layer. The connectivity layer can be tethered or wireless.

Regardless the solution picked for the IoT functionality have tradeoffs. The tradeoffs are best depicted in figure 3.1.

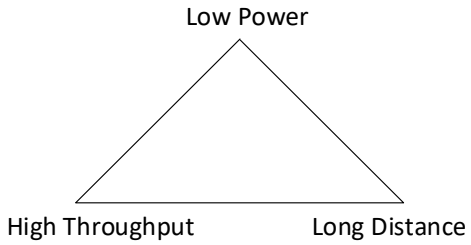

Figure 3.1: IoT Functionality Tradeoff

3.1 Protocol types

Figure 3.2 is a high-level depiction of an IoT sensor that communicates with a middleware platform to reach the application as part of the IoT ecosystem. The three different protocol classifications shown in Figure 1 the device, communication and application protocols. The true demarcation of where one protocol function begins and ends is dependent upon what your particular use case is.

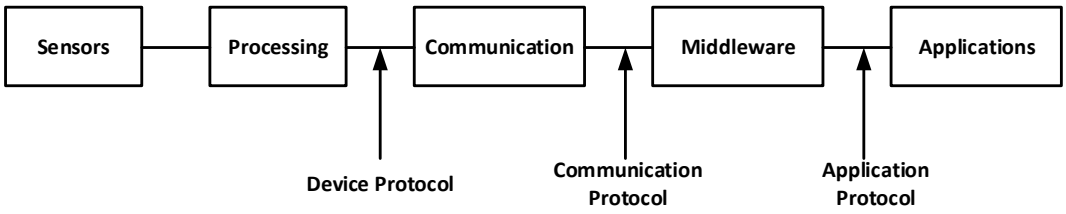

Figure 3.2: IoT Protocol Decision locations

There are numerous data link or device protocols used for IoT. Each of the data link protocols is designed to solve a particular problem and there is currently no single data link protocol that solves everything.

All data links can be classified as either tethered or wireless. Whether the selected data link protocol is tethered or wireless, it should match the objective for the problem you are trying to solve. If you need mobility, then a data link protocol providing mobility should be looked into. The mobility protocols available for use involve both licensed and license exempt spectrum usages. Most, if not all, licensed spectrum data link protocols have monthly recurring costs and these need to be understood. Also, other data link protocols are closed systems with one hardware vendor, or the protocol is not published.

It is necessary to look into the details of any device and protocol that you are considering. For instance, some data link protocols are not well suited for software updates, patches or configuration changes. Others utilize a mesh or star topology as part of the data link protocol where coverage and potential throughput need to be understood.

Then there are different frequencies for the wireless data link protocols. Some are meant for short range low data speeds using sub GHz frequencies. Others are meant for short range but higher data speeds using ISM frequencies. Also, there are protocols offering long range and low data rates and others with long range and high data rates.

As you can quickly gather there are numerous issues that should be thought about and weighed regarding their technical and business merits. It's important to always remember that once you begin using a data link protocol changing to another will prove difficult and time consuming which both equate to money.

When determining which data link protocol to utilize keep in mind that there are numerous IoT devices and protocols which have not been commercially successful with limited roll outs resulting in stranded systems.

3.2 IoT Device Protocols

Often the IoT device, communication and application protocols are intermixed in articles and some of the protocols are defined across multiple layers of OSI model making it difficult to truly separate its position in OSI layer. Additionally, some of the IoT Device protocol are part of a closed system limiting the use to a particular protocol and or vendor.

Therefore Table 1 is a list of most of the IoT Device Protocols that are in use presently. Table 1 has four categories in it. The first category is the tethered group which includes protocols associated primarily with wired connections. The second category is listed as wireless and while the remaining two are also wireless,

this category fundamentally covers the device protocols using license exempt frequencies. The third category involves cellular which includes wireless broadband. The fourth category lists the primary cellular IoT technologies that are used.

If you have not heard of some of the protocols listed in table 1 you are not alone. If you do not know a particular protocol that is listed, it would be good to possibly investigate it a little more. As always, the devil is in the details and the table provided is meant to help start culling the options you are trying to figure out for your use case.

IoT Device Protocols						
Tethered	802.3 Ethernet	IPv4/IPv6	BarCode	RS-232	RS-422	RS-485
	4-20mA	SPI	I2C	HART	Modbus	LonTalk
	Fieldbus	ARCNet	ProfiNet	Foundation Fieldbus	Profibus	Interbus
	BACNet	LonWorks	CEBus	X10	PLC	G.hn
Wireless	WiFi (WiFi6)	BLE	Bluetooth	ZigBee	RFID	NFC
	6LoWPAN (6Lo)	802.15.4	Zwave	Sigfox	LoRa	LwM2M
	Wireless HART	DASH7 (DA7)	RuBee	ANT	EnOcean	Weightless P
	ISA100	WiFi HaLow 802.ah	Ingenu RPMA	Telensa	Nwave	Neul
	NB-Fi	Wi SUN/ 802.15.4g				
Cellular	GSM	CDMA	UMTS	LTE	WiMax	Satellite
Cellular IoT	Cat 4	Cat 0	LTE Cat M1	LTE NB IoT	EC-GSM-IoT	DECT/ULE-ultra low energy

Note: some protocols include higher OSI layers

Table 3.1: IoT Device Protocols

To complicate things some of the protocols listed in Table 1 include some higher OSI layers. Furthermore, some of the IoT device protocols are part of a closed system limiting the use to a particular protocol and or vendor even though they claim to be an open standard.

Table 3.2 is a list of some of the IoT communication (session) protocols. Again, as with the device protocol list some of the protocols listed in Table 2 span multiple OSI layers.

IoT Communication Protocols					
XMPP	HTTP/REST	SNMP	SMS	HTTP/2	SOAP
CoAP	MQTT	SMQTT	IEEE 1451	AMQP	LLAP
Hart-ip	IBS	DNP3	IEC61850	CANopen	DDS
IEC 60870	IEC 61968	IEC 61968	Multispeak	SSI	ZeroMQ
Websocket	IEC61334	UPnP	IoTivity	DeviceNet	IEEE 1905.1
BACNet	Modbus	LonWorks	Sinec H1	MTConnect	Continua HDP
IEEE P2413	Weave				

Note: some protocols include higher OSI layers

Table 3.2: IoT Communication Protocols

In addition to device protocols and session protocols there are application protocols shown in Table 3.3. Table 3.3 lists some of the more prevalent IoT application protocols that are present today.

Application Protocols				
Juniper Mist	Haystack	AllJoyn	Thread	Tingsquare Mist
EEBus Spine	Dotdot 1.0	IoTivity	ONVIF	KNX
HomeKit	Symphony Link	MyriaNed	Insteon	Senet
IoLink	Home Pod	AWS IoT	Google IoT	Azure IoT
Home Connect	SmartThings Hub	Amazon Echo	Google Home	Nest

Table 3.3: IoT Application Protocols

Besides device, communication and application protocols there are numerous dashboards and OSI layer 7 applications that are available for IoT systems. These dashboard systems are how the user or device manager views their IoT world from. Therefore, when making a choice of which IoT device or devices to utilize, you need to include in the decision selection process how you will interface with the devices and system through a management layer. A key consideration about which dashboard or application you select needs to be based on what your use case is.

3.3 IoT Network Topologies

Each IoT device protocol and higher layers has a particular network topology it was initially designed for. When investigating an IoT solution there are numerous types of network topologies that can be used and each with its own set of unique benefits.

Figure 3.3 is a high-level depiction of the various network topologies. Some IoT technologies and implementations scenarios utilize a hybrid approach where there are multiple topologies used in different branches of the system.

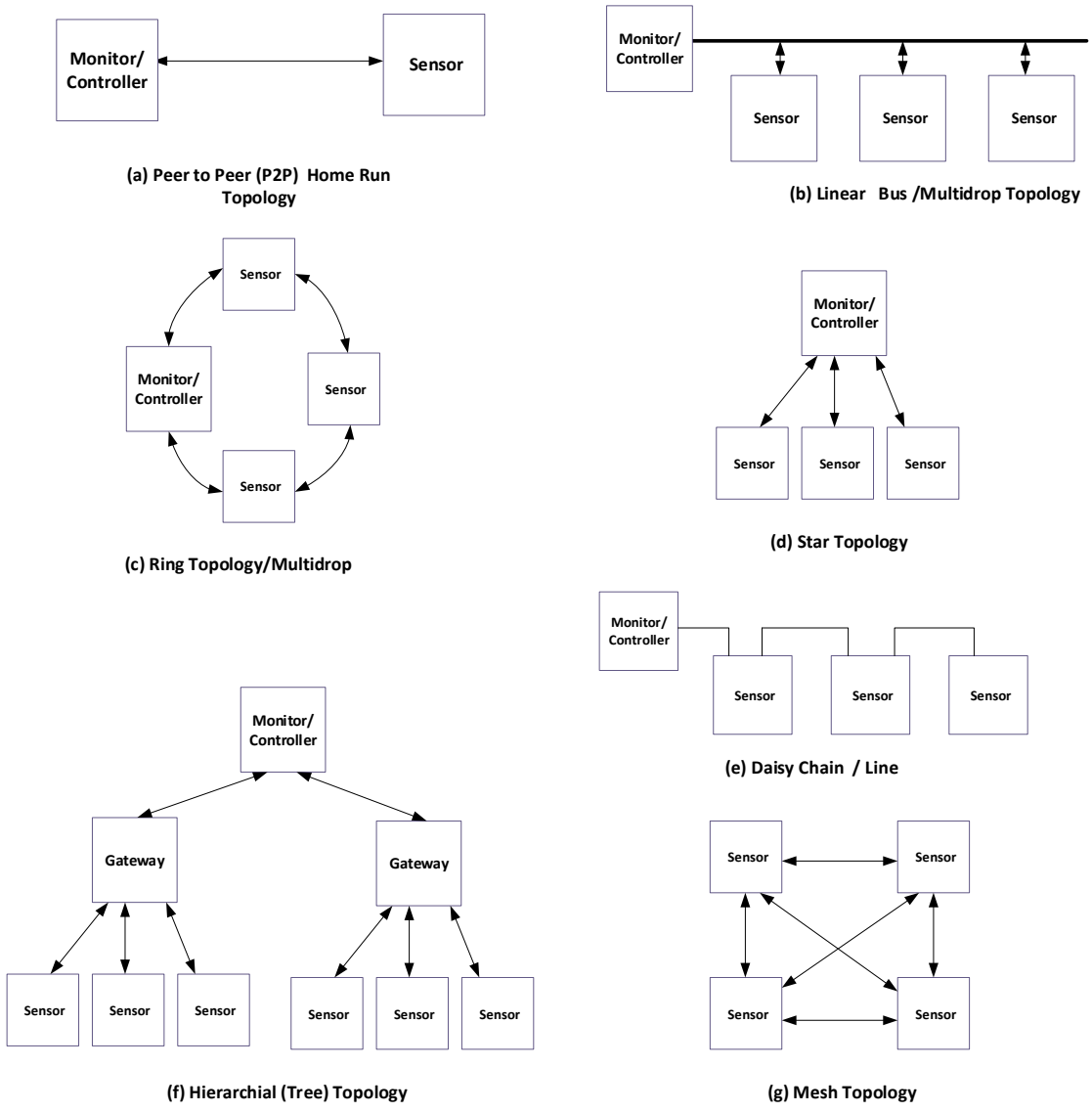

Figure 3.3: Network Topologies

In addition to the network topologies shown in figure 2 there is another topology which involves clouds. The use of cloud network as part of your solution may or may not be appropriate depending on your use case. However, it is likely some part of your IoT solution will incorporate the use of a cloud solution.

As you would expect there are numerous sources for clouds available. However, keep in mind that not all cloud solutions are the same and that interoperability and real portability needs to be considered otherwise you will be locked into one service provider or solution.

Regardless a cloud environment for IoT can be looked at in three layers: cloud, fog and mist and they are shown in figure 3.4. In figure 3.4 the main layer is the cloud and it is a more traditional network model using remote servers running in a virtual environment instead of being run locally. The cloud service

provides the ability to perform heavy computing, storage, and analytics. Devices can connect to the cloud directly or via an intermediary like a fog or mist environment.

A fog environment shown in figure 3 is meant to extend cloud computing closer to the edge of the network. This has many advantages for an IoT environment reducing latency, performing less intensive computing functions and minimizing the amount of data that is sent to the cloud which is not needed. Fog environments are more geographically dispersed than cloud networks. The fog environment can be thought of as an intermediate level cloud.

The next lower level cloud environment shown in figure 3.4 is called a mist or rather edge computing. Mist computing is meant to provide edge computing for the network. Devices making up a mist environment perform smaller functions that do not need to be elevated to the fog or cloud environments. Many mist clouds can reside within a fog environment. The mist environment is a true distributed computing environment.

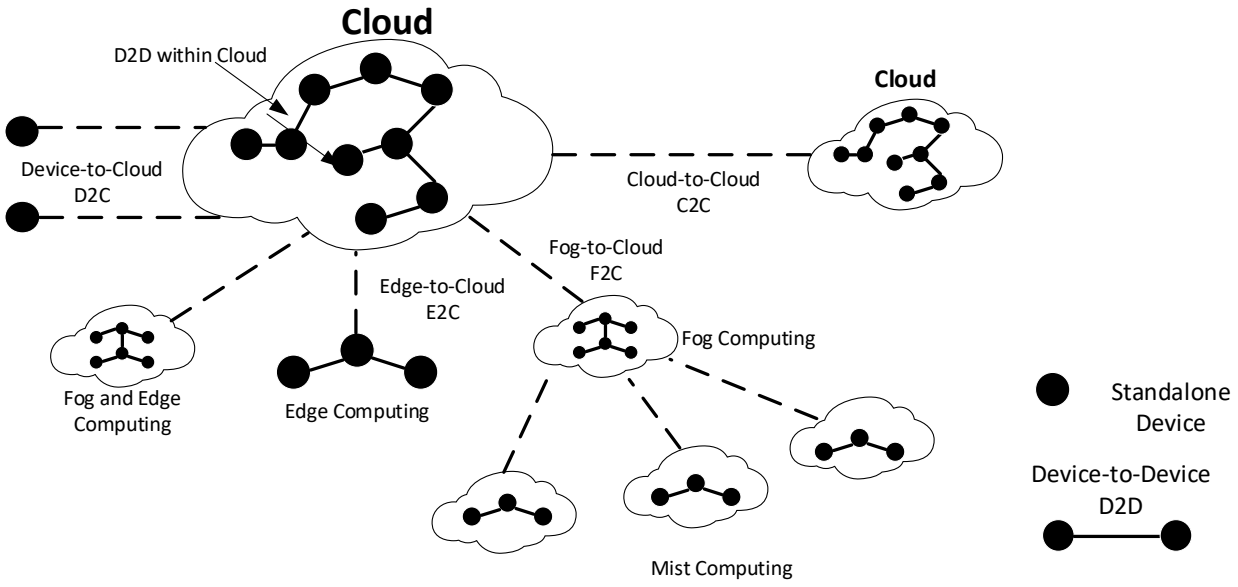

Figure 3.4: Cloud Topologies

3.4 Connectivity

There are many device protocols used for IoT each with its own unique advantages and some disadvantage. IoT devices However all IoT device protocols either utilize a tethered or untethered method.

A tethered method involves a wired connection. A wired connection can utilize two or more wires where the simpler control systems utilize two wire. A basic two wire system can be a simple relay contact for a door. Tethered connections have been predominantly used in Industrial IoT and building automation use cases. Tethered connections however are making way for untethered or wireless connections which offer more flexibility and features than many of the traditional wired connections.

Tethered connections are thought of as more secure however if the upstream controller is connected to the internet or other sensors have wireless capability then a comprehensive security review should be part of your regular audit process.

Untethered is another name for wireless. The vast majority of IoT protocols are wireless. And wireless devices operate either in license exempt bands link WiFi or licensed bands like cellular. However, there are multiple frequency bands that IoT devices utilize. Which frequency band used helps determine the potential distance or range that you can have the IoT device installed from its controller or upstream device and other IoT devices.

3.5 US IoT Spectrum

Figure 3.5 is a brief overview of the US spectrum which applies to wireless IoT devices. In reviewing figure 4 the different frequency bands which are available for IoT use are vast. In addition, all the frequency bands with the exception of the cellular spectrum are license free. IoT devices however are not solely limited to the frequencies listed in figure 4.

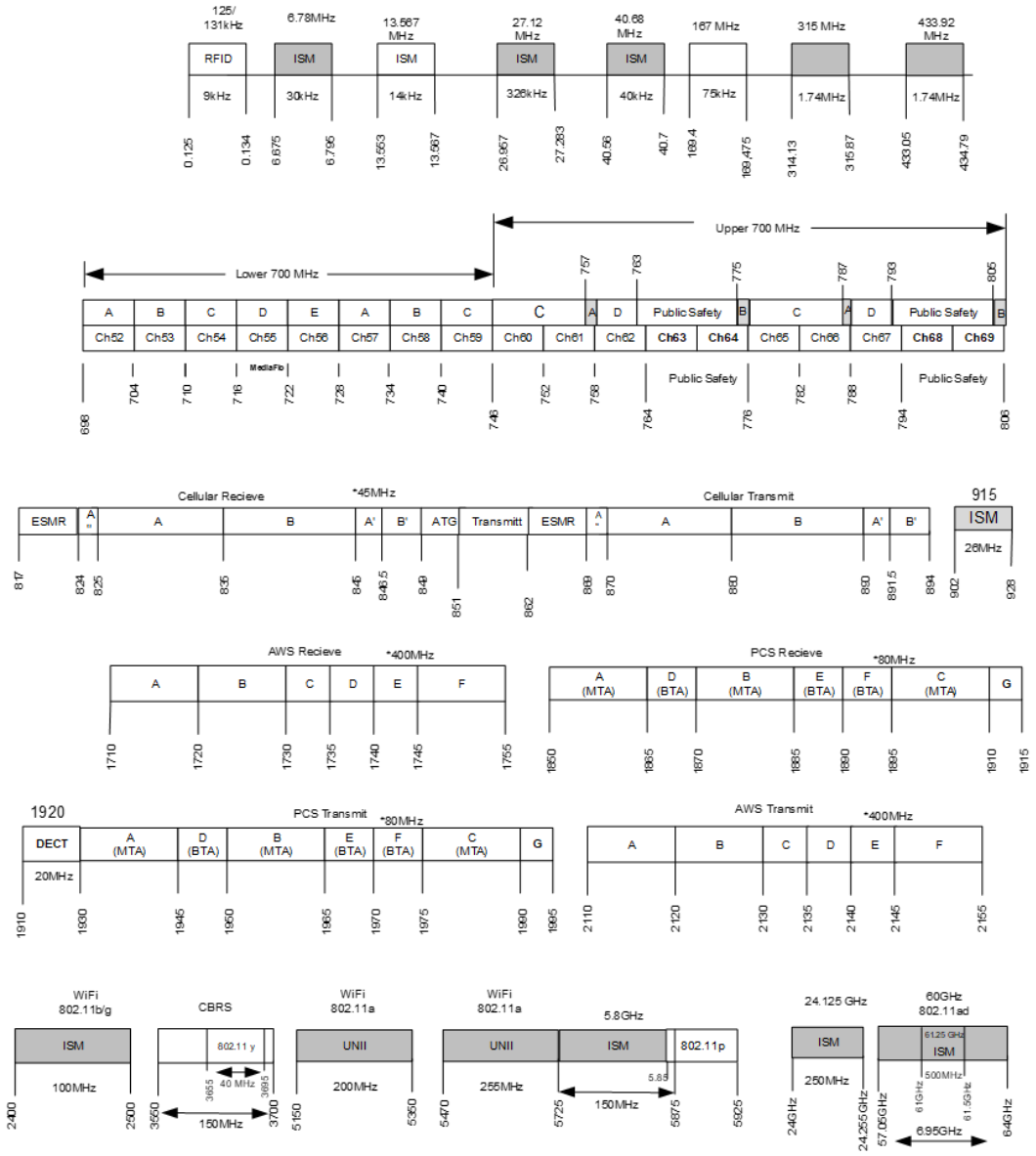

Figure 3.6: US IoT Spectrum

3.6 IoT Decision

While technology is an important component of the IoT decision there are other issues that need to be addressed when selecting an IoT solution. These issues are covered in the IoT 7 Critical Musts found at www.nextgconnect.com/iot-documents.

The seven areas that you should address as part of your IoT solution and they are:

1. Objective/Purpose
2. Security (cyber/physical)
3. Data Acquisition/Functions
4. Standards and Compliance Regulations
5. Business (CapEx/OpEx/Revenue)
6. Interface /User Experience
7. Technology

The order of the critical musts is not as important and making sure they are all addressed. However, I would keep the objective/purpose as the first since everything else is determined from that point onward.

Unfortunately, the device protocol is tightly coupled with the particular IoT hardware. Many IoT device protocols are open standards allowing many vendors to offer products that support that protocol. However, there are many proprietary protocols also called closed systems which are available leading to a limited selection of vendors to pick from. Whether you choose an open source or closed source will be primarily driven from your use case.

Since there are many choices and options to consider a deeper understanding of these protocols, their configuration options and security needs to be understood in order to select the best protocol or protocols to used based on your use case requirements.

4. Tethered (wired)

There are numerous tethered, wired, device protocols used in IoT. Each has their own advantages and disadvantages. This section ~~on~~ provides a brief description of various protocols, standards and technologies that are used in a wired environment. The list while long is not exhaustive.

What follows is a brief description of many of the device protocols used in IoT which are tethered, i.e. using wires. Along with the brief description of the device protocol a URL link or links are provided to enable you to do a deeper dive in the device protocol itself.

The description this is included for each IoT data link protocol is brief and is meant to provide a better idea of which data link protocols may best be suited for your particular use case(s). There is a table at the end of this section that has all the tethered protocols in this article listed for ease of comparison.

4.1 Ethernet

IEEE 802.3 is otherwise known as the Ethernet standard which describes both hardware as well as the data transmission itself. Ethernet is both layer 1 and layer 2 in the OSI model. Ethernet is commonly used for local area networks (LANs) and is not the internet.

Ethernet is a protocol for wired local area networks (LANs) enabling devices to communicate with each other. Ethernet supports the network architectures defined in IEEE 802.1. There can be multiple network topologies in which the nodes can be connected. 802.3 defines the nodes, the topologies in which nodes can be connected and the media using which the nodes can be connected in a specific topology.

Nodes can be computers, hubs, switches, routers and other devices like printers. Each of these nodes are points to and from which the communications take place. The physical connection to the wired network is usually provided by an RJ45 ethernet connector.

In addition, the operating systems like Windows, Apple iOS and Linux, have ethernet capability incorporated into the basic software. This means that additional drivers do not need to be loaded in to have the functionality.

There are several network topologies that can be used for ethernet connectivity. Based on the requirements, it can be a Point-to-point, Coaxial Bus or Star topology. Point-to-point is the simplest configuration with a network cable linking two network nodes. Coaxial bus topology is where the systems used a coaxial cable and all the network nodes are along the length of the cable. Star topology is the widely used where multiple point-to-point connections from a central hub extend out to a series of routers or switches that divert the data to the end nodes connected to them.

Coaxial cables, twisted pair cables and fiber optic cables are the media used to propagate signals within the Ethernet systems. The type of the cable determines the speed of the data transmissions. Most common types of the twisted pair Ethernet cables in use are cat5, cat5e, Cat 6, 6a, 7 and Cat 8 and the RJ45 connectors.

Table 4.1.1 is a high-level comparison of different cables used with ethernet connections. The speeds listed for each category is defined in TIA-EIA 568. However, the speeds can and will be different than advertised due to network topology and policies put in place.

Category	Shielding	Max Speed Mbps (100 meters)	Max Bandwidth (MHz)
Cat 3	Unshielded	10	16
Cat 5	Unshielded	10/100	100
Cat 5e	Unshielded	1000	100
Cat 6	Shielded/ Unshielded	1000	>250
Cat 6a	Shielded	10000	500
Cat 7	Shielded	10000	600

Table 4.1.1 Category comparisons

802.3 uses a mechanism called CSMA/CD (Carrier Sense Multiple Access /Collision Detection) where a sender listens to see if another carrier is present on the channel and transmits data only when the channel is free. If a carrier is detected, the transmitter waits for a random interval of time before trying to resend the frame.

Metro Ethernet and Carrier Ethernet are two other versions of **e**thernet used for higher performance in large metropolitan areas. Often businesses use Metropolitan Ethernet to connect multiple locations of the business to a network via an Ethernet private line. This gives some significant advantages over using the public networks in terms of speed, security and cost. Carrier Ethernet or Carrier grade Ethernet is typically used to provide communication point to point communication between two points or sites or to provide links for local area networks, etc. Carrier Ethernet has developed to include a number of other services and it enables long-distance data sharing among businesses and other organizations.

The following URL links can be used to do a deeper dive.

- www.wikipedia.org/wiki/Ethernet_frame
- www.standards.ieee.org/standard/802_3-2000.html

4.2 Internet Protocol (IP)

The internet protocol (IP) is an essential protocol used today in fostering machine communications over different kinds of networks. IP is usually associated with the Internet however it is also used with intranets. The IP protocol is used in both wired and wireless communications. IP is the most prolific communication protocol that is used.

There are essentially two IP protocol versions and they are Internet Protocol version 4 (IPv4) and Internet protocol version 6 (IPv6). Both IPv4 and IPv6 are connectionless protocols used in packet networks. IPv4 is described under RFC 791 and IPv6 is described under RFC 2460.

IPv6 is meant to replace IPv4. However, the embedded base of IPv4 devices ensures that IPv4 will be used for quite some time going forward. Both of these IP protocols are very useful, and they are currently coexisting together. Both IPv4 and IPv6 support Transmission Control Protocol (TCP) as well as User Datagram Protocol (UDP).

IPv4 and IPv6 while both supporting TCP and UDP are not the same and there are some fundamental differences between them. One of the main differences is the amount of IP addresses that can be assigned under IPv6 as compared to IPv4. In IPv4 there are about 4.3 billion addresses as compared to IPv6 which has 3.4×10^{38} addresses.

To facilitate more addresses the IPv6 address is now 128 bits long as compared to IPv4 which was 32 bits in length. In IPv4 the subnet is 32 bits made up of four 8-bit octets.

IPv4 address example:

IP address: 192.168.1.104

Subnet Mask: 255.255.255.0

Gateway Address: 192.168.1.1

Where the network address is 192.168 and the host address is 001.104 (1.104).

IPv6 address example:

IP Address: fdf4:4983:18be:12ff:e9af:7e13:9447:7fb8

Each character in an IPv6 address represents 4 bits (a nibble) and the IPv6 the subnet is expressed as an integer from 1 to 128. Additionally, with IPv6 there are no gateways.

IPv6 address example:
Network address (48 bits): fdf4:4983:18be
Subnet (16 bits): 12ff

Device Address (64 bits): e9af:7e13:9447:7fb8

Figure 4.2.1 is an example of IPv6.

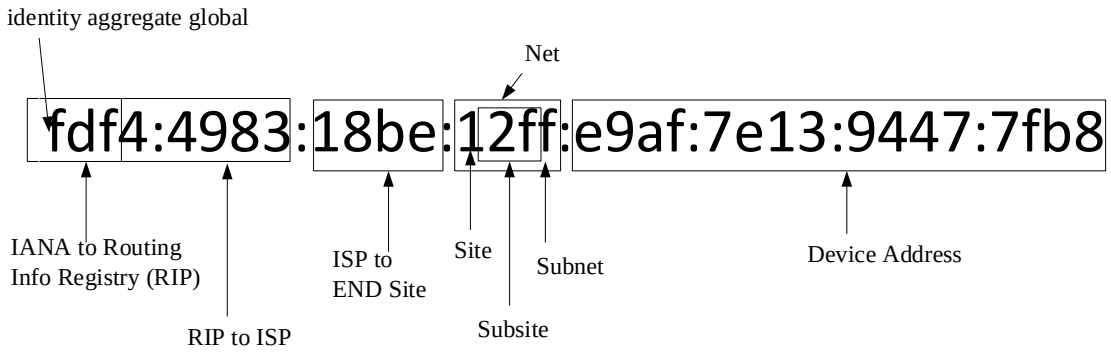

Figure:4.2.1 IPv6

The additional addresses with IPv6 means that it is now possible to assign a separate IP address to each IoT device. In essence IPv6 removes the need to have network address translations (NAT) improving amongst things latency.

IPv6 besides having a longer address also has a different format and an example of the two formats is shown in figure 4.2.2

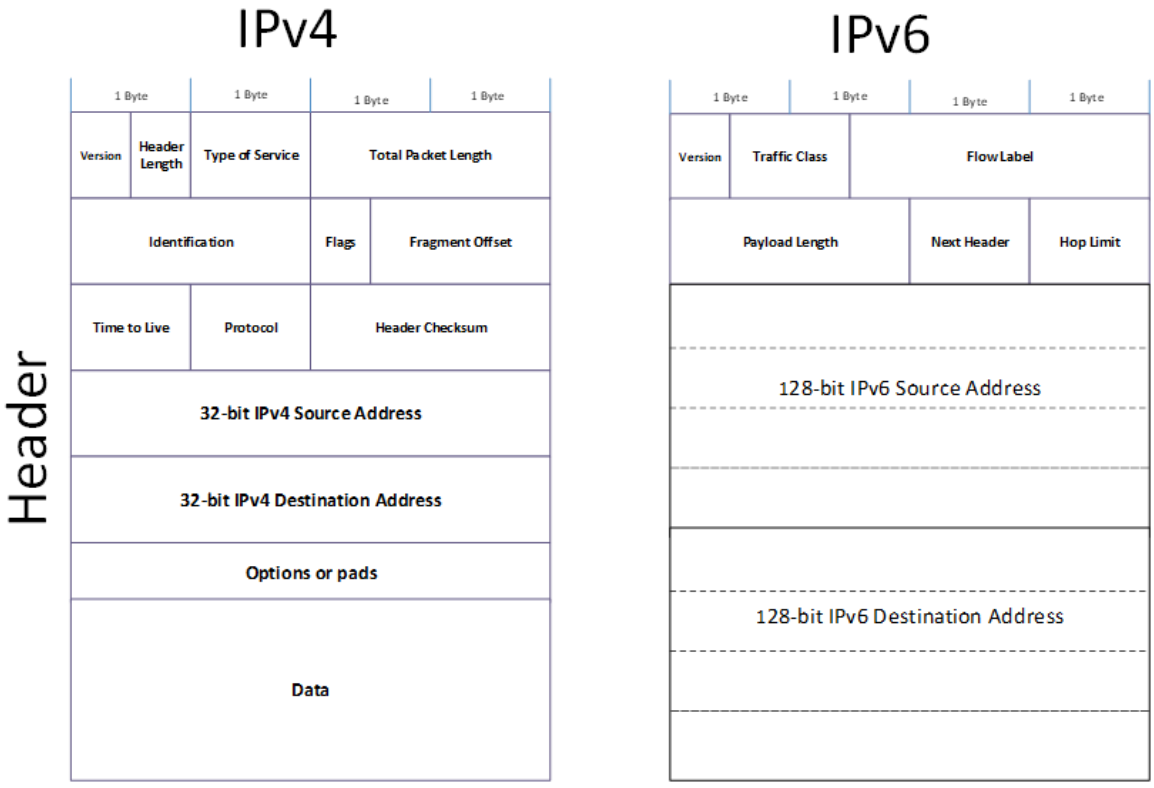

Figure 4.2.2 IPv4 and IPv6

The IPv6 format now allows for several key attributes:

- Quality of Service
- Network layer security (IPsec built in)
- Reduced processing for headers
- More efficient routing

The following URLs can be used to perform a deeper dive if needed besides examining the RFC documents.

- www.wikipedia.org/wiki/IPv6
- www.wikipedia.org/wiki/IPv4

4.3 Barcodes

A barcode is a label that is read by a machine, camera. Barcodes are included in the protocol list because they are used extensively with IoT solutions as well as being able to store and provide data when queried

via an optical scanner. A barcode has specific information about the item that it is attached to. Information can include contents, source, links to an application or web site. The barcode once imprinted cannot be easily changed. There are six general bar code types which are used in the industry.

One of the bar code types is shown in figure 4.3.1. In this case the barcode when scanned links the user to a web site.

Figure 4.3.1 Barcode.

Another barcode is the Quick Response Code (QR) in ISO/IEC 18004. An example of a QR code is shown in figure 4.3.2. The QR code consists of a series of black squares that are arranged in a pattern that can be read and decoded by an optical device that can read QR codes. The QR code in figure 4.3.2 points to the www.nextgconnect.com web site.

Figure 4.3.2 QR code

There are other symbology bar codes like PDF417 and AZTEC.

Bar codes are used extensively in industrial, commercial and retail applications.

The following URL links can be used to do a deeper dive.

- www.gs1.org/standards
- www.wikipedia.org/wiki/Barcode
- https://www.qrcode.com
- www.wikipedia.org/wiki/QR_code
- https://www.scandit.com/types-barcodes-choosing-right-barcode/

4.4 RS-232

RS-232 legacy protocol that is primarily associated with connecting computers to printers, although that was not its initial intention.

It is included in the IoT device protocol list because an IoT controller may connect to a printer or other device using a RS-232 connection including modems.

RS-232 is an open standard following EIA-232. RS-232 can utilize synchronous or asynchronous communication and uses serial communication and is the foundation for Universal Asynchronous Receiver/Transmitter (UART) communication.

RS-232 utilizes two types of connectors the DB-25 and DB-9**.** In DB-25, there are 25 pins available which are used for many of the application.

With RS-232 you have a transmitter referred to as the data terminal equipment (DTE) and the receiver as the data communication equipment (DCE) as shown in figure 4.4.1 However in RS-232 a device can be both a DTE and DCE facilitating 5 wires to be used to connect the two devices instead of the 3 shown.

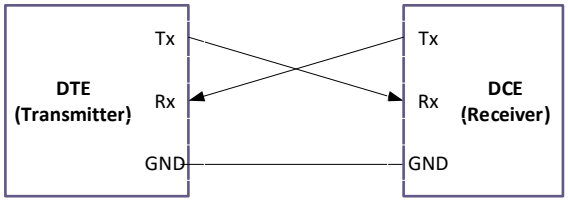

Figure: 4.4.1 RS-232

RS 232 is used for low-data-rate and short-range applications. The range for a RS-232 signal is typically 25 ft (8m) before some type of signal booster is required. This protocol is particularly effective in equipment used in noisy environments such as factories, process control plants, and utilities sites.

Besides printers RS-232 is also used for communicating with PLC, CNC machines, and servo controllers. It is also still used by some microcontroller boards, receipt printers, point of sale system (PoS), etc.

The following URL can be used to perform a deeper dive if needed.

- www.wikipedia.org/wiki/RS-232

4.5 RS422

RS-422 is a standard following EIA/TIA-422. RS-422 is used to send data faster and over longer distances than RS-232. RS-422 is a balanced interface that can support not only point to point but multi-drop. RS-422 is specified as a simplex multidrop standard, which means you can have one transmitter and 10 receivers

operating in full or half duplex mode on the same bus. RS-422 can also have higher data speeds than RS-232 where data speeds up to 10Mbps when under 50feet and 100kbps up to 4000 feet.

An example of a RS-422 circuit with 2 slaves is shown in Figure 4.5.1.

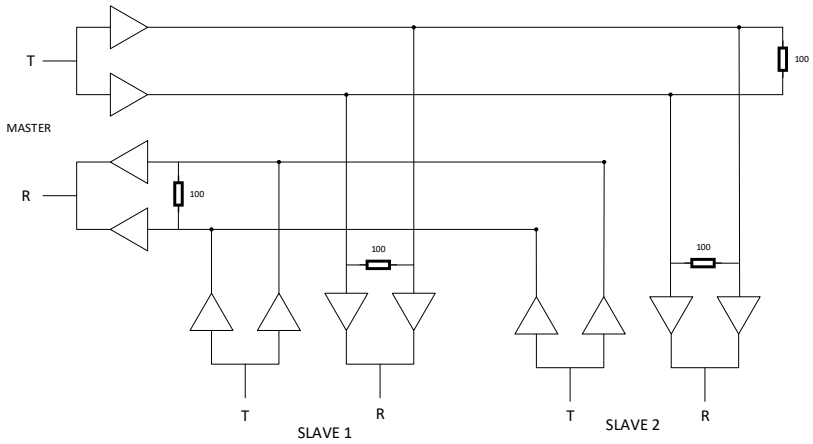

Figure 4.5.1 RS-422

The RS422 typically utilize a D-sub 25-pin and D-sub 9-pin as the connector.

You can extend a RS-232 connection by placing a RS-422 amplifier in line.

Also, a RS-485 driver can be used in RS-422 applications however RS-422 drivers cannot be used in RS-485 applications.

The following URL can be used to perform a deeper dive if needed.

- www.wikipedia.org/wiki/RS-422

4.6 RS-485

RS-485 is a serial communications standard used in many IIoT and building automation applications. RS-485 builds upon RS-422 capabilities. RS-485 is TIA/EIA-485 and can support multipoint systems besides peer to peer (PTP) and multidrop. The RS-485 can support up to 32 devices, 32 transmitters and 32 receivers, and uses a master-slave /token-passing (MS/TP) configuration.

RS-485 uses twisted pair(s) of wire and can be found in single, half duplex and duplex configurations. Many of the IIoT and building automation installations utilize a half-duplex configuration often called two-wire but a third wire is used for a reference or ground reference.

MS/TP device is either a master or a slave where the master initiates the queries and the slaves respond to the specific queries.

Figure 4.6.1 shows a half and full duplex RS-485 configuration.

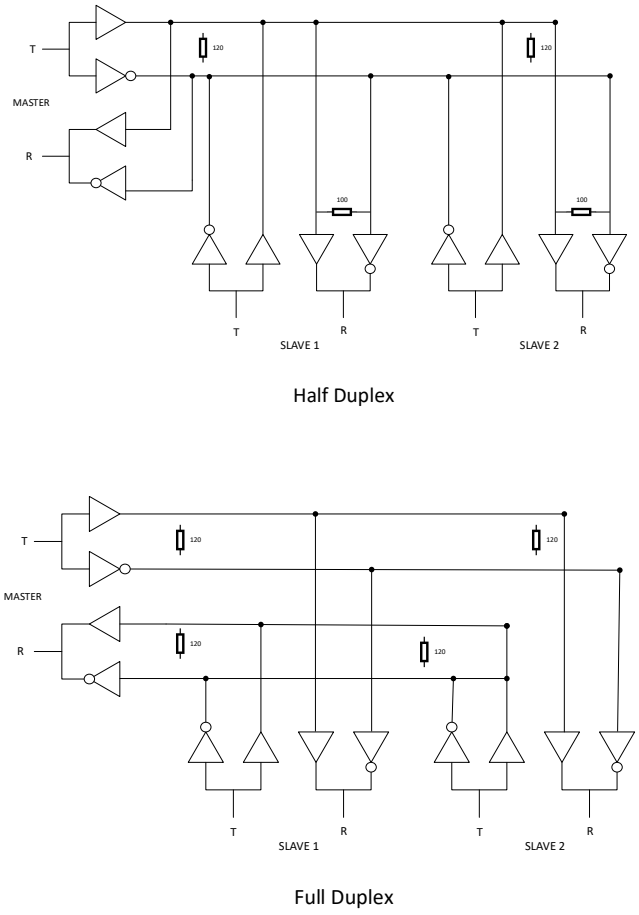

Figure 4.6.1 RS-485

RS-485 is used in IIoT and building automation configurations because it can support long distance runs with high data rates. Specifically, RS-485 can support 100kbps at 4000ft.

Since RS-485 uses differential signaling, it resists electromagnetic interference from motors and welding equipment. RS-485 however does not define or specify any communication protocol.

A RS-485 driver can replace a RS-422 however a RS-422 cannot replace a RS-485.

The following URLs can be used to perform a deeper dive if needed.

- www.wikipedia.org/wiki/RS-485
- http://www.rs485.com/rs485spec.html

4.7 4-20mA

The 4-20mA IoT protocol is also called a current loop protocol and is a serial protocol. The 4-20mA protocol has been used in the Industrial IoT industry for decades and is one of the most utilized protocols. It is used for transmitting sensor data or information for a host of processing and monitoring applications. The 4-20mA's popularity is achieved largely due to its simplicity and robustness to work in high noise environments leading it to be an excellent choice for factory automation control.

Devices utilizing the 4-20mA protocol include Supervisory Control and Acquisition (SCADA) systems and Programmable Logic Controller (PLC)s. The 4-20mA protocol utilizes a two (2) wire method though more wires could be used. Some example where 4-20mA protocol can be found involves controlling HVAC systems, control values, or monitor environmental conditions.

A key factor with using a 4-20mA protocol is the high level of adoption for this protocol and it is an open standard protocol following ISA 50.00.0. The high level of adoption has large vendor pool to select different IoT devices from which use this protocol.

.

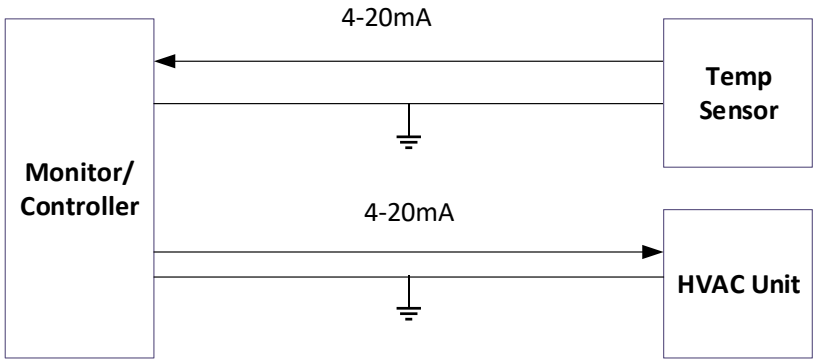

Figure 4.7.1: 4-20mA architecture

A 4-20mA architecture is shown in Figure 4.7.1. The 4-20mA protocol utilizes four primary components for device control.

- Sensor or transducer.
- Transmitter or signal conditioner.
- Power source, DC.
- Receiver or monitor.

In figure 4.7.1 a temperature sensor reports are values to a monitor/controller by a proportional signal. Specifically, with 4-20mA protocol a 4mA signal represents 0% while a 20mA represents 100% of what the sensor is able to report. Therefore, 10mA value could represent 30 degree C. The monitor/controller then acts as a transmitter and sends a signal to the HVAC unit turning it on. Alternatively, the controller could communicate with a motor and adjust its speed based on other sensor inputs.

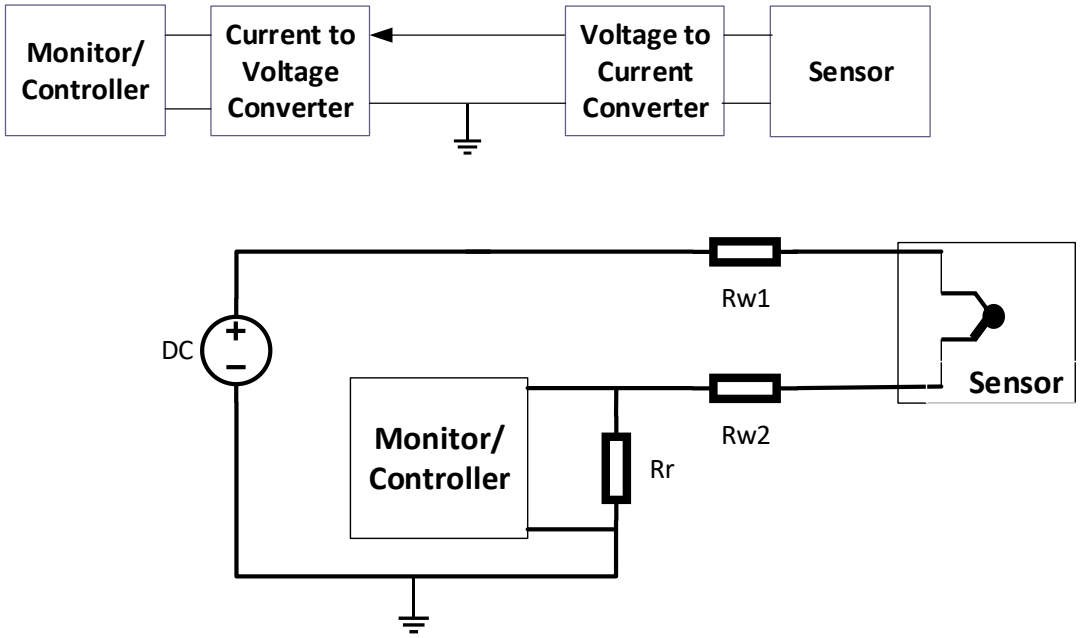

Figure 4.7.2 4-20mA sensor

Looking a little more into 4-20mA is Figure 4.7.2. In figure 4.7.2 a temperature sensor is being monitored. Power is supplied to the wire connecting the sensor. Based on the temperature the resistance of the sensor changes and so does the current moving along the wire. The sensors output voltage is converted to a current value somewhere between 4-20mA which is dependent upon the DC power and the wire resistance Rw1. Back at the monitor the current difference is converted to voltage by the use of Rr and measured.

However, 4-20mA is a proportional protocol and this means we need to calibrate each circuit. For instance, the power source needs to supply a constant current over the entire loop. Therefore the voltage applied needs to be equal or greater than all the voltage drops in the system caused by Rw1, Rw2 and Rr to for the constant current, usually 20mA. However, as the resistance of the sensor changes a new voltage drop is introduced reducing the current in the loop. The current that flows over Rr produces a voltage and this voltage is then read by the monitor. The monitor then uses a look up table that was calibrated for the loop itself to equate a voltage level to a particular sensor reading. Typically the 4mA would represent the sensor's zero-level output, and 20mA would represent the sensor's full-scale output. So, in this case we could have 4mA equated to 0 degrees C and 20mA equating to 35 degrees C or higher for the temperature sensor. The current values of course are converted to voltage and since Rw1, Rw2, the sensors resistance, and Rr are all part of the circuit path having an absolute value is not needed as along as the circuit is calibrated based on the values present.

The 4-20 mA current loop is a popular standard for IIoT and building automation applications because of its simplicity and ability to function in hostile signal environments. However, it requires home runs for every connection which may present an issue with reconfiguration or construction that may not be related.

The following are some URL links for more information. However, the wiki URL for this protocol is a great start.

- www.wikipedia.org/wiki/Current_loop

- www.isa.org
- https://www.analog.com/en/products/interface-isolation/4-20-ma.html

4.8 SPI

Serial Peripheral Interface (SPI) is a serial communication protocol that is used over very short distances. The communication is synchronous and is also full duplex. The communication being serial uses shift registers for sending and receiving data. SPI communication usually uses 4 wires and is a master slave configuration with only one master and many slaves. SPI data rates can be 10Mbps.

One common use of the SPI protocol is with SD cards for data transfer. However, there is no formal SPI standard and this at times leads to some compatibility issues.

Figure 4.8.1 shows a typical SPI master slave configuration with only one slave.

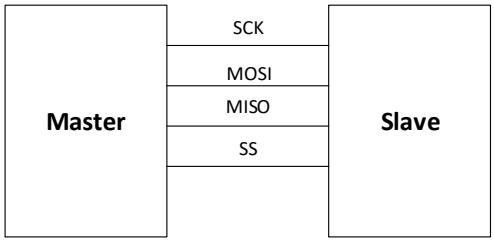

Figure 4.8.1 SPI

The four signals shown in Figure 4.8.1 are:

Signal	Name	Purpose
SCK or SCLK	Serial clock	Serial clock from master
MOSI or SDO	Master Output Slave Input	Data Output from master
MISO or SDI	Master Input Slave Output	Data Output from slave
SS	Slave Select	Signal to select Slave to communicate with

With SPI there is usually only one master however there can be multiple slaves either in a multiple slave select configuration as shown in figure 4.8.2 or as a daisy chain configuration, figure 4.8.3.

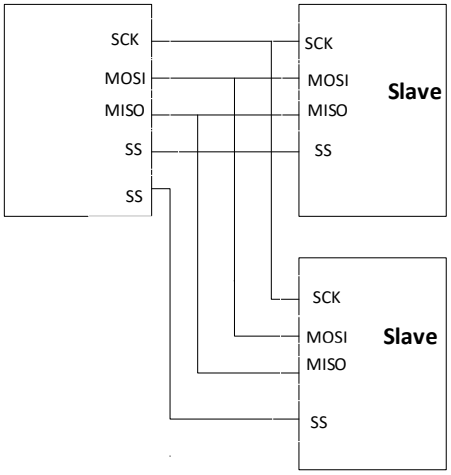

Figure 4.8.2 Multiple Slave Select

With multiple salve select the master has many slave select outputs which are individually connected to each slave. Only two slaves are shown for drawing ease.

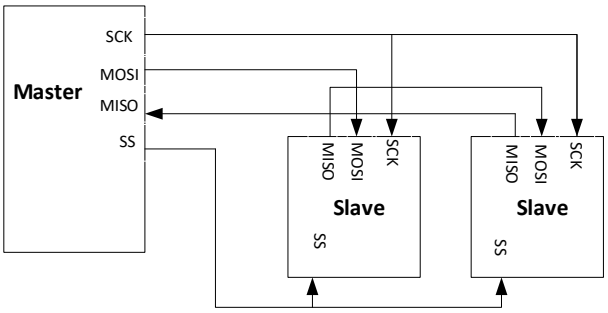

Figure 4.8.3 Daisy Chain

In the daisy-chain configuration, all slaves share a common SS line. Also, in a daisy chain configuration the data is shifted out of the master into the first slave, and then out of the first slave into the second slave. Only two slaves were used in figure 4.x.3 for drawing ease.

The following URL's can be used to obtain more information if needed.

- https://www.arduino.cc/en/Reference/SPI
- https://en.wikipedia.org/wiki/Serial_Peripheral_Interface

4.9 I2C

Inter IC (I2C) is a serial protocol that is commonly used for connecting sensors and is a synchronous bidirectional communication method. I2C is defined in specification I2C-Bus Specification 6.0 and UM10204. I2C is used in a master slave configuration allowing for multiple devices to be connected with

only two wires. The I2C communication speed has a standard rate of 100kbps however it can achieve 3.4Mbps and is defined by the current master.

With I2C each device is either defined as a master or slave. The master defines the clock speed of the bus for data transfer. With I2C there can be more than one master and there is an arbitration method used between multiple masters and multiple slaves for data transfer on the bus. With I2C you can have multiple masters connected to multiple slaves or there can be multiple masters controlling a single slave.

Figure 4.9.1 is an example of the most basic I2C configuration with one master and one slave device. Each I2C bus consists of two signals which are a SCL and SDA. The SCL is the clock signal and SDA is the data signal and both are bidirectional.

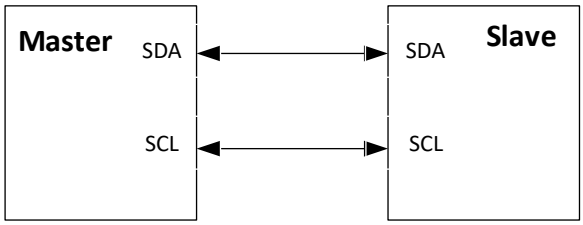

Figure 4.9.1 I2C

The I2C message uses a 7- or 10-bit address allowing for a master to communicate with up to 128 slaves using 7 bit address or 1023 with a 10 bit address. Figure 4.9.2 is an example of multiple I2C devices connected together where the device can be a master or a slave.

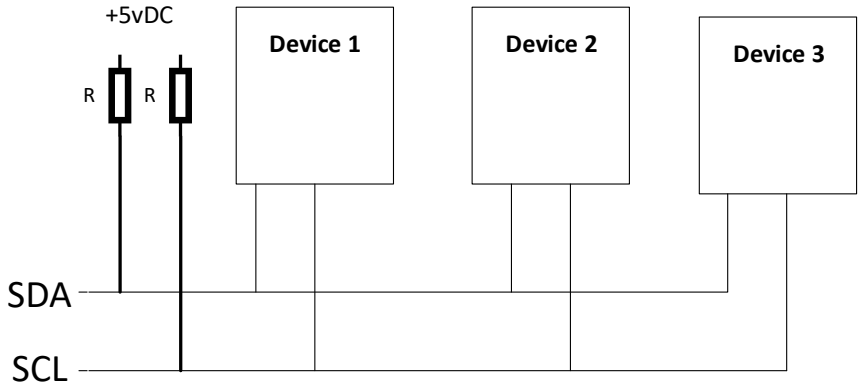

Figure 4.9.2 Multiple I2C devices

The various devices on the I2C bus are either masters or slaves as shown in figure 4.9.2. I2C slaves are devices that respond to the masters requested to transfer data. A slave however cannot initiate the transfer of data. The master I2C device drives the clock, SCL, that is used for determining the data speed.

In figure 4.9.2 pull up resistors are shown connected to the SDA and SCL lines. When all the devices are inactive the signals for both the SDA and SCL are pulled high. The SDA line is pulled low either by the

master to initiate the communication or by the slave to respond to the master. The pull up resistor is not needed in figure 4.9.1 since there is one master and one slave.

The following URLs can be used to perform a deeper dive.

- https://www.i2c-bus.org/
- https://i2c.info/i2c-bus-specification
- https://en.wikipedia.org/wiki/I%C2%B2C

4.10 HART

The Highway Addressable Remote Transducer (HART) Protocol for sending and receiving digital information across wires. HART is primarily used in IIoT and building automation applications. The HART protocol is proprietary and is depicted in Figure 4.10.1 where it shows Frequency Shift Keying (FSK) being superimposed on top of the 4-20mA signal. The inclusion of FSK with HART enables two-way communication as well enabling more information to be sent between the controller and the device.

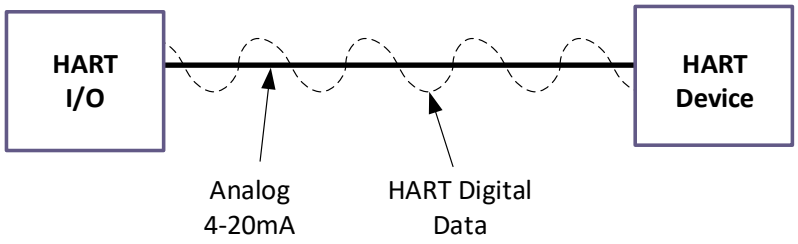

Figure 4.10.1 HART Protocol

The Hart protocol is essentially a Master/Slave protocol and has the ability to be used in different network topologies like point to point (PTP) as well as multidrop for up to 15 devices.

The HART protocol using FSK can communicate at 1200 bps without interrupting the 4-20mA signal. Additionally, the 4-20mA signal does not interfere with the FSK signal allowing for smart devices to be placed in the loop providing more frequent and robust information.

At this writing the HART protocol was at version 7.3.

The following URL link is best for obtaining more information.

- www.fieldcommgroup.org

4.11 Modbus

Modbus is a serial communication protocol that initially started out using RS-232 then expanded to include RS-485 and now TCP/IP. Modbus is typically used in Supervisory Control and Data Acquisition (SCADA) systems. Modbus is an open standard that is run by the Modbus organization. Currently the Modbus standard is defined in the Modbus application protocol V1.1b3.

Modbus signaling is independent of the media and has not changed since its inception using a simple protocol data unit (PDU). Modbus also follows a master slave format allowing for Peer to Peer (PTP) as well as multi-drop. Figure 4.11.1 is a high-level depiction of a simple Modbus communication scheme connecting just two devices where the client is the master and the slave is the server.

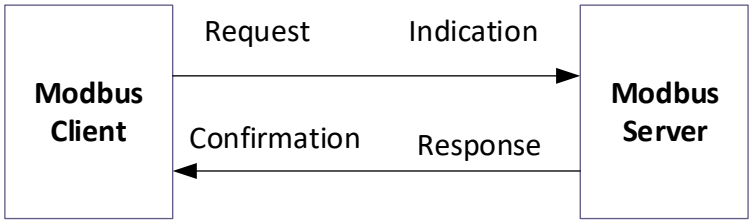

Figure 4.11.1 Modbus

In a Master/slave configuration the device operating as a master will poll one or more devices operating as a slave. An important distinction is the slave device will only respond to requests and not initiate them. The master initiates the transactions or queries. Each slave in a network is assigned a unique unit address from 1 to 247. When the master requests data, the first byte it sends is the Slave's address. This way each slave knows after the first byte whether or not to ignore the message.

Modbus connections can use one of two basic transmission modes, ASCII or Remote terminal unit (RTU). Depending on whether you are using ASCII or RTU the messages may be coded differently. Modbus RTU and Modbus ASCII talk the same protocol. The only difference is that the bytes being transmitted over the wire are presented differently. For example, with Modbus ASCII the messages are in a readable ASCII format but with a Modbus RTU a binary code is used. The primary Modbus method uses RTU with ASCII being demoted.

Therefore, to talk with a MODBUS device you must use the same mode as configured in the client and server.

With Modbus RS-232 is used for short distance point-to-point communication as well as RS 422, which is a bidirectional extension of RS232 but is able to support longer distances between devices. If you are using Modbus with RS-232 devices, you can have only two devices total no more.

RS485 used with Modbus for multipoint/drop communication where many devices are connected to the same signal cable, bus, using a master slave method. In the master slave configuration, there is one master with Modbus and numerous slaves.

You cannot have more than one Master on a with a Modbus RTU (RS-485) network. Since you have only one master if you use a gateway as the master you can only have one gateway. But you can have multiple gateways configured as slaves residing on the same Modbus RS-485 network. However, you cannot use multiple gateways to read more points from the same Modbus slave device.

Modbus also supports TCP/IP through Modbus TCP which is also called Modbus IP or Modbus Ethernet. The main difference between Modbus RTU and Modbus TCP is that Modbus uses 802.3 while Modbus RTU utilizes a serial protocol. Figure 4.11.2 is an example of a mixed Modbus environment where the Modbus RTU uses RS-485 or RS-232. Modbus TCP uses Ethernet.

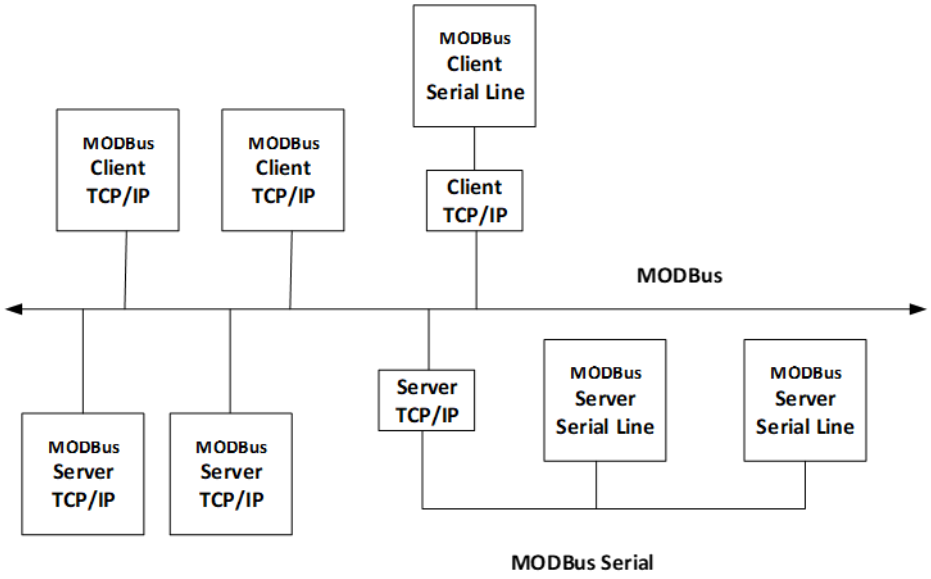

Figure 4.11.2 Modbus TCP and Modbus RTU

With Modbus TCP, controllers on Ethernet to be the Master to hundreds of Modbus TCP devices. Modbus TCP allows for unlimited clients eliminating the RS485 limitation of 32 devices., With Modbus TCP, there is the ability for a network to use multiple Clients/Masters.

The following URLs can be used for obtaining more information.

- www.modbus.org
- www.wikipedia.org/wiki/Modbus
- https://www.automation.com/library/articles-white-papers/fieldbus-serial-bus-io-networks/introduction-to-modbus

4.12 LonTalk

LonTalk is the communication protocol used with LonWorks and is part of IEC14908.

The LonTalk protocol uses Carrier Sense Multiple Access (CSMA) as a data collision avoidance method and is able to support data rates from 610bps to 1.25Mbps. Using CSMA LonTalk can avoid or minimize throughput degradation due to collisions. The LonTalk CSMA is designed to perform better than Ethernet. However, LonTalk while part of ANSI/EIA 701.9 uses a chipset that is sole sourced.

LonTalk can support more than 500 transactions per second and is able to use a variety of media which include.

- Twisted pair
- Power line (powered or unpowered)
- Radio frequency 400 and 900-MHz bands
- Coaxial cabling
- Fiber optics

The following URLs can be used for obtaining more information.

- https://www.echelon.com/
- www.lonmark.org

4.13 Fieldbus

Fieldbus is a protocol that is used for both IIoT and building automation applications. Fieldbus is an open standard following IEC 61158. The IEC 61158 standard has over 75 different communication technologies all of which perform similar functions. However, the particular function is vendor specific.

Fieldbus is designed to connect the field instruments, sensors/motor controls/etc., with the control and monitoring host system as part of a distributed control system (DCS).

Fieldbus has the advantage over 4-20mA in that it is digital and not analog. Fieldbus can also connect multiple devices in parallel using the same pair of wires providing not only communication but power as well unlike 4-20mA systems which are serial.

Fieldbus uses bidirectional digital communication in real time for a better closed loop control system. Because Fieldbus uses digital signaling it does not have the aliasing that can occur with repeated digital to analog (D/A) and analog to digital (A/D) conversations resulting in better resolution and accuracy of measured values. Figure 4.13.1 shows an example of a Fieldbus configuration.

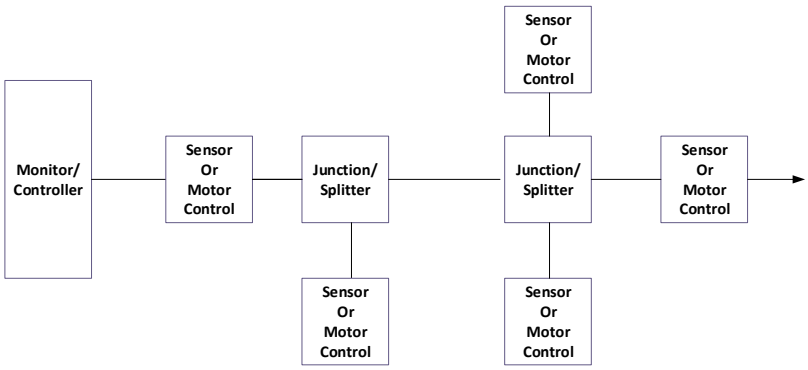

Figure 4.13.1: Fieldbus

The Fieldbus network is typically a peer to peer (PTP) however FF (Ref section 4.16) can support a host of other topologies including daisy chain (multidrop), star, ring, branch and tree. Fieldbus can typically support up to 31/32 field instruments connected to a controller. Data speeds of 1.2k to over 1Gbps are possible with a Fieldbus network providing the means to leverage immense amounts of data generated by modern automation.

The following is just a list of some of more common protocols utilizing Fieldbus.

- Foundation Fieldbus
- Profibus
- LonWorks

The following URL links are a good start for obtaining more information.

- www.fieldcommgroup.org
- www.wikipedia.org/wiki/Fieldbus

4.14 ARCNET

Attached Resource Computer NETwork (ARCNET or ARCnet) is a data link layer technology that does not have an application layer that is defined. It's often called a fieldbus technology but it is not part of the fieldbus standard. ARCNET has its own standard and that is ANSI 878.1.

ARCNET devices are typically integrated into IoT devices since ARCNET is a layer 1 and 2 device.

ARCNET supports a variety of network topologies including Bus, Star and tree. It also supports P2P, Daisy chaining and multidrop. However, it does not support Ring or loop topologies.

ARCNET is a protocol that is designed for fast and deterministic processes and is well suited for IIoT even though it was initially a BAS protocol. ARCNET can support 256 units with data speeds of 10Mbps.

ARCNET can be configured with both twisted pair or cable. RG-62U or thin ethernet is the cable that is used with ARCNET and this uses either passive or active hubs depending on the network topology used. ARCNET also uses twisted pair wiring, RS-485, and this can be either DC or AC coupled facilitating a host of possible applications. ARCNET can also use multimode fiber.

ARCNET is unique in that each ARCNET node can be a master at any time. The node becomes a master when it sends data. To make this happen a ARCNET node or device needs a controller chip and a cable transceiver suitable for the media technology being used. The part of the ARCNET node that has the controller chip and transceiver is referred to as the network interface module (NIM). The NIM is modular so that it can be changed to match the media that is being interfaced with.

As part of the bus communication scheme ARCNET uses a token passing protocol where a node can send a message only when it receives a token. When the token is received the node then becomes a temporary master on the network and the length of the message that the node can send is limited in size and duration to avoid having any one node control the network. The packet length can vary from 1 to 508 bytes.

The following URLs can be used to help obtain more information.

- https://en.wikipedia.org/wiki/ARCNET
- www.arcnet.de

4.15 PROFINet

Process **Fi**eld **Net** (PROFINet) uses the 802.3 ethernet topology. PROFINet also incorporate wireless standards Wi-Fi and Bluetooth as part of its specification IEC61158, (PROFINET 2.4). PROFINet is

technically fully compatible with a standard office ethernet network however it is designed to use industrial ethernet and not be comingled with an office LAN.

PROFINet uses both TCP/IP as well as PROFINet communication over 802.3. Specifically, TCP/IPP is used for data that is not time critical usually in the of 100ms or more. However, the PROFINet signaling is used for real-time (RT) applications needed 10ms responses and Isochronous RT where responses are needed within 1ms.

Figure 4.15.1 shows the OSI stack for PROFINet. In figure 4.15.1 it is shown that 802.3 is used for the delivery however PROFINet or TCP/IP can be used for layer 3 and 4.

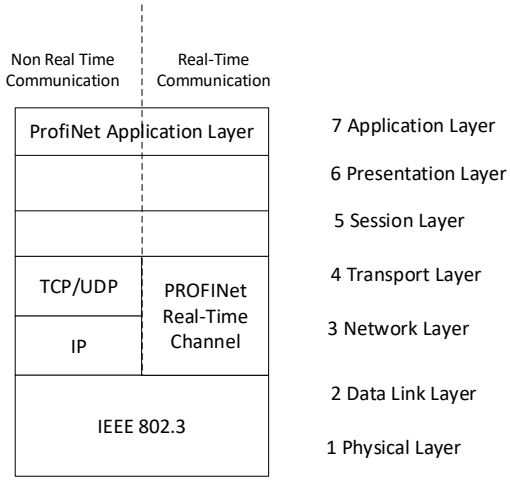

Figure 4.15.1 PROFINet OSI Stack

PROFINet has a defined architecture that is shown in Figure 4.15.2. Figure 4.15.2 shows three types of devices the make up the PROFINet architecture and they are:

1) **IO-Controller:** This controls the automation tasks like configuration, process data, and, alarms.
2) **IO-Supervisor:** This is used for diagnostics, status, control and setting parameters for individual IO-Devices. The IO-Supervisor typically resides on a PC or server and is software only.
3) **IO-Device**: This a field device which can be a sensor, motor control or other device that is monitored and controlled by an IO-Controller. Additionally an IO-Device can also consist of several modules and sub-modules.

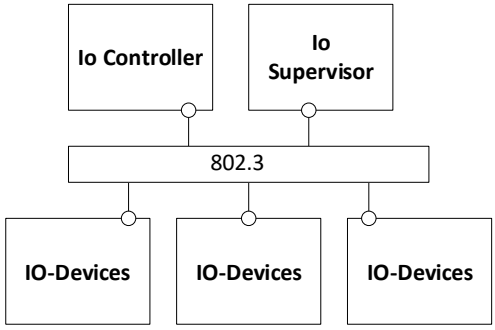

Figure 4.15.2 PROFINet Topology

However, PROFINet can also work with PROFIBus and other FieldBus systems. Figure 4.15.3 is an example of a PROFIBus and FieldBus system being controlled by a PROFINet system. Because the other systems are not based on PROFINet protocol a Proxy or Gateway needs to be used for the protocol conversion.

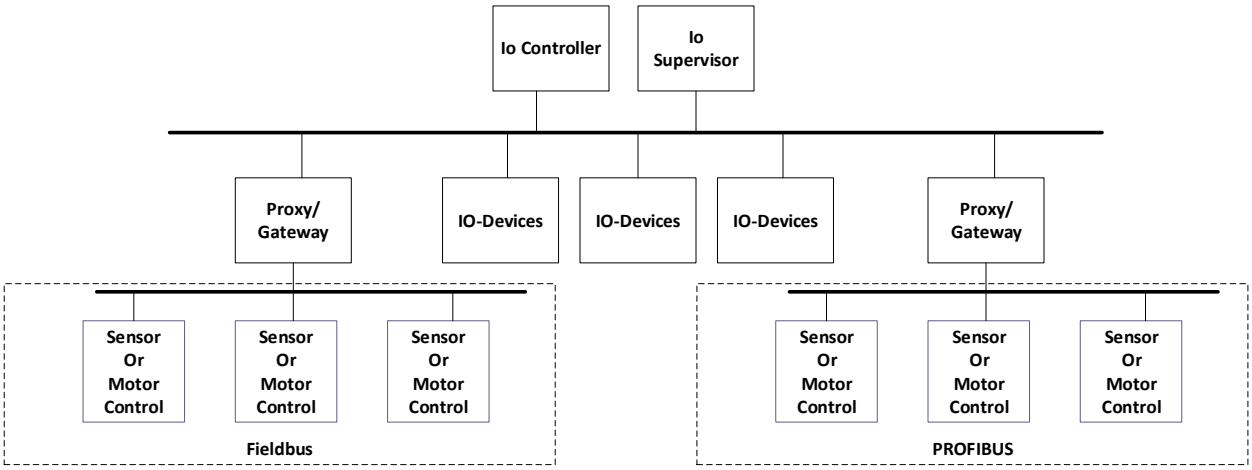

Figure 4.15.3 PROFINet with multiple protocols

The following URL can be used for obtaining more information about PROFINet.

- www.profinet.com

4.16 Foundation Fieldbus

Foundation Fieldbus (FF) is a Fieldbus protocol that is used for both IIoT and building automation applications. FF is an open standard following IEC 61158 at the application layer however there are two versions of FF and they are H1 and HSE.

FF H1 and FF HSE are both designed to connect field instruments, sensors/motor controls/etc., with the control and monitoring host system as part of a distributed control system (DCS).

FF H1 was designed specifically for process automation and it follows the IEC 61158 specification using Manchester coding(check) and can connect up to 32 devices per segment. FF H1 can support speeds up to 31.25kbps and communicate over distances up to 9.5km but at reduced speeds. FF H1 is also able to support devices that are in hazardous locations requiring intrinsically safe power conditions.

FF HSE is also meant for process automation and has the same messages as FF H1. However, FF HSE is different than FF H1 in that it uses IEEE 802.3 as its communication media. FF HSE is meant to connect higher-level, smart devices, that produce and process high amounts of data. Because FF HSE uses IEEE 802.3 it can support data speeds up to 1Gbps over 100m. There is no limit to the number of devices that can be connected in a FF HSE network however it requires a separate external power source making it not suitable for use for intrinsically safe environments.

Both H1 and HSE can coexist within the same control network as shown in Figure 4.16.1. However, FF H1 and FF HSE have different signaling and require an interface module. Although FF HSE is able to communicate faster than FF H1 when the two protocols share the same network the FF HSE speed will be dictated by the FF H1.

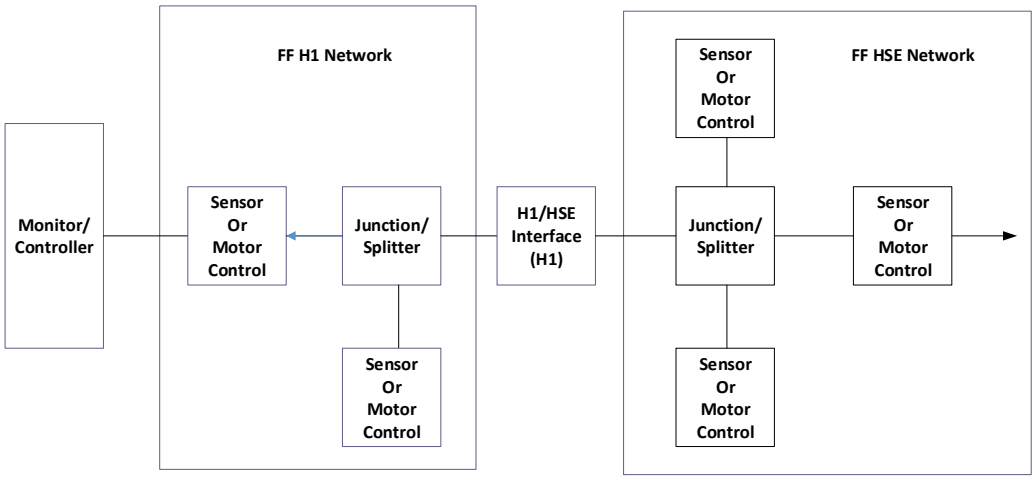

Figure 4.16.1 Foundation Fieldbus

The following URL link is best for obtaining more information.

- www.fieldcommgroup.org

4.17 Profibus

There are two versions of Profibus Process Automation (PA) and Decentralized Peripherals (DP). Both Profibus PA and DP are used in IIoT and building automation applications and are part of the Fieldbus technology family. Profibus PA is a variant of Profibus DP and each were designed for a specific purpose.

Profibus PA and DP are identical in protocol format, but their methods of transport is different. Profibus PA uses Manchester Coded Bus Power (MBP) over a wire pair while Profibus DP uses RS-485 to exchange data. However, you can have a Profibus PA and Profibus DP networked together with the use of a DP/PA coupler or DP/PA link that is used to perform protocol conversion between a PA and DP network as shown in Figure 4.17.1.

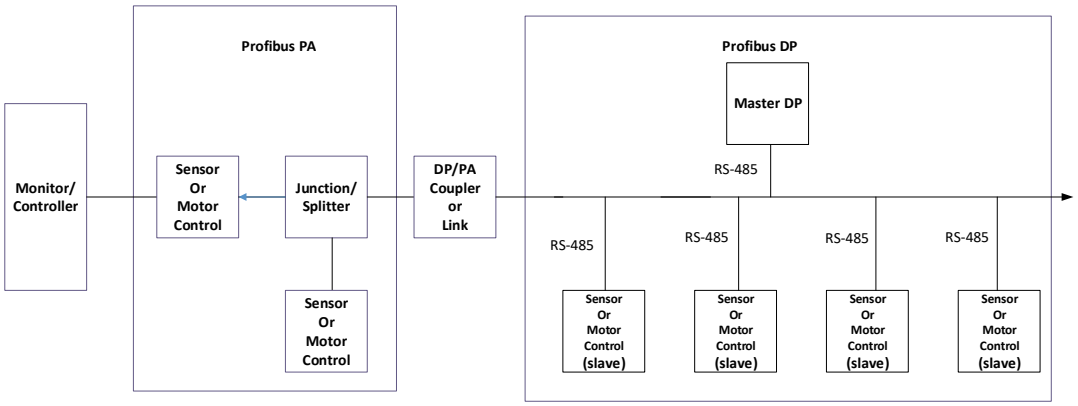

Figure 4.17.1 Profibus PA and HP

Profibus PA is used primarily for measurement and process control applications. The Profibus PA is meant to improve upon conf conventional systems such as 4-20 mA and HART in process automation and may be used in intrinsically safe applications since the data and power are transported over the same two wires between host systems such as the DCS or a PLC and field instrumentation. Profibus PA has a data transmission rate of 31.25 kbit/s.

Profibus DP is typically used in machine automation to operate sensors and actuators. Profibus uses a line configuration, bus, using half-duplex RS-485 transceivers which can use single, shielded or twisted pair cables. A decentralized system is achieved by using a bus architecture to communicate between the main controller (master) and the various I/O channels called slaves. The bus architecture enables Profibus DP to operate at 9.6kbps to 12Mbps and can transmit thoughts of I/O point information in milliseconds. This makes Profibus DP very applicable for processes that require fast action like turbine servos. However, with Profibus DP the I/O devices need a separate power source.

A simple comparison of Profibus PA and DP is included in Table 4.17.1.

	Profibus PA	Profibus DP
Physical	IEC 61158	RS-485
Length	1900m	1200m
# devices	32	32 or 126 with 4 repeaters
Data Rate	31.25kbps	9.6 – 12000kbps
Power source	In line	Aux 24vDC supply

Table 4.17.1 Profibus Comparison

The following URLs can be used for obtaining more information.

- www.wikipedia.org/wiki/Profibus
- www.profibus.com

4.18 INTERBUS

INTERBUS is a common protocol that is based on the Fieldbus IEC61158 standard following a master-slave access method in a ring topology architecture. Interbus utilizes serial data transmission in the process of communicating between different systems, computers, controller which are connected to subscribers which are actuators and sensors.

INTERBUS allows for rapid data transfer in the system because of its low protocol overhead and 500kbps – 2Mbps data rate.

The INTERBUS structure allows for a maximum of 512 subscribers /slaves to be connected and part of the network at the same time. The last subscriber on the ring, farthest away, also closes the ring as shown in figure 4.18.1. Each subscriber has its own power supply.

In figure 4.x.1 INTERBUS topology is shown which has several elements.

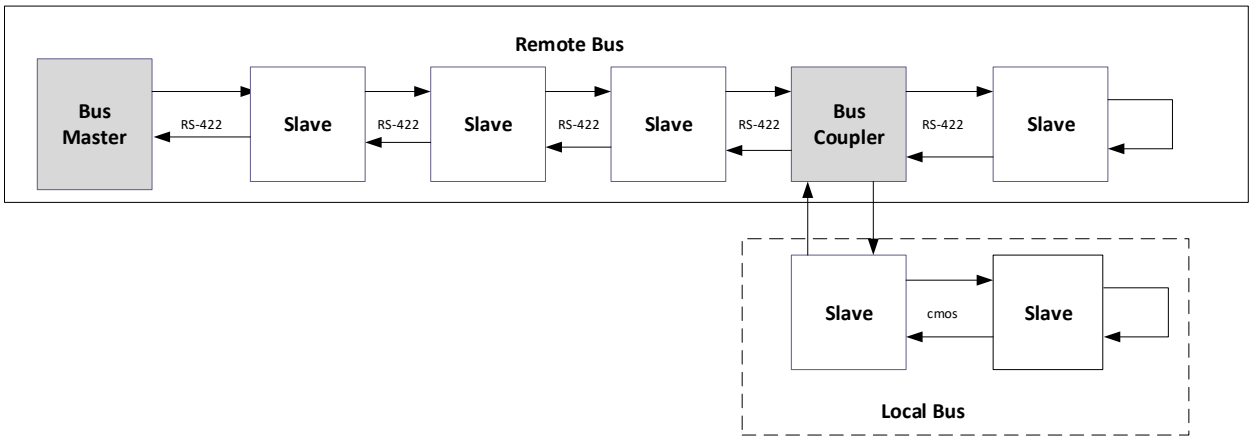

Figure 4.18.1 INTERBUS Topology

The backbone of an INTERBUS is called the remote bus which has a bus master that controls data communications. Each remote bus subscriber/device (i.e., a slave) is connected to the follow-on subscriber through full-duplex, RS-422 as shown in figure 4.18.1. The remote bus allows for 512 connections and 256 subscriber (slaves).

In INTERBUS each subscriber is connected to the next device via an RS-422 link. An additional local bus is connected to the main remote bus via a bus coupler in figure 4.18.1 however it uses CMOS in this example and not RS-422.

A subscriber with INTERBUS may also function as a bus-coupler. In addition, multiple local-bus loops are allowed which can also be a Tree, Ring, Star or point to point bus. With a local bus a total of 8 devices can be connected.

With INTERBUS the address of the subscriber is determined by the devices relative position in the bus and not on a fixed address.

The following URLs can be used to perform a deeper dive in INTERBUS if needed.

- www.phoenixcontact.com
- http://interbus.de
- http://www.interfacebus.com/INTERBUS_Field_Bus_Description.html

4.19 BACnet

Building Automation and Control Networks (BACnet) is a communication protocol used for building automation and was designed to provide interoperability between different vendors. BACnet follows ISO 16484-5 and is an open standard. BACnet is typically used for building automation involving HVAC control, fire systems, lighting, security, and elevators.

BACnet however does not directly control any device it just provides a communication method. To achieve this BACnet uses an object orient structure that consists of objects, properties and services.

BACnet adopted a collapsed OSI for 4 layers as shown in Figure 4.19.1.

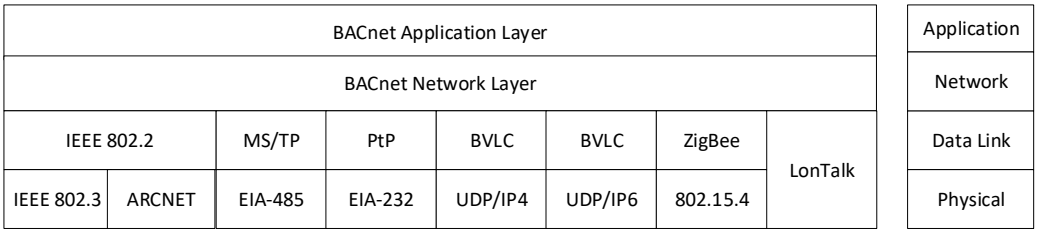

Figure 4.19.1 BACnet

Reviewing figure 4.19.1 BACNet supports multiple data link protocols and they are.

- Ethernet
- ARCNet
- MS/TP
- PtP
- LonTalk
- UDP/IP (enables WiFi and access to other IP networks)
- Zigbee

The BACnet collapsed OSI layers are shown in Figure 4.19,1 and they are:

- Application: This provides for monitoring and control of BACnet networks.
- Networks: Provides network to network communication.
- Data link: This is station to station comms within one network
- Physical – conversion of electrical signals into data

Additionally, there are three classes of BACnet devices and each has their own particular role in a BACnet ecosystem.

1. BACnet devices: These are sensor/actuator that understand BACnet protocol, referred to as native.
2. BACnet router: These enable different BACnet capable network to be connected to each other.
3. BACnet gateway: Provides a connection method to connect to non-compliant BACnet networks.

Figure 4.19.2 shows a native BACnet system all of one protocol. Figure 4.19.3 depicts the use of a BACnet router to communicate between two different BACnet networks.

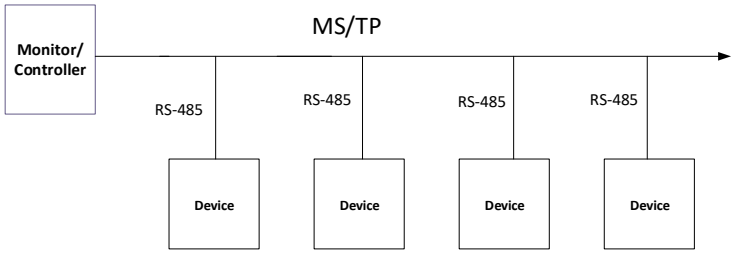

Figure 4.19.2 BACnet single protocol

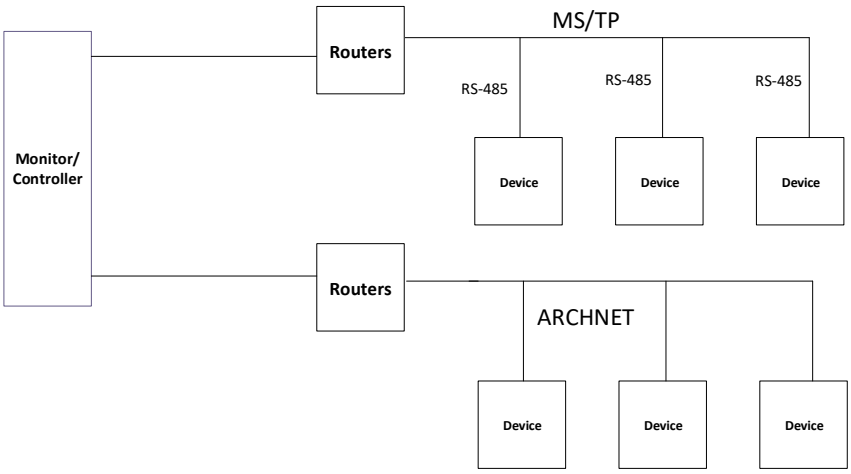

Figure 4.19.3 BACnet multiple protocol

Figure 4.19.4 shows the use of a BACnet gateway which is used to connect to a non-compliant BACnet network in this case a Modbus network.

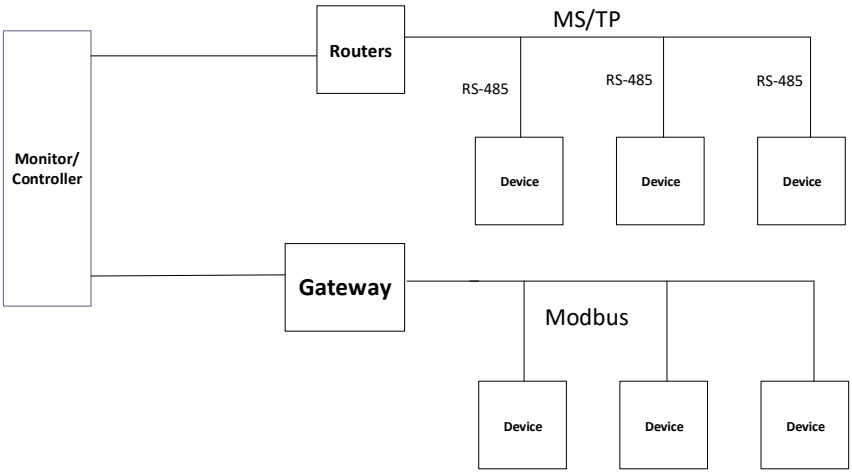

Figure 4.19.4 BACnet with non-BACnet systems

The following URL's can be used to get more information about BACnet.

- http://www.bacnet.org/
- https://www.bacnetinternational.org/

4.20 LonWorks

LonWorks describes several items that are typically part of a building automation system. LonWorks is an open standard using ANSI/EIA 701.9 and is similar to CEBus.

Lon in meant to refer to local operating network. LonWorks includes LonWorks devices which are the nodes which can be actuators or sensors. LonWorks uses LonTalk as the communication protocol.

LonWorks uses a peer to peer (PtP) or master-slave data communication scheme which is a Fieldbus derivative following IEC 61158. However, LonWorks can operate in what is called a Free Topology environment meaning you can mix and match topologies using LonWorks allowing you to utilize existing infrastructure.

LonWorks utilizes LonTalk as the method to communicate with other LonWorks devices. LonWorks via LonTalk supports numerous media types for providing connectivity including twisted pair, power line, coaxial cable, radio frequency (RF) ,infrared and fiber-optics.

LonMark networks are made up of nodes call LonMark Nodes/devices and each has a CPU (Neuron) and a transceiver that uses LonTalk. LonWorks is able to support from 2 to 32385 devices in the network.

The following URLs can be used for obtaining more information.

- https://www.echelon.com/
- www.lonmark.org

4.21 CEBus

The Consumer Electronics Bus (CEBus) defines a set of protocols and electrical standards for various household and office devices to send and receive commands and data.

CEBus is an open standard, ANSI/EIA-600, that was and is envisioned to be a universal low-cost communication method using multiple media. The ANSI/EIA-600 specifications define CEBus physical layer with EIA-600.3, the Node communications protocol with EIA 600.4, router protocols with EIA 600.5 and 600.6 and the CAL commands with EIA 600.8.

CEBus has two fundamental components which are a transceiver and a microcontroller.

CEBus however can communicate through a variety of media that include powerlines (PL), low voltage twisted pairs (TP), Coax (CX), InfraRed (IR), Radio (RF) and Fiber Optic (FO). When operating over a radio link it is at 915MHz.

CEBus also uses a robust set of control and operation tables called Common Application Language (CAL) contexts and are at the layer 7 level. The CAL tables define the control and monitoring commands that are used by remote devices to communicate with the CEBus devices. CAL commands enable CEBus devices from different manufacturers to communicate with each other. The CAL commands can be sent to the CEBus devices in the same room, from other rooms or even from the internet if allowed.

The CEBus protocol is similar to 802.3, Ethernet, in that it's a peer to peer network that uses Carrier Sense Multiple Access/Collision Detection and Resolution (CSMA/CDCR). Using CSMA/CDCR the network node waits until the media path used is clear avoiding simultaneous transmissions. With CEBus data can be sent at 10kbps and the delay is under 100ms.

However, there is also another EIA standard that comes within the CEBus realm and that is Standard EIA 709.2. EIA 709.2 unlike EIA600 defines how CEBus can use two- and three-phase electrical powerlines having data rates of 5.65kbps.

With EIA-600 the use of security protocols is optional.

The following are some URL links for more information. However, the ANSI/EIA link may prove more beneficial.

- www.ansi.org
- www.wikipedia.org/wiki/CEBus

You can also go to the CEBus Industry Council (CIC) for more information.

- http://www.cebus.org

4.22 X10

X-10 was the first power line technology used for home automation. X10 is restricted to power line communication unlike other home and building automation technologies and is not an open standard. X10

has two basic types of devices, one that receives commands and the other that functions in two-way communication.

XlO modules that are plugged into a standard electrical and sends it communication signal over the power wiring in a home or building to control light switches, power outlets and some relays. The X10 signal can be delivered locally from a controller or by a remote control method. Figure 4.21.1 is a simple example of a X10 system.

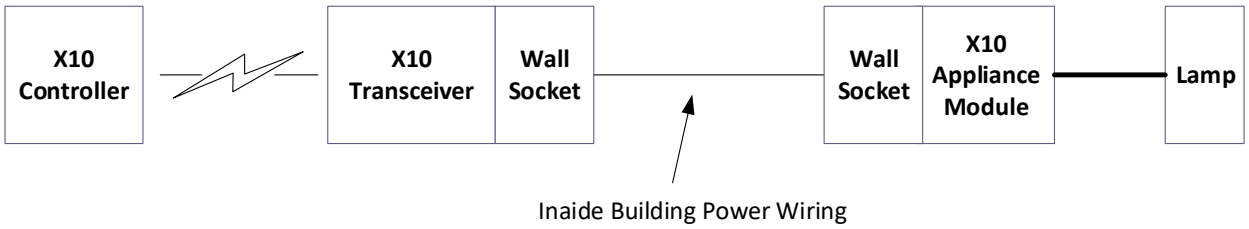

Figure 4.21.1 X10 Network

All packets of data sent using the X10 protocol consists of 3 elements:

- 4-bit House Code (A-P)
- 4-bit Unit Code(s): (1-16)
- 4-bit Command: (Binary number)

The house and unit codes allow for 256 unique addresses or 256 devices in a X10 network.

X10 systems can also include filters, couplers, and repeaters in to improve the reliability and performance.

Devices will react to any command or series of commands sent its way by X10 protocol, provided it's the correct house and unit code. You can also command multiple units at the same time by simply addressing each specific device before inputting the broadcast command.

With X10 a single bit is transmitted twice on each cycle of the 60 Hz AC signal at 120kHz (RF). The bursts are synchronized using the zero-volt crossing point for the AC signal. This limits the transmission rate to 60bps. The X10 message consists of two packets of 11 bits and each packet is sent after a 3 cycle wait meaning it takes 47 cycles to send on message, 0.8 seconds to the devices from the broadcast controller.

The signals from the control unit to the transceiver can be either infrared or RF. If it is RF the device operates at 310MHz.

The following URLs can be used for obtaining more information.

- www.x10.com
- www.wikipedia.org/wiki/X10

4.23 PLC

Power Line communication (PLC), IEEE 1901, also called broadband power line (BPL) and ethernet over power (EOP) is a physical system for transmitting data over existing AC medium voltage (MV) and low voltage (LV) electrical distribution wiring. A PLC system involves a basic process of simply superimposing the signal to be transmitted on the ac line typically using orthogonal frequency division multiplexing (OFDM) for modulating the data.

Electric utilities utilize PLC for transmitting and receiving control signals for the power grid as well as reading electric utility meters called automatic meter reading (AMR). Three AMR protocols used are Turtle, EMETCON and TWAC where TWAC is the most widely used. PLC is also a Powerline (PL) communications technology that uses the electrical power wiring within a home.

A PLC system is shown in figure 4.23.1 and contains the following major components.

1. Injector: used to convert IP traffic into a format that can be injected into Low and Medium Voltage lines (LV/MV) and is typically located in a substation using OFDM.
2. Coupler: Used to connect the injector to the LV/MV power line.
3. Repeater: Regenerates the OFDM signal so it can travel farther.
4. Extractor: removes the OFDM signal from the LV/MV power line
5. Modem: used to modulate and demodulate OFDM to/from TCP/IP

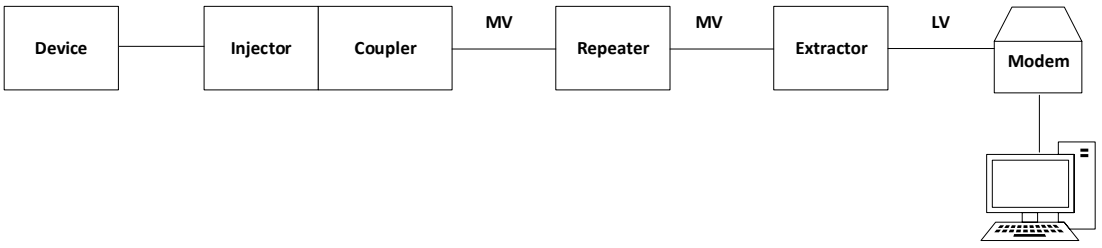

Figure 4.23.1 PLC System

HomePlug (HP) is another power line communication technology that is part of the IEEE 1901 standard. HP is used for home networking. There are several HP standards which include:

1. HomePlugAV is used for internet access on the power line (BPL) with a data rate of 200Mbps. It superimposes OFDM on the power line with an RF carrier from 2 to 28 MHz with up to1155 subcarriers.
2. HomePlugAV2 is another BPL and can achieve data rate of 1Gbps. It also superimposes an OFDM signal on the power line with an RF carrier from 30 to 86 MHz with up to 4096 subcarriers.
3. HomePlug Green PHY is designed for Smart Grid application for controlling HVAC, appliances, and smart electric meters for solar panels. It too uses OFDM and can achieve a data rate of 3.8Mbps.

The following URLs can be used for obtaining more information.

- www.wikipedia.org/wiki/IEEE_1901
- www.homeplug.org
- https://en.wikipedia.org/wiki/Power-line_communication

4.24 G.hn

G.hn is an ITU standard that is defined by ITU-T G9960. The G.hn is meant for multi-tenant/dwelling and home networking using a multiple input multiple output for diverse wiring options using a point-to-multipoint approach. G.hn is the next generation home audio visual (AV) network standard for distribution of internet protocol (IP) content across a diverse set of media including existing AC power lines, coax cables, twisted-pair telephone and fiber.

G.hn supports IoT through providing internet access as well as connecting various TV sets, DVD players, set-top boxes, other video equipment, appliances and sensors. G.hn uses OFDM signaling within its domain however it can also interface with other platforms using IP signaling.

A G.hn network is composed of multiple domains which can be established over any type of wiring power lines, coaxial cables, twisted pair and Plastic Optical Fiber. A domain is different media that is used in a G.hn network. Table 4.24.1 shows different media used by G.hn

Medium	Tethered/RF	Stat Freq MHz	Stop Freq MHz	Mbps
Power line (siso)	Tethered	2	25,50 or 100	250 -500
Power line (mimo)	Tethered	2	25,50 or 100	400-1000
Coax	Tethered	5	50 or 100	700-1000
Coax	RF	300	3000	800-1000+
Twisted Pair	Tethered	2	50 or 100	250-700
Fiber	Tethered	2	100 or 200	700-1000

Table 4.24.1 G.hn media

For distribution of broadband services in Multi-tenant/dwelling Units (MDUs) twisted pair wiring is a common media. Within a G.hn domain there can be 250 end points (EP). Within each domain there is a domain master (DM) and several domain masters are controlled by a global master (GM).

Figure 4.24.1 is an example of a G.hn architecture using a GM several DMs and EPs.

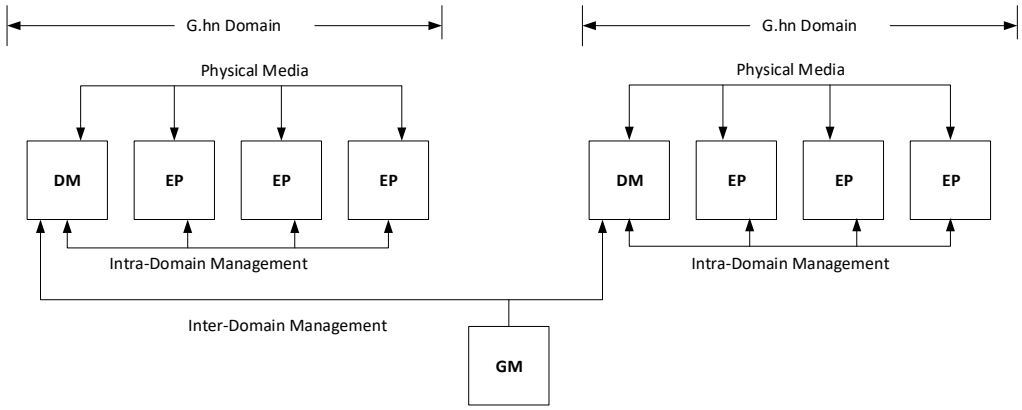

Figure 4.24.1 G.hn architecture

Figure 4.24.2 is a diagram depicting a G.hn network that has multiple media types within the MDU or home network.

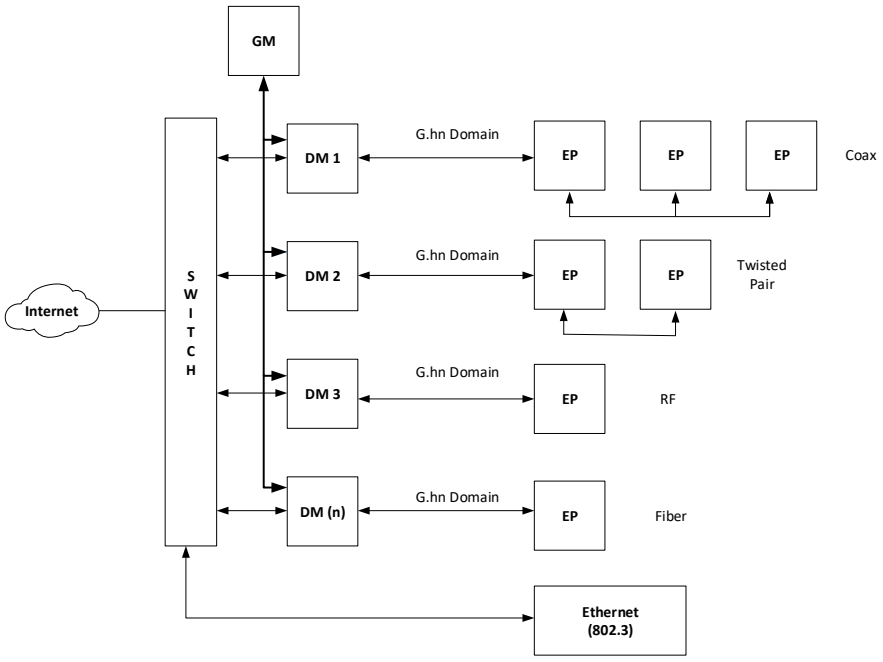

Figure 4.24.2 G.hn multiple media

The following URLs can be used for obtaining more information.

- www.wikipedia.org/wiki/G.hn
- https://www.broadband-forum.org

- http://www.homegridforum.org

4.25 Tethered Device Protocols

Table 4.24.1 highlights the differences between various wired protocols that are used.

Bus Technology	Standard	Power w/comm	Comm type	Data Rate	Max distance	# devices per segment	Comms Relationship
Foundation Fieldbus H1	IEC 61158	Y	Digital	31.25kbps	1.9km, 9.5km	32	Client/Server Pub/Sub Sink/Source
Foundation Fieldbus HSE	IEC 61158 IEEE 802.3	N	Digital	100M - 1Gbps	100m	unlimited	Client/Server Pub/Sub Sink/Source
Profibus PA	IEC 61158	Y	Digital	31.25k	1.9km, 9.5km	32	Master/Slave
Profibus DP	IEC 61158	N	Digital	9.6k-12Mbps	1512m	247	Master/Slave Pub/Sub
Modbus	V1.1b3	N	Digital	9.6k-12Mbps	1512m	247	Master/Slave Pub/Sub
ProfiNet	IEC 61158 PROFINET 2.4	N	Digital	100M - 1Gbps	100m	unlimited	Master/Slave
Hart	Bell202, 4-20mA	Y	Digital over analog	1.2-9,6k	3km	15	Master/Slave
ARCNET	ANSI 878	N	Digital	10Mbps	610m	256	Master Master
LonTalk	ANSI/CEA 709.1	N	Digital	1.25Mbp	500m	64	Peer to Peer

Note: all are IEC 61158 at the application layer with the exception of ARCNET and LonTalk.

Table 4.24.1: Tethered Device Protocols

I trust that you found this article useful.

Clint Smith, P.E.
Next G Connect
CTO
csmith@nextgconnect.com

Who we are:

NGC is a consulting team of highly skilled and experienced professionals. Our background is in wireless communications for both the commercial and public safety sectors. The team has led deployment and operations spanning decades in the wireless technology. We have designed software and hardware for both network infrastructure and edge devices from concept to POC/FOA. Our current areas of focus include 4G/5G, IoT and security.

The team has collectively been granted over 160 patents in the wireless communication space during their careers. We have also written multiple books used extensively in the industry on wireless technology and published by McGraw-Hill.

Feel free to utilize this information in any presentation or article with the simple request you reference its origin.

If you see something that should be added, changed or simply want to talk about your potential needs please contact us at info@nextgconnect.com or call us at 1.845.987.1787.