DNP3 SPECIFICATION

Volume 1

DNP3 INTRODUCTION

Version 2.00 Draft G

11-November-2002

DISCLAIMER STATEMENT

DNP User Group documents and publications are not consensus documents. Information contained in this and other works has been obtained from sources believed to be reliable, and reviewed by credible members of the DNP User Group and/or the DNP User Group Technical Committee. Neither the DNP Users Group nor any authors/developers of DNP documentation guarantee, and each such person expressly disclaims responsibility for ensuring, the accuracy or completeness of any information published herein, and neither the DNP Users Group nor its authors/developers shall be responsible for any errors, omissions, or damages arising out of use of this document.

Likewise, while the author/developer and publisher believe that the information and guidance given in this work serves as an enhancement to users, all parties must rely upon their own skill and judgment when making use of it. Neither the author nor the publisher assumes any liability to anyone for any loss or damage caused by any error or omission in the work, whether such error or omission is the result of negligence or any other cause. Any and all such liability is disclaimed.

This statement was developed by the DNP User Group Technical Committee and represents the considered judgment of a group of software developers with expertise in the subject field. The DNP User Group is a global forum for users and implementers of the protocol and promotes implementers and developer information and interaction exchange. This work is published with the understanding that the DNP User Group and its authors/developers are supplying information through this publication, not attempting to render engineering or other professional services. If such services are required, the assistance of an appropriate professional should be sought. The DNP User Group is not responsible for any statements and/or opinions advanced in this publication.

NOTICE OF RIGHTS - DNP USERS GROUP

The DNP Users Group has made every reasonable attempt to ensure the completeness and accuracy of this document. However, the information contained in this manual is subject to change without notice, and does not represent a commitment on the part of the DNP Users Group. Copies of the latest documentation are available through the DNP Users web site at www.dnp.org.

TRADEMARK AND COPYRIGHT NOTICES

REVISION HISTORY

Version	Date	By Whom:	Reason for Changes
1.00	11-November-2002	DNP3 Technical Committee	Original release.

Contents

1 DNP3 PURPOSE AND HISTORY

This section discusses the creation and history of DNP3. The structure and operation of the protocol may be easier to understand when taken in the context of the problems the designers of DNP3 intended to solve.

1.1 The Tower of Babel

Westronic Incorporated developed DNP3 between 1992 and 1994, intending it to be the first truly open, truly useful protocol standard in the utility industry. Westronic was a manufacturer of remote terminal units and a system integrator based in Calgary, Canada. It had made a reputation converting between the hundreds of proprietary utility protocols in use at the time. This was not an easy task, however, and Westronic management had become frustrated with trying to make its devices compatible with so many proprietary protocols.

A proposal was made that Westronic should develop its own protocol, but then release it to the industry. The new protocol would incorporate the best features of the many protocols Westronic had encountered, plus some new ideas. Westronic would place the specification under the control of an independent users' group. Both utilities and vendors would be invited to be members, including Westronics' competitors. Westronic would not receive any money for the sale and distribution of the specification.

1.2 Voices in the Wilderness

Westronic was not the first to propose an open standard for the utility industry, but the designers of DNP3 did not find any of the existing efforts suitable. At the time when Westronic was considering DNP3, there were two main candidates available for an open protocol:

- The Electrical Power Research Institute (EPRI) had recently released the *Utility Communications Architecture* version 1.0. However, version 1.0 listed a choice of protocol profiles only, and did not define any object models or services suitable for performing SCADA functions. At that point in the development of UCA, very few utilities or vendors had provided input to the specification, and there were some serious concerns about bandwidth usage. These drawbacks and others eventually led to the development of UCA 2.0. UCA 2.0 became an IEEE technical report in 1998 and at this writing is in the process of becoming an International Standard.
- The International Electrotechnical Commission (IEC) had developed the first few documents in the IEC 60870-5 series of specifications, including the specifics of the data link layer and general definitions for the application layer. (At that time it was called just 870-5.) Westronic had been participating in this effort, but felt that it was progressing too slowly. Furthermore, the IEC had provided many options in the specification and Westronic was worried the standard would not be restrictive enough to ensure interoperability. The IEC eventually released the 60870-5-101-companion standard in 1995 to address these issues.

In 1992, the IEC work seemed to be the more complete of the two efforts, and had wider industry support at the time. Westronic therefore decided to base DNP3 on the IEC work already

completed. Even now, the feature sets of IEC 60870-5-101 and DNP3 are very similar, because the design teams built them on the same basic research.

UCA was not forgotten. Westronic (by then called Harris Distributed Automation Products) circulated versions of the DNP Basic 4 Document Set including a paper called "On the Road to Utility Communications Architecture". The thesis of this paper was that by standardizing on DNP, utilities would at least be reducing from many protocols to one. This would make it easy for utilities to later change to use UCA. However, very few design elements of UCA found their way into the DNP specification, other than a generally layered architecture.

1.3 Keeping it Small

The designers of DNP3 built it with several goals in mind, but the one that had the most impact on the final protocol was the industry's desire to limit the amount of bandwidth used. At that time, utilities considered a link running at 1200 bits per second to be fairly quick. (Yes, there are areas where this is still true). LANs were for office computing only, and the thought of trusting one's SCADA network to a third party telecom provider was heresy.

Power utilities had heard about layered protocols and the OSI model, but they were unconvinced of their value in a SCADA protocol. The Internet was beginning to boom, of course, but most utilities considered those protocols were for business computing only. They were not for a SCADA network. Those who followed such things may also have heard that there was a backlash against the OSI model brewing. Protocols like ATM and Frame Relay promised higher performance by eliminating layers. No utility at that time would have used these protocols in their network, but they probably heard that "layers are bad".

Therefore, the designers of DNP3 gave themselves a design goal to reduce bandwidth and use as few layers as possible.

This goal combined with the desire to be compliant with IEC 60870-5 resulted in the "Transport Function" as it now exists: a header that is not quite part of the data link layer, and yet not quite a complete transport layer. A later section will discuss the Transport Function in more detail.

1.4 Paranoia is Good

While requesting less bandwidth, utilities refused to compromise on the requirement that a SCADA protocol be extremely reliable. Early bit-oriented protocols had acquired a bad reputation because a change of a single bit could result in a device operating the wrong switch. This led to utilities requiring in bid specifications that vendors build select-before-operate, "I tell you twice" functions into all protocols. A few bad experiences made utilities paranoid about reliability to the point of writing it into contracts.

Therefore, when designing a frame format to use, the DNP3 designers chose the most reliable format they could find. The IEC had done extensive modeling on reliability and documented the results in IEC 60870-5-1. Rather than re-invent the wheel, the designers picked the most reliable of the several formats described in that specification, Frame Type 3 (FT3).

In the years that have passed, this decision has proven to be a good one. Many vendors have cursed the calculations necessary for the many Cyclic Redundancy Checks (CRCs). Many system engineers have cursed the extra bandwidth overhead. However, DNP3s reputation for reliability started well and has only improved with the years.

1.5 Do What You Do Best

Because the designers of DNP3 were from a systems integration company, they tried to incorporate into DNP3 the best features of all the utility protocols they had encountered. These features included:

- *Broadcasting.* The ability to send a single message to multiple devices.
- *Select-Before-Operate – Or Not.* The ability to choose to use extra reliability when operating an output, or to choose not to use it.
- *Time-Stamped Data.* Some of the most popular utility protocols, such as Modbus, had no way to accurately time-stamp data. Vendors and utilities were forced to develop proprietary work-around solutions. Other protocols supported timestamps on binary data only. DNP3 permits timestamps on almost all data. This is a feature that is only now beginning to become popular as utilities are starting to gather other types of historical data beyond the standard binary "sequence of events" log.
- *Accurate Time Synchronization.* Many earlier protocols had no way to account for transmission and software delays when synchronizing. The method used in DNP3 is an amalgamation of several different protocols' solutions.
- *Quality Flags.* Representing a maker of data concentrators, the designers ensured that there was a way to see whether data was valid, and why. Some protocols, designed by IED vendors whose data was always online, did not include this feature.
- *Multiple Data Formats*. The ability to report data in a variety of formats: 16-bit, 32-bit, with a flag, without a flag, floating-point, BCD, packed, unpacked, and so on.
- *Scan Groups.* The ability to define and ask for a large set of otherwise unrelated data using a single request.
- *Layer Separation.* Separating the function of "getting the data there" from the actual SCADA functions.
- *Report-by-Exception*. More than any other feature, the ability to *reliably* report only the changes in data has helped make DNP3 successful.
- *Internal Indications*. As several protocol efforts that are more recent than DNP3 have discovered, it is extremely useful to have a global set of flags returned in each response. These flags indicate the health of the device and the results of the last request.

Most of these features had been seen elsewhere, but this was the first time an open utility protocol had attempted to do them all.

1.6 Tell Me Again Why That's In There?

Unfortunately, the "best practices" approach to developing DNP3 was not perfect, causing a number of features to be added that were not really in widespread use. A number of them existed only in Westronic equipment. At various times, vendors have questioned the need for:

- So many different types of Counters, particularly Delta Counters
- So many different types of binary output operations, especially control queuing
- So many different ways to format data, i.e. many qualifier codes

- Pattern Masks
- Binary Coded Decimal Analogs
- Storage Objects
- The ability to either WRITE or OPERATE an output
- So many layers of confirmation and segmentation

The first result of this "abundance of riches" was the publishing of the *DNP Subset Definitions,* which told vendors what they *really* had to do to implement DNP3. Over the years, the DNP Technical Committee was forced to issue a number of technical bulletins to clarify the use of those items that had not been adequately addressed by the *Subset Definitions*. All of these documents have been incorporated into the revised version 2.0 of the specification.

1.7 The Intelligent Network

Another trend in the early 1990's was the move to put larger processors and more memory in SCADA devices. Marketing and sales people were talking about "the intelligent network". By this, they meant pushing many of the functions previously performed only by master stations into remote devices. These devices would be more independent and make more decisions on their own. Those who join the utility industry these days are sometimes confused by the term "IED" meaning Intelligent Electronic Device. They say, "Aren't all computing devices intelligent?" Yes, but it wasn't always this way.

In terms of DNP3 design, the idea of "the intelligent network" translated to the following features:

- *Spontaneous Reporting*. A device could transmit whenever it wanted, not just when polled by the master. On multi-drop links, this led to the need for a collision avoidance mechanism.
- *Meta-Data*. The DNP3 designers called a spontaneous message an "Unsolicited Response", which shows the mindset in those days. Most devices only sent data in response to a poll request. Therefore, the master always knew what data was coming. For a device to send an unsolicited message, it had to include not only the SCADA data itself, but also information *describing* the data so the master knew what it was. The term these days for such information is *meta-data.* It appears in such modern technologies as Extended Markup Language (XML). At that time, though, it was a very new concept for the utility industry.
- *Wild-Carding.* Because the remote device was more intelligent, the designers gave it more choice in the amount and format of the data it reported. A master could ask very simple questions, like "Give me all your data" or "Give me your analog changes" and get very complex answers. Again, because the answer did not exactly match the question, meta-data was required in the response.
- *Self-Description.* The idea that a device could tell the master what data it had available, and how to present it, was already around thanks to UCA 1.0. The DNP3 designers tried to incorporate some of this ability into DNP3. The Device Profile Object and the use of floating-point with the units transmitted were considered very advanced. Perhaps they were too advanced, because they appeared in very few implementations.
- *Vendor-Specific Expansion.* The DNP3 specification includes the Private Registration Object, which permits vendors to add proprietary extensions to the basic standard. The

Private Registration Object Descriptor permits a standard implementation to parse these extensions even though they are proprietary. These objects, too, have not been very popular, but a few vendors have used them to good effect.

- *File Transfer.* The designers gave DNP3 file transfer capabilities so that an intelligent device could download new configuration or software, or upload oscillography files. At the time DNP3 was developed, few devices had flash memory, and only specialized fault recorder devices performed oscillography. Now both are widespread.
- *Program Control.* The ability to start and stop individual programs and processes on a remote device was common in the factory automation industry. DNP3 provides a rudimentary mechanism to do this.

The dream of the "intelligent network" has had mixed results. Some of these features, like spontaneous reporting, meta-data, prioritization and wild-carding, have worked very well. They are probably some of the main reasons for DNP3's popularity. Other features, like self-description, file transfer, floating-point, program control, and collision avoidance, were not completely thought out. The DNP Technical Committee was forced to revise these and issue technical bulletins clarifying their use. Some features have died a death of obscurity.

However, history should not be a harsh judge. Many people take such features for granted these days, but it is important to remember that DNP3 was there first.

1.8 Wish List for Data Comm Geeks

Because of the intense pressure to reduce bandwidth, and because the DNP3 designers had more expertise in SCADA than in general data communications, a number of common communications features were "left out" of the DNP3 definition. Many designers have subsequently mourned the absence of these features. Some of them the DNP Technical Committee has attempted to "add on" afterward. Others the Committee could only achieve now at the cost of obsoleting all existing implementations.

The following list of missing data communications features illustrates how well the DNP architecture works despite the limitations imposed at its birth:

- *A network layer.* At one point, the designers actually wrote a specification for a DNP network layer, but Westronic management did not approve it. In retrospect, this is just as well, because the IP network layer now used is far more popular.
- *Application layer addresses*. The ability to select a particular logical device within a physical one would have been useful. Most devices that support this feature have found a way around it through local software mechanisms that use the Data Link address and/or physical port number as a key.
- *Application and transport layer sequence number initialization.* This has caused much grief over the years and has been addressed as well as possible without causing obsolescence. Data communications experts should note, therefore, that DNP3 is not quite connection-oriented and not quite connectionless, but somewhere in between.
- *Long sequence numbers.* DNP3 sequence numbers are very short, which is good for bandwidth but not for detecting duplicates. This is the reason TCP is required when using DNP3 over WANs, which turns out to be a very robust solution.
- *Sequence number in Data Link Confirms.* Without a sequence number, it is impossible to determine which Data Link frame a Confirm frame is answering. On a serial point-to-point link, this is not a problem, but on a WAN, Confirm frames could arrive out of order

or be lost. Using TCP in WANs addresses the issue on IP networks, but in theory, it could still cause problems in serial radio networks. In practice, it generally works anyway. This problem was inherited from IEC 60870-5 and cannot be changed without obsolescence.

- *Sliding window.* One constant of DNP3 has been that only one transaction can be outstanding at a time. In theory, a device could send several response fragments very quickly for a particular request, but over the years the Tech Committee has decided that interoperability is best served by enforcing a confirmation between each fragment.
- *Access security.* The designers of DNP3 purposely avoided dealing with this issue because of its complexity. Fortunately, it may be possible to add security features without completely re-writing the protocol.
- *Version Control.* Most protocols tend to have an octet reserved to show the version of the protocol in use. This was not included in the original DNP3 definition due to bandwidth reasons, but it will reappear as part of the new self-description solution.
- *Overall length field.* Segmentation and fragmentation would have been a lot easier and more robust, and the LAN implementation would have been easier if each fragment had a length field at the beginning. It was not included for bandwidth reasons. Again, various software solutions make it work anyway, so perhaps it was the right decision.

1.9 So Is It IEC Compliant or Not?

As discussed earlier, there were two reasons why the DNP3 designers wanted it to be compliant with the IEC 60870-5 specifications:

- They wanted to take advantage of the excellent technical work done on reliability in the 60870-5 data link layer specifications.
- They wanted to increase the acceptance of the protocol by showing it was based on standards work that was already well known.

They were so successful in both efforts that even now, some people are confused about whether DNP3 and IEC 60870-5 are interoperable.

The answer is that they are not interoperable, although the DNP3 data link layer could be considered compliant to IEC 60870-5 Parts 1 and 2. DNP3 was based on the drafts available at the time of IEC 60870-5 Parts 1 through 5. These Parts of the specification described the data link layer in great detail and the application layer in general. There were several options specified for the data link layer.

The DNP3 designers chose those options of Parts 1 and 2 they thought were most appropriate. Unfortunately, when the IEC 60870-5-101 companion standard was released with the details of the application layer, it specified *different* data link layer options than those the DNP3 designers had chosen.

Therefore, DNP3 is considered compliant with IEC 60870-5-1 and 60870-5-2, but not 60870-5-101.

Table 1.9–1 shows the differences in the data link layers of the two protocols.

Table 1.9–1 Comparison of IEC 60870-5 and DNP3 Data Link Layers

Feature	Options Permitted in IEC 60870-5-1 and 2	Chosen by DNP3	Chosen by IEC 60870-5-101
Addressing	Single address, length system-dependent	Two-octet Source address and two-octet Destination address. Considered a single four-octet "structured" address for compliance purposes.	Single address, choice of either zero, one or two octets in length
Frame Format	Choice of FT1.1, FT1.2, FT2, FT3	FT3, transmitted asynchronously	FT1.2
Reliability Mechanism	Varies per frame type	Multiple 16-bit CRCs over each 16 octets of a 255 octet frame. Start and Stop bits, but no parity.	Parity bits and one-octet checksum (not CRC) calculated over 255 octets
Hamming Distance	Varies per frame type	6 for the original FT3. Some debate about the value as currently used. See further discussion in this section.	4
Acknowledge-ments	Either Fixed-length or single-octet	Fixed 10-octet only	Either fixed-length or single-octet
Procedures	Balanced (no master) or Unbalanced (master polls)	Balanced only	Either Balanced or Unbalanced
Method for Multi-Drop Links	Unbalanced mode	Collision Avoidance	Unbalanced mode

1.9.1 The Hamming Distance Debate

Some critics of DNP3 have disputed DNP3's right to claim a Hamming Distance of six. The "Hamming Distance" of a protocol is the number of bit errors required in a frame before a receiver could incorrectly identify a corrupted incoming frame as a valid frame. Critics argue that the original calculation was made assuming the FT3 frame was transmitted synchronously, while DNP3 uses the FT3 frame format asynchronously.

The main concern in this debate is inter-character gaps. If a gap is permitted between the octets of a sixteen-octet block, noise could be introduced that might be mis-interpreted as valid data. Critics claim that the DNP3 specification has never required that all octets of a block be transmitted together, and this reduces the theoretical reliability of the protocol to below that of the FT1.2 frame.

However, years of use in hundreds of systems have proven DNP3's reliability to be more than sufficient for utility purposes. This may be due to the fact that most DNP3 devices tend to start a timer or other mechanism that will discard an incoming frame when inter-character gaps appear.

The inter-character concern with the DNP3 frame is similar to a problem that occurs in some IEC 608705-101 systems. The FT1.2 frame's reliability relies on the use of parity bits in each octet. However, many utilities mistakenly use the protocol with modems that do not add, or actually remove, such parity bits. The IEC is preparing a 60870-5 standard that clarifies parity bits *must* be used.

1.9.2 One Address or Two?

The other main issue concerning DNP3 compliance to 60870-5 was the structure of the address field. The IEC definition of the address field states that it is a single address, always addressing one end of the link. This is the way 60870-5-101 uses the address field.

By including both a source and destination in every message, the DNP3 designers permitted the use of multiple masters on the same link, and peer-to-peer communications. This proved to be a powerful argument in the acceptance of DNP3. Furthermore, since 60870-5-2 did not specify a particular length of address, a four-octet address that just happened to be "structured" with two sub-addresses could still be considered compliant.

1.9.3 The Verdict

Although it was the topic of lively debate when DNP3 was first released, the question of whether DNP3 complies with IEC 60870-5 is essentially a moot point today. DNP3 may be considered compliant to Parts 1 and 2. One could even argue that DNP3 complies with the spirit, if not the letter, of Part 5, the general application layer definition. However, the format of the IEC 60870-5-101 application layer is very different from that of DNP3. It is clear the two protocols could never interoperate.

It is better to consider the two protocol suites as cousins with a common family tree and leave it at that.

1.10 Transport Who?

The naming of the Transport Function always confuses newcomers to DNP3. Is it a true transport layer, is it a part of the data link layer, or is it something truly different?

The answer is that it really is something different, although it most closely resembles an additional field in the data link layer. It does not have its own addressing or acknowledgements, as a separate layer would. There was no network layer in the original protocol definition, so the transport header was terminated at the end of each physical link, just like the data link header. It doesn't have the long sequence numbers and other features that would really enforce transmitting frames in sequence. Therefore it doesn't seem to be a transport layer.

However, if it were a field of the data link header, it would be included in every data link frame, and it is not. Only those frames containing application layer data contain a transport header.

The reasons this strange "half-layer" exists are both political and technical. The designers of DNP3 decided they wanted the application layer data broken into small segments suitable for passing over noisy links. This capability would at a minimum require a new data link layer field. However, they did not want to add a new field for two reasons:

1. It would eliminate DNP3's chances to be considered compliant to IEC 60870-5. As discussed earlier, this was considered critical to DNP3's acceptance by the industry.
2. Changing the structure of the FT3 frame could possibly compromise the calculated reliability of the frame.

Therefore, the transport header was placed in front of the application layer header, in the user data field of the data link layer frame.

However, the next question was, "What to call it?" As noted in this section, it had some of the characteristics of a layer, but not all of them. Furthermore, the designers knew there would be resistance to any additional layers in the protocol. It was bad enough that they were dedicating *a whole octet* to the segmentation and reassembly functions.

Therefore, the name "Transport Function" was chosen, thus causing years of questions on hotlines, email and training presentations.

Whatever it is, it makes DNP3 distinct. Along with application layer fragmentation, it permits a small, low-powered device to report a nearly unlimited amount of data reliably over a noisy link.

1.11 Development of Organizational Features

One feature of DNP3 that newcomers do not always appreciate is the organization that stands behind it. Over the years, the DNP User's Group has contributed at least as much to the protocol's success as the technical features of the protocol itself.

In roughly chronological order, here are the organizational features that helped DNP3 become popular:

- Membership that included both utilities and vendors
- Low membership fees
- Low cost of the specifications
- A structure consisting of steering, technical, and marketing committees
- The *DNP Subset Definitions* document
- Suggested wordings for utilities to specify DNP
- Agreement that any non-backward compatible change must be approved by the General Membership
- The DNP hotline, later to become an Internet chat session
- Booths at major trade shows
- Publishing membership lists so utilities could see the lists of vendors
- The DNP bulletin board, later to become a web site
- The DNP email mailing list
- The DNP Technical Committee mailing list, which can include non-committee members
- Publishing the technical committee minutes so the process remains open
- Technical bulletins clarifying areas of dispute when interoperability issues arose
- Agreement, first informal and then formal, that the president should always be from a utility
- The *DNP IED Level 1 and Level 2 Test Procedures*
- The *DNP WAN/LAN Specification*

1.12 Summary

The following design goals, whether formally stated or not at the time, had a major impact on the structure of DNP3:

- Include the best features of the utility protocols in use at the time.
- Push the intelligence in the network toward the remote device.
- Try to comply as much as possible with existing standards efforts, especially IEC 60870-5.
- Use as little bandwidth as possible.
- Make it more reliable than anything that came before.

It is easy to see that these goals are necessarily contradictory. The resulting protocol was not perfect and has been "patched up" over the years. However, it remains popular, open, reliable, and mostly backward-compatible.

1.13 A Footnote: Naming the Protocol

Few people seem to know what to call this protocol. Everyone knows it is DNP, but is it "DNP 3", "DNP 3.0", "DNP V3.0", or any combination of the above? Also, there are several different subset levels of implementation, and a few non-backward-compatible changes have been made over the years.

As of Technical Bulletin TB2000-003: "Change Management", the official name of the protocol is DNP3-xxxx, where xxxx is the year of release of the Test Procedures to which a device complies. The subset level is specified afterward, as in "DNP3-2000 Level 2".

This naming convention represents an evolution of the name over the years:

- The original Basic 4 documentation referred to the protocol as "DNP V3.00". No one has ever liked saying the "V" part, so that name has never caught on.
- For those who were wondering: DNP V1.00 and DNP V2.00 are proprietary Westronic protocols that were rarely used even at the time DNP3 was released.
- The user's group is called just the "DNP User's Group". That saves it from having to worry about version numbers in marketing information.
- The *Subset Definitions* defined the format DNP-Lx, where x was the subset level. That never caught on either, since utilities preferred to spell out the words "Level x" in their bid specs.
- When the *Test Procedures* were first published in 2000, there had to be some mechanism to distinguish between an implementation that was compliant to the procedures and earlier implementations that were not. The Technical Committee therefore decided to use the year in the specification, similar to the format used by the ISO, IEEE, and IEC.
- The intent is that there will never be a DNP 4.0, or even a DNP 3.1. To help illustrate this commitment to backward compatibility, the User's Group changed the name from "DNP V3.00" to "DNP3". The name therefore remains recognizable while eliminating the "software version" impression that the decimal point gave.

Some people rightly complain that it is redundant to say “DNP3 protocol”, since the “P” in DNP3 stands for “Protocol” already. However, this truism doesn’t seem to discourage people from using the phrase, and it is likely to be heard for years to come.

Now if one could just figure out how a protocol that originally had no network layer ended up with the name “Distributed *Network* Protocol”. One long-standing DNP3 user points out that the networks in question are SCADA networks, which “bear scant resemblance to other things that people usually call networks.” Perhaps this is the case.

Ah well, a protocol by any other name is just as interoperable and reliable.

2 DNP3 OVERVIEW

2.1 Basic Messages and Data Flow

The below is a brief, but incomplete overview of DNP3 messages and data flow. Its purpose is to prepare the reader for what follows in the Application Layer, Transport Function and Data Link Layer volumes of the DNP3 Specification.

This initial discussion of DNP3 uses the master-outstation model illustrated in Figure 2.1-1. This section omits many details to purposely keep the description straightforward.

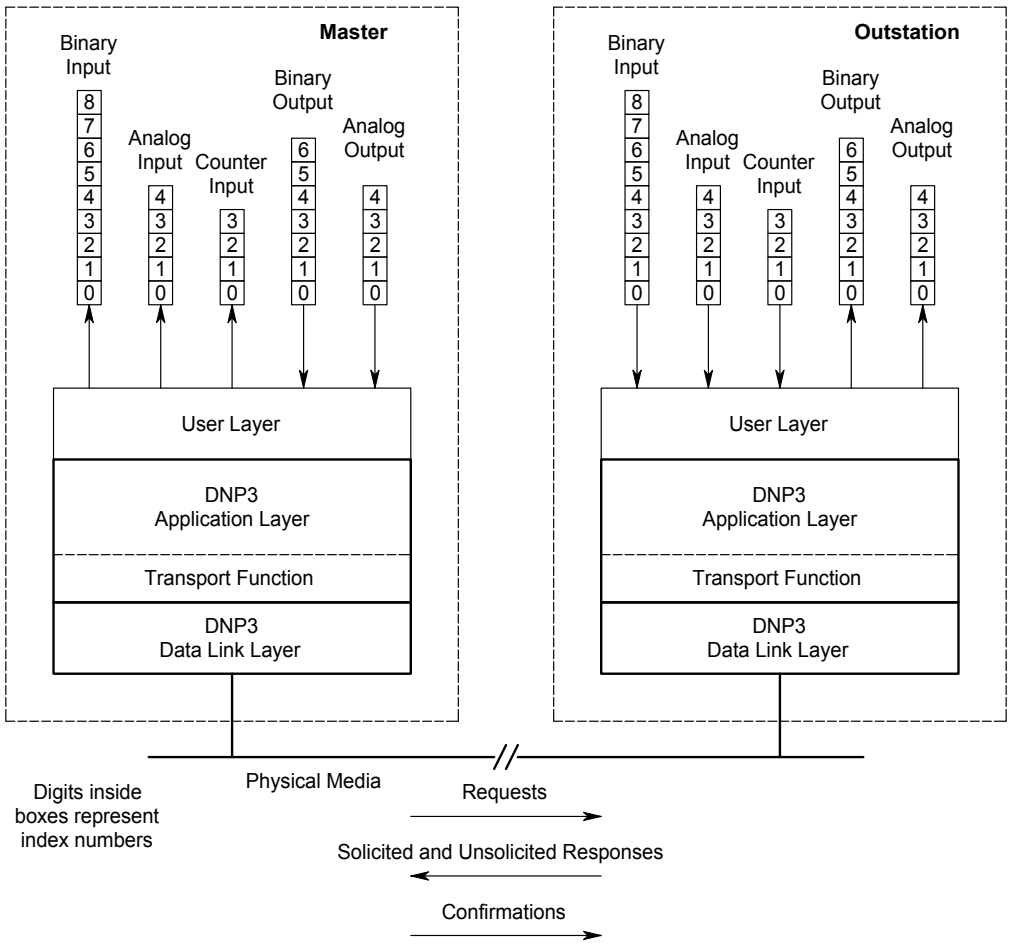


Figure 2.1-1

The User Layer in the master on the left side of the figure initiates a data transfer by causing its Application Layer to send a request to the outstation. The request contains a function code and zero or more DNP3 objects that specify what data is wanted. The Application Layer passes the request to the Transport Function for partitioning into transmission-sized units and then on to the

Data Link Layer. The Data Link Layer adds addressing and error detection information and transmits the packet to the outstation over the physical media.

At the outstation on the right side, the Data Link Layer receives the octets from the physical layer and checks for errors that were introduced while the packet was in transit. If no errors are detected, the addressing and error detection information added by the transmitting Data Link Layer is stripped from the message, and the remaining octets are passed to the Application Layer. If necessary, the Transport Function reassembles multiple packets into a complete request. The Application Layer then interprets the function code and DNP3 objects in the message and indicates to the User Layer what data is desired.

The User Layer in the outstation initiates a response based upon what data the master requested. It fetches data, classifies it and presents that data to the Application Layer. The Application Layer creates a message with data formatted into DNP3 objects, passes it through the Transport Function, and then on to the Data Link Layer for transmission to the master using methods similar to those employed by the master to send its request.

Upon receipt of the response at the master, the layers perform address and error checking and reassembly into a complete message for the Application Layer. This layer parses the DNP3 objects in the response and presents the information to the User Layer. The User Layer can then store or operate on that data in a way that is suitable for the end user.

The master always initiates control commands. These actuate device outputs or variables internal to the outstation. The DNP3 user-to-Application Layer interface and transmission procedures are similar to those discussed for data acquisition.

A transaction consists of a single request followed by a single response. A master sends a request and waits for the response, or a timeout, before issuing another request. Multiple transactions may simultaneously occur within a system. For example, consider the case where two masters each make requests to the same outstation.

In some systems the master does not always directly initiate data transfer. DNP3 has provision for the outstation to automatically send data when it detects a condition worthy of transmitting without a specific master request. “Unsolicited responses” is the terminology applied to this type of operation because the request is implied.

2.2 Layering

2.2.1 General

The ISO (International Organization for Standardization) defines a communication architecture that separates functions into seven layers called the Open System Interconnection (OSI) reference model. DNP3 protocol is based upon a simplified model termed the Enhanced Performance Architecture (EPA) that consists of only three layers: Application, Data Link and Physical. Figure 2.1-1 shows how DNP3 fits the EPA structure and communication model.

In theory, each layer in a layer stack performs a set of functions required to communicate with the same layer in another device, relying on the next lower layer for more primitive functions. At the sending device, each layer below the Application Layer receives data from the layer above for transmission. The layer adds more information that enables the equivalent layer in the receiver to properly process message. At the receiving device, layers examine their layer specific information added by the corresponding layer at the transmission site and process the message appropriately. The layer control information is stripped, and the message is passed to the next higher layer.

The Transport Function within the Application Layer performs a layer-like function of partitioning large messages into smaller messages that the Data Link Layer is capable of handling. The Transport Function is sometimes referred to as a "pseudo layer". In DNP3 the Application Layer, Transport Function and the Data Link Layer in the transmitter add information to the message for enabling the same layer or pseudo layer in the receiver to process the message.

2.2.2 Fragments, Segments and Frames

Figure 2.2-1 illustrates the partitioning of large messages at the Application Layer into smaller units and the addition of header information at each layer.

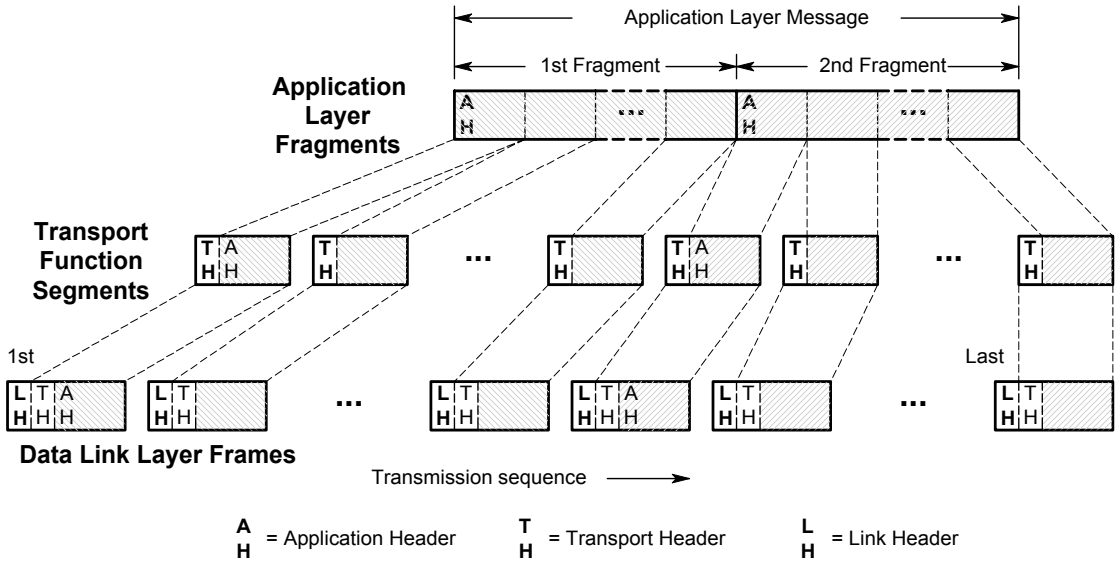


Figure 2.2-1

Figure 2.2-1 shows a fragmented Application Layer message, segmentation of each fragment by the Transport Function, and how segments fit into Data Link Layer frames. This diagram does not show timing and confirmation details, but serves to demonstrate how the higher level parts nest inside the lower layer structures. It also shows the relative positions of the Application Layer headers, the Transport Function headers and the Data Link Layer headers.

Table 2.2–1 provides a summary of the terminology and some brief information associated with each layer or function.

Table 2.2–1

Layer or Function	Unit Name	Information
Application Layer	Application Fragment	Permits the setting an upper limit on the memory requirements for message reception. Requests must fit into a single fragment. Responses may require more than one fragment.
Transport Function	Transport Segment	Segmentation breaks a fragment into pieces that fit into a Data Link Layer frame. Each segment contains a Transport Header, but only the first segment of any fragment contains an Application Header. Each segment may have a maximum of 250 octets including the Transport Header.
Data Link Layer	Data Link Frame	A Frame may have as many as 292 octets including its header and CRC octets. Frames are designed for superior error detection.

2.3 Message Sequences

Figure 2.3-1 illustrates a hypothetical sequence and the time relationship of fragments and frames as they move between layers, and between the master and outstation in a polled environment. Readers just beginning to learn the DNP3 specification are cautioned to only view the diagram to gain a general overview. Later, after studying the details refer back to this figure when it will be more meaningful.

Figure 2.3-2 illustrates a hypothetical sequence and the time relationship of fragments and frames as they move between layers, and between the master and outstation in a polled environment. Readers just beginning to learn the DNP3 specification are cautioned to only view the diagram to gain a general overview. Later, after studying the details refer back to this figure when it will be more meaningful.

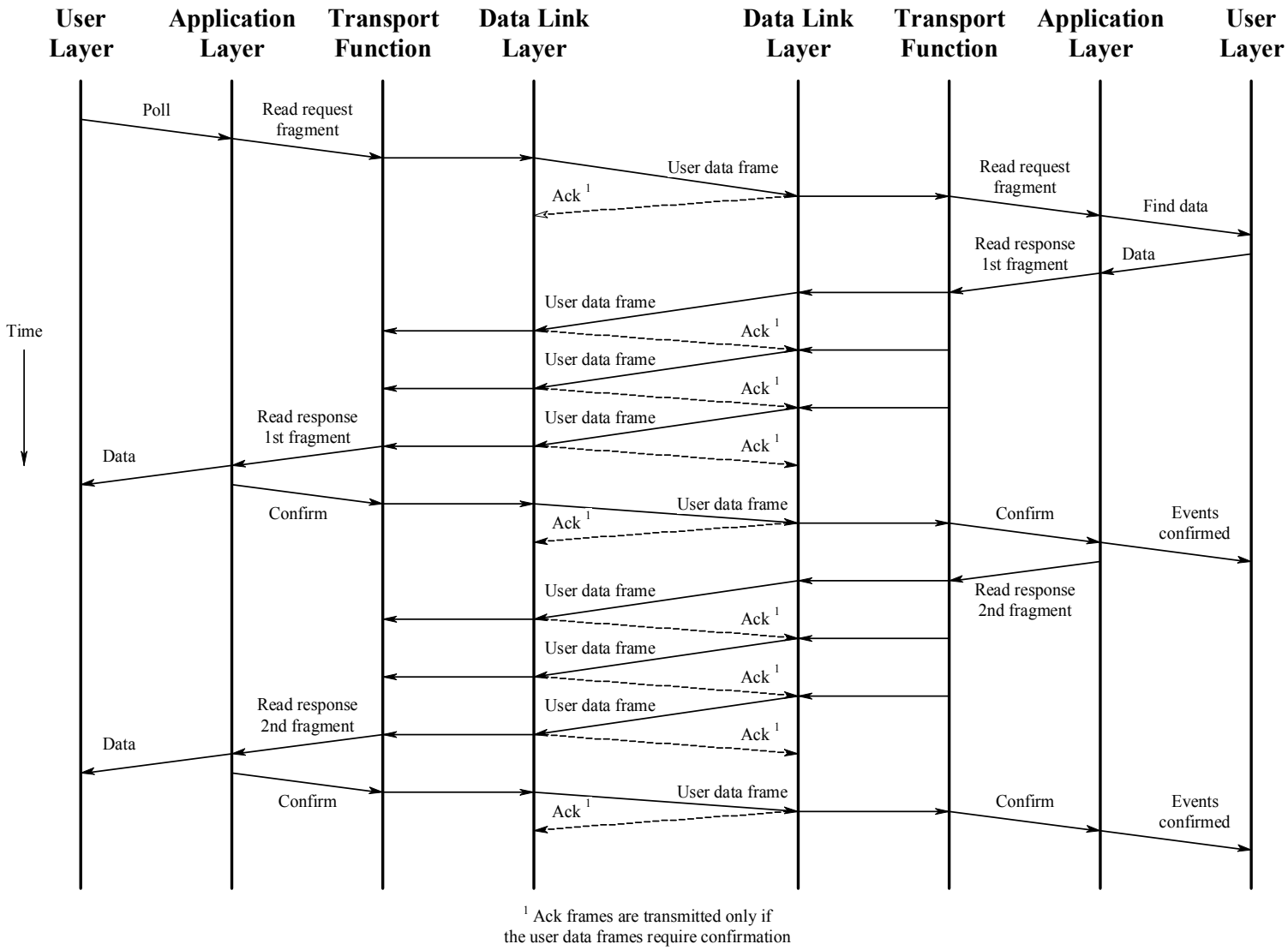


Figure 2.3-1 Polled Sequence with Link Layer Confirmation

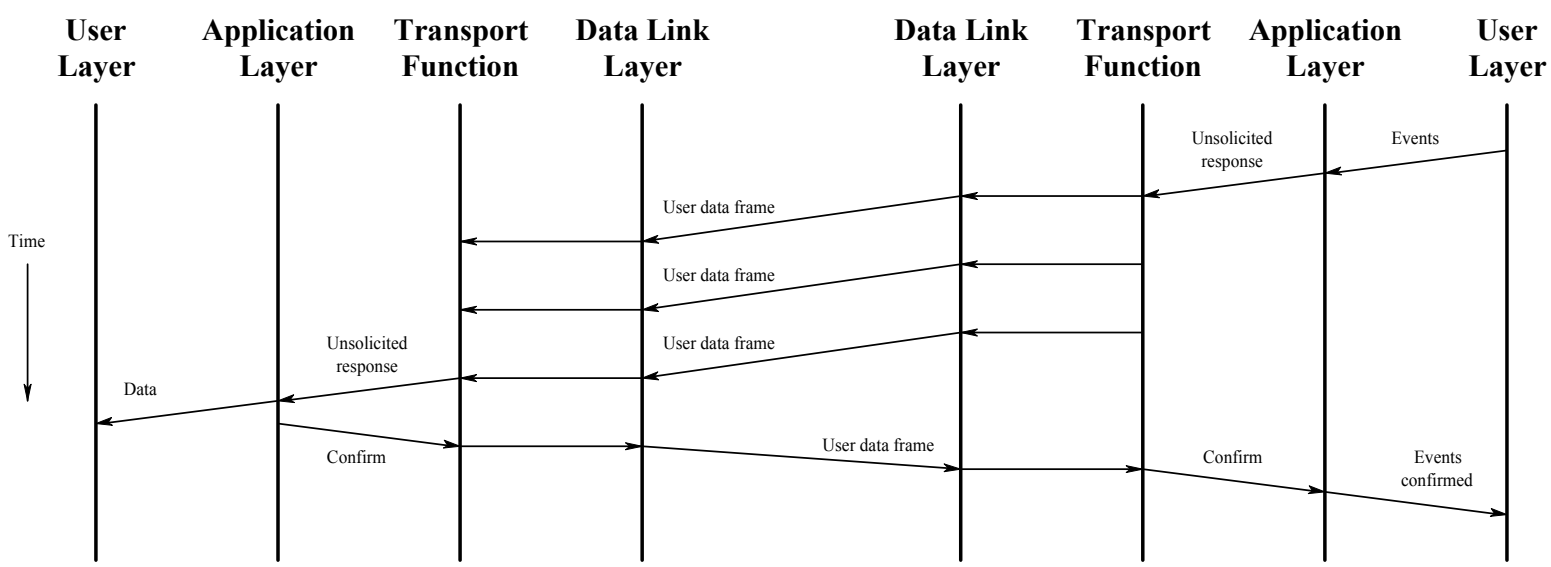


Figure 2.3-2 Unsolicited Response Sequence

2.4 Data Loss and Efficiency

One of the fundamental goals of DNP3 is to prevent loss of data transferred from an outstation to the master. Of special concern is transferal of all binary input states, in sequence, and without missing any transitions.

To increase the efficiency, DNP3 provides for report-by-exception whereby changes are transmitted soon after they occur, and an occasional integrity poll is issued to ensure that databases in the outstation and master are synchronized. When an outstation transmits changes, it must request application layer confirmation. Only after the master confirms receipt of the changes, can the outstation assume the changes arrived at the master.

Outstation devices that are able to report all of their current data in a single frame are not required to support report-by-exception.

3 ORGANIZATION OF DNP3 SPECIFICATION

The complete DNP3 Specification is organized into separate volumes wherein details of the DNP3 protocol are documented.

Volume 1:	DNP3 Introduction
Volume 2:	Application Layer
Volume 3:	Transport Function
Volume 4:	Data Link Layer
Volume 5:	Layer Independent Topics
Volume 6:	Data Object Library
Volume 7:	Networking, LAN/WAN
Volume 8:	Interoperability

4 CONVENTIONS USED IN THE DNP3 SPECIFICATION

4.1 Tips

Tips appear inside a box with an arrow to the left.

Tip

This is a tip box. These are used to highlight special information that is not part of the protocol specification but can help the reader.

Tip boxes also hold implementation suggestions.

4.2 Examples

Examples are preceded with a box describing what is illustrated below.

E

This example shows a request for all of the static binary inputs. Assume there are 18 binary inputs.

4.3 Wording – Required vs Option

The words **Must** and **Shall** are used to indicate *is required*.

The word **May** is used to indicate *is permitted, but is not required* or *is possible if certain relevant conditions are true*.

4.4 Single Master, Single Outstation Perspective

The DNP3 protocol is suitable for systems with one or more master stations, one or more outstations, and peer-to-peer arrangements. In general, this specification was written from the perspective of a single master and single outstation to make the documents easier to understand without the additional complexities involved.

A separate section is devoted to discussion of multi-master systems and their special considerations and requirements. Statements appear elsewhere only when it is necessary to emphasize specific characteristics or behavior for systems with multiple master or outstation devices.

4.5 Octet Order

Unless specified elsewhere, the least significant octet in multi-octet data values is transmitted first.

5 GLOSSARY

5.1 Words and Terms

The following provides definitions of words and terms used in the DNP3 specification.

DNP3 Event	The occurrence of something significant happening. Events are saved at the outstation as information in vendor-specific structures and reported to the master using DNP3 event objects. An event remains in the outstation until confirmation has been received indicating that a description of the event has arrived at the master, after which, the outstation must discard it. With a few exceptions, DNP3 does not define which events are worthy of transmission.
DNP3 Event Object	An object that has a group number and variation that is used to report an event in the outstation to the master.
DNP3 Object	The encoding within a message that refers to a single instance of a group and variation. DNP3 objects can associate with individual point indexes, a set of indexes or to an entire device.
Fragment	A packet of octets that is sized to fit into the buffers of the receiving device's Application Layer. Each fragment contains an application header and a portion of an Application Layer request, response or confirmation.
Frame	A packet of octets transmitted from the Link Layer in one device to the Link Layer in another device over the Physical Layer. Each frame contains a link header, CRC octets and sometimes a segment from the Transport Function.
Input	Refers to values that are measured, read or generated by the device and are reported by an outstation to a master. Examples are the level of fluid in a tank, the open-close state of switch and the calculated sum of the power on all three phases of a power line. Input sometimes refers to the physical source of the value such as a voltage sensor.
Local Issue or Local Matter	The subject of interest that is restricted to an individual device or system and not generally known to other devices, systems, vendors or persons. The method of measuring analog quantities in an outstation is a local issue.
Local Mode	An operating condition whereby outputs are prevented from being controlled by a master. The outputs can be operated locally at the device where the output point is physically located.
Master	A process that desires to obtain data or information in an outstation or that wants to change variables or to control outputs in an outstation. May also refer to a device that contains a master process.
Null Response	A response message wherein the application layer fragment consists of only Application Control, Function Code and Internal Indications octets.
Octet	A group of 8 contiguous digital information bits.
Output	Refers to values in an outstation or lower level device that are controlled by commands from the master. Examples are an analog signal that sets the desired pressure for a gas manifold and electrical contacts which when activated cause a circuit breaker to trip or close. Output sometimes refers to the physical device that receives a control signal such as a circuit breaker.
Outstation	A process that has data, variables or information that another process wants to obtain or wants to set to a new value. May also refer to a device that contains an outstation process.
Point	An instance of a point type.

Point index	The zero-based numeric identifier that differentiates unique instances of points having the same point type within a DNP3 device.
Point type	The classification for entities having a common set of characteristics and attributes. Examples are binary inputs, analog inputs, counters, binary outputs and analog outputs.
Poll	A poll is a request for data from a master.
Polling	Polling is an interrogate-reply scheme whereby a master schedules the transmission of requests for data to an outstation. Upon receipt of the request, the outstation returns the requested data in a response. The scheduled time of each poll and the specific data requested are a local matter.
Primary Station (when used in context of the Data Link Layer)	The device (master or outstation) that initiates a message transaction between its Data Link Layer and that of a secondary device. The secondary, or non-initiating station, sometimes, but not always, depending upon which function code is used by the primary, sends a response to complete the transaction.
Private	Belonging to or restricted to an individual device or system and not generally known to other devices, systems, vendors or persons. An example of a private application is a control loop implemented within a utility's outstation.
Remote Mode	An operating condition whereby outputs may be controlled from a remotely located master. The outputs may also be operated locally if the system permits this.
Report-by-Exception	A schema whereby changes only are reported from an outstation. The data that remains constant is reported at infrequent intervals, via an integrity poll, as a means of assuring that the data in the master matches the data in the outstation. Report-by-exception is used for both polled and unsolicited responses.
Request	An Application Layer message that asks an outstation to perform a specific action. A poll is only one type of request. There are other types of requests; e.g., actuate a control output and set the time.
Response	An Application Layer message from an outstation that is returned to the master as the result of a request from the master.
Secondary Station (when used in context of the Data Link Layer)	The device (master or outstation) that receives a request from a primary station.
Segment	A packet of octets that is sized to fit into a Link Layer frame. Each segment contains a transport header and a portion of a fragment from the Application Layer.
Subset	
Unsolicited Response	An Application Layer message from an outstation to a master for which no explicit request was received. The request is implied by the act of a master enabling the unsolicited operating mode in an outstation.

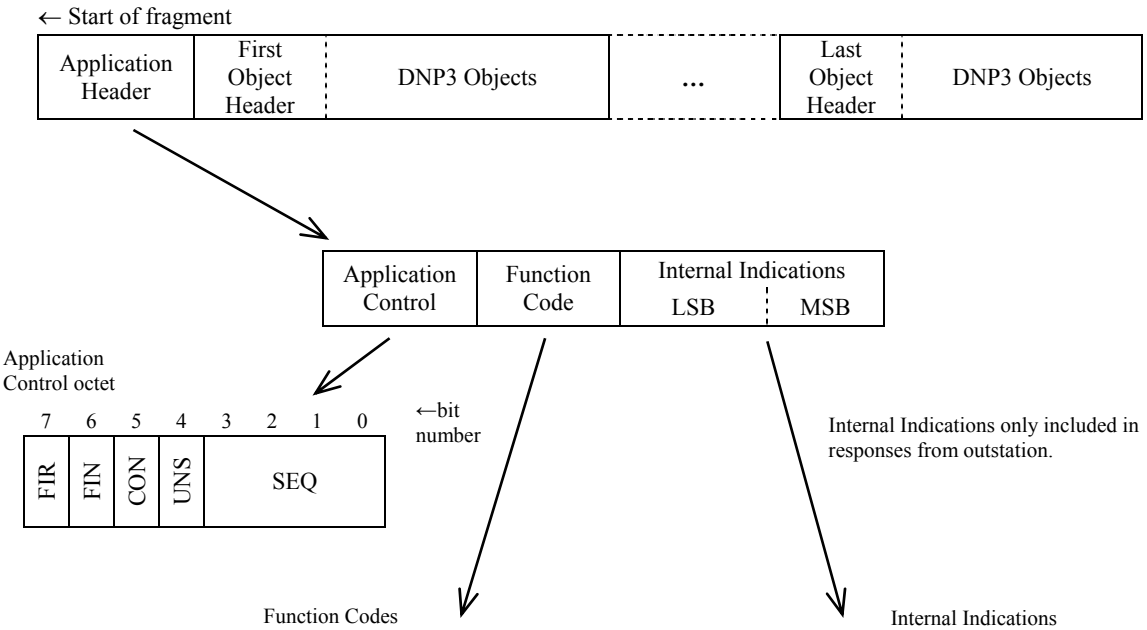
5.2 Acronyms and Abbreviations

The following provides a brief description of acronyms and abbreviations used in the DNP3 specification.

CON	A bit in the Application Layer's control octet that specifies whether an Application Layer confirmation is required.
CRC	Cyclic Redundancy Check code that is generated according to a specific algorithm, and transmitted with the message, for the purpose of detecting data corruption during communication via the Physical Layer.
CROB	Control Relay Output Block. A structured data block appearing in request and response messages associated with actuating on-off type output devices.
DNP	Distributed Network Protocol.
DNP3	The third generation of DNP.
FIN	Final Data Link frame or final Application Layer fragment in a message.
FIR	First Data Link frame or first Application Layer fragment in a message.
IIN	Internal Indications. This bit field appears in response headers that indicate certain states or error conditions with the outstation.
LSB	Least Signifcant Byte. DNP3 uses the term octet instead of byte, therefore this abbreviation means the least significant octet. It is applied when there are two or more contiguous octets that together are used to hold a value and the lower order octet is intended.
MSB	Most Signifcant Byte. DNP3 uses the term octet instead of byte, therefore this abbreviation means the most significant octet. It is applied when there are two or more contiguous octets that together are used to hold a value and the higher order octet is intended.
RBE	Report-by-exception. An methodology whereby an outstation transmits only changes instead of an entire set of data.
SEQ	Sequence number that differentiates subsequent Data Link frames or Application Layer fragments. Sequence numbers associated with unsolicited responses are distinct from sequence numbers used for solicited responses.
URBE	Unsolicited report-by-exception.
UNS	A bit in the Application Layer's control octet that specifies whether a fragment (response and confirmation) pertains to an unsolicited message. When this bit is set, the sequence number in the SEQ field refers to the unsolicited sequence number.

6 DNP3 QUICK REFERENCE

Application Layer



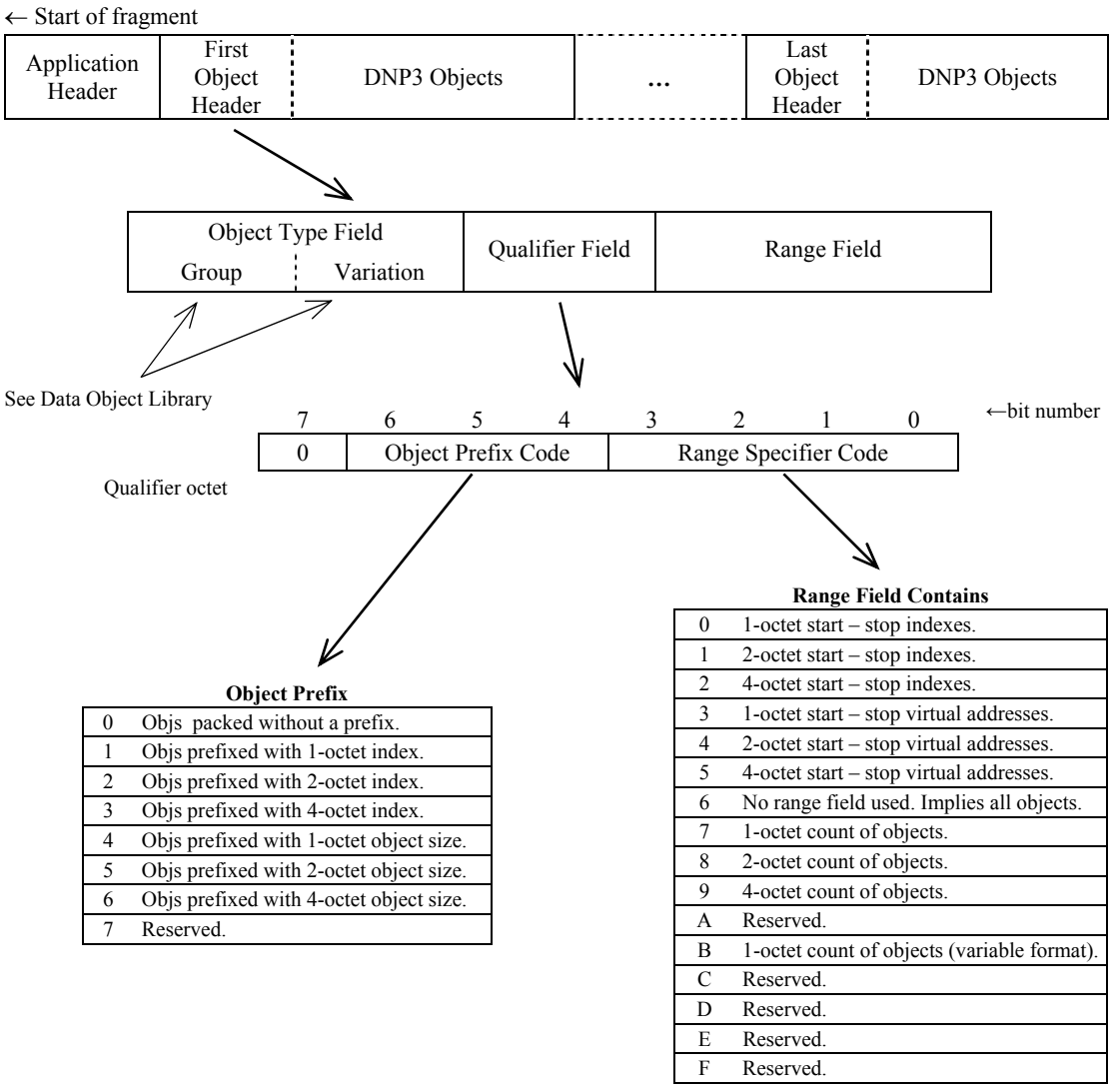
Function Codes

Requests (Hex)			
0	Confirm	10	Initialize application
1	Read	11	Start application
2	Write	12	Stop application
3	Select	13	Save configuration
4	Operate	14	Enable unsolicited
5	Dir operate	15	Disable unsolicited
6	Dir operate – No resp	16	Assign class
7	Freeze	17	Delay measurement
8	Freeze – No resp	18	Record current time
9	Freeze clear	19	Open file
A	Freeze clear – No resp	1A	Close file
B	Freeze at time	1B	Delete file
C	Freeze at time – No resp	1C	Get file information
D	Cold restart	1D	Authenticate file
E	Warm restart	1E	Abort file
F	Initialize data		
Responses (Hex)			
81	Response		
82	Unsolicited response		

Internal Indications

LSB	
IIN1.0	All stations
IIN1.1	Class 1 events
IIN1.2	Class 2 events
IIN1.3	Class 3 events
IIN1.4	Need time
IIN1.5	Local control
IIN1.6	Device trouble
IIN1.7	Device restart
MSB	
IIN2.0	Function code not supported
IIN2.1	Object unknown
IIN2.2	Parameter error
IIN2.3	Event buffer overflow
IIN2.4	Already executing
IIN2.5	Configuration corrupt
IIN2.6	Reserved 1
IIN2.7	Reserved 2

Application Layer



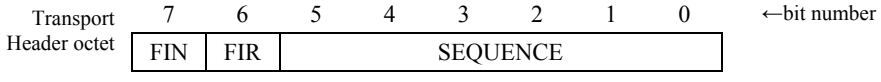
Object Prefix

0	Objs packed without a prefix.
1	Objs prefixed with 1-octet index.
2	Objs prefixed with 2-octet index.
3	Objs prefixed with 4-octet index.
4	Objs prefixed with 1-octet object size.
5	Objs prefixed with 2-octet object size.
6	Objs prefixed with 4-octet object size.
7	Reserved.

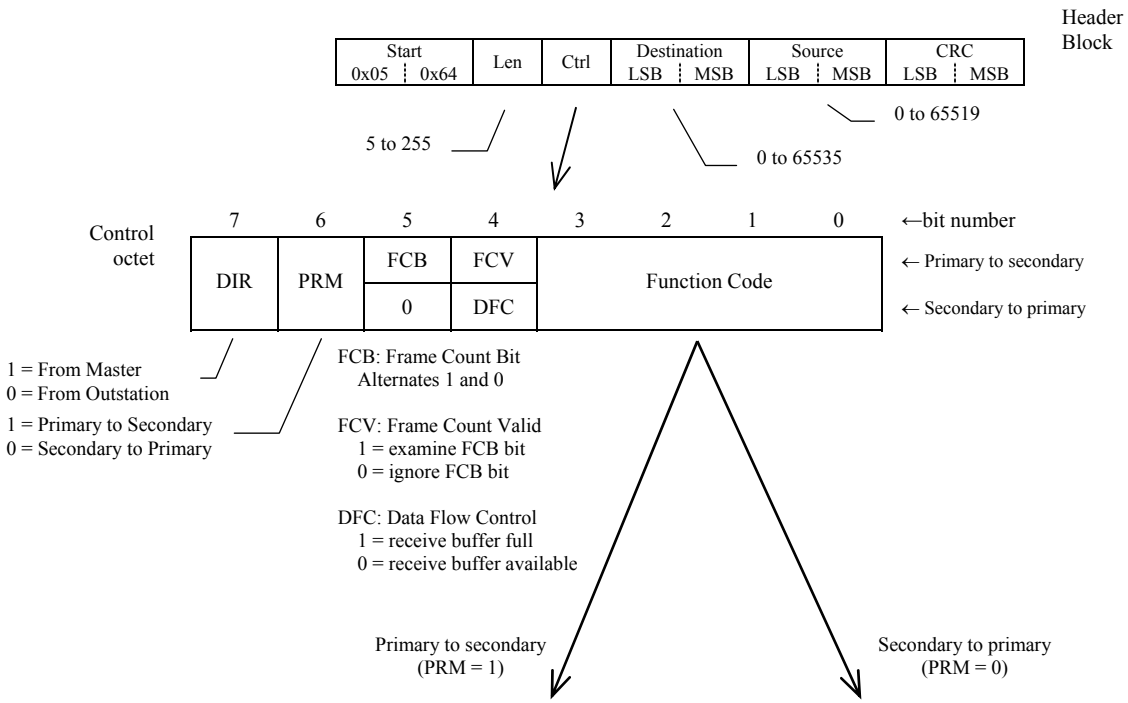
Range Field Contains

0	1-octet start – stop indexes.
1	2-octet start – stop indexes.
2	4-octet start – stop indexes.
3	1-octet start – stop virtual addresses.
4	2-octet start – stop virtual addresses.
5	4-octet start – stop virtual addresses.
6	No range field used. Implies all objects.
7	1-octet count of objects.
8	2-octet count of objects.
9	4-octet count of objects.
A	Reserved.
B	1-octet count of objects (variable format).
C	Reserved.
D	Reserved.
E	Reserved.
F	Reserved.

Transport Function



Data Link Layer



Primary Function Code	Function Code Name	FCV Bit
0	RESET_LINK_STATES	0
1	–	–
2	TEST_LINK_STATES	1
3	CONFIRMED_USER_DATA	1
4	UNCONFIRMED_USER_DATA	0
5	–	–
6	–	–
7	–	–
8	–	–
9	REQUEST_LINK_STATUS	0
A	–	–
B	–	–
C	–	–
D	–	–
E	–	–
F	–	–

Secondary Function Code	Function Code Name
0	ACK
1	NACK
2	–
3	–
4	–
5	–
6	–
7	–
8	–
9	–
A	–
B	LINK_STATUS
C	–
D	–
E	–
F	NOT_SUPPORTED

Valid Data Link Layer Control Codes

Outstation to Master	Master to Outstation	Function Code Name	Type	Comment
00	**80**	**ACK**	Sec-to-Pri	
01	**81**	**NACK**		Link reset required
0B	8B	LINK_STATUS		
0F	8F	NOT_SUPPORTED		
10	90	ACK		Receive buffers full
11	91	NACK		Receive buffers full
1B	9B	LINK_STATUS		Receive buffers full
1F	9F	NOT_SUPPORTED		Receive buffers full
40	**C0**	**RESET_LINK_STATES**	Pri-to-Sec	FCB = 0 (secondary ignores FCB)
44	**C4**	**UNCONFIRMED_USER_DATA**		FCB = 0 (secondary ignores FCB)
49	C9	REQUEST_LINK_STATUS		FCB = 0 (secondary ignores FCB)
52	D2	TEST_LINK_STATES		FCB = 0
53	**D3**	**CONFIRMED_USER_DATA**		FCB = 0
60	E0	RESET_LINK_STATES		FCB = 1 (secondary ignores FCB)
64	E4	UNCONFIRMED_USER_DATA		FCB = 1 (secondary ignores FCB)
69	E9	REQUEST_LINK_STATUS		FCB = 1 (secondary ignores FCB)
72	F2	TEST_LINK_STATES		FCB = 1
73	**F3**	**CONFIRMED_USER_DATA**		FCB = 1

Most commonly used are shown in **bold** face.

DNP3 Exchange Samples

Reset Link Example	
--► 05 64 05 C0 01 00 00 04 E9 21	Reset link states
◄-- 05 64 05 00 00 04 01 00 19 A6	Ack

Integrity Poll Example	
--► 05 64 14 F3 01 00 00 04 0A 3B C0 C3 01 3C 02 06 3C 03 06 3C 04 06 3C 01 06 9A 12	Request class 1, 2, 3 and 0 data
◄-- 05 64 05 00 00 04 01 00 19 A6	Link layer confirm
◄-- 05 64 05 40 00 04 01 00 A3 96	Reset link states
--► 05 64 05 80 01 00 00 04 53 11	Ack
◄-- 05 64 53 73 00 04 01 00 03 FC C1 E3 81 96 00 02 01 28 01 00 00 00 01 02 01 28 05 24 01 00 01 00 01 02 01 28 01 00 02 00 01 02 01 28 B4 77 01 00 03 00 01 20 02 28 01 00 00 00 01 00 00 20 A5 25 02 28 01 00 01 00 01 00 00 01 01 01 00 00 03 00 2F AC 00 1E 02 01 00 00 01 00 01 00 00 01 00 00 16 ED	Response. IIN = device restart, need time, class 1 & 2 events. 4 binary input events, 2 analog input events, 4 binary inputs and 2 analog inputs.
--► 05 64 05 80 01 00 00 04 53 11	Link layer confirm
--► 05 64 08 C4 01 00 00 04 A4 CF C1 C3 00 20 3F	Application layer confirm

Reset Restart IIN Bit	
--► 05 64 0E C4 01 00 00 04 7D A4 C0 C4 02 50 01 00 07 07 00 64 11	Request write IIN1.7 = 0
◄-- 05 64 0A 44 00 04 01 00 59 5E C2 C4 81 10 00 93 AD	Null response

Set Time and Date	
--► 05 64 12 C4 01 00 00 04 0E 0B C0 C5 02 32 01 07 01 F8 B8 6C AA F0 00 98 98	Request write time and date
◄-- 05 64 0A 44 00 04 01 00 59 5E C3 C5 81 00 00 55 93	Null response

Key: --► Master station transmissions (Address 1024 decimal).
◄-- Outstation transmissions (Address 1).