DNP3: SECURITY AND SCALABILITY ANALYSIS

Mandar Patwardhan
B.E, PVPIT, Sangli, 2006

PROJECT

Submitted in partial satisfaction of the requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE

at

CALIFORNIA STATE UNIVERSITY, SACRAMENTO

SPRING
2012
DNP3: SECURITY AND SCALABILITY ANALYSIS

A Project

by

Mandar Patwardhan

Approved by:

Isaac Ghansah, Ph.D., Committee Chair

Scott Gordon, Ph.D., Second Reader

Date
Student: Mandar Patwardhan

I certify that this student has met the requirements for format contained in the University format manual, and that this project is suitable for shelving in the Library and credit is to be awarded for the Project.

__________________________, Graduate Coordinator
Dr.Nikrouz Faroughi

Date

Department of Computer Science
Abstract

of

DNP3: SECURITY AND SCALABILITY ANALYSIS

by

Mandar Patwardhan

DNP3 ('Distributed Network Protocol') is mainly used in electrical sectors as a mechanism to enable communication between different data acquisition and control equipment. In a typical electrical company there is a main control station that manages and supervises many of its substations which are distributed over various geographic locations. DNP3 provides the rules for remotely located computers and master station computers to communicate data. DNP3 is a reliable protocol in the way it works. But it was not designed to be secure from attackers and hackers that can target the underlying SCADA (Supervisory Control and Administration) system.

This report provides a detail analysis of two known methodologies by which security of SCADA communications can be improved as well as comparison of the two methodologies.

(1) Solutions that wrap the DNP3 protocols without making changes to the protocols

(2) Solutions that alter the DNP3 protocols fundamentally
This report also discusses an approach to scale DNP3 over TCP-UDP/IP protocol suite to utilize the power of Internet. This approach is covered in the IEEE standard for Electric Power Systems Communications 1815.

_______________________, Graduate Coordinator
Dr.Nikrouz Faroughi.

_______________________
Date
ACKNOWLEDGEMENTS

I would like to thank my project advisor, Dr. Isaac Ghansah for his guidance and constant support throughout the project. His suggestions have been of great value for the project.

Words alone cannot express the gratitude, I owe to my loving parents for their constant support throughout my life.
TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acknowledgements</td>
<td>vi</td>
</tr>
<tr>
<td>List of Tables</td>
<td>ix</td>
</tr>
<tr>
<td>List of Figures</td>
<td>x</td>
</tr>
<tr>
<td>1. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Background of DNP3</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Why is DNP3 ideal for SCADA communication?</td>
<td>3</td>
</tr>
<tr>
<td>1.3 Security issues in DNP3</td>
<td>4</td>
</tr>
<tr>
<td>1.4 Approaches to improve security</td>
<td>5</td>
</tr>
<tr>
<td>1.5 Approaches to improve scalability</td>
<td>6</td>
</tr>
<tr>
<td>1.6 Report organization</td>
<td>7</td>
</tr>
<tr>
<td>2. DNP3 ANALYSIS IN DETAIL</td>
<td>8</td>
</tr>
<tr>
<td>2.1 DNP3 origins</td>
<td>8</td>
</tr>
<tr>
<td>2.2 Relationship between master and outstation</td>
<td>9</td>
</tr>
<tr>
<td>2.3 DNP3 format structure</td>
<td>10</td>
</tr>
<tr>
<td>2.4 DNP3 message breakdown</td>
<td>11</td>
</tr>
<tr>
<td>2.5 DNP3 application layer</td>
<td>12</td>
</tr>
<tr>
<td>2.6 Responsibility of DNP3 application layer</td>
<td>22</td>
</tr>
<tr>
<td>2.7 DNP3 transport layer</td>
<td>25</td>
</tr>
<tr>
<td>2.8 Responsibility of DNP3 transport layer</td>
<td>27</td>
</tr>
<tr>
<td>2.9 The data link layer</td>
<td>27</td>
</tr>
<tr>
<td>2.10 Introduction to points, indexes, groups and variations</td>
<td>37</td>
</tr>
<tr>
<td></td>
<td>Title</td>
</tr>
<tr>
<td>---</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>1</td>
<td>Important function codes</td>
</tr>
<tr>
<td>2</td>
<td>Object prefix codes</td>
</tr>
<tr>
<td>3</td>
<td>Range specifier codes</td>
</tr>
<tr>
<td>4</td>
<td>Primary to secondary function codes</td>
</tr>
<tr>
<td>5</td>
<td>Secondary to Primary function codes</td>
</tr>
<tr>
<td>6</td>
<td>Function code table</td>
</tr>
<tr>
<td>7</td>
<td>Managing keys</td>
</tr>
<tr>
<td>8</td>
<td>States in state machine descriptions</td>
</tr>
<tr>
<td>9</td>
<td>DNP3 master messages with authentication codes</td>
</tr>
<tr>
<td>10</td>
<td>DNP3 outstation messages with authentication codes</td>
</tr>
<tr>
<td>11</td>
<td>Comparison of the two security approaches</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

1. Figure 1 SCADA communication ................................................................. 1
2. Figure 2 Master/Slave communication .......................................................... 3
3. Figure 3 Master outstation relationship showing what is exchanged .............. 9
4. Figure 4 DNP3 protocol stack ....................................................................... 11
5. Figure 5 DNP3 message breakdown ............................................................... 12
6. Figure 6 Application layer fragment structure ............................................. 14
7. Figure 7 Application layer request header .................................................... 14
8. Figure 8 Application layer response header .................................................. 14
9. Figure 9 Application control octet ................................................................. 15
10. Figure 10 Internal indications in application layer header ......................... 17
11. Figure 11 Object header ............................................................................. 18
12. Figure 12 Qualifier octet in an object header ............................................. 18
13. Figure 13 Object prefix in an object header ............................................... 19
14. Figure 14 Index list .................................................................................. 20
15. Figure 15a Application layer put together-part 1 ....................................... 23
16. Figure 15b Application layer put together-part 2 ....................................... 24
17. Figure 16 Transport function ................................................................... 25
18. Figure 17 Transport segment .................................................................... 26
19. Figure 18 Transport header fields ............................................................. 26
20. Figure 19: Data link layer ......................................................................... 28
21. Figure 20 Transaction diagram for DNP3 data link layer .......................... 29
22. Figure 21 DNP3 data link layer frame format ........................................... 31
46. Figure 46 DNP3 on internet protocol stack...............................................................76
47. Figure 47 UDP cases................................................................................................81
Chapter 1

INTRODUCTION

1.1 Background of DNP3

In computer networking, a protocol defines the rules by which devices talk with each other and DNP3 (Distributed Networking Protocol) is a protocol for transmission of data between two points in a network. It is heavily used in electric and water companies, but it also functions well for other areas. [1]

DNP3 was developed for communications among various types of data acquisition and control environments. It is significant in SCADA (Supervisory Control And Data Acquisition) systems where it is used as a prime communication protocol between various SCADA Master Stations, RTU (Remote Terminal Units) and IED (Intelligent Electronic Devices). [1]

A typical SCADA station can be seen in the Figure 1.

![Figure 1: SCADA communication](image)
SCADA master basics:

It is the control center from which multiple substations or other remote installations are controlled or monitored. It interfaces with humans through HMI (Human Machine Interface) which may be local or remote. A master is connected to RTU(s) or IED(s). [1]

RTU basics:

It exposes as a slave to Master when DNP3 is used for communication. In turn it actually manages multiple IEDs.

IED basics:

It can be a data acquisition device or it may be responsible only for control. Possible input to IED may be configuration setting or command data while possible output can be values, conditions or status results. [1]

The Master can address individual IED(s) or RTU(s), or can initiate a broadcast message to all of them. Slave in return responds to requests that are addressed to them individually. The DNP3 protocol establishes the format for the Master’s request by placing it into slave address, a function code defining the requested action, any data that is to be sent and an error checking field. The slave’s response is also constructed using DNP3 protocol. It contains the fields confirming the action taken, any data to be returned and an error checking field. If an error occurs in receipt of the message or if slave is unable to perform the action an error message is constructed and sent as a response. Details can be seen in figure 2. [4]
The request message is contained in the application layer information within the message. A confirmation response is required to this message. The Slave station (RTU) sends an ACK (acknowledge) message to the Master. Since the last transaction contained an application level request for the transmission of data, the Slave station then performs the action requested and initiates a communication with the requested data. [6]

1.2 Why is DNP3 ideal for SCADA communication?

DNP3 is a non-proprietary protocol that is available to anyone by visiting the web site www.dnp.org. Only a nominal fee is charged for documentation, but otherwise it is available worldwide with no restrictions. This means a party can purchase master station and outstation computing equipment from any manufacturer and be assured that they will reliably talk to each other. Vendors compete based upon their computing equipment’s features, costs and quality factors instead of who has the best protocol. Utilities are not bound to one manufacturer after the initial sale.
DNP3 optimizes the transmission of data acquisition and control commands in SCADA systems. It differs from general purpose protocols for transferring email and hypertexting documents or the protocols used in SQL queries or used in multimedia broadcast and huge files.

Complex SCADA networks require data transfer protocols to meet all of the communication challenges while remaining easy to configure. The components shown in Figure 1 typically operate in harsh environment. High integrity SCADA systems require very short data latency, and accurate, high-resolution time tagging of reported data. [4] That’s why the architecture requires protocol that provides reliability and efficiency.

SCADA benefits from following features of DNP3. [2]

1. Request and response with multiple data types in single messages,
2. Segments messages into multiple frames to ensure error detection and recovery
3. Assigns priorities to data items and request data items based on their priority
4. Ability to respond without request (unsolicited)
5. Supports time synchronization and a standard time format
6. Allows multiple masters and peer-to-peer operations.
7. Allows user definable objects including file transfer.

1.3 Security issues in DNP3

How secure are SCADA systems

Many SCADA based systems in today’s operation were deployed decades ago with availability and personal safety as the primary concerns rather than security. After initial dependence on proprietary elements, it is now common to build SCADA systems using commercial hardware and software that speaks to open communication protocols, the technical internals of which are
often easily accessible. Networks of many vendors are connected to Internet. With the integration of Internet into the SCADA networks the vendors are now interconnected to achieve efficient information but vulnerable to malicious intent of attackers. Finally, teams of sophisticated hackers are now employed by criminal organizations or terrorists to break into these systems. [5]

Failure of critical infrastructure can produce devastating results. As an example of cyber attack in 2001 in Australia, a computerized sewage management system was hacked and dumped untreated sewage water into parks and rivers. [26] It is therefore very important that SCADA systems and the protocol running on it must be secured with the cyber security by identifying the possible threats that can occur in all the forms.

1.4 Approaches to improve security

There are some approaches that are widely under consideration when it comes to improve the security in DNP3. They can be classified as below: [6]

1. Solutions that wrap the DNP3 protocol without making changes to the protocols.

2. Solutions that alter the DNP3 protocol fundamentally.

The main goal is to address the threats related to confidentiality, integrity and authenticity in the DNP3 as a part of SCADA architecture, with a minimum performance impact on the communication link and without affecting the scalability of the SCADA systems and components. It should also be kept in mind that the expensive Masters Station and Substation devices as well as the applications supporting them shouldn’t undergo extensive modifications.
“The solutions that wrap DNP3 protocols include SSL/TLS and IPsec. They provide a quick and low cost security enhancement. The solutions that would require altering the DNP3 protocol tend to be more time consuming to implement and expensive but provide better end to end security. Such solutions can be deployed at either a protocol level or within an application.” [8]

The solutions that change the protocol fundamentally target to provide security at application layer. Application layer security adds user and device authentication as well as data integrity protection to the DNP3 protocol. Secure DNP3 is a bi-directional protocol that provides protection between master stations and outstations (PLC’s, RTU’s, and IED’s). [25]

1.5 Approaches to improve scalability

With the advent of Internet DNP3 is not limited to the particular LAN of the station in which it controls the communication. Chapter 4 explains an approach by which DNP3 communication can be carried out over TCP-UDP/IP networks. [21]

It employs the Internet Protocol suite for DNP3 message transport and also does not redefine any of the existing DNP3 layers and applies to all end devices that can interpret and generate DNP3 messages on a local (LAN) or wide area network (WAN).

The reasons for choosing the Internet Protocol suite as a transport mechanism for DNP3 are as follows:

1. Highly efficient integration of networks

2. Leverage existing Master/Outstation equipment and DNP3 standards
1.6 Report organization

This report is organized as follows

Chapter 1: Introduction
This chapter gives an introduction to the concepts of DNP3 and its use in SCADA communication. It also discusses the security shortcomings of the protocol and approaches to improve it.

Chapter 2: DNP3 detail analysis
In this chapter DNP3 is explored in greater details. Factors like its place in protocol stack, message format and its features and message exchange is deeply understood.

Chapter 3: Approaches to improve security
In this chapter the security shortcomings are put forward and some of the approaches by which security can be implanted in DNP3 (internal or external) are researched.

Chapter 4: Approaches to improve scalability
This chapter brings up some approaches to improve the efficiency of DNP3 by taking Internet protocol suite into consideration.

Chapter 5: Conclusion
This chapter comprises of summary of the project report, the conclusion arrived.

Glossary

References
2.1 DNP3 origins

DNP3 originally was developed by a company called Westronic, Inc. (now called GE Harris) in late 1990s. During 1993 “the DNP3 Basic 4.0” protocol specification documentation set was released to public. The ownership of the protocol was given to DNP Users group in October, 1993. Since then the protocol has gained high importance and is acceptable throughout the world.

In January 1995, a better and dedicated committee called DNP Users Group Technical Committee was formed to improve the features put forwarded by the DNP3 users group. The committee came up with a document called the ‘DNP subset Definitions’ document, which is followed as a standard for implementation of DNP3. [8]

In 1994, the IEEE Power Engineering Society’s Data Acquisition, Monitoring and Control Subcommittee formed a Task Force to review the communication protocols being used between Intelligent Electronic Devices (IEDs) and Remote Terminal Units (RTUs) in substations. The IEEE Task Force found the atmosphere in which SCADA systems work to be very confusing and changing constantly. The Task Force collected statistics from about 140 protocols and compared it with the requirements of SCADA systems. Two protocols were short listed as the most efficient and mostly matching all the requirements set by SCADA systems. DNP3 got additional support than the other selected protocol IEC 60870-101, which was more favorite among European vendors. The comparison resulted into a new Standard called IEEE Standard 1379-1997 (Trial Use Recommended Standard for Data Communication between Intelligent Electronic Devices
and Remote Terminal Units in Substations). This study was well received and got acceptance. The standard was updated in 1999 with a new ballot and became IEEE Standard 1379-200 “Recommended Practice for Data Communications between Remote Terminal Units and Intelligent Electronic Devices in Substations.” [8]

2.2 Relationship between DNP3 master and outstation

Main role of DNP3 is that it optimizes the flow of data acquisition and control commands in SCADA systems. It significantly is different from general purpose protocols used in mail servers or querying databases.

Figure 3: Master outstation relationship showing what is exchanged [6]
Figure 3 shows the relationship between Master and outstation (can be an RTU). Master and outstation have their own databases. This figure gives us an idea of how the Master and Outstation exchange the information using DNP3 as a communication protocol through the processes running on corresponding Master and Outstation systems. The Master is shown on the left while the outstation is represented on the right.

As shown in the figure 3, the square blocks at near the outstation depict the data stored in the outstation database. Different data types are organized as arrays. Each array represents the states which the physical devices can fall into, which can be binary. Analog input arrays are the input quantities measured or computed. Similarly, following the same pattern, the counter array holds values in kilowatt hours. Control inputs are organized as into an array representing physical or logical on or off. Elements of the array are labeled 0 to N-1 where N is the number of blocks shown for respective data types. [6]

The right hand side of the figure 3 represents DNP3 master which has a similar database as of DNP3 outstation. The values stored in Master database are used for representing mostly the control information like system states, closed loop control, alarm notification, billing and other purposes. Master keeps its database updated by sending requests to outstations, who return the value stored in their own database. This is done thorough polling. [6]

2.3 DNP3 format structure

As shown in Figure 4, DNP3 stack has 3 three layers: The application layer, the data link layer and the physical layer. To support the RTU functions and messages larger than the maximum frame length, a transport pseudo layer is added. [8]
2.4 DNP3 message breakdown

In the next section we will take a detail look into the DNP3 message structure in a layered approach. Figure 5 shows a complete breakdown of a DNP3 message.
2.5 DNP3 application layer

The Application Layer stands on the top of the EPA (Enhanced Protocol Architecture) and OSI (Open System Interconnect) models. It interfaces with the DNP3 user’s software and with the lower layers. The Application Layer mainly performs standardized functions, data formats, and procedures for exchanging data acquisition values, attributes, and control commands. DNP3 user’s software makes use of the services offered by DNP3 Application Layer to send and receive messages with peer DNP3 device.
2.5.1 Application layer message structure

Masters forms a message and transmits it to an outstation to perform a command or an activity. When the outstation receives this message, it initiates its own action and generates a message and transmits it back to master. The information sent by the outstation can be a data result as explained in previous section. In some cases, an outstation may spontaneously transmit a message which isn’t asked or queried by master. This is called as ‘unsolicited reporting’ by outstations.

2.5.1.1 Application layer fragments

In networking terms a fragment “is a block of octets containing request or response information transported between peers (here, a master and an outstation.) Each DNP3 application layer fragment contains a function code (as described in later sections) specifying how the receiver shall process the fragment. DNP3 limits the amount of memory reserved for sending and receiving messages (messaging buffers) by specifying the maximum length of each fragment and also allowing response messages into a single or multi fragments. Small messages, requiring only a few octets, can fit into a single fragment, whereas larger messages may require multiple fragments. Fragments when transferred into higher or lower layers are broken or reassembled depending on if they are passed into higher or lower layers.

2.5.1.2 Application layer fragment structure

As shown in Figure 6 each application fragment begins with an application header that contains message control information. The header also contains a field called internal indications.
The application header is unable to convey entire information required for message exchange and is often accompanied with another entity called DNP3 objects. These two together form a DNP3 message.

2.5.1.3 Application request header

An application request header is used in requests from masters and has two fields in Figure 7.

```
<table>
<thead>
<tr>
<th>← Application Request Header          →</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Control</td>
</tr>
<tr>
<td>(1 octet)</td>
</tr>
</tbody>
</table>
```

Figure 7: Application layer request header [25]

2.5.1.4 Application response header

Application response header has three fields as shown in Figure 8.

```
<table>
<thead>
<tr>
<th>← Application Response Header        →</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Control</td>
</tr>
<tr>
<td>(1 octet)</td>
</tr>
</tbody>
</table>
```

Figure 8: Application layer response header [25]
2.5.1.5 Application control octet – Common to both request and response header (Figure 9)

<table>
<thead>
<tr>
<th>Bit #</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fields</td>
<td>FIR</td>
<td>FIN</td>
<td>CON</td>
<td>UNS</td>
<td>SEQ</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 9: Application control octet [25]

FIR- Indicates if this is the first fragment of the message

FIR = 0 - Not the first fragment
FIR = 1 - Is the first fragment

FIN- Indicates if this is the last fragment of a message

FIN = 0 - More fragments to follow
FIN = 1 - Final fragments of the message

CON- Indicates if receiver’s application layer would return an application confirmation message

CON = 0 Application layer confirmation message shall not be returned
CON = 1 Application layer confirmation message shall be returned

UNS- Indicates if the message contains unsolicited response or a confirmation of an unsolicited response

UNS = 0 indicates that the sequence number is associated with a master request
UNS = 1 indicates that sequence number is associated with an unsolicited response message.

The UNS bits are set by an outstation when it contains an unsolicited message. Masters set this bit in Application Layer confirmation fragments that confirm the receipt of an unsolicited response.
SEQ- 4 bit wide. SEQ field verifies if the fragments arrived are in order or if it has received a duplicate fragment. It has a range from 0 to 15. The number increments by one count, modulo 16 for non-repeated Master fragments, next fragments and unsolicited response fragments when the UNS field is set by outstation. [25]

2.5.1.6 Function code octet – Common to request and response header

Function code octet identifies the purpose of the message. Request messages from masters use the function codes from 0 till 128. Response messages from outstations use function codes with values from 129 to 255. [11]

<table>
<thead>
<tr>
<th>Message Type</th>
<th>Code</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Confirmation</td>
<td>0x00</td>
<td>CONFIRM</td>
<td>Confirm Function Code: Master sends this to an Outstation to confirm the receipt of a fragment.</td>
</tr>
<tr>
<td>Request</td>
<td>0x01</td>
<td>READ</td>
<td>Outstation shall return the data specified by the objects in the request.</td>
</tr>
<tr>
<td>Request</td>
<td>0x02</td>
<td>WRITE</td>
<td>Outstation shall store the data specified by the objects in the request.</td>
</tr>
<tr>
<td>Request</td>
<td>0x03</td>
<td>SELECT</td>
<td>Outstation shall select the output points specified by the objects in the request preparation. For an operate command</td>
</tr>
<tr>
<td>Request</td>
<td>0x04</td>
<td>OPERATE</td>
<td>Outstation shall activate the output points selected by the previous select function code.</td>
</tr>
<tr>
<td>Request</td>
<td>0x0D</td>
<td>COLD_RESTART</td>
<td>Complete reset of all hardware and software in the device.</td>
</tr>
<tr>
<td>Request</td>
<td>0xF</td>
<td>INITIALIZE_DATA</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-----</td>
<td>----------------</td>
<td></td>
</tr>
<tr>
<td>Request</td>
<td>0x11</td>
<td>START_APPL</td>
<td></td>
</tr>
<tr>
<td>Request</td>
<td>0x14</td>
<td>ENABLE_UNSOLICITED</td>
<td></td>
</tr>
<tr>
<td>Request</td>
<td>0x17</td>
<td>DELAY_MEASURE</td>
<td></td>
</tr>
<tr>
<td>Response</td>
<td>0x81</td>
<td>RESPONSE</td>
<td></td>
</tr>
<tr>
<td>Response</td>
<td>0x82</td>
<td>UNSOLICITED_RESPONSE</td>
<td></td>
</tr>
<tr>
<td>Response</td>
<td>0x83</td>
<td>AUTHENTICATE_RESPONSE</td>
<td></td>
</tr>
</tbody>
</table>

Outstation shall start running the applications specified by the objects in the request.

Enables outstations to initiate unsolicited responses.

Outstation shall report time it takes to process and initiate the transmission of its response.

Solicited response code. Master shall interpret this fragment as Application Layer response to an Application Layer request sent by the Master.

Master shall interpret this fragment as an unsolicited response that was not prompted by an explicit request.

Authentication purposes sent from outstation.

### 2.5.1.7 Internal indications – Present only in response headers

The internal indications field as described earlier field is shown in Figure 10 appears in application response headers immediately following the function code octet. Each octet has 8 bit fields numbered 0 through 7 where bit 0 is the least significant bit.

<table>
<thead>
<tr>
<th>← Internal indications →</th>
</tr>
</thead>
<tbody>
<tr>
<td>First octet</td>
</tr>
</tbody>
</table>

↑ first octet transmitted

Figure 10: Internal indications in application layer header [25]

### 2.5.1.8 Object headers- Figure 6 shows optional set of object headers and DNP3 objects following the application request and response headers.
Consider a scenario in which a master wants to read analog values shown in one of the arrays in Figure 6 from an outstation or a remote terminal unit (RTU) device. The master achieves this by formulating the request message using the function code READ. The object header then specifies the following information. [25]

The analog input point type data is wanted.

The integer or floating-point format the outstation should use when sending its data.

DNP3 objects are not included in the request, only an object header (or headers), because the master is not sending values; it only sends enough information for the outstation to know which values and format are desired. The response message uses function code RESPONSE and contains the same or a similar object header (or headers) followed by the DNP3 objects. [25]

Each object consists of a single analog value from a single point index [9] as shown in Figure 11.

<table>
<thead>
<tr>
<th>←</th>
<th>Object Header</th>
<th>→</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object Type Field</td>
<td>Qualifier Field</td>
<td>Range Field</td>
</tr>
<tr>
<td>Group (1 octet)</td>
<td>Variation (1 octet)</td>
<td>(dependent upon qualifier)</td>
</tr>
</tbody>
</table>

Figure 11: Object header [25]

- **Object group**: Specifies the datatype in request or response
- **Object variation**: Specifies data format of a DNP3 variation. DNP3 variation is described later
- **Qualifier field** as shown in Figure 12

<table>
<thead>
<tr>
<th>Bit # Fields</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Res</td>
<td>Object Prefix Code</td>
<td>Range Specifier Code</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 12: Qualifier octet in an object header [25]
Res- Reserved for future

Object prefix code- Specifies the value that appears before each DNP3 object. It’s either an index number or an object size as shown in Figure 13.

![Figure 13: Object prefix in an object header [25]](image)

It is sometimes necessary to place a prefix before each of the objects to associate the object data with an index in the outstation. [9]

<table>
<thead>
<tr>
<th>Code (Hex)</th>
<th>Description</th>
<th>Size of object prefix</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Objects are packed without an index prefix</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>Objects are prefixed with an index</td>
<td>1-octet</td>
</tr>
<tr>
<td>2</td>
<td>Objects are prefixed with an index</td>
<td>2-octet</td>
</tr>
<tr>
<td>3</td>
<td>Objects are prefixed with an index</td>
<td>4-octet</td>
</tr>
<tr>
<td>4</td>
<td>Objects are prefixed with an object size</td>
<td>1-octet</td>
</tr>
<tr>
<td>5</td>
<td>Objects are prefixed with an object size</td>
<td>2-octet</td>
</tr>
<tr>
<td>6</td>
<td>Objects are prefixed with an object size</td>
<td>4-octet</td>
</tr>
<tr>
<td>7</td>
<td>Reserved for future use.</td>
<td>-</td>
</tr>
</tbody>
</table>

When a master sends the event data it has no information about which points have changed, so it attaches an index number with the object data to reference it as shown in Figure 14.
In some cases, when it reads a list of non-sequential binary input points, the master needs to convey a list of point indexes to the outstation; thus it ends up sending only the index list and no data filled objects. As shown in below figure 14, the master uses object prefix code 1, 2, or 3 and sends “null objects” (having zero octets) prefixed by a point index. [25]

![Figure 14: Index list](image)

Range specifier code- Indicates whether a field is used and if used also the length of each field.

<table>
<thead>
<tr>
<th>Code (Hex)</th>
<th>Description</th>
<th>Number of octets in range field</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Range field contains 1-octet start and stop indexes.</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>Range field contains 2-octet start and stop indexes.</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>Range field contains 4-octet start and stop indexes.</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>Range field contains 1-octet start and stop virtual addresses.</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>Range field contains 2-octet start and stop virtual addresses.</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>Range field contains 4-octet start and stop virtual addresses.</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>No range field is used. This implies all values.</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>Range field contains 1-octet count of objects.</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>Range field contains 2-octet count of objects.</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>Range field contains 4-octet count of objects.</td>
<td>4</td>
</tr>
<tr>
<td>A</td>
<td>Reserved for future use.</td>
<td>—</td>
</tr>
<tr>
<td>B</td>
<td>Variable format qualifier, range field contains 1-octet count of objects.</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>Reserved for future use.</td>
<td>—</td>
</tr>
<tr>
<td>D</td>
<td>Reserved for future use.</td>
<td>—</td>
</tr>
<tr>
<td>E</td>
<td>Reserved for future use.</td>
<td>—</td>
</tr>
<tr>
<td>F</td>
<td>Reserved for future use.</td>
<td>—</td>
</tr>
</tbody>
</table>
Codes 0 to 2
This code range is used when the DNP3 objects are packed contiguously in their index order. The index of the first is shown by the starting index. The index of the next object increases by one. The index of the last object called the stop index. When the start and stop index values are identical, there is only one object present in the range.

Codes 3 to 5
This range is used for specifying contiguous address locations in a vendor-specific virtual memory space. These codes are not widely used, but when a vendor implements them, they are most often used with 8-bit unsigned integer objects. As similar to previous code range address of the first object shall be the start address. The address of each succeeding object shall increase by one than the previous. The address of the last object is the stop address. When the start and stop address values are identical, the range has only one object. [25]
Range specifier code 6
Indicates the master desires values from all of the points specified by the object group. [25]

Codes 7 to 9
Indicate the range field contains a count of data objects or a count of object indexes.
An outstation responds with lesser objects with fewer objects if a master’s request uses qualifier codes 0x07, 0x08, or 0x09 and there is a practical reason for sending a lesser number.
2.6 Responsibility of DNP3 application layer

The application layer describes the message format, attributes, services it offers to the lower layers and the procedures. It listens and builds messages on the need for or availability of user application data. If the data is too large to send it breaks down the message and sends each message sequentially. After building of messages the application layer presents a buffer comprising of an application layer fragment to the pseudo-transport layer for segmentation.

The application layer initially forms the data blocks and they are termed as application service data units (ASDU). The application header as described in 2.14.1 is also called Application Protocol Control Information (APCI). The response header accompanies with an additional 2 byte field called internal indications (IIN). This field is used when the request cannot be processed due to errors. The application layer sequence number allows the receiving application layer to detect fragments that are out of sequence, or dropped fragments. [8] Figure 15a and 15b show a detail breakdown of Application Layer message format.
Figure 15a: Application layer put together-part 1 [30]
Figure 15b: Application layer put together-part 2 [30]
2.7 DNP3 transport layer

The transport layer often named pseudo transport layer in DNP3 terminology sits beneath the application layer and just above the data link layer. All messages to and from the Application layer pass through the transport function on their way and from and to other station as can be seen from Figure 16.

Figure 16: Transport function [25]

The size of a DNP3 application layer message fragment may be larger than the MTU (Maximum Transmission Unit) assigned in a Data Link Frame. The transport layer disintegrates Application layer fragments into Data Link layer sized data units for transmission and reassembles these transport segments to form a transport layer message. [25]

At the transmission site, an Application fragment is broken into smaller portions and an application layer header is attached which contains the sequence number information to keep packets in order. The header and payload in form of application data constitutes a transport
segment that is further passed to data link layer. Transport segments, figure 17, are always passed one at a time.

The receiving application checks for the sequence number of the received messages. Out of order packets are discarded. [25]

<table>
<thead>
<tr>
<th>Header</th>
<th>Application Layer Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>← 1 octet →</td>
<td>← from 1 to 249 octets →</td>
</tr>
</tbody>
</table>

Figure 17: Transport segment [25]

2.7.1 Transport header

Transport header contains a single octet. The header is the first octet in a transport segment. A header is then pre-attached to each Application Layer data before the whole message is submitted to Data Link layer for transmission. When received the, Data Link Layer cuts out the transport header and assembles an application fragments if required.

<table>
<thead>
<tr>
<th>Bit #</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fields</td>
<td>FIN</td>
<td>FIR</td>
<td>SEQUENCE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 18: Transport header fields [25]

FIN- This is a single bit, indicates if this is the final or last transport segment.

FIN = 0 indicates more segments to follow

FIN = 1 indicates the final transport segment in the series of transport segment

FIR- This field is single bit, indicates if this is the first transport segment in the fragment.

FIR = 0 indicates this is not the first segment in a series of transport segment.

FIR = 1 indicates this is the first segment in a series of transport segment.
Sequence number field- This 6 bit field is used to verify that transport segments are received in correct order and to guard against duplicated or missing transport segments. It has a range from 0 to 63. Sequence numbers increment by one, modulo 64, for each segment in a series of transport segments that hold a single Application Layer fragment.

2.8 Responsibility of DNP3 transport layer

The DNP3 transport layer provides segment formats used for message communication. It mainly performs the breakdown of application layer packets so that they are compatible with the size allowed by data link layer segments and reassemble them when they are sent towards application layer. Each broken application layer fragment is called a Transport Protocol Data Unit (TDPU). Maximum size allotted is 250 bytes. 1 byte of that unit is header and 249 bytes of maximum allowed data. As already mentioned, it has a rolling segment sequence number to verify that segments are received in correct order and mark against duplicates or missing ones. [8]

2.9 The data link layer.

Overview- The data link layer depicted in Figure 19 provides an interface between the Transport function and physical media or network transport layer. The main functions of Data Link Layer are station addressing and error detection. This layer adds last DNP3 specific overhead octets when transmitting over the communication channel. [25]

DNP3 protocol has been designed to operate in a byte-oriented mode over communications channel and also in packet-oriented mode when using TCP/IP and UDP/IP. The DNP3 Data Link Layer is made compatible with both connection-less and connection oriented systems.
2.9.1 DNP3 data link layer

This layer plays two purposes in DNP3: It’s a two way transport for applications to ship data between participating devices. It serves by encoding the transport layer segments passed from upper layer and sends them to the physical media for transmission. The transport segments are extracted and passed to upper layers. It also manages the data link frame synchronization and error control.

DNP3 data link layer thus offers

- Encapsulation of transport segments and sending to lower layers
- Decoding of data link frames received from communication channel to upper layers.
- Error detection using a CRC (Cyclic Redundancy Check) for every 16 data octets and one for the data link frame header. These CRC words are checked in each received frame so that only valid data are passed to the Transport Function.
• Source and destination addressing. DNP3 requires both addresses to enable multiple masters and multiple outstations to share the communication channel. [25]

2.9.2 DNP3 data link transaction model

Transactions are used for conveying data and performing link related functions in the Data Link layer. A transaction consists of one or two messages.

a. A request message from a Master to outstation

b. An optional response message from the outstation back to Master[25]

A request message is shown in dark arrow between Masters to Outstation in Figure 20 from primary to secondary. It always contains a command for Outstation with a function code included in the message to specify the command. Some of the function codes are intended for managing the link. The message contents are strictly private and hidden from any other DNP layers.

The response message of an transaction may be a simple acknowledgement of the request message or some additional information about outstations Data link layer. PRM as described in the next section is the primary message bit which indicates who (Master/Outstation) originated the message.

![Figure 20: Transaction diagram for DNP3 data link layer [25]](image-url)
The Data Link Layer carries data from the upper layers which is called user data. The data link layer always transmits data in a master to outstation request message. The Data Link Layer in each DNP3 device acts as both a primary and as a secondary station. The behavior of a secondary is independent of its behavior as a primary. The following example is directly quoted from reference [25]

“For example, when a master station issues a poll, its Application Layer passes information to the Data Link Layer requesting it to initiate a transaction to send the information to the Application Layer in an outstation. The master is the primary, and the outstation is the secondary. When the Application Layer in the outstation has data to return to the master, it passes that information to its Data Link Layer requesting it to initiate a transaction to send the data to the Application Layer in the master. In this instance, the outstation is the primary and the master is the secondary.” [25]

2.9.3 DNP3 data link frame format

A data link frame as shown in Figure 21 has a fixed length header followed by optional data block. Each block ends with a 16 bit CRC.
Data Link Layer header fields: The header consists of 2 start octets, 1 length octet, 1 link control octet, a two octet destination address and a two octet source address.

Start field: 2 octets, First octet is 0X05 and second is 0X64

Length field: This field is 1 octet in length and specifies the count of non-CRC octets that follow header and data blocks. This count includes the CONTROL, DESTINATION and SOURCE fields in the header and USER DATA fields in the body. CRC fields are not included. Minimum value is 5 (only header) while maximum value is 255.

Control Field: This field as shown in Figure 22 is one octet in length and contains relevant information regarding frame’s direction, transaction, error and flow control and function.
• DIR bit field: This field indicates the transmission origin of the data link frame.

  DIR = 1 frame from an Master
  DIR = 0 frame from an outstation

• PRM bit field: The primary message bit field indicates the direction of the data link frame with respect to the initiating station.

  PRM = 1 indicates a Data Link layer transaction is being initiated by either a master or an outstation with FCV and FCB fields applied.
  PRM = 0 indicates a Data Link layer transaction is being completed by either a master or an outstation with DFC field applied.

• FCB bit field: This field is used to message loss and duplicates.

• DFC bit field: The Data Flow Control (DFC) is used to report an inadequate Data Link layer buffer or a busy data link

  DFC = 1 indicates receive buffers were not available or that secondary’s Data Link Layer was busy.
  DFC = 0 indicates receive buffers were available and the secondary’s Data Link Layer was ready.

• Function CODE field: This important field specifies the actual function for each data link frame. Value of this field is dependent on the way in which message is flowing i.e. master to outstation or vice versa.
Table 4: Primary to secondary function codes. [25]

<table>
<thead>
<tr>
<th>Primary function code</th>
<th>Function code name</th>
<th>Service function</th>
<th>FCV bit</th>
<th>Response function codes permitted from secondary</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>RESET_LINK_STATES</td>
<td>Reset of remote link</td>
<td>0</td>
<td>0 or 1</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>Obsolete</td>
<td>—</td>
<td>15 or no response</td>
</tr>
<tr>
<td>2</td>
<td>TEST_LINK_STATES</td>
<td>Test function for link</td>
<td>1</td>
<td>0 or 1 (no response is acceptable if the link states are UnReset)</td>
</tr>
<tr>
<td>3</td>
<td>CONFIRMED_USER_DATA</td>
<td>Deliver application data, confirmation requested</td>
<td>1</td>
<td>0 or 1</td>
</tr>
<tr>
<td>4</td>
<td>UNCONFIRMED_USER_DATA</td>
<td>Deliver application data, confirmation not requested</td>
<td>0</td>
<td>No response</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>Reserved</td>
<td>—</td>
<td>15 or no response</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>Reserved</td>
<td>—</td>
<td>15 or no response</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>Reserved</td>
<td>—</td>
<td>15 or no response</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>Reserved</td>
<td>—</td>
<td>15 or no response</td>
</tr>
<tr>
<td>9</td>
<td>REQUEST_LINK_STATUS</td>
<td>Request status of link</td>
<td>0</td>
<td>11</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>Reserved</td>
<td>—</td>
<td>15 or no response</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>Reserved</td>
<td>—</td>
<td>15 or no response</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>Reserved</td>
<td>—</td>
<td>15 or no response</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>Reserved</td>
<td>—</td>
<td>15 or no response</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td>Reserved</td>
<td>—</td>
<td>15 or no response</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>Reserved</td>
<td>—</td>
<td>15 or no response</td>
</tr>
</tbody>
</table>
Table 5: Secondary to Primary function codes. [25]

<table>
<thead>
<tr>
<th>Secondary function code</th>
<th>Function code name</th>
<th>Service function</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>ACK</td>
<td>Positive acknowledgment</td>
</tr>
<tr>
<td>1</td>
<td>NACK</td>
<td>Negative acknowledgment</td>
</tr>
<tr>
<td>2</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td>3</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td>4</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td>5</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td>6</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td>7</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td>8</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td>9</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td>10</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td>11</td>
<td>LINK_STATUS</td>
<td>Status of the link</td>
</tr>
<tr>
<td>12</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td>13</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td>14</td>
<td>—</td>
<td>Obsolete</td>
</tr>
<tr>
<td>15</td>
<td>NOT_SUPPORTED</td>
<td>Link service not supported</td>
</tr>
</tbody>
</table>

Destination field: 2 Octet field having the destination address.

Source field: 2 octet field having the source address

CRC fields: This 2-octet cyclic redundancy check is appended to each block in a frame. The START, LENGTH, DESTINATION and SOURCE fields are all included when calculating the CRC for header.
The 2-octet CRC check as shown in Figure 24 and 25 is generated from a polynomial and then inverted before placed in the block for transmission.

<table>
<thead>
<tr>
<th>first octet</th>
<th>last octet</th>
<th>CRC LS</th>
<th>MSB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header or user data octets</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 24: CRC ordering in link layer format [25]

![Request Message](image)

Figure 25: CRC state example [25]

Frame error detection:

The receiving device’s Data Link layer shall examine the frames before acceptance. The DNP3 software examines the following conditions and if any of it is false, discards the frame. [25]

1. The starting octets shall be 0X05 followed by 0X64
2. The destination address shall match one of the address that is programmed to accept by Data Link layer
3. The source address shall match one of the address that is programmed to accept by Link layer
4. Correct CRC values shall appear at the designated locations
5. The frame length shall match the value in the length field.
6. Function codes shall comply with those specified for primary to secondary requests table numbered 4 and secondary to primary response table numbered 5.
7. The FCV bit shall be set in primary to secondary frames depending on function code.
Figure 26 shows the data-link layer put together

![Data Link Layer Diagram]

Figure 26: Data link layer put together [30]
2.10 Introduction to points, indexes, groups and variations – DNP3 features

2.10.1 Points and point types

A DNP3 point is a unique object type which may be physical or logical. A point can be an input point or an output point in the form of analog or binary. A single analog input is an example of a point. It is associated with a specific measured signal or computed analog quantity.

As mentioned earlier in Figure 6 point types here are shown in Figure 27 as arrays representing related characteristics, similar functionality. They exhibit the relationship with hardware in form of Master or Outstation database. Using the example from the previous paragraph, an analog input belongs to the analog input point type.

DNP3 represents each point type as an array of points which are referenced by an index like a single dimensional array in programming languages.

![Figure 27: Point array types](25)
In Figure 27 five point types are mentioned. The transmitting entity encodes the point message before sending while the receiving entity parses and interprets this data. The index and group numbers identify a unique point, and the variation describes its data format.

2.10.2 Indexes

As mentioned earlier DNP3 references point arrays using index numbers to identify points having the same point type. Like normal arrays index numbers start with zero and increase as we traverse this array. [25]

2.10.3 Groups

Groups are the entities which classify the data types in a message. Each group number shares a common point type and method of data generation or creation. Another use of group numbers are for classifying the data type needed while outstation report values or file transfers between a Master and outstation. [25]

Some group numbers are as follows:

Group 30—current value of the point
Group 31—frozen value of the point
Group 32—change of current value event
Group 33—change of frozen value event

When a message is transmitted by DNP3 which has response data, the group number and variation associated with every message is also identified. Each message may have a group number from the above group list which enables the participating component to classify the type of value associated. Each counter, binary/analog inputs have a group number associated with them.
2.10.4 Variations

Variations are the encoding formats of the data types in DNP3. Variations are offered on a group basis as for example shown below for group 30 (Variations offered for group 30).

2.10.5 DNP3 objects

DNP3 objects can resemble to the objects known in object oriented programming. DNP3 objects are the formatted and momentary representation of data at a particular point or a group or variation number for transport in a message.

“For example, a message with 6 objects holds the current value of 6 analog input points, expressed as 32-bit integers. Here, the group number is 30, the variation is 3 and there are 6 instances of 32-bit analog input objects.” [25]

2.10.6 Static event and class data

Static Data

Static data as the name suggests is a value of a point, group or variation which is recent and directly computed. E.g. Static binary data can be ‘1’ or ‘0’ and analog data can be a value. The most important part is that it is considered to be most recent at that period of time and does not change. [25]

Events

DNP3 events are the indications when something changes significantly. It can be a state change, values in Outstation databases crossing the thresholds or the inputs data provided by Masters to Outstations cross the allowed dead band. It can be a snapshot at a particular instance of time or some newly arrived information. These scenarios are not served by static data explained earlier and therefore are presented as events. [25]
Classes

DNP3 uses the concept of classes to organize static data and events into several categories:

- Class 0: Static data
- Class 1, 2, 3: Events

The following example is directly quoted from reference [25]:

“The points of most data types may be assigned to one of the four classes. If a point is assigned to Class 0, the point’s present value shall be reported by the outstation in its response to a Class 0 poll, but the outstation shall not store or report any events for that point. If a point is assigned to one of the event classes, the outstation shall store and report events for that point, and the point’s present value shall also be reported by the outstation in its response to a Class 0 poll.”

Table 6: Function code table [25]

<table>
<thead>
<tr>
<th>Present value included in response to static data-type poll?</th>
<th>Point not assigned to any class</th>
<th>Point assigned to Class 0</th>
<th>Point assigned to Class 1</th>
<th>Point assigned to Class 2</th>
<th>Point assigned to Class 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
2.11 Outstation event buffering

Outstation event buffering as shown in Figure 28 is a mechanism which prevents events from getting lost or checks for duplicate events. A small messaging buffer is placed in an array like data structure at the outstation. This data structure holds the events generated for some period of time with a maximum time threshold. The events reside in the structure until they are processed or until the time threshold occurs. Once the events are processed the outstation gets a delivery notification from the master. This guarantees events are not lost. Once the event is processed it is removed from the buffer and duplicate events are avoided. [25]

Figure 28: Event buffering [25]
2.12 Report by exception

Many of the object groups have data that changes all the time and is called the ‘change data’. Change data may be points that have changed for a specifically corresponding group. Like object group 1 represents binary inputs (static data) and object group 2 represents binary input change data. When a point in object group 1 is detected to have changed, change event in object group 2 for the same point member is created. This, including only points that have changed in response messages allows smaller and efficient messages. This reporting scheme is called report by exception (RBE) [21]

For each change point, a time can be associated with the change; each detection of a data value that changes is a change event. In DNP3, object groups, and the data points within them, can be further organized into classes. There are four classes defined in DNP3. Class 0 represents all static (not change event data). Classes 1, 2, and 3, represent different priorities of change event data. Assuming class 1 contains the highest priority change event data and class 3 contains the lowest priority change event data, a class 1 poll would ideally be performed as often as possible, a class 2 poll would be performed less often, and a class 3 poll would be performed even less often.
Chapter 3

DNP3 SECURITY

Need for security- What to secure, Why to secure

DNP3 protocol was designed for SCADA communication only. In early days SCADA systems were not connected to internet. They were secure from cyber-attacks. But then with the advent of Internet the need to connect SCADA systems to fast outside networks raised so that vendors can carry out critical operations on a remote locations or dedicated workstations geographically distant from SCADA network. DNP3 was not designed with the security in mind. Industrial Systems are nowadays exposed to new kinds of malicious threats. This resulted into severe cyber-attacks on a SCADA network where DNP3 haplessly failed to the malicious intent of attackers without protection. Several scientific works [12] have showed how SCADA i.e. the systems which control industrial installations, are exposed to cyber-attacks. These, due to the intrinsic nature of SCADA systems, cannot be avoided by using traditional Information and Communication Technology (ICT) security measures. [12]

Modern Critical Infrastructures like Power Plants, Water Grids are still largely dependent on ICT to harness new features and carry out operations. In particular, according to a relatively new trend, several of the maintenance and management operations related to such installations, are conducted remotely taking advantage of public networks i.e. Internet. This has contributed by making the process fast and efficient but on the other hand it has exposed such critical operations to new sources of possible threats. In fact, connecting together the critical systems by using the Internet have opened new virtual gates to those interested in damaging such critical
infrastructures. The ICT security measures like firewalls, antivirus or intrusion detection is simply not sufficient enough and proves to be too cumbersome to maintain.

The flow of data in every SCADA system is totally dependent on the communication protocol (e.g. Modbus, DNP3 etc.). By using these protocols it is possible, for example, to force the opening of a valve, or perform other related operation. [12]

Following are list of points that DNP3 lacks when security is main issue.

1. Master and slave do not authenticate each other. This scenario is extremely dangerous, since, potentially, an attacker could take the control of the critical installation.

2. Another threat that can be considered is an attacker could send to the master fake slave reply packets, providing it with false information about the state of the system. In this case, the attacker will be able to eventually hide dangerous state of the system

Approaches to improve DNP3 security

As mentioned in chapter 1, the approaches that can be examined to improve the DNP3 security enhancements in SCADA communications to reduce the vulnerability of cyber attacks are

1. Solutions that alter DNP3 fundamentally [8]

2. Solutions that wrap the DNP3 protocols without making changes to the protocols.[8]

While making sure that the protocol is secure and resilient from the cyber-attacks (discussed further) it should also be kept in mind that the security doesn’t prove to be an additional overhead to the extent that it affects the efficiency of the protocol.

3.1 Solutions that alter DNP3 fundamentally

3.1.1 Purpose of this approach
This approach examines changing the protocol internally by adding an authentication mechanism in its application layer. The main objective of altering the DNP3 protocol internally is to provide authentication.

The protocol model when this mechanism is added will serve the following:

- “A DNP3 outstation can use to unambiguously determine it is communicating with a user who is authorized to access the services of the outstation.
- A DNP3 master can use to unambiguously determine that it is communicating with the correct outstation.” [25]

3.1.2 Threats addressed

The following security threats are successfully defended with this approach.

Spoofing – “a spoofing attack is a situation in which one person or program successfully masquerades as another by falsifying data and thereby gaining an illegitimate advantage. [14]”

Modification- “The attacker removes a message from network traffic, alters it, and reinserts it. This is called an active attack, because it involves an attempts to change information; in comparison, a passive attack, such as password sniffing, seeks information but does not itself modify the valid information, although it may be used in conjunction with an active form of attack for various purposes.” [15]

Replay- A replay attack is a form of network attack in which a valid data transmission is maliciously or fraudulently repeated or delayed. This is carried out either by the originator or by an adversary who intercepts the data and retransmits it, possibly as part of a masquerade attack by IP packet substitution (such as stream cipher attack).” [16]
Eavesdropping- “The attacker makes independent connections with the victims and relays messages between them, making them believe that they are talking directly to each other over a private connection, when in fact the entire conversation is controlled by the attacker.” [17]

Non repudiation- “Non-repudiation refers to a state of affairs where the purported maker of a statement will not be able to successfully challenge the validity of the statement.” [18]

3.1.3 Application layer security

The solutions that change the protocol fundamentally target to provide security at application layer. Application layer security as shown in Figure 29 adds user and device authentication as well as data integrity protection to the DNP3 protocol. Secure DNP3 is a bi-directional protocol that provides protection between master stations and outstations.

Figure 29: Application layer security [31]
Features of Application layer security as shown in Figure 29

“DNP3 may be used over a variety of different physical networks and may be “bridged” from one to the other. Only authentication at the Application Layer shall be used to provide end-to-end security.” [25]

Application layer authentication provides a unique feature which protects against the internal malicious application(s) residing inside the master or slave systems. It also authenticates individual users. The mechanism is based on a challenge and response mechanism.

Description of the process

When the messages are authenticated the ability to distinguish between users is achieved something which is not possible in non-authenticated DNP3 messages. The authentication requires additional DNP3 function codes and object variations added to standard. [25]

Figure 30: Design of application layer security [25]
The actual authentication is based on two concepts:

A challenge and response process between initiator and receiver, as discussed below

The use of a Keyed-Hash Message Authentication Code (HMAC) that both the outstations and the masters calculate based on each ASDU, or protocol message that is to be authenticated. [25]

An HMAC algorithm is a mathematical calculation that takes a protocol message as input, produces a smaller piece of data as output. It has the following characteristics:

- The hash is very sensitive to even small changes. Easy to detect if modified.
- Makes use of shared secret key which is shared by participating entities.
- It is very difficult to reverse engineer.
- Due to high randomness of the hash values it is difficult to find two messages that produce the same HMAC. [25]

Steps in authentication

Initiating the challenge

The challenge is initiated by a Master or an outstation. Challenger initiates a challenge to protect the application layer fragment it is trying to protect. The challenger issues the challenge immediately after receiving the critical ASDU, before taking any action on it. Outstation assumes all the operation it performs as critical. Challenge is mainly issued when the ASDU contains events related data which is considered extremely critical. A keyed –HMAC is also included when a challenge is initiated. [25]

Replying to the challenge
The device that receives the challenge is responsible for setting up a reply. The responder performs an HMAC algorithm specified in the challenge message to produce the reply using shared Session Key known to both devices [25].

Authentication success

Upon receiving the reply, the challenger performs the same calculation on the same data used by the responder. If the results match as shown in Figure 31, the challenger permits communications to continue. If the challenger was protecting a particular ASDU, it processes the ASDU. [25]
Figure 31: Successful challenge of critical ASDU [25]
**Authentication failure**

If the value computed by challenger does not match the challenger behaves as if it has not processed the ASDU and an error message is generated. The requested operation is also denied. [25]

---

**Figure 32: Unsuccessful challenge of critical ASDU [25]**
Aggressive mode

Aggressive mode is an efficient technique to save bandwidth in which responder while attempting a critical operation just ‘anticipates a challenge and sends over unsolicited HMAC value in the same message it attempts to send. It eliminates the challenge and reply messages.

“Aggressive mode provides a lower level of security because not as much data is changing in each message. However, the value of aggressive mode is considered high enough for DNP3 that all DNP3 implementations of this authentication mechanism are required to support it. Per IEC/TS 62351-5,( IEC/TS 62351-5 specifies messages, procedures and algorithms for securing the operation of all protocols based on or derived from the standard IEC 60870-5) however, all DNP3 implementations are also required to permit it to be disabled by configuration.” [25]

![Figure 33: Successful aggressive challenge](image-url)
Figure 34: Unsuccessful aggressive challenge [25]
Managing session keys

As in case of any secured communication Session Keys are used the most frequently to hash the data. A different Session Key is used in each direction, so that if the key for one direction is compromised, it does not compromise communications in the other direction. The master initializes the Session Keys immediately after communications is established and regularly changes the Session Keys thereafter. The session key is changed periodically so that it can be kept safe from crypt-analysis attacks. Another key called Update keys are used to encrypt new session keys along with challenge data. The combined message is called Key Change message.

Sequence for changing the Session keys is as follows (Figure 35):

- The master initiates the process by sending a Key Status Request message having no data. It does include a User Number that indicates the particular Update Key and set of Session Keys being queried.
- The outstation replies with a Key Status Response message containing the current status of the keys and some challenge data.
- On receiving the response from outstation the master updates the Session Keys with a Key Change message. With the key change it also contains a reply which serves as reply to challenge and allows the outstation to authenticate the party with which it is communicating.

- A new Key Status message is sent by outstation indicating if the key was changed successfully or not.

If master or outstation device detect a failure they assume that most recent session keys have been compromised and they mutually refuse to authenticate any further challenges. [25]
Key Status Request

Master

Key Status Reply

Key Change (Response 1)

Key Change OK

Normal Requests

Key Status Request

Key change timer expires

Figure 35: Session key initialization and periodic update [25]
Normal Request
Master Outstation
Response Lost
Key Status Request Lost
Key Status Request
Key Status
Protocol Request timer
Expires
Key Change
Key Status
(OK, Challenge)
Key Status Timer Expires

Figure 36: Communication failure after session key exchange [25]
State diagrams of the master and outstation during the authentication process

The state diagrams (Figure 37 & 38) summarize the general operation of the authenticated protocol. The security Idle and Wait for reply states are common to both the Master and outstation.

Figure 37: State transitions for master [25]
Figure 38: State transitions for outstation [25]
An event not pertaining to Security Idle or for not pertaining current user under consideration is normally considered as error. The event may be queued for further consideration.

“For e.g. If there are three users at the master, the master begins by sending a Request Key Status message for the first user and entering Wait for Key Status for that user. The initialization event for the other two users is queued up by the master. The second user does not enter Wait for Key Status until the first user has reached Security Idle by either succeeding or failing to initialize its Session Keys.” [25]

Master authentication message definitions

DNP3 masters shall implement the authentication mechanism using the function codes and objects as described in Table 8. The first column of the table shows how these function codes and objects correspond to the IEC/TS 62351-5 specification.
Table 9: DNP3 master messages with authentication codes

<table>
<thead>
<tr>
<th>IEC/TS 62351-5 message</th>
<th>Description</th>
<th>Message from master contains</th>
<th>DNP3 function codes</th>
<th>Outstation responds with</th>
<th>DNP3 function codes</th>
<th>DNP3 objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Challenge</td>
<td>Requests authentication of the preceding outstation DNP3 Response or Unsolicited Response</td>
<td>0x20 Authentication Request</td>
<td>g120v1 Authentication Challenge object</td>
<td>0x83 Authentication Response</td>
<td>g120v2 Authentication Reply object</td>
<td></td>
</tr>
<tr>
<td>Reply</td>
<td>Provides authentication of a Challenge from the outstation</td>
<td>0x20 Authentication Request</td>
<td>g120v2 Authentication Reply object</td>
<td>0x83 Authentication Response</td>
<td>If authentication was successful</td>
<td></td>
</tr>
<tr>
<td>Aggressive Mode Request</td>
<td>Provides authentication for the current DNP3 Request</td>
<td>Whatever function code is in the DNP3 Request</td>
<td>g120v3 Authentication Aggressive Mode Request object. Shall be first object. ... Objects appropriate for standard DNP3 request ... g120v9 Authentication HMAC object. Shall be last object</td>
<td>0x83 Authentication Response</td>
<td>If authentication failed</td>
<td></td>
</tr>
<tr>
<td>Key Status Request</td>
<td>Requests the current status of the Session Keys</td>
<td>0x20 Authentication Request</td>
<td>g120v4 Session Key Status Request object</td>
<td>0x83 Authentication Response</td>
<td>g120v5 Session Key Status object</td>
<td></td>
</tr>
<tr>
<td>Key Change</td>
<td>Changes the symmetric Session Keys subsequently used by master and outstation for authentication</td>
<td>0x20 Authentication Request</td>
<td>g120v6 Session Key Change object</td>
<td>0x83 Authentication Response</td>
<td>If authentication was successful</td>
<td></td>
</tr>
<tr>
<td>Error</td>
<td>Indicates the authentication provided in the challenge reply from outstation was incorrect or that the outstation's aggressive mode DNP3 response did not correctly authenticate</td>
<td>0x21 Authentication Request—No Ack</td>
<td>g120v7 Authentication Error object</td>
<td>None</td>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>

NOTE—Shaded cells indicate optional messages.
**Table 10: DNP3 outstation messages with authentication codes [25]**

<table>
<thead>
<tr>
<th>IEC/TS 62351-5 message</th>
<th>Description</th>
<th>Message from outstation contains</th>
<th>Message initiated because master sent</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>DNP3 function codes</td>
<td>DNP3 objects</td>
</tr>
<tr>
<td>Challenge</td>
<td>Requests authentication of the preceding master DNP3 Request</td>
<td>0x83 Authentication Response</td>
<td>g120v1 Authentication Challenge object</td>
</tr>
<tr>
<td>Reply</td>
<td>Provides authentication of a Challenge from the master</td>
<td>0x83 Authentication Response</td>
<td>g120v2 Authentication Reply object</td>
</tr>
<tr>
<td>Aggressive Mode Request</td>
<td>Provides authentication for the outstation’s current DNP3 Response</td>
<td>0x81 Response or 0x82 Unsolicited Response</td>
<td>g120v3 Authentication Aggressive Mode Request object. Shall be first object. Objects appropriate for standard DNP3 response g120v9 Authentication HMAC object. Shall be last object</td>
</tr>
<tr>
<td>Key Status</td>
<td>Response providing the outstation’s current status of the Session Keys</td>
<td>0x83 Authentication Response</td>
<td>g120v5 Session Key Status object</td>
</tr>
<tr>
<td>Key Change</td>
<td>Changes the symmetric Session Keys subsequently used by master and outstation for authentication</td>
<td>0x83 Authentication Response If authentication was successful</td>
<td>g120v5 Session Key Status object</td>
</tr>
<tr>
<td></td>
<td>Indicating authentication provided in the previous challenge reply from master was incorrect or that an aggressive mode request did not authenticate</td>
<td>0x83 Authentication Response If authentication failed</td>
<td>g120v7 Authentication Error object</td>
</tr>
</tbody>
</table>

NOTE—Shaded cells indicate optional messages.
DNP3 authentication challenge, reply or error message will have the same DNP3 application sequence number as the critical DNP3 fragment that was challenged.

Figure 39: DNP3 select-operate authentication using sequence numbers [25]

Figure 40 illustrates that when the aggressive mode is used, the sequence numbering is identical to normal non-authenticated DNP3.
Figure 40: DNP3 select-operate authentication in aggressive mode [25]
Figure 41 illustrates an example of a failed DNP3 select/operate authentication.

![Figure 41: DNP3 select-operate authentication in aggressive mode - failed][25]

3.1.4 DNPSec: It can be considered another approach which makes fundamental changes to DNP3 protocol header format to add authentication. As similar to earlier approach DNPSec authentication and integrity framework provides verification for frame origin; it also guarantees against modification and anti-replay attacks. The confidentiality framework will provide encryption to protect against eavesdropping. This requires some modifications to DNP3 LPDU (link layer protocol data unit) to provide these capabilities. CRC is a common technique used in DNP3 for detecting data transmission errors. CRCs occupy 34 bytes out of 292 bytes of the DNP3 LPDU for integrity. These bytes will be utilized in a different way in the DNPSec framework. [6]
DNPSec can be best explained with 2 important components that make it

1. The structure to construct the frame and transfer data securely between Master and outstation.

2. The key exchange during installation and connection setup between Master and outstation.

The fields as shown in Figure 42 are described as follows:

1. New Header: Unsigned 4 bytes field containing destination address( 2 bytes); the MH flag bit to recognize if the message is from Primary master (0) or secondary master (1); SK bit to indicate the slave if message contains new session key (1) . 14 bits are reserved.

2. Key Sequence number- Unsigned 4 byte field containing a counter value that increases by one for each message sent by the Master and places the value in the key sequence number field. Range 1 to \((2^{32}-1)\). Once the value is reached the master terminates the session key.

3. Original LH header- DNP3 data-link header without 2 CRCs. The payload is protected using encryption.
4. Authentication data field containing Integrity Check Value (ICV) computed over the key sequence number, original LH header and payload data fields. ICV provides integrity services and is provided by HMAC-MD5-96. [6]

The DNPSec fields are [6]:

0 – 3 New Header (4 bytes)
   • DA: 0-1 Destination Address (2 bytes)
   • MH: 2(bit 0) 0: Primary Master Host,
     1: Secondary Master Host
   • SK: 2(bit 1) 0: Fetch the database for the session key,
     1: The frame contains a KSN value from the Master.
   • 2-3 Reserved (2 bytes)

4 – 7 Key Sequence Number (4 bytes)

8 – 15 Original LH Header (8 bytes)
   • 8 – 9 Sync (2 bytes)
   • 10 - 10 Length (1 byte)
   • 11 – 11 Link Control (1 byte)
   • 12 – 13 Destination Address (2 bytes)
   • 14 – 15 Source Address

16 – 271 Payload data (256 bytes)
   • 16 – 265 TPDU data
   • 266 – 271 padding dummy data

272 – 291 Authentication Data (20 bytes)

Key Management in DNPSec with reference to Figure 43
The Master manages a secure database M_Keydb (shown as a squared table under Master in Figure 43). The database has 4 most important tables the shared session key, the time stamp, outstation address and Key Sequence Number (KSN) The outstation address is used as an index for querying the database. Using a function called M_GenKey the Master generates a new and unique session key every time when old key expires. This keeps the process free from crypt-analysis attack. [6]. At the slave database S_Keydb two separate keys are maintained for communicating with Primary and Secondary. [6]

![Figure 43: DNPSec communication](image-url)
3.2 Solutions that wrap DNP3 into external secure protocol

3.2.1 SSL/TLS

This is an approach which is exactly opposite of the previous approach. As mentioned earlier this approach does not change the protocol internally but tries to run it on a secured medium. SSL (Secured Socket Layer) /TLS (Transaction Layer Security) secures the communication channel to carry out traffic over the TCP-IP. This mechanism secures the communication channels in order to carry out reliable communication over TCP/IP and is in use for about a decade providing private network for the Internet users. SSL/TLS secures communication between a client and a server by allowing mutual authentication and provides integrity by using digital signatures and privacy via encryption.

How SSL works with DNP3

The secured socket layer is placed between a reliable connection dependent protocol like a TCP-IP and DNP3 packets are sent over it. This mechanism provides a secure communication between a remote DNP3 Master and a SCADA DNP3 outstation. Security techniques like mutual authentication, digital signatures and encryption are achieved automatically with this technique at a low cost as they come with SSL-TLS itself. The protocol is also designed for specific algorithms used for cryptography, digests and digital signatures. [20]

SSL Session Establishment

The SSL session is established when a handshake sequence happens between a client and server as shown in Figure 44.
The steps in authentication are:

1. Negotiate the Cipher Suite to be used during the transfer
2. Establish and share a session key between communicating entities
3. Authenticate the server to the client and vice versa. [20]

3.3 Comparison of both the approaches

This report discussed two approaches to add security to DNP3 protocol [25]. In this section the approaches described in previous two sections are compared to inspect their strengths and weaknesses.

According to SSL-TLS approach DNP3 messages are transferred over a network which is already secured. Thus this solution tends to be very quick and low cost as the implementers do not have to worry about changing the internal details of DNP3. The implementers and vendors only need to secure their own network and not worry about the internal details of the protocol. The vendors
can secure the entire network by using a VPN or other equivalent techniques. SSL-TLS allows mutual authentication by using its own techniques like digital signature and encryption. It also guards against man in middle and replay attacks.

Another advantage of using this approach is it provides an excellent watchdog when the Master/Outstation application software uses web monitors and application servers. SSL/TLS is well established in techniques or applications which use web browsers, web and application servers. Since application servers provide transaction security of their own any request originating from Master or Outstation application would already be secure. In addition to these features if a vendor decides to change his basic communication protocol from DNP3 to its contemporary protocol like Modbus the underlying systems would not take a security hit. General acceptance of DNP3 over SSL-TLS is growing. This assumption can be stated with the fact that recently the IEC (International Electro technical Commission) committee accepted SSL/TLS as a part of their security standard. In the future DNP3 technical committee can accept the same approach for DNP3 security standard.

On the flip side however this approaches suffers from some disadvantages. If the Master wants to broadcast a message to all the outstations in the network then running SSL/TLS over a TCP based network is undesired. As a result running DNP3 over SSL-TLS would not be an ideal solution. Also this approach cannot guard against non-repudiation attacks (assurance about senders identity). In this approach the network on which it runs is secured but the actual DNP3 packets that are transferred are not secured (as done in the approach which changes the protocol fundamentally). As a result a direct cyber-attack on the Master or Outstation system may generate a big threat. The SSL-TLS approach is totally dependent on the underlying cryptography or tool used. If the algorithm is sacrificed then this approach may fail.
In the second approach the security is provided by making fundamental changes in the protocol at its application level. This approach as already discussed adds authentication at the application layer. Rather than securing the channel, the messages that move over the channel are secured. This can be considered an intense approach and it provides a more robust security. Security provided is independent of underlying channels and message exchanges can be done over a TCP/IP or UDP/IP based network. Additional headers are appended at the application layer when the DNP3 message is sent over the network. The biggest advantage of this approach over SSL-TLS is that since the messages instead of network, are protected, this technique can be considered as end to end security solution.

On the other hand this approach is very specific to DNP3 protocol and individual messages need to be secured in communication. This tends to be time consuming and cumbersome. As this approach makes fundamental changes in protocol it is very expensive to implement as fundamental changes need to be made to the DNP3 Master and Outstation application programs. Also due to the solutions specific nature to DNP3 if a vendor decides to change the underlying protocol to Modbus, this approach would not help. Adding authentication puts some overhead on the application layer processing as it has to manage an extra challenge-response data as a part of communication. This places an additional overhead on the communication and may bring down efficiency of DNP3. A summary of comparison is shown in table 11.
<table>
<thead>
<tr>
<th>Feature</th>
<th>DNP3 over SSL/TLS</th>
<th>DNP3 application layer security</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal or external to protocol</td>
<td>External (No modification to the protocol messages)</td>
<td>Internal (Modifications done to protocol messages)</td>
</tr>
<tr>
<td>How is it done</td>
<td>Network is secured</td>
<td>Messages are secured</td>
</tr>
<tr>
<td>Provides protection against these attacks</td>
<td>Man in Middle, Replay attacks</td>
<td>Spoofing, modification, reply, eavesdropping, non-repudiation</td>
</tr>
<tr>
<td>Cost</td>
<td>Relatively low (changes are done only to the network)</td>
<td>Relatively high (changes need to be made to outstation/master systems to process changed DNP3 messages)</td>
</tr>
<tr>
<td>Ease of use</td>
<td>Easy to implement</td>
<td>Intense and difficult to implement</td>
</tr>
<tr>
<td>Scalability</td>
<td>Can be scaled to other protocols like Modbus/IEC</td>
<td>DNP3 specific</td>
</tr>
<tr>
<td>Extensiveness</td>
<td>Messages are not secured after receiving and before sending</td>
<td>Complete and End to End security</td>
</tr>
<tr>
<td>Acceptance</td>
<td>Still under review but getting widely accepted</td>
<td>Fully accepted and included in the security specification of DNP3</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>------------------------------------------------</td>
<td>----------------------------------------------------------------</td>
</tr>
<tr>
<td>User authentication</td>
<td>Not done</td>
<td>Users are also authenticated</td>
</tr>
<tr>
<td>Performance hit on DNP3 efficiency</td>
<td>No adverse performance hit</td>
<td>Performance may be affected due to extra challenge and response exchange</td>
</tr>
</tbody>
</table>
4.1 Scope of improvement in DNP3

As discussed in the introduction of Chapter 2 DNP3 was never developed with the idea of running it on TCP-IP/UDP networks. DNP3 was originally designed for serial point-to-point SCADA communication (e.g., RS-232) in a standalone SCADA station with limited support for half-duplex serial networks (e.g., RS-485). With the advent of internet many vendors recognized a need for devices to exchange DNP3 messages over high-speed digital networks. [24]

The approach described in the reference [25] investigates carrying DNP3 communication over internet. While the scalability of DNP3 is increased tremendously due to this an effort is also made to not to redefine any of the existing DNP3 layers and also not to change the end devices that take active part in communication like DNP3 Master and Outstations.

The Internet Protocol suite is designed to be highly platform independent, network independent and seamlessly weave all the private public networks together with full transparency. It may scale from simple LAN to highly complicated WAN networks. Large amounts of data can be transferred across the whole web due to this integration.

4.2 Internet protocol stack with DNP3

The Internet Protocol suite and DNP3 use a layering paradigm; each piece of the protocol stack in one station logically communicates with the corresponding piece in the other station. It is therefore easy to build DNP3 “on top of” the Internet Protocol suite as shown in Figure 46 since the Internet layers appear transparent to the DNP3 layers. [24]
4.3 Layer requirements

The DNP3 Link, Transport, and Application Layers require no change of specification except in the time synchronization method for network operation. Link frames are transported unchanged over the Internet Protocol suite under the control of the connection management layer.

Figure 46: DNP3 on internet protocol stack [25]
The main requirements are as follows:

1. **Confirmations**

   The components would not confirm the data link layer frames while transferring DNP3 frames over IP suite. The Application Layer confirmation would be taken as final. This is done to keep the process of confirmation transparent while dealing with any kind of network underneath.

2. **Message Transfer**

   The process of establishing a connection while dealing with TCP/IP networks would remain the same. Once a connection is established DNP3 link layer should be able to transfer the packets. For UDP/IP networks the communication would be carried out using socket connection. The basic DNP3 operations like Master polling the outstation for status updates, report by exception are kept independent from connection establishment. [25]

   “For example, a typical case is for the master to establish a TCP connection with an outstation, perform a class 1,2,3,0 request, enable unsolicited messages, and operate in an unsolicited mode from then on.” [25]

4.4 **Internet protocol suite requirements for DNP3**

   DNP3 would make use of entire Internet protocol suite to provide data transfer. An internet protocol suite is composed of many protocols that work together to enable communication. [25]

   1. **Configuration requirements**

   As mentioned earlier DNP3 devices would make use of both TCP and UDP. A master station would initiate a connection and decide to route the messages using a TCP connection or send the DNP3 message as a UDP datagram.
As a part of initiating process Master would decide a TCP handshake connection endpoint/dual end point or just a UDP endpoint.

2. Registered port number

The port number 20000 has been registered with the IANA (Internet Assigned Numbers Authority) for use with DNP3. All devices shall support communications using this port number. [25]

3. Connection management

This is the interfacing layer between DNP3 protocol layers and IP suite layers. This layer can be compared to the data link layer in OSI model. Connection management layer sets up TCP connections if a TCP in case of connection oriented packet transfer. The connection management layer provides the interface between the DNP3 protocol layers and the Internet Protocol suite. It also accepts UDP connections. [25]

4.5 DNP3 over TCP-IP

The main qualities of TCP like reliability of data transfer, connection oriented communication (both devices shall participate in the defined synchronization scheme in order to establish the connection.) and acceptance by wide area networks are useful for DNP3 communication. TCP provides a serial data stream between devices for the DNP3 Link Layer and thus enables common integration with existing serial codes. [25]

The process of setting up a connection is as follows.

Master initiates an endpoint by doing a TCP active open. An outstation listens on a listening end point by doing a TCP passive open.
Dual end point: In this mode a Master and an Outstation (if it supports) can listen as well as accept connections. If the device wants to send data it can do an active open. The advantage of this mode is any side participating in the communication can send or receive data. Master uses this to poll a remote outstation while outstation may do this to send status report or send warnings or signals.

The main advantage of this mode is while the outstation is listening on the data and an emergency arises where it has to report data to the master in an unsolicited way it may do an active open and initiate data transfer.

Once the transfer is complete the devices return to the listening mode. If both sides initiate connection then the priority is given to the master and other connections are aborted. [25]

4.6 DNP3 over UDP-IP

There are some characteristics of UDP which make it an efficient communication medium for DNP3 communication. The networks which are already highly reliable don’t need TCP as the underlying network. In these cases DNP3 communication can be used over UDP to avoid communication overhead.

Another advantage of running DNP3 over UDP is a case where Master has to broadcast data to all the outstations on the network. It won’t have to worry about connection establishment and acceptance.

UDP can be used in DNP3 networking in following scenarios as shown in Figure 47

The following example is directly quoted from reference [25]

“Case A: When the communication channel uses TCP, an outstation shall provide a UDP end point at port 20000 to receive broadcast messages from the master station. This permits the
master to send, for example, freeze or disable unsolicited commands to all logical devices within an outstation. Outstations do not respond to broadcast messages.”

“Case B: When the communication channel uses UDP only, an outstation shall provide a UDP end point at port 20000 (or optionally other port numbers) to receive all messages from the master station. Responses shall be returned to the master with the destination port number set to one of the values B1/B2/B3 as shown in Figure 48:”

“Case C: Additionally, an outstation configured for unsolicited responses shall send its initial null response to either C1 or C2. If the initial unsolicited response is sent to a configured port number, the outstation may, based on its configuration, use the source port number from subsequent requests or continue sending all responses, solicited and unsolicited, using this configured number as the destination port. This method provides maximum control for a system designer to assign specific port numbers for individual outstations.”
4.7 TCP connection and keep alive mechanism

If the device listening a TCP connection suddenly goes down and comes back up there is no way for the by which the other end can know about this. This can become a big shortcoming in DNP3 communication because a master always needs to know if all the stations are up. As explained earlier it does this by constant polling in standalone SCADA network station. But when it is communicating on a TCP enabled network it may not know about the status of inactive outstation.
till the time it actually sends a message. For the Master to know in these circumstances DNP3 devices are required to maintain a keep alive timer to monitor the status of an active TCP connection between them. Every time a message is send and received the timer is restarted from the receiving end. [25]
Chapter 5

CONCLUSION

5.1 Summary

This report started with an analysis of DNP3 protocol and its efficient role in SCADA communication. It stated how SCADA systems communicate; the main components that participate in the architecture. Then it gradually described how DNP3 fits in the whole SCADA architecture. It also described the DNP3 protocol stack with detailed layered and diagrammatic description of each layer and the responsibilities of it. The features of DNP3 which makes it SCADA favorite like points, indexes, objects, classes, events and RBE were described in detail in the concluding part of DNP3 analysis. Then, the focus shifted on the security of DNP3 and SCADA systems. The report brought out some security vulnerabilities that are present in DNP3 and the proposed threats the SCADA systems can face if those vulnerabilities are not correctly addressed. The report presented two main approaches that can be adopted to improve the security in DNP3. The first approach deals with improving the internal security of the protocol by providing authentication at the application layer of the protocol. The second approach deals with improving the security by providing external safeguards to the protocol like SSL/TLS or DNPSec framework. In the next section the report talked about improving the efficiency of the protocol. In order to improve efficiency of an already efficient protocol, this report tried to bring out approaches which can enable vendors to implement the protocol on TCP-UDP/IP protocol suite. This new improvement in DNP3 can help the SCADA systems to harness the power of Internet and could only improve the scalability and reach of DNP3.
5.2 Future work

On 7th February, 2012, the IEEE announced that work was proceeding to update IEEE 1815™, the standard defining the DNP3 Specification. The revisions include a significant update to the Secure Authentication section of the specification. The IEEE balloting procedure to adopt these updates is expected to commence in January 2012, with publication of the revised IEEE 1815 coming later in 2012. This IEEE announcement has been reported in a number of control system technical journals and websites. [29]
## GLOSSARY

1. SCADA  Supervisory Control And Data Acquisition  
2. DNP3  Distributed Networking Protocol  
3. RTU  Remote Terminal Unit  
4. IED  Intelligent Electronic Devices  
5. TCP  Transmission Control Protocol  
6. IPSec  Internet Protocol Security  
7. IP  Internet Protocol  
8. TLS  Transaction Layer Security  
9. SSL  Secured Socket Layer  
10. IEEE  Institute of Electrical and Electronics Engineers  
11. IEC  International Electro technical Commission  
12. ASDU  Application Service Data Units  
13. APCI  Application Protocol Control Information  
14. IIN  Internal Indications  
15. TPDU  Transport Protocol Data Units  
16. LPDU  Link Protocol Data Units  
17. CRC  Cyclic Redundancy Check  
18. RBE  Report by exception  
19. ICT  Information and Communication Technologies  
20. ICV  Integrity Check Value  
21. KSN  Key Sequence Number
REFERENCES


[4] SCADA and substation Control communication by Andrew West [Online]

Available: ist.psu.edu


Available: http://m.controleng.com/index.php?id=2819&tx_ttnews%5Btt_news%5D=9392&cHash=6c086b25f6


Protocol (DNP3)

[10] DNP users group online library [Online]
Available: www.dnp.org

Available: IEEE online library

Available: IEEE online library


Available:  http://httpd.apache.org/docs/2.0/ssl/ssl_intro.html#AC96

Available:  www.dnp.org , Online Library

[22] DNP3 Application notes AN2004-001, [Online]
Available:  www.dnp.org, Online Library

[23] DNP3 Vol3 Transport Function [Online]
Available:  www.dnp.org, Online Library

Available:  www.dnp.org, Online Library

Available:  www.dnp.org, Online Library

[26] Secure SCADA [Online]
Available:
http://www.smartgridnews.com/artman/publish/industry/DNP_Secure_Authentication_Essential_to_Smart_Grid_Progress.html


Available: [29] IEEE journal on release of DNP3 secure authentication Version 5 by Lorene Cunningham

Available to members at www.dnp3.org


[31] NOJA Power DNP3 Security: Security and Challenges by Allan Scott