Jim Bougie

# Hands on Relay School
# INTRO TO COMMUNICATIONS

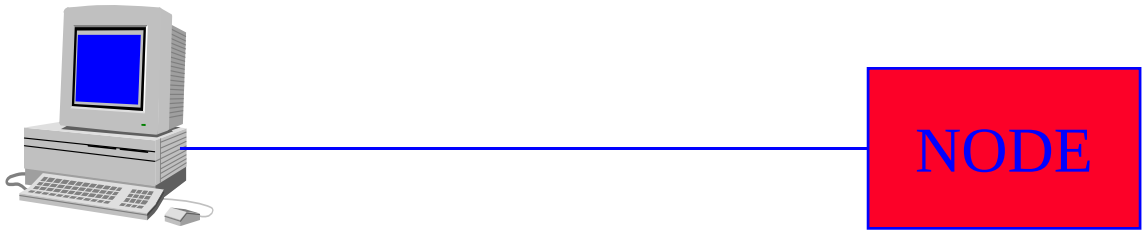
# Agenda

- Communications Media
  - RS 232
  - RS 485
  - Ethernet
- Protocols
  - Proprietary
  - Modbus
  - DNP
  - IEC61850

# RS 232

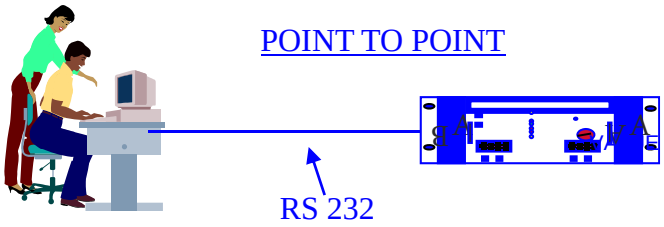
- Found in many formats, mainly 9 / 25 pin, but RJ45 has been used.
- Is voltage based.
- Is an TIA (Telecommunications Industry Association) Standard
- Was originally developed on EIA subcommittee TR30.2 on Interface.
- Latest revision as of July 2009 is TIA-RS232-F

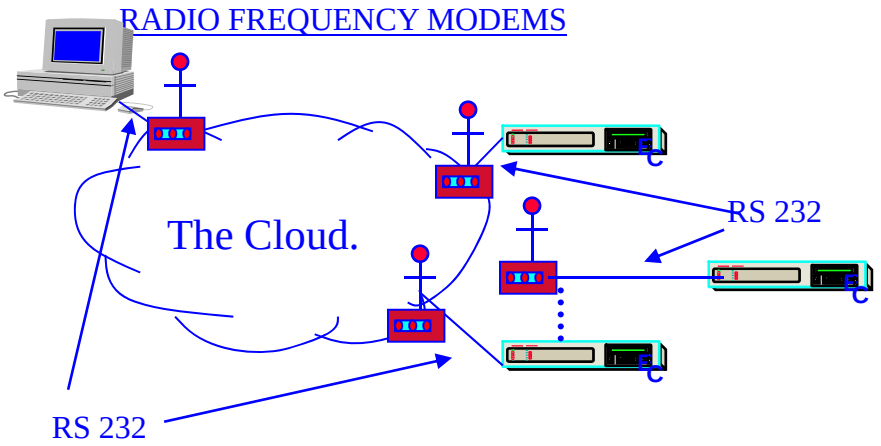
# RS 232

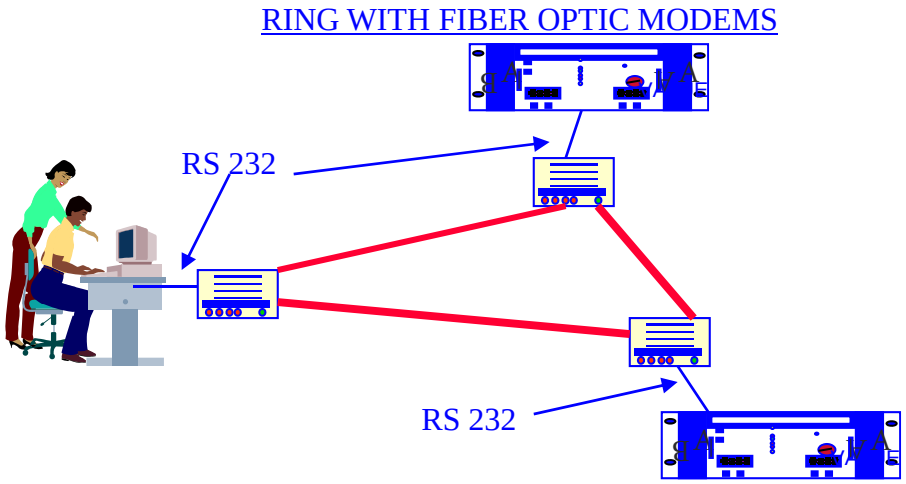



- RS 232 is a point to point connection network.
- Relays may have multiple RS 232 Ports, usually one reserve for local programming
- Is voltage based (referenced to a single common return [ground]).
- Is the most commonly used electrical interface.

# Typical RS 232 Devices



# RS 232 ANSI SPEC

- 2 Emulation's:
  - DTE - Data Terminal Equipment
    - Examples -
      - Personal Computer.
  - DCE - Data Communication Equipment.
    - Examples
      - Modem (Automatic Calling Equipment)

# SPEED RS 232

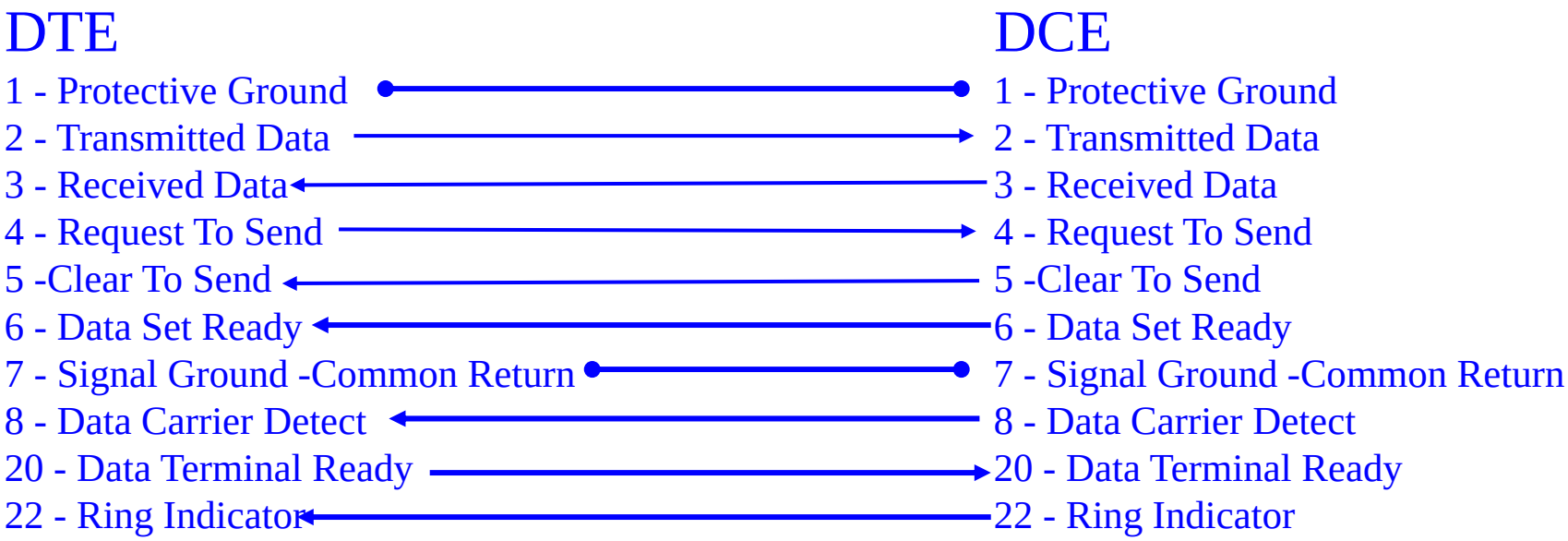
- Modem Speeds are from:
  - 110 baud (Not Really Used)
  - 115.2 K baud

- A Baud is a bit representation.
  - One Baud does not necessarily mean 1 bit.
  - One Baud means one change of state of the line.
  - Baud rate = 9600 = 1 change of state every 100 micro seconds.

# RS 232 Connector

- Connector Style is not specified.
  - Originally Specified for 25 pins.
  - IBM developed de-facto standard of 9 Pins
- Physical Interface Connector
  - DB 25
  - DB 9
  - Screw Terminals
  - RJ 11 Telephone Connectors.
  - RJ 45 Ethernet Connectors

# RS 232 DB 25 Wiring Diagram




| | | |
|---|---|---|
| RTS - | Space= Transmit mode | Mark = Receive Mode |
| CTS - | Space = Send Data | Mark = Do Not Send Data |
| DSR - | Space = Device Off Hook | Mark = Device ON Hook |
| DTR - | Space = Device On Line | Mark = Device Off Line |
| DCD - | Space = Good Data | Mark = Error In Data |
| RI - | Mark = Phone Ringing | Space = Not Ringing |

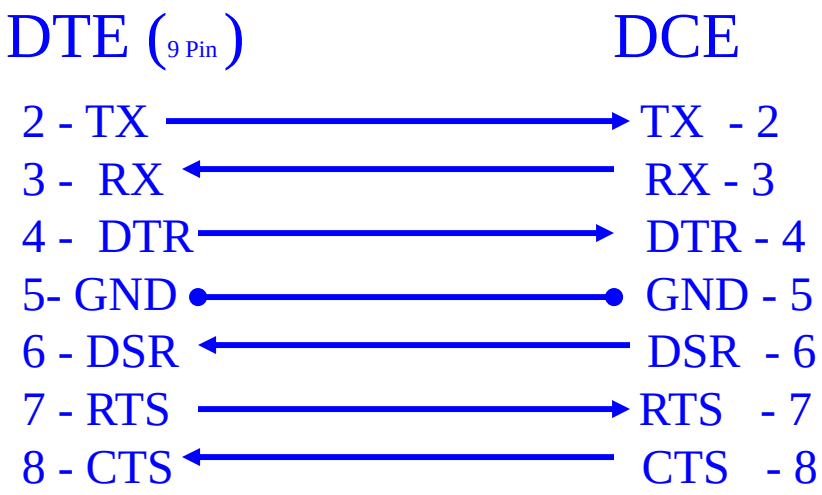
NOTE: both RTS/CTS and DTR/DSR are rarely used

# RS 232 (DB 9)

DTE ●——● DCE

Note: If both devices are DTE's or DCE's re-pin the cable as necessary!

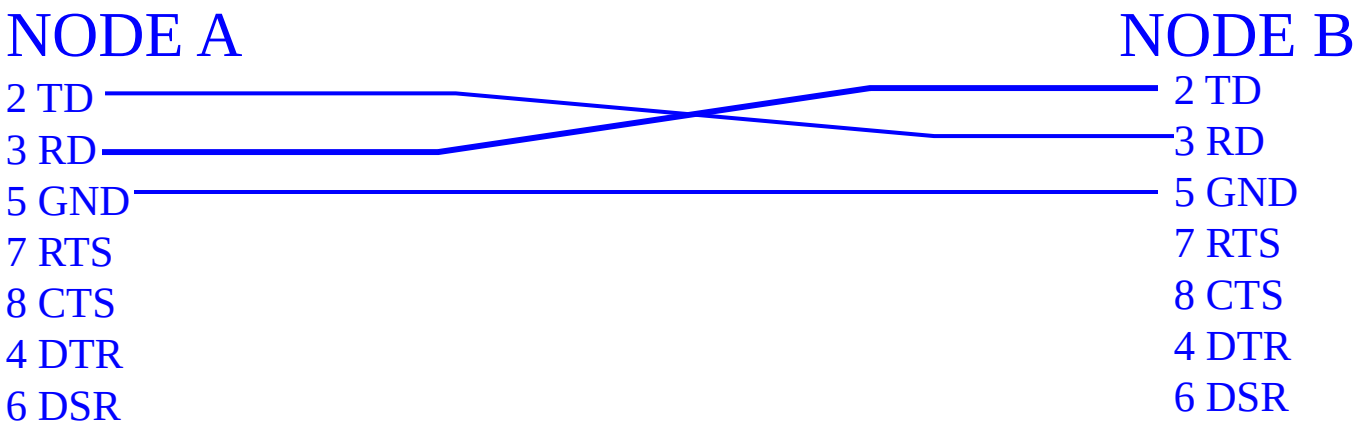
50 feet cable length maximum



# RS 232 - Physical Interface

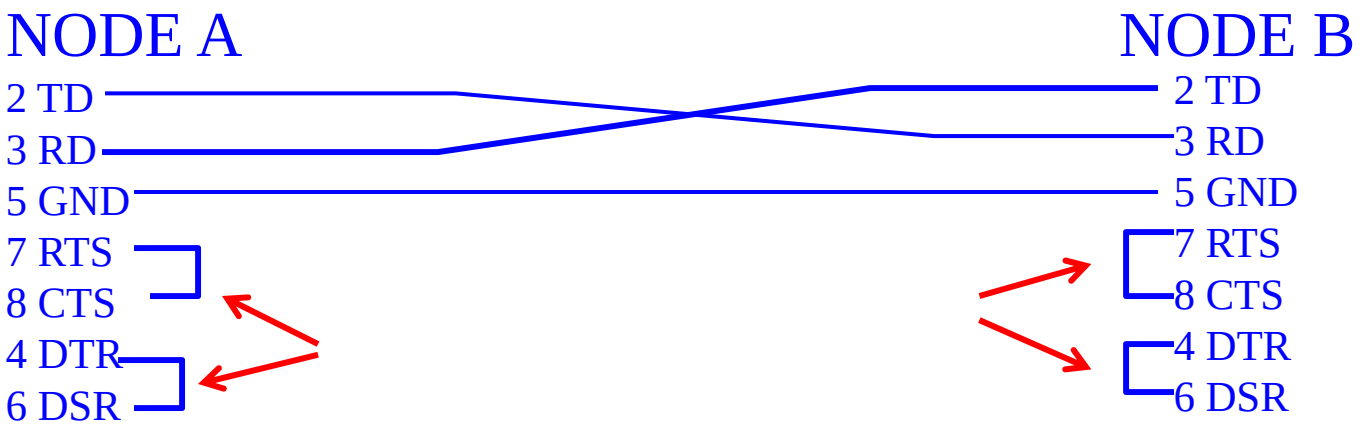
- Signal
  - Transmit (TX)
  - Receive ( RX)
  - Ground (GND)
- Control
  - Clear To Send ( CTS)
  - Request To Send ( RTS)
  - Data Set Ready ( DSR)
  - Data Terminal Ready ( DTR)
  - Carrier Detect (CD)

# What is a NULL MODEM Cable?




- There are times to connect DTE - DTE or DCE-DCE.

# Why the Jumpers?




- There are times when handshaking is required by the software and not by the hardware.

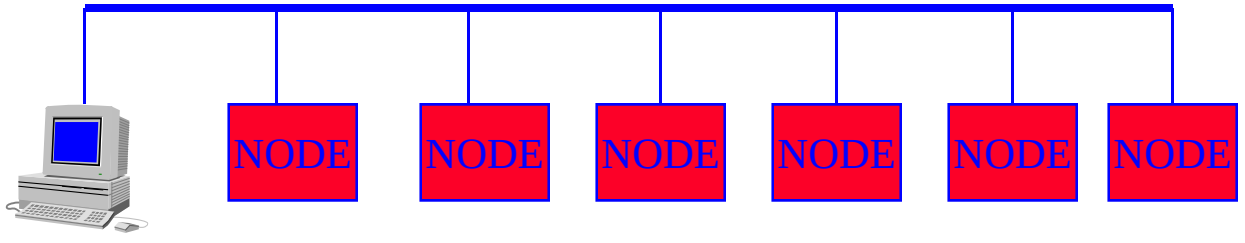
# RS 232
# Advantages/Disadvantages

- Advantages
  - Easy to implement
  - Easy to troubleshoot
  - Designed for “long-range” communication equipment

- Disadvantages
  - Susceptible to noise
  - Relativity short distances
  - Designed for single devices

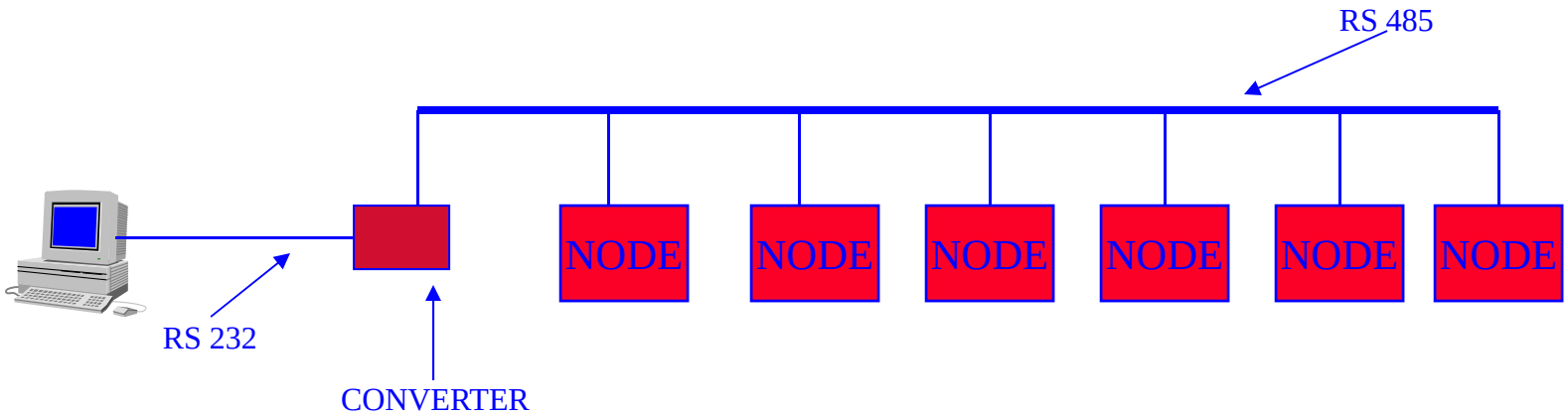
# RS 485

In Contrast to RS 232, RS 485 allows interconnection of multiple devices.




- ADVANTAGES
  - EASY TO IMPLEMENT
  - HIGH EXPANDABILITY
  - Less Susceptible to noise
- DISADVANTAGES
  - HIGHER WIRING COSTS
  - MORE DIFFICULT TO TOUBLESHOOT

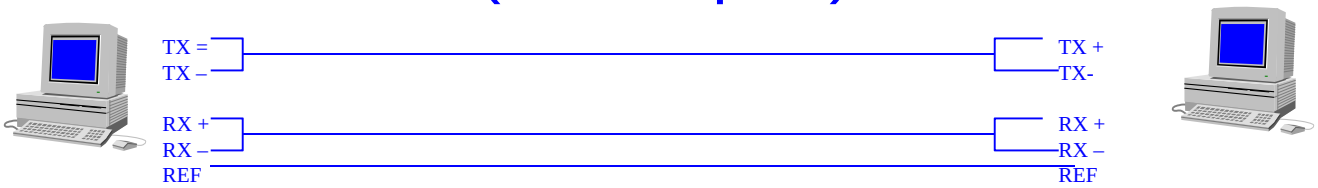
# AN IBM PC HAS AN RS 232 PORT



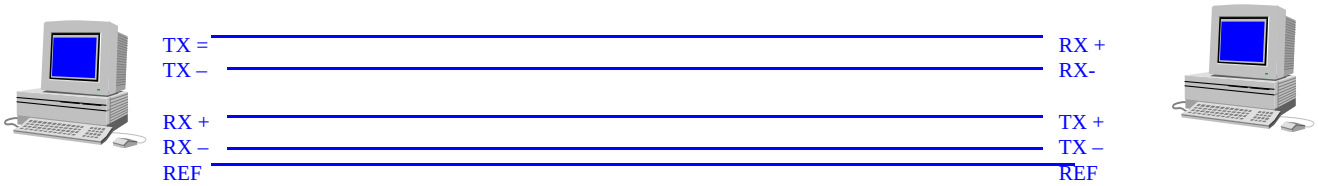

- A Converter may be needed to transform the RS 232 Interface to an RS 485 Interface.
- Many Manufacturers of interfaces are available.

# RS 485

- 2 Variants of RS 485
  - 2 Wire (Half Duplex)




  - 4 Wire ( Full Duplex)



# Terminology

- HALF DUPLEX- data can be transmitted in both directions, but not at the same time.
- FULL DUPLEX- data can be transmitted in both directions (TX/RX) at the same time.

# RS 485 Loading

- Balanced communication
  - Sensed between + and -
    - A, B
    - +, -
  - Able to connect up to 32 Loads
    - The amount of data can affect polling times
  - A Terminal negative with respect to B Terminal
    - 1 or MARK
  - A Terminal positive with respect to B terminal
    - 0 or SPACE

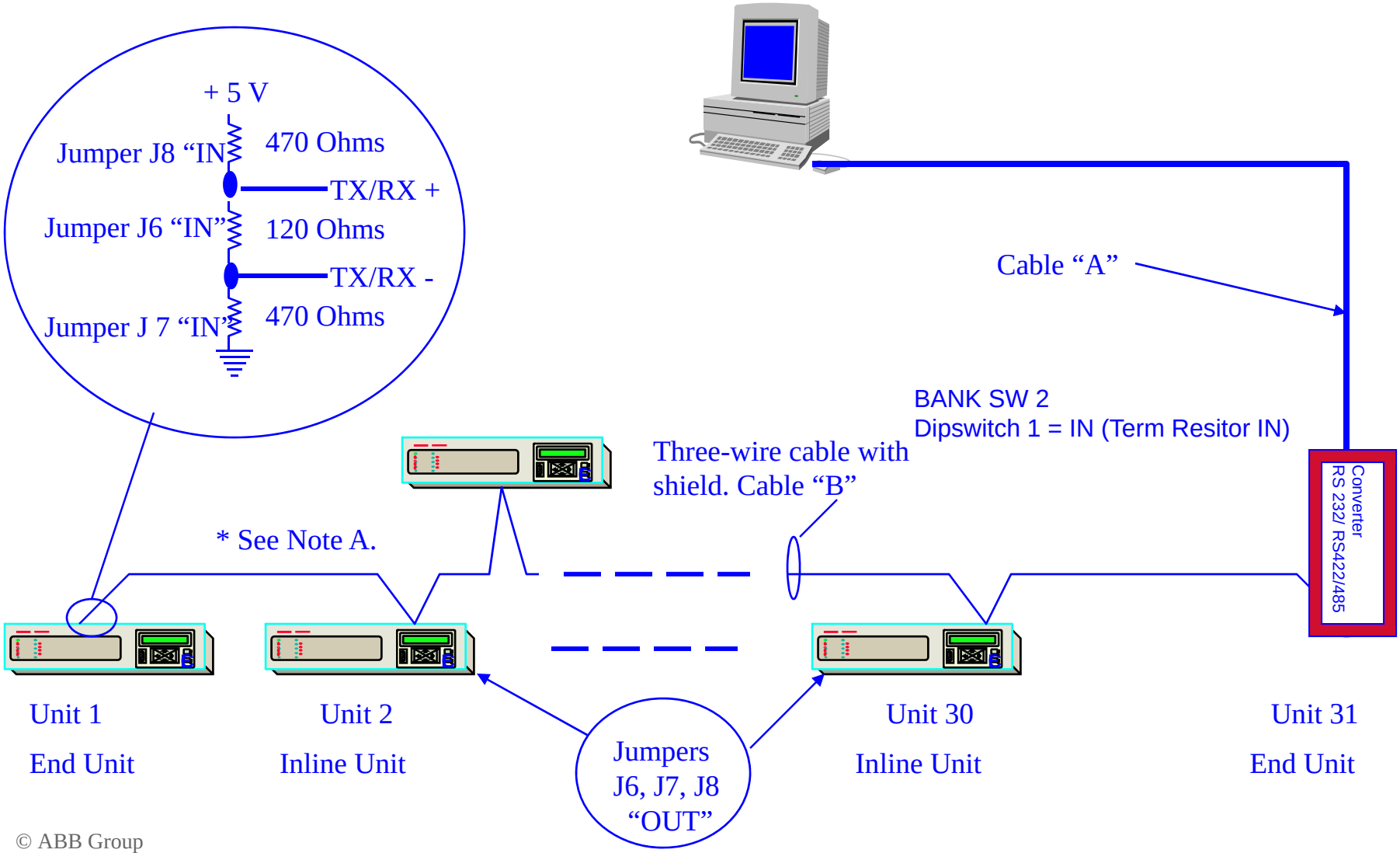
# RS 485 Loading

- A Driver must be able to drive:
  - Impedance of 60 ohms (54 ohms worst case).
- Check Manufacturer Recommendations ( depending on cable) 1000 ft. drive distance - 4000 ft. max.

# What About Grounding?

- It is recommended that a shield is terminated at one point.
- Some relays have isolated ports, This may require a separate GROUND conductor interconnecting each node on the cable.
- "RS 485 EIA Spec states: "The circuit reference may be established by a third connector connecting the common leads of the equipment OR it may be provided by connections in each using equipment to an earth reference."

# Topology Diagram for RS 485 Multi-drop Architecture



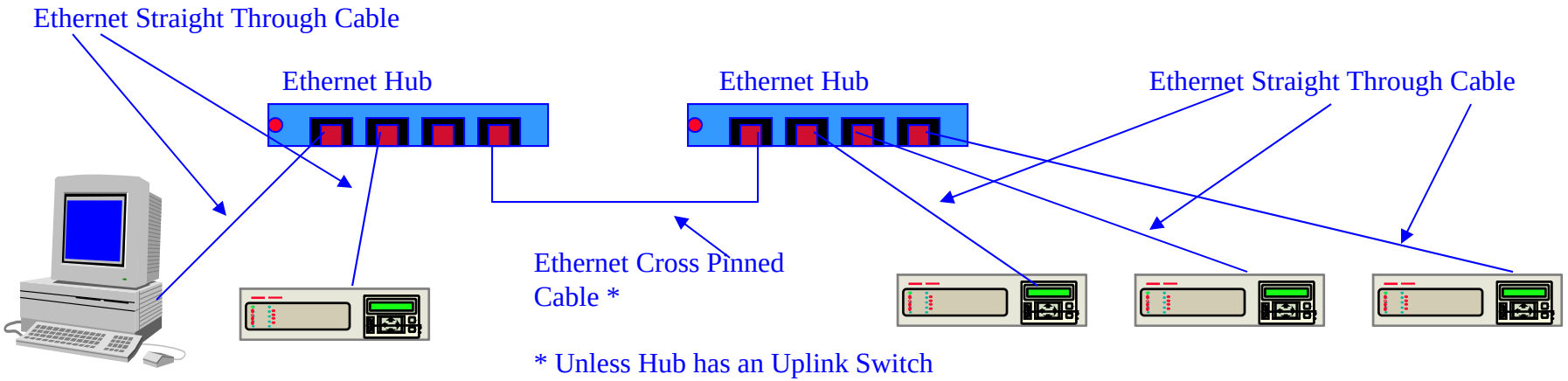
# Ethernet

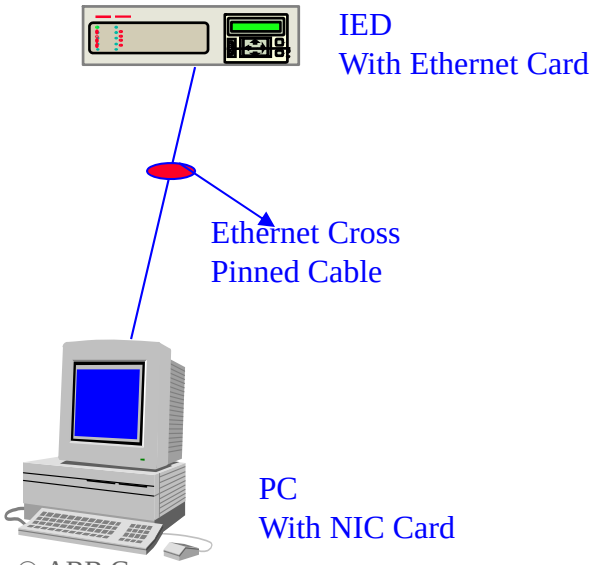
- IEEE 802.3
- Ethernet nodes are typically connected via copper (CAT 5 Cable).
- Ethernet nodes found in substations are usually connected via Fiber Optics.
- Copper: 10baseT, 100baseTX
  - RJ45 Connector
- Fiber: 100baseFX
  - Numerous connectors styles
    - Make sure you know which one you need

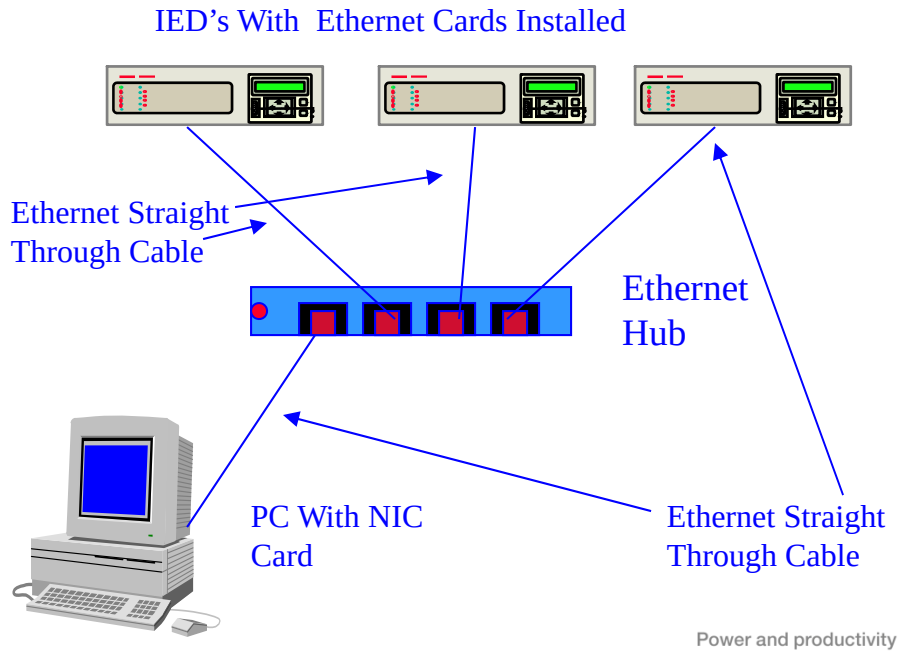
# Cable Connection

- There are two types of Copper Ethernet ports:
  - MDIX :Typically what a Hub or Switch uses.
  - MDI : Typically what a NIC card uses.
- A typical Ethernet Hub emulates the MDI interface.
- This means for interconnection one must know what type of CAT 5 Cable is required:
  - Straight Through Pinout.
  - Cross Pinned.
- Most modern Ethernet devices have auto sense capability eliminating the need for cross over cables

# Ethernet Copper Connectivity



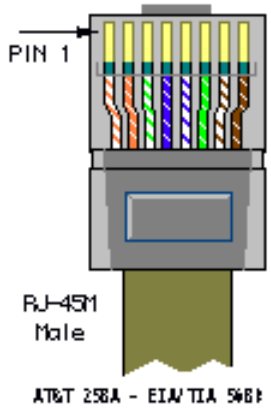



# CAT 5 Cable Pinouts

**Category 5 wiring standards:**
EIA/TIA 568A/568B and AT&T 258A define the wiring standards and allow for two different wiring color codes.

| Crossover Cable | |
|---|---|
| **RJ-45 PIN** | **RJ-45 PIN** |
| 1 Rx+ | 3 Tx+ |
| 2 Rc- | 6 Tx- |
| 3 Tx+ | 1 Rc+ |
| 6 Tx- | 2 Rc- |



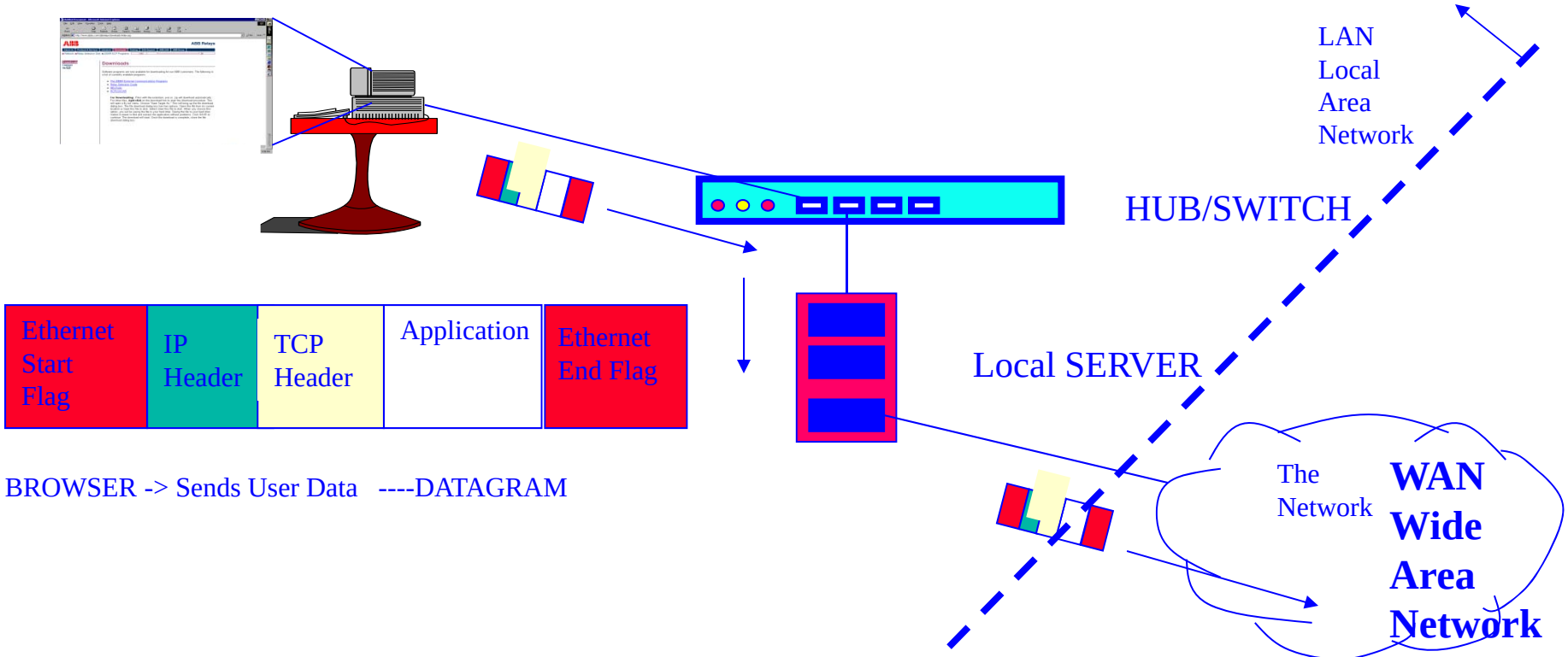

| Straight Through Cable | |
|---|---|
| **RJ-45 PIN** | **RJ-45 PIN** |
| 1 Tx+ | 1 Rc+ |
| 2 Tx- | 2 Rc- |
| 3 Rc+ | 3 Tx+ |
| 6 Rc- | 6 Tx- |

•Pairs may be solid colors and not have the stripe.
•Category 5 cable must use Category 5 rated connectors.

| Pin # | EIA/TIA 568A | AT&T 258A, or EIA/TIA 568B | Ethernet 10BASE-T | Token Ring | FDDI, ATM, and TP-PMD |
|---|---|---|---|---|---|
| 1 | White/Green | White/Orange | X | | X |
| 2 | Green/White | Orange/White | X | | X |
| 3 | White/Orange | White/Green | X | X | |
| 4 | Blue/White | Blue/White | | X | |
| 5 | White/Blue | White/Blue | | X | |
| 6 | Orange/White | Green/White | X | X | |
| 7 | White/Brown | White/Brown | | | X |
| 8 | Brown/White | Brown/White | | | X |


Courtesy of Enterasys Knowledge @ http://knowledgebase.enterasys.com/esupport/esupport

# General Ethernet Architecture




- The Internet uses Internet Protocol – TCP/IP. The message is layered and sent to gather request data

# The Layers of the OSI Model

- Layer 1 – Physical Layer
  - The actual hardware
- Layer 2 – Data Link Layer
  - Data Transfer Method
    - Frames the data
    - Assures error free transmission
    - Timing
  - Logical Link Control (LLC)
    - Maintains the link between two computers
  - Media Access Control
    - Used to send data between two computers
    - Hardware address

# The layers of the OSI Model

- Layer 3 – Network Layer
  - IP network protocol. Routes messages using the best path available
- Layer 4 – Transport Layer
  - TCP, UDP. Ensures properly sequenced and error free transmission.
- Layer 5 – Session Layer
    - User interface to the Network
    - Determines when the session is begun or opened, how long it is used, and when it is closed
    - Controls the data
    - Supports Security
    - Supports name look up

# The layers of the OSI Model

- Layer 6 – Presentation Layer
  - Makes the type of data transparent to the layers around it.
  - Used to translate data to computer
- Layer 7 – Application Layer
  - Provides services software applications need
  - Email, DNP, etc.

# Understanding the TCP/IP Model

- There are 4 interconnected layers:
  - Application (Modbus, DNP)
  - Transport (TCP, UDP)
  - Internet (IP address)
  - Network Access (media, MAC)

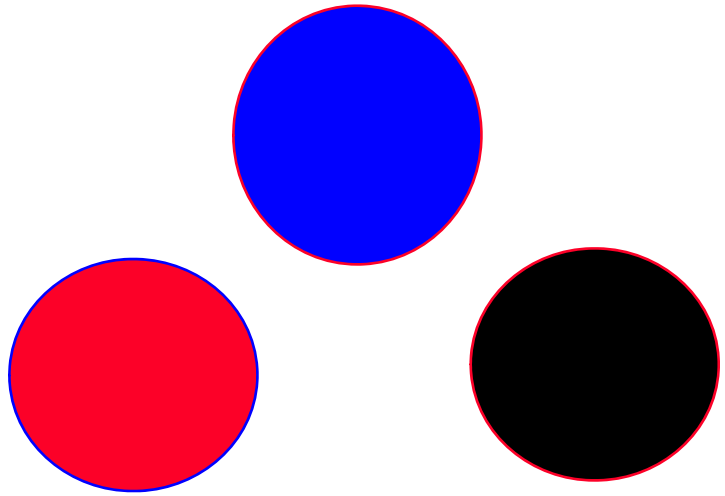
# IP addressing

- Network Address
  - Identifies the network
- Host Address
  - Identifies a device inside a network

IP address: 192.168.1.5

Subnet mask: 255.255.255.0

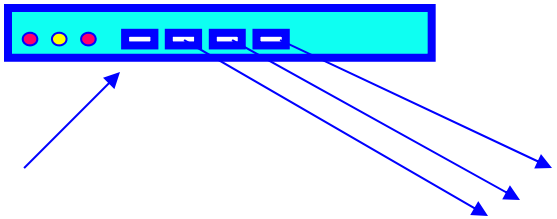
Gateway: 192.168.1.1



192.168.2.5

192.168.3.5

# HUB

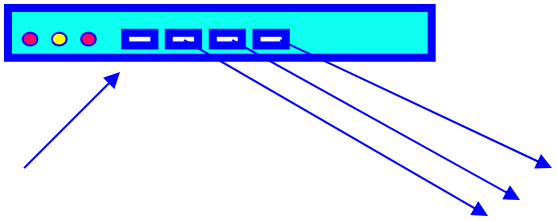


Message Generated

Message Regenerated to each port.

- Ethernet is in essence a Point to Point Interface.
- If 2 Devices are connected a cross-pinned cable may be necessary for interconnection.
- If more than 2 devices are connected, a hub/switch is required

# HUB



Message Generated

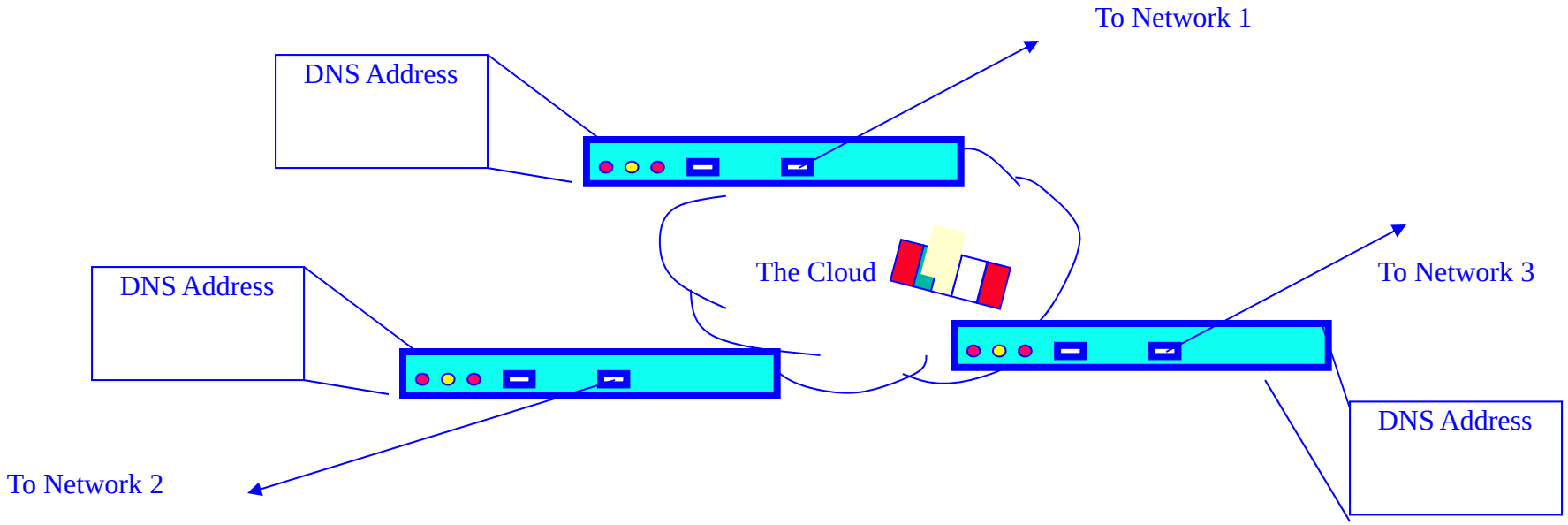
Message Regenerated to each port.

- Performance of Hub deteriorates on large network because of traffic.

# Switch

- A Switch is a device that channels incoming data from any of multiple input ports to the specific output port that will take the data toward its intended destination.
- Performs Layer 2 of the OSI layer functionality or Network Layer of the Ethernet Model.

# Router




A router is a device or, in some cases, software in a computer, that determines the next network point to which a datagram should be forwarded toward its final destination.

# Agenda

- Communications Media
  - RS 232
  - RS 485
  - Ethernet
- **Protocols**
  - Proprietary
  - Modbus
  - DNP
  - IEC61850

# Propriety Protocols

- 1st of Protocols
  - Used were for SCADA Systems
    - All components were one manufacturer
    - Slow Baud rates
    - Only basic information available

# Propriety Protocols

- 2nd generation of protocols
  - Used for programming equipment
    - Electronic equipment
  - Used by the 1st integration systems
    - Was available
  - More information was available
    - But only limited amount sent to SCADA

# Propriety Protocols

- Automation Schemes
  - Before all automation was hardwired
  - Electronic Devices able to talk to each other
    - Reduced wiring
    - Longer distances
      - Communication systems
  - Limited to one manufacturer
    - Led to “universal” protocols
      - Modbus
      - DNP
      - IEC61850

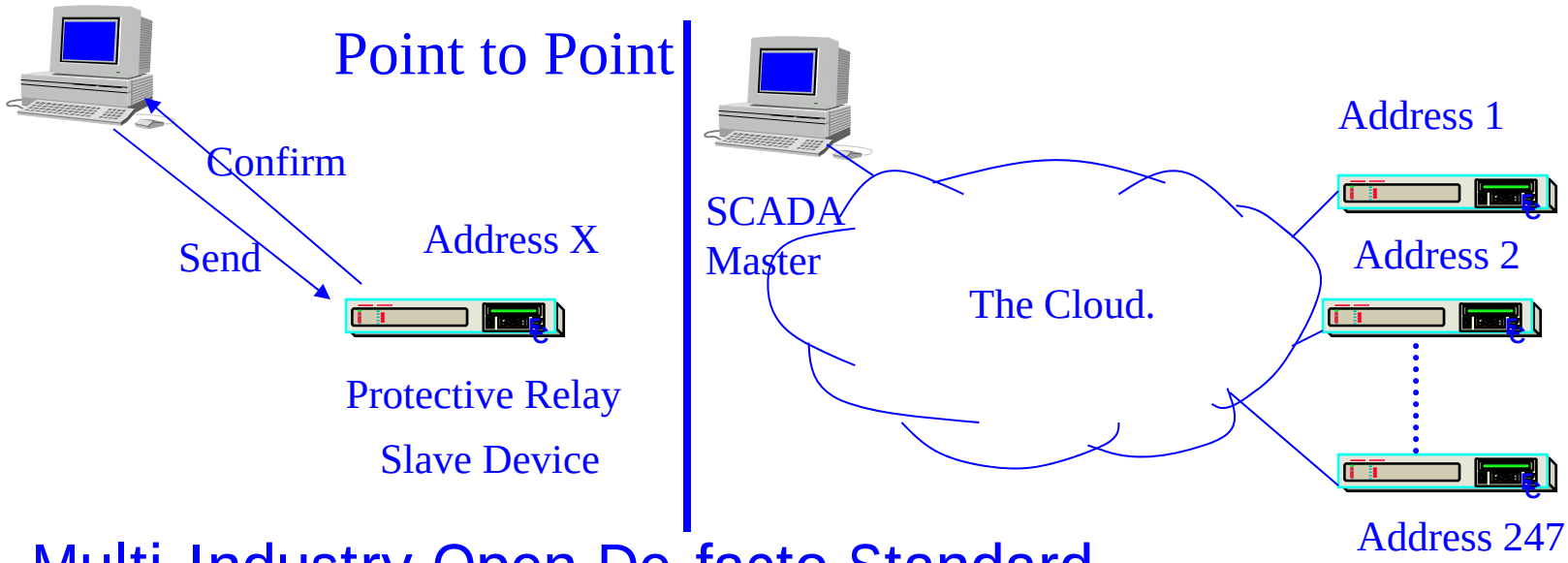
# Modbus

- Modbus was invented by Modicon Inc. In 1978.
  - As a method to connect PLC's to a host (Master/Slave or Parent/Child)
  - Easy to implement with two emulations:
    - RTU ( Remote Terminal Unit) Emulation
    - ASCII Emulation
- Modbus is available through several Physical Interfaces (RS232/RS485/Ethernet, etc).

# What Makes Modbus a Non-Utility Protocol

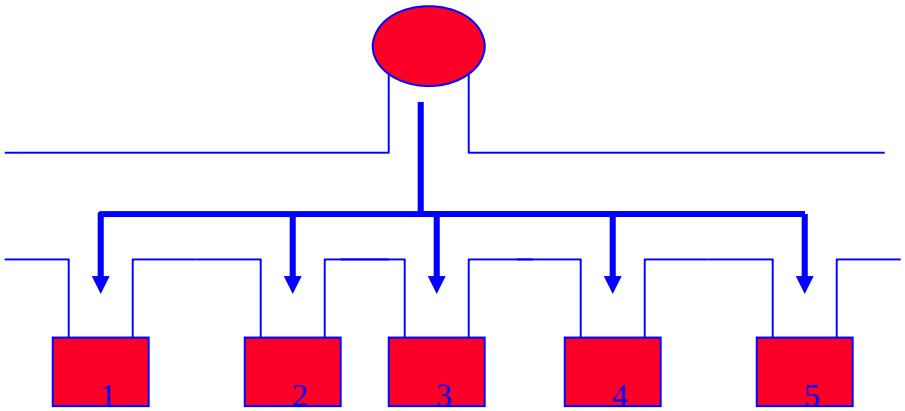
- It has no Time Synch imbedded in the protocol.
- It has no concept of frozen points.
- It has no concept of select before operate.
- The manufacturer or implementer of the protocol must engineer these features into the protocol/device.

# Modbus Protocol

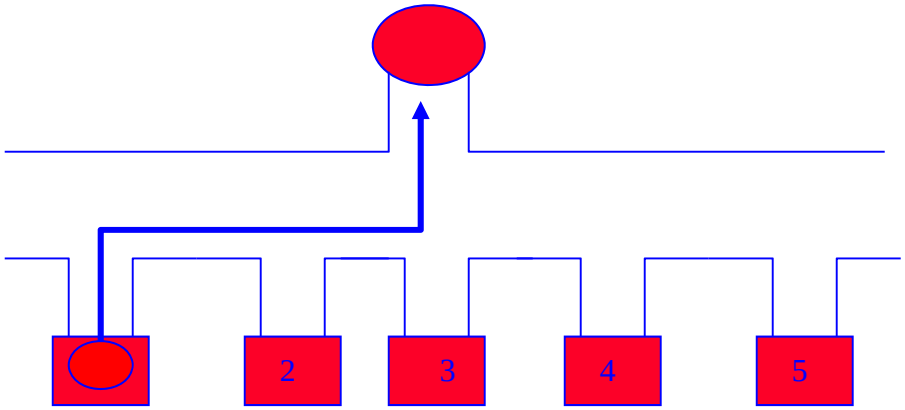



- Multi-Industry Open De-facto Standard.
- Master-Slave Protocol.
- Two Emulations
  - Modbus ASCII ( Master/Slave Mode) - 10 bit Asynchronous
  - Modbus RTU (Master/Slave Mode) - 11 bit Synchronous

# Modbus Example




**The Master node (Circle) contains a polling list. The master transmits its request to a specific node and waits for a response. All nodes hear the transmitted request.**




**The addressed Slave responds with the information. If the slave data cannot be transmitted immediately, a not ready response is generated and the Master must poll the Slave again with the same request.**

1

# Modbus Command Format

**Data Sent From Master**


**Data Received From Master**


( Device Address = 0 ( Broadcast), 1 - 247)

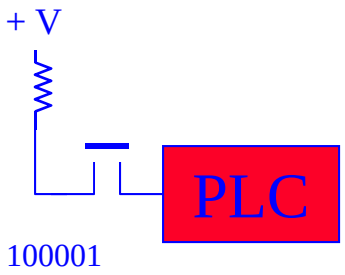
# Modbus Emulation

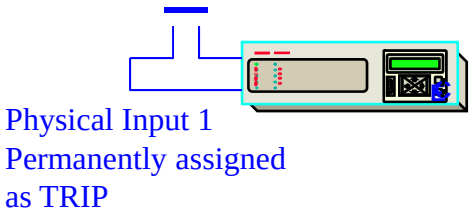
- ASCII Mode-
  - Asynchronous Communication
  - Hexadecimal ASCII Characters 0-9, A-F ( 30 - 39, 41,46)
  - 10 bit protocol
    - 1 Start Bit
    - 7 Data Bits
    - 1 Parity ( if enabled)
    - 1 Stop Bit ( if Parity) or 2 Stop Bits ( if no Parity enabled).
  - Longitudinal Redundancy Check

# Modbus Emulation

- RTU Mode
  - Synchronous  Communication
  - Data 8 Bit Binary, Hexadecimal 0 - 9, A- F
  - 11 Bit Protocol
    - 1 Start Bit
    - 8 Data Bits, LSB sent first
    - 1 Bit Parity ( if selected)
    - 1 Stop Bit ( if Parity) or 2 Stop Bits ( if No Parity Selected
  - CRC-16 Error Check

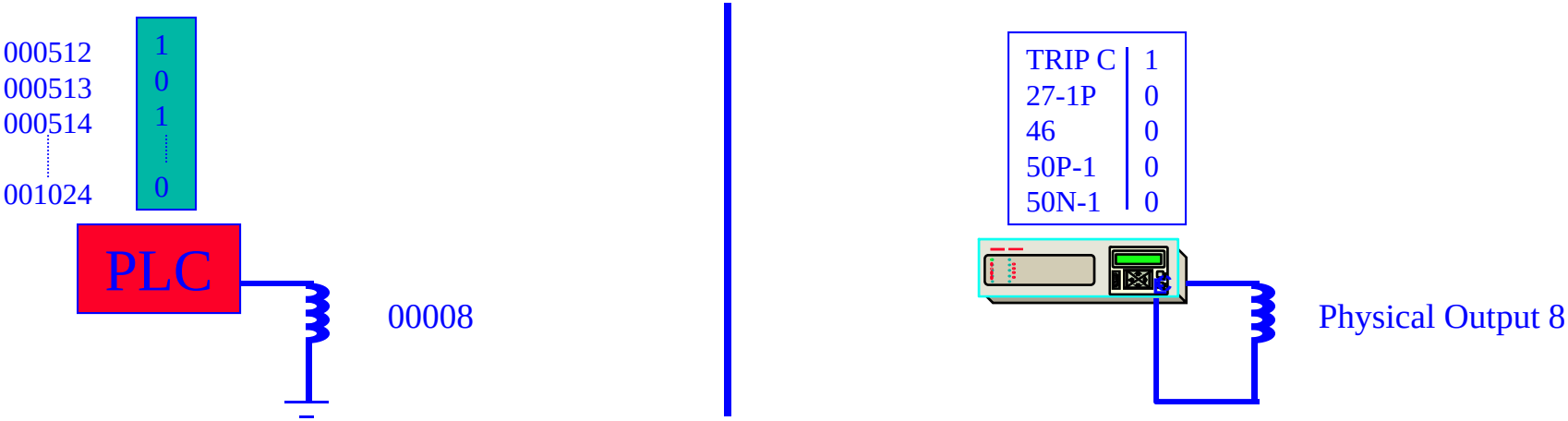
# 1 XXXX Memory



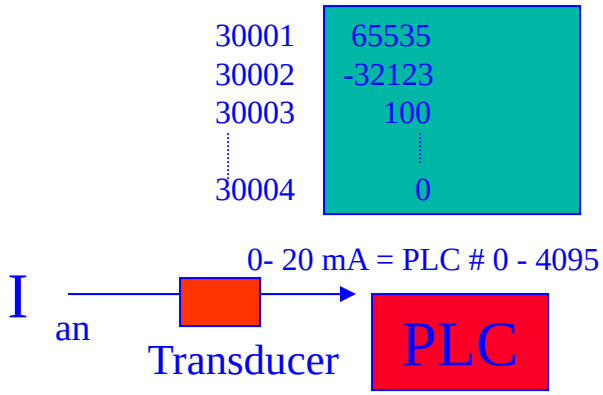

- PLC 1 XXXX memory is analogous to the Physical Inputs on a protective relay.
- 1 XXXX memory is a discrete bit.
- PLC's may have XXXX = 16 to 65535 discrete inputs per device ( 1 X memory).

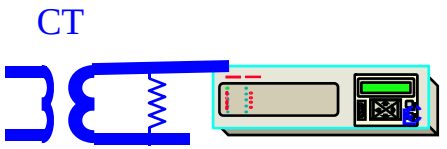
# 0 XXXX Memory




- PLC 0 XXXX memory has duality:
  - Internal Memory bit-wide
  - Output memory
- Many protective relays have similar capability.
  - Internal memory is analogous to ULO [User Logical Inputs/Outputs]
  - Output memory is analogous to the physical outputs on the relay.

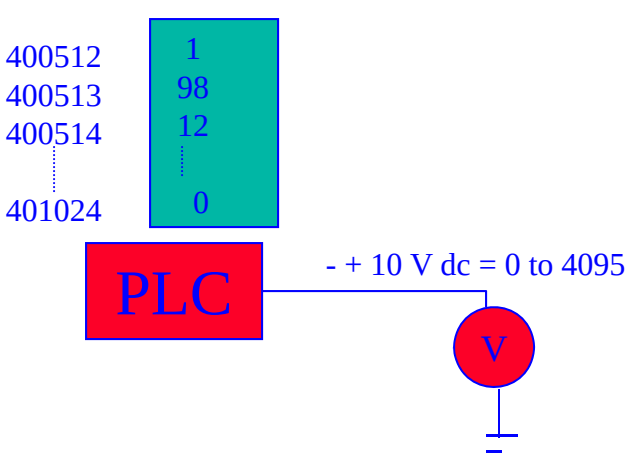
# 3XXXX Memory




- A relay may have physical inputs matching the 3XXXX register definition.
- 3 XXXX data is defined as a word wide Physical Input from the field mapped to memory.

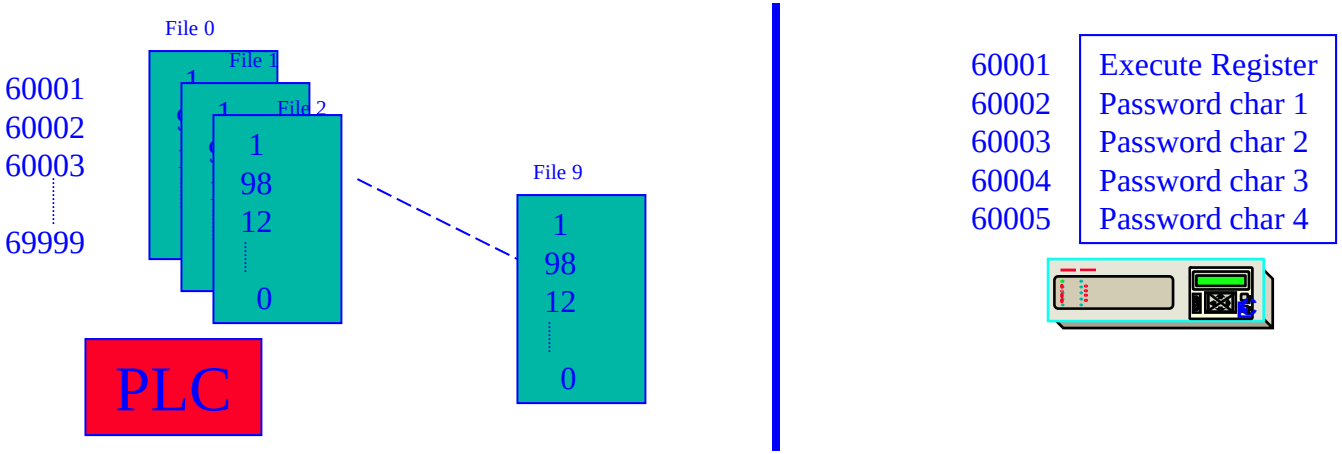
# 4XXXX Memory




| | |
|---|---|
| Fault Number | 1 |
| Year | 98 |
| Month | 12 |
| Day | 11 |
| Hours | 23 |

- PLC 4 XXXX memory has duality:
  - Internal Memory word-wide
  - Output memory
- Many protective relays have similar capability.
  - Internal memory is analogous to metering and fault capabilities of the relay.
  - 4X Physical Output mapping is not applicable for the protective device.

# 6 XXXX Memory

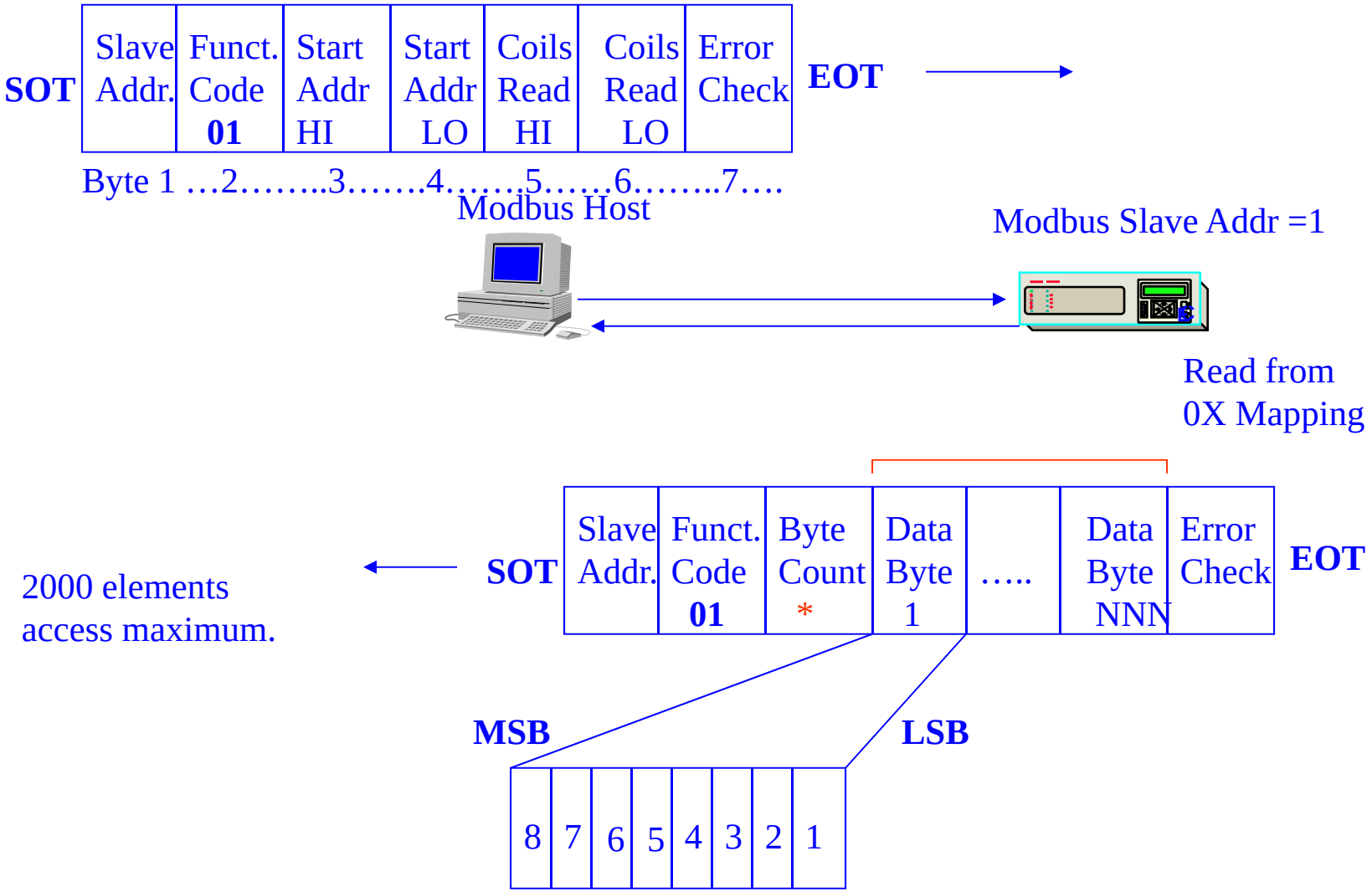



- 6XXX memory is defined as extended memory. Some PLC's have this memory. It is able to be paged in to 4 XXXX memory.
- A few protective relay store configuration parameters in this memory area for network access.

# Function 01 -Read Coil Status

- Reads 0X ( Coil) references from the slave.
- All bytes are in hex ( coding is dependent on RTU or ASCII emulation).
- Memory Start Address is offset by one.
- If amount of data is not a multiple of 8, most significant bits are padded with 0's.

# Function 01 - Read Coil Status



# What Happens if ……?

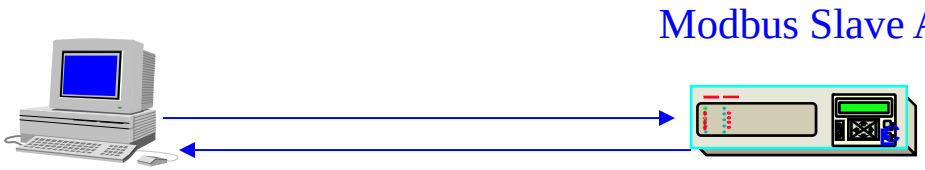
- The issue with static data is:
  - What happens between access reads of the IED?
    - If something changes?
    - If something doesn't change?
- IF two changes occur during read the event is lost using static data.
  - Breaker Trips - Pickup Alarm (PUA) energizes and de-energizes briefly.

# How is this anomaly resolved?

- Latched Bits (which can be reset via a control write)
- Momentary Change Detect Bits ( which change status is reset on a read of the element).
- This is a manufacturer's function and not one of the protocol.
- NOT ALL MODBUS IMPLEMENTATIONS ARE ALIKE.

# Function 01 - Read Coil Status

## Example - Read Output 1-6, with two bit status.




Read from
0X Mapping

Obtain Output 8 Through Output 3 Status Indication (01037 to 01048 per the memory map).

Host Sends : 01 01 04 0C 00 00 14 - - = LRC or CRC Code

Addr = 01
Function = 01
Address = 1037 ( which is 1036 in hex = 040C)
Amount of Data Requested = 12 Coils

Relay Responds: 01 01 02 A1 02 -

Addr = 01
Function = 01
Data Bytes Received = 2
Data Received = A1 02

# Function 02 -Read Input Status

- Reads 1X ( Input) references from the slave.
- All bytes are in hex ( coding is dependent on RTU or ASCII emulation).
- Memory Start Address is offset by one.
- If amount of data is not a multiple of 8, most significant bits are padded with 0's.

# Function 02 - Read Input Status

| SOT | Slave Addr. | Funct. Code **02** | Start Addr HI | Start Addr LO | Coils Read HI | Coils Read LO | Error Check | EOT |
|---|---|---|---|---|---|---|---|---|

Byte 1 …2……..3…….4……5……6……..7….

Modbus Host

Modbus Slave Addr =1

2000 bits maximum.

Read from 1X Mapping

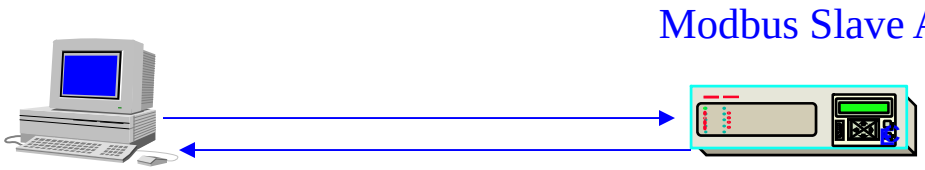
| SOT | Slave Addr. | Funct. Code **02** | Byte Count * | Data Byte 1 | ….. | Data Byte NNN | Error Check | EOT |
|---|---|---|---|---|---|---|---|---|

**MSB** **LSB**

| 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
|---|---|---|---|---|---|---|---|

# Function 02 - Read Input Status

## Example - Read User Logical Input 1-6, with two bit status.




Obtain ULI1 Through ULI 6 Status Indication (10559 per the memory map).

Host Sends : 01 02 01 2E 00 14 - - = LRC or CRC Code

Addr = 01
Function = 02
Address = 559 ( which is 558 in hex = 012E)
Amount of Data Requested = 12 Inputs

Relay Responds: 01 02 02 A1 02 -

Addr = 01
Function = 01
Data Bytes Received = 2
Data Received = A1 02

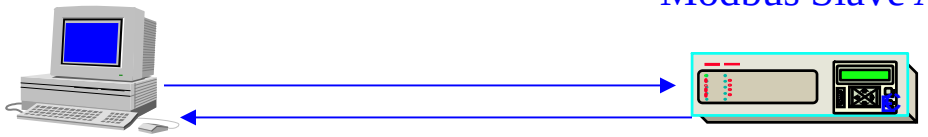
# Function 03 -Read Holding Registers

- Reads 4X holding registers from the slave.
- All bytes are in hex ( coding is dependent on RTU or ASCII emulation).
- Memory Start Address is offset by one.
- Data is returned in register format ( 16 bits/2 bytes per register).
- Maximum registers read are 125 per query.
- Registers are sent Hi byte- Lo byte per register.

# Function 03 - Read Holding Registers

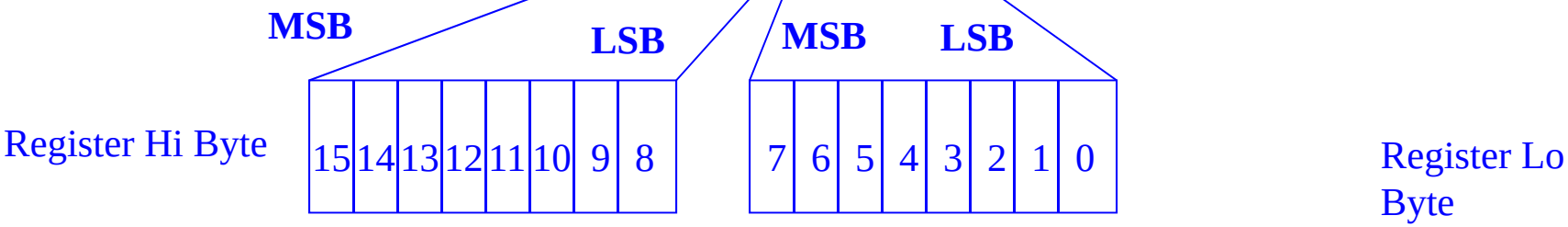
| SOT | Slave Addr. | Funct. Code **03** | Start Addr HI | Start Addr LO | Regs Read HI | Regs Read LO | Error Check | EOT |
|---|---|---|---|---|---|---|---|---|
| | Byte 1 | …2 | ……..3 | …….4 | ……5 | ……6 | ……..7 | …. |

Command
Allows for
125 Registers
Max.




Read from
4X Mapping

| SOT | Slave Addr. | Funct. Code **03** | Byte Count * | Data Byte Hi | Data Byte Lo | ---------- | Data Byte Lo | Error Check | EOT |
|---|---|---|---|---|---|---|---|---|---|



# DNP PROTOCOL HISTORY

- Created by Westronics (Now GE ) in 1990
- Released into Public Domain in 1993.
- Users Group created in 1993.
- DNP Technical Committee Created in 1995
  - Published Subset Documentation
  - Established Parameters for future protocol conformance committee

# DNP 3.0

- Dependent on the Implementation DNP 3.0 Can:
  - Request and Respond with Multiple Data Messages in a single message.
  - Segment messages into multiple frames.
  - Respond with changed data.
  - Request data based on data priority
  - Support time synchronization
  - Allow multiple masters and peer to peer operation
  - Allow user defined objects and file transfer.

# DNP 3.0

- DNP 3.0 Supports the ISO OSI (International Standard Organization Open Systems Interconnect) Model. Layers fully supported are:
  - Physical (Layer 1)
  - Data Link (Layer 2)
  - Application (Layer 7)
- Pseudo-Supported and Defined Layers are:
  - Transport (Layer 4)

# DNP - Definition of Terms

- Object Categories - Data which conforms to different data types:
  - Static : Current Value of Field or Software Point.
  - Event: Historical Data.
  - Frozen Static: A Field or Software Value which is not actively updated due to a Data Freeze Request.
  - Frozen Event:  Data generated as a result of a Data Freeze Event but historically archived upon a change.

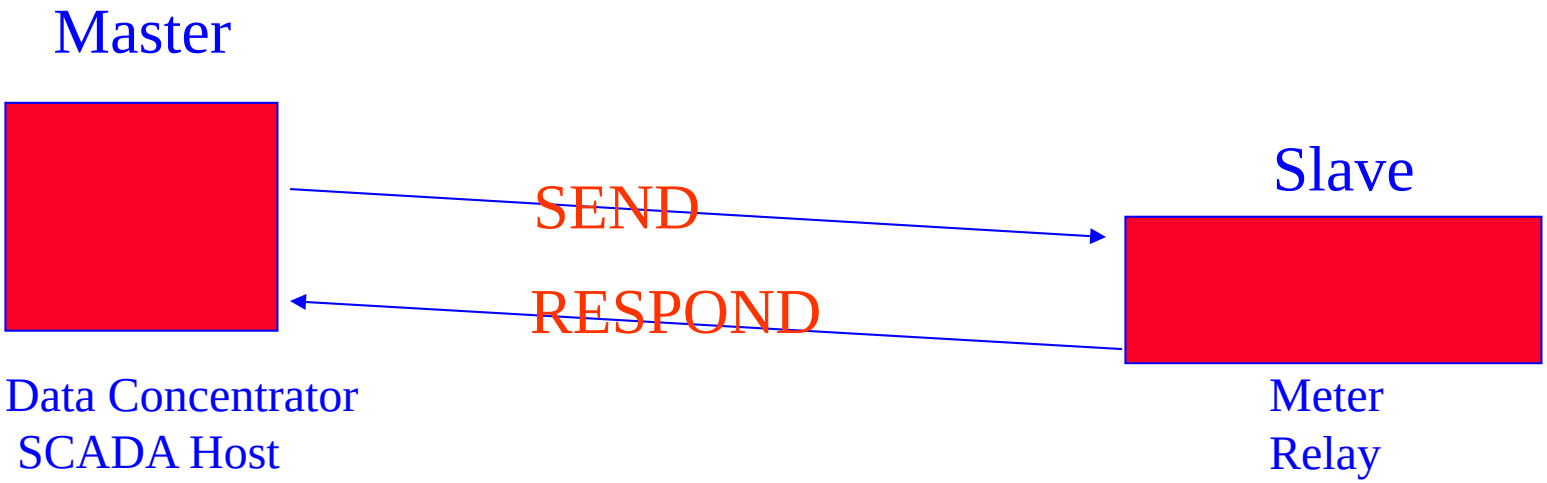
# Object Types per Object Category

| DATA TYPE | OBJECT | VARIANTS |
|---|---|---|
| Binary Input | 1,2, | \|1,2 \| \|1,2,3\| |
| Binary Output | 10,12 | \|1,2 \| \|1,2,3\| |
| Counter | 20,21,22,23 | \|1 - 8\| \|1 – 12 \| \|1 – 8 \| \| 1 – 8 \| |
| Analog Input | 30,31,32,33 | \|1 – 4\| \|1 – 6 \| \|1 – 4\| \|1 – 4\| |
| Analog Output | 40,41 | \| 1, 2 \| \| 1, 2\| |
| Time (relative or absolute) | 50,51,52 | \| 1,2 \| \| 1,2 \| \| 1,2\| |

# Object Types Per Object Category

| DATA TYPE | OBJECT | VARIANT |
|---|---|---|
| Class | 60 | \| 1 – 4\| |
| Files | 70 | \| 1 \| |
| Devices | 80,81,82,83 | \|1 \| \| 1\| \| 1 \| \| 1, 2 \| |
| Applications | 90 | \| 1 \| |
| Alternate Numeric | 100, 101 | \| 1,2,3 \| \|1,2,3\| |
| Future Use | [ 110 – 254] | - |

# Level 1 - DNP 3.0




- Master Requests - Slave Responds
- Slave MUST Accept Requests for:
  - Data Object Reads
  - Binary/Analog Output Object Reads *
  - Control Operations for Binary/Analog Outputs
  - Cold and Internal Indication Restarts
  - Delay Measurements
  - Writes to Date and Time

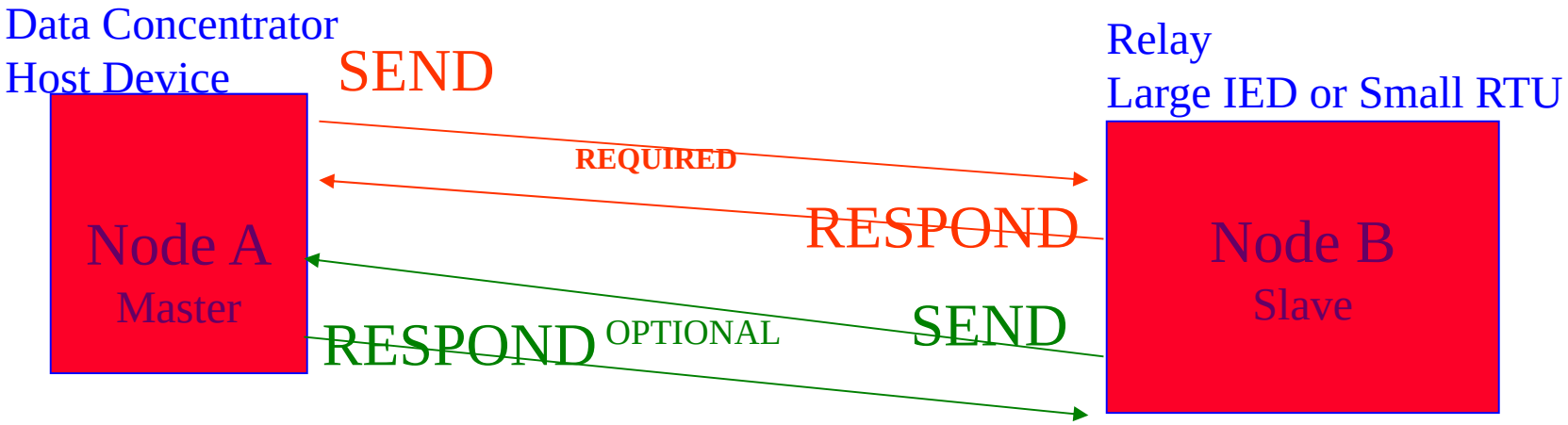
# Level 1 - DNP 3.0

- Master Must Accept (with multiple object variations)
  - Binary/Analog Input and Events
  - Counter and Counter Events
  - Binary/Analog Output Status
- Master Device Must be able to break the message into component pieces (parse).

# Level 1 - DNP 3.0

- OPTIONAL FEATURE Implementation
  - Slave OPTIONALLY MAY send unsolicited responses.
  - Slave OPTIONALLY MAY NOT generate parsed data objects if master requests such information.
  - Slave OPTIONALLY MAY respond without time object attachment.
  - Slave OPTIONALLY MAY send unsolicited responses **AND** the capability MUST be configurable.

# Level 2 - DNP 3.0




- Node A Requests - Node B Responds (Standard)
  Node B Requests - Node B Responds (Optional)
- Slave MUST Accepts Requests for:
  - FREEZE on Binary Counter Options
  - Parse of Read Requests of various objects and OPTIONALLY MAY report Frozen Counter objects.
- Master and Slave MUST incorporate Level 1 DNP features.

# Introduction to DNP

- Event Based
  - Binary change of state
    - multiple change detection
    - SOE
  - Analog % change
  - Event classes
  - Event buffer

# Introduction to DNP

- Object based
  - Data specification
  - Object Types
    - Value
    - Change
    - Frozen
  - Additional attributes

# DNP Event Data

- Each Event Index Corresponds to Present Value Index
- New Value
- Time stamp
- Class 1, 2, 3

# DNP Data Retrieval Types

- Polled
  - Present Values: Class 0
  - Event Data: Class 1, 2, 3
- Unsolicited
  - Event Data

# Introduction to DNP

- Modbus Data Request

Master requests specific memory area from slave

Slave responds with all data in region

# Introduction to DNP

- DNP Polled Static

Master requests all data of a type of Class 0

Slave responds with all data of type or all Classes

# Polled Static

- Polling Mechanism: Static Data Report
- Considerations
  - Only Class 0 Data Reported
  - No Unsolicited Data Reports

# Introduction to DNP

- DNP Polled Report-by-Exception

Master performs periodic Class 0 poll

Slave responds to Class 0 poll with all data

Master polls for events

Slave reports event data

# Polled Report-by-Exception

- Polling Mechanism: Static Data Report
  - Frequent Event Polls
  - Infrequent Integrity Polls
- Considerations
  - Class 1, 2, and 3 data reported from event polls
  - No Unsolicited Data Reports

# Introduction to DNP

- DNP Polled Report-by-Exception

Slave reports unsolicited event data

Master performs occasional Class 0 poll

Slave responds to Class 0 poll with all data

# Unsolicited Report-by-Exception

- Polling Mechanism
  - IEDs send Unsolicited Data
  - Occasional Class 0 Polls
- Considerations
  - Unsolicited Data used for Event Data
  - Static Polls for Data Synchronization

# Introduction to DNP

- DNP Polled Report-by-Exception

Slave reports unsolicited event data

Master does not poll

# Introduction to DNP

- Optimized Communication
  - Event-driven polling
    - class 0
    - class 1, 2, 3
  - Minimum message size

# Introduction to DNP

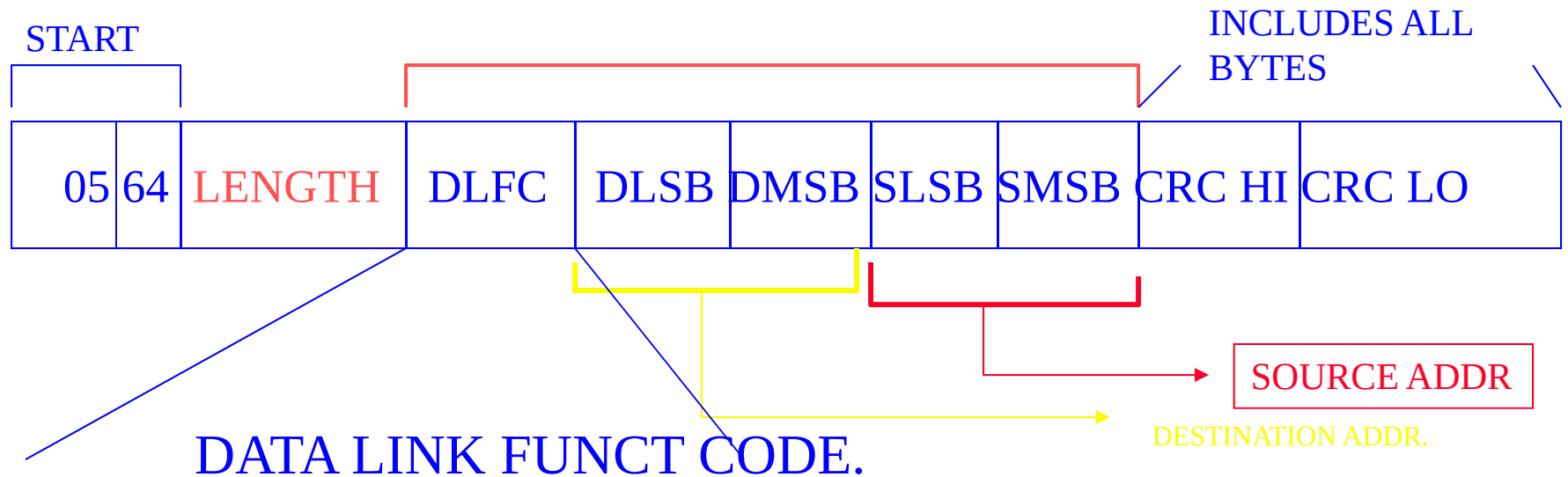
- High Data Integrity
  - 16-Bit CRC every 16 bytes
  - Data link confirmations
  - Application confirmations

# Introduction to DNP

- Structured Evolution
  - Subset definitions
  - Object definitions
  - Standard documentation
  - Conformance testing
  - User's group
  - Technical committee

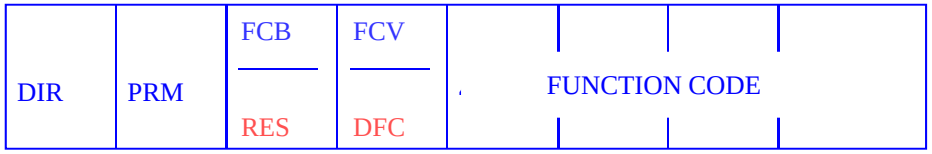
# DATA LINK MESSAGE

- Data Link Header Length May be from 5 to 255 Bytes Long.




Data Link Header Length May be from 5 to 255 Bytes Long. 5 4 3 2 1 0

# Data Link Header Control Field




FROM PRIMARY TO SECONDARY HOST TO RELAY

FROM SECONDARY TO PRIMARY RELAY TO HOST

7 6 5 4 3 2 1 0

DIR = DIRECTION - 1 = Master To Relay 0 = Relay to Master
Gives Direction Of the Frame with respect to the master.

PRM= Data Flow Control 1 = Frame from Initiating Station 0 = Frame from Responding Station
Gives Direction of the Frame in relation to the sending station.

FCB = Frame Count Bit Toggles with each SEND/CONFIRM COMBINATION (on same Master/ IED transaction.
Used to prevent duplication of frames and loss of frame transmission. (Sent From Host)

FCV = Frame Count Valid 1 = Frame Count Bit Valid 0 = Ignore Frame Count Bit.
Enables Function of Frame Count Bit. (Sent From Host)

RES = Reserved Bit - No Function Defined

DFC = Data Flow Control 1 = Send Causes Data Link Buffer Overflow in Relay. 0 = Primary Can Send Data.
Prevents Overflow of Data buffers in Relay ( Returned on Host Request)

**FUNCTION CODE -** Identifies the Type of Message.

**PRM = 1**

| FC | FRAME TYPE | SERVICE FUNCTION | FCV BIT |
|---|---|---|---|
| 0 | Send Confirm Expected | Reset of Remote Link | 0 |
| 1 | Send Confirm Expected | Reset of User Process | 0 |
| 2 | Send Confirm Expected | Test Function For Link | 1 |
| 3 | Send Confirm Expected | User Data | 1 |
| 4 | Send No Reply Expected | Unconfirmed User Data | 0 |
| 9 | Request Respond Expected | Request Link Status | 0 |

**PRM = 0**

| FC | FRAME TYPE | SERVICE FUNCTION |
|---|---|---|
| 0 | CONFIRM | ACK |
| 1 | CONFIRM | NACK |
| 11 | RESPOND | Link Status (DFC) |

SECONDARY FRAMES

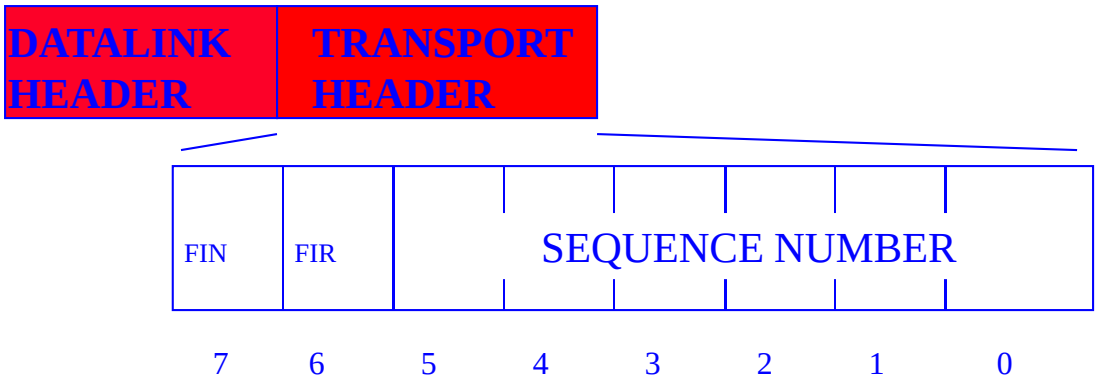
# What Does the Data Link Layer Do?

- DNP can allow a host and and IED (Unsolicited Request) to act as a master.
- The Data Link Layer:
  - Synchronizes data exchanges.
  - Controls Message Retries.
  - Connects and Disconnects Dial Up Sessions.
  - Controls the Physical Layer.
  - Provides message services (priority, error notification).
  - Establishes and disconnects a DIAL UP Connection.
  - Sets the Frame Construction in a DNP 3.0 session.
  - Performs Collision Avoidance of messages ( in an unsolicited response node).

# Transport Layer

- The TRANSPORT LAYER indicates the length of a communication session.
- Why is this needed?
  - Long Messages exceeding 255 bytes are segmented in multiple messages.
  - The Length of DLC data is 5 bytes
  - The Length of the TL is 1 byte
  - The remaining data length is 255 - 5 -1 = 249 bytes.
  - In case any data frames are corrupted, a retransmission of the corrupted frame may occur.
  - Allows assembly of large messages by a host device.

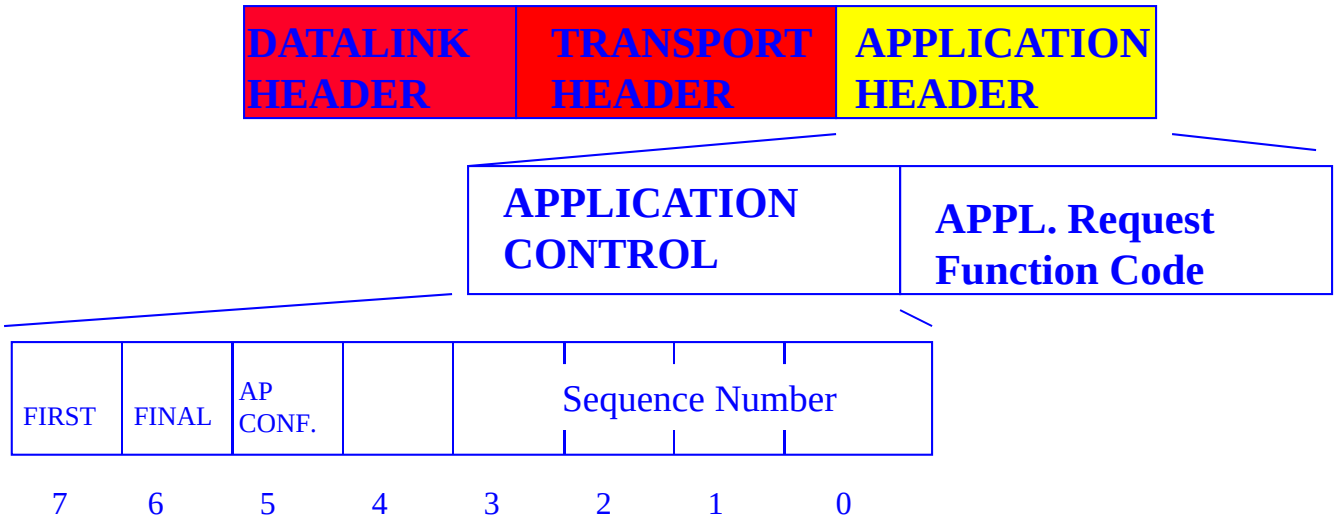
# TRANSPORT LAYER




FIN = Final Indication 1 = FINal Frame in sequence 0 = More Frames Follow
FIR = FIRst Frame 1 = First Frame In a Sequence 0 = Not The First Frame
0 <= Sequence Number <= 63 (Number rolls over if more frames than 63)

- Byte indication of number of frame in transmission sequence and first/last frame.

# Application Header (Request)




FIN = Final Indication 1 = FINal Frame in sequence 0 = More Frames Follow
FIR = FIRst Frame 1 = First Frame In a Sequence 0 = Not The First Frame
AP CONF. = Application Confirm 1 = Sending Node Expects Confirm 0 = No Confirm Expected.
UNSOL = Unsolicited 1 = IED Responds 0 = Not Unsolicited
SEQUENCE NUMBER 0 <= X<= 15 - Master Station Requests (Rollover at 15)
16 <= X <= 31 - Unsolicited Request Sequence Numbers

- Contains a Header and A Function Code

# Application Header (Request)

- There are 8 Types of Function Codes:
  - 1. Transfer Function Code
    - 0 (00x) - Confirm.
    - 1 (01x) - Read.
    - 2 (02x) Write.
  - 2. Control Function Code
    - 3 (03x) -Select.
    - 4 (04x) -Operate.
    - 5 (05x) -Direct Operate.
    - 6 (06x) -Direct Operate - No Acknowledge.

# Application Header (Request)

- 3. Freeze Function Code
  - 7 (07x) Immediate Freeze.
  - 8 (08x) Immediate Freeze - No Acknowledgement.
- 4 Transfer Function Code
  - 9 (09x) Freeze and Clear
  - 10 (0Ax) Freeze and Clear - No Acknowledgement.
  - 11 (0Bx)  Freeze with Time
  - 12 (0Cx)  Freeze with Time - No Acknowledgement

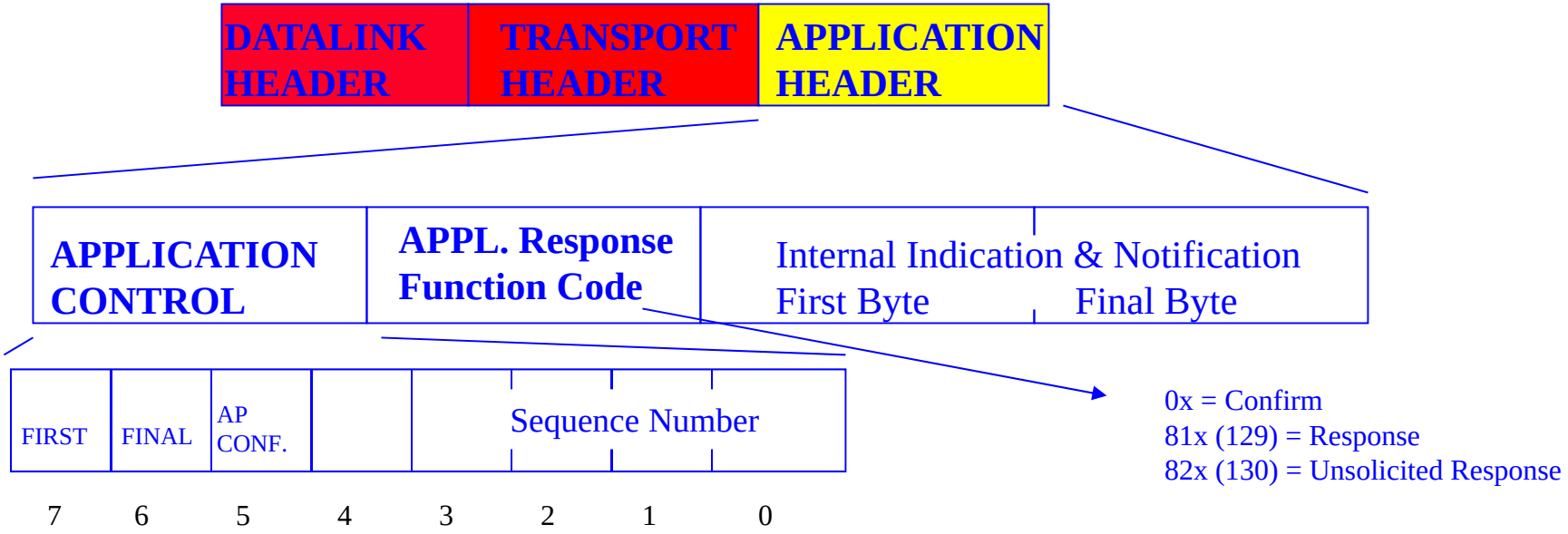
# Application Header (Request)

- 5. Application Control Function Codes
  - 13 (0Dx) Cold Restart
  - 14 (0Ex) Warm Restart
  - 15 (0Fx) Initialize Data to Defaults
  - 16 (10x) Start Application
  - 17 (11x) Stop Application
- 6. Save Function Codes
  - 18 (12x) = Save Configuration
  - 19 (22x) = Enable Unsolicited Messages

# Application Header (Request)

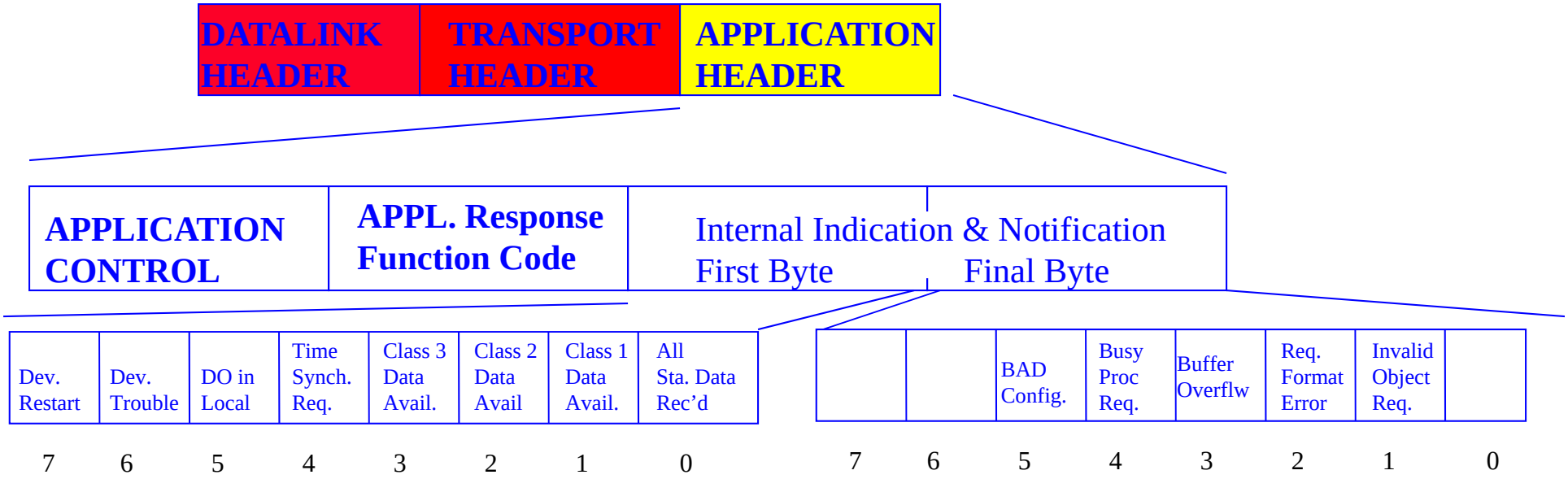
- 7. Transfer Function Codes
  - 21 (15x) -Disable Unsolicited Messages
  - 22 (16x) - Assign Class to Data Object
- 8. Time Synchronization
  - 23 (17x) - Delay Measurement

# Application Header (Response)

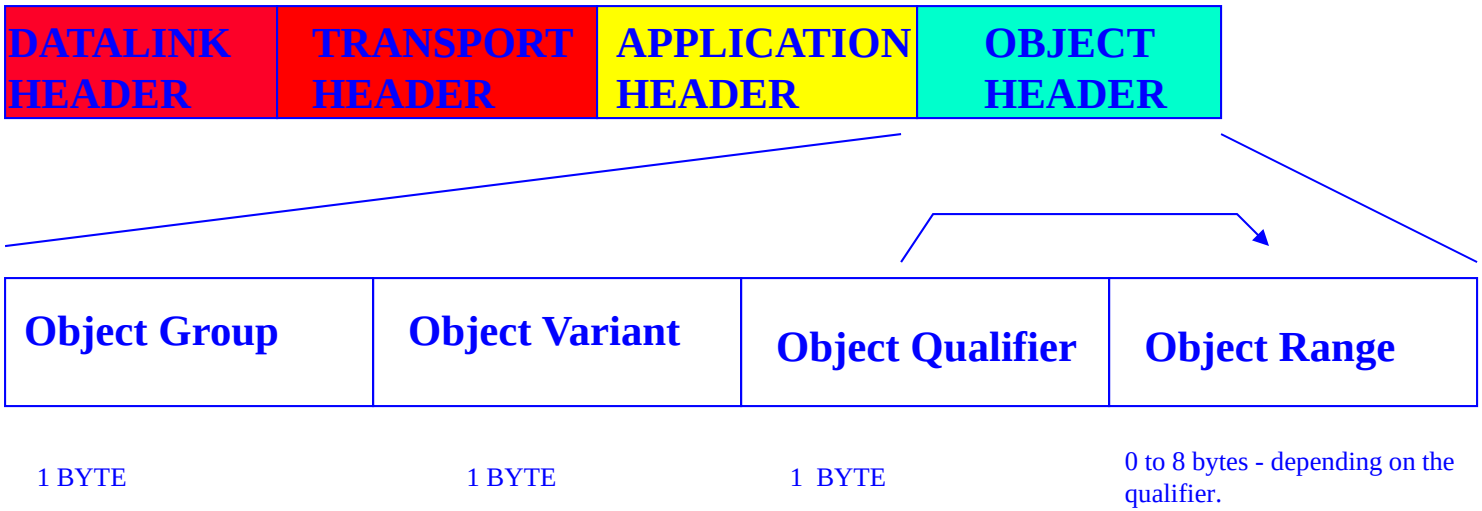



FIN = Final Indication 1 = FINal Frame in sequence 0 = More Frames Follow
FIR = FIRst Frame 1 = First Frame In a Sequence 0 = Not The First Frame
AP CONF. = Application Confirm 1 = Sending Node Expects Confirm 0 = No Confirm Expected.
UNSOL = Unsolicited 1 = Unsolicited Response from the IED
SEQUENCE NUMBER 0 <= X<= 15 - Master Station Requests (Rollover at 15)
16 <= X <= 31 - Unsolicited Request Sequence Numbers

# IIN Field (Response)



# Application Layer - Object Header




- Object Header is of variable length and differs when requesting information in a variety of formats.

# Object Header

- An object is a type of information requested.
- There are 12 Object Types.
- Variants:
  - Objects report data in differing formats.
  - A variant describes the format of the Object.
  - Different Objects have differing defined variants.
  - For Example
    - Object 1 -Binary Input (Static Data).
      - Variant 1 - Static Data.
      - Variant 2 - Static Data With Point status.
  - Variant 0 is the manufacturers default variant.
- Qualifier:
  - Data may be read or written using a variety of addressing schemes or requesting a variety of points.
  - A qualifier “defines” the return or write format of the data.
  - A Qualifier is a decoded single byte.

# Class Data Reporting

- Data May be Obtained in a variety of methods:
  - Ask for each point by Object and Variant.
  - Have the IED report the data by Unsolicited Response.
  - Have the Host report the data in Classes.
    - Class 0
    - Class 1
    - Class 2
    - Class 3

# Typical IED (Slave) DNP Settings

- Communication Settings
  - Communication Ports and Baud Rates for serial ports
  - IP addresses and TCP / UDP port numbers
- DNP Address
  - Master / Client
  - Slave / Server (IED)
- Class for Event Data (0, 1, 2, 3)

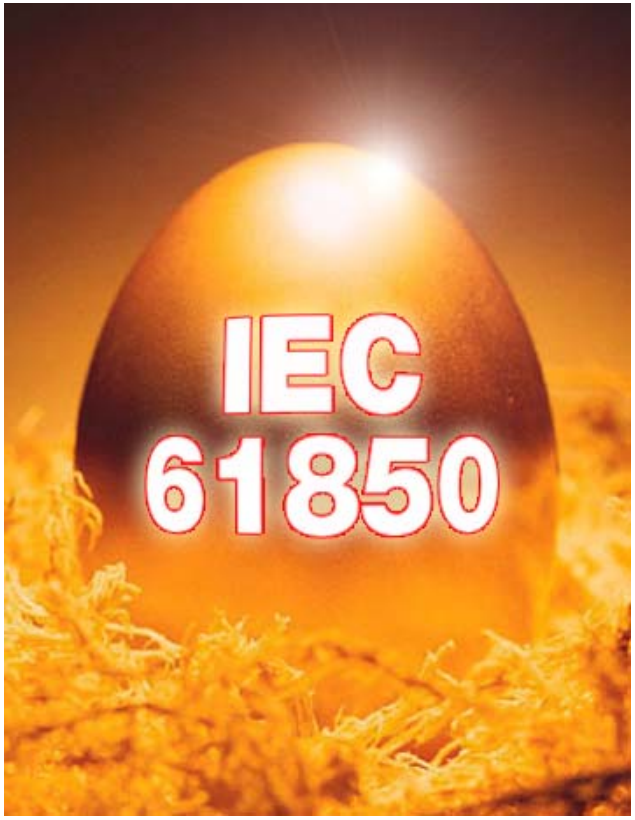
# Typical IED (Slave) DNP Settings

- Analog Variations,16 / 32 bit
- Select / Operate Time Out
- Number of Data Link Retries
  - On / Off
  - Number of Retries if used and Timeout
- Min / Max Delay From DCD (0 – 1000 ms)

# Typical IED (Slave) DNP Settings

- Hardware handshaking for Serial Connections
  - RTS / CTS and DCD functions
  - May not be applicable to all IEDs
- Settings for Analog Events
- Unsolicited Reporting Enable / Disabled

# IEC61850

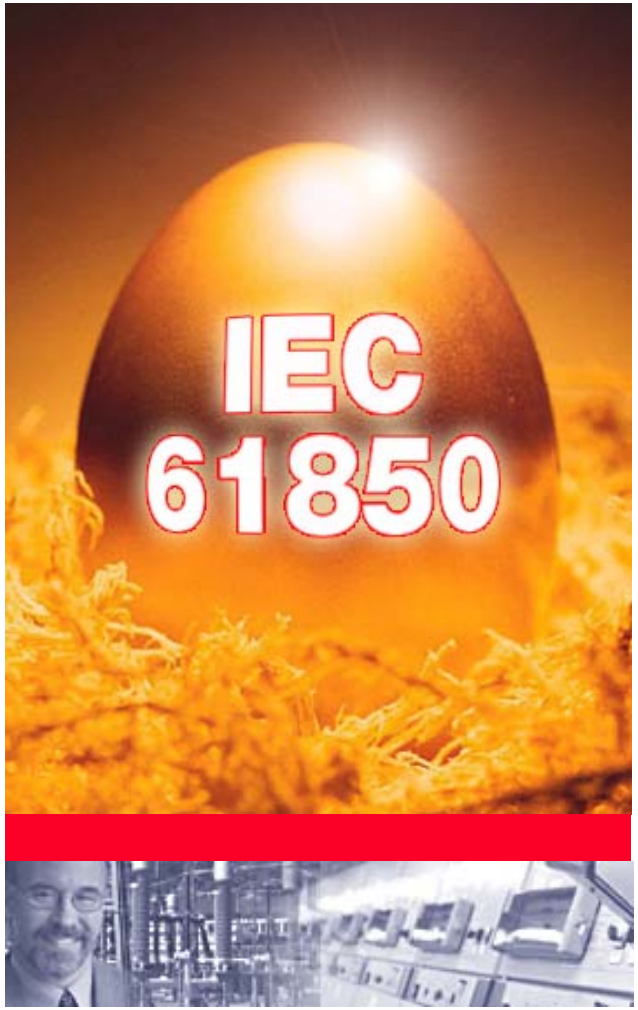



- A collection of specifications, protocols, mappings and models to address :
- Standarized way of representing data and documenting it (Data Model)
- Basic equipment specification for Substation Automation system
- Basic Substation Automation system performance requirements
- Protocol to report status information (Client-Server)
- Protocol to exchange information between IEDs at high speeds
- Protocol to distribute analog information for metering, protection and control functions (Process Bus)

# IEC61850 HISTORY

- UCA Project Origin (North American Standardization Initiative):
  - Utility Communications Architecture (UCA) - enterprise-wide unified scheme to share all operating and management information
  - 1994 - EPRI member utilities called for common standard for IEDs in substations
  - EPRI RP 3599 defined requirements; looked at UCA compatible technologies for substations
  - 1996 - UCA demonstration initiative by AEP and other major utilities.
    - Pushed to identify Ethernet protocol to be used for all data sharing, plus high-speed control
    - Solicited IED vendor and user participation
    - Specified replacing control wiring with LAN
- IEC 61850 Origin (first an EU Standardization Initiative):
  - 1980s - Large European manufacturers were selling expensive LAN-based substation control systems (SCS)
  - Each design unique, and user committed to one vendor's equipment
  - Later - IEC developed Standard 870-5 (now 60870-5)
  - 1995 - IEC TC 57 began 61850 Standard to define next generation of LAN-based substation control

# IEC 61850 based SA systems



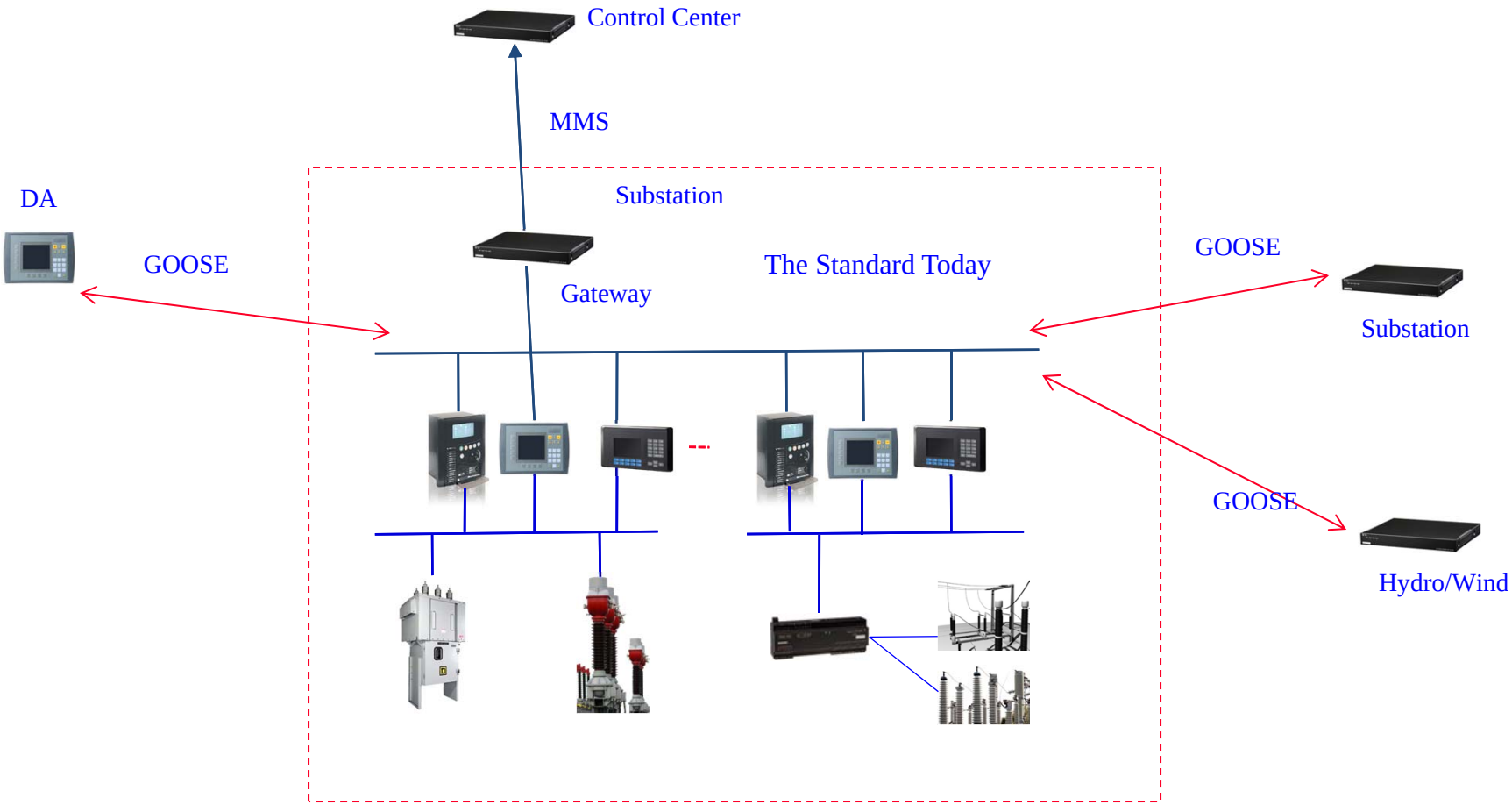

**Basics:**

- Fast Ethernet (100 MBps to 1 GBps)
- MMS
- Station Bus 61850 8-1
- Process Bus 61850 9-2
- Data Model
- Substation Configuration Language

**Much more than a protocol:**

- Modularization and structuring of data
- On-line meaningful information
- Free allocation of functions in IEDs
- Complete description of configuration
- Structured engineering & services
- Testing, validation, and certification

# IEC 61850 Uses

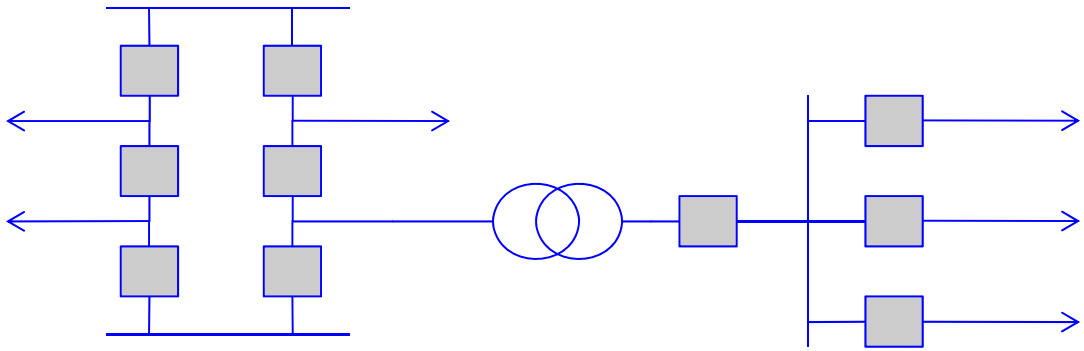


# IEC 61850 - What the Standard Provides

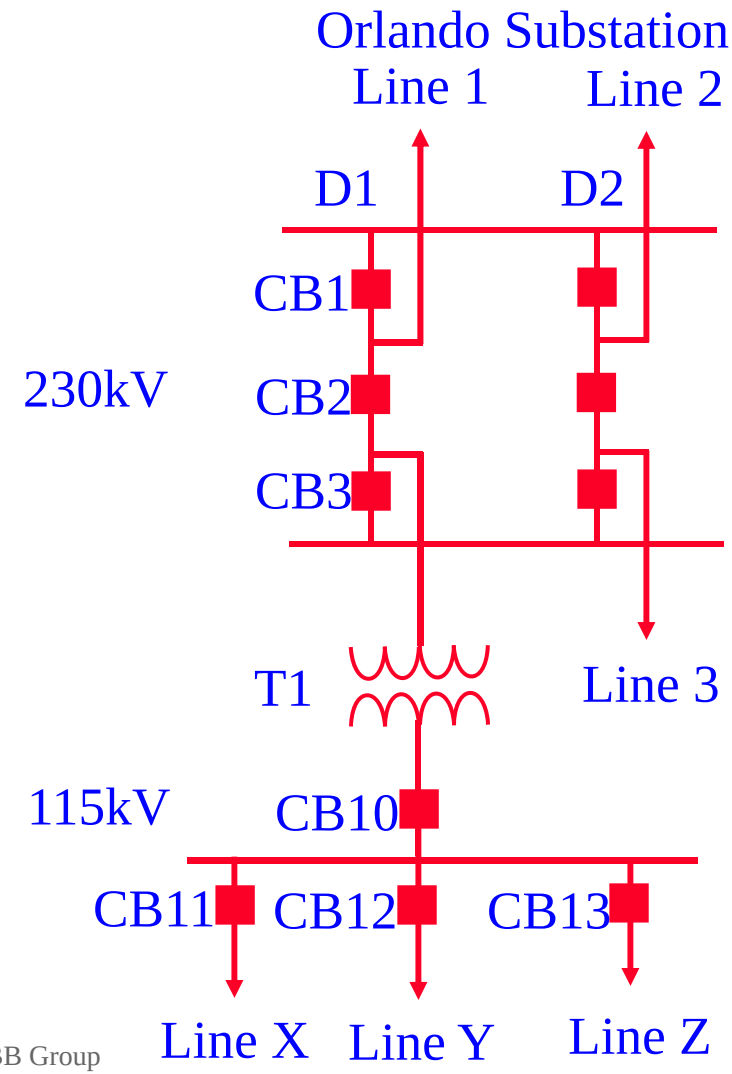
- Interoperability
  - Exchange information between IED's (Intelligent Electronic Device) from several manufacturers
  - IEDs use this information for their own function
- Long Term Viability
  - Future proof
  - Follow progress in mainstream communication technology
  - Follow evolving system requirements needed by customers

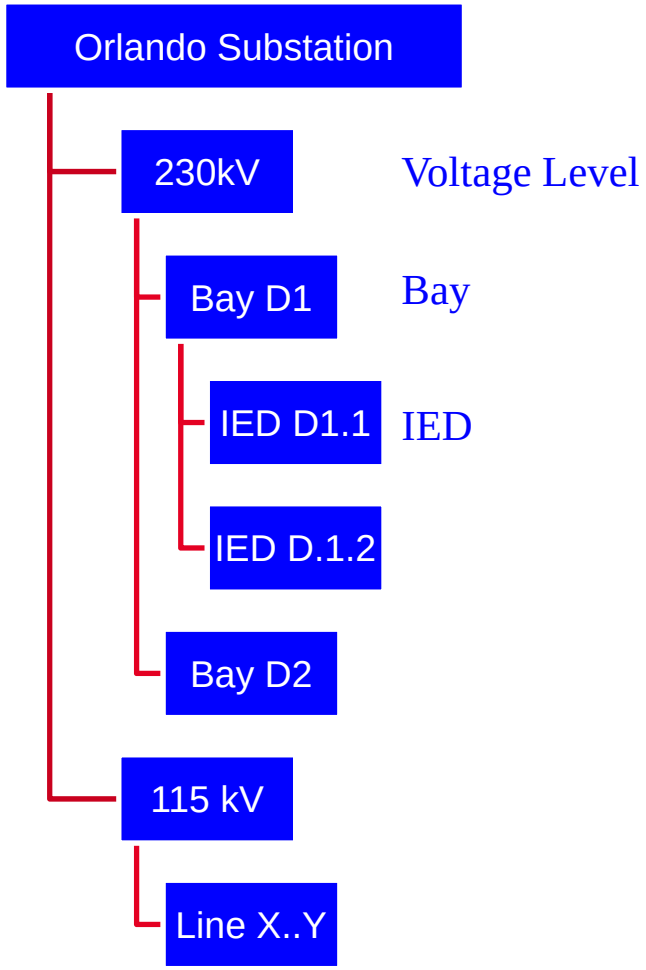
# INTEGRATION - Where does IEC 61850 help?

- It is about AVOIDING ...
  - Media and protocol converters when using multiple protocols
  - Understanding each device's unique memory/point/register map
  - Programming the data concentrator to accept that information
    - Data types
    - Reporting structure
- Connecting the information to the application
  - Point 1 from Device 1 = 52A



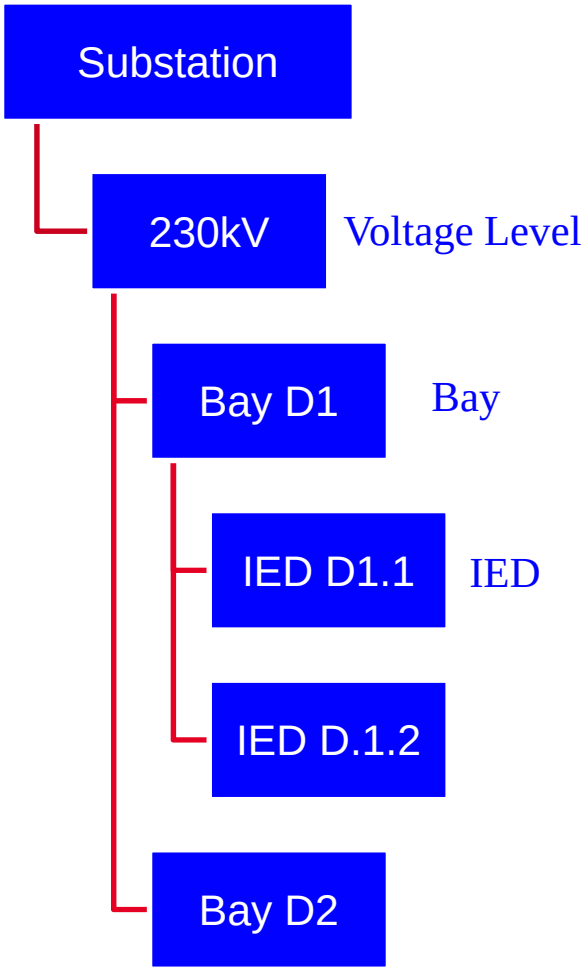
# Substation Modeling - The Substation Structure



# Substation Modeling
# The Substation Structure

## Substation Structure




## IED Data Model