MODBUS

Introduction

- An industrial protocol - developed by Medicon in 1979 to make communication possible between automation devices

- Originally implemented as an application-level protocol intended to transfer data over a serial
  - Expanded to include implementations over serial, TCP/IP, and the user datagram protocol (UDP).
  - Port 502 reserved in TCP/IP stack
  - Open Protocol

Client Server Model:

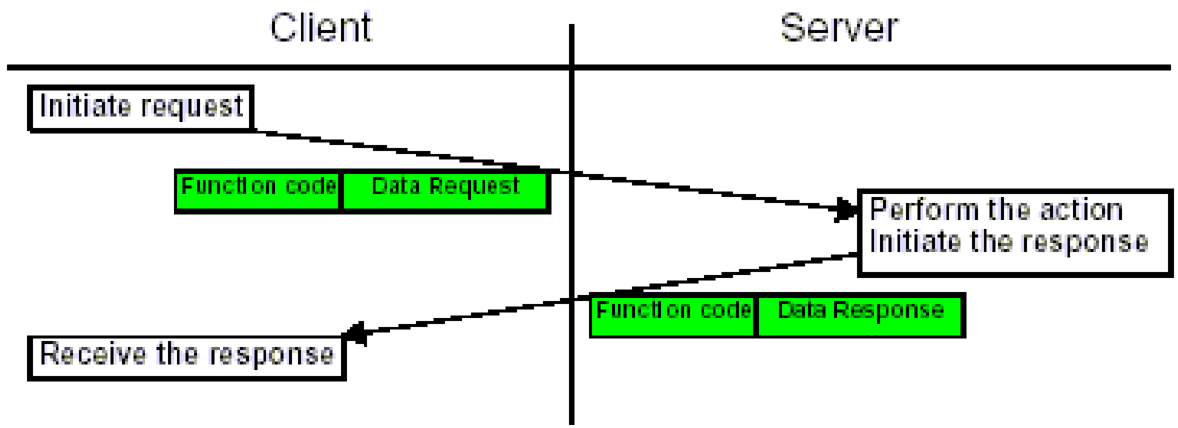


Figure 4: MODBUS transaction (error free)

- If an error

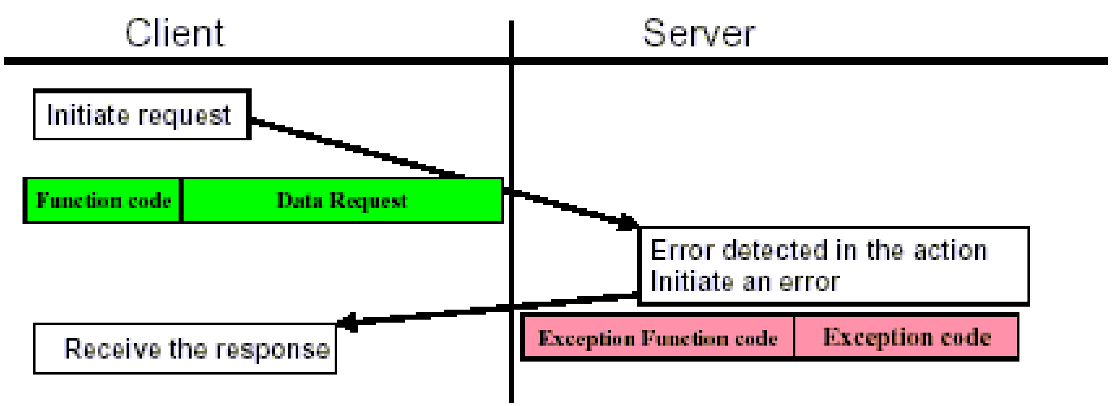
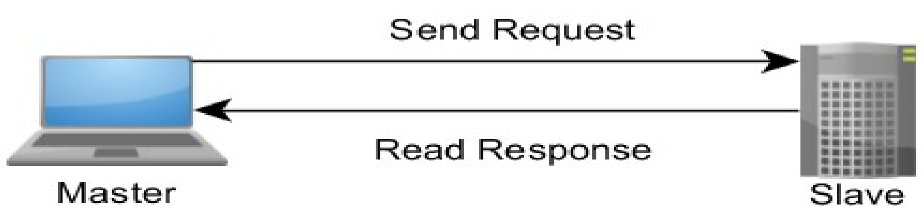


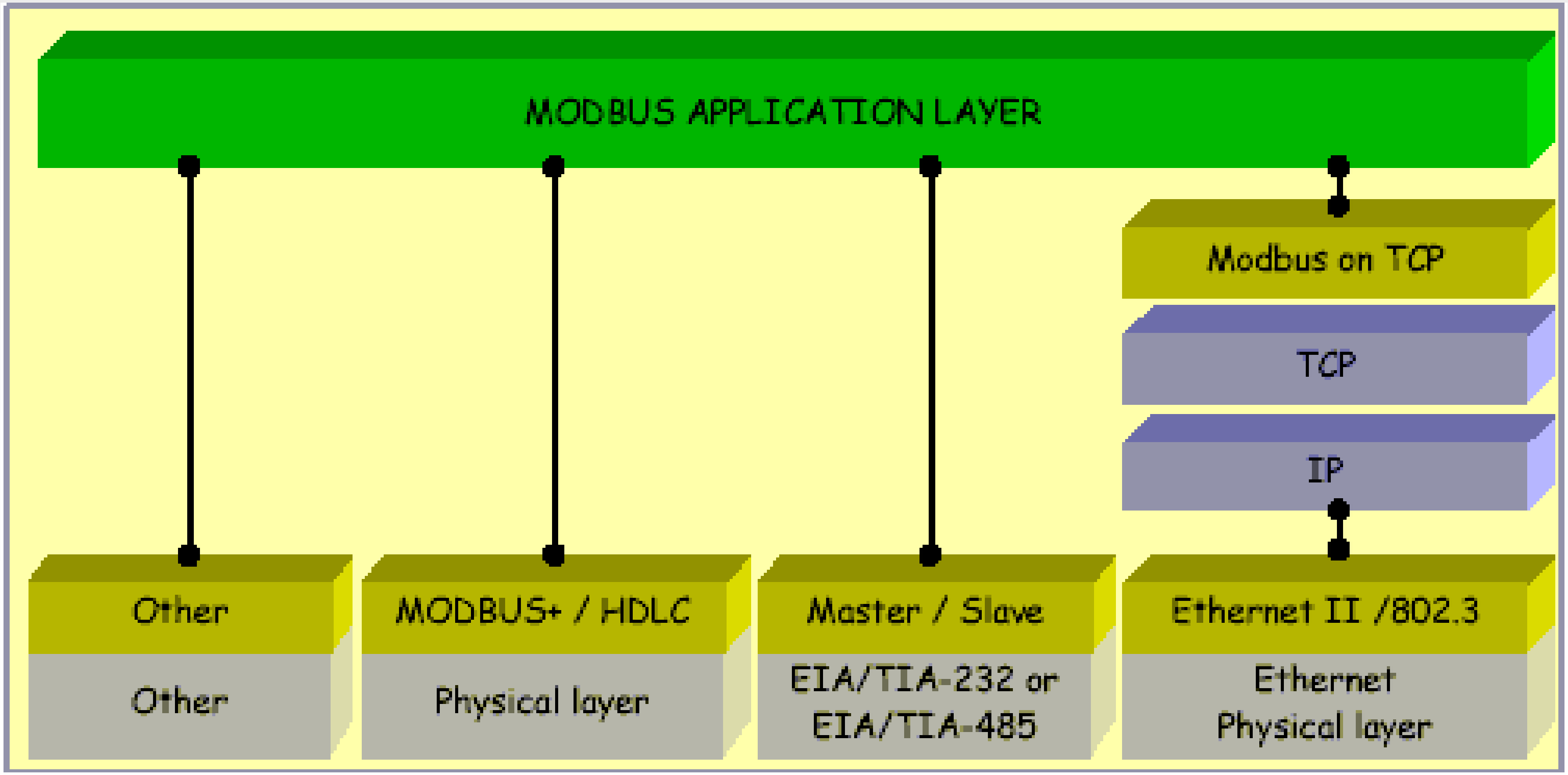
Figure 5: MODBUS transaction (exception response)

Client Server Model:

- Request-response protocol implemented using a master-slave relationship (Client – Server)
  - MODBUS Transaction.
- One device must initiate a request and then wait for a response
  - Initiating device (the master) is responsible for initiating every interaction
  - Typically, master is a Human Machine Interface (HMI) or Supervisory Control and Data Acquisition (SCADA) system and
  - The slave is a sensor, programmable logic controller (PLC), or programmable automation controller (PAC)

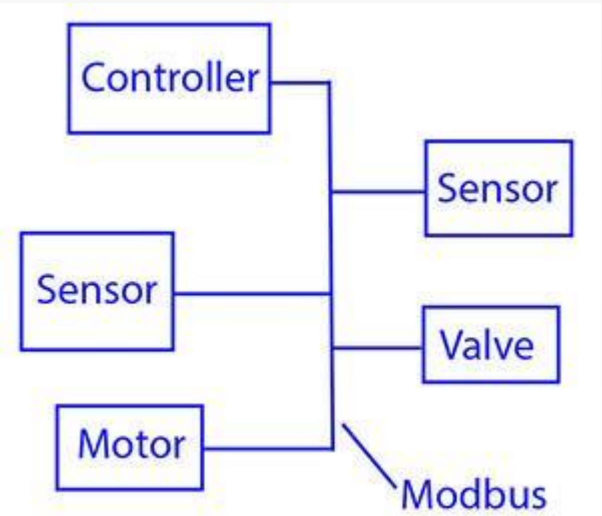


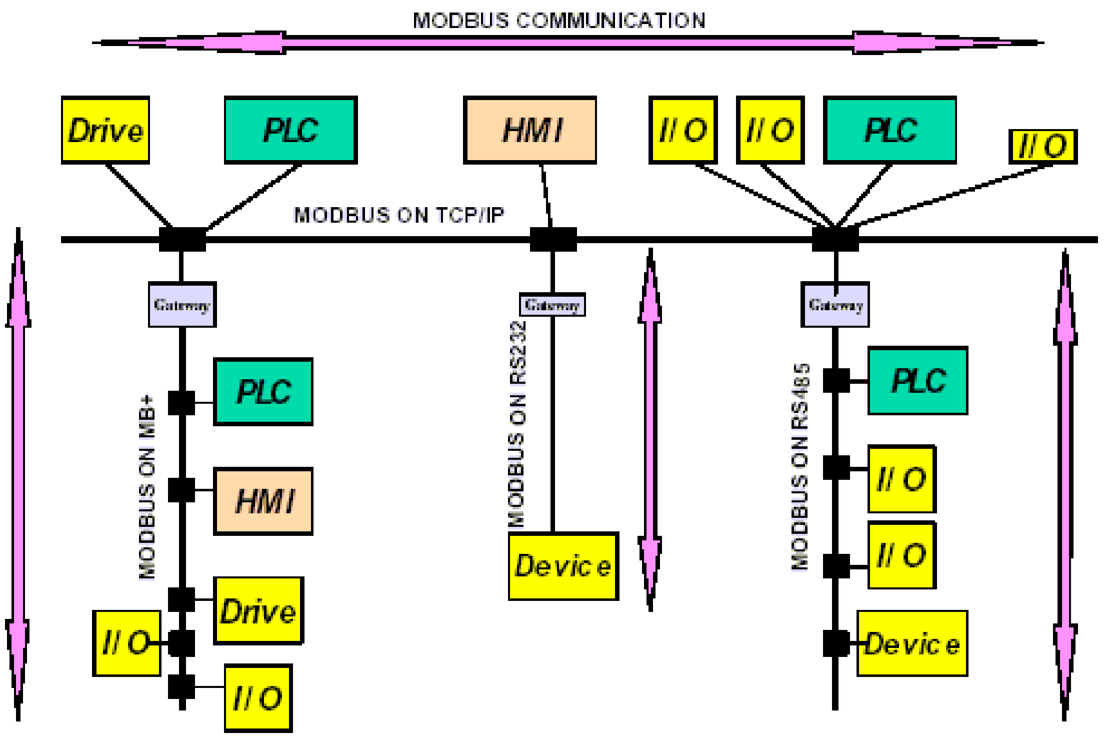
MODBUS Application Layer:



MODBUS on Different Networks

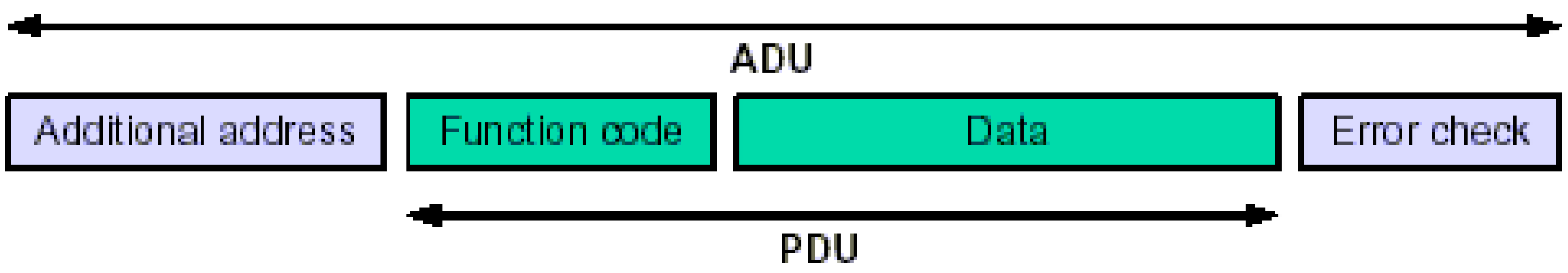
- Allows an easy communication within all types of network architectures.
- Gateways – For communication between several types of buses or network using the MODBUS protocol.





MODBUS PDU/ADU:

- Defined a simple protocol data unit (PDU) independent of the underlying communication layers – 253 bytes
- Mapping of MODBUS protocol on specific buses or network can introduce some additional fields on the application data unit (ADU) – 256 bytes



Interfaces

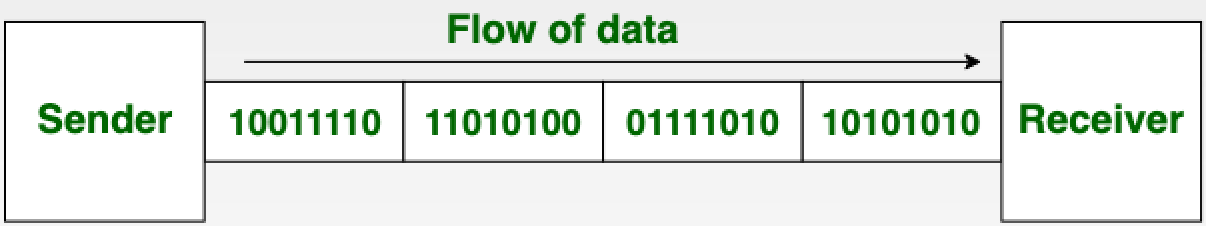
- Transmission of data from the source to a device or from a device to the destination
- Parallel Transmission:
  - Multiple lines carrying bits simultaneously
  - High data rate, but expensive
- Serial Transmission:
  - Bits transmitted serially
  - Synchronous vs. Asynchronous

Serial I/O Protocols

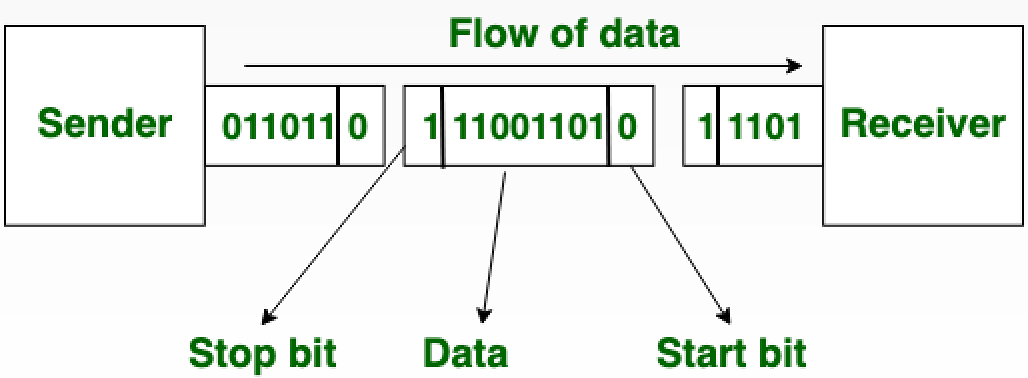
- **Synchronous:**
  A master clock controls the transmission as a continuous stream
- **Asynchronous:**
  Random delays between data pieces

Synchronous	Asynchronous
Requires processing to extract clock	No clock recovery needed
Overhead applies to entire block	One character at a time (8 bits max) 20% overhead/character (1 start and 1 stop bit)
Error detection and correction built into protocol	Error detection possible, correction done separately

Synchronous Vs Asyschronous



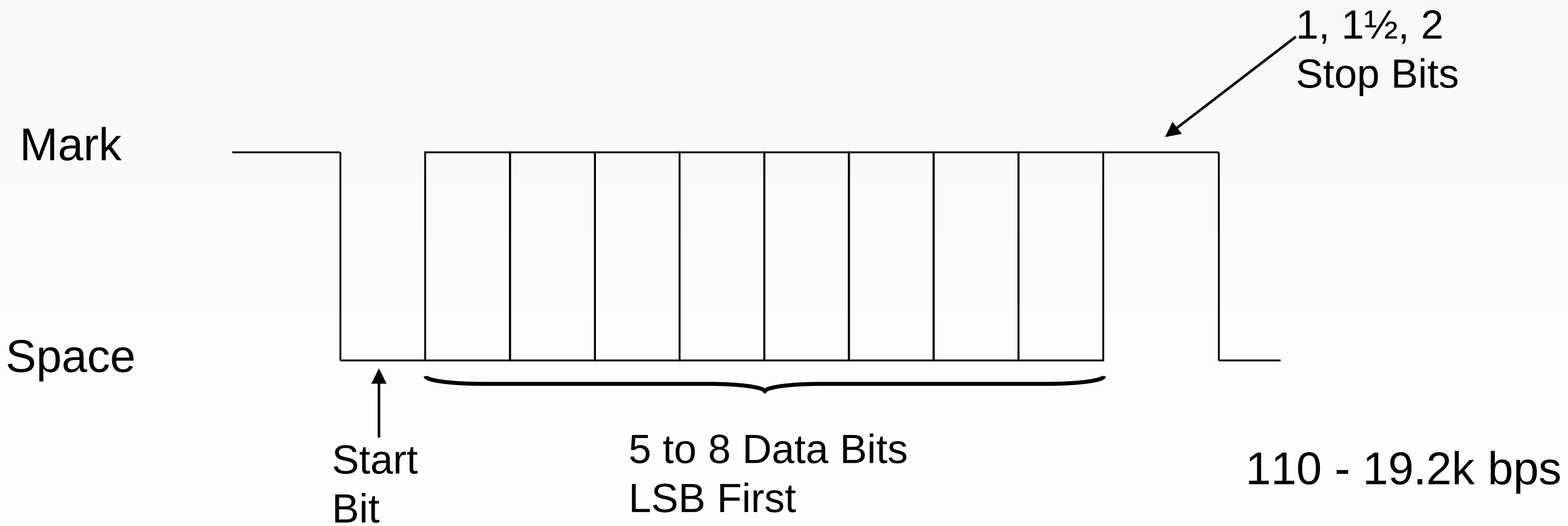
Synchronous Transmission



Asynchronous Transmission

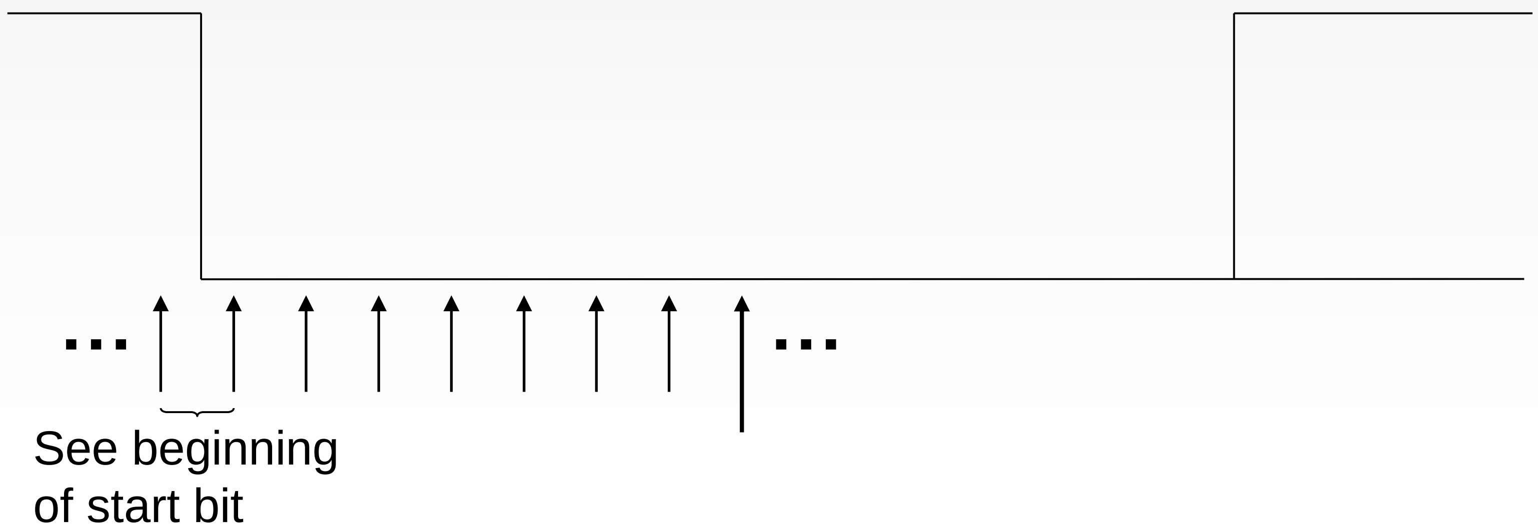
Asynchronous Protocols

- RS-232-C
  - 20MA Current Loop
- RS-422, RS-423, RS-485
  RS: Recommended Standard by EIA
  (Electronic Industries Association)



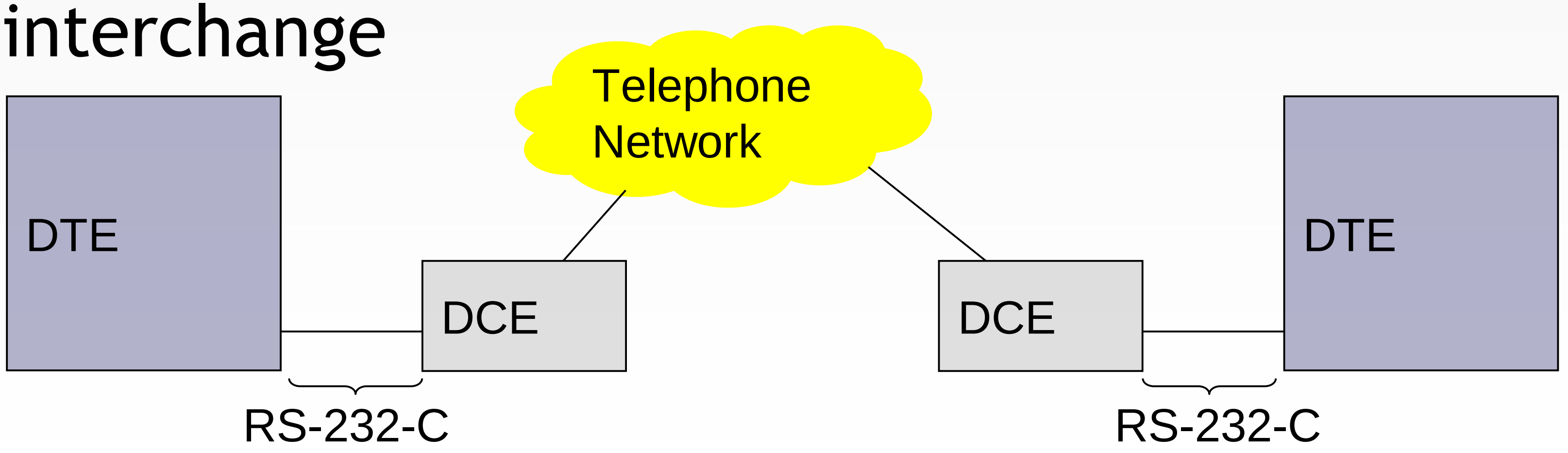
Start and Stop Bits

- Start bit permits local synchronization
- Stop bit provides validity check and the opposite level for the start bit



RS-232-C Interface

- EIA in cooperation with Bell Systems, independent modem and computer manufacturers
- Standard for interface between **Data Terminal Equipment (DTE)** and **Data Communication Equipment (DCE)** employing serial bit interchange



RS-232-C

- Standards contain
  - Electrical signal characteristics
  - Interface mechanical characteristics
  - Functional description of interchange circuits
  - Standard subsets for specific groups of communication systems applications
- Mechanical
  - DB-25 or DB-9 connectors
  - Cable
    - Female connected to DTE, male to DCE
    - Maximum 15 meters

RS-232-C

- Lines/Pins:

1 Shield	Shield	
7 GND	Signal ground	
2 XMIT	Transmit from DTE to DCE (Modem)	
3 RCV	Receive from DCE (Modem)	
4 RTS	Request to send, from terminal to modem	*
5 CTS	Clear to send, from modem to terminal	
6 DSR	Data set ready, from modem to terminal	Data set (modem) online
20 DTR	Data terminal ready, from term. to modem	Tie to power
22 RI	Ring indicator, from modem to terminal	“Say hello!”
8 CD	Carrier Detect, from modem to terminal	“I hear the other end”

RS-232-C

- Electrical Problems:
  - ±12V supply needed, inconvenient
  - Cable capacitance: Maximum 50 ft if cable is 40-50pF/ft!
  - Ground reference
    - System has poor common-mode noise rejection
    - Cross-talk and increase of bias distortion
    - Especially bad if clock lines used (SYNC)
  - Not suitable for long distances

-

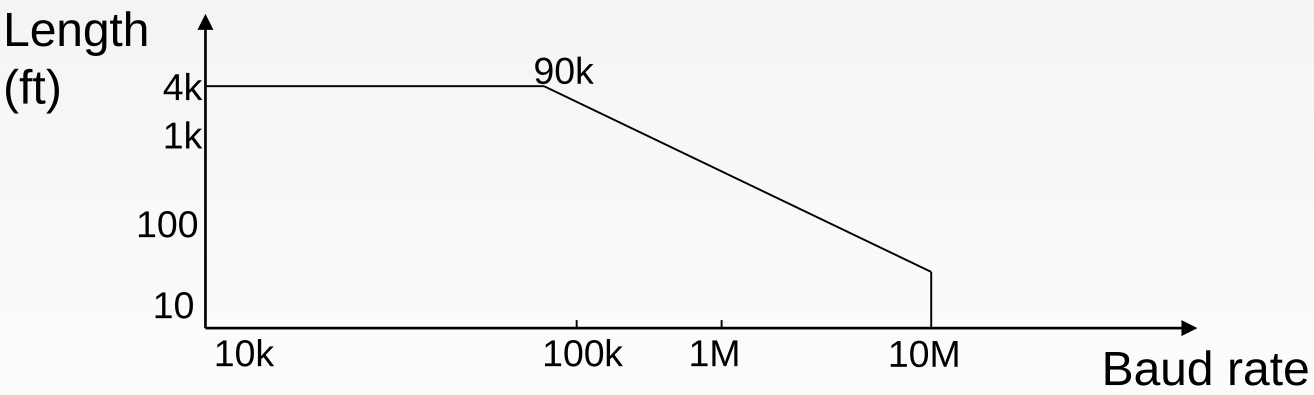
- Motivation for new standards RS-422, 423

RS-423

- Created for transition from RS-232 to RS-422
- Uses unpopular 37-pin connectors per RS-449
- Unbalanced like RS-232-C
- Valid margins: +2V/+6V and -2V/-6V
- For less than 20kbps

RS-422

- Fully balanced, differential inputs
- Supports data rates ≥ 20kbps



- Using 24G Twisted-pair, 100Ω load
  - Amplitude drop less than 6dB
  - Rise time less than ½ bit time

RS-485

- Like 422, 485 is also balanced
- 485 handles multiple drivers and receivers
- Better noise rejection
- Sensitivity of ±200mV in receivers

Distributed Network Protocol (DNP3.0)

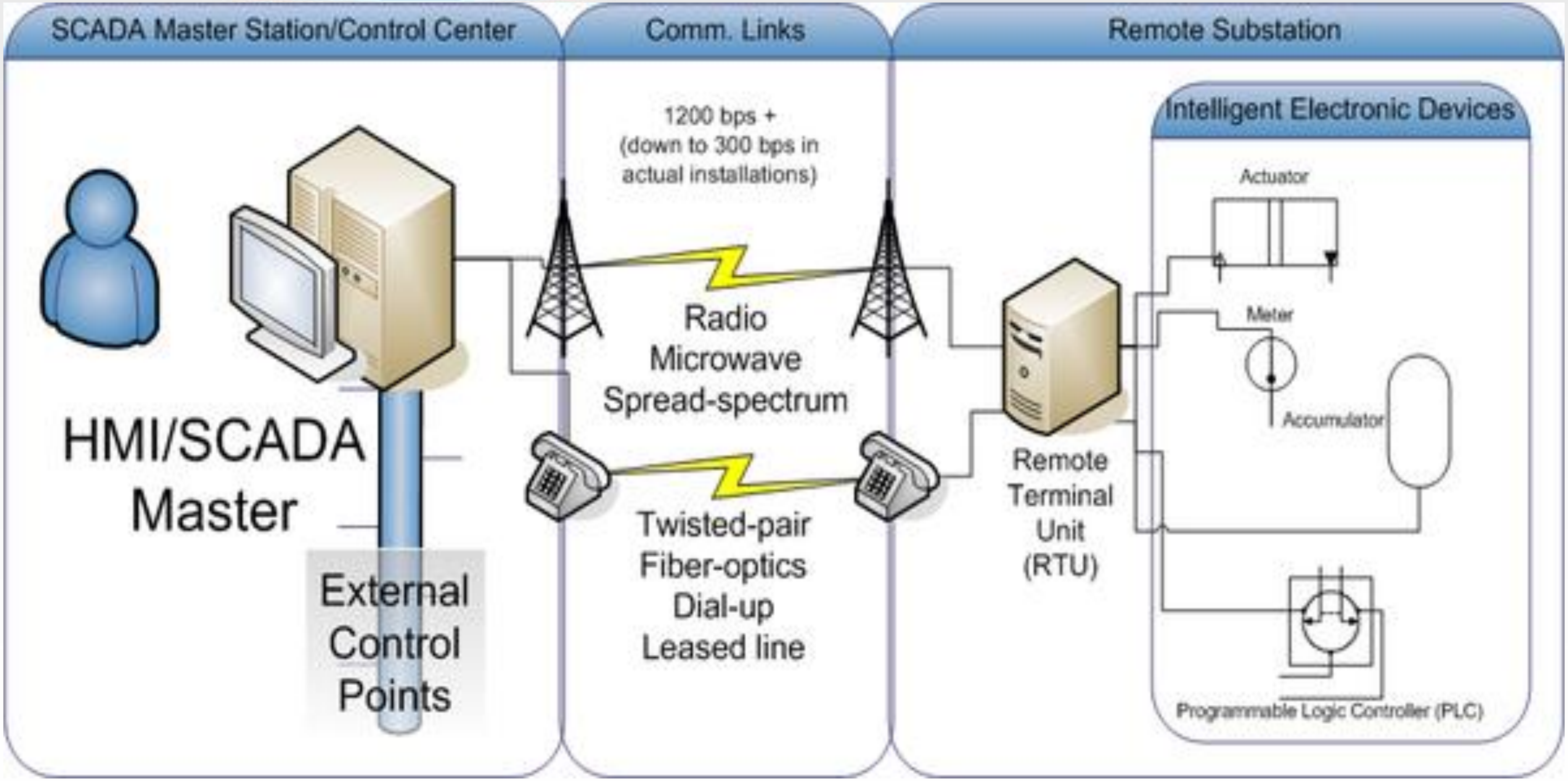
DNP3.0/IEEE Std 1815™

- DNP was originally developed by GE-Harris Canada in 1990 and released in 1993
- Now managed by the DNP Users Group: http://www.dnp.org
- The DNP Users Group includes master station, RTU and IED vendors, and representatives of the electric utility and system consulting communities.

Introduction

- DNP3 used for communications between SCADA masters (control centres) and remote terminal units (RTUs) and/or intelligent electronic devices (IEDs)
- DNP: Distributed Network Protocol
- SCADA: Supervisory Control And Data Acquisition
- Protocol defined in “Basic 4” document set from DNP Users Group
- Based on IEC 60870-5.
- International counterpart: IEC 60870-5-101

Overview



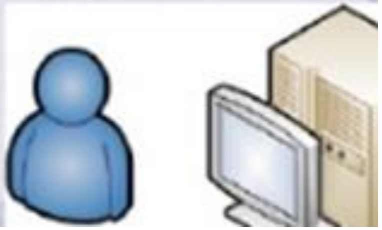
SCADA Master Station/Control Center
Comm. Links
Remote Substation
1200 bps +
(down to 300 bps in
actual installations)
Intelligent Electronic Devices
Actuator
Meter
Accumulator
Radio
Microwave
Spread-spectrum
HMI/SCADA
Master
Twisted-pair
Fiber-optics
Dial-up
Leased line
Remote
Terminal
Unit
(RTU)
External
Control
Points
Programmable Logic Controller (PLC)

Overview

- SCADA Master Stations/Control centres
  - Connected to HMI and other control centres
- Remote terminal units
  - Interface between IEDs and master stations
  - May exhibit limited autonomous control
- Intelligent electronic devices
  - Sensors and meters
  - Relays and other actuators
  - Programmable Logic Controllers: PLCs

SCADA Master:

- Control centre from which multiple substations or other remote installations are controlled and monitored
- Connected to other control centres using ICCP, a separate protocol
- Interfaces with human through HMI (Human-Machine Interface), which may be local or remote.
- Connected to RTUs and/or IEDs



ICCP: Inter-Control Centre Communications Protocol

RTU:

- Remote Terminal Unit
- Appears as IED to SCADA master when DNP used for communications
- Manages multiple actual IEDs
- Attached IEDs referenced using absolute addressing scheme
- Addresses only have meaning to SCADA master

Remote Terminal Unit

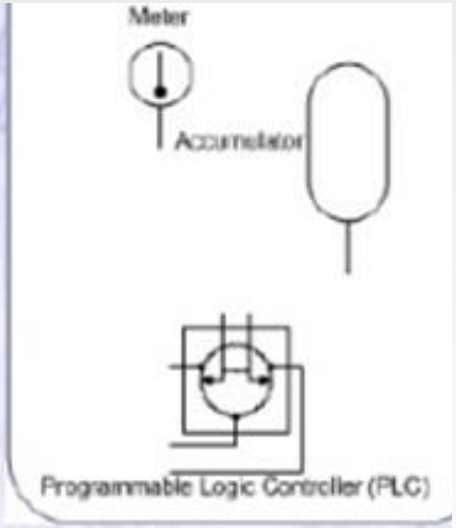
Sample RTUs:

Radio RTU

Cellular RTU

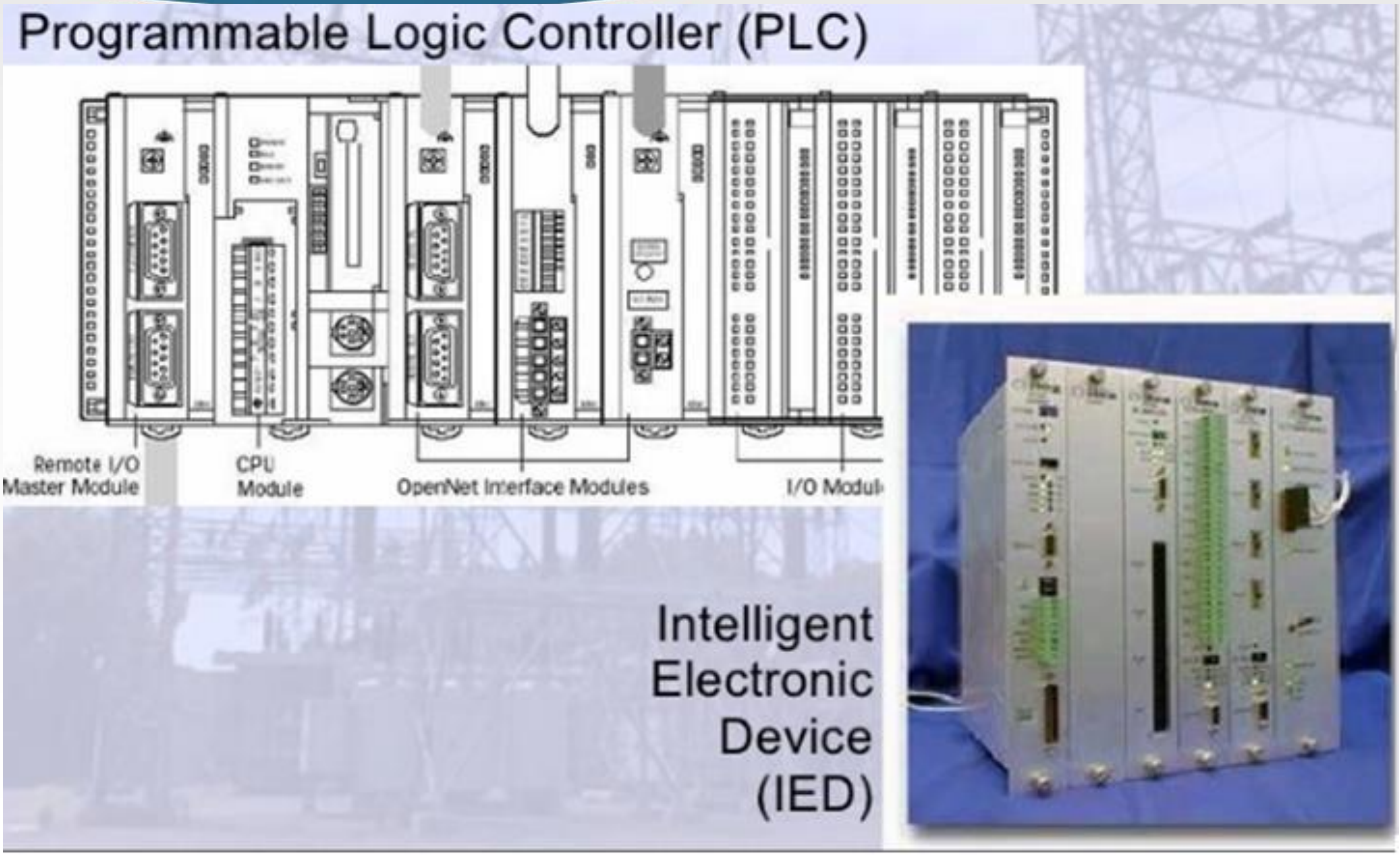
Serial RTU

IEDs:



- Intelligent Electronic Device
- May be data acquisition device only
- May be responsible for control
- Possible inputs: configuration, setting, and command data
- Possible outputs: values, conditions, status, and results
- May be PLCs programmed with ladder logic

Sample IEDs:


Programmable Logic Controller (PLC)
Remote I/O
Master Module
CPU
Module
OpenNet Interface Modules
Intelligent
Electronic
Device
(IED)

Parameters Monitored/Measured:

- IEDs and RTUs can control and monitor a variety of physical processes and other information:
  - Accumulate measurements like kilowatt hour consumption
  - Monitor voltage and current
  - Monitor temperatures (useful for automatically controlling tunnel fires)
  - Switch electrical breakers on and off
  - Transfer configuration files to/from SCADA master

DNP 3.0 Protocols Standard:

- The DNP3 protocol standard defines several aspects of SCADA Master-RTU/IED communications:
  - Frame and message formats
  - Physical layer requirements
    - 1200 bps+
    - Busy link indicator for collision avoidance
  - Data-link layer behavior
    - frame segmentation
    - Transmission retry algorithm
  - Application layer
    - file transfer
    - time synchronization
    - start/stop service

DNP 3.0 Layers

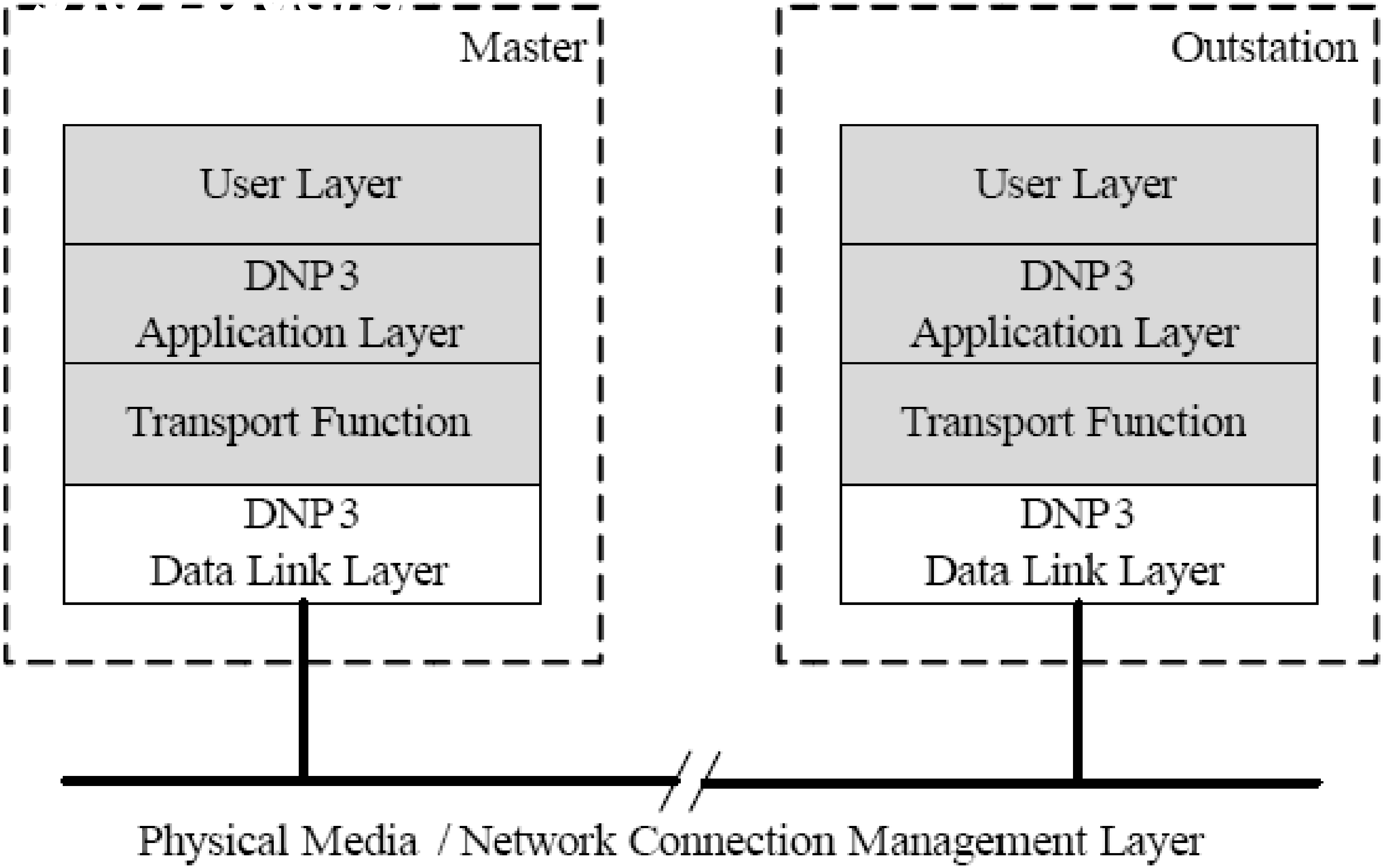
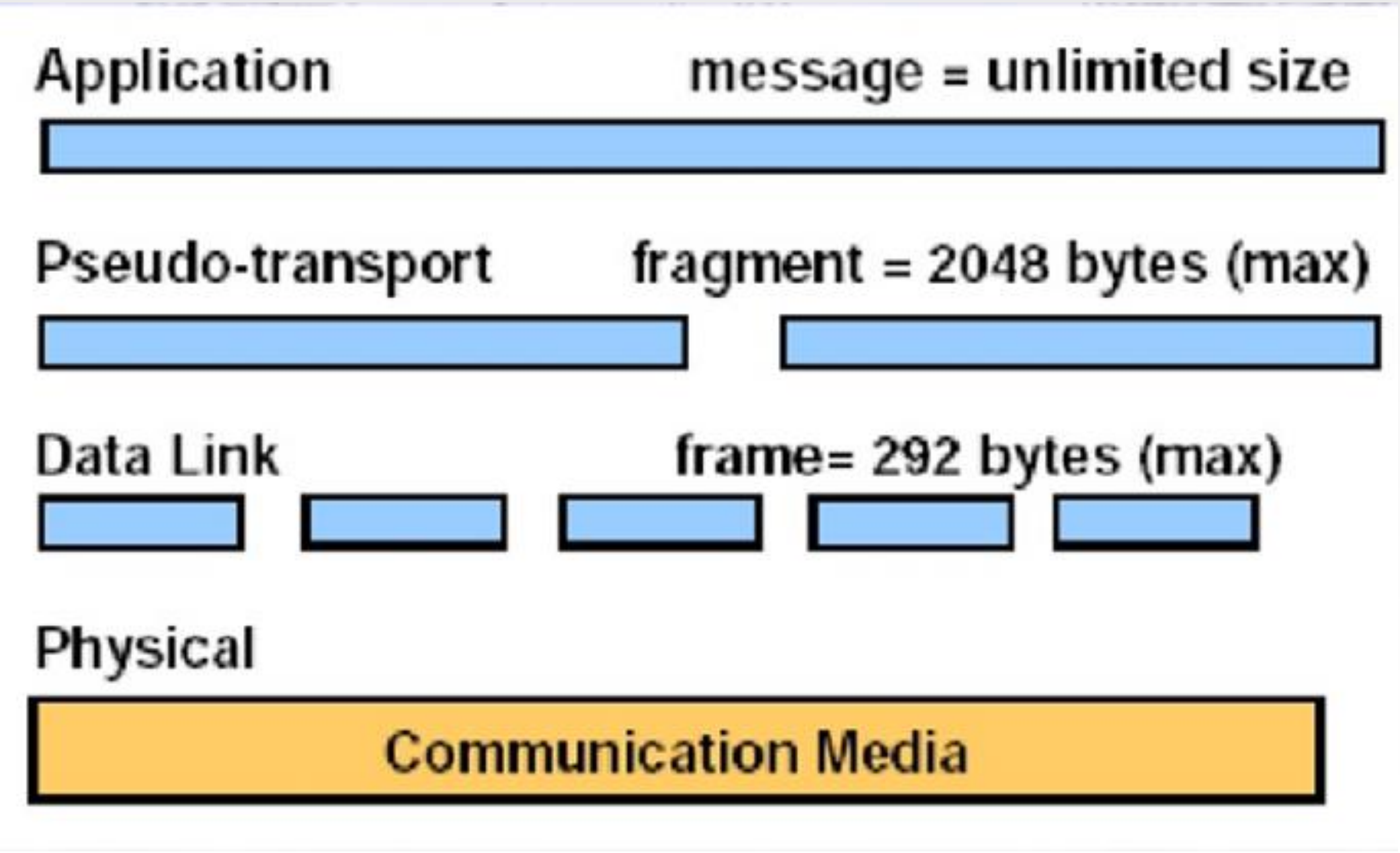


Figure 9-1—DNP3 protocol stack

Message Formats:



DNP 3.0 Capabilities:

- DNP3 can request and respond with multiple data types in single messages
- Response without request (unsolicited messages)
- It allows multiple masters and peer-to-peer operations
- It supports time synchronization and a standard time format
- It includes only changed data in response messages

Benefits of DNP3.0:

- Interoperability between multi-vendor devices
- Fewer protocols to support in the field
- Reduced software costs
- No protocol translators needed
- Shorter delivery schedules
- Less testing, maintenance and training
- Improved documentation
- Independent conformance testing
- Support by independent users group and third-party sources (e.g. test sets, source code)

Controller Area Networks (CAN)

What is CAN?

Controller Area Network:

- **Two-wire, bidirectional serial-bus communication method**
- **Originally developed in the mid 1980s by Bosch for automotive use**
- **Main design objective: economical solution for implementing high-integrity networking in real-time control applications**
- **Now standardized internationally:**
  - CAN 2.0A: ISO11519 — low speed
  - CAN 2.0B: ISO11898 — high speed
  - CAN Validation: ISO16845
- **Usage**
  - Many current and potential non-automotive application opportunities

Non-automotive CAN Applications

- Electronically controlled production and packaging equipment
  - Machine tools; machines for molding, weaving, knitting, and sewing; systems for folding and wrapping; etc.
- Industrial freezers, printing machines
- Ships, locomotives, railway systems
- Farm and construction machinery
- Semiconductor manufacturing equipment
- Building automation: HVAC systems, elevators, etc.
- Hospital patient-monitoring systems

Many others:

For Details/Applications: www.canopen.us

Key Reasons for using CAN

- **Reliability**
  - Error-free communication
- **Economy**
  - Low wiring cost
  - Low hardware cost
- **Scalability**
  - Easy expandability
  - Low node-connection costs
- **Availability**
  - More chips with CAN hardware
  - More off-the-shelf tools
  - Higher-level protocols
- **Popularity**
  - Knowledge base expanding

Main Features of CAN

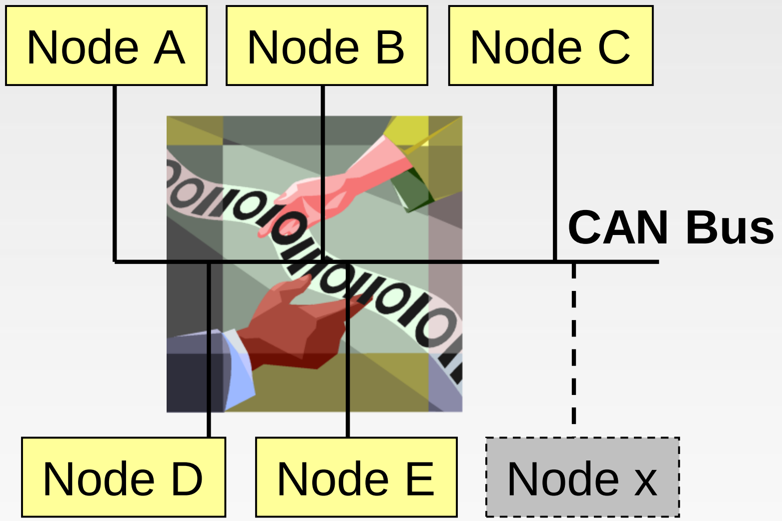
Features	Benefits
Has a multiple-master hierarchy	For building intelligent and redundant systems
Provides transfer rates up to 1 Megabit/sec	For adequate real-time response in many embedded control applications
Allows 0-8 bytes of user data per message	To accommodate diverse design requirements
Puts multiple transmit or receive message boxes at each node and assigns each an identifier	For flexibility in system design

Main Features of CAN

Features	Benefits
Eliminates addresses of transmitting and receiving nodes in data messages	To save bus bandwidth, simplify software, and allow simultaneous transmission of node-to-node and broadcast messages
Causes receiving nodes to filter messages based on their assigned identifiers (IDs)	▪ To simplify node hardware and software ▪ To permit message prioritization ▪ To allow the hardware to arbitrate the CAN bus
Automatically retransmits messages if corruption occurs	For accurate communication, even in noisy environments
Provides error detection, signaling and fault-confinement measures	To ensure highly reliable network operation

Design Factors to Consider

- **Distance/environment**
  - CAN 2.0B: 1Mbps, up to 40m
  - CAN 2.0A: 125kbps, up to 500m
  - Suitable for difficult environments:
    - industrial, automotive, and more
- **Reliability requirements**
  - Integrated error detection and confinement
  - Automatic retransmission of corrupted message
  - Probability of undetected bad message is $< 4.7 \times 10^{-11}$
- **Number of nodes**
  - Depends on Physical layer; >100 is feasible
- **Number of masters**
  - Every node can initiate communication and negotiate for the bus
- **Net data transfer rate**
  - Up to 577Kbps net at 1Mbps total data transfer rate
- **Message priority**
  - Message with lowest numerical value identifier wins if two nodes try to transmit at the same time



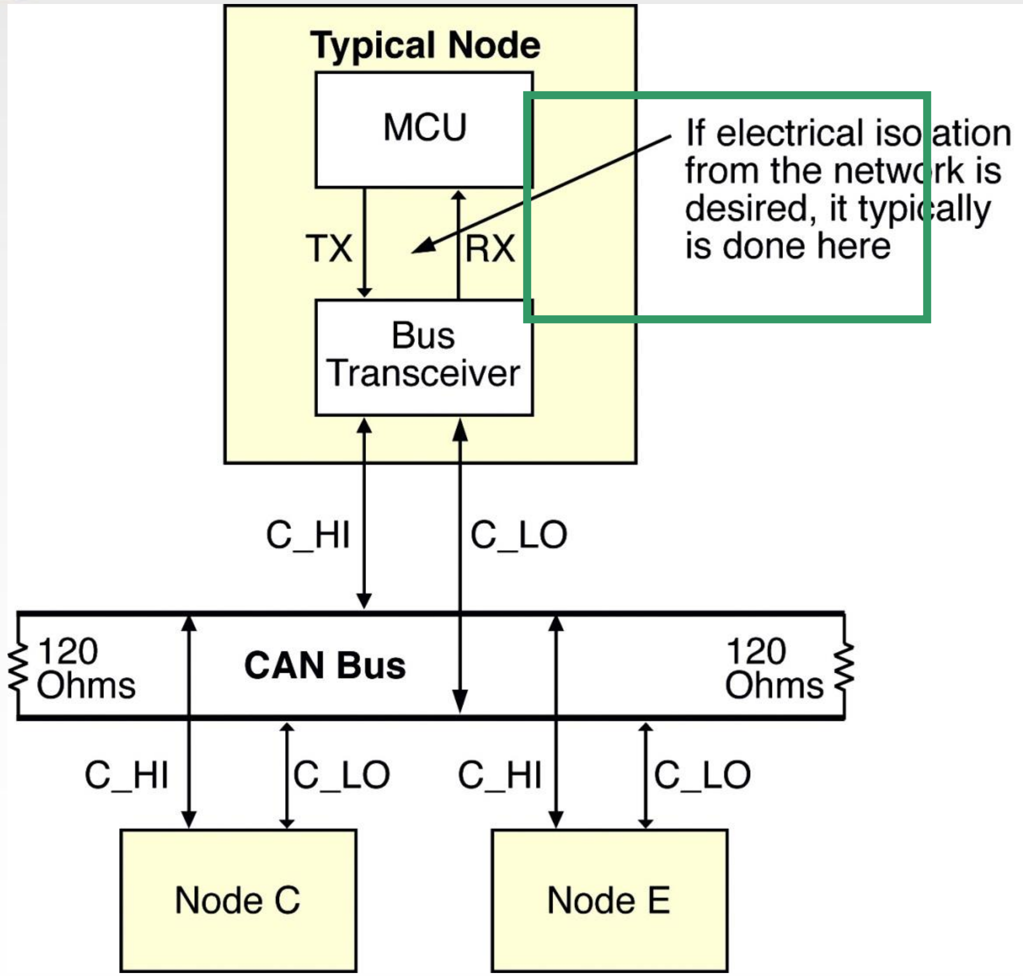
Data Flow

CAN Bus Traffic:

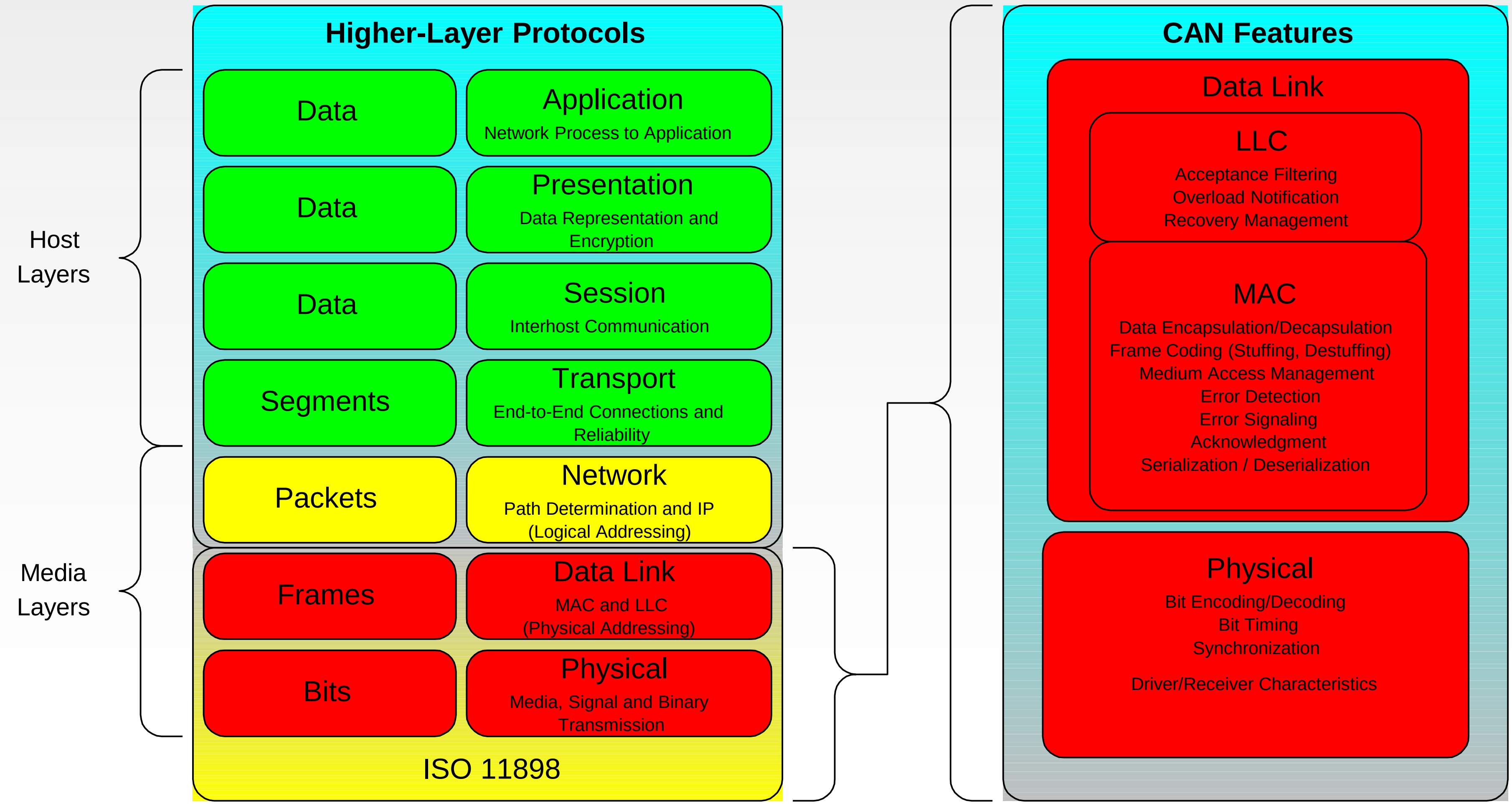
- The transmitter at a CAN node broadcasts the data frame to all nodes on the bus.
  - Nodes configured to accept the data save it
  - Other nodes do nothing with the data
- CAN 2.0A has an 11-bit message identifier and operates at a maximum frequency of 250kbps.
- CAN 2.0B has 11-bit or 29-bit message identifiers and operates at up to 1Mbps.

Physical Interface

- Dominant low (voltage) line (Logic 0)
- Recessive high line (Logic 1)
- Bus must be terminated
- Most common Physical-layer choice: ISO11898-2



CAN in the OSI Model



Higher-layer CAN Protocols