MODBUS

La Storia

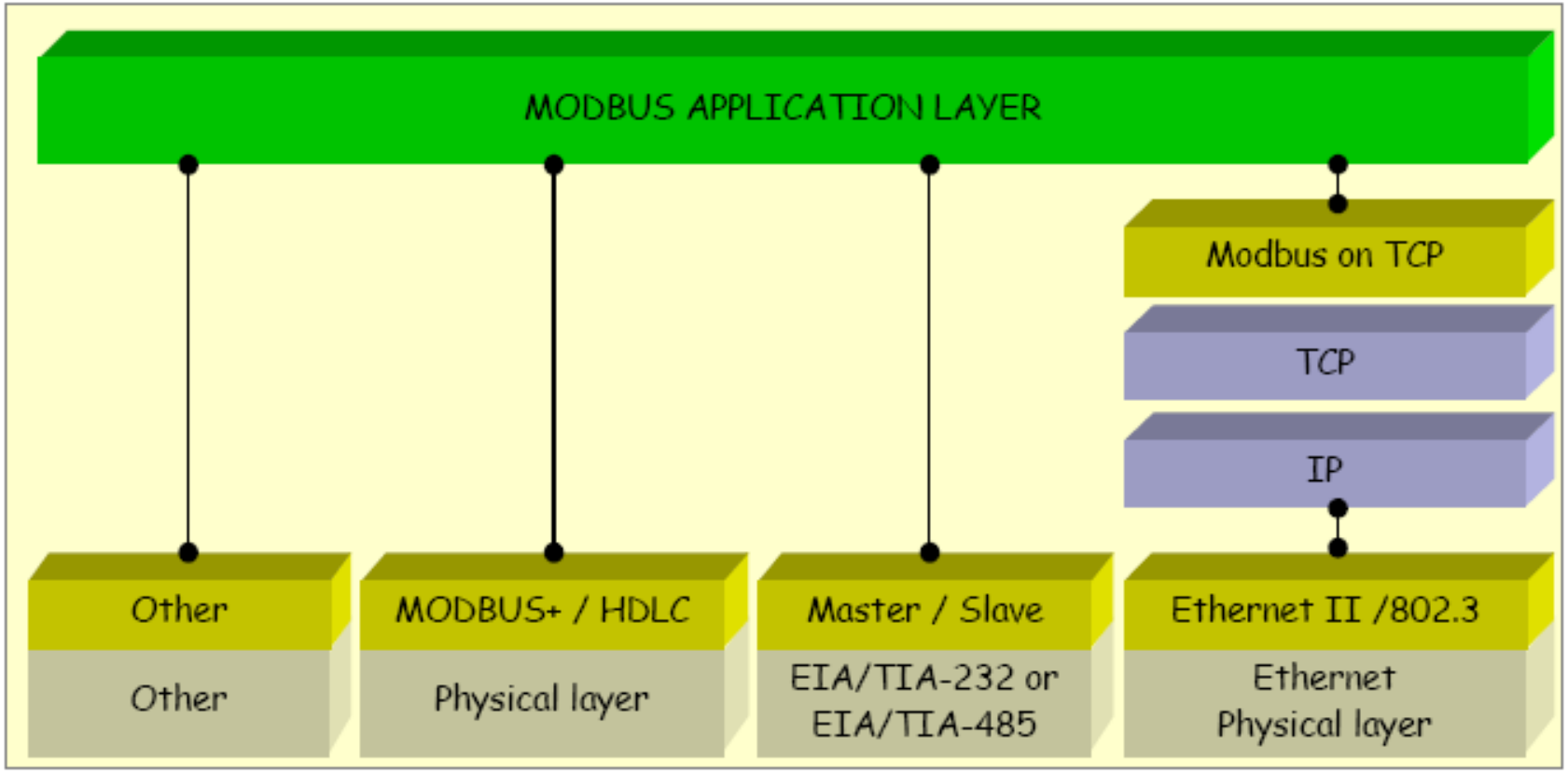
- Diventa uno STANDARD nel 1979
- Nato come protocollo di comunicazione SERIALE si è successivamente adattato alle specifiche TCP/IP
- Permette una comunicazione Client/Server o Master/Slave
- Semplicità del Protocollo

Argomenti trattati

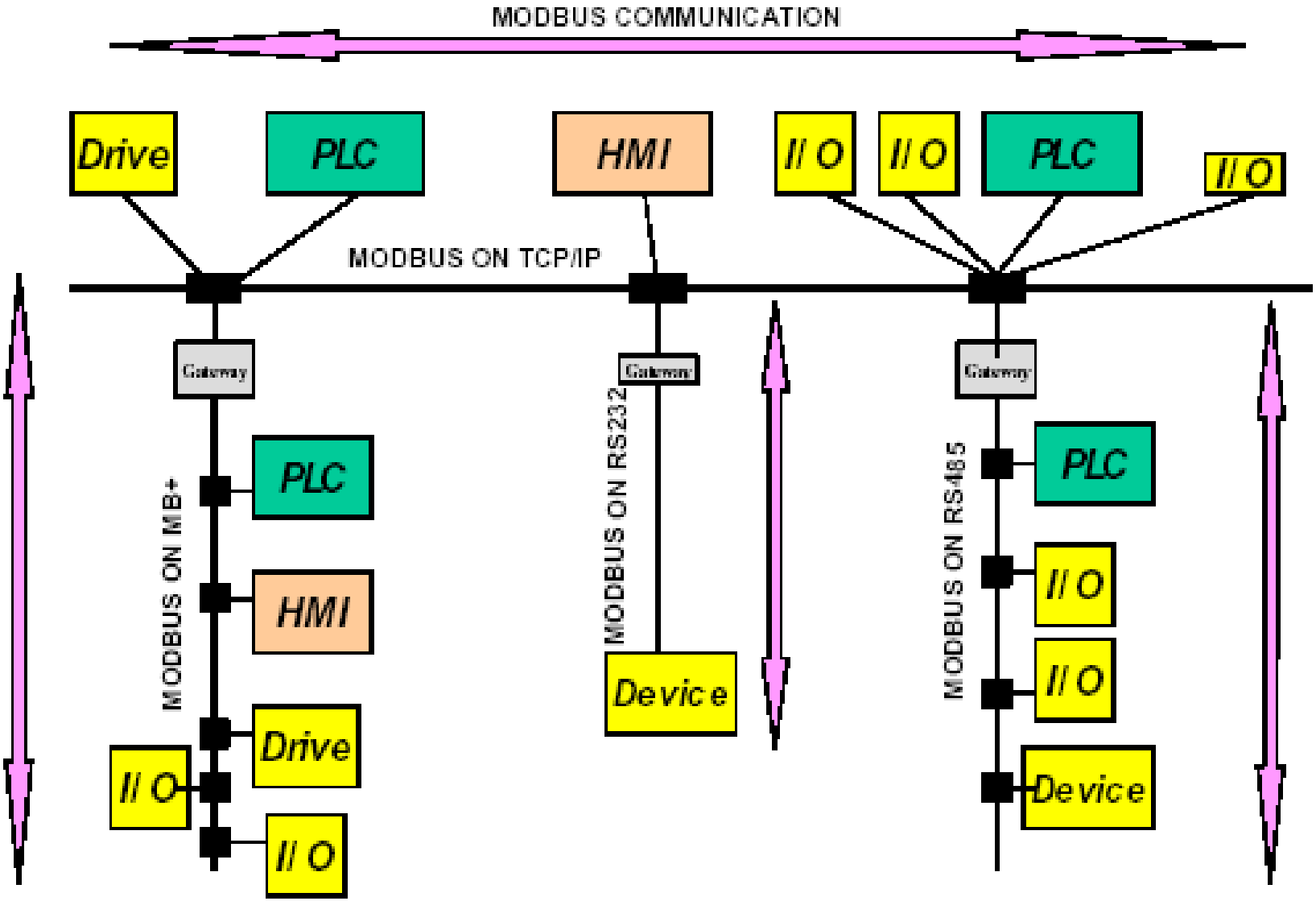
- Specifiche generali
  - Strato Applicazione
    - Costruzione del PDU
    - Organizzazione dei dati
- Implementazione SERIALE
  - Costruzione del pacchetto
- Implementazione su TCP/IP
  - Costruzione del pacchetto

Specifiche Generali Modbus

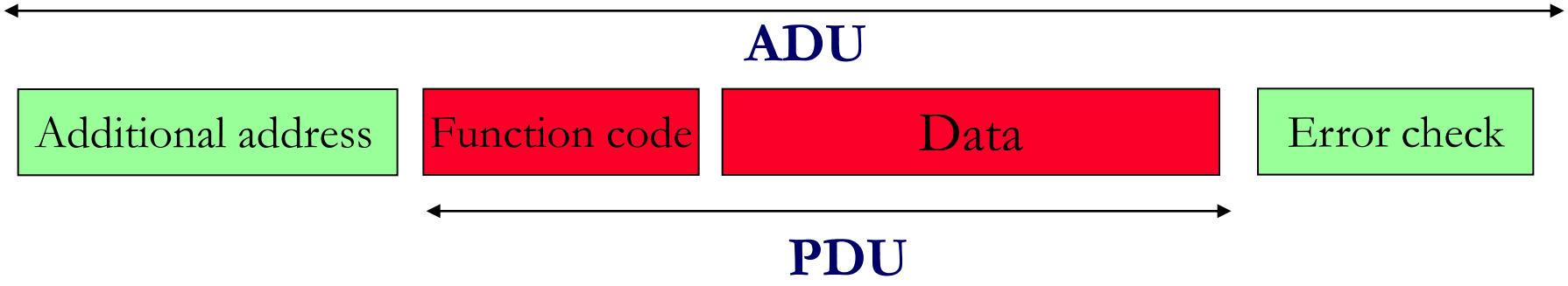
- Protocollo di comunicazione dello strato APPLICAZIONE, livello 7 ISO/OSI
- Realizza una comunicazione **client/server** tra i dispositivi connessi su tipi di reti e bus **diversi**



Architettura Network



Modbus Frame

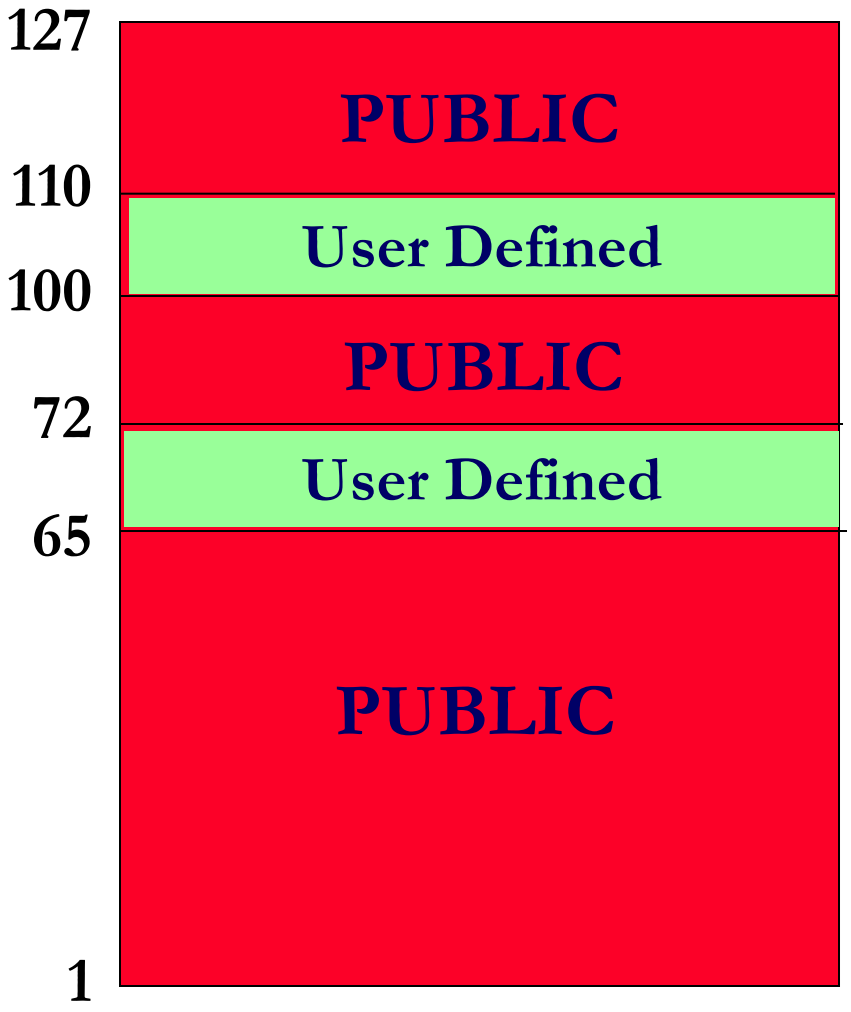


- Il protocol data unit (PDU) presenta una struttura semplice e indipendente dagli strati di comunicazione sottostanti
- I campi dell' application data unit (ADU) subiscono delle variazioni in funzione della rete o del bus utilizzato
- Dimensione massima del PDU = 253 bytes

FUNCTION CODE

- Indica al dispositivo indirizzato quale tipo di azione deve svolgere
- Dimensione : 1 byte
- Codifiche valide : da 1 a 255 decimale (da 128 a 255 riservate per le eccezioni)
- Si dividono in tre categorie:
  - Public Function codes
  - User-Defined Function Codes
  - Reserved Function Codes

Function Code - Categorie

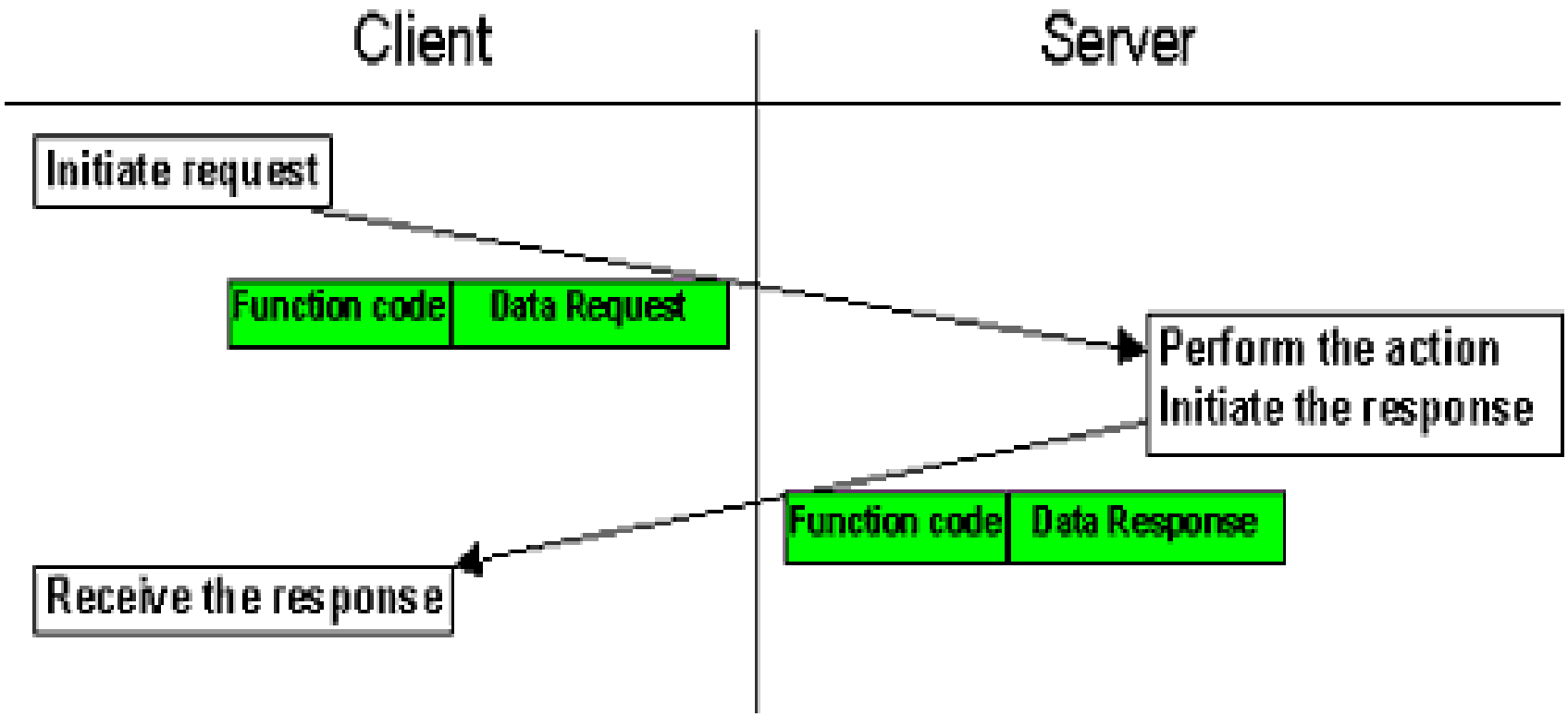


- Public:
  - Include le codifiche riservate
  - Garantiti, unici
  - Ben documentati
  - Riconosciuti dalla modbus.org
- User-Defined:
  - Personalizzati dall'utente

DATA

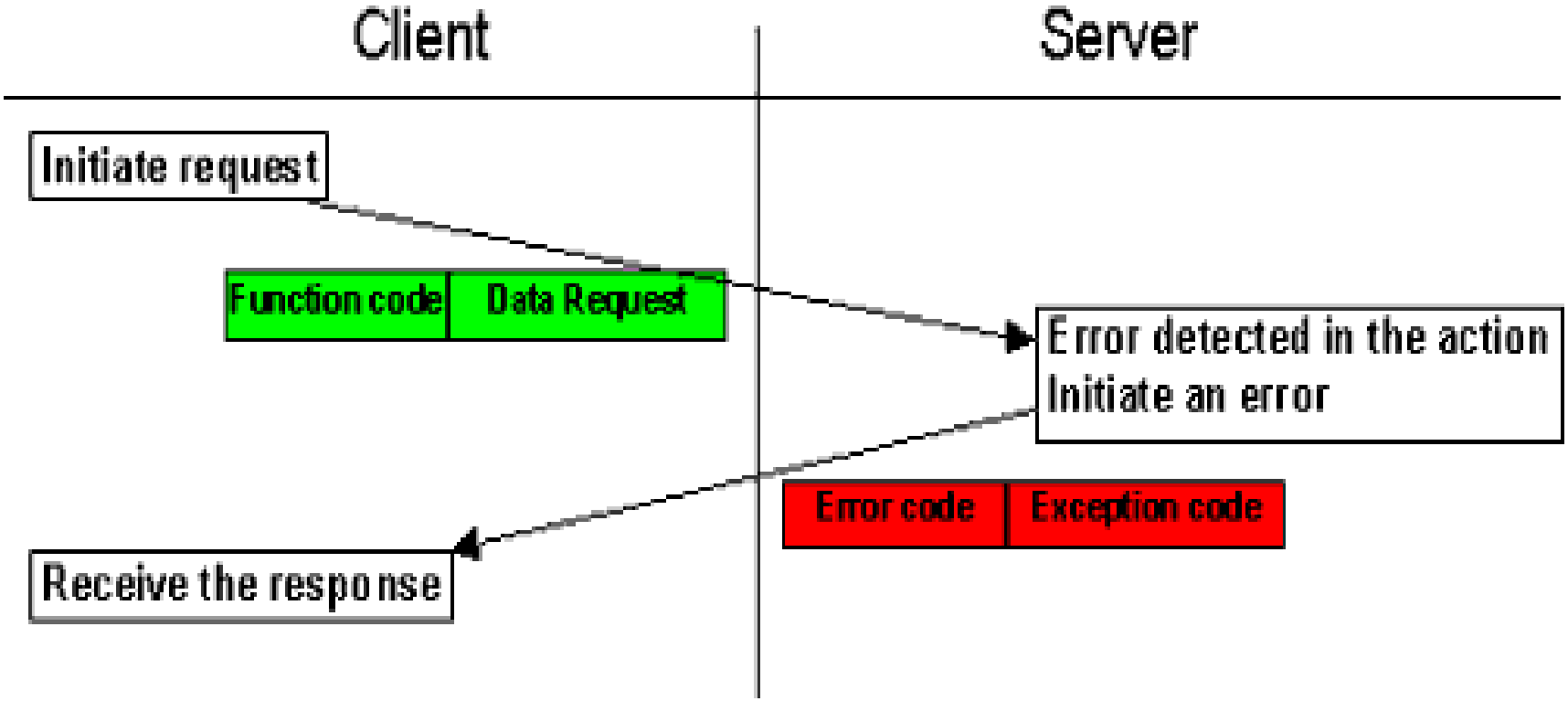
- Nei messaggi di richiesta, se esiste, contiene informazioni aggiuntive al function code.
- Nei messaggi di risposta contiene i dati richiesti dal client
- Dimensione: 0-252 bytes (msg di richiesta)
  1-252 bytes (msg di risposta)

Transazione (1)



senza errori

Transazione (2)



con errori

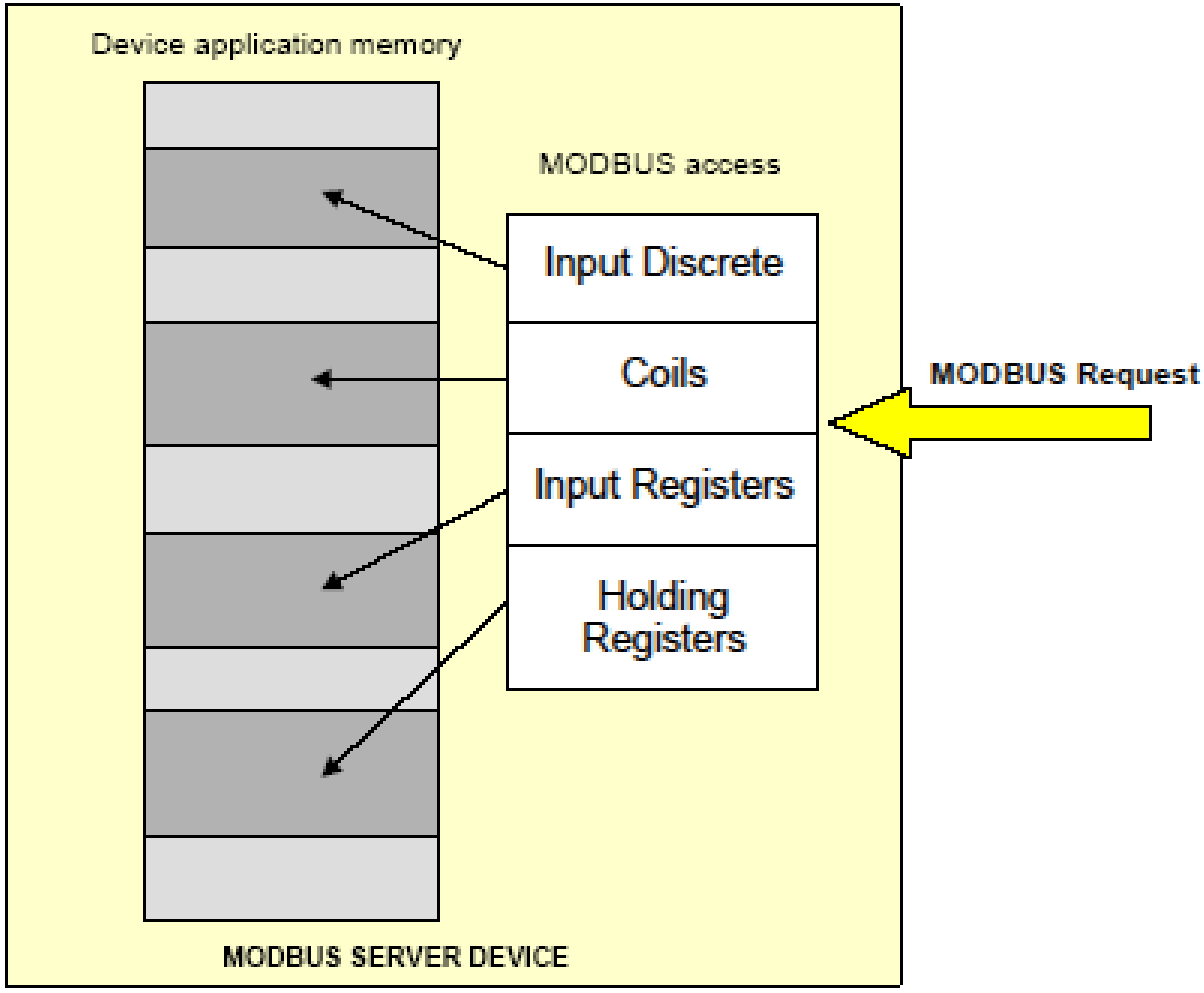
Tipo di dati

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.

Data Model

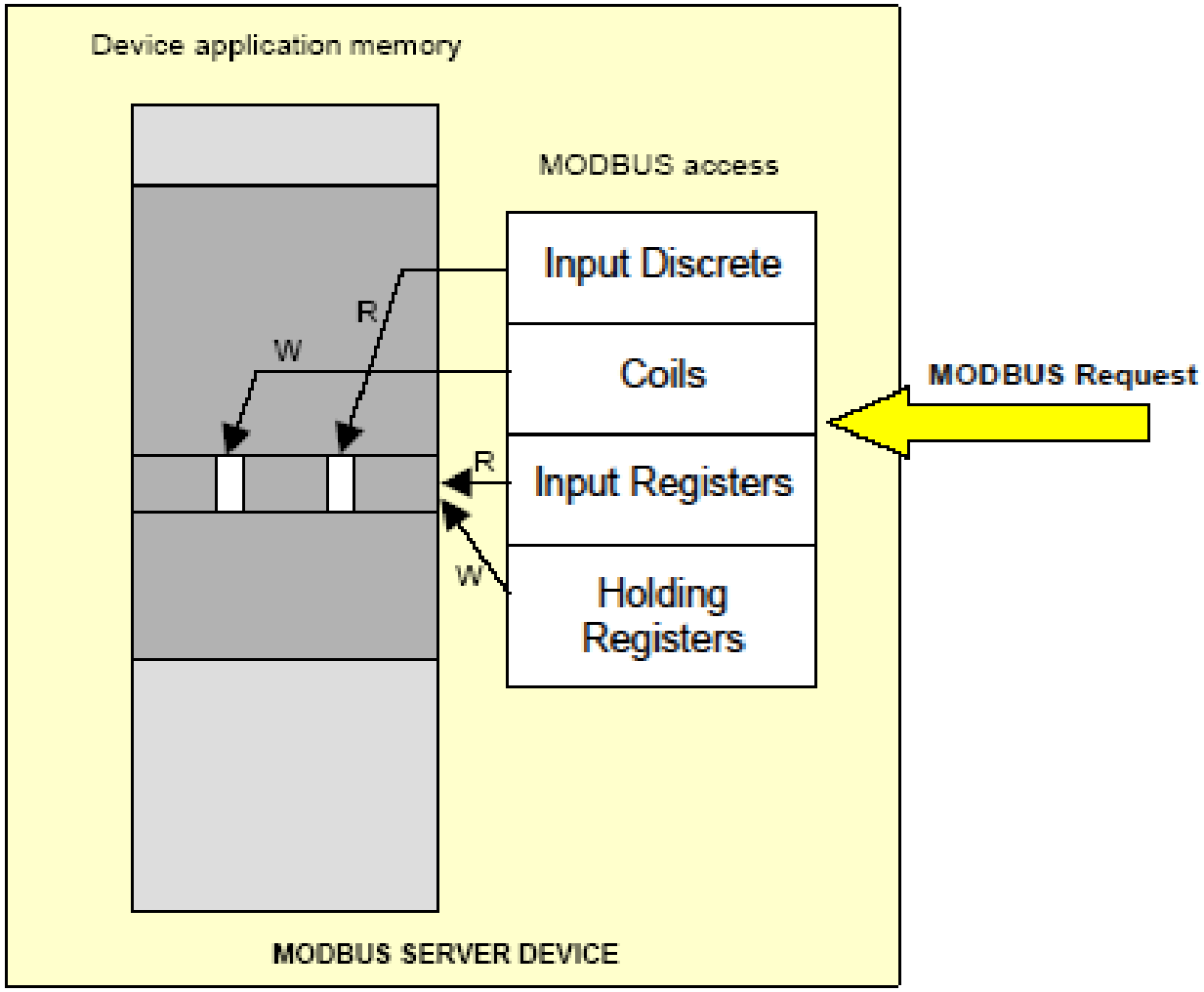
- 4 tipi di dati
- Per ogni tipo di dati il protocollo permette la gestione di 65536 oggetti
- Differenza tra mappatura fisica e logica
- Nel frame compare il riferimento logico agli oggetti
- Link tra memoria fisica del dispositivo e riferimento logico

Memoria dei dispositivi (1)



Organizzazione fisica dei dati a blocchi separati all'interno di un dispositivo

Memoria dei dispositivi (2)

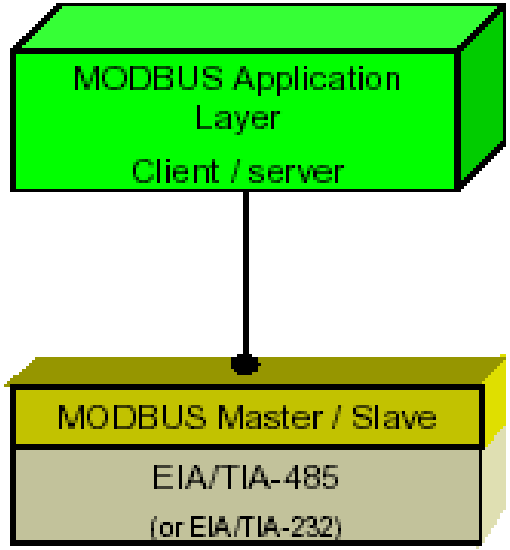


Organizzazione fisica dei dati in un unico blocco all'interno di un dispositivo

MODBUS OVER SERIAL LINE

Modbus over Serial Line

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)



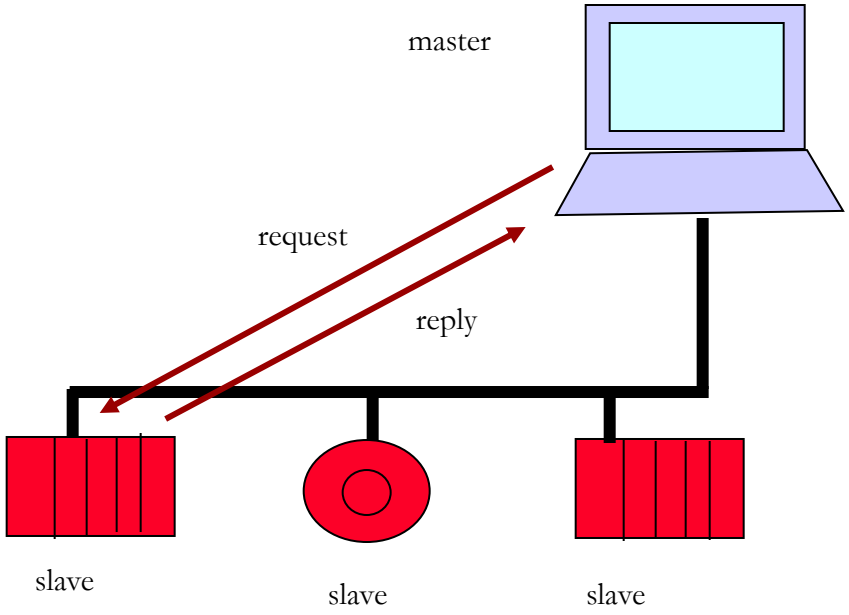
Data link Layer si divide in

- protocollo master-slave
- modalità di trasmissione RTU o ASCII

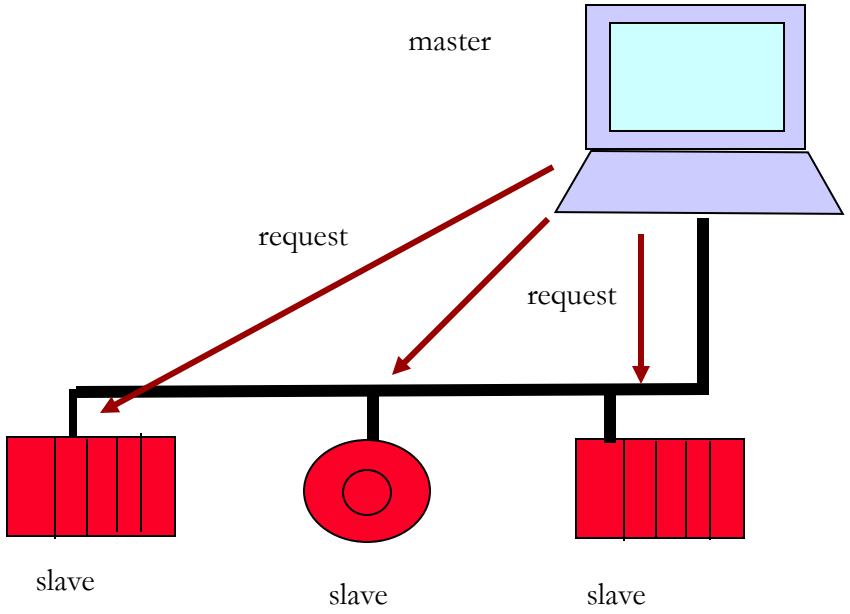
Protocollo Master/Slave

- In un dato intervallo di tempo si ha:
  - Un solo master
  - Massimo 247 slaves
  - Comunicazione iniziata dal master
    - Unicast
    - Broadcast
  - Nessuna comunicazione tra gli slaves

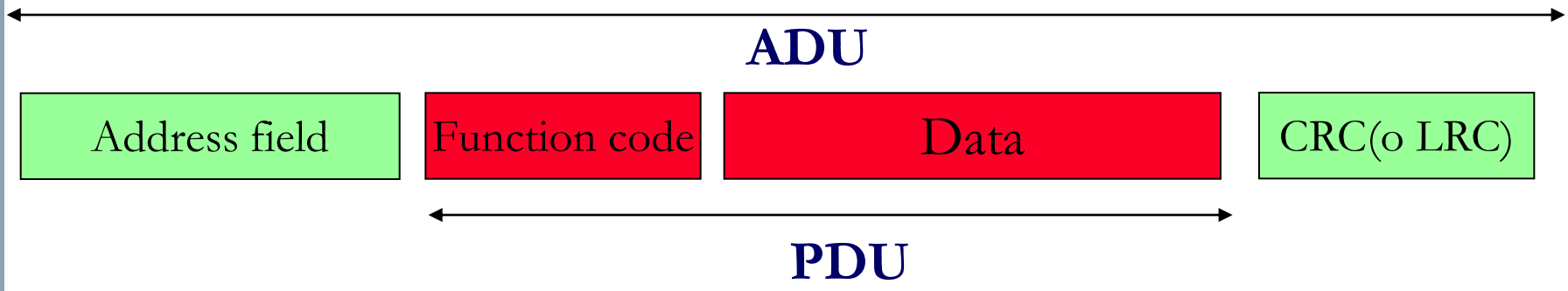
Unicast



Broadcast



ADU nel Serial Line



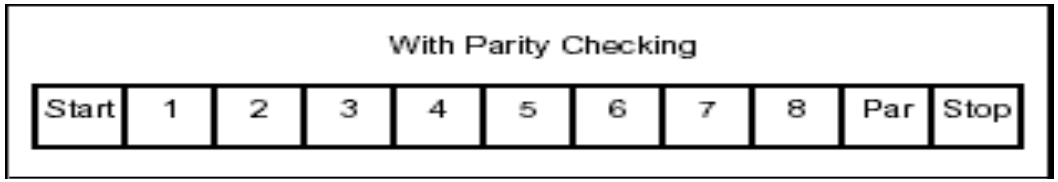
Address Field

- 0 per broadcast
- 1-247 dec per gli slave
- 248-255 dec riservato

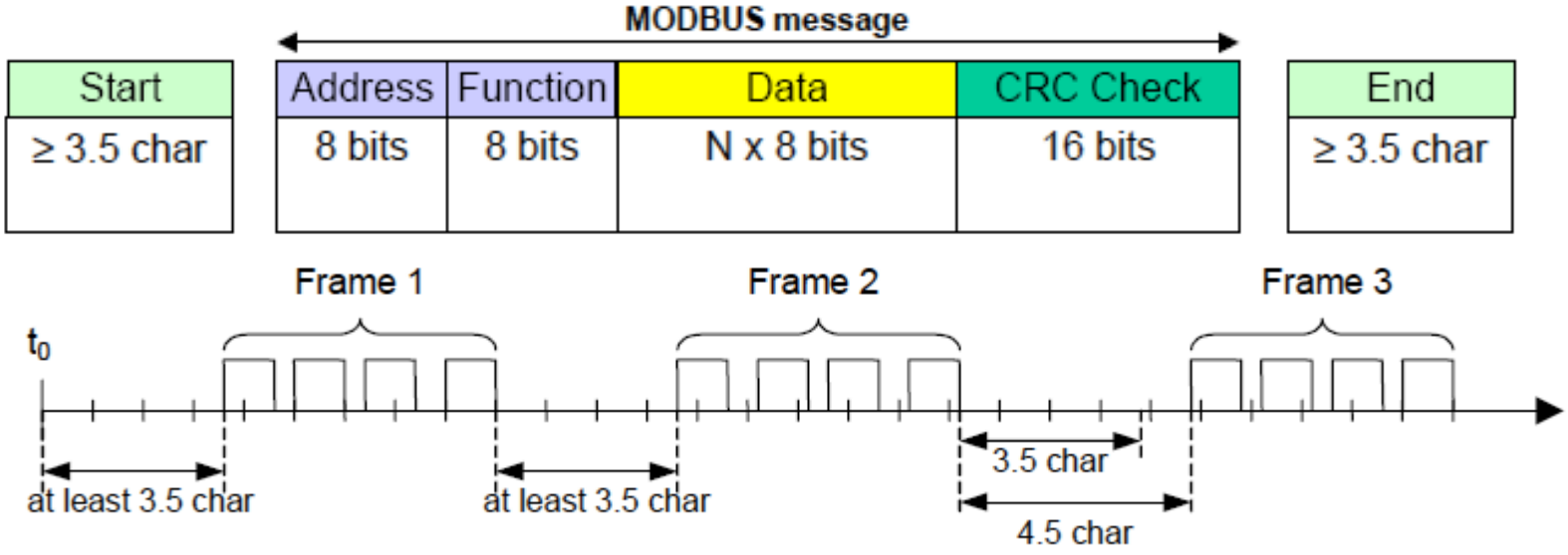
Modalità di trasmissione RTU (remote terminal unit)

- Ogni byte di 8-bit contiene due caratteri esadecimali di 4 bit
- Throughput di dati maggiore di ASCII per la stessa percentuale di bit/secondo

- Carattere RTU



- Svantaggio: intervallo di tempo tra due frame



CRC = Ciclical Redundancy Checking

Si applica solo ai bit di dati

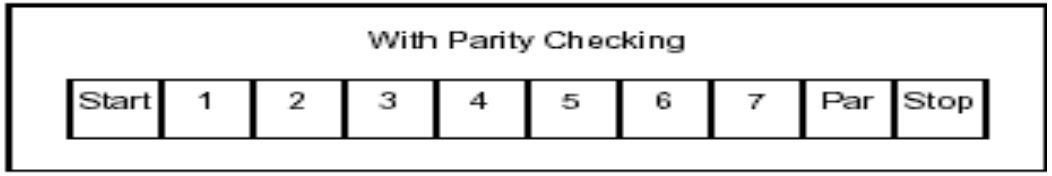
1. Pre-loading di un registro a 16 bit con tutti 1
2. XOR tra i bit del registro e gli 8 bit di dati
3. Il risultato shifta nei LSB bit, i MSB vengono messi a zero
4. Se il LSB è 1 -> XOR con un valore fissato, altrimenti no
5. Si ripete tutto dal passo 2 per 8 shift consecutivi
6. Dopo 8 shift, XOR tra il byte successivo e il contenuto del registro
7. Si ripete tutto per altre 8 volte
8. Il risultato è il CRC

Modalità di trasmissione ASCII

- Ogni byte di 8 bit è codificato con 2 caratteri ASCII

  es: 0x5B è codificato con due caratteri 0x35 e 0x42 (0x35 = "5" e 0x42 = "B" in ASCII)

- Carattere ASCII



- Vantaggio: intervallo di tempo fino a 1 sec tra due caratteri di un messaggio

Start	Address	Function	Data	LRC	End
1 char :	2 chars	2 chars	0 up to 2x252 char(s)	2 chars	2 chars CR,LF

LRC = Longitudinal Redundancy Check

Si applica ai soli bit di dati

1. Si sommano insieme i bytes del messaggio, scartando i riporti
2. Si fa il complemento a due della somma
3. Il risultato è il LRC

Velocità del canale

- La velocità di trasmissione è impostabile
  - Minima : 9600 bps
  - Massima: 115 Kps

MODBUS ON TCP-IP

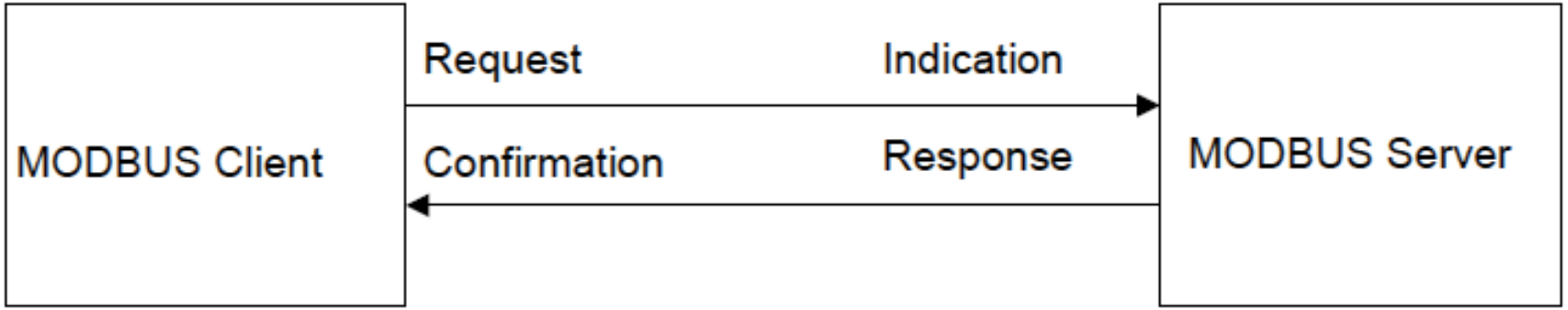
Utilizzato per …

- monitorare e programmare i dispositivi
- far comunicare i dispositivi intelligenti con i sensori
- monitorare i dispositivi di campo con l'uso di PC e HMI
- Scambiare informazioni in tempo reale tra
  - Due applicazioni
  - L'applicazione di un dispositivo e un altro dispositivo
  - Applicazioni HMI/SCADA e i dispositivi
  - Un PC e un programma di dispositivo provvisto di servizi on-line

Vantaggi

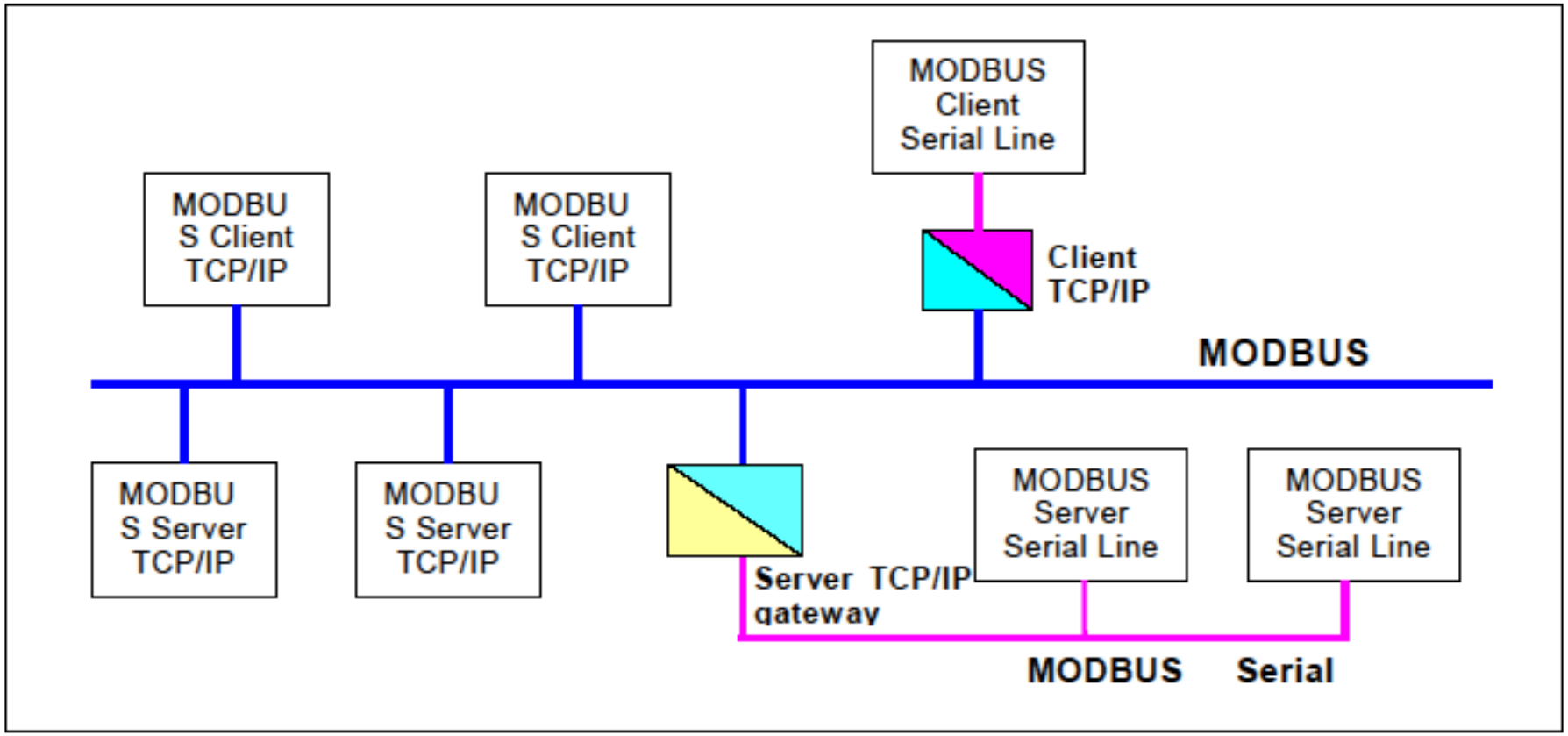
- Semplicità, Bassi costi di implementazione
- Hardware minimo richiesto
- Facile da sviluppare
- Un dispositivo remoto può essere acceduto da **qualunque parte del mondo** attraverso Internet

Modello Client/Server

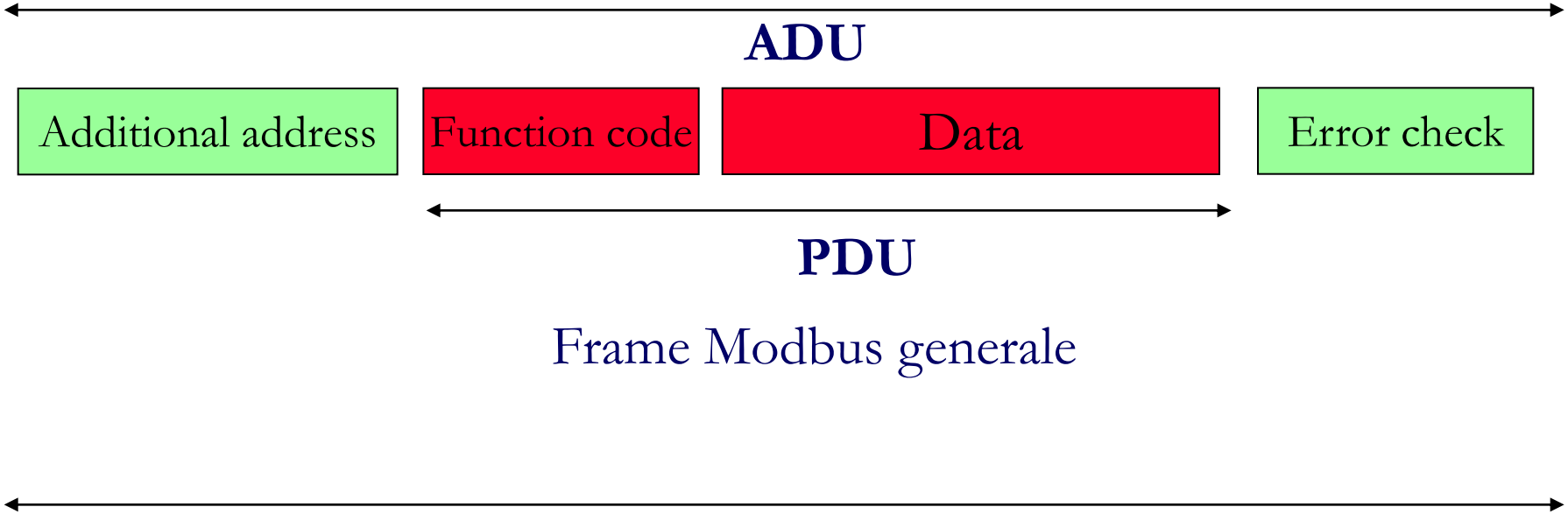


Comunicazione client/server basata su quattro tipi di messaggi

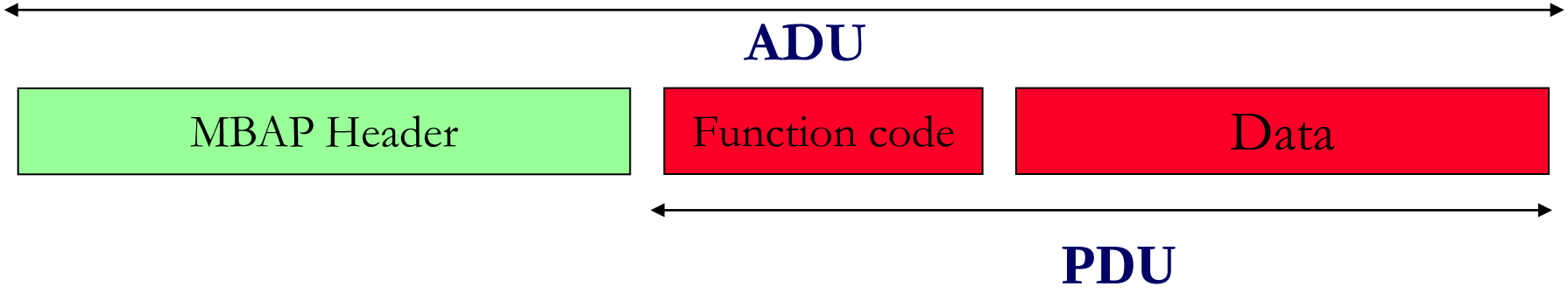
Architettura di comunicazione



ADU nel MB-TCP/IP



Frame Modbus generale

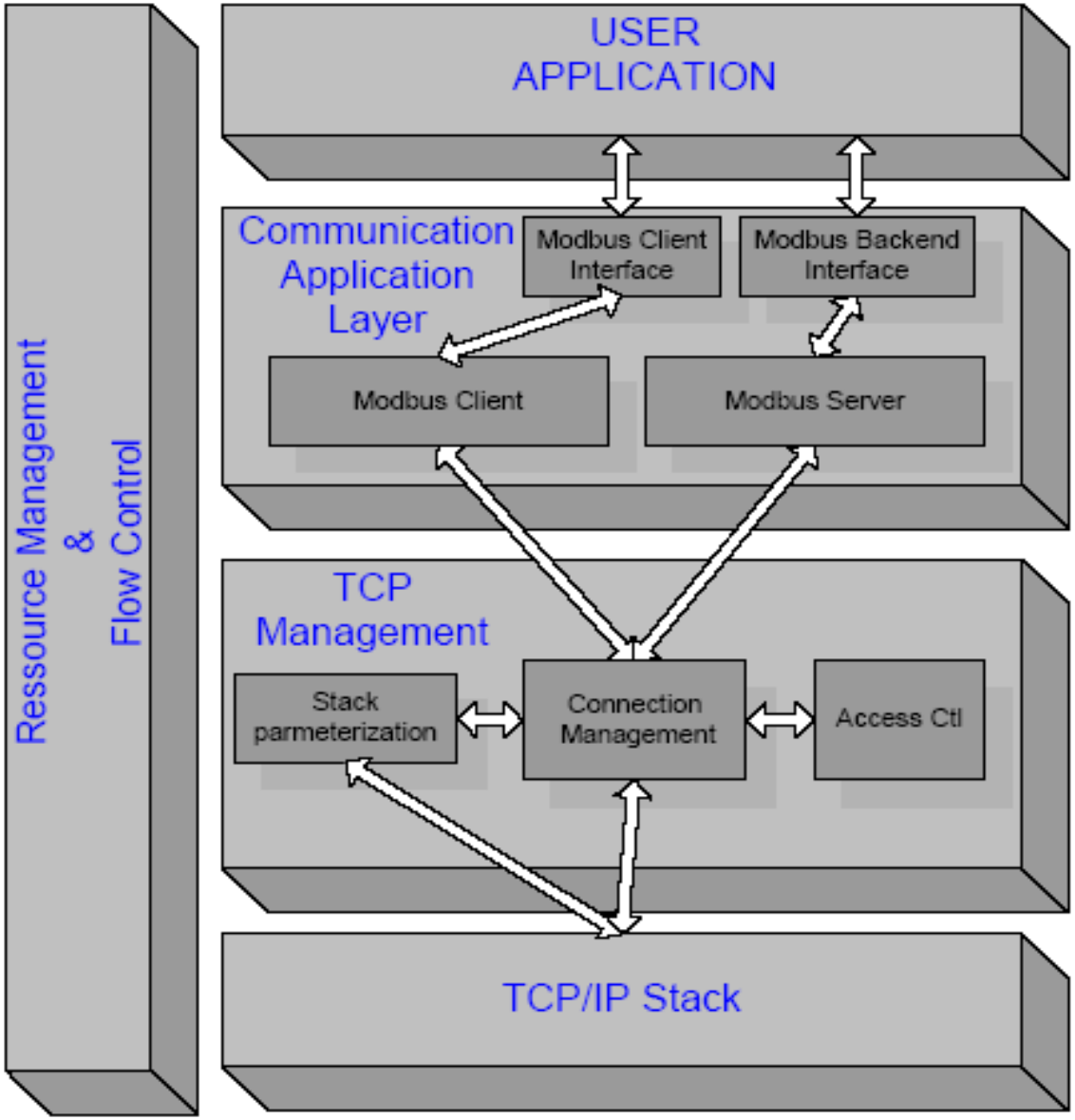


Frame di richiesta/risposta su TCP/IP

MBAP Header

Fields	Length	Description -	Client	Server
Transaction Identifier	2 Bytes	Identification of a MODBUS Request / Response transaction.	Initialized by the client	Recopied by the server from the received request
Protocol Identifier	2 Bytes	0 = MODBUS protocol	Initialized by the client	Recopied by the server from the received request
Length	2 Bytes	Number of following bytes	Initialized by the client (request)	Initialized by the server (Response)
Unit Identifier	1 Byte	Identification of a remote slave connected on a serial line or on other buses.	Initialized by the client	Recopied by the server from the received request

Architettura dei componenti



- MODBUS CLIENT
  - Permette all' applicazione utente di controllare dispositivi remoti
  - Costruisce il messaggio di richiesta
  - Gestisce le transazioni fino alla conferma
- MODBUS SERVER
  - Attiva una serie di azioni locali in base alla richiesta ricevuta
  - Ascolto della porta 502 TCP
  - Costruzione della risposta

Gestione della comunicazione e del flusso di dati

Connection Management

- Stabilisce la comunicazione tra client e server
- Porta 502 TCP dal lato server, porta n>1024 dal lato client
- Due modalità di connessione
  - Manualmente dall' applicazione utente
    - Grande flessibilità
    - buona conoscenza di TCP richiesta
  - Automaticamente dal modulo di connessione
    - Trasparente per l'utente
    - Definisce due gruppi di connessione (proritario e non prioritario)

Access Control Module

- Controlla ogni nuova connessione verificando la lista degli indirizzi IP autorizzati
- In alcuni casi è necessario specificare l'accesso per ogni indirizzo IP remoto
- Gli IP non configurati vengono rifiutati