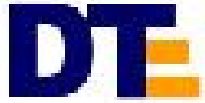


# BUSES DE CAMPO-FIELDBUS

---

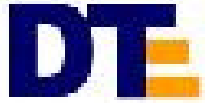
- Sistema de comunicaciones
  - Control distribuido
  - Sistema de cableado
  - Protocolo
  - Estandarización
  - Documentación
  - Recursos para desarrollo
  - Disponibilidad de documentación



# PROTOSCOLOS FIELDBUS

---

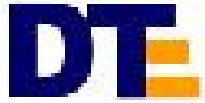
- CONTROL INDUSTRIAL
  - WORLDFIP
  - PROFIBUS
  - FIELBUS FOUNDATION
  - MODBUS
  - ASI
  - INTERBUS
  - HART
  - DEVICENET



# PROTOSCOLOS FIELDBUS

---

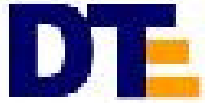
- VEHÍCULOS
  - CAN
- DOMÓTICA
  - X10
  - EIB-KNX
  - LONWORKS
- GENÉRICOS
  - TCP/IP-POE POWER OVER ETHERNET



# PROTOSCOLOS FIELDBUS

---

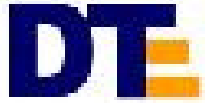
- INALÁMBRICOS
  - ISM
  - WIFI
  - WIMAX
  - ZIGBEE
- LÍNEA PORTADORA-POWERLINE
  - LONWORKS
  - EIB-KNX



# MODBUS-CARACTERÍSTICAS

---

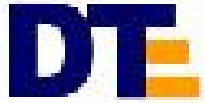
- FUNCIONAMIENTO SOBRE LÍNEA SERIE
- ESTRUCTURA MAESTRO-ESCLAVO
- NÚMERO MÁXIMO DE NODOS 247
- CABLEADO PAR TRENZADO
- ESTÁNDAR
- DOCUMENTACIÓN DISPONIBLE
- NO PRECISA HARDWARE ESPECIAL



# MODBUS-DOCUMENTOS

---

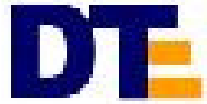
- MODBUS OVER SERIAL LINE
  - PROTOCOLO SOBRE LÍNEA SERIE
- MODBUS APPLICATION PROTOCOL
  - DESCRIPCIÓN DEL NIVEL DE APLICACIÓN
- MODBUS MESSAGIN IMPLEMENTATION GUIDE
  - PROTOCOLO SOBRE TCP/IP



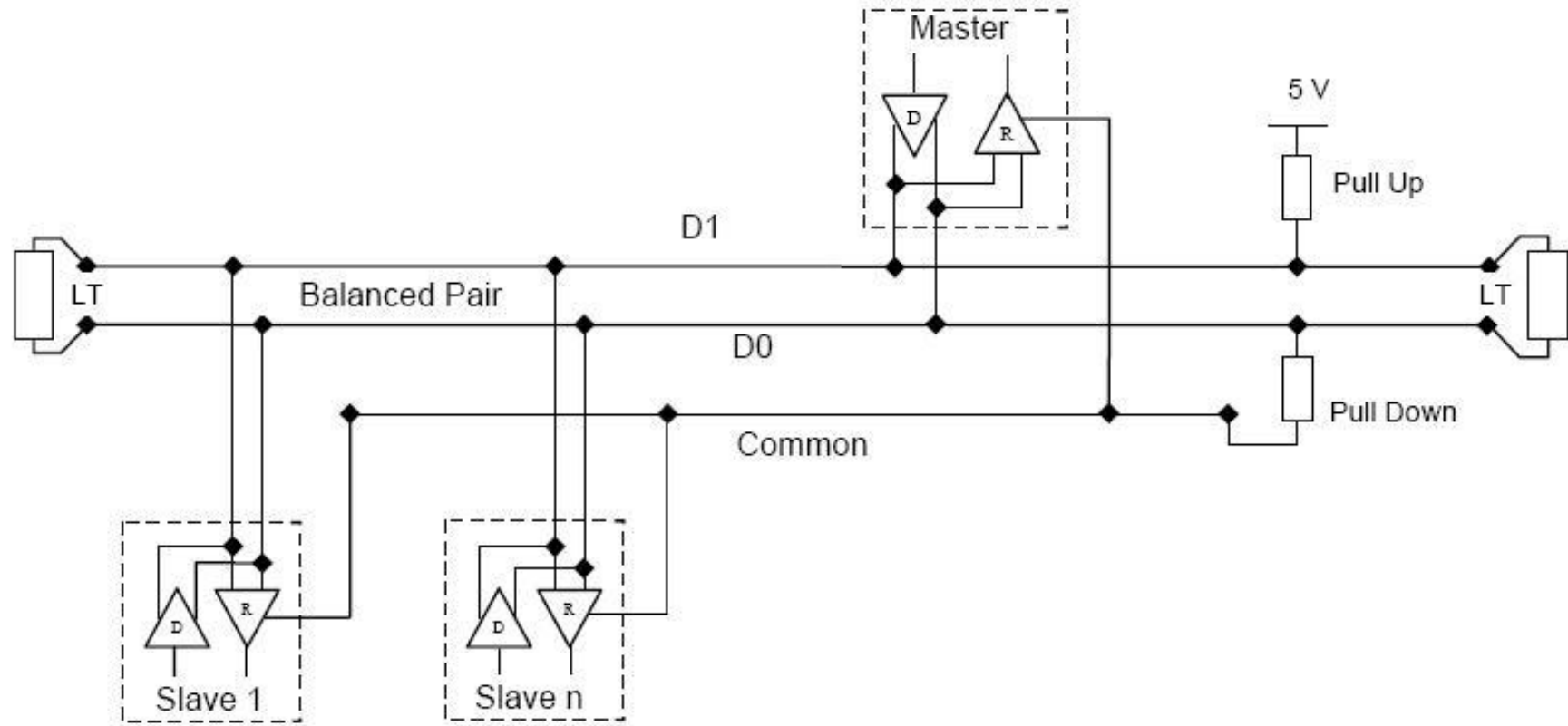
# MODBUS-SIMULADORES

---

- MTESTER
  - MAESTRO MODBUS
  - CLIENTE DE DATOS
- MOD-RSSIM
  - ESCLAVO MODBUS
  - SERVIDOR DE DATOS



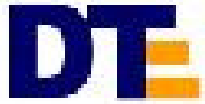
# MODBUS-CABLEADO 2W



General 2-Wire Topology

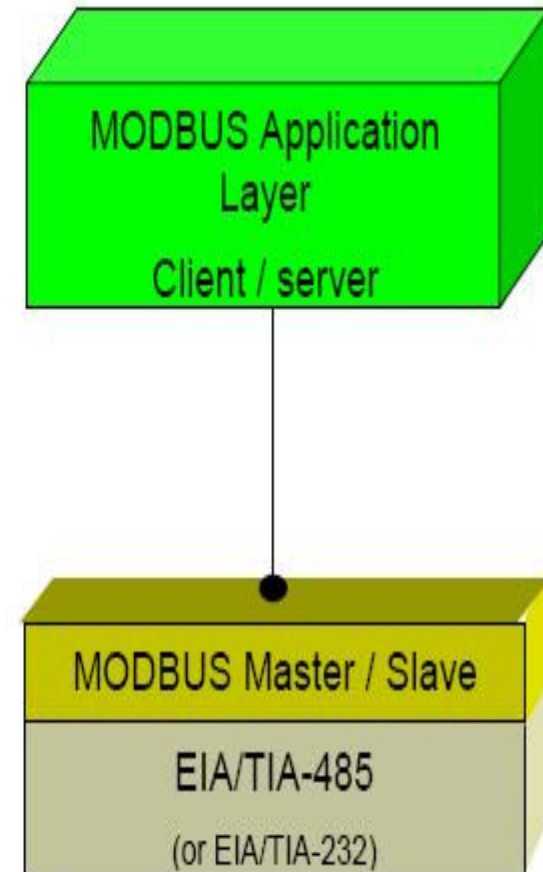




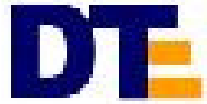


# PROTOCOLO MODBUS

Layer	ISO/OSI Model	
7	Application	MODBUS Application Protocol
6	Presentation	Empty
5	Session	Empty
4	Transport	Empty
3	Network	Empty
2	Data Link	MODBUS Serial Line Protocol
1	Physical	EIA/TIA-485 (or EIA/TIA-232)

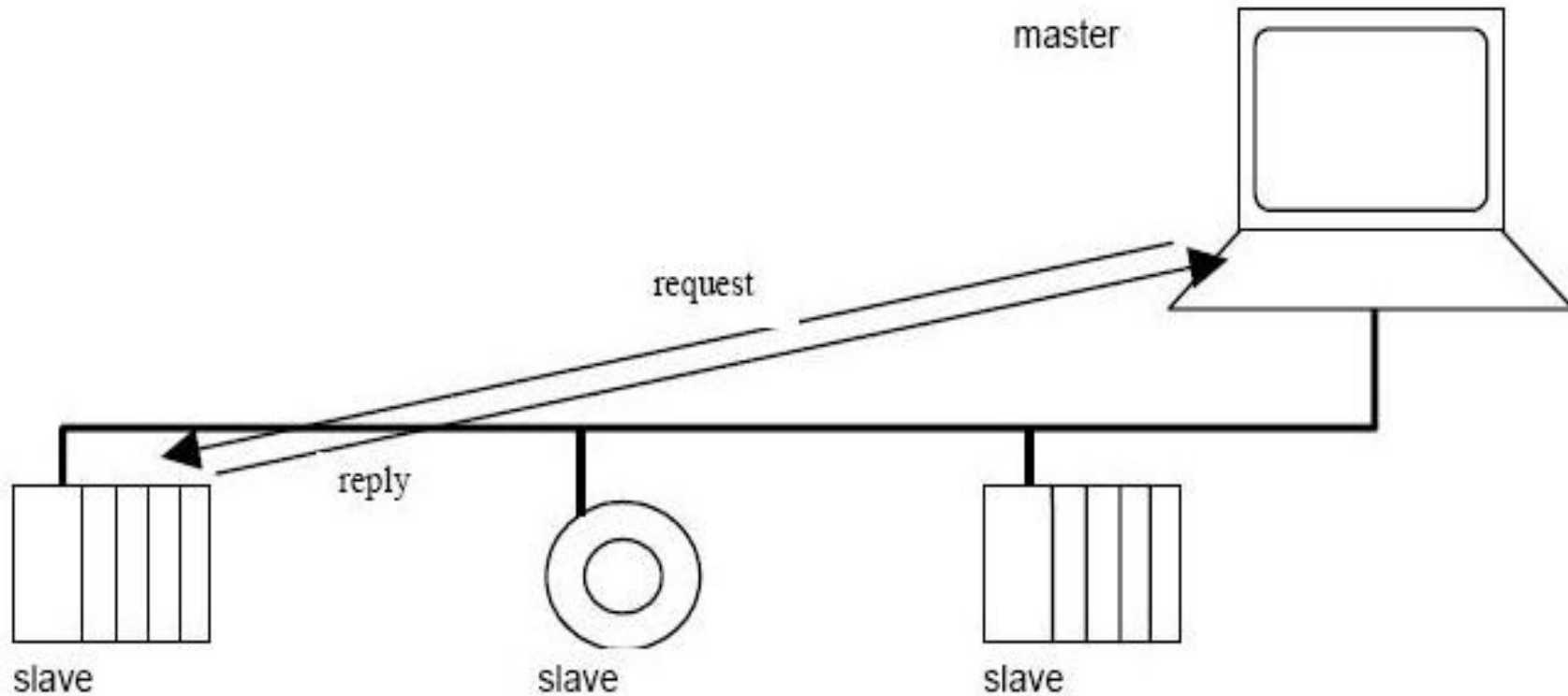


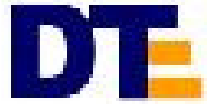
MODBUS Protocols and ISO/OSI Model



# MODBUS-UNICAST

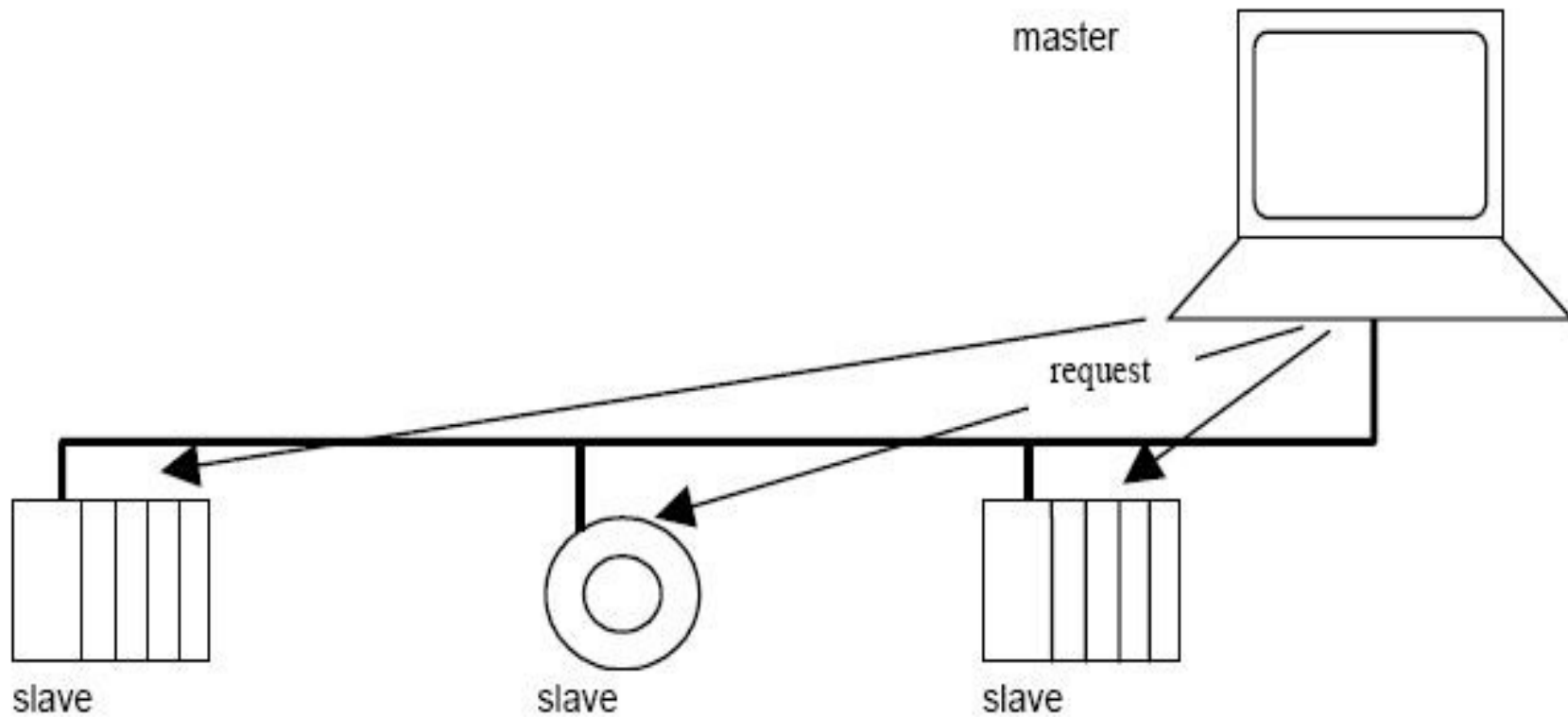
- DIRECCIONES 1..247

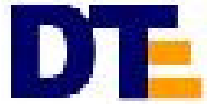




# MODBUS-MULTICAST

- DIRECCIÓN 0





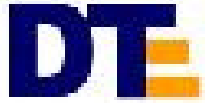
# MODBUS-PDU

## MODBUS SERIAL LINE PDU



**MODBUS PDU**

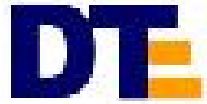
MODBUS frame over Serial Line



# MODBUS-FORMATOS

---

- ASCII
- RTU
  - CARÁCTER 11 BITS
  - 1 BIT DE INICIO
  - 8 BITS DE DATOS
  - PARIDAD PAR
  - 1 BIT DE PARO
  - OPCIONAL (NO PARIDAD 2 BITS DE PARO)



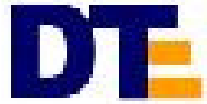
# MODBUS-CARÁCTER

With Parity Checking

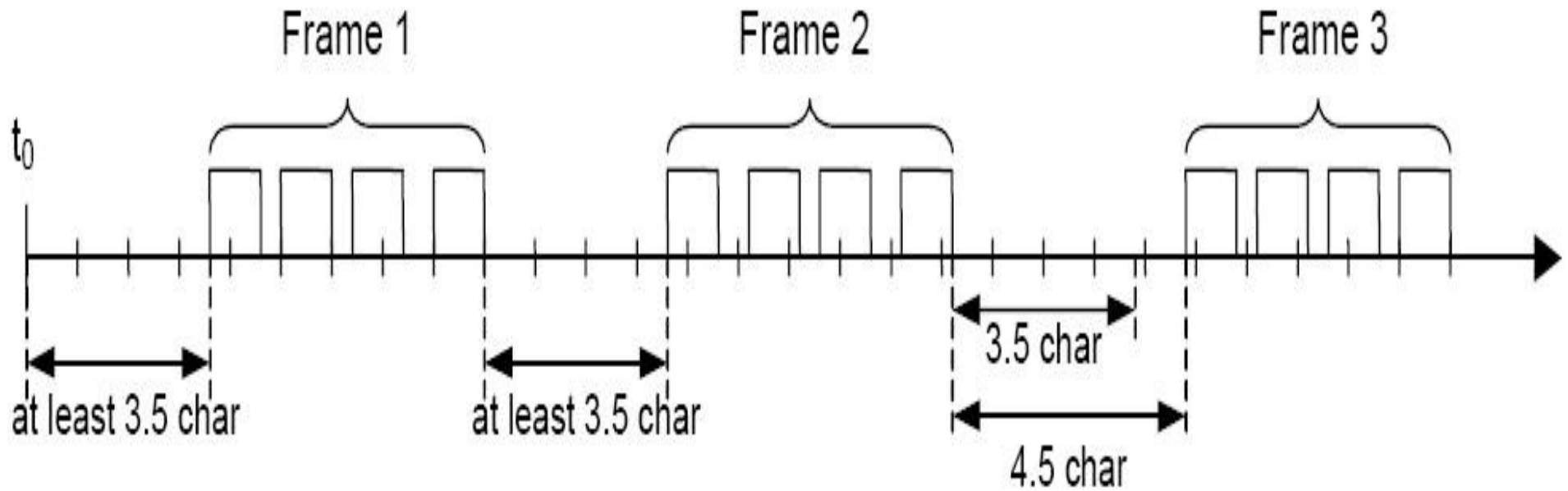
Start	1	2	3	4	5	6	7	8	Par	Stop
-------	---	---	---	---	---	---	---	---	-----	------

Without Parity Checking

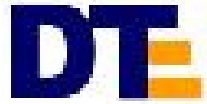
Start	1	2	3	4	5	6	7	8	Stop	Stop
-------	---	---	---	---	---	---	---	---	------	------



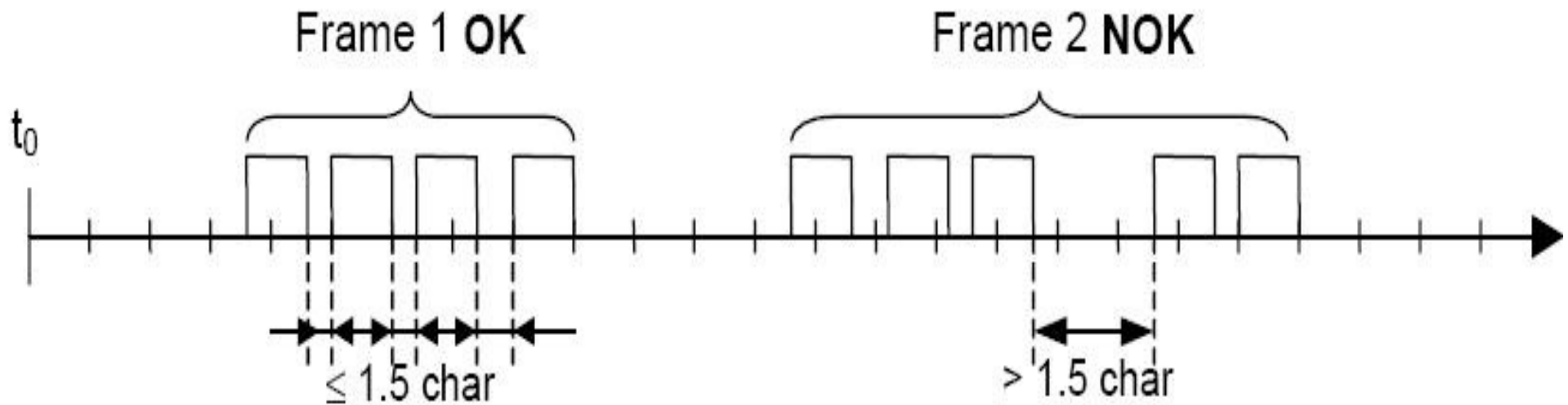
# MODBUS-TEMPORAL1

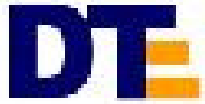






# MODBUS-TEMPORAL2

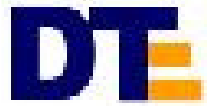




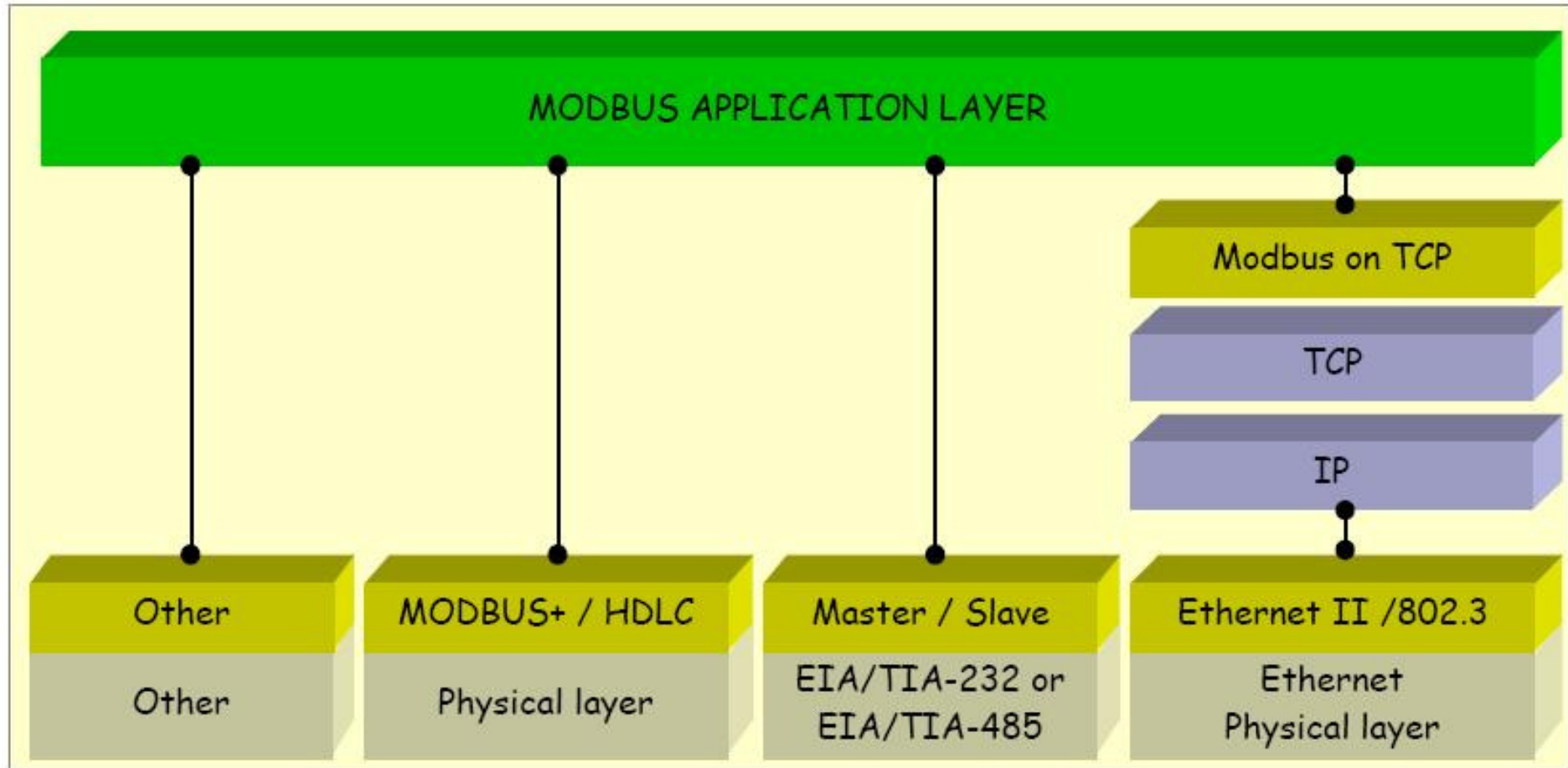
# MODBUS-TRAMA

Slave Address	Function Code	Data	CRC
1 byte	1 byte	0 up to 252 byte(s)	2 bytes CRC Low   CRC Hi

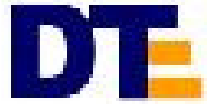
RTU Message Frame



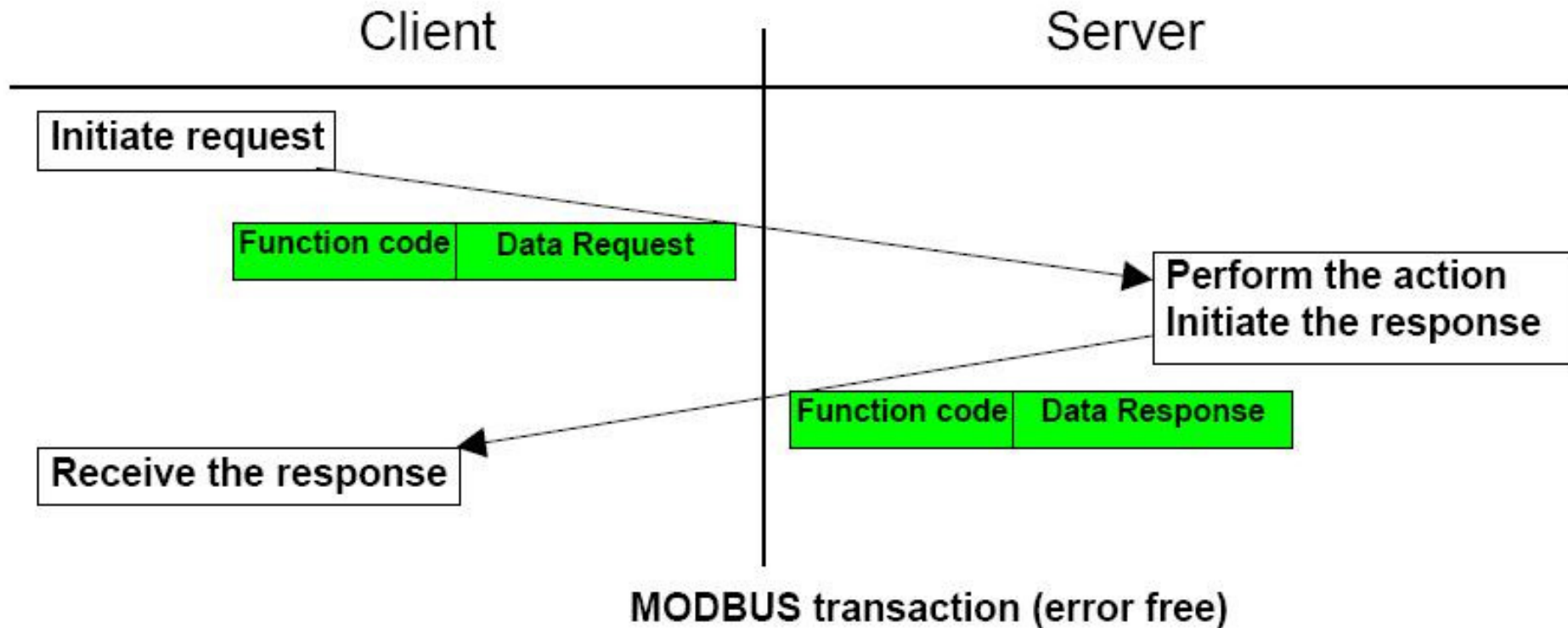
# MODBUS-OSI

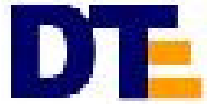


MODBUS communication stack

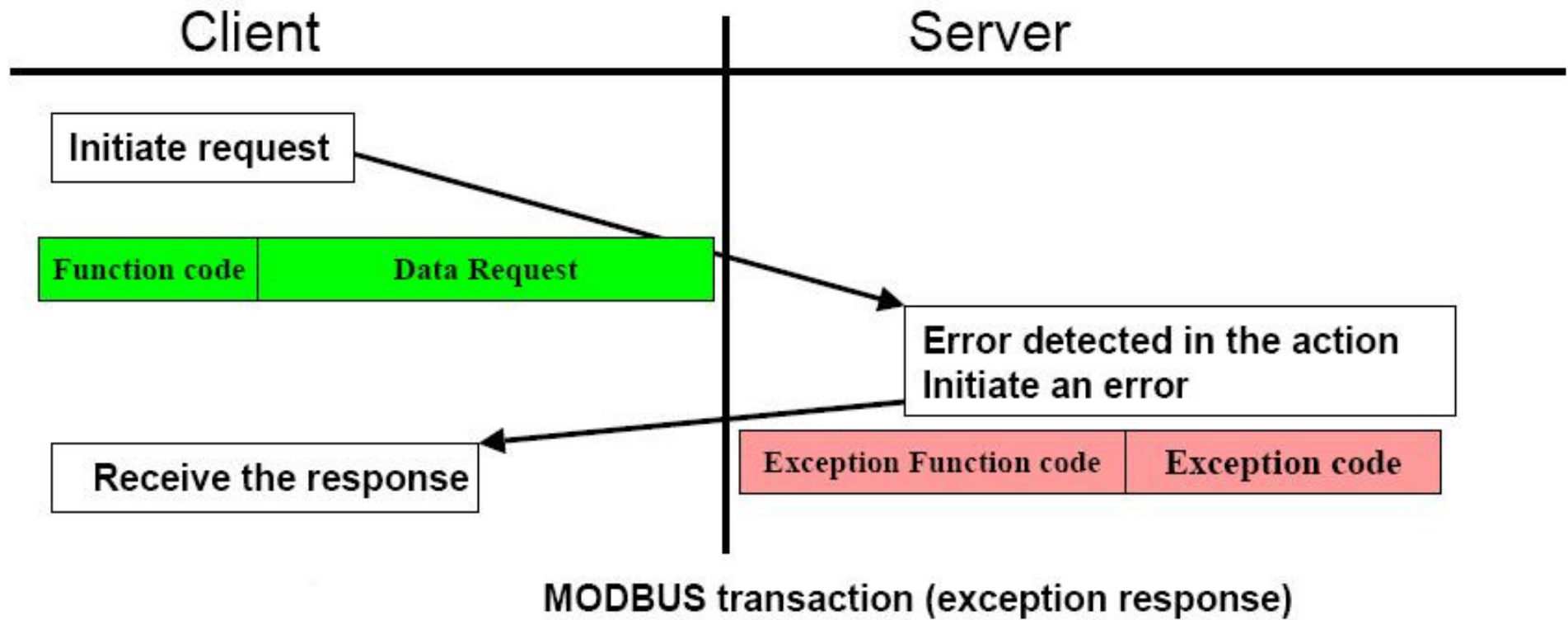


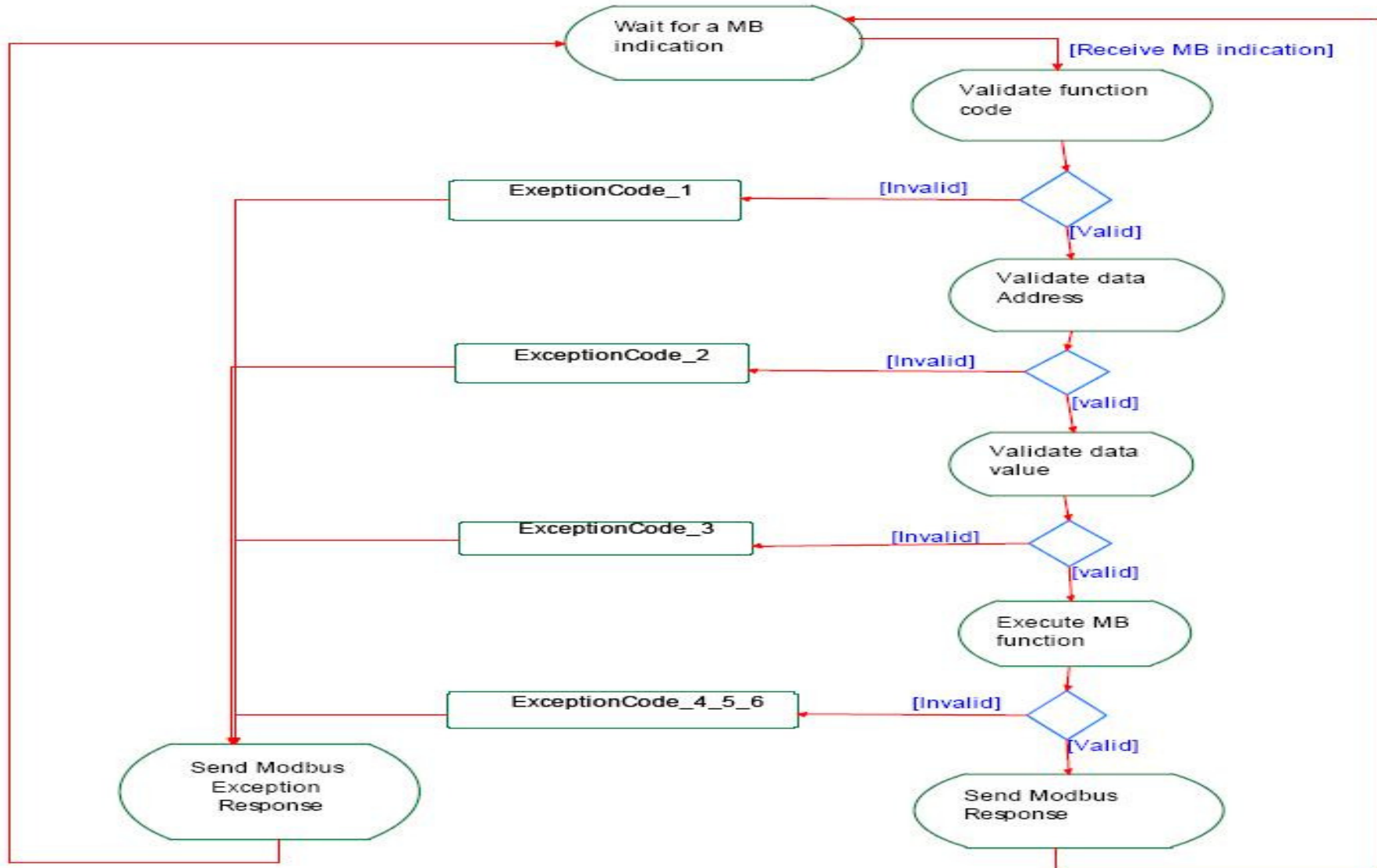
# MODBUS-TRANSACCIONES



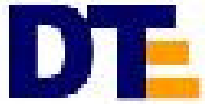


# MODBUS-EXCEPCIONES





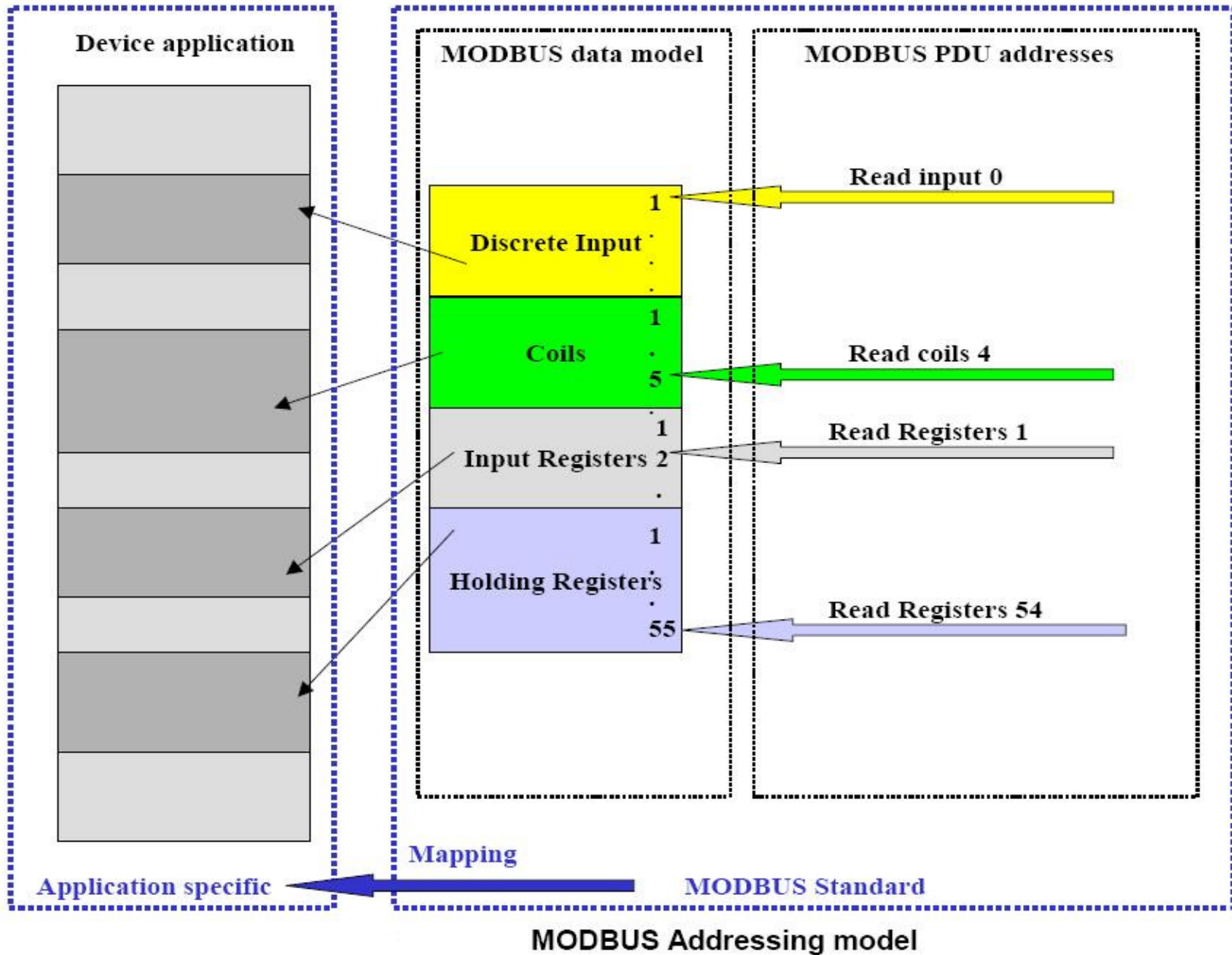
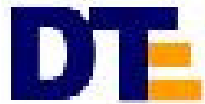
MODBUS Transaction state diagram



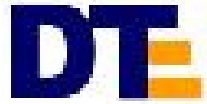
# MODBUS-MODELO DE DATOS

---

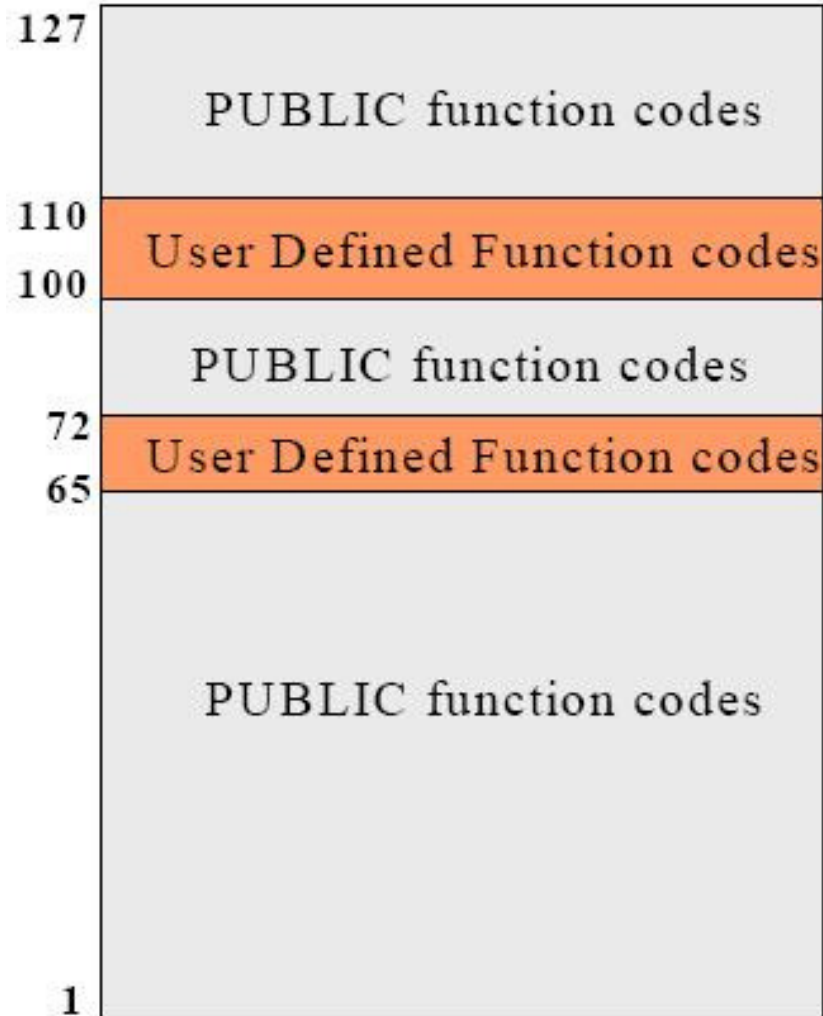
Primary tables	Object type	Type of access	Comments
Discretes Input	Single bit	Read-Only	This type of data can be provided by an I/O system.
Coils	Single bit	Read-Write	This type of data can be alterable by an application program.
Input Registers	16-bit word	Read-Only	This type of data can be provided by an I/O system
Holding Registers	16-bit word	Read-Write	This type of data can be alterable by an application program.

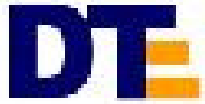






# MODBUS-FUNCIONES





# MODBUS-FUN1 READ COILS

## Request

Function code	1 Byte	<b>0x01</b>
Starting Address	2 Bytes	0x0000 to 0xFFFF
Quantity of coils	2 Bytes	1 to 2000 (0x7D0)

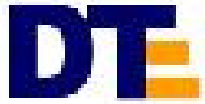
## Response

Function code	1 Byte	<b>0x01</b>
Byte count	1 Byte	<b>N*</b>
Coil Status	<b>n</b> Byte	n = N or N+1

\*N = Quantity of Outputs / 8, if the remainder is different of 0  $\Rightarrow$  N = N+1

## Error

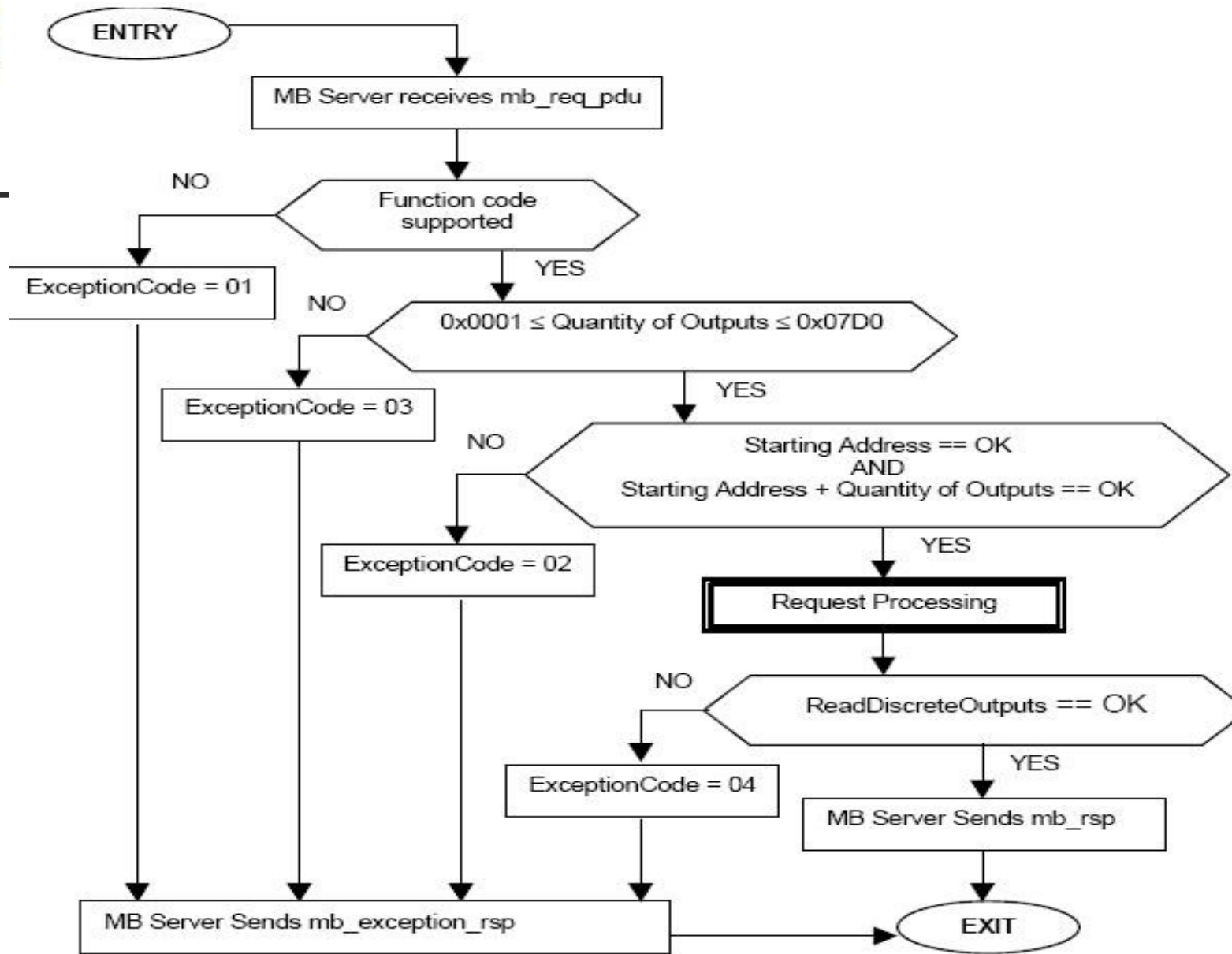
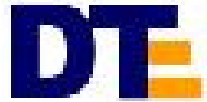
Function code	1 Byte	<b>Function code + 0x80</b>
Exception code	1 Byte	01 or 02 or 03 or 04



# MODBUS-READ COILS

read discrete outputs 20–38:

Request		Response	
<i>Field Name</i>	<i>(Hex)</i>	<i>Field Name</i>	<i>(Hex)</i>
Function	<b>01</b>	Function	<b>01</b>
Starting Address Hi	<b>00</b>	Byte Count	<b>03</b>
Starting Address Lo	<b>13</b>	Outputs status 27-20	<b>CD</b>
Quantity of Outputs Hi	<b>00</b>	Outputs status 35-28	<b>6B</b>
Quantity of Outputs Lo	<b>13</b>	Outputs status 38-36	<b>05</b>



Read Coils state diagram



# MODBUS-READ DISCRETE INPUT

## Request

Function code	1 Byte	0x02
Starting Address	2 Bytes	0x0000 to 0xFFFF
Quantity of Inputs	2 Bytes	1 to 2000 (0x7D0)

## Response

Function code	1 Byte	0x02
Byte count	1 Byte	N*
Input Status	N* x 1 Byte	

\*N = Quantity of Inputs / 8 if the remainder is different of 0  $\Rightarrow$  N = N+1



# MODBUS-READ HOLDING REG

## Request

Function code	1 Byte	<b>0x03</b>
Starting Address	2 Bytes	0x0000 to 0xFFFF
Quantity of Registers	2 Bytes	1 to 125 (0x7D)

## Response

Function code	1 Byte	<b>0x03</b>
Byte count	1 Byte	2 x <b>N*</b>
Register value	<b>N*</b> x 2 Bytes	

\*N = Quantity of Registers

## Error

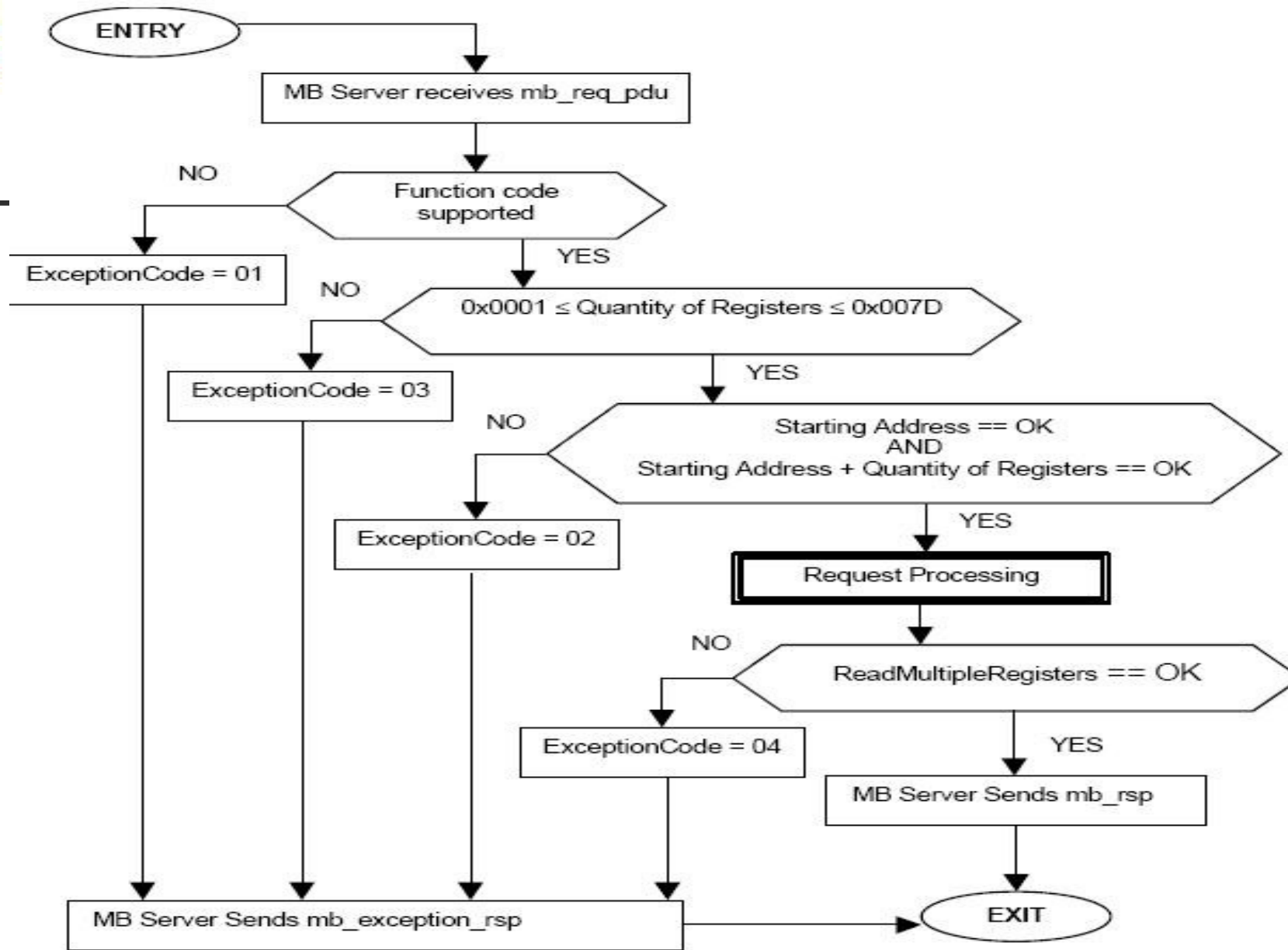
Error code	1 Byte	<b>0x83</b>
Exception code	1 Byte	01 or 02 or 03 or 04



# MODBUS-READ HOLDING REG

read registers 108 – 110:

Request		Response	
<i>Field Name</i>	<i>(Hex)</i>	<i>Field Name</i>	<i>(Hex)</i>
Function	<b>03</b>	Function	<b>03</b>
Starting Address Hi	<b>00</b>	Byte Count	<b>06</b>
Starting Address Lo	<b>6B</b>	Register value Hi (108)	<b>02</b>
No. of Registers Hi	<b>00</b>	Register value Lo (108)	<b>2B</b>
No. of Registers Lo	<b>03</b>	Register value Hi (109)	<b>00</b>
		Register value Lo (109)	<b>00</b>
		Register value Hi (110)	<b>00</b>
		Register value Lo (110)	<b>64</b>



**Read Holding Registers state diagram**





# MODBUS-READ INPUT REG

## Request

Function code	1 Byte	<b>0x04</b>
Starting Address	2 Bytes	0x0000 to 0xFFFF
Quantity of Input Registers	2 Bytes	0x0001 to 0x007D

## Response

Function code	1 Byte	<b>0x04</b>
Byte count	1 Byte	2 x N*
Input Registers	N* x 2 Bytes	

\*N = Quantity of Input Registers



# MODBUS-WRITE SINGLE COIL

## Request

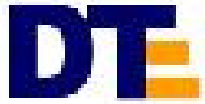
Function code	1 Byte	<b>0x05</b>
Output Address	2 Bytes	0x0000 to 0xFFFF
Output Value	2 Bytes	0x0000 or 0xFF00

## Response

Function code	1 Byte	<b>0x05</b>
Output Address	2 Bytes	0x0000 to 0xFFFF
Output Value	2 Bytes	0x0000 or 0xFF00

## Error

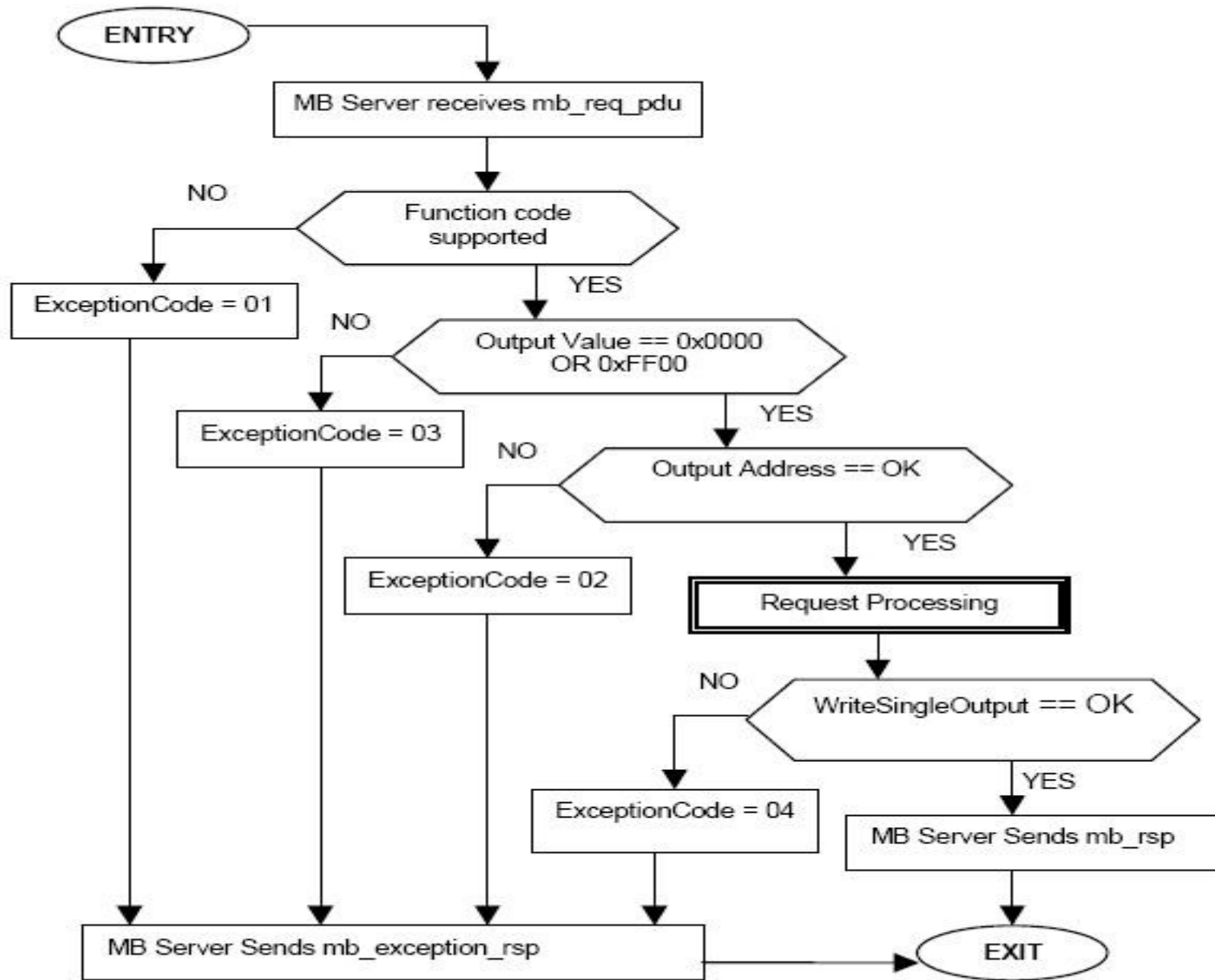
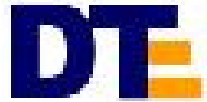
Error code	1 Byte	<b>0x85</b>
Exception code	1 Byte	01 or 02 or 03 or 04



# MODBUS-WRITE SINGLE COIL

write Coil 173 ON:

Request		Response	
<i>Field Name</i>	<i>(Hex)</i>	<i>Field Name</i>	<i>(Hex)</i>
Function	<b>05</b>	Function	<b>05</b>
Output Address Hi	<b>00</b>	Output Address Hi	<b>00</b>
Output Address Lo	<b>AC</b>	Output Address Lo	<b>AC</b>
Output Value Hi	<b>FF</b>	Output Value Hi	<b>FF</b>
Output Value Lo	<b>00</b>	Output Value Lo	<b>00</b>



**Write Single Output state diagram**



# MODBUS-WRITE SINGLE REG

## Request

Function code	1 Byte	<b>0x06</b>
Register Address	2 Bytes	0x0000 to 0xFFFF
Register Value	2 Bytes	0x0000 or 0xFFFF

## Response

Function code	1 Byte	<b>0x06</b>
Register Address	2 Bytes	0x0000 to 0xFFFF
Register Value	2 Bytes	0x0000 or 0xFFFF

## Error

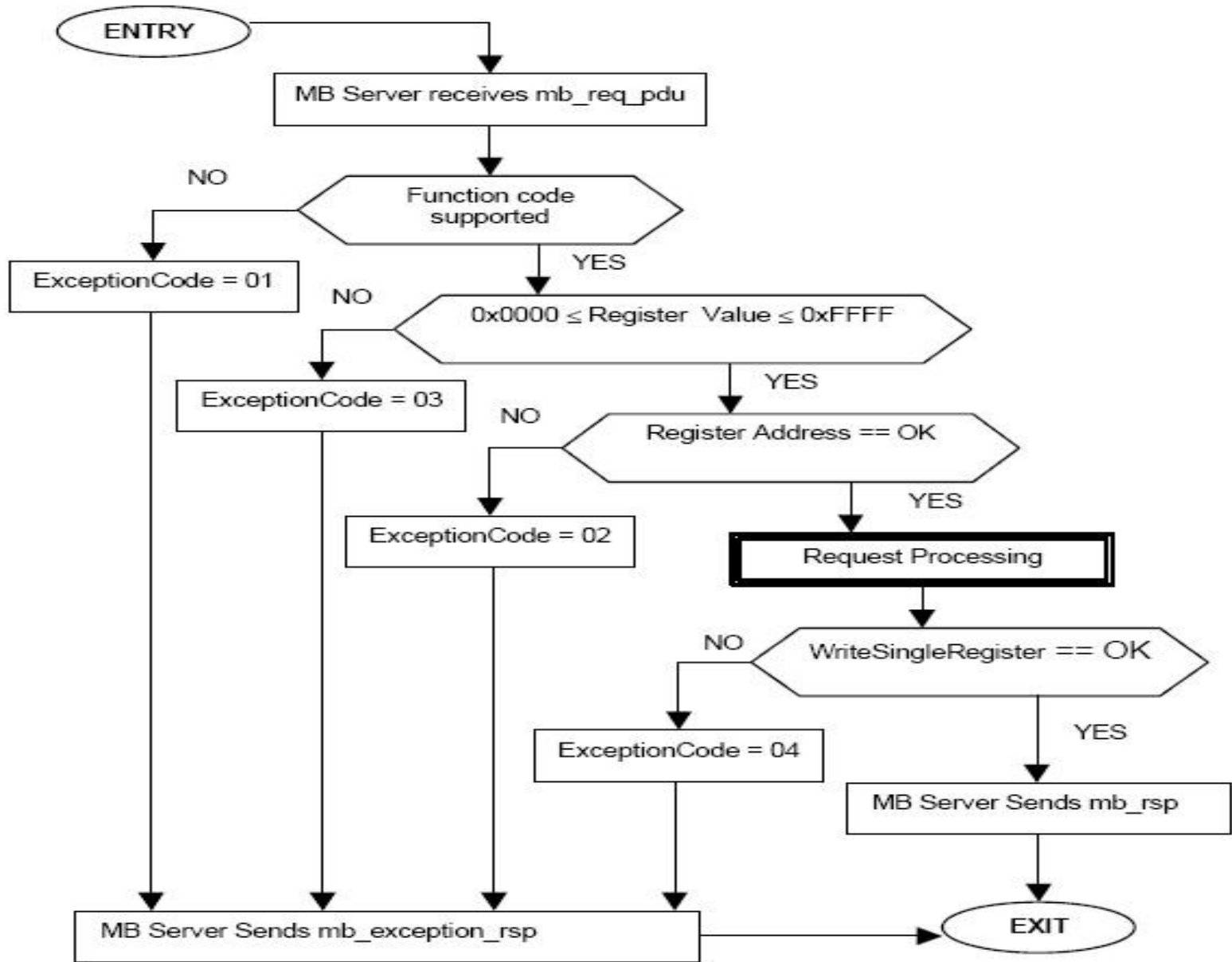
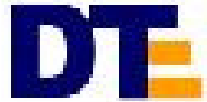
Error code	1 Byte	<b>0x86</b>
Exception code	1 Byte	01 or 02 or 03 or 04



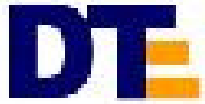
# MODBUS-WRITE SINGLE REG

write register 2 to 00 03 hex:

Request		Response	
<i>Field Name</i>	<i>(Hex)</i>	<i>Field Name</i>	<i>(Hex)</i>
Function	<b>06</b>	Function	<b>06</b>
Register Address Hi	<b>00</b>	Register Address Hi	<b>00</b>
Register Address Lo	<b>01</b>	Register Address Lo	<b>01</b>
Register Value Hi	<b>00</b>	Register Value Hi	<b>00</b>
Register Value Lo	<b>03</b>	Register Value Lo	<b>03</b>



Write Single Register state diagram



# MODBUS-WRITE MULT OUTPUTS

## Request PDU

Function code	1 Byte	<b>0x0F</b>
Starting Address	2 Bytes	0x0000 to 0xFFFF
Quantity of Outputs	2 Bytes	0x0001 to 0x07B0
Byte Count	1 Byte	<b>N*</b>
Outputs Value	<b>N* x 1 Byte</b>	

\*N = Quantity of Outputs / 8, if the remainder is different of 0  $\Rightarrow$  N = N+1

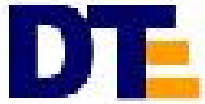
## Response PDU

Function code	1 Byte	<b>0x0F</b>
Starting Address	2 Bytes	0x0000 to 0xFFFF
Quantity of Outputs	2 Bytes	0x0001 to 0x07B0

## Error

Error code	1 Byte	<b>0x8F</b>
Exception code	1 Byte	01 or 02 or 03 or 04

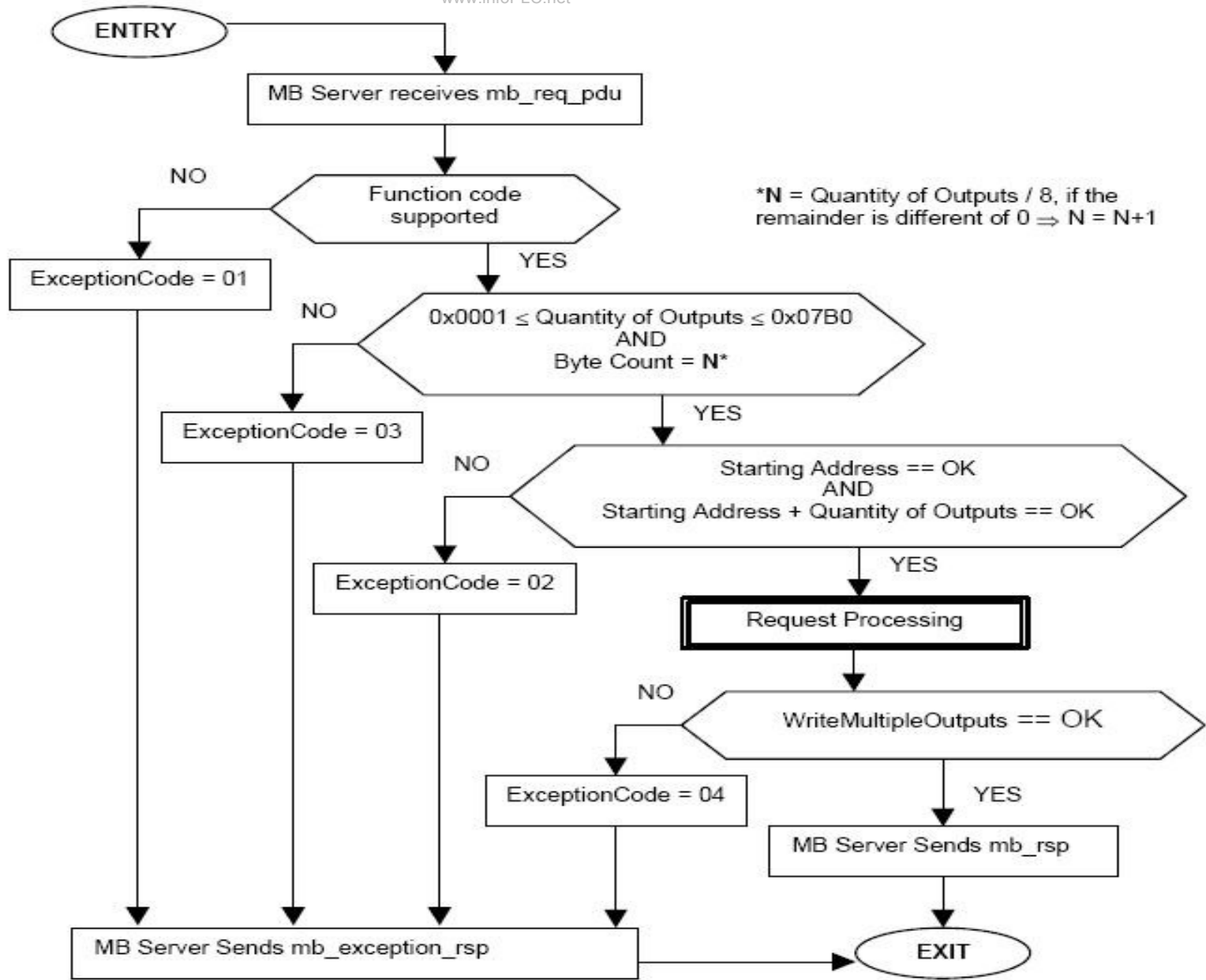




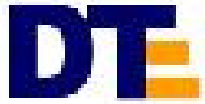
# MODBUS-WRITE MULT OUTPUTS

Bit:                    1    1    0    0    1    1    0    1    0    0    0    0    0    0    0    1  
 Output:                27   26   25   24   23   22   21   20   -   -   -   -   -   -   29   28

Request		Response	
Field Name	(Hex)	Field Name	(Hex)
Function	0F	Function	0F
Starting Address Hi	00	Starting Address Hi	00
Starting Address Lo	13	Starting Address Lo	13
Quantity of Outputs Hi	00	Quantity of Outputs Hi	00
Quantity of Outputs Lo	0A	Quantity of Outputs Lo	0A
Byte Count	02		
Outputs Value Hi	CD		
Outputs Value Lo	01		



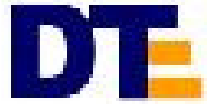
Write Multiple Outputs state diagram



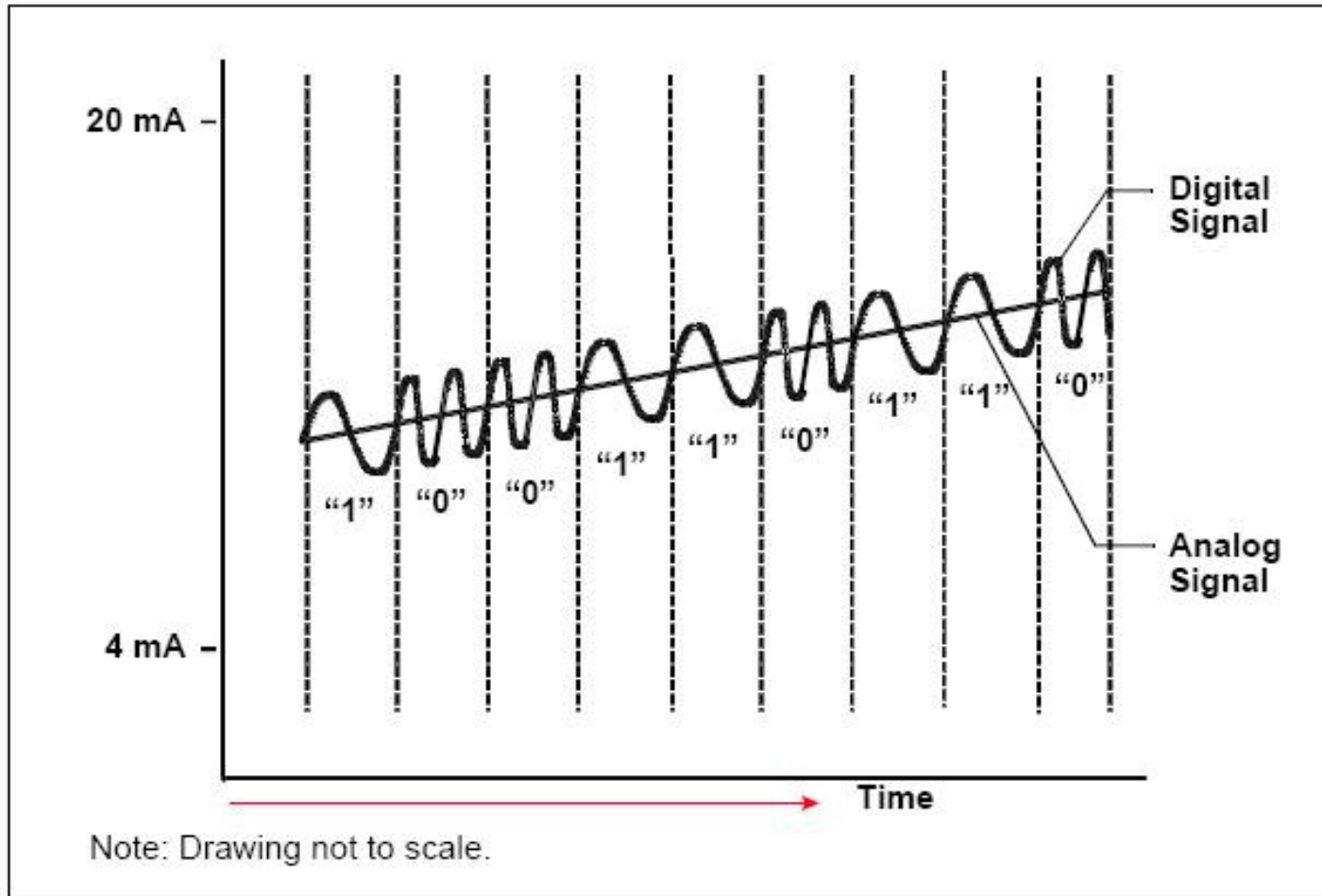
# HART

---

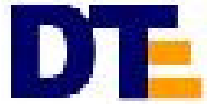
- Maestro-Esclavo
- Combina
  - 4-20mA analógico
  - Protocolo digital modulado FSK
    - 0 lógico 2200Hz
    - 1 lógico 1200 Hz
- Baja velocidad (3 mensajes/s)



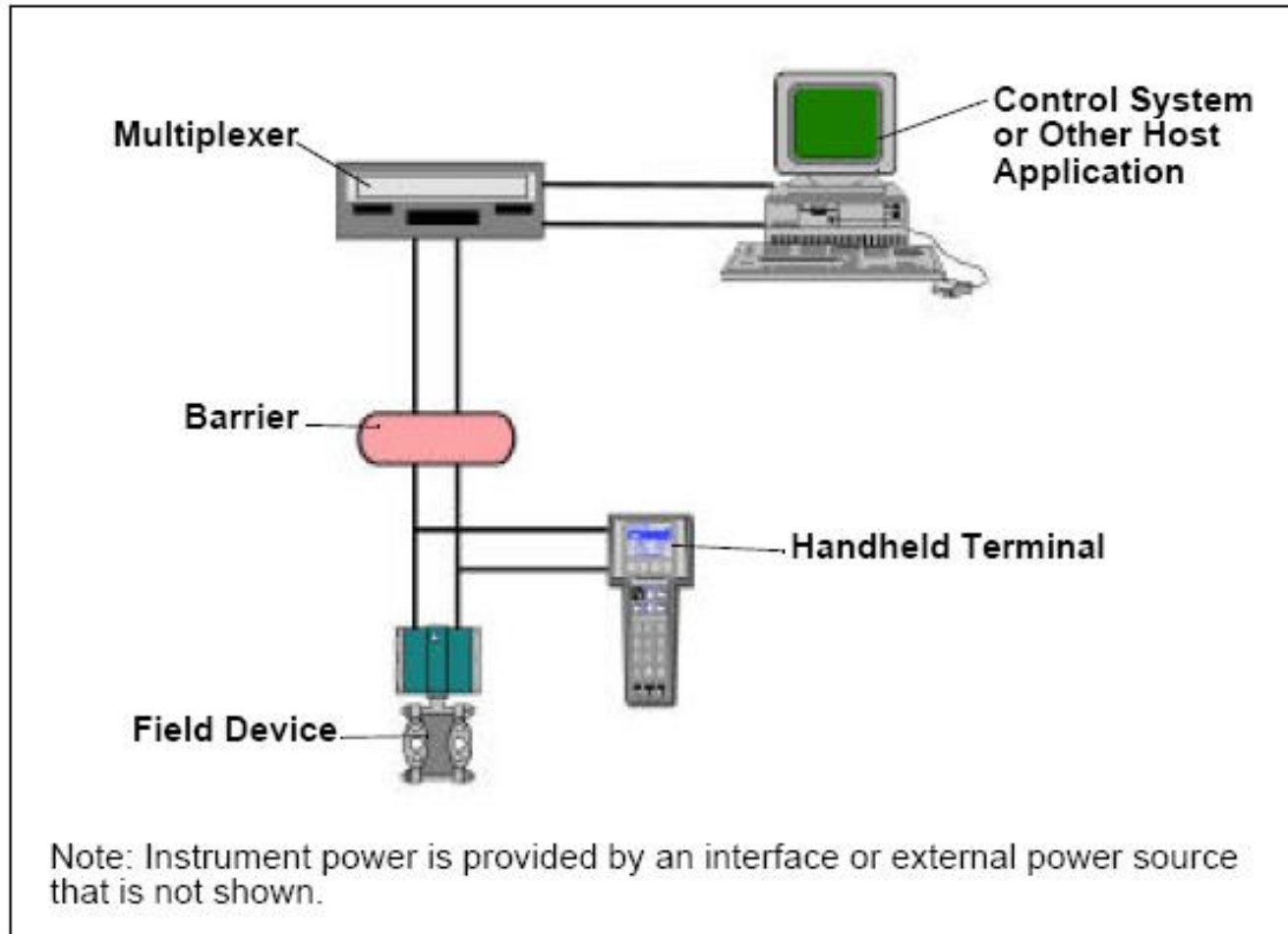
# HART DIAGRAMA I/T



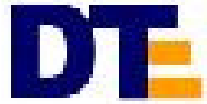
Simultaneous Analog and Digital Communication



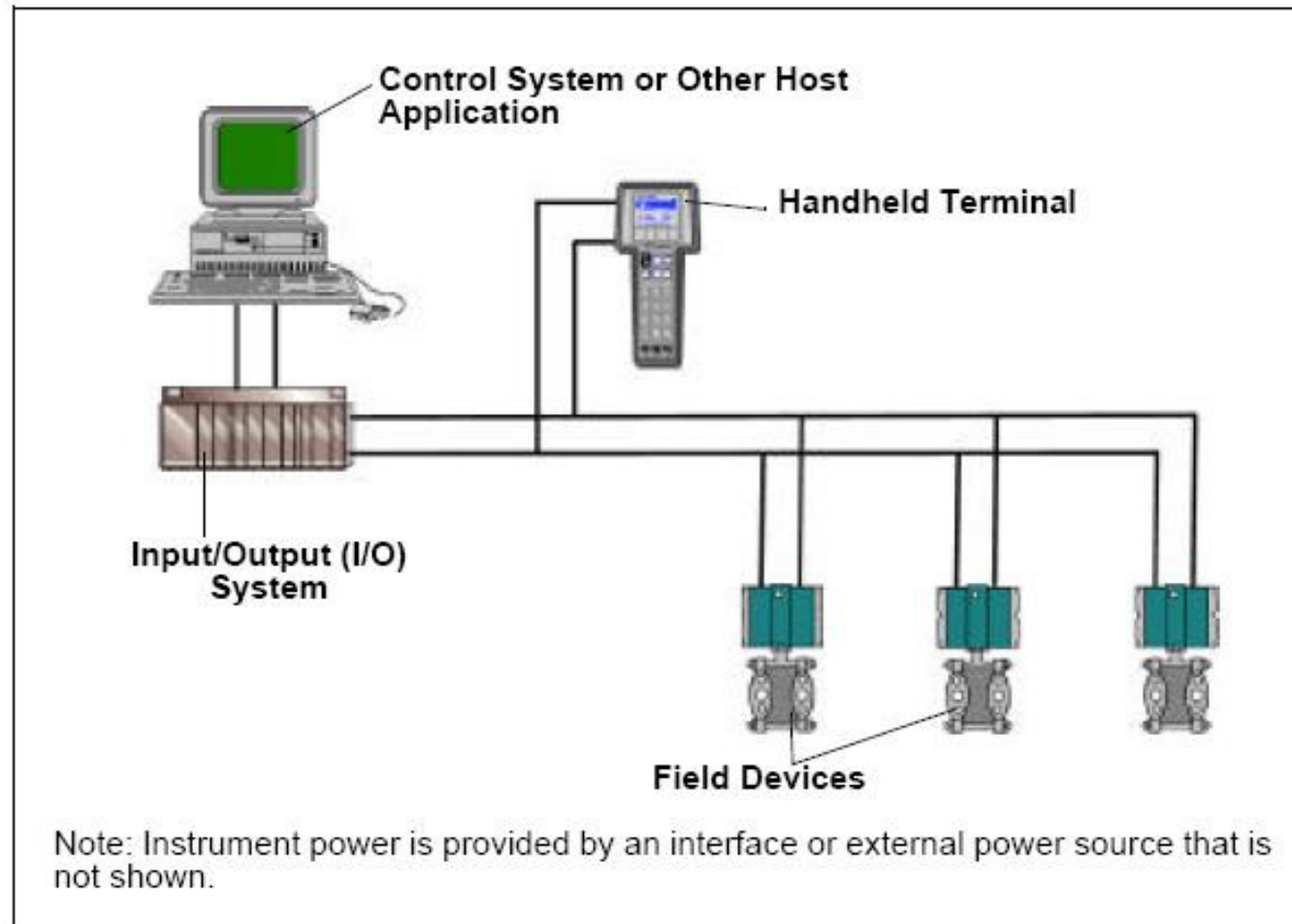
# HART PUNTO A PUNTO



Point-to-Point Mode of Operation



# HART MULTIPUNTO



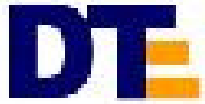
Multidrop Mode of Operation



# HART-COMANDOS

Universal Commands	Common Practice Commands	Device-Specific Commands
<ul style="list-style-type: none"> <li>• Read manufacturer and device type</li> <li>• Read primary variable (PV) and units</li> <li>• Read current output and percent of range</li> <li>• Read up to four predefined dynamic variables</li> <li>• Read or write eight-character tag, 16-character descriptor, date</li> <li>• Read or write 32-character message</li> <li>• Read device range values, units, and damping time constant</li> <li>• Read or write final assembly number</li> <li>• Write polling address</li> </ul>	<ul style="list-style-type: none"> <li>• Read selection of up to four dynamic variables</li> <li>• Write damping time constant</li> <li>• Write device range values</li> <li>• Calibrate (set zero, set span)</li> <li>• Set fixed output current</li> <li>• Perform self-test</li> <li>• Perform master reset</li> <li>• Trim PV zero</li> <li>• Write PV unit</li> <li>• Trim DAC zero and gain</li> <li>• Write transfer function (square root/linear)</li> <li>• Write sensor serial number</li> <li>• Read or write dynamic variable assignments</li> </ul>	<ul style="list-style-type: none"> <li>• Read or write low-flow cut-off</li> <li>• Start, stop, or clear totalizer</li> <li>• Read or write density calibration factor</li> <li>• Choose PV (mass, flow, or density)</li> <li>• Read or write materials or construction information</li> <li>• Trim sensor calibration</li> <li>• PID enable</li> <li>• Write PID setpoint</li> <li>• Valve characterization</li> <li>• Valve setpoint</li> <li>• Travel limits</li> <li>• User units</li> <li>• Local display information</li> </ul>

HART Commands

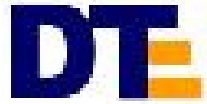


# INTERBUS

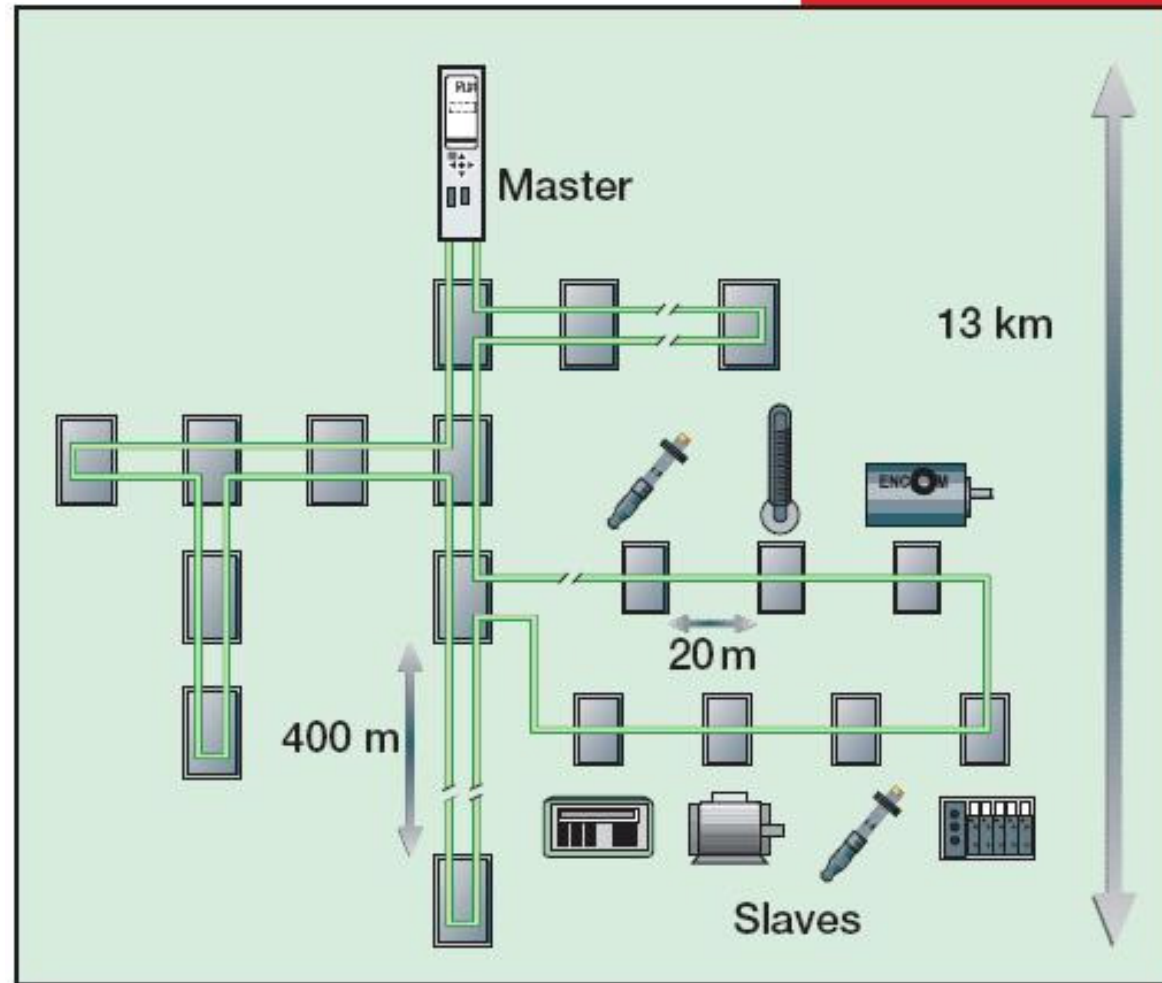
---

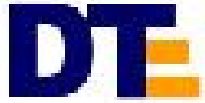
- Topología anillo activo
- Maestro-Esclavo con longitud fija msg.
- Todos los nodos con repetidor
- Velocidad 500 Kbps
- Longitud bus
  - 400 m entre dispositivos
  - 13 Km longitud total
- Aplicación: Sensor/actuador, control ind





# INTERBUS TOPOLOGÍA

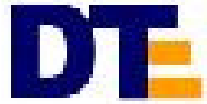




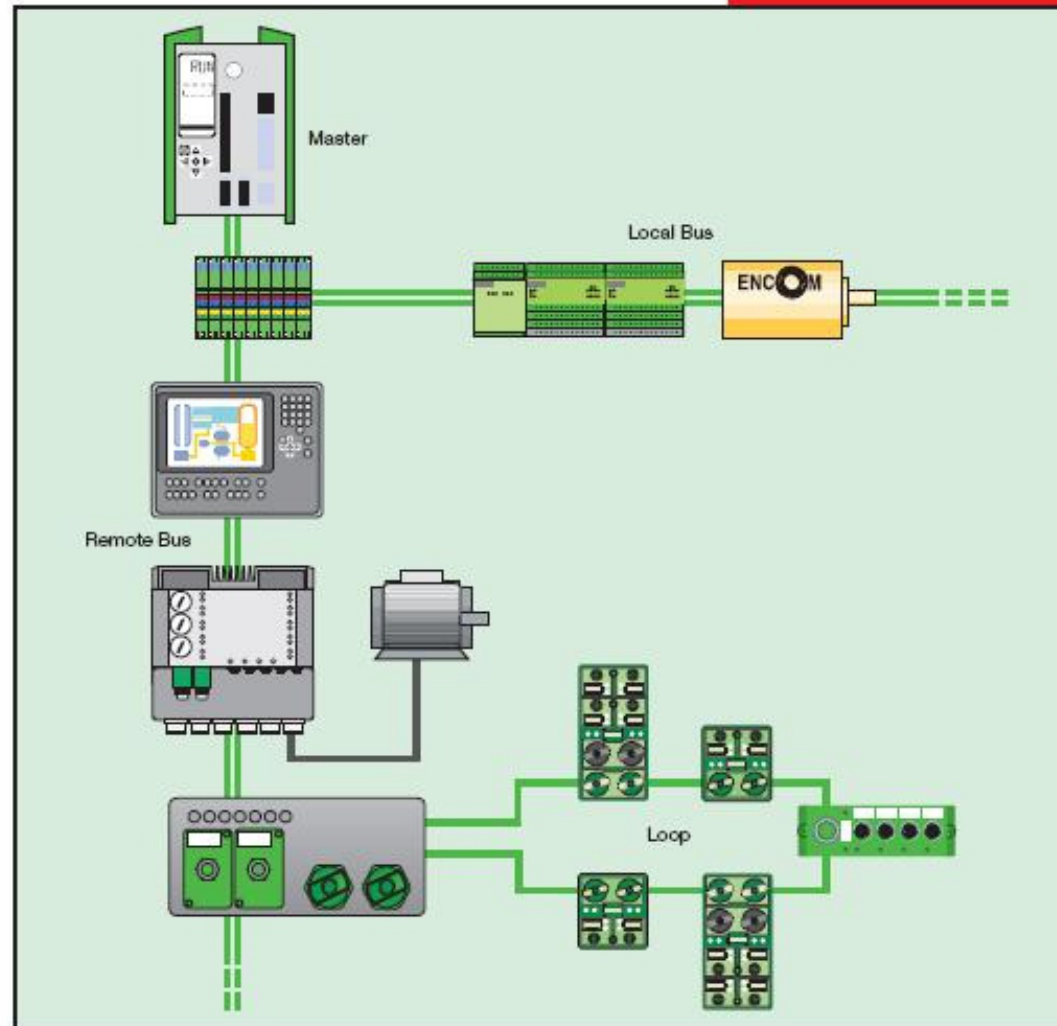
# INTERBUS LAZO

---

- Distancia entre dispositivos
  - Mínima 20 cm
  - Máxima 20 m
- Distancia total 200 m
- Máximo 63 dispositivos
- Alimentación y comunicación mismo cable
- Potencia máxima 1.8A



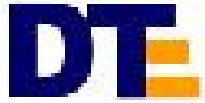
# INTERBUS LAZO



Francisco Simón Muñiz

*Individual components of an INTERBUS network*

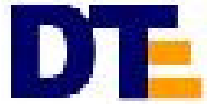
3.51



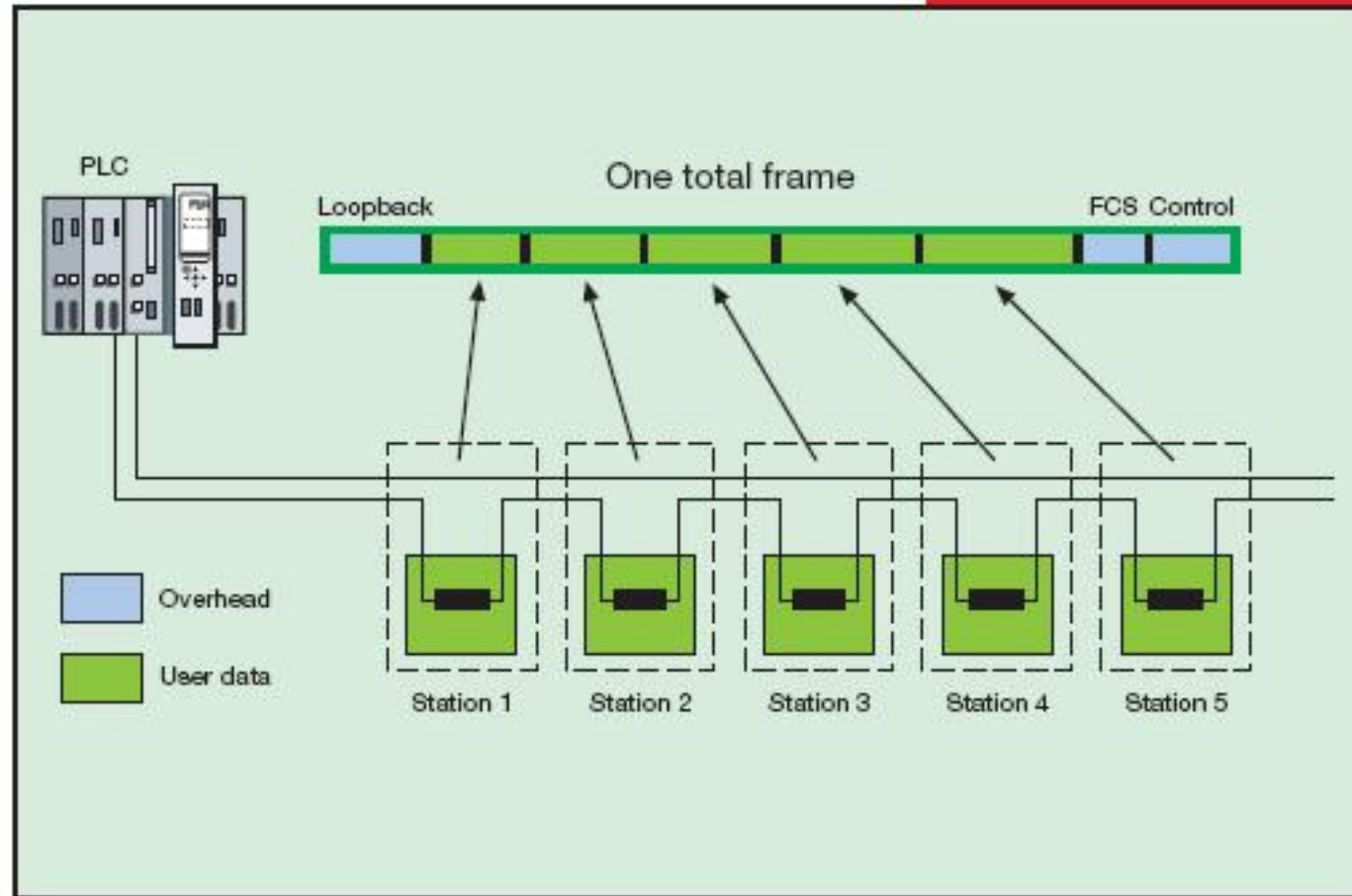
# INTERBUS TRANSMISIÓN

---

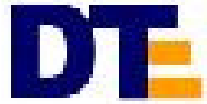
- Maestro-Esclavo
- Una sola trama en comunicación
- Un espacio en la trama para cada dispositivo
  - Variables de salida – escribe el maestro
  - Variables entrada – escribe el dispositivo
- Baja sobrecarga del protocolo



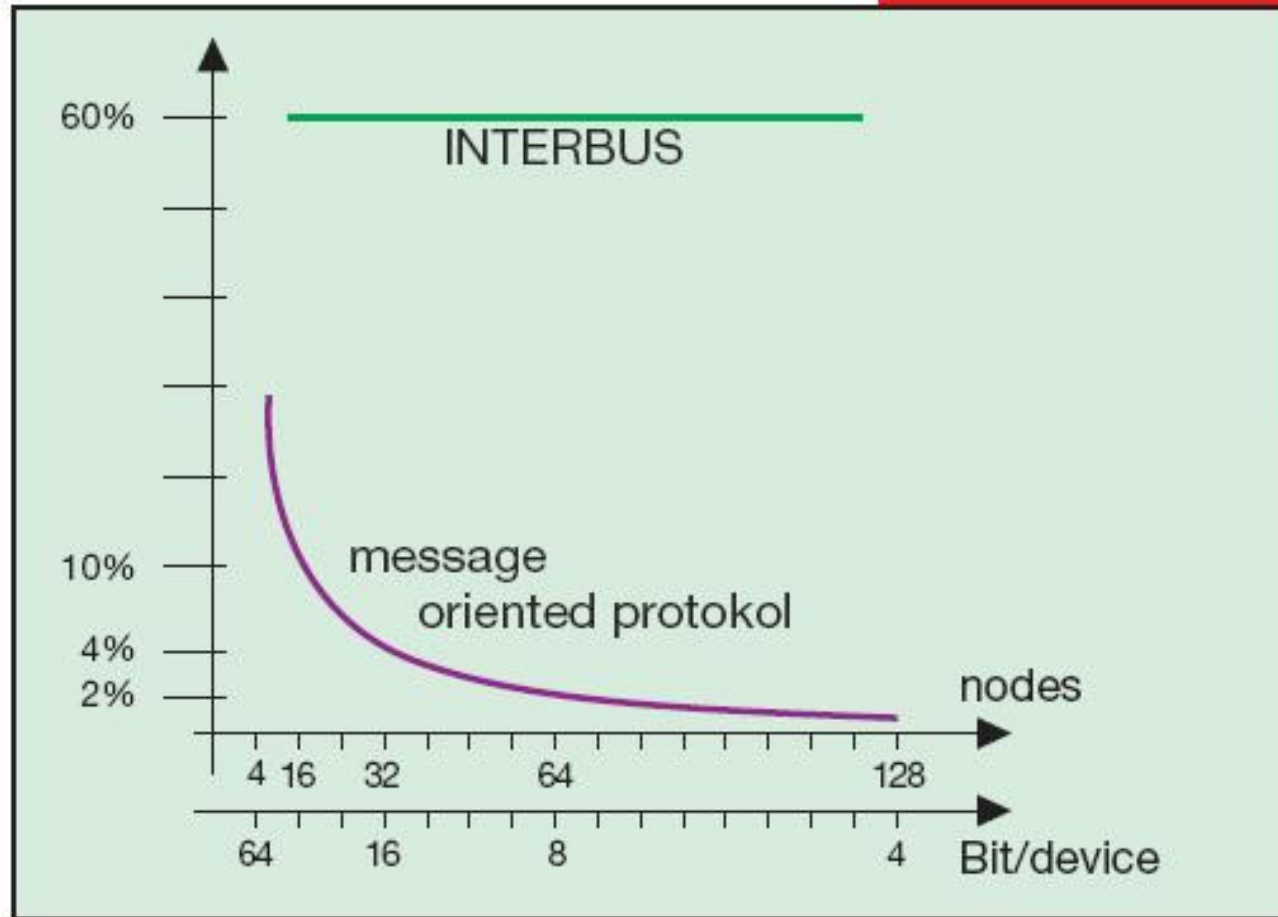
# INTERBUS TRANSMISIÓN



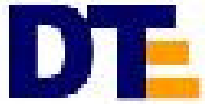
*Physical transmission method - summation frame method*



# INTERBUS EFICIENCIA



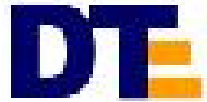
*Efficiency of different transmission methods*



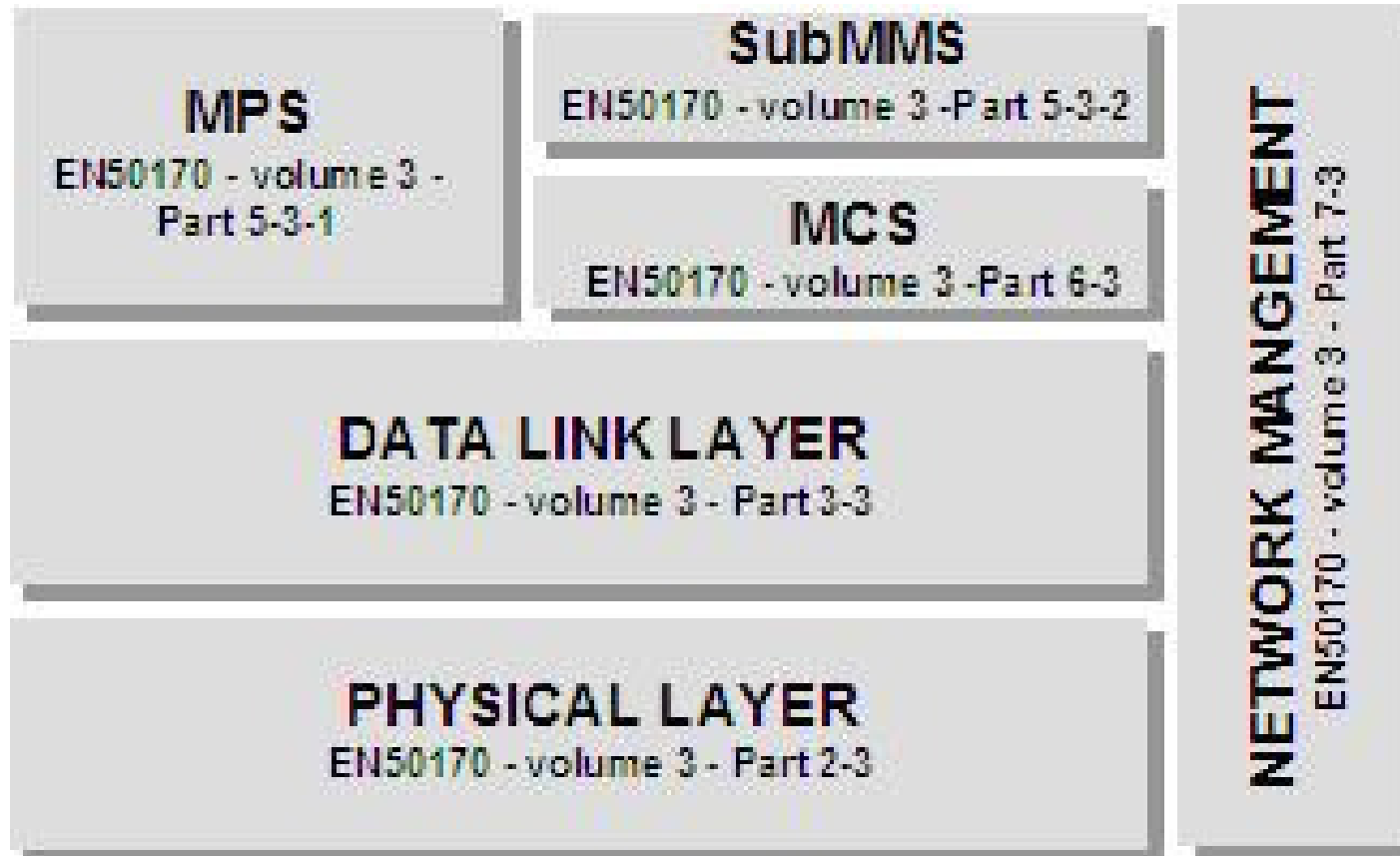
# WORLDFIP

---

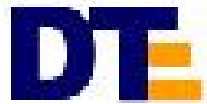
- Protocolo Francés
- Estándar EN 50170
- Velocidad
  - 31.25 kb/s baja velocidad
  - 1 Mb/s (2.5 Mb/s) alta velocidad
- Direccionamiento por variables
  - Un identificador 16 bits por variable
  - Hasta 65536 variables



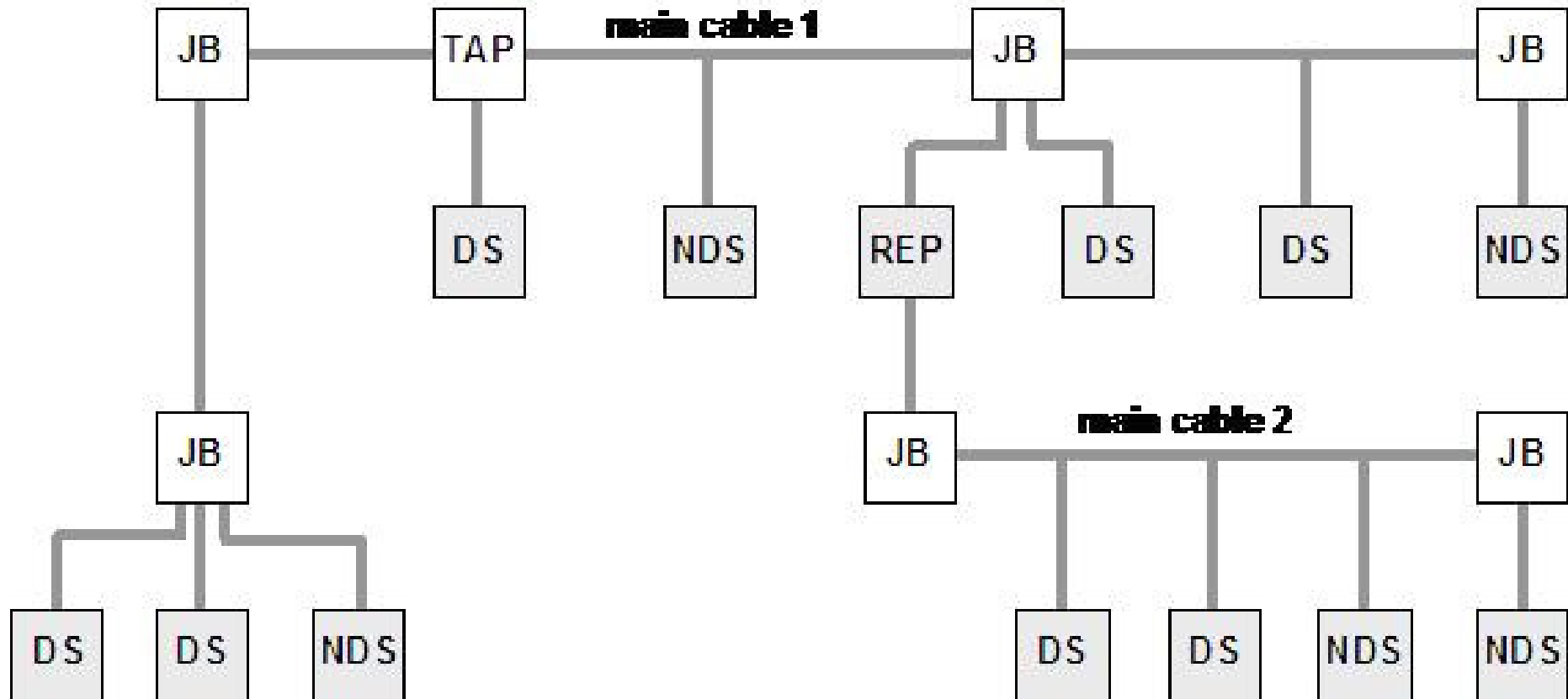
# WORLDFIP ESTÁNDAR

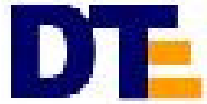




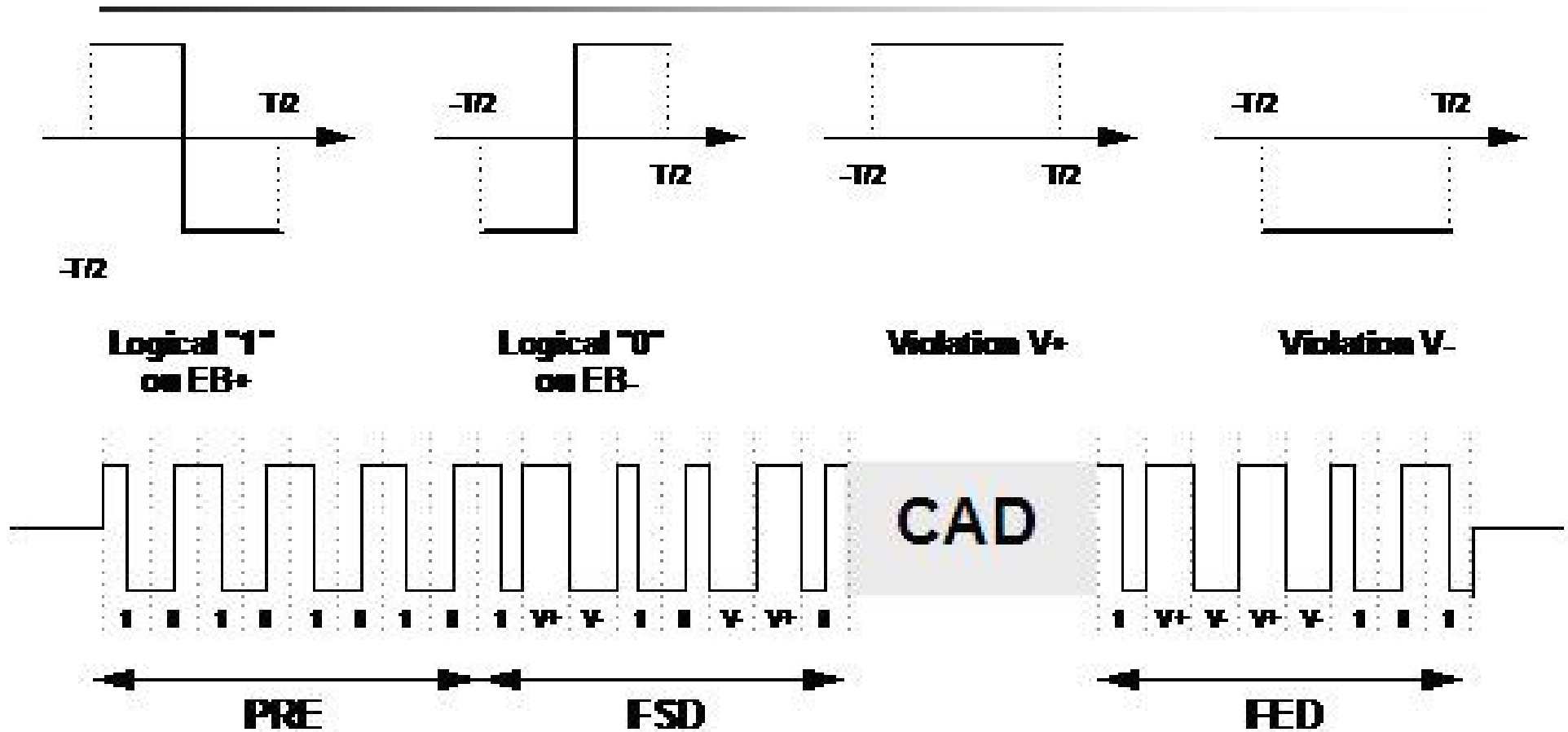


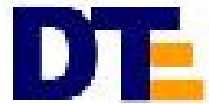
# WORLDFIP NIVEL FÍSICO



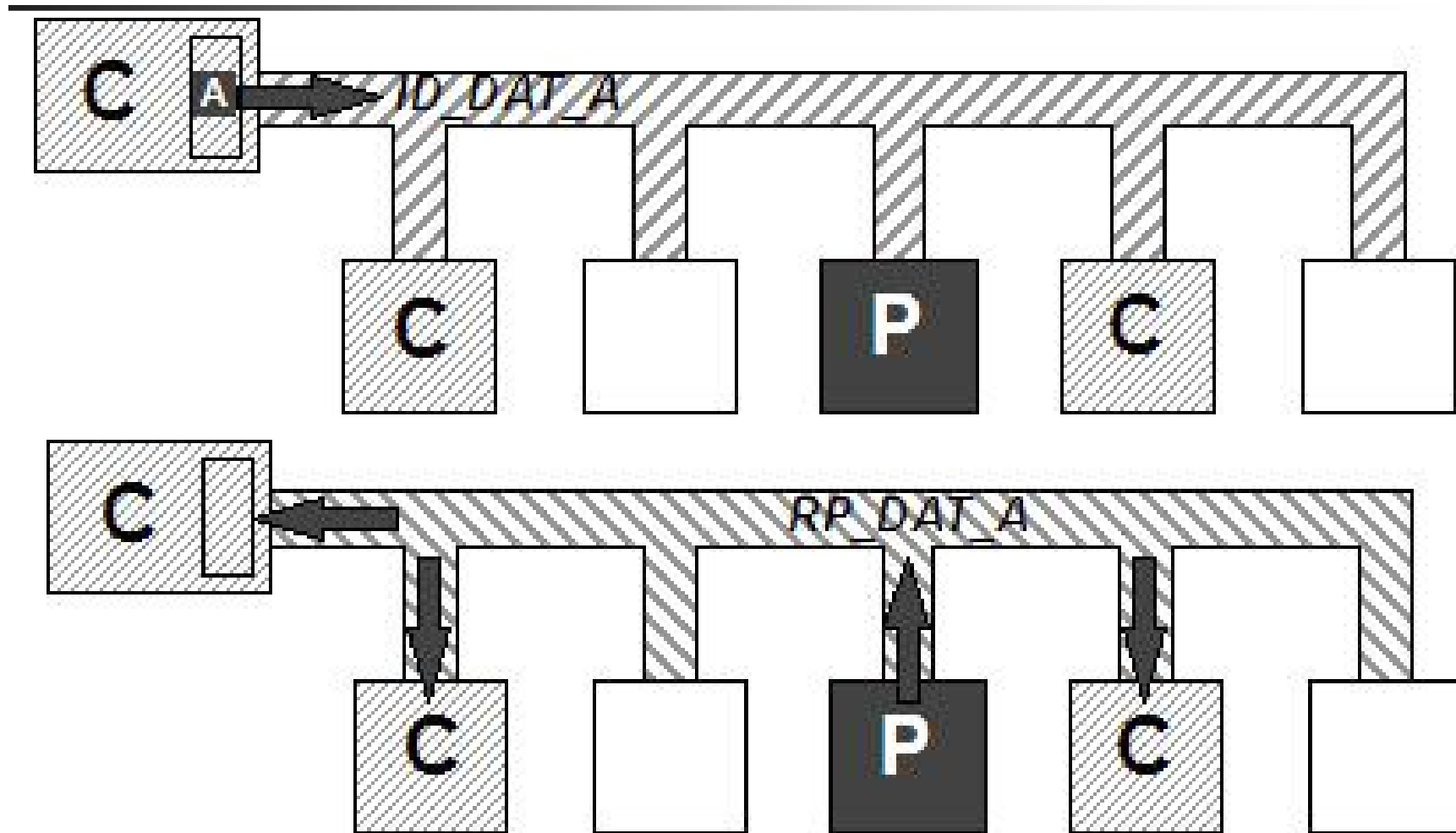


# WORLDIFIP CODIFICACIÓN

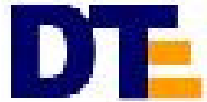




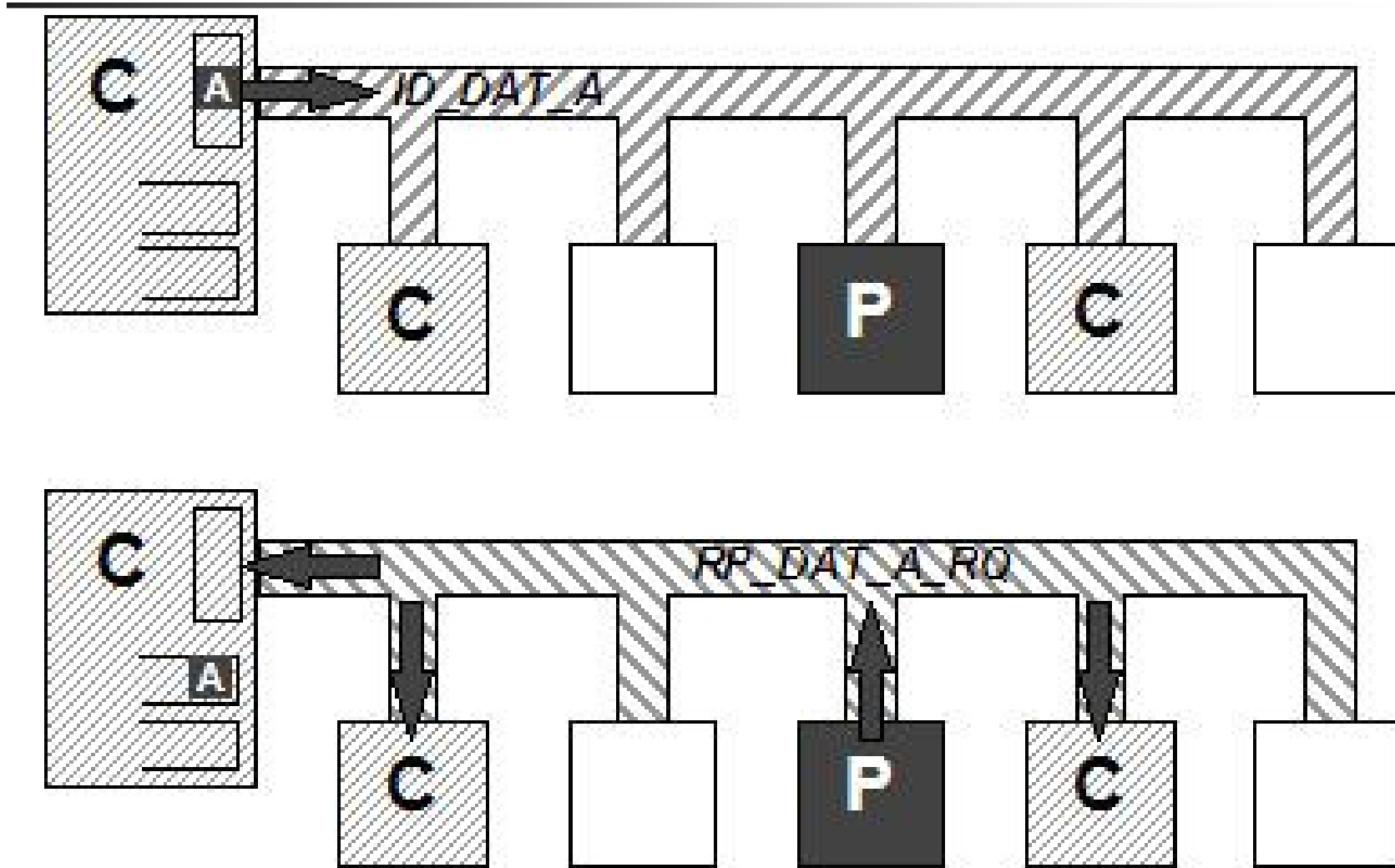
# WFIP Productor-Consumidor

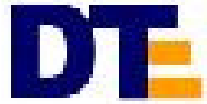




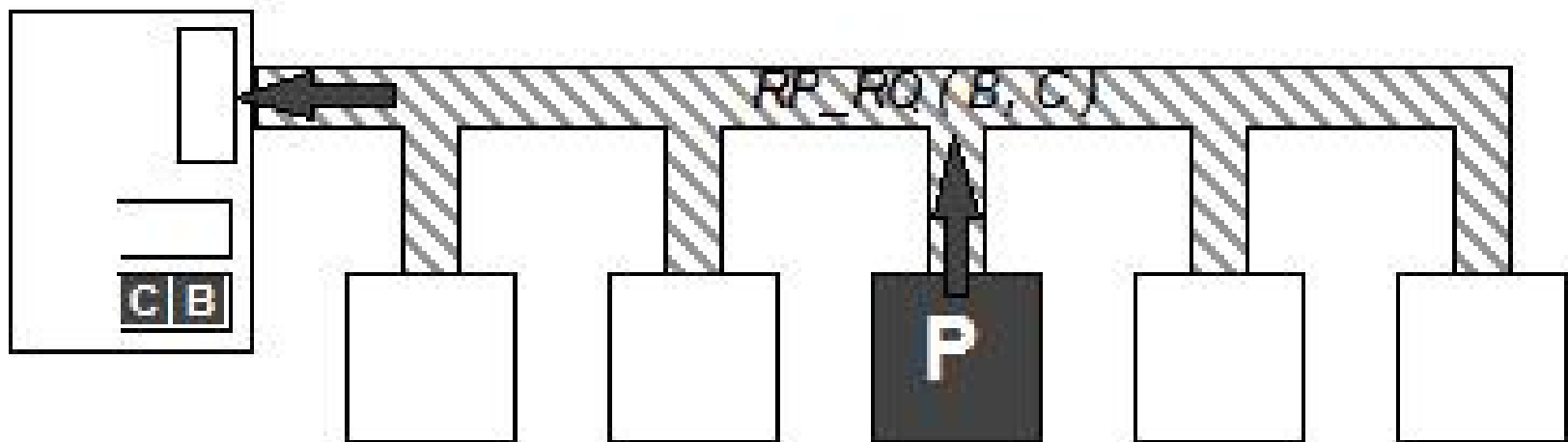
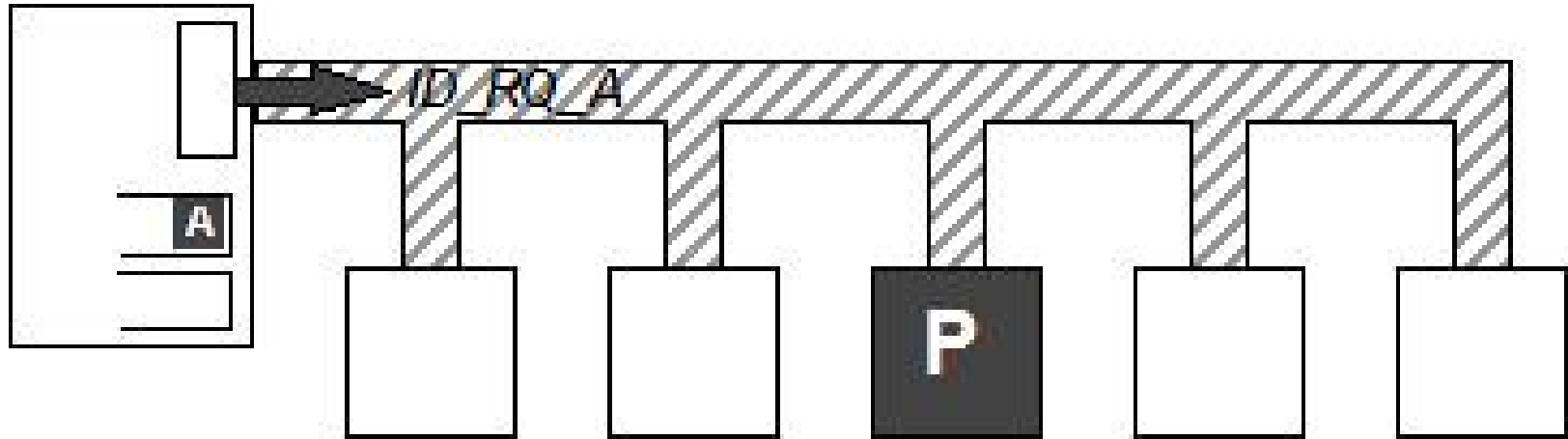


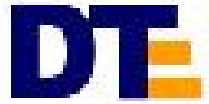
# WFIP Transferencias Aperiódicas





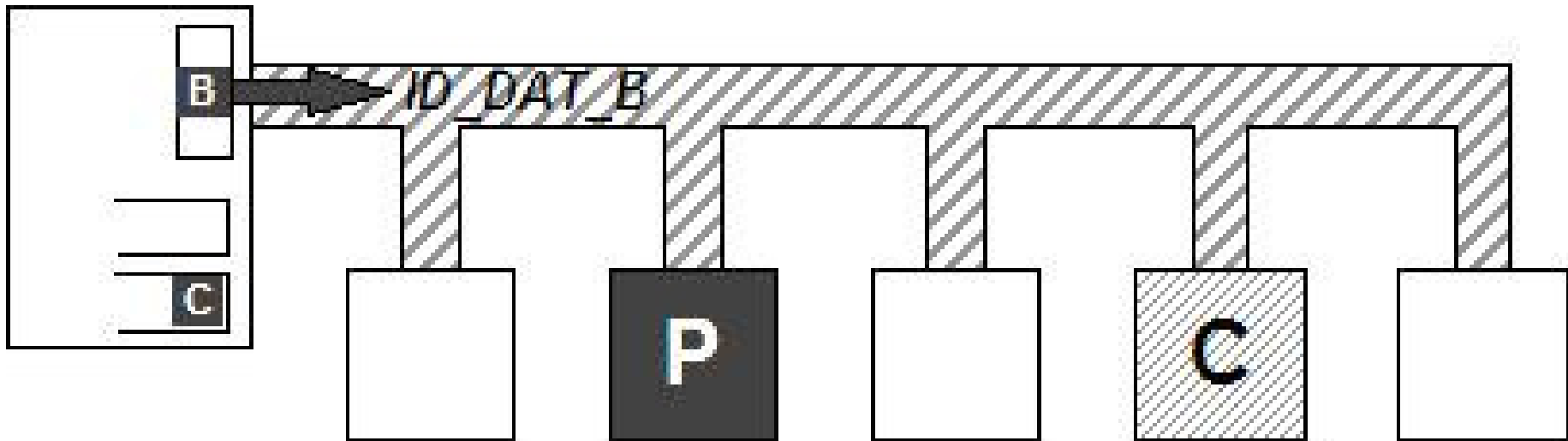
# WFIP Transferencias Aperiódicas

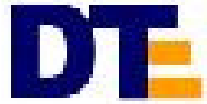




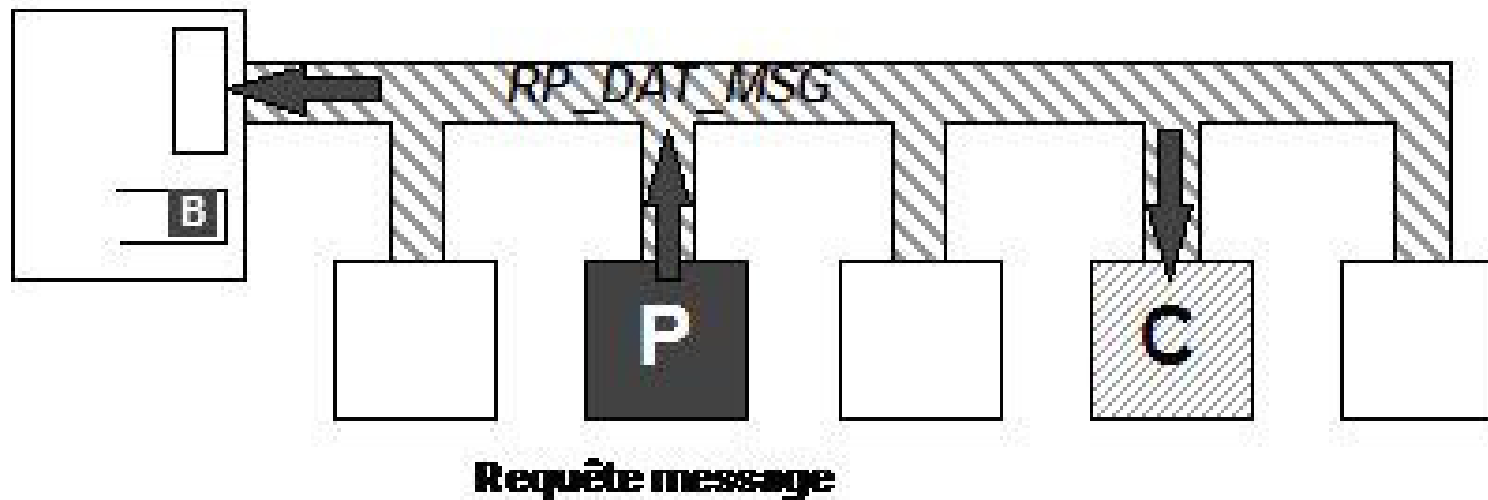
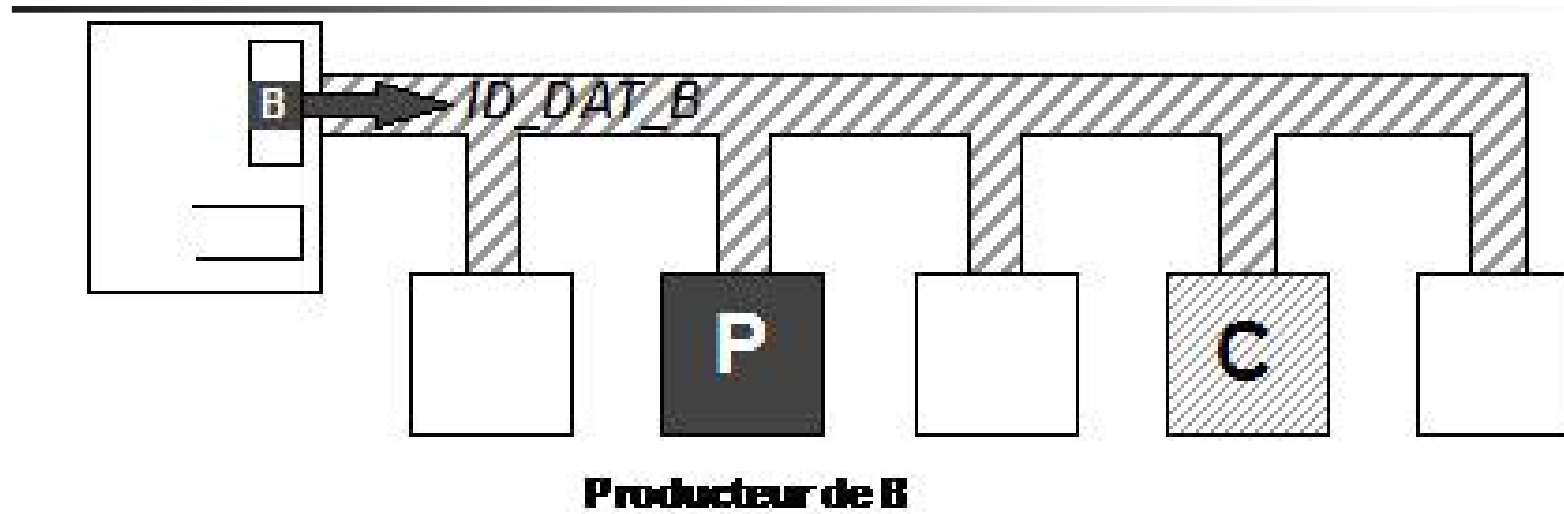
# WFIP Transferencias Aperiódicas

---

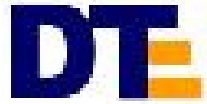




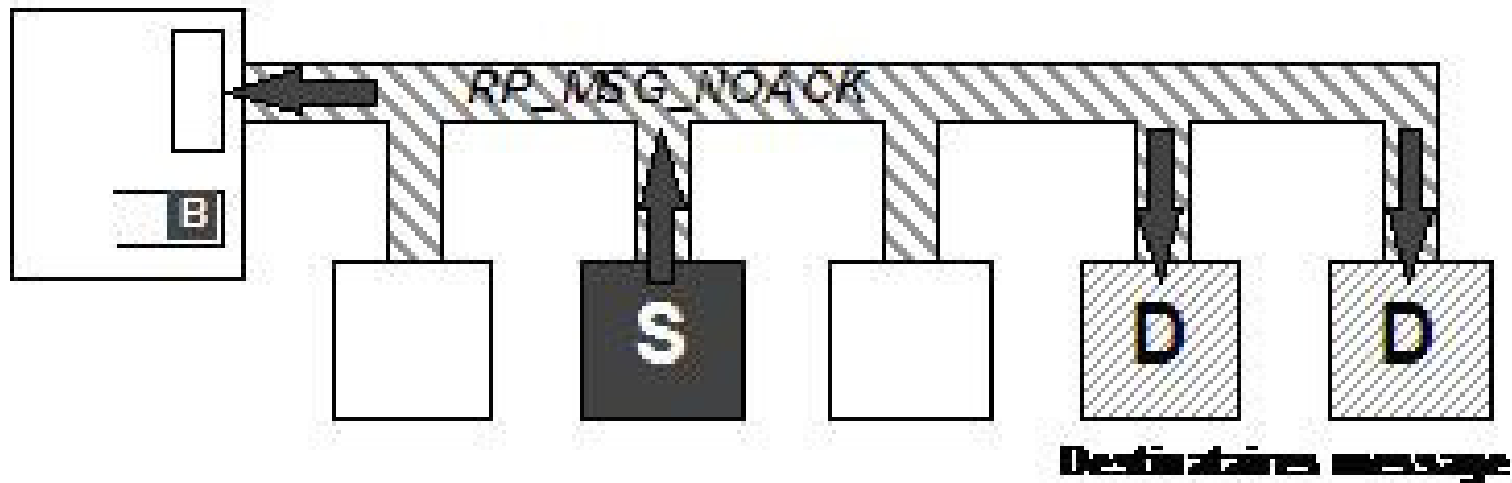
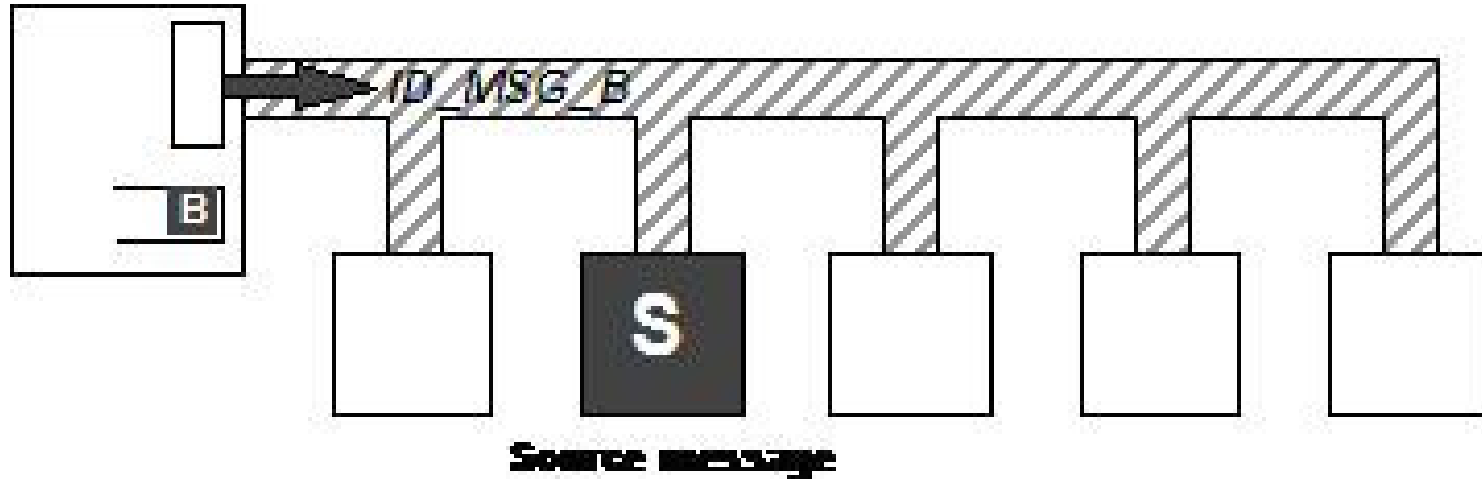
# WF Mensajes Sin Reconocimiento







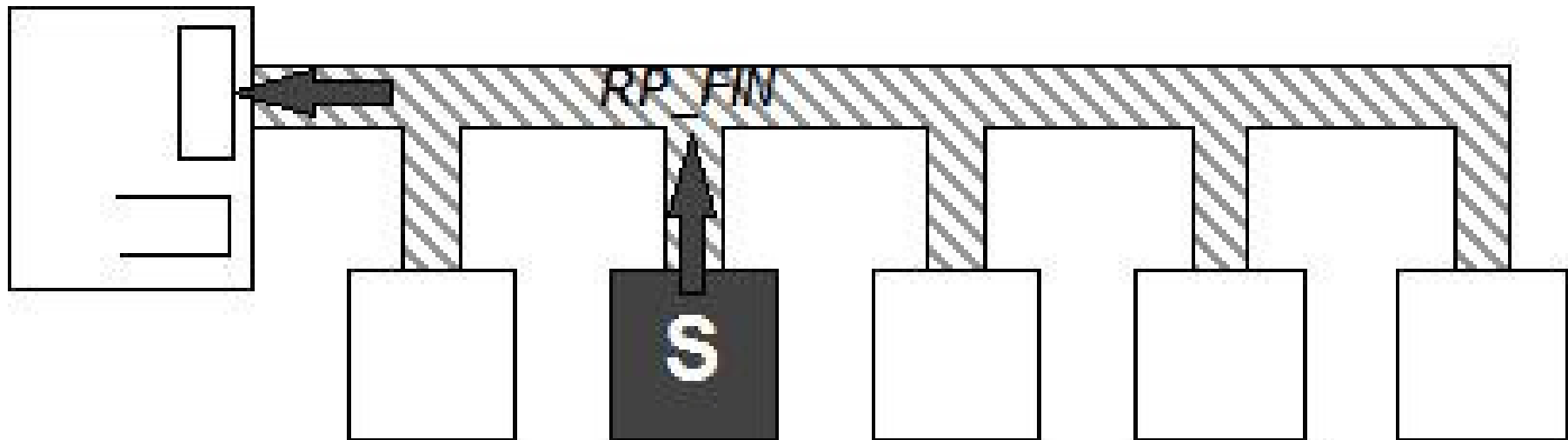
# WF Mensajes Sin Reconocimiento

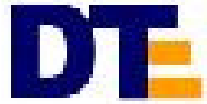




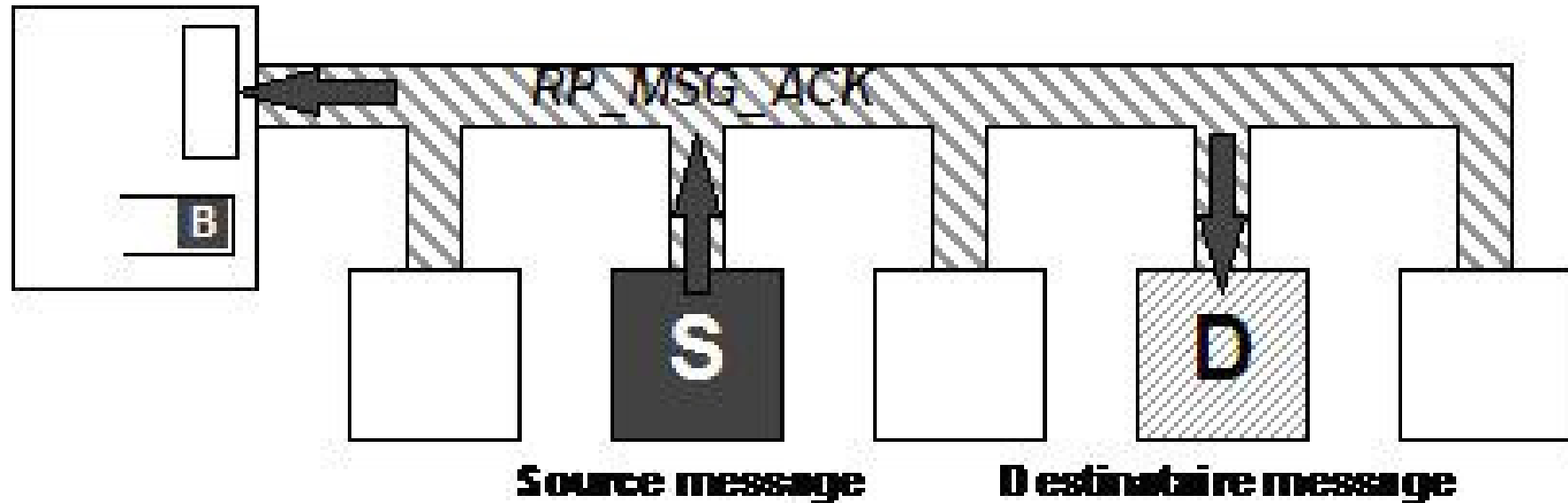
# WF Mensajes Sin Reconocimiento

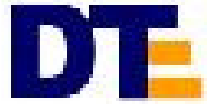
---



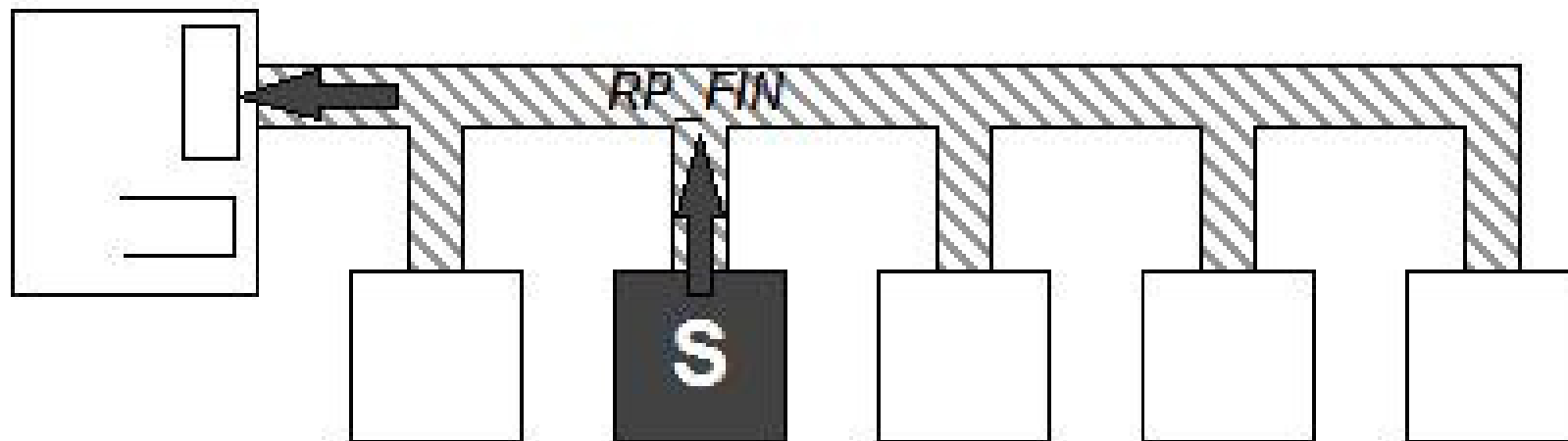
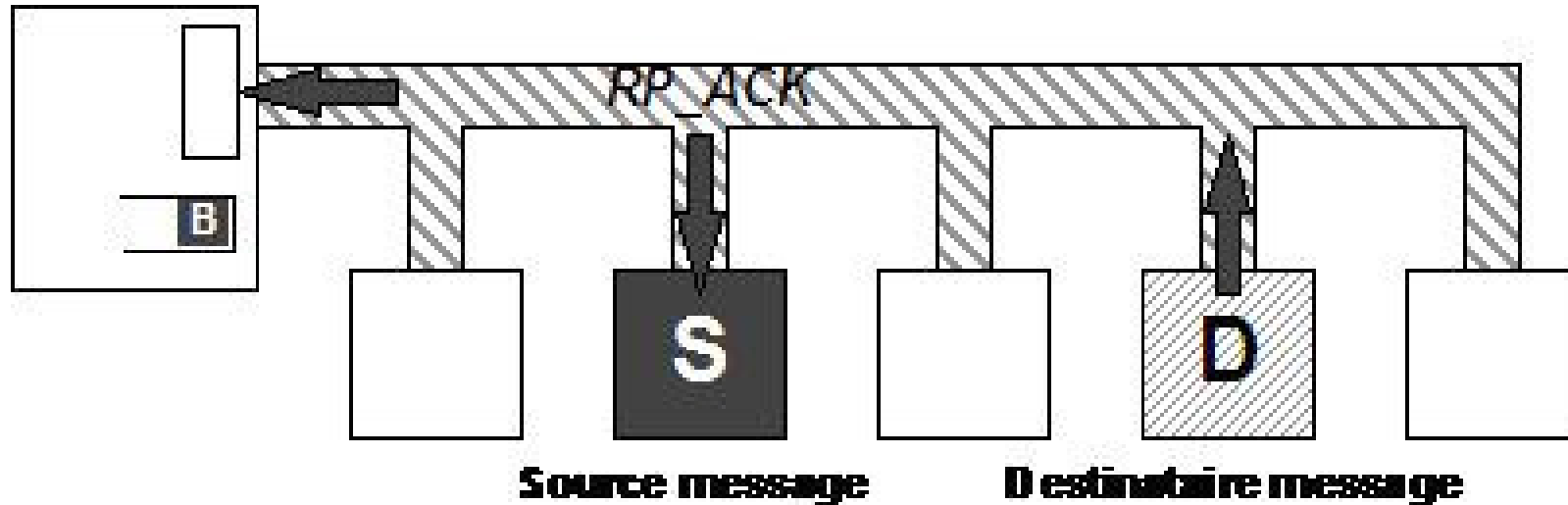


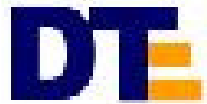
# WF Mensajes con Reconocimiento



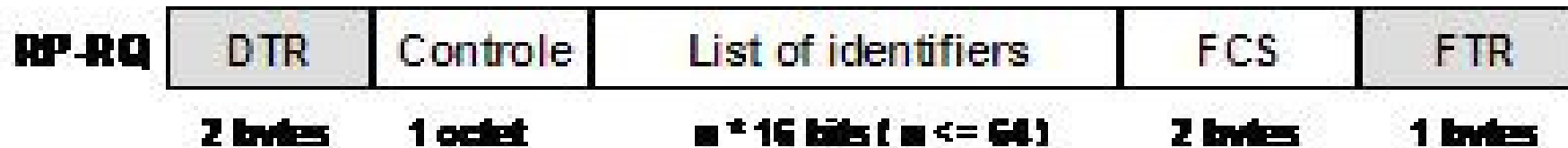
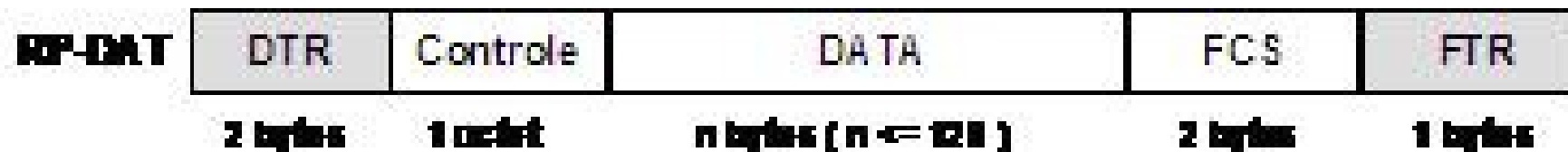
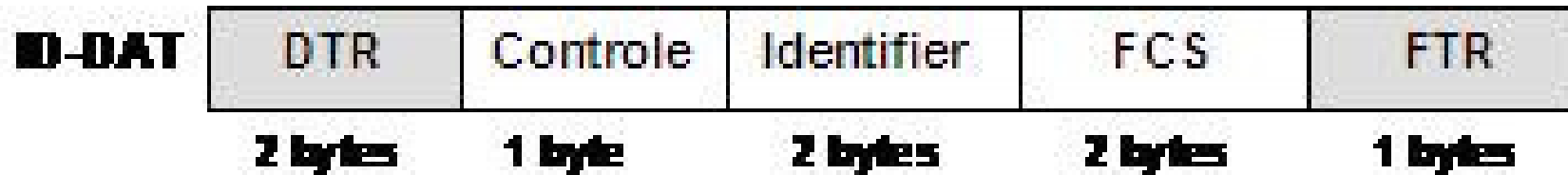


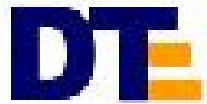
# WF Mensajes con Reconocimiento





# WORLDFIP TRAMAS



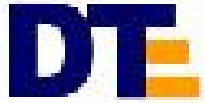


# WORLDFIP TRAMAS

<b>RP-MSG-ack</b>	DTR	Controle	dest. adr.	srce adr.	Message	FCS	FTR
	2 bytes	1 byte	3 bytes	3 bytes	max 256 bytes	2 bytes	2 bytes

<b>RP-ACK</b>	DTR	Controle	FCS	FTR
	2 bytes	1 byte	2 bytes	1 bytes

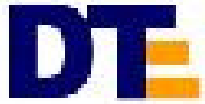
<b>RP-FIN</b>	DTR	Controle	FCS	FTR
	2 bytes	1 byte	2 bytes	1 bytes



# PROTOCOLO CAN

---

- Multi-maestro
- Niveles OSI 1,2 (físico y enlace)
- Se identifican mensajes y prioridad
- Métodos sofisticados de detección de errores
- Codificación NRZ+bit Stuffing
- Control de acceso: CSMA/CD+CR  
(acceso múltiple al medio con detección y resolución de colisión)

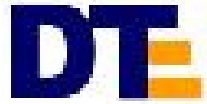


# CAN-CAPA FÍSICA

---

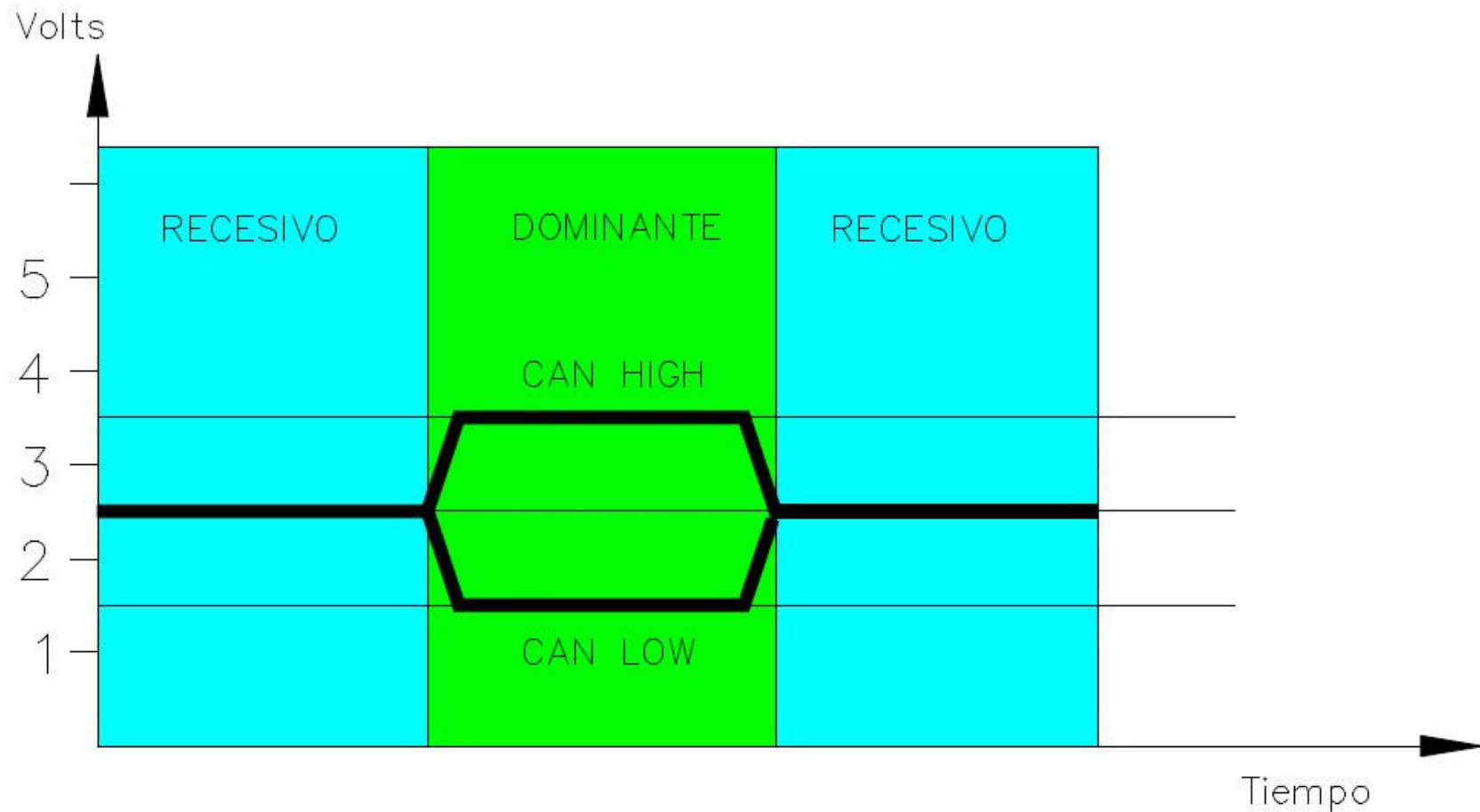
- ISO 11898
- Bus de dos hilos
  - CANH
  - CANL
- Niveles lógicos
  - Dominante CANH=3.5V CANL=1.5V
  - Recesivo CANH=CANL=2.5V

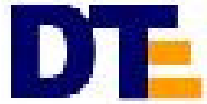




# CAN-CAPA FÍSICA

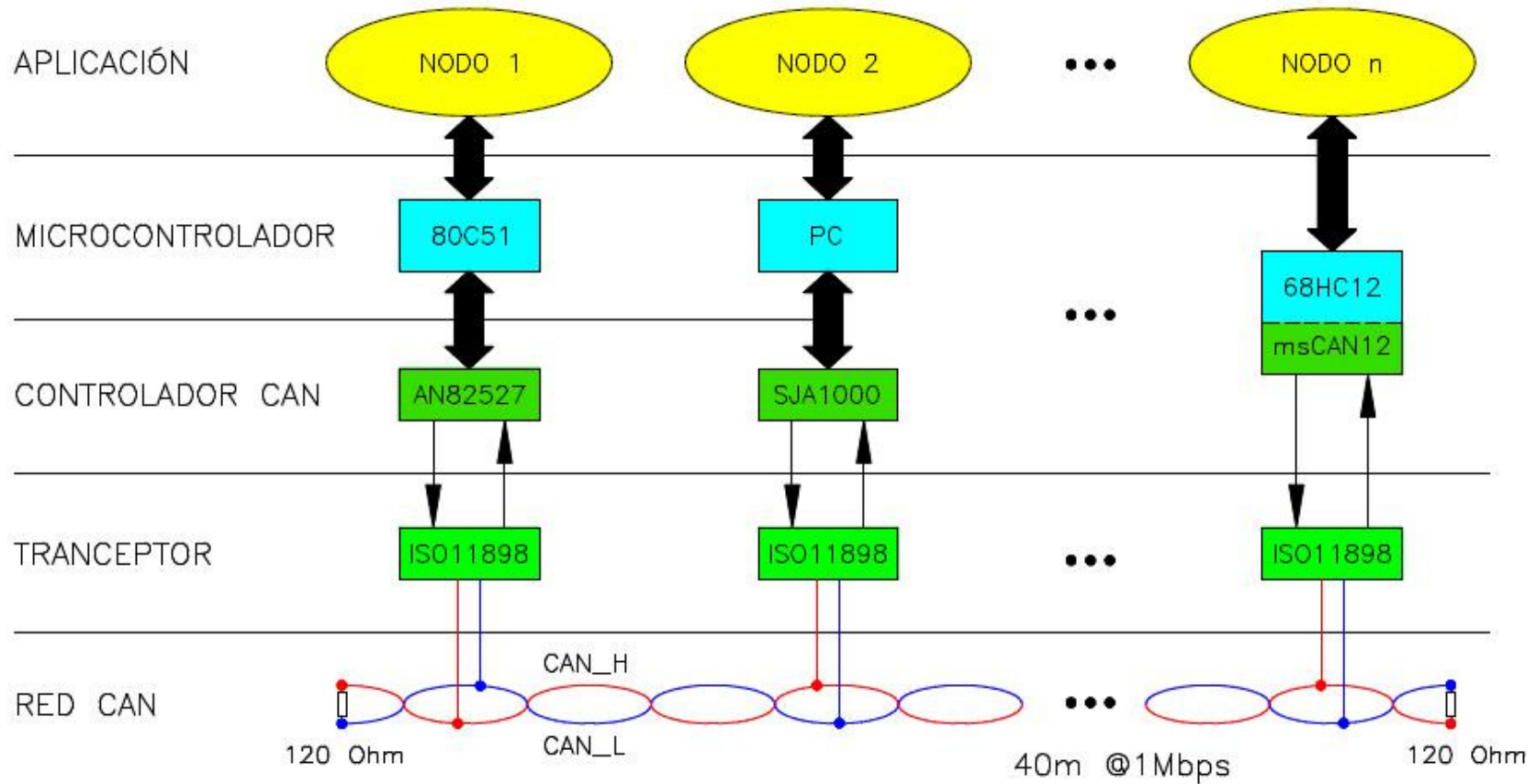
Estándar ISO11898

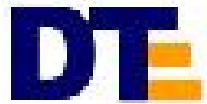




# CAN-CAPA FÍSICA

## Estructura de una red CAN

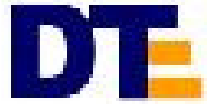




# CAN-CAPA FÍSICA

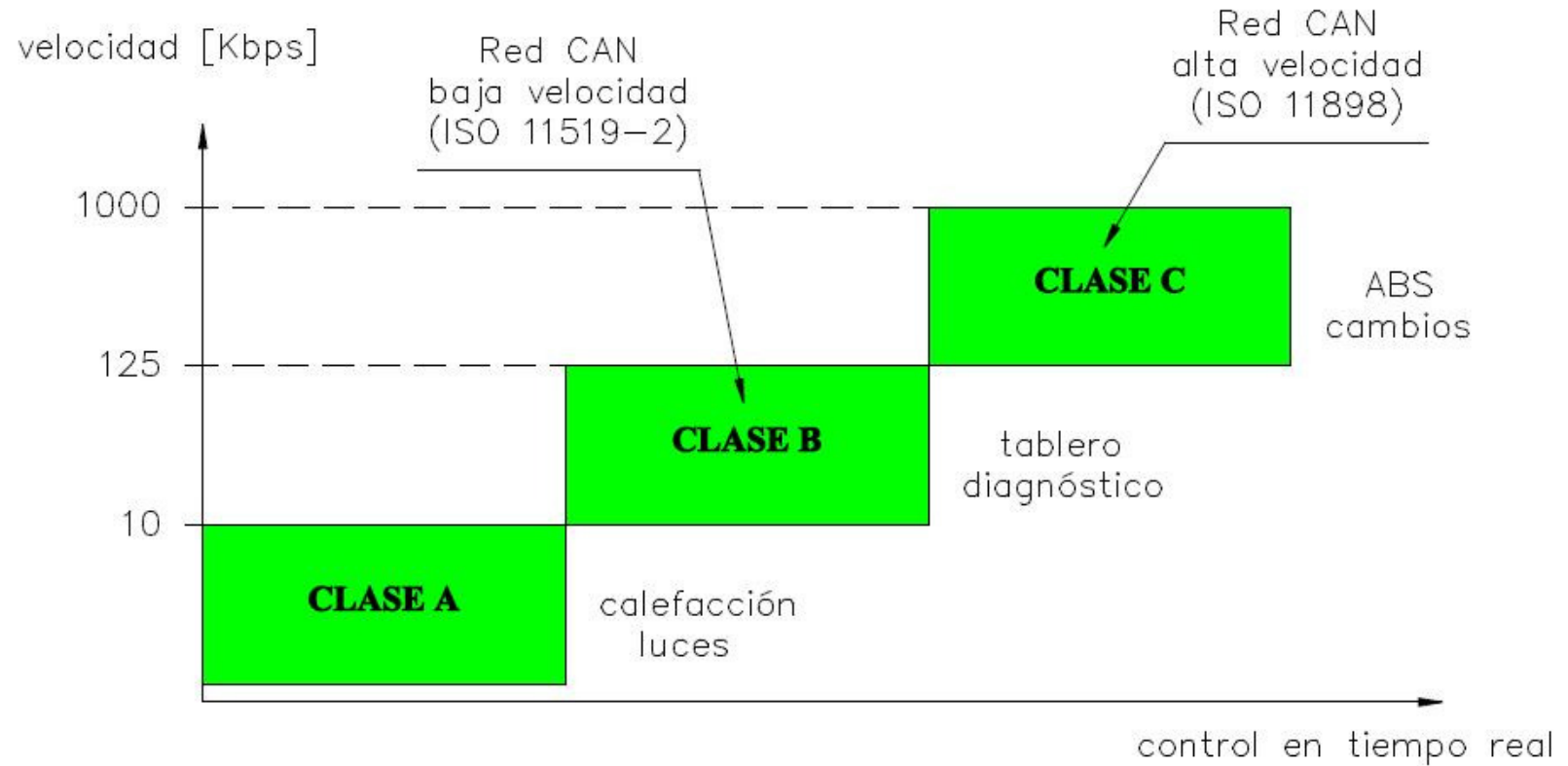
*Tabla 1. Velocidad-Distancia en CAN*

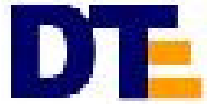
Velocidad	Tiempo de Bit	Longitud Máxima
1 Mbps	1 $\mu$ S	30 m
800 Kbps	1,25 $\mu$ S	50 m
500 Kbps	2 $\mu$ S	100 m
250 Kbps	4 $\mu$ S	250 m
125 Kbps	8 $\mu$ S	500 m
50 Kbps	20 $\mu$ S	1000 m
20 Kbps	50 $\mu$ S	2500 m
10 Kbps	100 $\mu$ S	5000 m



# CAN-CAPA FÍSICA

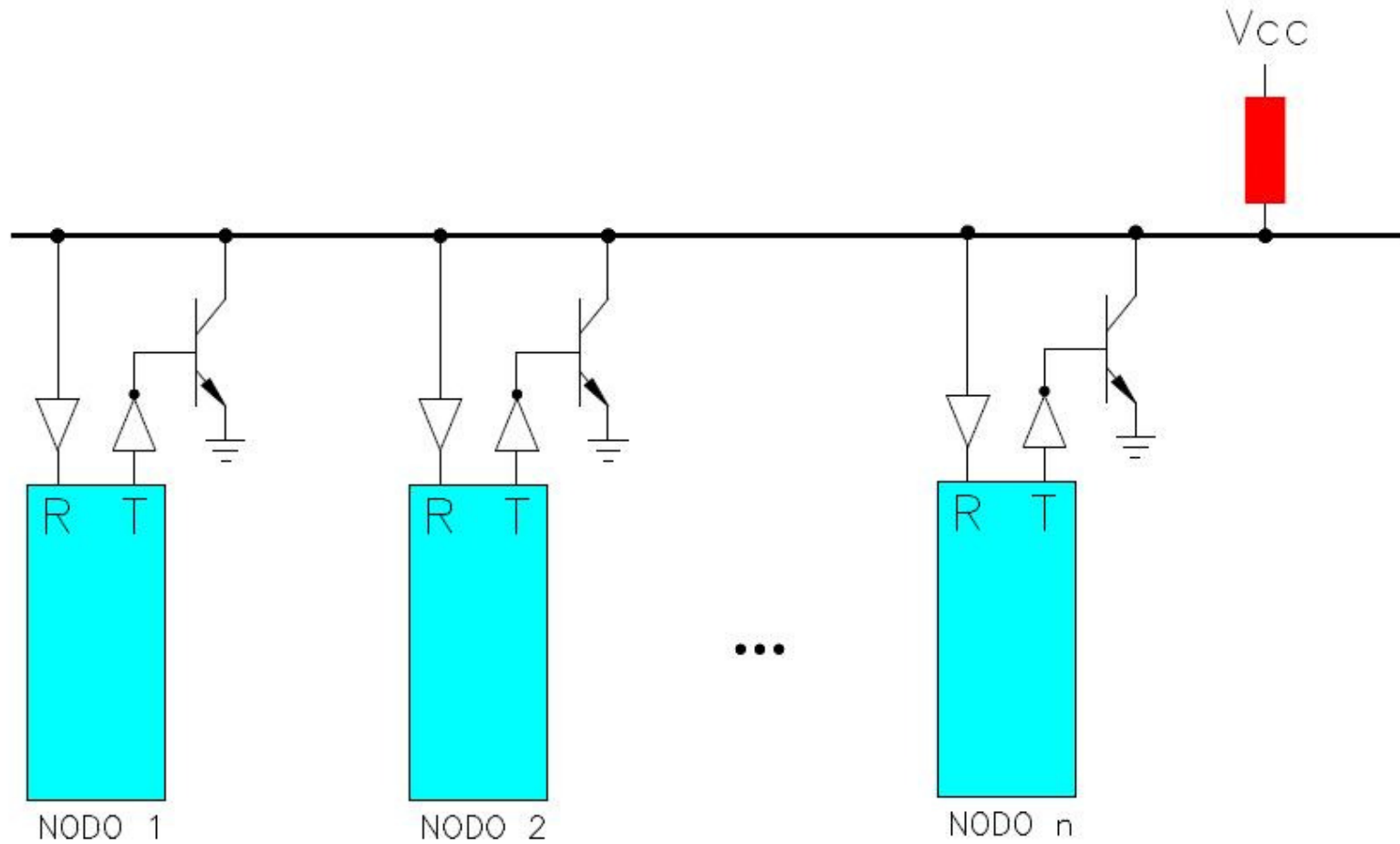
## Estándar Medio físico





# CAN-CAPA ENLACE

## Control de bus de la red CAN





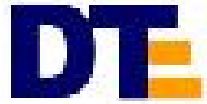
# CAN-CAPA ENLACE

NODO			BUS
1	2	3	
D	D	D	D
D	D	R	D
D	R	D	D
D	R	R	D
R	D	D	D
R	D	R	D
R	R	D	D
R	R	R	R

## BUS CAN LÓGICA AND CABLEADA

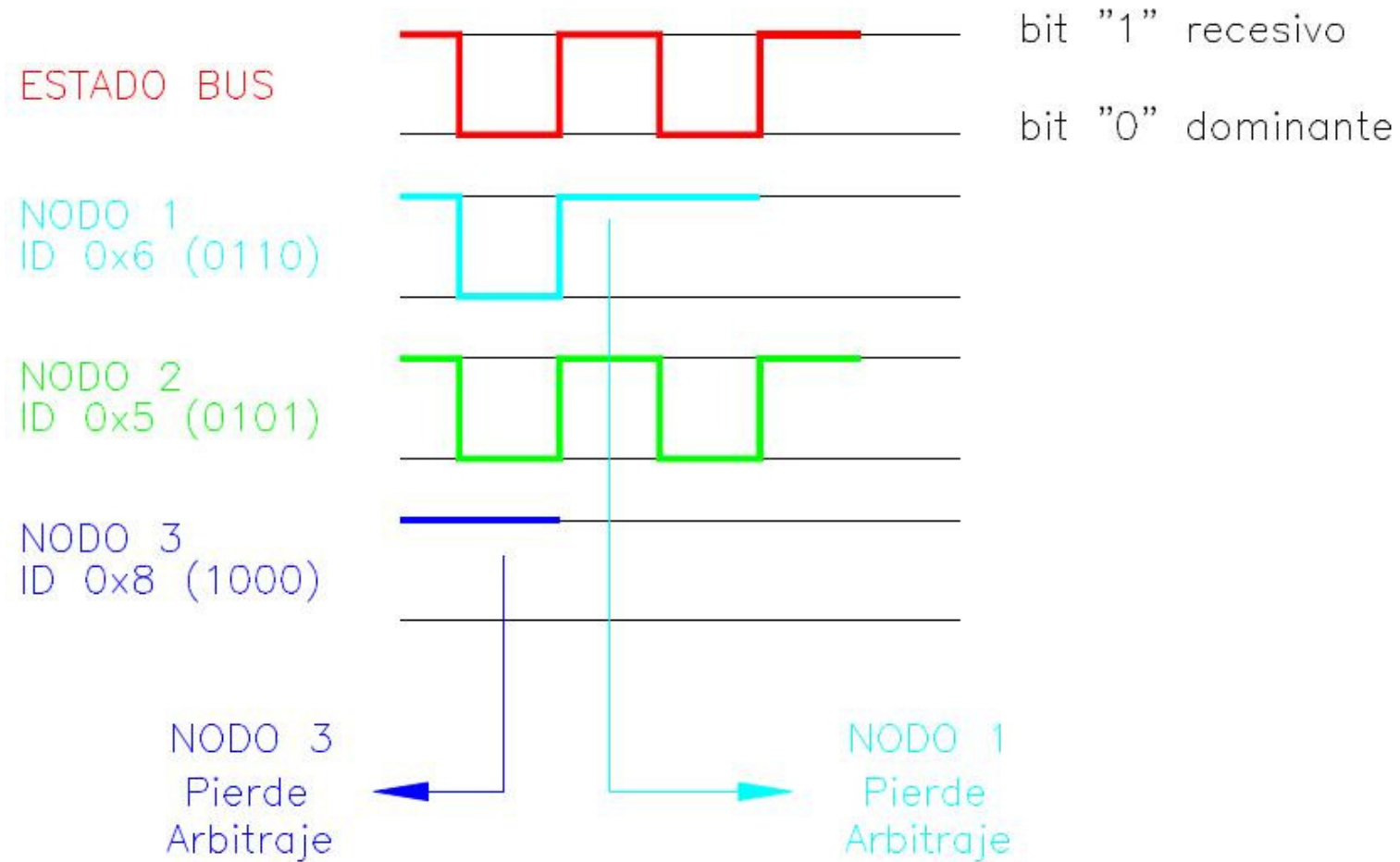
Si sólo un nodo pone su salida dominante ("0")  
=> el bus está en estado dominante

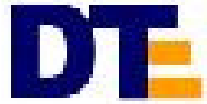
Si todos los nodos ponen salida recesiva  
=> bus en estado recesivo



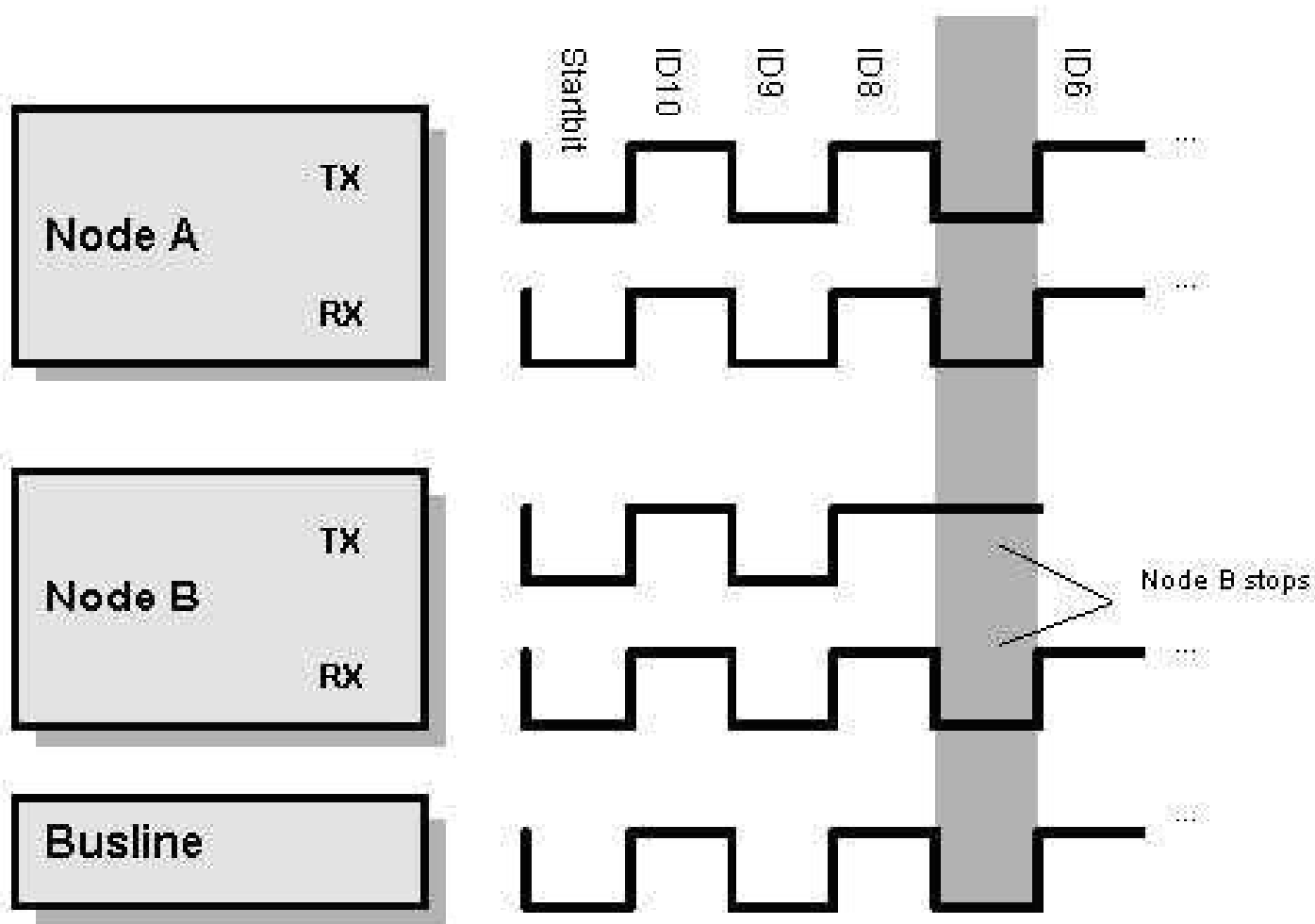
# CAN-CAPA ENLACE

## Arbitraje de bus CAN

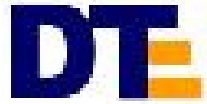




# CAN-CAPA ENLACE







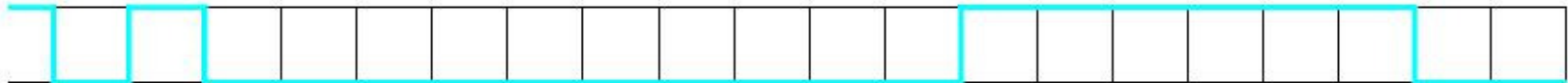
# CAN-CAPA ENLACE

## Bit Stuffing

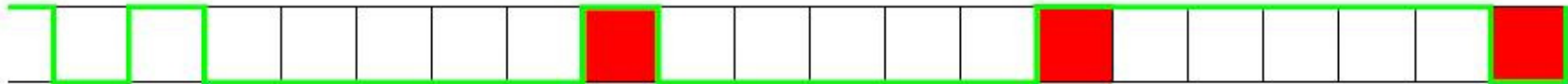
NUMERO DE BITS  
CON IGUAL POLARIDAD

1 2 3 4 5 6 7 8 9 10 1 2 3 4 5 6

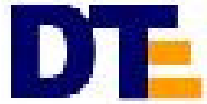
DATOS A  
TRANSMITIR



DATOS  
TRANSMITIDOS



1 2 3 4 5 S 6 7 8 9 10 S 1 2 3 4 5 S



# CAN-TIPOS DE TRAMAS

---

## Tipos de Tramas en CAN

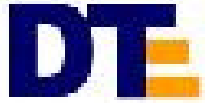
**TRAMA  
DE  
DATOS**

**TRAMA  
REMOTA**

**TRAMA  
DE  
ERROR**

**TRAMA  
DE  
SOBRECARGA**

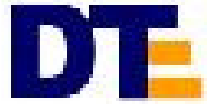
**ESPACIO  
INTERTRAMA**



# CAN-TRAMAS

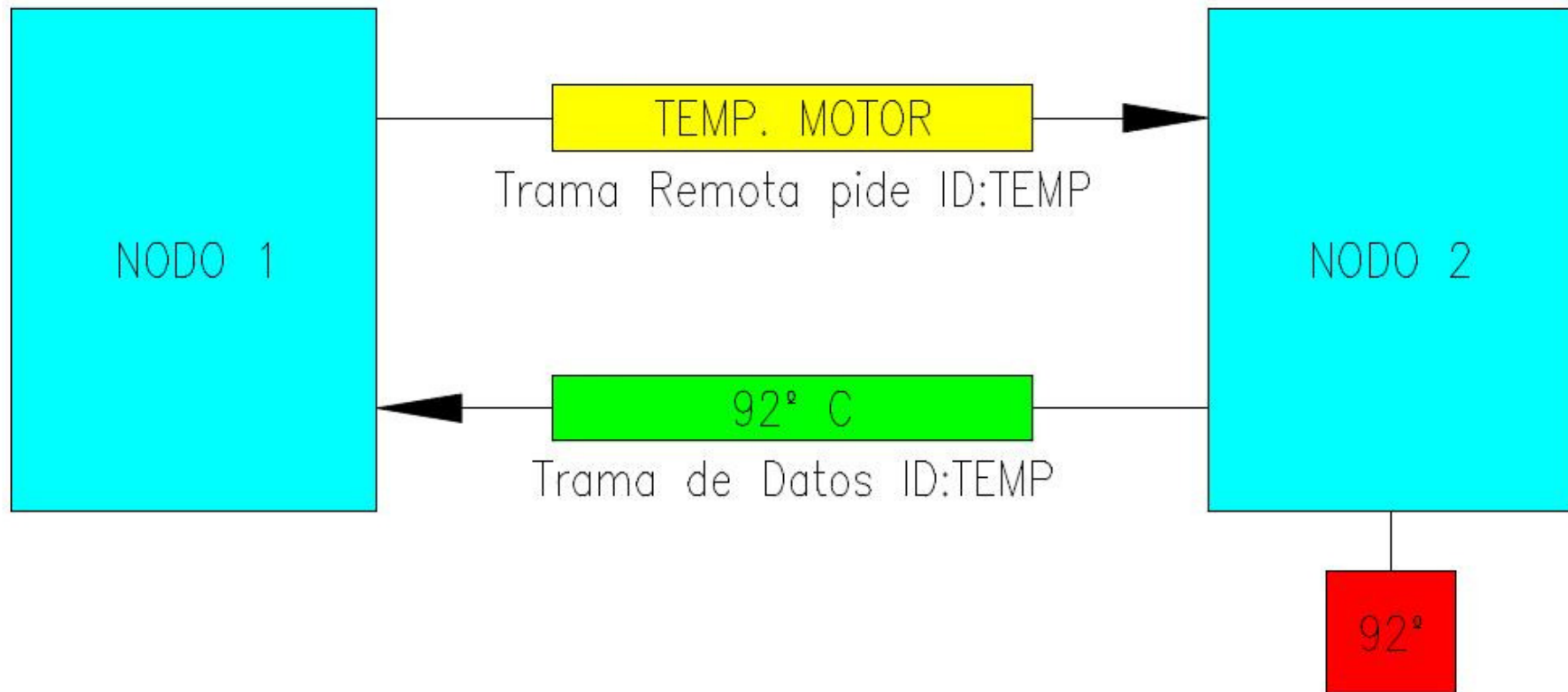
---

- TRAMA DE DATOS
  - Se usa para enviar datos nodo-nodo
  - Es la trama habitual
  
- TRAMA REMOTA
  - Se usa para solicitar datos de un nodo
  - No contiene datos solo el ID del dato



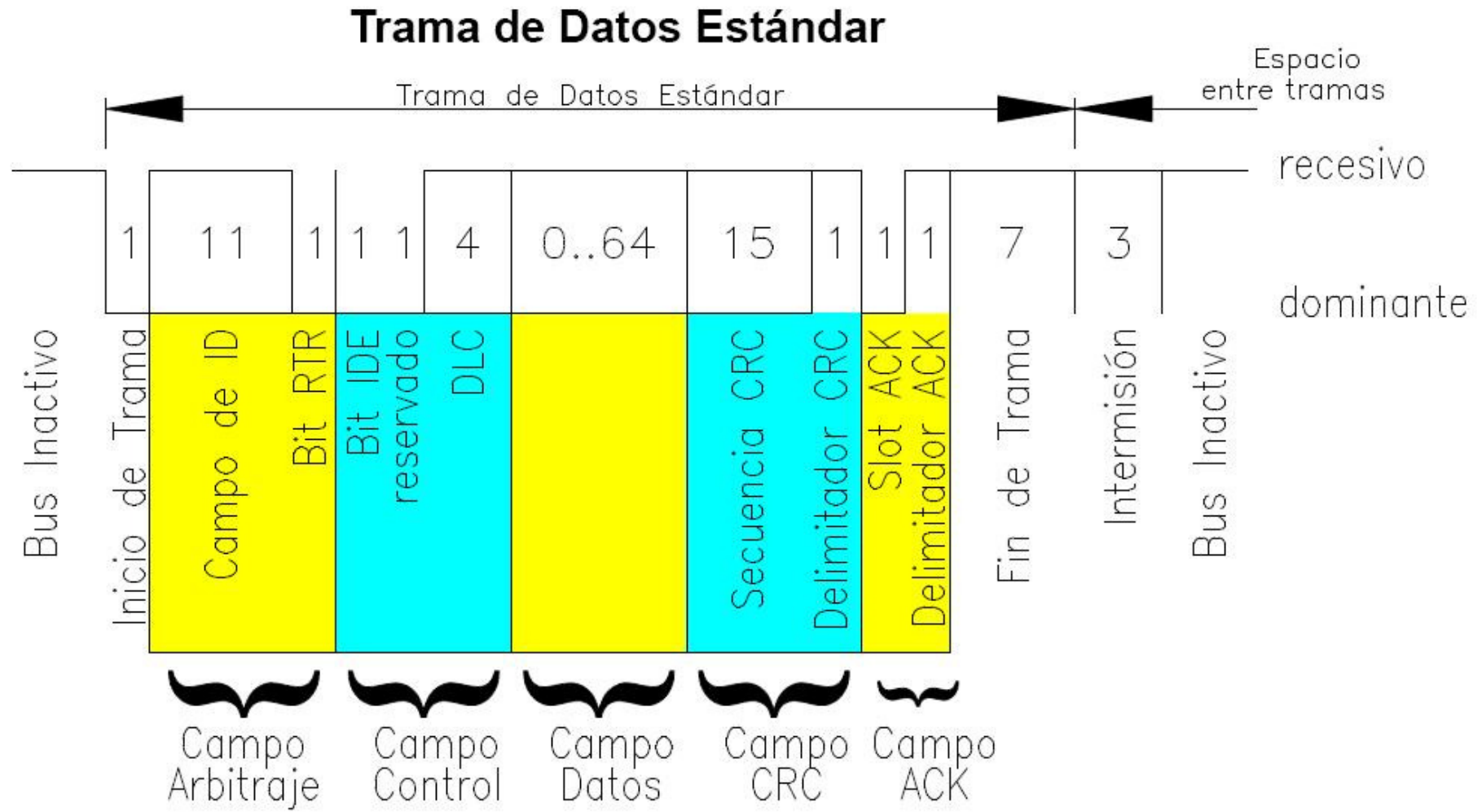
# CAN-TRAMAS

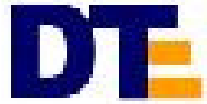
## Trama Remota Estándar (ejemplo)





# CAN-TRAMA ESTÁNDAR





# CAN-SINCRONISMO

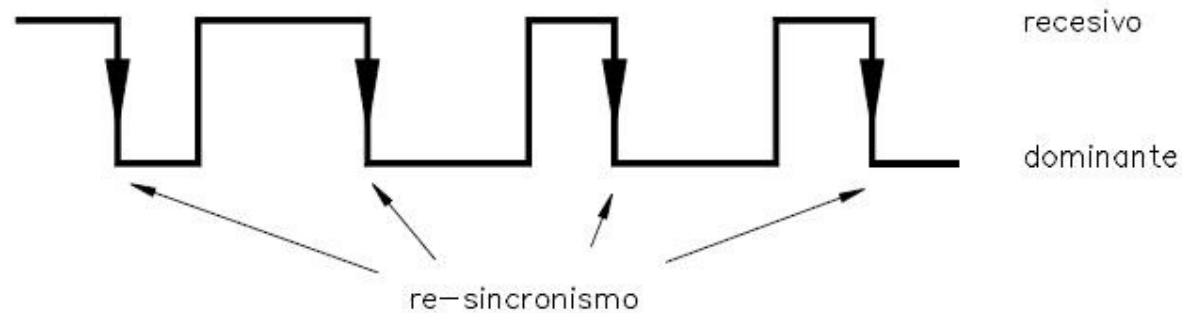
## Sincronismo de Nodos

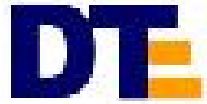
### TRAMA CAN



↑ Todos los nodos se sincronizan en el flanco descendente del bit SOF (sincronización HARD)

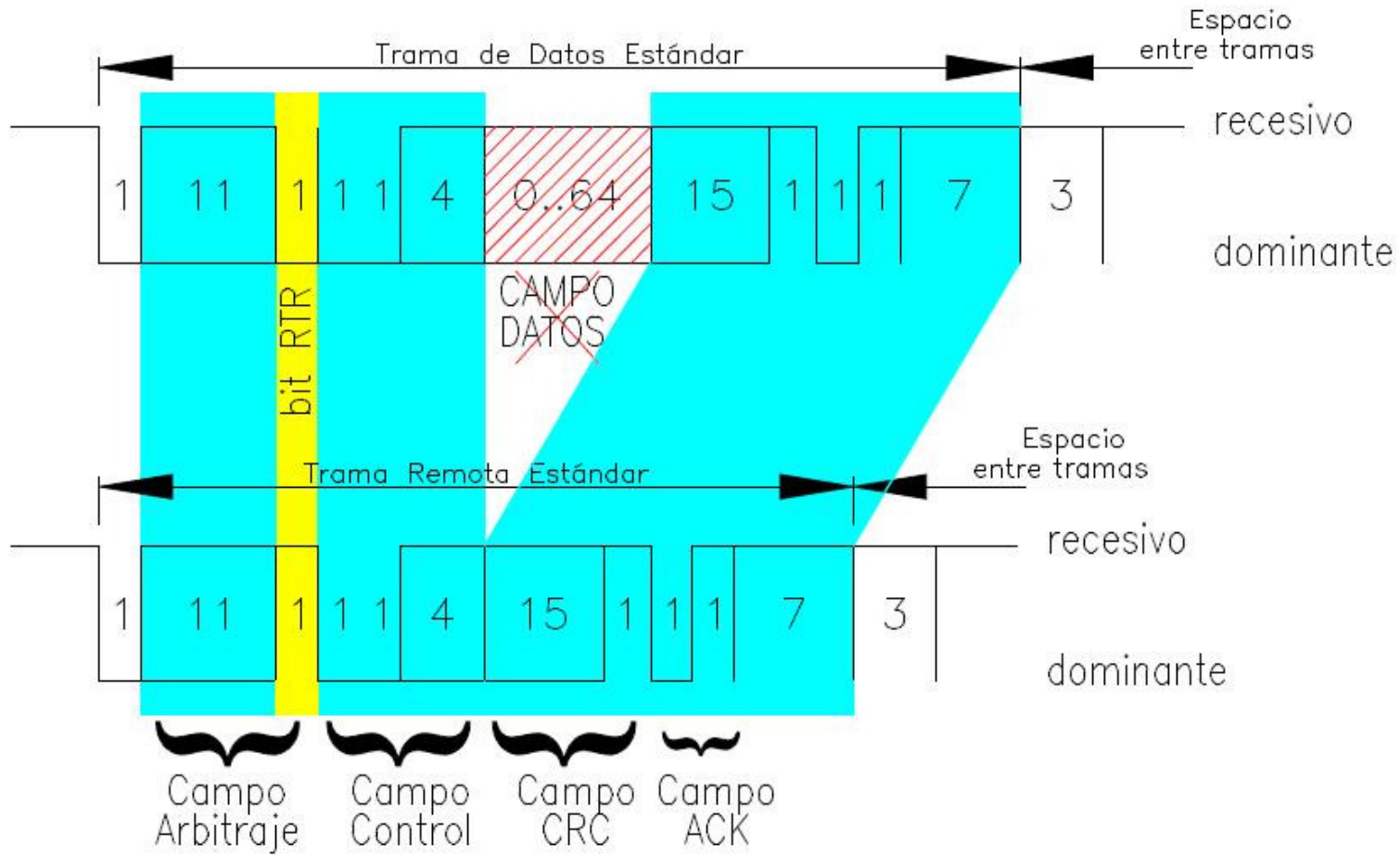
### BITS EN LA TRAMA

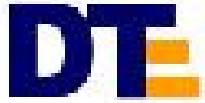




# CAN-TRAMA REMOTA

## Trama Remota Estándar

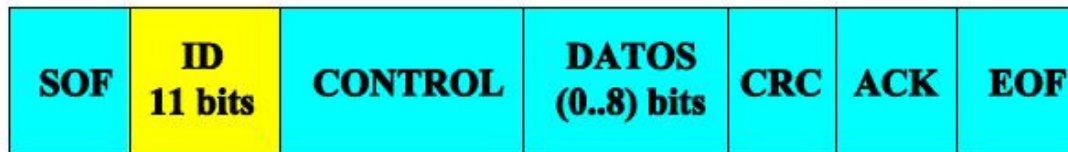




# CAN-FORMATO EXTENDIDO

## Protocolo CAN Extendido

### TRAMA ESTÁNDAR



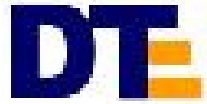
### TRAMA EXTENDIDA



ESTÁNDAR: Largo identificador es de 11 bits  $\Rightarrow$  2048 ID's posibles.

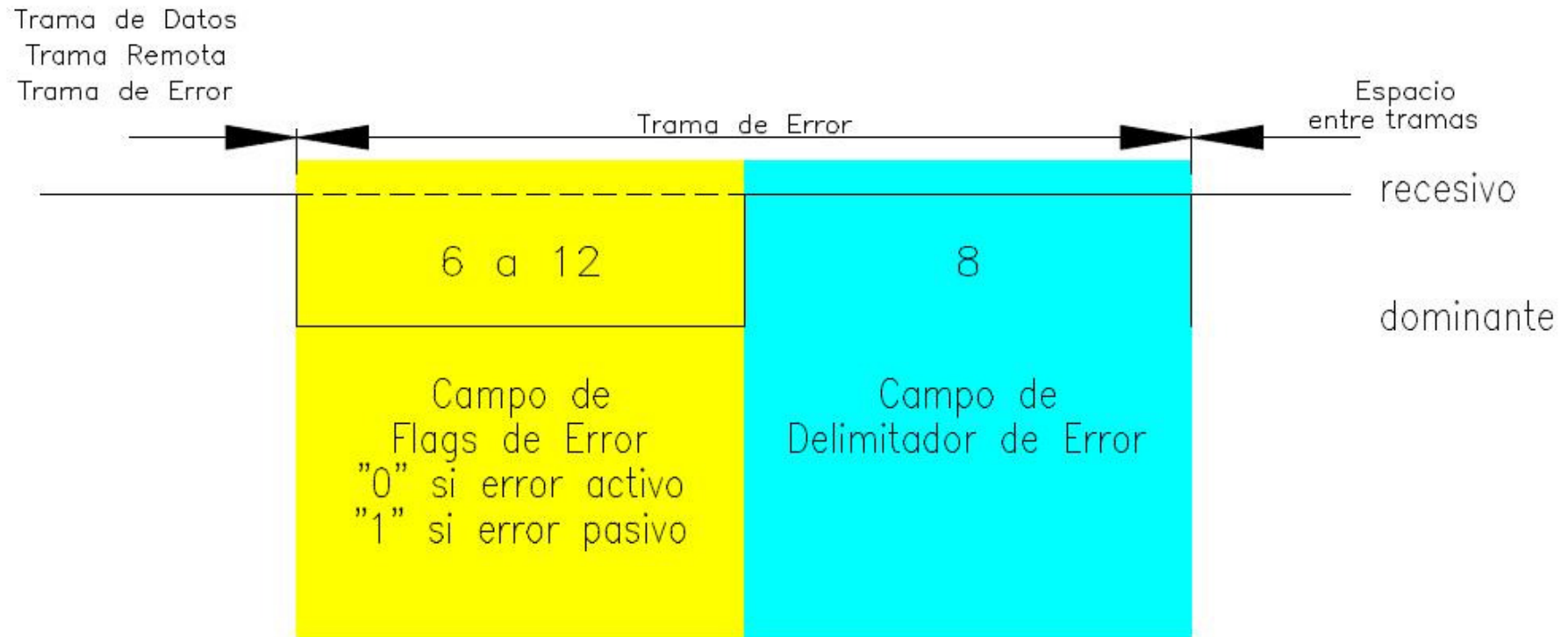
EXTENDIDO: Largo identificador es de 29 bits  $\Rightarrow$  más de 536 millones de ID's posibles.

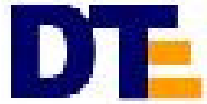




# CAN-TRAMA ERROR

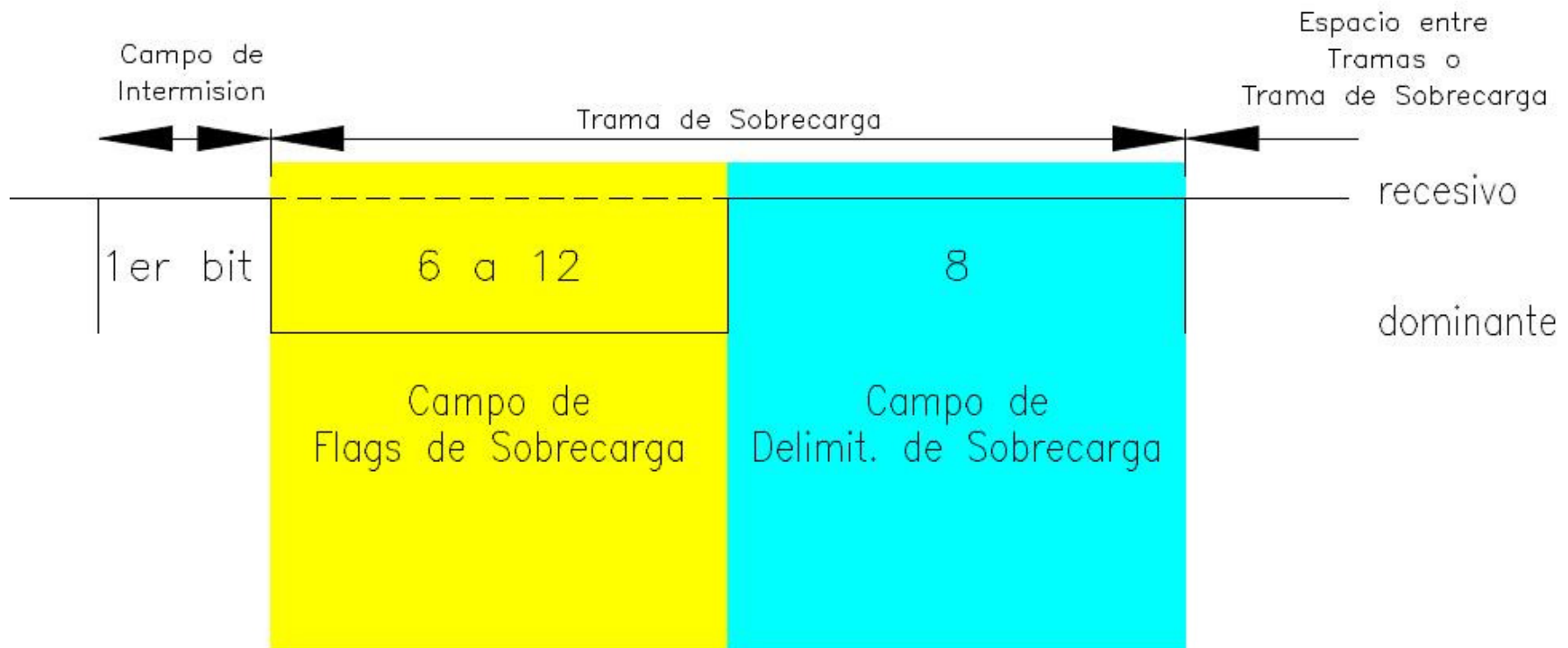
## Trama de Error

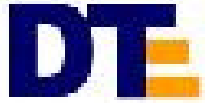




# CAN-TRAMA SOBRECARGA

## Trama de Sobrecarga

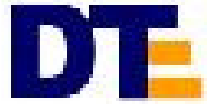




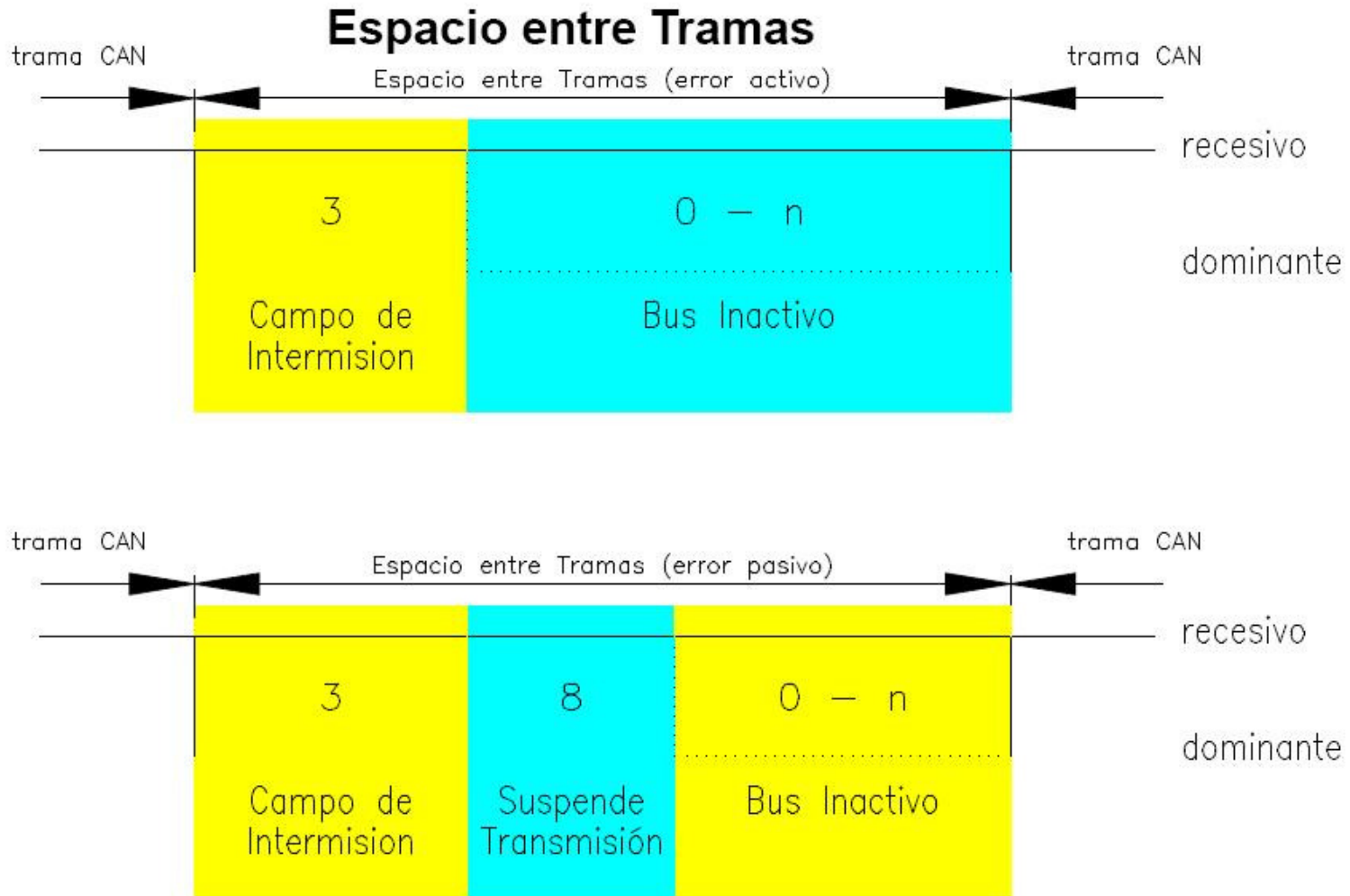
# CAN-ESPACIO ENTRE TRAMAS

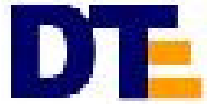
---

- Al menos 3 bits recesivos (intermission)
  
- Nodo error Activo
  - Inicia transmisión
  - Reposo
  
- Nodo error pasivo
  - Debe esperar 8 bits recesivos adicionales



# CAN-ESPACIO ENTRE TRAMAS





# CAN-TIPOS ERRORES

---

## Tipo de Errores Detectados

**ERROR  
DE  
CRC**

**ERROR  
DE  
FORMA**

**ERROR  
DE  
STUFFING**

**ERROR  
DE  
ACK**

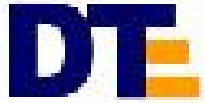
**ERROR  
DE  
BIT**



# CAN-ERRORES

---

- CRC
  - Detectado en receptor
  - Descarta trama
  - Transmite trama error
- ACK
  - Detectado por transmisor
  - No recibe ACK de ningún nodo
  - Retransmite la trama
  - No se genera trama de error



# CAN-ERRORES

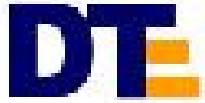
---

## ■ FORMA

- Se produce si se detecta bit dominante en
  - Delimitador de CRC
  - Delimitador de ACK
  - Fin de trama
- Se transmite trama de error

## ■ STUFFING

- 6 bits consecutivos del mismo valor
- Se envía trama de error

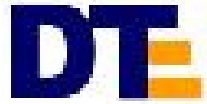


# CAN-ERRORES

---

- BIT
  - Realizado por el transmisor
  - Bit recibido distinto transmitido
  - No se considera en los campos
    - Arbitraje
    - Slot de reconocimiento
  - Se transmite trama de error
  - Se retransmite la trama





# CAN-CONTADORES ERRORES

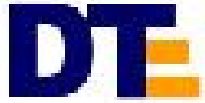
CER = CONTADOR ERROR DE RECEPCIÓN  
CET = CONTADOR ERROR DE TRANSMISIÓN

CER  $\leq$  127  
Y  
CET  $\leq$  127

CER  $>$  127  
Ó  
CET  $>$  127

CET  $>$  255

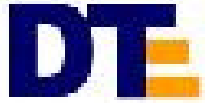




# CAN-ESTADO ERRORES

---

- **ERROR ACTIVO**
  - Estado normal
    - Puede transmitir y recibir mensajes
    - Puede transmitir tramas de error activas
- **ERROR PASIVO**
  - Algún contador  $>127$
  - Las tramas de error que se envían son pasivas



# CAN-ESTADO ERRORES

---

- BUS APAGADO
  - Si errores transmisión > 256
  - Todas las actividades se detienen
  - Se libera el bus (recesivo)
  - Debe reiniciarse por la CPU