M.Tech. Credit Seminar Report, Electronics Systems Group, EE Dept, IIT Bombay, submitted November '03

Comparison of protocols used in remote monitoring: DNP 3.0, IEC 870-5-101 & Modbus

Jay Makhija (03307905)
Supervisor: Prof. L.R.Subramanyan

Abstract

A brief introduction to IEC 870-5-101, DNP 3.0 and Modbus is presented together with the explanation of functionalities provided by these protocols. Comparison between these protocols, considering various aspects like architecture, functions performed by individual layers, device addressing and configuration parameter's etc is done. Here objective is to compare protocols not to conclude which one is better. Selection of protocol depends mainly on application, specific requirements and functions to be carried out, which protocol to be used for what purpose is discussed.

1. Introduction

In a SCADA system, the RTU accepts commands to operate control points, set analog output levels, and provide responses, it sends status, analog, and accumulator data to the SCADA master station. The data representations sent are not identified in any fashion other than by absolute addressing. The addressing is designed to correlate with a database contained in the SCADA master station, and the RTU has no knowledge of which unique parameters it is monitoring in the real world. It simply monitors certain points and stores the information in a local addressing scheme. Figure 1 illustrates the data and control flow between a master station and one or more RTUs.

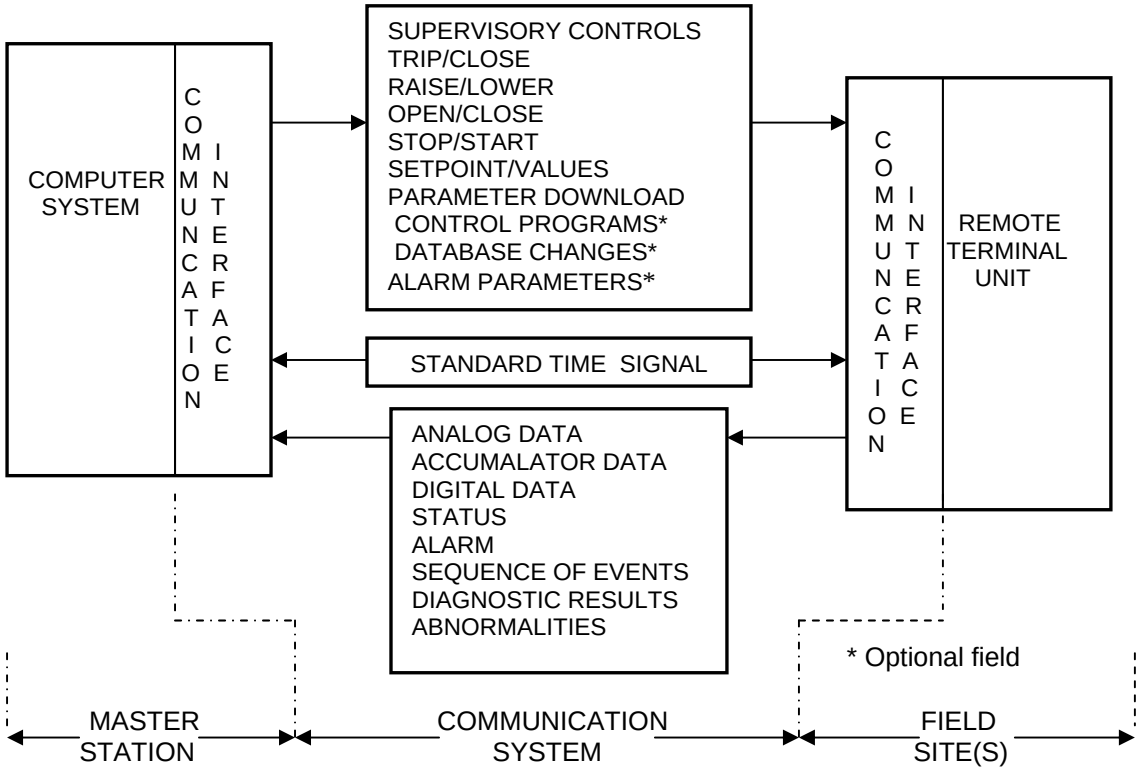


Figure 1— Master/RTU functional data/control flow [ref. 2]

The SCADA master station is the part of the system that should “know” that the first status point of RTU number x (say 27) is the status of a certain circuit breaker of a given substation. This represents the predominant SCADA systems and protocols in use in the utility industry today. Each protocol consists of two message sets or pairs. One set forms the master protocol, containing the valid statements for master station initiation or response, and the other set is the RTU protocol, containing the valid statements an RTU can initiate and respond to. In most but not all cases, these pairs can be considered a poll or request for information or action, and a confirming response.

The SCADA protocol between master and RTU forms a viable model for IED-to-RTU communications; therefore, the DNP 3.0 and IEC 870-5-T101 (1995) protocols in the practice are SCADA-based protocols.

1.1 Need for Standards

The communication protocol allows two devices to communicate with each other. Each device involved in the communication must essentially support not only the same protocol but also the same version of the protocol. Any differences involved in the implementation of protocol at the either of ends will result in the communication errors.

There is usually very little problem for devices to communicate with each other when all devices are from same supplier and support same protocol. Because of using the unique protocol, used by the vendor, the utility is restricted to one supplier for support and purchase of future devices. This presents a serious problem.

With the arrival of “open systems concept”, it is desired that devices from one vendor be able to communicate with those of other vendors i.e. devices should “interoperate”. To achieve interoperability one has to use industry standard protocols. Having industry standard, where the vendors design their devices such that all functionality and capabilities are possible with the protocol, they provide the utilities, with the flexibility of buying the best devices for each application.

Using standard communication protocol is a very important decision that leads to cost reduction and maximized flexibility within the utility sector. Broadly benefits for the utilities are:

- Availability of ”open system connectivity”
- Vendor independence
- Reliable products at optimized costs
- Easily available knowledge and specification

Benefits drawn for vendors by standardization are:

- Lower costs of installation and maintenance
- A large market and thus opportunity to compete on price performance instead of technical details only
- Cost effective project implementation

The price paid for the gain of above-mentioned advantages is:

- More overheads are there, thus requiring higher speed for the same efficiency or information throughput
- There is possibility that utility will not be able to realize the full functionality of the device while using an industry standard protocol.

1.2 Enhanced Performance Architecture (EPA) [ref.1]

Both DNP and IEC 870-5 protocol are based on a three-layer Enhanced Performance Architecture (EPA) reference model for efficient implementation within RTUs, meters, relays, and other IEDs. Additionally, EPA defines basic application functionality for a user layer, which is situated between the OSI Application Layer and the application program. This user layer adds interoperability for such functions as clock synchronization and file transfers. Figure 2 shows the EPA layer organization.

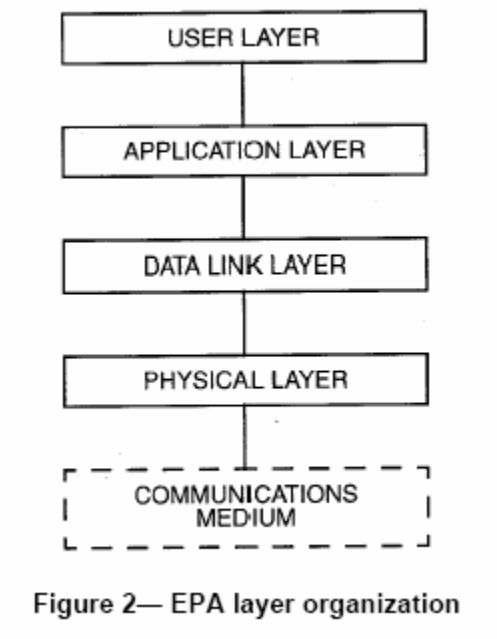


Figure 2— EPA layer organization

The DNP 3.0 layer stack adds a pseudo-transport layer, for the efficient transmission of large sized application data. Pseudo-transport layer functioning is explained in section 2.3.

1.3 Balanced and Unbalanced Transmission [ref. 9]

In balanced transmission at the link layer, all devices are equal. Collision is avoided by one of the following:

- Full duplex point to point connection (RS232)
- Designated master polls rest of slaves on network (two wire RS485 and disable data link confirms in slaves)
- Physical layer (CSMA/CD)

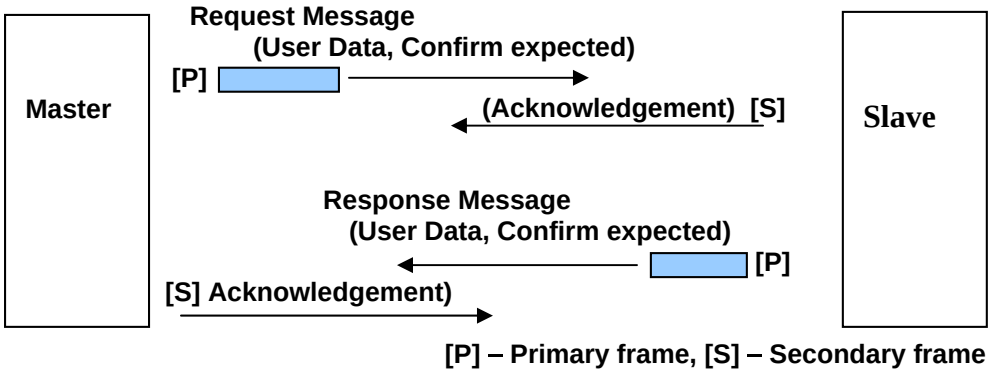


Figure 3 – Balance Mode Transmission

In unbalanced transmission only Master device can transmit primary frames. Collision avoidance is not necessary since slave device cannot initiate exchange, or retry failed messages if the slave device responds with

NACK: requested data not available

the master will try again until it gets data, or a response time-out occurs

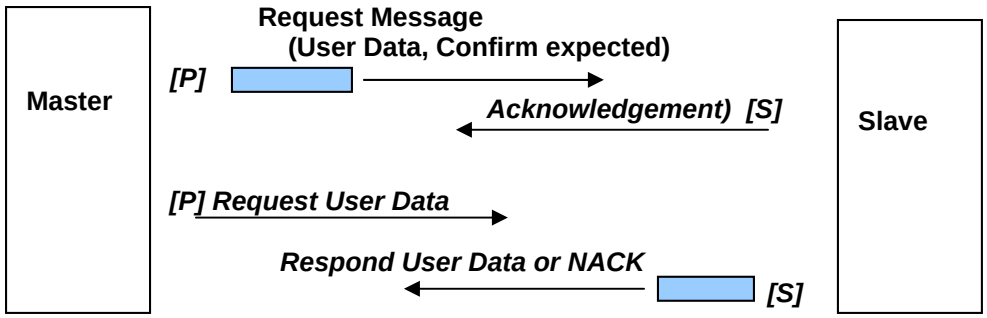


Figure 4 – Unbalance Mode Transmission

2. DNP 3.0 [ref. 1 & 3]

DNP was originally created by Westronic, Inc. (now GE Harris) in 1990. In 1993, the "DNP 3.0 Basic 4" protocol specification document set was released into the public domain, turned over to Users Group in 1993

Core Specification Documents

DNP 3.0 - Basic 4 Document Set DNP 3.0 Data Link Layer
DNP 3.0 - Transport Functions
DNP 3.0 - Application Layer Specification
DNP 3.0 - Data Object Library

The DNP 3.0 is specifically developed for interdevice communication involving SCADA RTUs, and provides for both IED-to-RTU and master-to-IED/RTU. DNP 3.0 was developed with the following goals: **[ref. 1]**

a) *High data integrity:* The DNP 3.0 Data Link Layer uses a variation of the IEC 870-5-1 (1990) frame format FT3. Both data link layer frames and application layer messages may be transmitted using confirmed service.

b) *Flexible structure:* The DNP 3.0 Application Layer is object-based, with a structure that allows a range of implementations while retaining interoperability.

c) *Multiple applications:* DNP 3.0 can be used in several modes, including:
1) Polled only.
2) Polled report-by-exception.
3) Unsolicited report-by-exception (quiescent mode)
4) Mixture of the modes 1) – 3).
It can also be used with several physical layers, and as a layered protocol is suitable for operation over local and some wide area networks.

d) *Minimized overhead:* DNP 3.0 was designed for existing wire-pair data links, with operating bit rates as low as 1200 b/s, and attempts to use a minimum of overhead while retaining flexibility. Selection of a data reporting method, such as report-by-exception, further reduces overhead.

e) *Open standard:* DNP 3.0 is a non-proprietary, evolving standard controlled by a users' group whose members include RTU, IED and master station vendors, and representatives of the electric utility and system consulting community.

2.1 Physical Layer

The physical layer is primarily concerned with the physical media over which the protocol is being communicated. For example, it handles state of the media (clear or busy), and synchronization across the media (starting and stopping). Most commonly, DNP is specified over a simple serial physical layer such as RS-232 or RS-485 using physical media such as copper, fiber, radio or satellite. More recent applications have implemented DNP3 over an Ethernet connection.

2.2 Data link layer

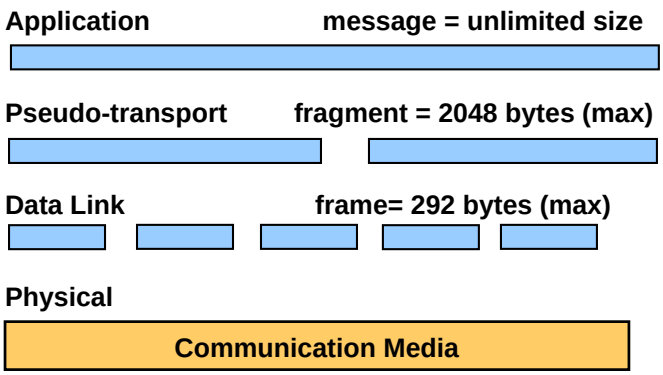
The DNP 3.0 data link layer specification describes the frame format, services, responsibilities, and transmission procedures for the data link layer. It describes the required services to be provided by a DNP 3.0 physical layer. *DNP 3.0 is essentially media-independent when the physical layer interface meets these requirements*. For instance, if unsolicited messaging is used, the physical layer shall provide an indication of whether the link is busy, which is necessary for collision avoidance. The DNP 3.0 data link layer specification also relates the DNP 3.0 data link layer to IEC 870-5-1 (1990) and IEC 870-5-2 (1992) standards. The primary difference is that DNP 3.0 uses the FT3 frame format for asynchronous, rather than synchronous, transmission. DNP 3.0 also adapts the IEC 870-5 addressing to include both a source and destination address in the frame. This addition enables the use of multiple master stations and peer-to-peer communications using DNP 3.0.

2.3 Transport functions

The pseudo-transport layer segments application layer messages into multiple data link frames. For each frame, it inserts a single byte function code that indicates if the data link frame is the first frame of the message, the last frame of a message, or both (for single frame messages). The function code also includes a rolling frame sequence number which increments with each frame and allows the receiving transport layer to detect dropped frames.

The transport header is removed by the device at each end of a physical layer, like the data link overhead, so it is not a true end-to-end transport layer. However, it is not actually part of the data link overhead but is counted as the first octet of cyclic-redundancy-checked user data carried by the data link layer. All confirmation and reliability is provided by the data link layer, not by the transport function. This function results in reduced layers and overhead, and retains a high level of data integrity, yet provides a richer set of application layer functions.

2.4 DNP Message Buildup [ref. 9]



2.5 Application Layer

The DNP 3.0 application layer specification describes the message format, services, and procedures for the application layer. The application layer responds to complete messages received (and passed up from the transport layer), and builds messages based on the need for or the availability of user data. Once messages are built, they are passed down to the pseudo-transport layer where they are segmented and passed to the data link layer and eventually communicated over the physical layer. The total length of received messages is indicated by pseudo-transport layer as it appends data link layer frames, each with their own indicated length.

When the data to be transmitted is too large for a single application layer message, multiple application layer messages may be built and transmitted sequentially. However, each message is an independent application layer message; their only association with each other is an indication in all but the last message that more messages follow. Because of this possible

fragmentation of application data, each application layer message is referred to as a **fragment** and a message may either be a **single-fragment message** or a **multi-fragment message**.

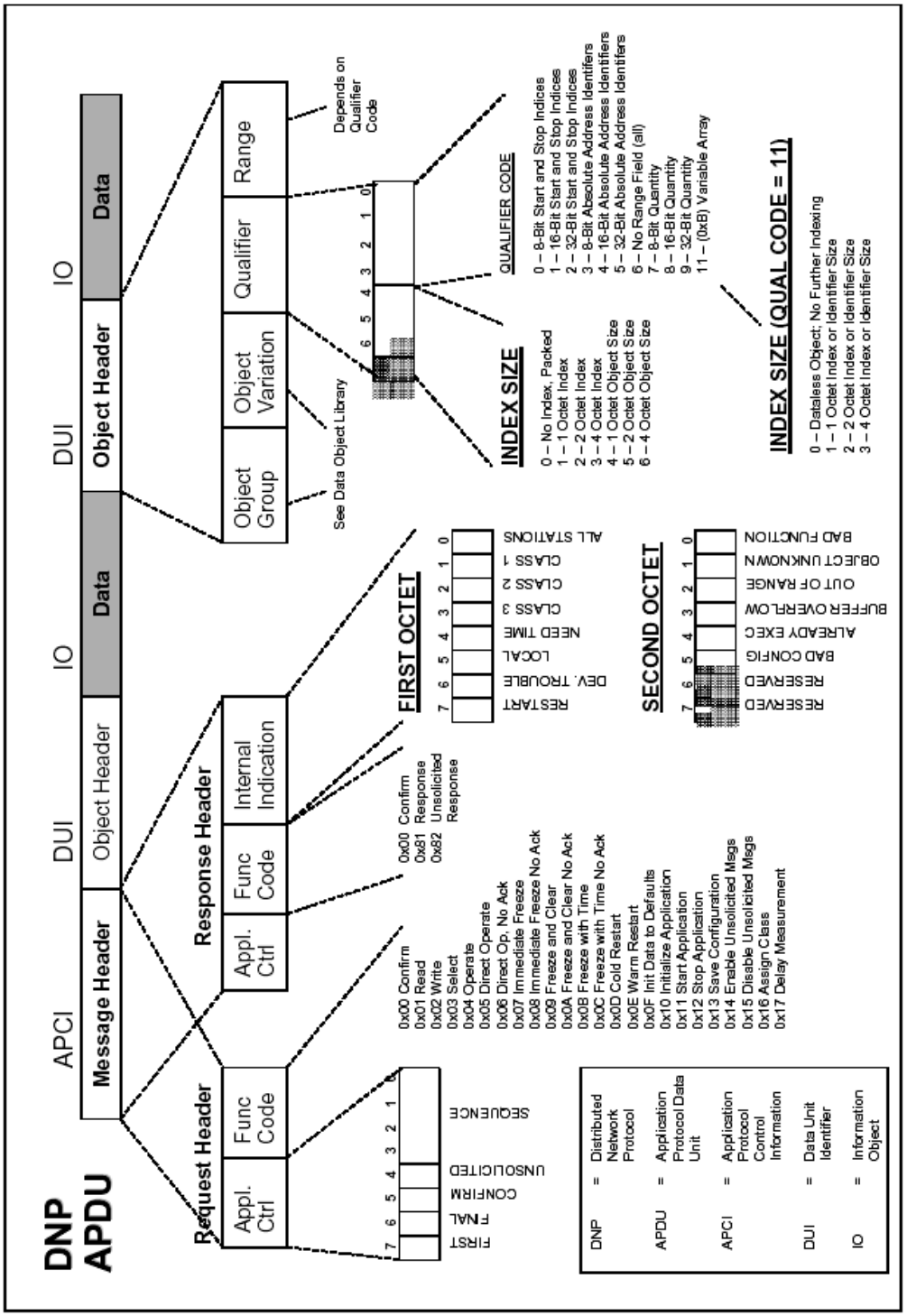


Figure 5 - DNP APDU frame details [ref. 10]

Application layer fragments from Master DNP3 stations are typically **requests** for operations on data objects, and application layer fragments from Slave DNP3 stations are typically **responses** to those requests. A Slave DNP3 station may also transmit a message without a request (an **unsolicited response**).

As in the data link layer, application layer fragments may be sent with a request for a confirmation. An **application layer confirmation** indicates that a message has not only been received, but also been parsed without error. (On the other hand, a data link layer confirmation, or ACK, indicates only that the data link frame has been received and that it passes CRC error checks.)

Each application layer fragment begins with an **application layer header** followed by one or more object header/object data combinations. The application layer header contains an application control code and an application function code. The **application control code** contains an indication if the fragment is one of a multi-fragment message, contains an indication if an application layer confirmation is requested for the fragment, contains an indication if the fragment was unsolicited, and contains a rolling application layer sequence number. The application layer sequence number allows the receiving application layer to detect fragments that are out of sequence, or dropped fragments.

The application layer header function code indicates the purpose, or requested operation, of the message. While DNP3 allows multiple data types in a single message, it only allows a single requested operation on the data types within the message. Example function codes (as shown in fig. 4) include: Confirm (for application layer confirmations), read and write, select and operate (for select-before-operate, or SBO, controls), direct operate (for operation of controls without SBO), freeze and clear (for counters), restart (both cold and warm), enable and disable unsolicited messages, and assign class. The application layer header function code applies to all object headers, and therefore all data within the message fragment.

2.6 Data object library

The DNP 3.0 data object library document describes the format of data presented within an application layer message. A variety of qualifier codes and variations of data permit an implementation of DNP 3.0 to make optimal use of bandwidth. DNP objects are not general-purpose objects; they are defined specifically for RTU operation.

2.7 Subset definitions

The DNP 3.0 subset definitions document describes three basic levels of DNP 3.0 objects and services that can be used to determine interoperability between devices, or to specify a minimum required level of implementation in a request for proposals. The intended use of these subsets is as follows:

Level 1 (L1): A minimum implementation, intended for a simple IED.
Level 2 (L2): Intended for a more sophisticated IED or a small RTU.
Level 3 (3): Intended for a larger RTU or data concentrator.

3. IEC 870-5-101 [ref. 1, 4 & 6]

The IEC Technical Committee 57 (Working Group 03) have developed a protocol standard for telecontrol, teleprotection, and associated telecommunications for electric power systems. The result of this work is IEC 870-5. Five documents specify the base IEC 870-5. The documents are:

IEC 870-5-1 Transmission Frame Formats
IEC 870-5-2 Data Link Transmission Services
IEC 870-5-3 General Structure of Application Data
IEC 870-5-4 Definition and coding of Information Elements
IEC 870-5-5 Basic Application Functions

IEC 870-5-101 (T101) is a companion standard generated by the IEC TC57 for electric utility communication between master stations and RTUs. The IEC 870-5-101 is based of the five documents IEC 870-5-1-- 5. Like DNP 3.0, T101 provides structures that are also directly applicable to the interface between RTUs and IEDs. It contains all the elements of a protocol necessary to provide an unambiguous profile definition so that vendors may create products that interoperate fully.

3.1 Physical Layer

At the physical layer, T101 additionally allows the selection of ITU-T (formerly CCITT) standards that are compatible with EIA standards RS-232 and RS-485, and also support fiber optics interfaces. **[ref. 1]**

T101 specifies frame format FT 1.2, chosen from those offered in IEC 870-5-1 (1990) to provide the required data integrity together with the maximum efficiency available for acceptable convenience of implementation. FT 1.2 is basically asynchronous and can be implemented using standard Universal Asynchronous Receiver/Transmitters (UARTs). Formats with both fixed and variable block length are admitted. Also, the single control character 1 transmission is allowed.

3.2 Data Link Layer

At the data link layer, T101 specifies whether an unbalanced or balanced transmission mode is used together with which link procedures (and corresponding link function codes) are to be used. Address for each link is also provided.

The link transmission procedures selected from IEC 870-5-2 (1992) specify that SEND/NO REPLY, SEND/CONFIRM, and REQUEST/RESPOND message transactions shall be supported as necessary for the functionality of the end device. Additionally, T101 defines the necessary rules for devices that will operate in the unbalanced (multidrop) and balanced (point-to-point) transmission modes.

3.3 Application Layer

Application layer define Application Service Data Unit (ASDUs) from a given general structure in IEC 870-5-3 (1992). The sizes and the contents of individual information fields of ASDUs are specified according to the declaration rules for information elements defined in the document IEC 870-5-4 (1993). Type information defines structure, type, and format for information object(s), and a set has been predefined for a number of information objects.

The predefined information elements and type information does not allow the addition by vendors of new information elements and types. Information elements in the T101 profile have been defined for protection equipment, voltage regulators, and for metered values, to interface these devices as IEDs to the RTU. T101 utilizes the following basic application functions, defined in IEC 870-5-5 (1995), within the user layer. **[ref. 1]**

a) Station initialization
b) Cyclic data transmission
c) General interrogation
d) Command transmission
e) Parameter loading
f) File transfer
g) Data acquisition by polling
h) Acquisition of events
i) Clock synchronization
j) Transmission of integrated totals
k) Test procedure

Finally, T101 defines a mechanism for interoperability within a particular system. It is recognized that the companion standard defines parameters and alternatives from which subsets are chosen to implement particular telecontrol systems. Figure 6 shows the ASDU frame details showing various fields that are either fixed or variable per ASDU.

.

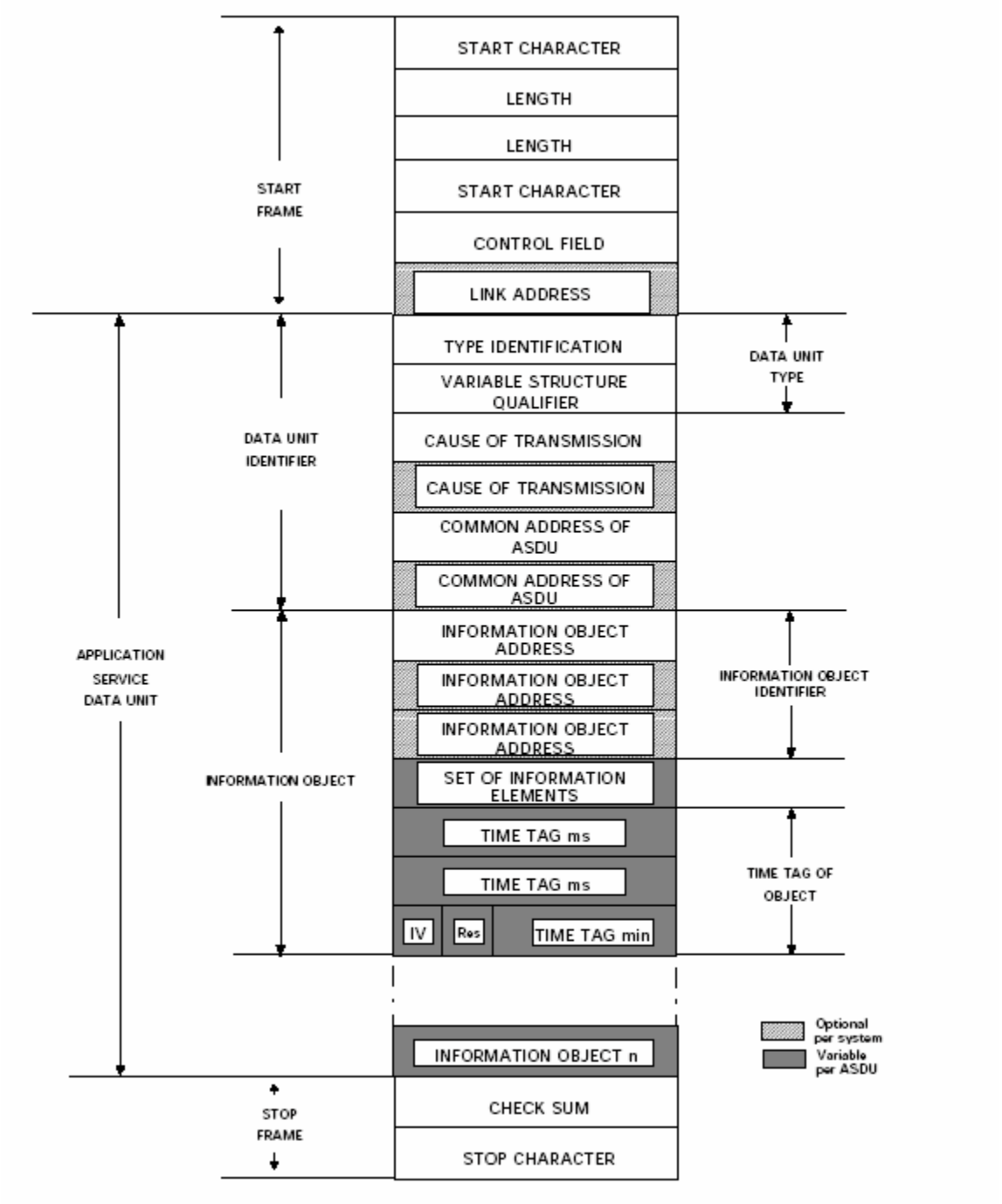


Figure 6 - ASDU frame details for IEC 870-5-101 [ref. 4]

As a guide for achieving interoperability within a system, T101 provides a checklist that a vendor can use to describe a device from a protocol perspective. Wherever choices can be made, such as baud rate, common address of ASDU field length, link transmission procedure, basic application functions, etc., there is a list that can be checked off, indicating the subset of supported services. Also contained in the check off list is the information, which may be contained in the ASDU in both the control and monitor directions. Table 2 presents information on functions and/or messages, which are applicable to the IED/RTU communication functions, using a common name to relate similar operations in the implementation of DNP3.0 and IEC 870-5-101.

4. Modbus [ref. 5, 7 & 8]

Modbus is an application layer messaging protocol, positioned at level 7 of the OSI model, which provides client/server communication between devices connected on different types of buses or networks. The industry's serial de facto standard since 1979, truly open and the most widely used network protocol in the industrial manufacturing environment. The Modbus protocol provides an industry standard method that Modbus devices use for parsing messages.

The Internet community can access Modbus at a reserved system port 502 on the TCP/IP stack. Modbus is used to monitor and program devices; to communicate intelligent devices with sensors and instruments; to monitor field devices using PCs and HMIs. **[ref. 8]**

4.1 Communication between Modbus devices

Modbus devices communicate using a master-slave technique in which only one device (the master) can initiate transactions (called queries). The other devices (slaves) respond by supplying the requested data to the master, or by taking the action requested in the query. A slave is any peripheral device (I/O transducer, valve, network drive, or other measuring device), which processes information and sends its output to the master using Modbus. Masters can address individual slaves, or can initiate a broadcast message to all slaves. Slaves return a response to all queries addressed to them individually, but do not respond to broadcast queries

4.2 Modbus Register Map

Modbus devices usually include a Register Map. Modbus functions operate on register map registers to monitor, configure, and control module I/O. You should refer to the register map for your device to gain a better understanding of its operation.

4.3 Serial Transmission Modes of Modbus networks

The transmission mode defines the bit contents of the message bytes transmitted along the network and how the message information is to be packed into the message stream and decoded. Standard Modbus networks employ one of two types of transmission modes:

ASCII Mode
RTU Mode

The mode of transmission is usually selected along with other serial port communication parameters (baud rate, parity etc.) as part of the device configuration.

- **ASCII Transmission Mode**

In the ASCII Transmission Mode (American Standard Code for Information Interchange), each character byte in a message is sent as 2 ASCII characters. This mode allows time intervals of up to a second between characters during transmission without generating errors.

- **RTU (Remote Terminal Unit) Transmission Mode**

In RTU (Remote Terminal Unit) Mode, each 8-bit message byte contains two 4-bit hexadecimal characters, and the message is transmitted in a continuous stream. The greater effective character density increases throughput over ASCII mode at the same baud rate. **[ref. 8]**

4.4 Modbus Message Framing

A message frame is used to mark the beginning and ending point of a message allowing the receiving device to determine which device is being addressed and to know when the message is completed. It also allows partial messages to be detected and errors flagged as a result.

A Modbus message is placed in a message frame by the transmitting device. Each word of this message (including the frame) is also placed in a data frame that appends a start bit, stop bit, and parity bit. In ASCII mode, the word size is 7 bits, while in RTU mode; the word size is 8 bits. Thus, every 8 bits of an RTU message is effectively 11 bits when accounting for the start, stop, and parity bits of the data frame.

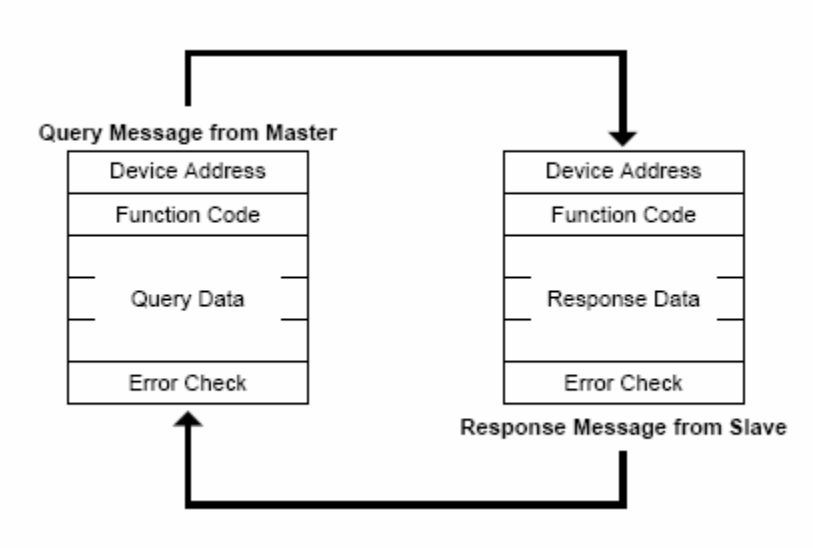


Figure 7: Master slave query response cycle [ref. 5]

One must not confuse the message frame with the data frame of a single byte (RTU Mode) or 7- bit character (ASCII Mode). The structure of the data frame depends on the transmission mode (ASCII or RTU). Note that on some other network types and on Modbus Plus, the network protocol handles the framing of messages and uses start and end delimiters specific to the network.

4.5 Modbus Addresses

The master device addresses a specific slave device by placing the 8-bit slave address in the address field of the message (RTU Mode). The address field of the message frame contains two characters (in ASCII mode), or 8 binary bits (in RTU Mode). Valid addresses are from 1-247. When the slave responds, it places its own address in this field of its response to let the master know which slave is responding.

4.6 Modbus Functions

The function code field of the message frame will contain two characters (in ASCII mode), or 8 binary bits (in RTU Mode) that tell the slave what kind of action to take. Valid function codes are from 1-255, but not all codes will apply to a module and some codes are reserved for future use.

4.7 Modbus Data Field

The data field provides the slave with any additional information required by the slave to complete the action specified by the function code. The data is formed from a multiple of character bytes (a pair of ASCII characters in ASCII Mode), or a multiple of two hex digits in RTU mode, in range 00H-FFH. The data field typically includes register addresses; count values, and written data.

- If no error occurs, the data field of a response from a slave will return the requested data.
- If an error occurs, the data field returns an exception code that the master's application software can use to determine the next action to take.

4.8 Modbus Error Checking

Modbus networks employ two methods of error checking: parity checking

1. Parity checking of the data character frame (even, odd, or no parity)
2. Frame checking within the message frame (Cyclical Redundancy Check in RTU Mode, or Longitudinal Redundancy Check in ASCII Mode)

4.9 Modbus Exceptions

If an unsupported function code is sent to a module, then the exception code 01 (Illegal function) will be returned in the data field of the response message. If a holding register is written with an invalid value, then exception code 03 (Illegal Data Value) will be returned in the response message.

4.10 Modbus TCP/IP [ref. 8]

Modbus/TCP was invented by Modicon/Group Schneider and is today is one of the most popular protocols embedded inside the TCP/IP frames of Ethernet. Modbus/TCP basically embeds a Modbus frame into a TCP frame in a simple manner. This is a connection-oriented transaction, which means every query expects a response.

This query/response technique fits well with the master/slave nature of Modbus, adding to the deterministic advantage that Switched Ethernet offers industrial users. The use of OPEN Modbus within the TCP frame provides a totally scaleable solution from ten nodes to ten thousand nodes without the risk of compromise that other multicast techniques would give.

Modbus TCP/IP has became an industry de facto standard because of its openness, simplicity, low cost development, and minimum hardware required to support it. Modbus TCP/IP uses TCP/IP and Ethernet to carry the MODBUS messaging structure. Running Modbus TCP/IP over the Internet, one won't get better than typical Internet response times. However, for communicating for debug and maintenance purposes, this may be perfectly adequate

5. Comparison of DNP 3.0, IEC 870-5-101 and Modbus

The following tables list the detail of the analysis.

Table - 1 provides comparison depending on the various features of the protocols.

Table - 2 presents information on functions and/or messages, which are applicable to the IED/RTU communication functions for DNP 3.0 and IEC 870-5-101, using a common name to relate similar operations in each of the implementations.

Table - 3 gives information on function as available in Modbus protocol.

Table 1 - Comparison of DNP 3.0, IEC 870-5-101 and Modbus [ref. 1, 9 & 11]

Feature	IEC 870-5-101	DNP 3.0	Modbus
Standardization	IEC Standard (1995) Amendments 2000,2001	Open industry specification (1993)	Not Applicable
Standardization Organization	IEC TC 57 WG 03	DNP user's group	Modicon Inc.
Architecture	3-layer EPA architecture	4-layer architecture Also supports 7 layer TCP/IP or UDP/IP	Application layer messaging protocol
Physical layer	Balanced Mode – Point to Point Multipoint to point Implementation by X.24 / X.27 standard Unbalanced Mode – Point to Point Point to Multipoint Implementation by V.24 / V.28 standard	Balanced mode transmission It supports multiple masters, multiple slave and peer-to-peer communication RS 232 or RS 485 implementation TCP/IP over Ethernet, 802.3 or X.21	Balanced mode of transmission RS 232 serial interface implementation Peer to peer communication TCP/IP over Ethernet
Data link layer	Frame format FT 1.2 Hamming distance – 4	Frame format FT3 Hamming distance-6	Two types of message frames are used:

Feature	IEC 870-5-101	DNP 3.0	Modbus ASCII mode and RTU mode
Application layer	Both IEC 870-5-101 and DNP 3.0 provides • Time synchronization • Time stamped events • Select before operate • Polled report by exception • Unsolicited responses • Data group/classes Limited to single data type per message Can control one point per message only No internal indication bits No application layer confirms for events	Remote starting / stopping of software applications Polling by data priority level Broadcast addressing Multiple data types per message are allowed Internal Indication field IID present in response header Application layer confirms events; use of CON bit is made	Does not give time stamped events. We have sequence of events (without time but not event list with time. Does not provide polled report by exception Checksum ensures proper end-to-end communication
Device Addressing	Link address could be 0, 1, 2 bytes Unbalanced link contains slave address Balanced link is point to point so link address is optional (may be included for security)	Link contains both source and destination address (both always 16 bits) Application layer does not contains address 32 b point addresses of each data type per device	Addresses field contains two characters (ASCII mode) or 8 bits (RTU mode) Valid address in range 1-247 Address 0 used for broadcast
Configuration Parameters required	Baud rate Device addresses Balanced / unbalanced Frame length Size of link address Size of ASDU address Size/structure of point number	Baud rate Device addresses Fragment size	Baud rate Mode – ASCII or RTU Parity mode

Feature	IEC 870-5-101	DNP 3.0	Modbus
Configuration Parameters required contd...	Size of cause of transmission		
Application Specific information model	A few application specific data types available Data objects and messages are not independent to each other	Permits vendors to create application specific extensions Data objects and messages independent to each other	Allows user to create application specific model
Cyclic transmission	Eliminates static data poll message from master Interrupted by event triggered communication request	Available but interval cannot be remotely adjusted	Not Applicable
Dominant market	Europe (South America, Australia and china)	North America (Australia and china)	Used worldwide for application with low volume data
Online configurations	Enable/ disable communication control objects Loading configuration Change report / logging behavior	Define group of data Selecting data for responding Enable/ disable communication control objects Loading configuration Change report / logging behavior	Efficient online configuration could be made by Modbus TCP/IP
Open for other encoding solutions	Not Available	Yes open for other encoding solutions like XML	Yes. One could write source code in programming languages like C, VC++ & JAVA etc.

Table 2 - Protocol message/function types for DNP and IEC 870-5-101 [ref. 1]

DNP 3.0 implementation		IEC 870-5-101 implementation	
Function **Code**	Description	**<Type ID> or (Tx cause)**	Description
0	Confirm	(P/N=0) (P/N=1)	Positive confirm Negative Confirm
1	Read	(1) <100> <101> <102> (5-6) (20) (21-36) (38-41)	Periodic, Cyclic Interrogation command Counter interrogation CMD Read command Request General interrogation Group interrogation Group counter request
2	Write	<120-126> (13) <110-113> <103>	File transfer Parameter of measured value Clock sync command
3 4 5 6	Select Operate Direct operate Direct operate no ack	<45-51> (6,8)	Single/double command Set point commands regulating Step CMD activation Deactivation
7 8 9 10 11 12	Immediate freeze Immediate freeze – no ack Freeze and clear Freeze and clear – no ack Freeze with time Freeze with time –no ack	<113>	Parameter activation (parameter equals time period memorization of integrand totals)
13 14 15 16	Cold restart Warm restart Init data to default Initiate application	(4)<70>	Initialized End of initialization
17 18	Start application Stop application	<105>	Reset process command
19	Save configuration	<120-126>(13) <113>	File transfer Parameter activation
20	Enable unsolicited		
21	Disable unsolicited		
22	Assign to class	(20-41)	Group interrogations
23	Delay measurement	<103>	Clock sync command
129	Response	(11) (12) <7> <7> <10> <1-21>	Return info – local CMD Return info – remote CMD Activation confirmation Deactivation confirmation Activation termination Process info – monitor direction
130	Unsolicited response	(1) (3) <104> <106>	Periodic, cyclic Spontaneous Test command Delay acquisition command

Table –3 Protocol Function Types for Modbus [ref. 5]

Function Code	Function
01	Read Coil Status
02	Read Input Status
03	Read Holding Registers
04	Read Input Registers
05	Force Single Coil
06	Preset Single Register
07	Read Exception Status
08	Diagnostics
09	Program 484
10	Poll 484
11	Fetch Comm. Event Ctr
12	Fetch Comm. Event Log
13	Program Controller
14	Poll Controller
15	Force Multiple Coils
16	Preset Multiple Registers
17	Report Slave ID
18	Program 884/M84
19	Reset Comm. Link
20	Read General Reference
21	Write General Reference
22	Mask Write 4X Register
23	Read/Write 4X Registers
24	Read FIFO

6. How to choose a protocol for an application?

To choose a protocol we have to look after the following questions:

Q.1) What is your application domain? Factory? Utility? Power?

IEC 870-5-101 and DNP 3.0 are comparable protocols mainly used in Utilities, Oil & Gas Industries and with some applicability in other domains. However Modbus is more of a general purpose protocol mainly intended at Industrial applications with direct register mapping and amount of data transfer is not large.

e.g. If it is power or energy industry, need to interface with SCADA systems with time-stamping and similar requirements makes IEC 870 and DNP suitable one.

Q.2) Is it communication from master to RTU or from master to numerical relays or master to IEDs or between two applications?

For different communication options we have options as explained below:

Communication within substations: There are protocols that are used for communicating with the devices meant for protection control and metering. The most common protocols are:

- Modbus
- IEC 870-5-103
- LON
- Profibus
- UCA
- IEC 61850

Some of the proprietary protocols are:
- SPA (ABB)
- VDEW (Siemens)
- K-BUS (Alstom)

Communication outside the substations: Communication protocols used for communication of data from substation to master control centers are:
- IEC 870-5-101
- IEC 870-5-104
- DNP 3.0
- IEC 60670-6 (TASE .2)
- IEEE P1525
- ELCOM90

Communication between applications: There are standards that are being defined for interfaces between various application e.g. IEC 61968

Q.3) What are the specific requirements with regard to amount of data, bandwidth, response time & longest distance between two devices?

When large area is to be covered than DNP 3.0 and IEC 870-5-101 presents good solution. DNP 3.0 sends small number of large sized data while IEC 870-5-101 sends large number of small sized data. Thus when voluminous data is to be communicated over a large distance DNP 3.0 becomes favored one. Also DNP networks works with higher baud rates than that of comparable IEC 870-5-101.

If you are looking for a simple protocol where memory requirements are less and a few number of data types, then MODBUS is a better option because of its smaller frame size compared to other protocols and simplicity in implementation. Modbus is a very fast protocol (it packs a lot of information in just one message). Regarding data integrity, it's very safe, since you always have to poll the process (there is no spontaneous sending).

Q.4) What kind of devices does one want to equip? Embedded devices, PLCs, PCs?

In small-embedded controllers and boards, where available memory is small Modbus becomes good option to implement.

Q.5) What functions one wants to carry out at the interface? Is it required to remotely parameterize the relays, download disturbance data and events or it is just required to retrieve some measurands?

One can see that while Modbus is "just a bunch of bits", IEC and DNP have much more functionality. If you choose Modbus, you will have to implement master and slave in order to obtain that functionality. And since you have to implement on both sides, you cannot assure that systems from different vendors will have the same.

If it is a simple register read/write and similar features is what you require and then it will be adequate to use Modbus for the project.

Q.6) What are the key players in your domain?

If, lets say it is Siemens with their PLCs Simatic S7, then Profibus could be the right solution - depending on the requirements.

Q.7) What is the geographical location? (Which country…)

Many times choice of the protocol between two options is made depending upon what country the target application will be in. For example, in North America DNP3 would be the obvious choice, while in Europe IEC 60870-5 would be the clear winner

Acknowledgement

I would like to thank our faculty adviser Prof. P.C. Pandey for his invariable support and for providing the opportunity to present the seminar. I would like to express my deep gratitude to Prof. L.R. Subramanyan for his expertise guidance and motivation throughout, for compilation of the report.

I am thankful to Mr. Jim Coats (Vice President, DNP User's Group), Mr. Ganesh Okade (Sunlux Technologies Ltd., India) and Mr. Karlheinz Schwarz (SCC, Germany) for their insightful suggestions on the topic.

Also I would thank IIT, BOMBAY for providing the needful resources for making the report and my colleagues who helped me to come up with this report.

References:

[1] IEEE Std 1379-1997, "IEEE Trial-Use Recommended Practice for Data Communications Between Intelligent Electronic Devices and Remote Terminal Units in a Substation", http://ieeexplore.ieee.org/iel4/5327/14435/00660326.pdf

[2] IEEE Std C37.1-1994, "IEEE Standard Definition, Specification, and Analysis of Systems Used for Supervisory Control, Data Acquisition, and Automatic Control", http://ieeexplore.ieee.org/iel1/3389/10055/00478424.pdf

[3] DNP 3.0 Protocol Functions Specification Set - "The Basic Four", available from DNP User Group, www.dnp.org

[4] "Norwegian IEC 870-5-101 User Conventions", Revision no. 2.0, www.statnett.no/Files/Open/IEC-R20_1.pdf

[5] PI–MBUS–300 –1996, "Modicon Modbus Protocol Reference Guide", http://www.eecs.umich.edu/~modbus/documents/PI_MBUS_300.pdf

[6] Official URL for IEC Protocols www.iec.ch

[7] Official URL for Modbus Protocols, www.modbus.org

[8] "Introduction to MODBUS", June '02, www.sena.com/support/technical_tutorial/

[9] Jim Coats, "Comparison of DNP and IEC 870-5-101", Presented at DA/DSM '97 meeting, DNP User's Group, Rev. Sept.1999, www.trianglemicrowoks.com

[10] Jim Coats, "DNP3 Protocol", Presented at AGA/GTI SCADA Security Meeting, Aug 2002, www.trianglemicrowoks.com

[11] Karlheinz Schwarz, "Comparison of IEC 870-5-101/-103/-104, DNP3 and IEC 60870-6-TASE.2 with IEC 61850", Feb '02, www.nettedautomation.com/news/n_44.html