Protocolo de Comunicação DNP3

Prof. Lucas Silveira

Introdução ao DNP3

O que é DNP3?

- DNP 3.0 ou DNP3 (Distributed Network Protocol) é um protocolo de comunicação de dados baseado na **norma IEC 60870-5** e foi criado em 1993, pela empresa Harris.
- É um protocolo para transmissões de dados entre `master stations` e `UTRs` (Unidades Terminais Remotas) ou `IEDs` (Dispositivos Eletrônicos Inteligentes).

O que é DNP3?

- O DNP 3.0 abrange funções das **camadas de aplicação, de enlace de dados e física**, em uma arquitetura de rede simplificada, denominada `EPA` (Enhanced Performance Architecture), e possui uma *pseudo camada de transporte* junto à camada de aplicação que faz a separação de mensagens maiores que 249 bytes.
- O DNP 3.0 é um *protocolo aberto*, porém, não disponibiliza as especificações e normas no domínio público. Logo:

> Para ter acesso às suas especificações é necessário ser membro do grupo ou comprar as normas que o descrevem.

- O site oficial do grupo de estudos DNP (DNP Users Group) é o www.dnp.org.

Breve Histórico do DNP3 e suas Áreas de Aplicação

- Desenvolvido em **1993** pela **GE-Harris**
- Usado principalmente nos setores de:
  - **Energia Elétrica**,
  - **Petróleo e Gás**,
  - **Água**, e
  - **Sistemas de Transporte**.
- É um *padrão aberto*: amplamente adotado no mundo todo.

Tipos de Dados possíveis de serem representados no DNP3

- Entradas digitais,
- Entradas analógicas com conversor A/D (Analógico/Digital) para 16 ou 32 bits,
- Contadores,
- Congelamento de contadores,
- Eventos com estampa de tempo (timestamp),
- Sincronismo de tempo,
- Saídas digitais e analógicas,
- Sequência de octetos (string),
- Entre outros.

Principais Tipos de Mensagens

- Solicitações de leitura
- Solicitações de comando
- Respostas de evento
- Mensagens de confirmação

Tipos de objetos de dados DNP3

No DNP3 Existem objetos estáticos, como é o caso dos objetos:

- 01 (Binary Input),
- 20 (Binary Counter),
- 30 (Analog Input) e
- 40 (Analog Output),

Que representam os **dados no instante da leitura.**

Tipos de objetos de dados DNP3

Já os objetos:

- 02 (Binary Input Change),
- 22 (Binary Counter Change),
- 32 (Analog Input Change) e
- 42 (Analog Output Change), por exemplo,

Representam os **dados de eventos**, ou seja, apenas os dados que mudaram de valor até a última varredura do mestre.

Classes de dados DNP3

O DNP3 pode agrupar estes objetos dinâmicos em 4 classes (classes 1, 2, 3 e 0), que podem ser utilizadas para priorização de eventos:

- Classe 1 `Changed Data` – Prioridade Alta
- Classe 2 `Changed Data` – Prioridade Média
- Classe 3 `Changed Data` – Prioridade Baixa
- Classe 0 `Static Data` – Todos os dados

Estrutura organizacional dos dados no DNP3

- Um **ponto de dados** é:

> *Um único valor de dados do tipo especificado por seu grupo de objeto.*

- Também, dentro de cada **grupo de objeto**, existem suas **variações.** Uma **variação** é:

> *Tipicamente usada para indicar um método diferente de especificar dados dentro do grupo de objetos.*

- Por exemplo, **variações de entradas analógicas** permitem transferência dos dados com:
  - um valor 16 bits do tipo inteiro, **ou como**
  - um valor 32 bits do tipo ponto flutuante.

Estrutura organizacional dos dados no DNP3

- O protocolo disponibiliza também **dois campos de Indicações Internas** (IIN – Internal Indications), que são:

> *Dois bytes que seguem o campo de função em todas as respostas e indicam alguns estados internos do escravo*, como:

- Existência de eventos (dados de classe 1, 2, 3 disponíveis),
- Reinicialização do escravo (Device Restart),
- Pedido de sincronização de tempo (Need Time),
- Dispositivo remoto com problema (Device Trouble),
- Estouro do buffer de eventos no escravo (Buffer Overflow),
- Dentre outros.

Níveis de Implementação de Dispositivos DNP3

- O DNP3 é implementado em **3 níveis nos dispositivos**, sendo que cada nível possui diferentes quantidades de objetos de dados disponibilizados.
- Essa característica deve ser especificada **na compra do equipamento** que utiliza protocolo DNP3 e depende das **necessidades do usuário**.
- Cada equipamento que utiliza o protocolo DNP3 possui um **documento de perfil do dispositivo** (Device Profile).

Níveis de Implementação de Dispositivos DNP3

A norma DNP 3.0 define três níveis de implementação:

- **Nível 1** (Level 1): Nível mais simples, que define a comunicação entre uma estação Mestre ou concentrador de dados e um IED de pequeno porte.
- **Nível 2** (Level 2): Melhoria do nível anterior, que define a comunicação entre uma estação Mestre ou concentrador de dados e um IED de grande porte ou uma Unidade Terminal Remota (UTR).
- **Nível 3** (Level 3): Melhoria do nível anterior, que define a comunicação entre uma estação Mestre e um dispositivo de médio porte (por exemplo, uma Unidade Terminal Remota ou um Concentrador de Dados)

Arquitetura de Camadas do DNP3

O protocolo DNP3 é baseado no modelo EPA (Enhanced Performance Architecture) e utiliza 4 camadas (Layers) de rede.

Camadas do protocolo:

- **Aplicação**: Interpreta comandos e dados
- **Transporte**: Gerencia pacotes de dados
- **Link de dados**: Garante a integridade da comunicação
- **Física**: Transmite bits pelo meio físico

Descrição das camadas

- A **Camada Física** do protocolo compreende os procedimentos de transmissão e recepção no meio físico.
- A **Camada de Enlace** de dados (Data Link Layer) possui dois propósitos:
  - *Prover transferência de informações* (ou quadro de dados) através da conexão física. Nela o quadro de dados do usuário é transformado em um quadro de dados de enlace, com o acréscimo do **cabeçalho** e de **CRCs.**
  - *Prover indicação de outros eventos*, como o estado do enlace.

Descrição das camadas

- É no cabeçalho da camada de enlace que estão os bytes de endereço fonte e destino.
- No nível de enlace não há distinção entre terminal remoto (escravo) e mestre.
- O frame DNP 3 consiste de um cabeçalho e uma seção de dados;
- A pseudo **Camada de Transporte** segmenta mensagens da camada de aplicação em múltiplos pacotes da camada de enlace.
- Na **Camada de Aplicação** são disponibilizados vários **objetos de dados** que podem ser mapeados nos pontos de leitura (entrada) e comando (saída) de uma UTR típica de automação do sistema elétrico.

Camada de Enlace (Data Link Layer) DNP3

- Utiliza os pacotes de *transmissão de dados*.
- Os pacotes de transmissão de dados seguem formatos definidos pela norma IEC 60870-5.
- Estes formatos são chamados `Transmission Frames` e possuem, geralmente, a seguinte estrutura:
  - L = Length (Campo de Tamanho) – `1 byte`
  - C = Control (Campo de Controle) – `1 byte`
  - A = Address (Campo de Endereço) – `1 ou mais bytes`
  - Link User Data (Campo de Dados) – `n bytes`

Camada de Enlace (Data Link Layer) DNP3

Campos do cabeçalho da camada de enlace:

- START = 2 bytes do cabeçalho (0x0564)
- LENGTH = 1 byte representando o tamanho da mensagem, incluindo os campos CONTROL,
- DESTINATION and SOURCE do cabeçalho e o campo USER DATA. Os campos de CRC fields não são incluídos na conta do tamanho do frame. O valor deste campo vai de 5 a 255.
- CONTROL = 1 byte. O campo de controle contém informações sobre a direção da mensagem, o tipo de serviço e suporta funções de controle (Control Field).

Camada de Enlace (Data Link Layer) DNP3

- DESTINATION = 2 bytes contendo o endereço de destino.
- SOURCE = 2 bytes contendo o endereço de fonte.
- CRC = 2 bytes de verificação (Cyclic Redundancy Check).
- USER DATA = Cada bloco de dados contém 16 bytes mais 2 bytes de CRC. O último bloco pode ter até 16 bytes mais 2 de CRC.

Camada de Transporte DNP3

- Quando a mensagem possui dados, é necessária a utilização de uma camada de transporte, que é definida assim:
  - TH = `1 byte` de cabeçalho
  - USER DATA = Cada bloco de dados contém 16 bytes mais 2 bytes de CRC. O último bloco pode ter até 16 bytes mais 2 de CRC.

Campos do Cabeçalho da Camada de Transporte DNP3

- FIN = 1 bit que indica se a mensagem é a última (1) ou se existem mais mensagens a seguir (0).
- FIR = 1 bit que indica se a mensagem é a primeira (1) ou se não é a primeira mensagem da sequência (0).
- SEQUENCE = número de sequência da camada de transporte. Varia de 0 a 63, em incrementos de 1 em 1.

Camada de Aplicação

A camada de aplicação consiste de:

- **APCI** (Application Protocol Control Information)

> Contém o cabeçalho de pedido (request header) ou o cabeçalho de resposta (response header).

- **ASDU** (Application Service Data Unit)

> Contém o cabeçalho de objetos (object header) e os dados relativos aos objetos.

Application Headers

O cabeçalho de pedido (request header) é composto de dois campos:

- AP (Application Control) – 1 byte
- FC (Function Code) – 1 byte

O cabeçalho de resposta (response header) é composto de três campos:

- AP (Application Control) – 1 byte
- FC (Function Code) – 1 byte
- IIN (Internal Indications) – 2 bytes

Application Control

O campo de controle é definido da seguinte forma:

- FIR = 1 bit que indica se o fragmento da mensagem é o primeiro (1) ou se não é o primeiro fragmento mensagem (0).
- FIN = 1 bit que indica se o fragmento da mensagem é o último (1) ou se existem mais fragmentos da mensagem a seguir (0).
- CON = 1 bit que indica se a mensagem precisa ser confirmada por quem a recebeu (1) ou se a mensagem não precisa ser confirmada (0).
- SEQUENCE = número de sequência do fragmento. Varia de 0 a 15, em incrementos de 1 em 1.

No caso de mensagens não solicitadas (Unsolicited), varia de 16 a 31.

FC (Hexa)	FC (Decimal)	Função
00	0	Confirm
01	1	Read
02	2	Write
03	3	Select
04	4	Operate
05	5	Direct Operate
06	6	Direct Op, No Ack
07	7	Immediate Freeze
08	8	Immediate Freeze No Ack
09	9	Freeze and Clear
0A	10	Freeze and Clear No Ack
0B	11	Freeze with Time
0C	12	Freeze with Time No Ack
0D	13	Cold Restart
0E	14	Warm Restart
0F	15	Init Data to Defaults
10	16	Initialize Application
11	17	Start Application
12	18	Stop Application
13	19	Save Configuration
14	20	Enable Unsolicited Msgs
15	21	Disable Unsolicited Msgs
16	22	Assign Class
17	23	Delay Measurement
18 - 78	24 - 120	Reserved for future use
79 - 80	121 - 128	Reserved for testing only

Function Codes para Pedidos (Requests)

Function Code

Os códigos de função utilizados para requisições DNP3.

Function Code

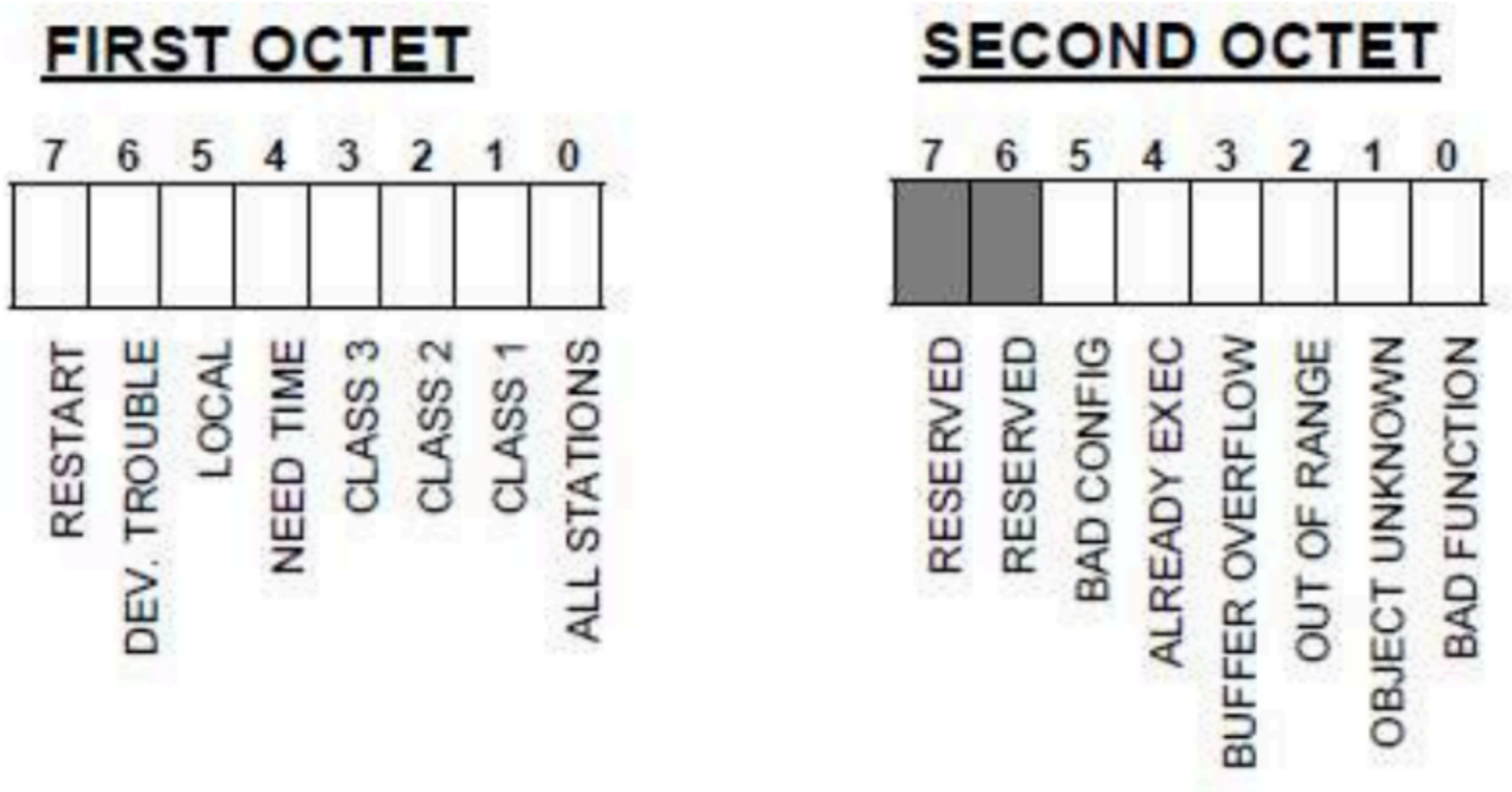
Os códigos de função utilizados para respostas DNP3.

FC (Hexa)	FC (Decimal)	Função
00	0	Confirm
81	129	Response
82	130	Unsolicited Response

Function Codes para Respostas (Responses)

Internal Indication

As indicações internas (IIN) são formadas por 16 bits:



Em que:

- ALL STATIONS = bit que indica recebimento de todas as estações.
- CLASS 1 = bit que indica que existem dados de classe 1 (Prioridade Alta) disponíveis.
- CLASS 2 = bit que indica que existem dados de classe 2 (Prioridade Média) disponíveis.
- CLASS 3 = bit que indica que existem dados de classe 3 (Prioridade Baixa) disponíveis.
- NEED TIME = bit que indica a necessidade de sincronização do relógio.
- LOCAL = bit que indica que uma ou mais saídas digitais não estão disponíveis.
- DEV. TROUBLE = bit que indica um defeito interno do IED.
- RESTART = bit que indica que o IED reiniciou.

- BAD FUNCTION = bit que indica que a função solicitada não existe.
- OBJECT UNKNOWN = bit que indica que o objeto solicitado não é reconhecido.
- OUT OF RANGE = bit que indica que os parâmetros dos campos de qualificador, de faixa ou de
- dados estão fora de faixa.
- BUFFER OVERFLOW = bit que indica que as filas de eventos ou de outras aplicações estão
- cheias.
- ALREADY EXEC = bit que indica que a operação solicitada ainda está sendo executada.
- BAD CONFIG = bit que indica que a configuração do IED está corrompida.

Object Header

O cabeçalho de objetos (object header) é composto de três campos:

- **Object** (2 bytes)

> Define o grupo de objetos e a variação dos mesmos.

- **Qualifier** (1 byte)

> Define o qualificador dos objetos e como o campo Range será interpretado.

- **Range** (0 a 8 bytes)

> Define a quantidade de pontos, endereço inicial e final, ou indentificadores dos pontos.

INDEX SIZE

0 – No Index, Packed
1 – 1 Octet Index
2 – 2 Octet Index
3 – 4 Octet Index
4 – 1 Octet Object Size
5 – 2 Octet Object Size
6 – 4 Octet Object Size

QUALIFIER CODE

0 – 8-Bit Start and Stop Indices
1 – 16-Bit Start and Stop Indices
2 – 32-Bit Start and Stop Indices
3 – 8-Bit Absolute Address Identifers
4 – 16-Bit Absolute Address Identifiers
5 – 32-Bit Absolute Address Identifers
6 – No Range Field (all)
7 – 8-Bit Quantity
8 – 16-Bit Quantity
9 – 32-Bit Quantity
11 – (0xB) Variable Array

INDEX SIZE (QUAL CODE = 11)

0 – Dataless Object; No Further Indexing
1 – 1 Octet Index or Identifier Size
2 – 2 Octet Index or Identifier Size
3 – 4 Octet Index or Identifier Size

Object Group

Uma lista de grupos de objetos, com suas variações e seus qualificadores pode ser visualizada no slide seguinte.

OBJECT			REQUEST (slave must parse)		RESPONSE (master must parse)	
Obj	Var	Description	Func Codes (dec)	Qual Codes (hex)	Func Codes	Qual Codes (hex)
1	0	Binary Input - All Variations	1, 22	00, 01, 06		
1	1	Binary Input	1	00, 01, 06	129, 130	00, 01
1	2	Binary Input with Status	1	00, 01 06	129, 130	00, 01
2	0	Binary Input Change - All Variations	1	06,07,08		
2	1	Binary Input Change without Time	1	06,07,08	129, 130	17, 28
2	2	Binary Input Change with Time	1	06,07,08	129, 130	17, 28
2	3	Binary Input Change with Relative Time	1	06,07,08	129, 130	17, 28
10	0	Binary Output - All Variations	1	00, 01, 06		
10	1	Binary Output				
10	2	Binary Output Status	1	00, 01, 06	129, 130	00, 01
12	0	Control Block - All Variations				
12	1	Control Relay Output Block	3, 4, 5, 6	17, 28	129	echo of request
12	2	Pattern Control Block	5, 6	17, 28	129	echo of request
12	3	Pattern Mask	5, 6	00, 01	129	echo of request
20	0	Binary Counter - All Variations	1, 7, 8, 9, 10, 22	00, 01, 06		
20	1	32-Bit Binary Counter	1	00, 01, 06	129, 130	00, 01
20	2	16-Bit Binary Counter	1	00, 01, 06	129, 130	00, 01
20	3	32-Bit Delta Counter	1	00, 01, 06	129, 130	00, 01
20	4	16-Bit Delta Counter	1	00, 01, 06	129, 130	00, 01

Modos de Operação do DNP3

- **Polled Mode:** Mestre solicita periodicamente dados ao escravo
- **Event Mode:** Escravo envia eventos quando ocorrem alterações significativas

Exemplo de Comunicação

- Simulação utilizando Python com visualização no Wireshark
- Simulação utilizando SCDADA LTS e IED SEL 751 com visualização no Wireshark

Exercícios

Exercício 1: Comunicação básica

Descreva o fluxo de comunicação entre um mestre e dois escravos em um sistema DNP3, considerando o modo de operação **Polled Mode.**

Exercício 2: Tipos de mensagens

Classifique as seguintes mensagens como leitura, comando ou resposta:

1. Mestre solicita leitura da temperatura
2. Escravo informa alteração de status de uma válvula
3. Mestre envia comando para fechar uma válvula

Exercício 3: Arquitetura

Explique a função de cada uma das camadas do protocolo DNP3.

Referências

- IEEE Std 1815 – Standard for Electric Power Systems Communications – Distributed Network Protocol (DNP3)
- Documentação oficial DNP Users Group
- Manuais de dispositivos SCADA

Obrigado!