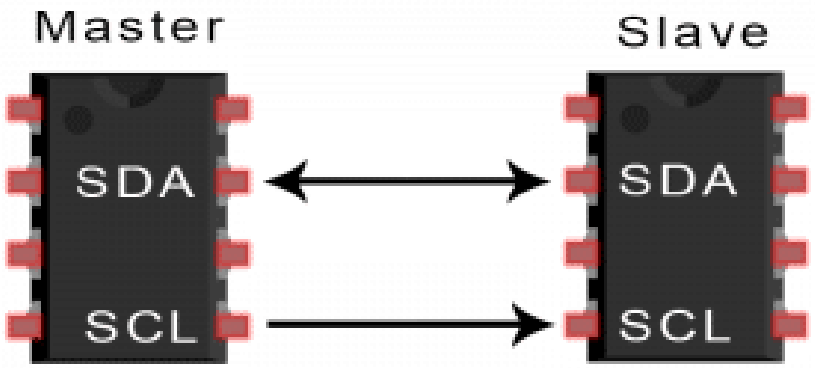
EMBEDDED NETWORKING PROTOCOLS

MODULE 6

INTER-INTEGRATED CIRCUIT

INTRODUCTION

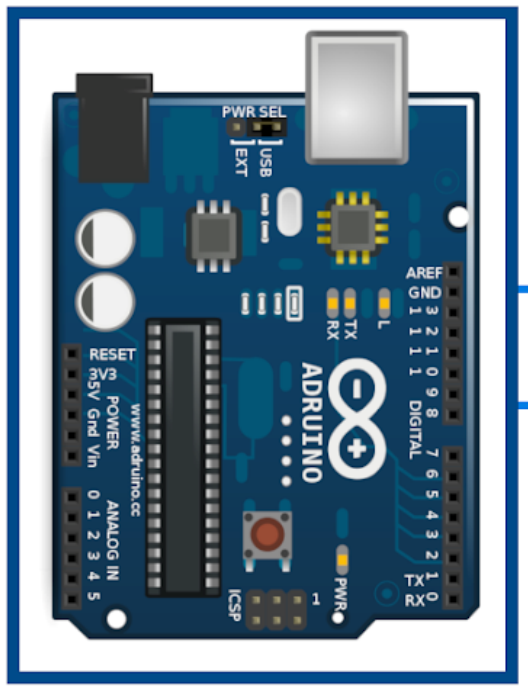
- I2C is invented in 1989 by Philips semiconductor
- It is a serial protocol with only two wire interface like UART
- Widely used for short distance and intra-board communication
- I2C has the best features of both SPI and UART protocols
- It can support *multi-master and multi-slave configuration*



https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol/

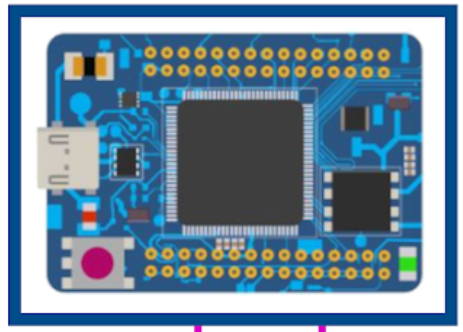
I2C

Microcontroller



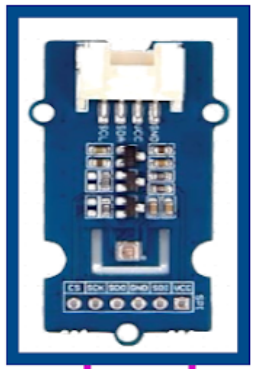
Arduino
(Master)

Microcontroller



(Master)

Gyrometer



(Slave)

VCC(5v)

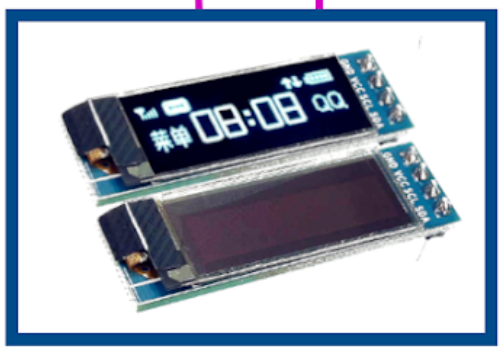
4.7k 4.7k

2 Pull up Resisters

SDA

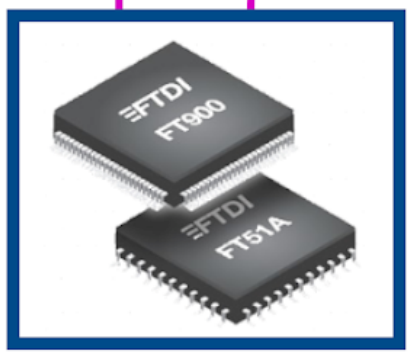
SCL

(Slave)



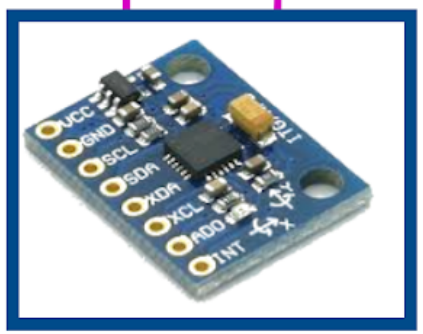
OLED Display

(Slave)



EEPROM

(Slave)



Temp. Sensor

INTER-INTEGRATED CIRCUIT

INTRODUCTION

- I2C protocol use only two pins Serial Data (SDA) and Serial Clock (SCL)
- Data is transmitted bit by bit along SDA in I2C
- SDA (Serial Data) – Bidirectional connection between master and slave
- SCL (Serial Clock) – clock signal

INTER-INTEGRATED CIRCUIT

INTRODUCTION

- I2C is also a synchronous protocol, hence the output is synchronized using clock signal
- Common clock signal is shared between master and slave
- Master always controls the clock signal
- Master device addresses the slave devices via SCL

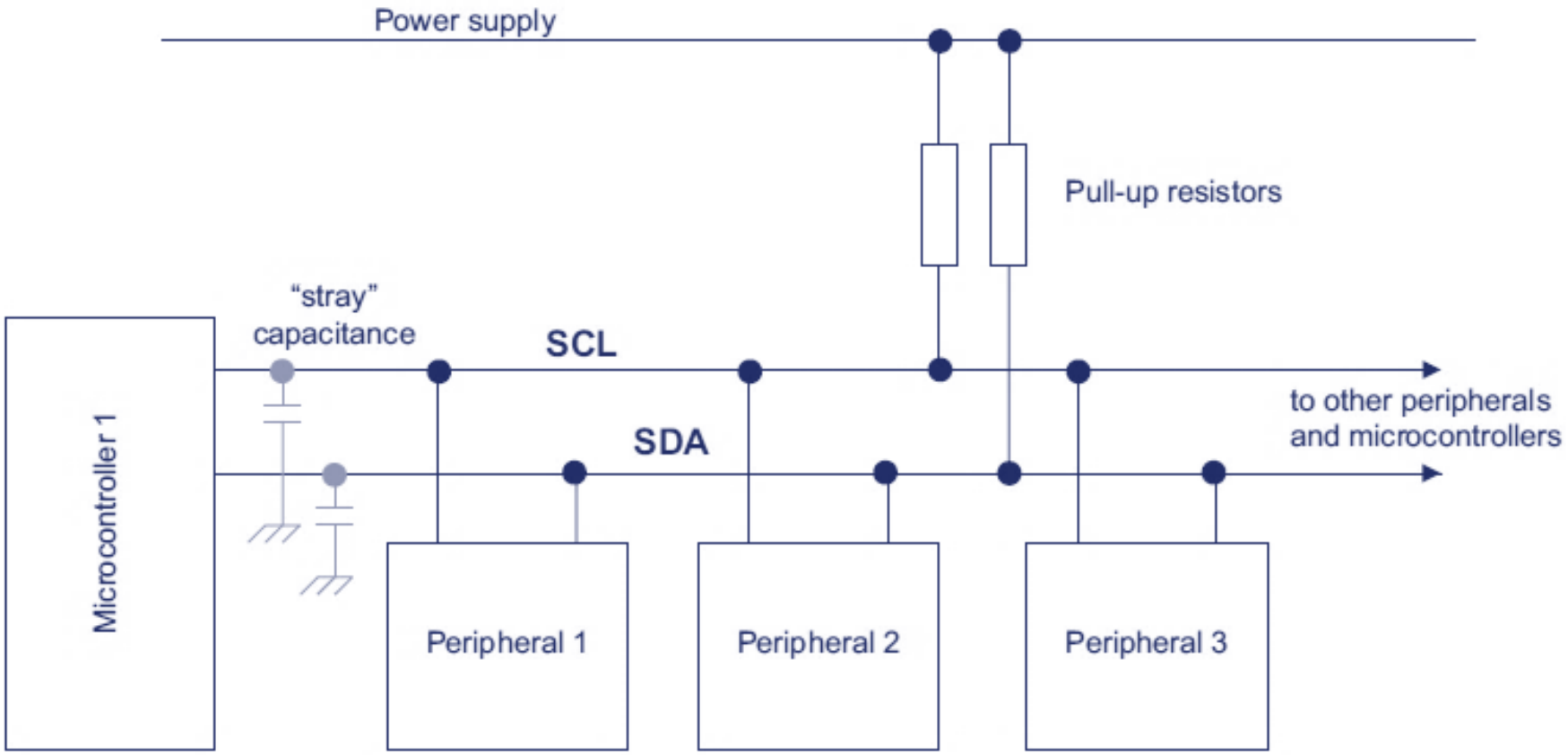
INTER-INTEGRATED CIRCUIT

HOW I2C WORKS

- Data is transferred as messages in I2C
  - Messages are broken to frames of data
  - Each message contains a binary address frame of the slave and data frames
  - One or many data frames with data can be transferred in one message
  - Message has start and stop conditions, read, write, ACK & NACK bit between each data frame

INTER-INTEGRATED CIRCUIT

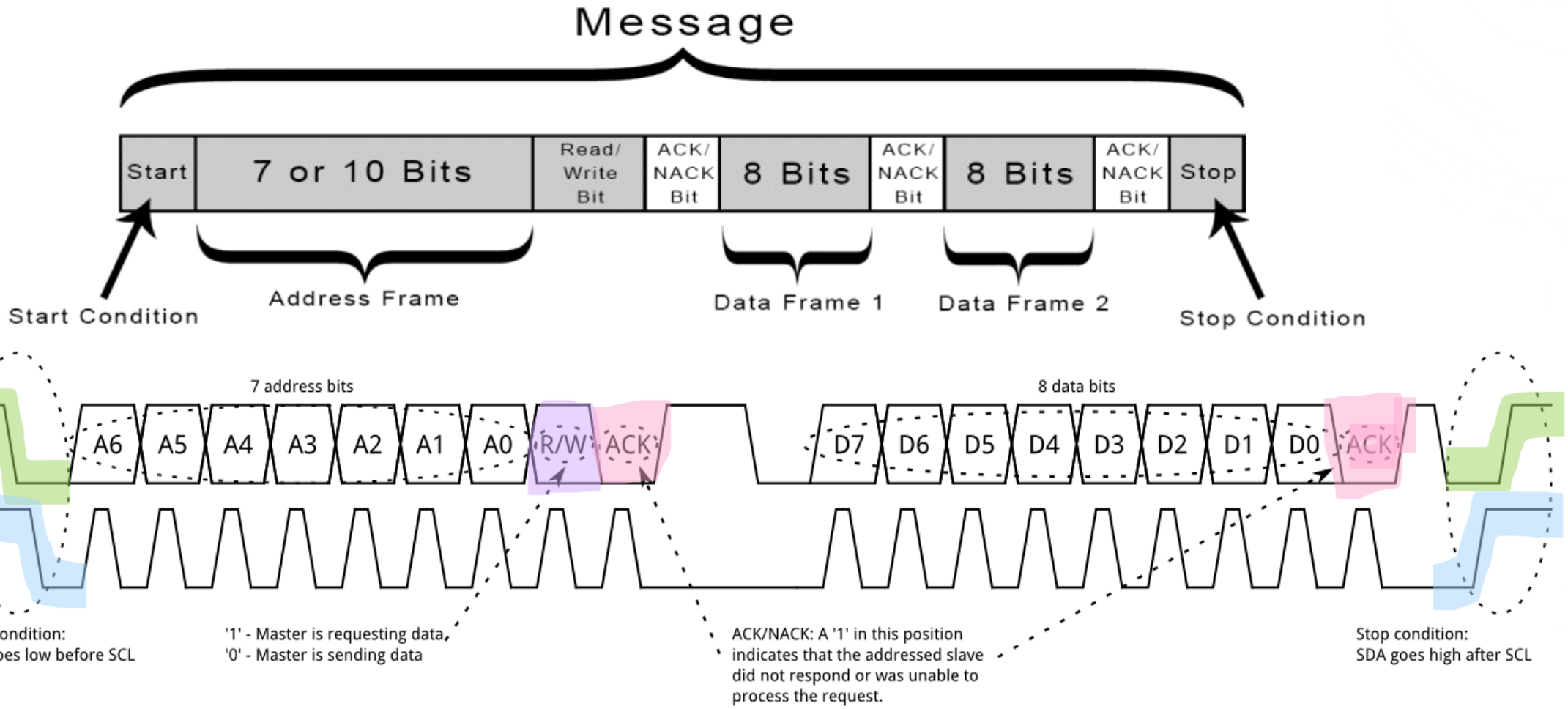
HOW I2C WORKS



An inter-integrated circuit-based system.

INTER-INTEGRATED CIRCUIT

HOW I2C WORKS



https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol/

INTER-INTEGRATED CIRCUIT

HOW I2C WORKS

- **Start Condition:** SDA pin switches from high to low voltage level before the SCL pin switches from high to low
- **Stop Condition:** SDA pin switches from low to high voltage after SCL pin switches from low to high
- **Address Frame:** 7- or 10-bit unique address is assigned to each slave for identifying when master wants to connect to it

INTER-INTEGRATED CIRCUIT

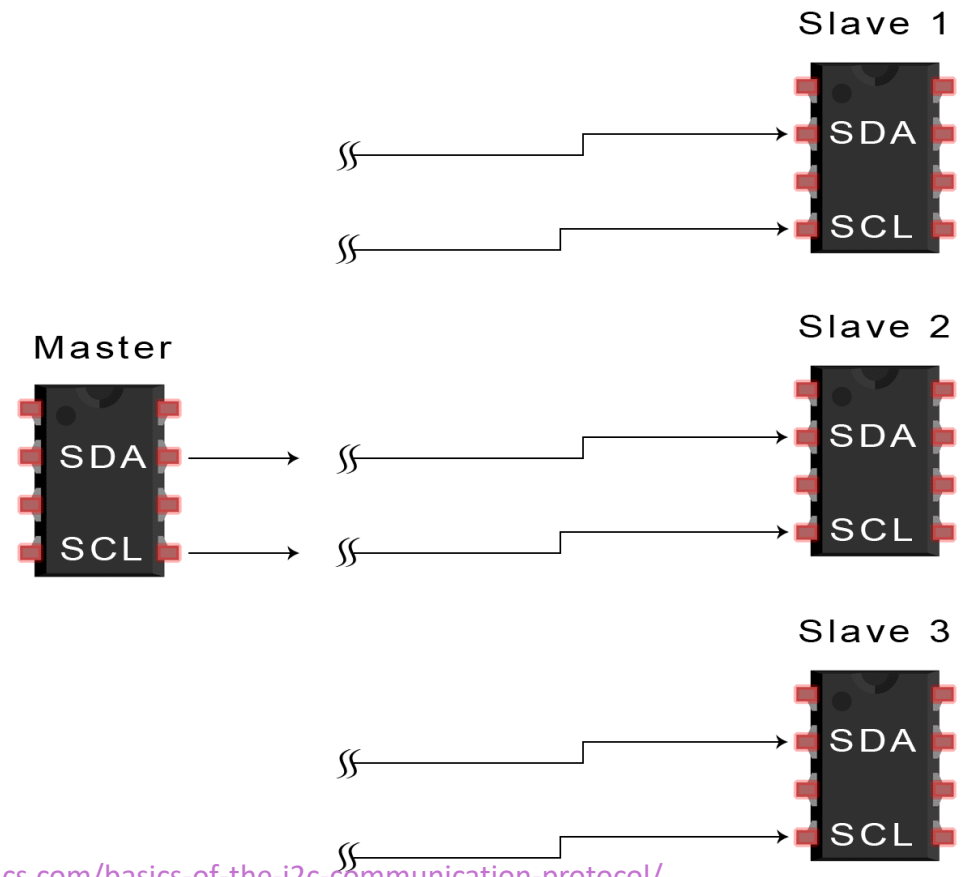
HOW I2C WORKS

- **Read/Write Bit:** This bit defines the state of master, whether it is sending data (low) or receiving data (high)
- **ACK/NACK Bit:** Each frame is followed by a ACK bit.
- If message is successfully received, an ACK bit is returned

INTER-INTEGRATED CIRCUIT

STEPS OF I2C DATA TRANSMISSION

STEP-1: The master sets the start condition by setting all SDA pin to low before setting SCL pin

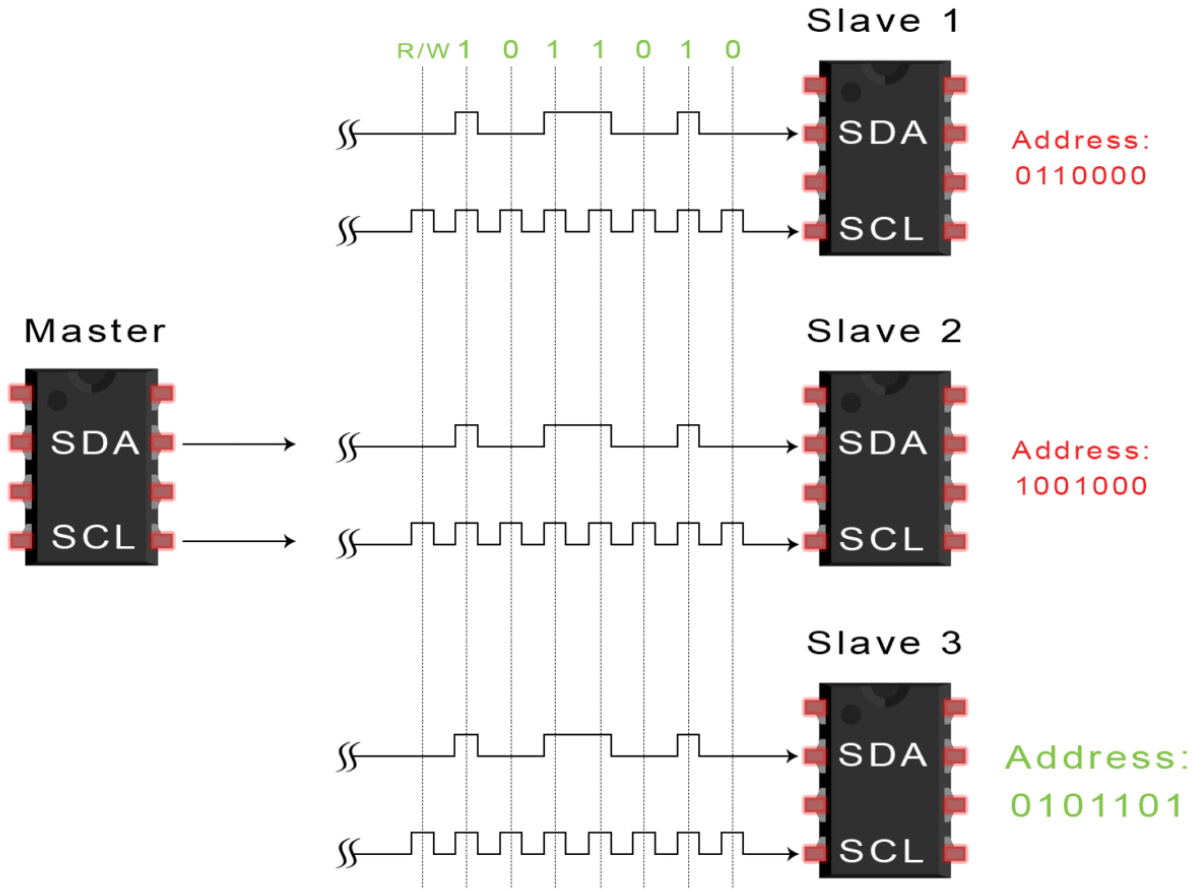


https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol/

INTER-INTEGRATED CIRCUIT

STEPS OF I2C DATA TRANSMISSION

STEP-2: The master sends 7 or 10 bit address to each slave with which it wants to communicate with, along with the read/write bit



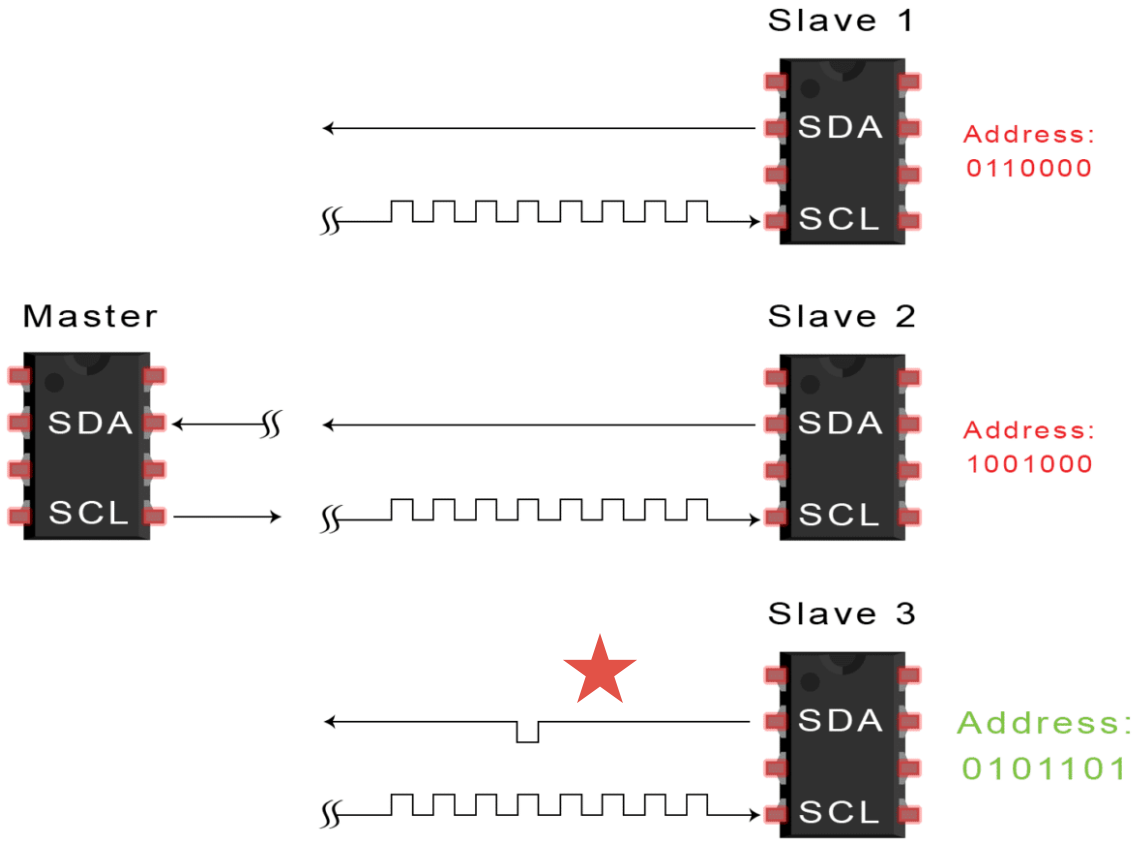
https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol/

INTER-INTEGRATED CIRCUIT

STEPS OF I2C DATA TRANSMISSION

STEP-3:

- After receiving address from master, slave compare the address with its own address
- If the address matches, *ACK bit is sent and SDA pin is switched to low*
- If address does not matches *slave keeps the SDA pin high*

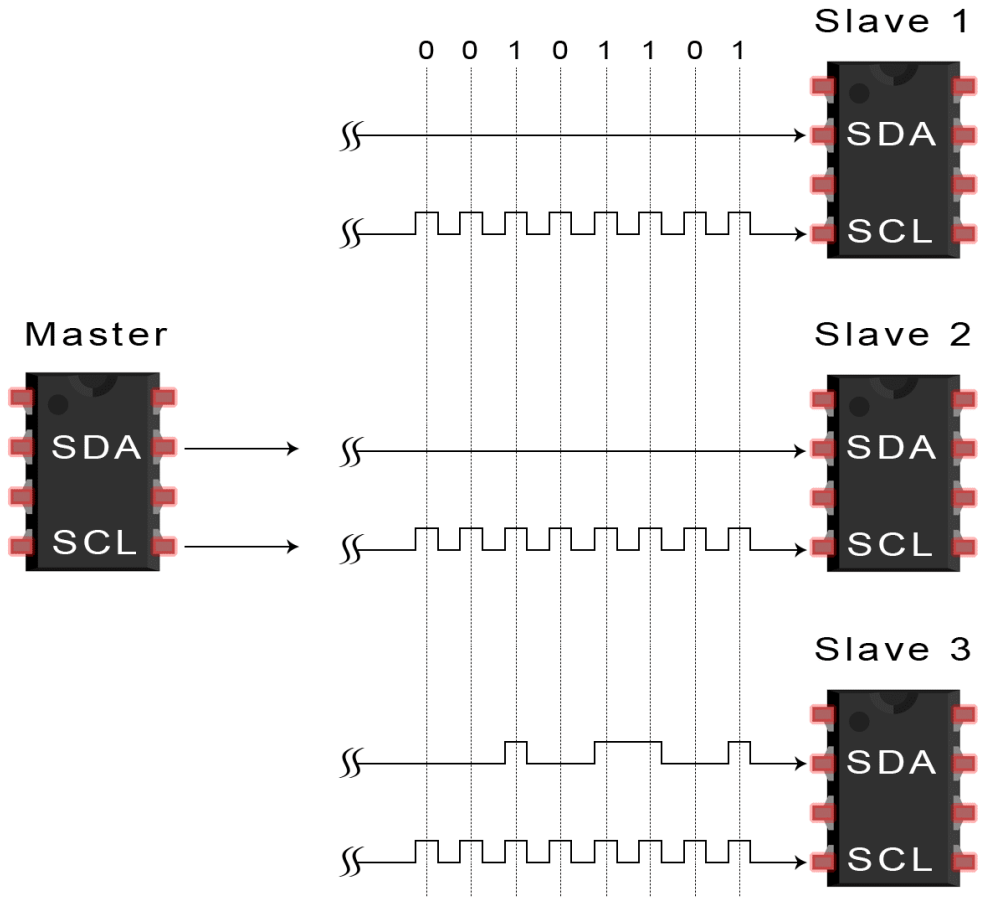


https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol/

INTER-INTEGRATED CIRCUIT

STEPS OF I2C DATA TRANSMISSION

STEP-4: The master sends or receives the data frame

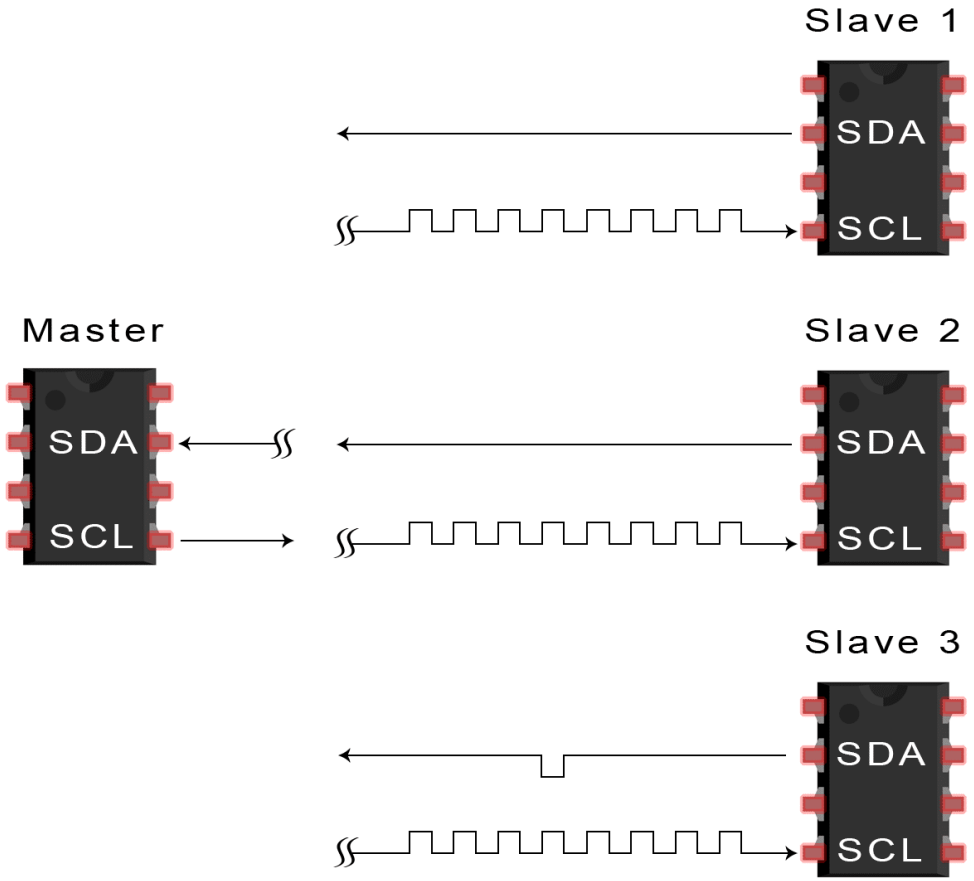


https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol/

INTER-INTEGRATED CIRCUIT

STEPS OF I2C DATA TRANSMISSION

STEP-5: After each successful data reception, the receiving device sent an ACK bit

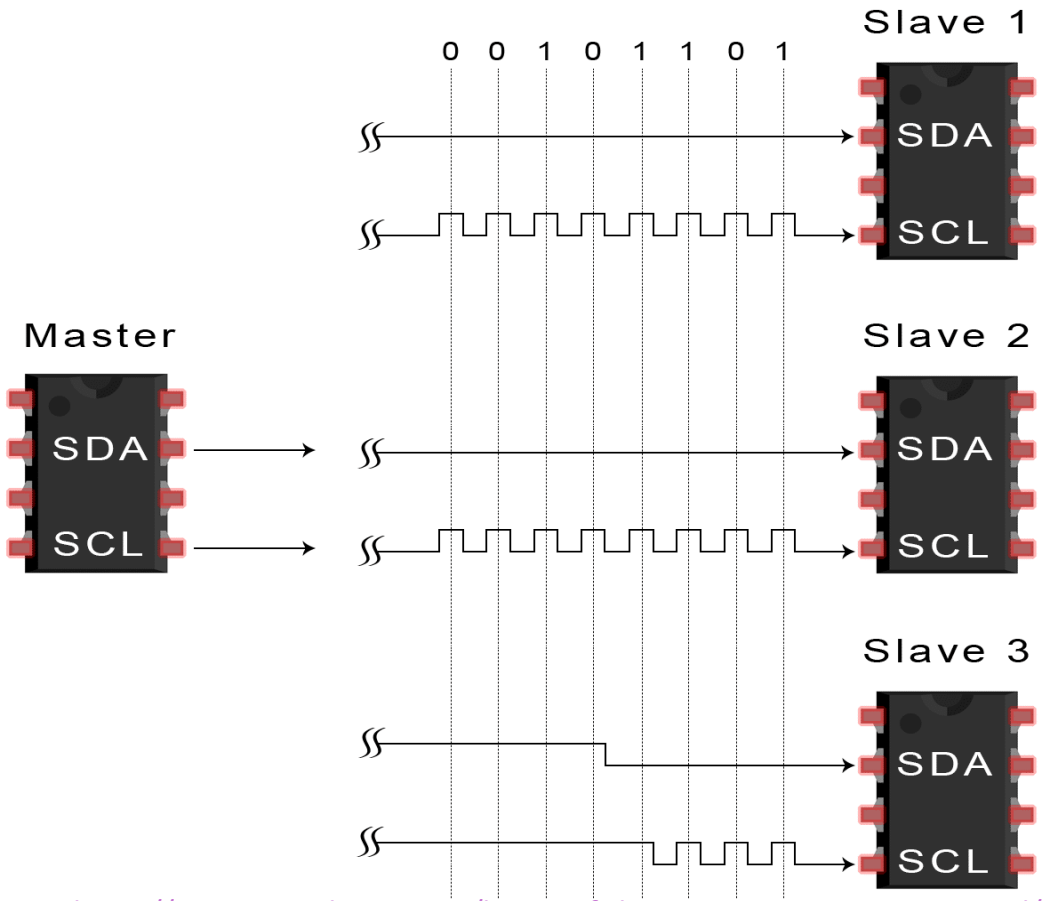


https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol/

INTER-INTEGRATED CIRCUIT

STEPS OF I2C DATA TRANSMISSION

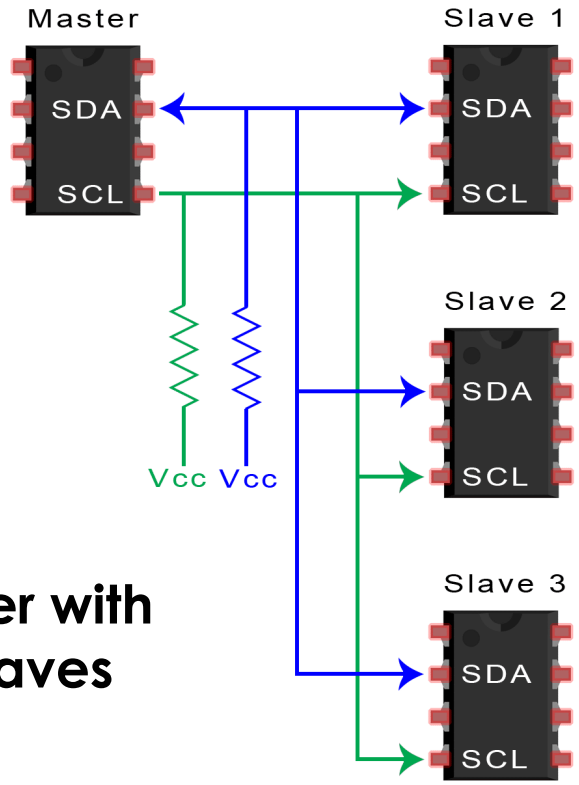
STEP-6: To end the data transmission, the master sets the SCL pin high before setting SDA pin high



https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol/

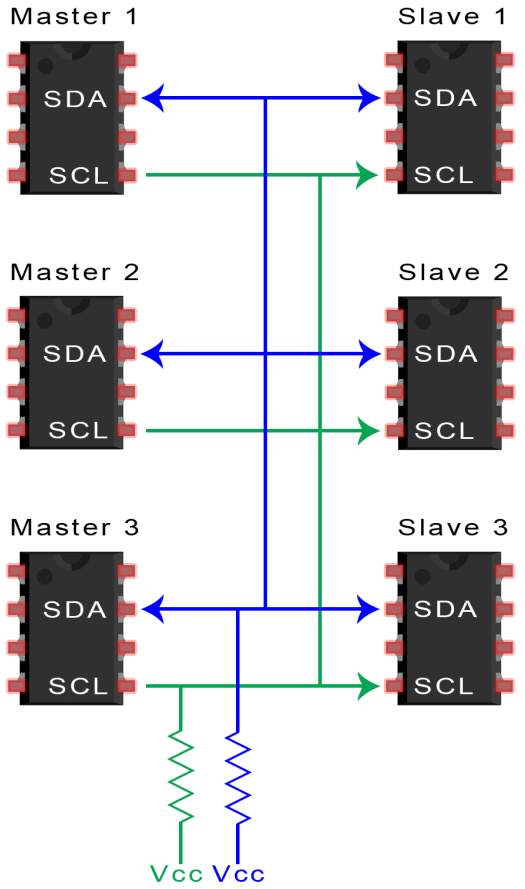
INTER-INTEGRATED CIRCUIT

STEPS OF I2C DATA TRANSMISSION



Single Master with Multiple Slaves

Multiple Master with Multiple Slaves



https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol/

INTER-INTEGRATED CIRCUIT

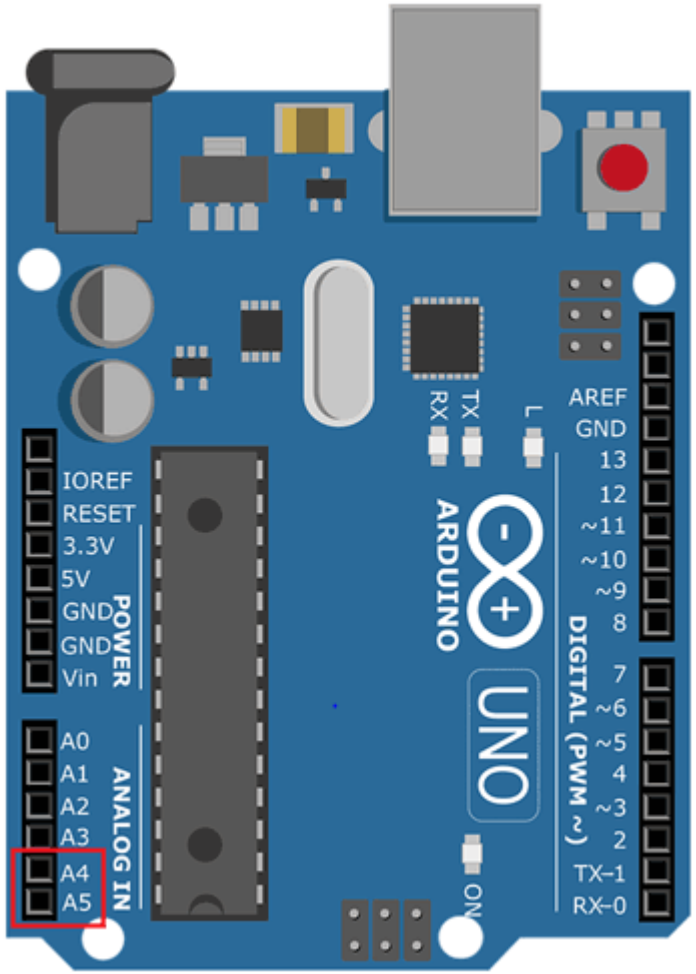
❑ ADVANTAGES:

- Only uses two wires
- Supports multiple masters and slaves
- Use chip addressing
- ACK/NACK bit for successful transmission
- it can work well with both slow and fast ICs.

❑ DISADVANTAGES:

- Slower data transfer rate than SPI
- Clock speed limitation & power dissipation because of pull-up resistor
- hardware complexity increases with multi master/slave configuration
- Frame overhead due to ack bits and device address bits

INTER-INTEGRATED CIRCUIT



I2C Line	Pin in Arduino
SDA	A4
SCL	A5

INTER-INTEGRATED CIRCUIT

Arduino function for I2C protocol (Wire Library)

```
Wire.begin()                    //Initialize the I2C bus
Wire.requestFrom(address, quantity)
                    //address- 7 bit address of the device to request bytes
                    //quantity- The number of bytes to request

Wire.receive()      //receives a byte of data transmitted from slave to master
Wire.send()                     //sends data to master from slave on request
Wire.onRequest(handler)         //register a function to call when master request data
                                //from slave
Wire.onReceive(handler) -       //Registers a function to be called when a slave device
                                //receives a transmission from a master
Wire.read()                     //reads a byte transmitted form slave to master
Wire.write()                    //writes data to slave on request from master
Wire.setClock()                 //modifies the clock frequency of I2C
```

Refer. URL :https://www.arduino.cc/en/reference/SPI

INTER-INTEGRATED CIRCUIT

I2C – Arduino Interface

Master read

```
include <Wire.h>
void setup()
{
Wire.begin();                    // join i2c bus (address optional for master)
Serial.begin(9600);              // start serial for output
}
void loop()
 {
Wire.requestFrom(8, 6);          // request 6 bytes from slavedevice #8
while (Wire.available())
 {
                                 // slave may send less than requested
char c = Wire.read();            // receive a byte as character
Serial.print(c);                 // print the character
  }
delay(500);
}
```

INTER-INTEGRATED CIRCUIT

I2C – Arduino Interface

Slave sender

```
#include <Wire.h>
void setup()
{
Wire.begin(8);                          // join i2c bus with address #8
 Wire.onRequest(requestEvent);          // register event
}
void loop()
 {
delay(100);
}
// function that executes whenever data is requested by master
// this function is registered as an event, see setup()
void requestEvent()
 {  Wire.write("hello ");               // respond with message of 6 bytes
                                        // as expected by master
}
```

CAN
Control Area Network

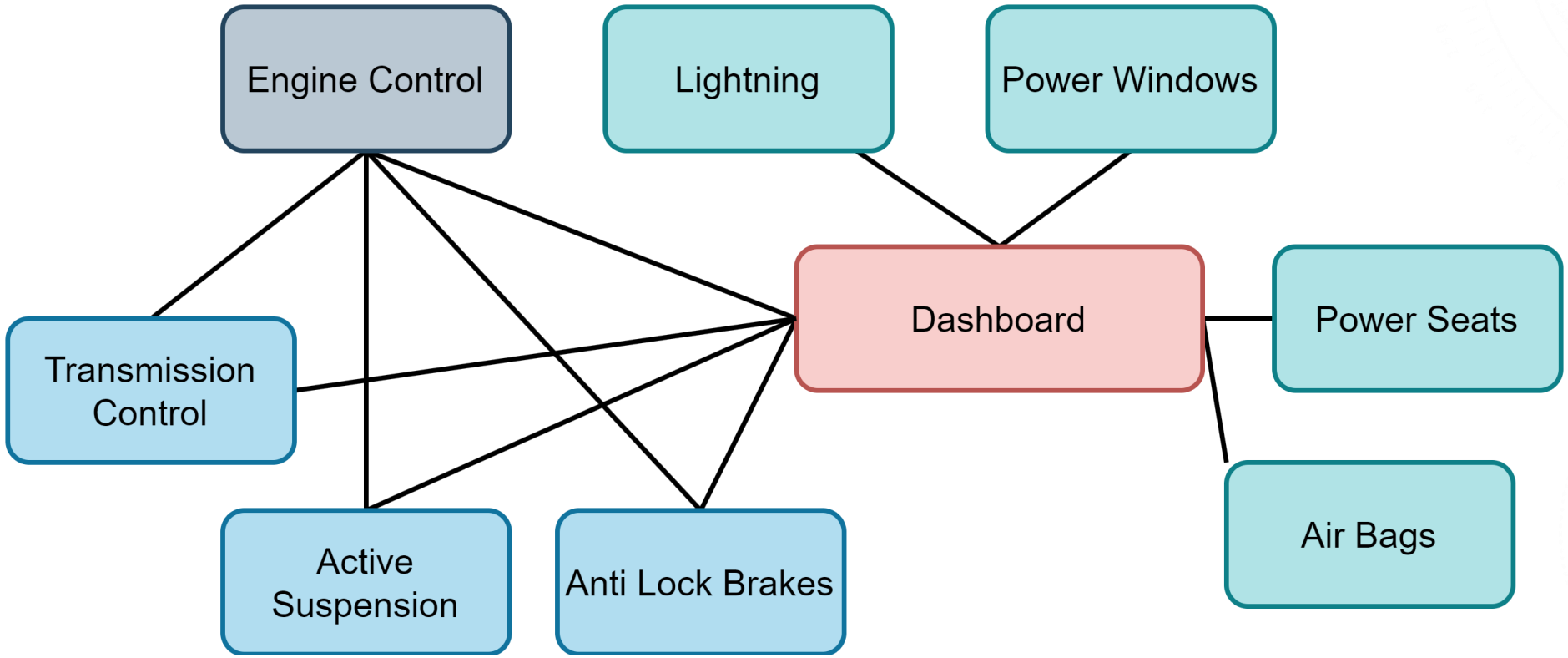
CONTROL AREA NETWORK

INTRODUCTION

- CAN bus allows all ECUs to communicate with each other via a single bus
- CAN bus has two wires- CAN low and CAN high

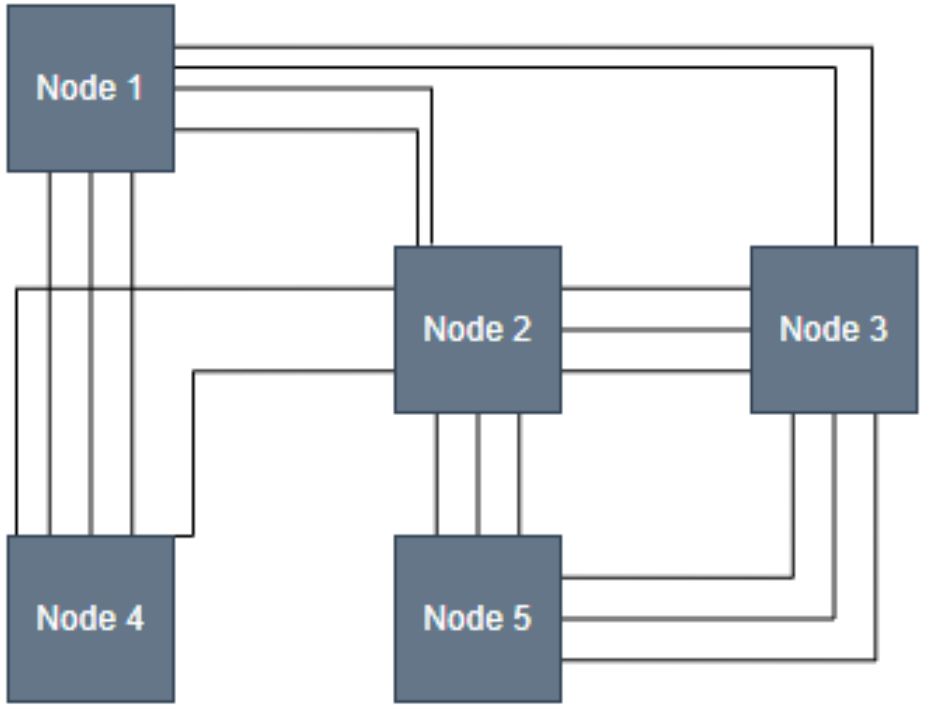
- **NEED FOR CAN:**
  - Simple and low cost – Communicate via one single bus
  - Fully centralized – one point of entry
  - Extremely robust – towards electromagnetic interference and electric disturbance
  - Efficient – CAN frame has ID for prioritization

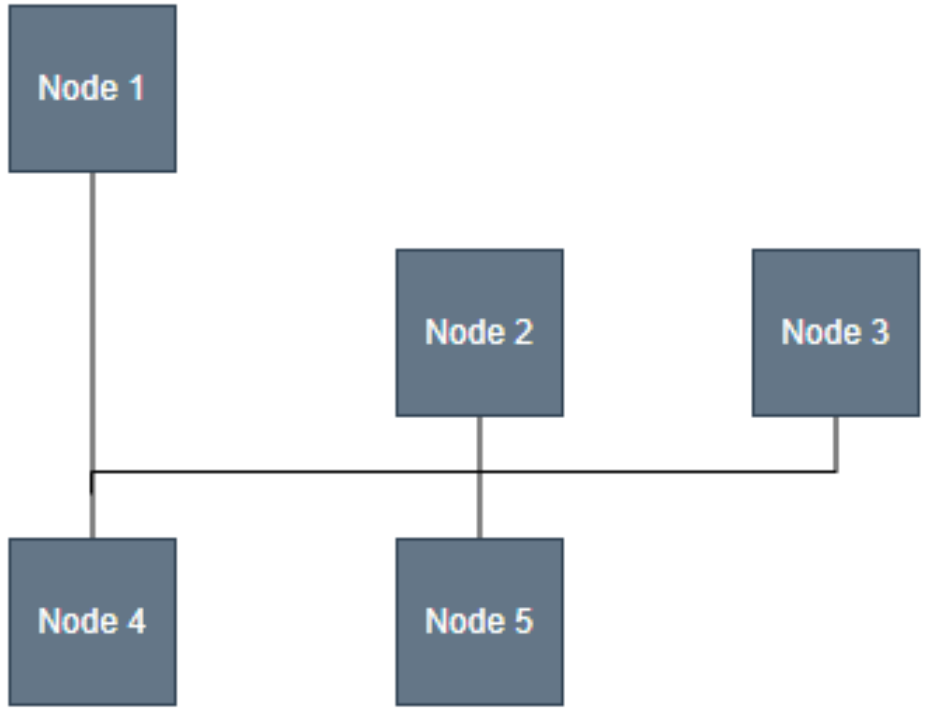
CONTROL AREA NETWORK



Pictorial view of the point to point wiring connection

CONTROL AREA NETWORK





CONTROL AREA NETWORK

INTRODUCTION

- CAN protocol is a set of rules for transmitting and receiving messages in a network of electronic devices.
- It is a method of communication used in automobiles to communicate between electronic devices like *active suspension, ABS, gear control, lighting control, air conditioning, airbags, central locking* etc.
- Every electronic device is connected using a common serial bus to receive and transmit data using CAN protocol

CONTROL AREA NETWORK

INTRODUCTION

- The node must have hardware and software embedded in them to have data exchange
- Host controller (ECU/MCU) is responsible for the functioning of the respective node.
- CAN controller convert the messages in accordance with the CAN protocols for transmitting via CAN transceiver
  - CAN controller can either be connected separately or embedded inside the host controller

CONTROL AREA NETWORK

ADVANTAGES

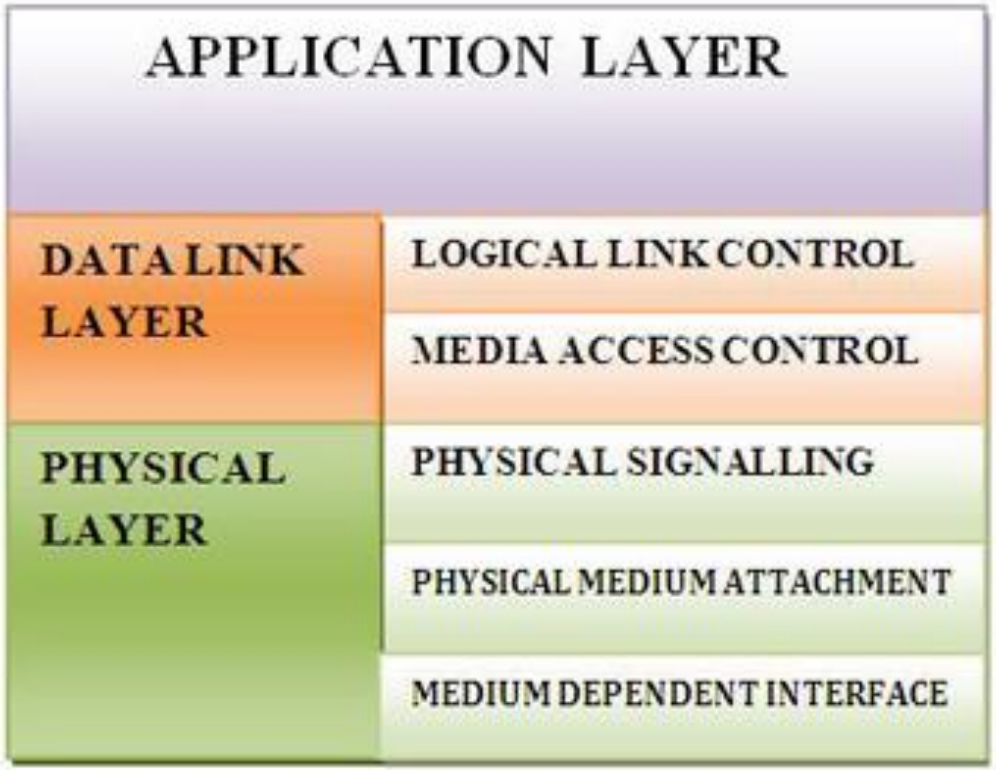
- **Low cost:** They are cheap on bulk production
- **Reliable:** Highly immune to EMI and excellent error detection and error handling
- **Flexibility:** can be easily connected / disconnected
- **Good Speed:** 1 MBit/s @ 40m bus length.
- **Multi-master communication**
- **Fault Confinement**
- **Broadcast capability**

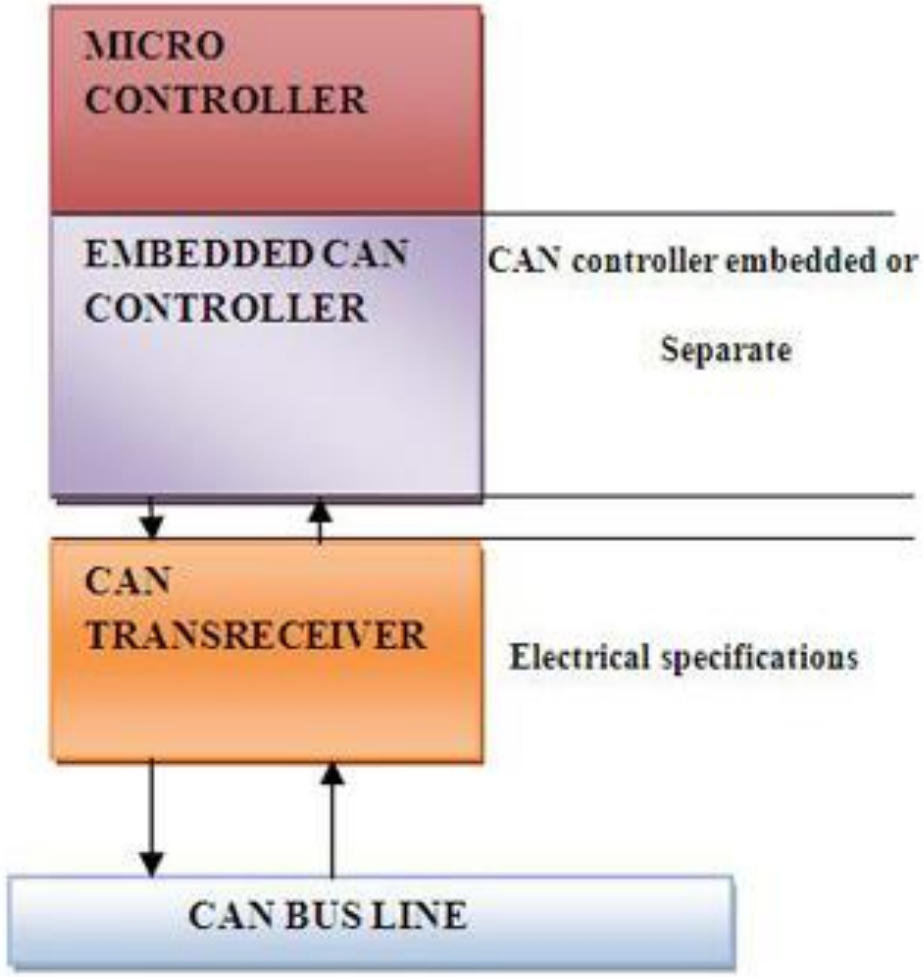
CONTROL AREA NETWORK

CAN ARCHITECTURE

- CAN uses the existing OSI reference model for data transmission
- CAN protocol uses lower two layers of OSI physical layer and data link layer.
- System designer defined the functionality of remaining five layers according to needs
- Host controller uses application layer of OSI model
- CAN controller incorporate logical link control (LLC) and MAC medium access control of data link layer.
- LLC allows filtering of messages by using unique ID

CONTROL AREA NETWORK





CONTROL AREA NETWORK

CAN ARCHITECTURE

- Once, framing is done it is followed by arbitration, error detection and acknowledgement handled by MAC layer
- CAN trans-receiver takes care of encoding and decoding.
- Finally, CAN trans-receiver synchronizes with the CAN bus for message transmission

CONTROL AREA NETWORK

HOW CAN PROTOCOL WORKS?

- Master-slave configuration is not supported in the CAN protocol architecture
- Each node *cannot access every other node* for *reading* and *writing data* on CAN bus
- Based on the availability of bus, CAN bus writes the CAN frame
- Frame is defined structure, carrying sequence of bit or bytes of data
- In CAN protocol *predefined unique ID* is used instead of *destination address*

CONTROL AREA NETWORK

HOW CAN PROTOCOL WORKS?

- Every byte of information is defined in the CAN protocol
- All nodes receive the CAN frame and based on the ID nodes decides whether to accept it or not
- Highest priority message gets the bus access and low priority nodes will wait till the bus is available
- Nodes can be added without re-programming

CONTROL AREA NETWORK

MESSAGE FRAMING

- Messages in CAN are sent in a format called frames.
- Framing of message is done by MAC sub layer of Data Link Layer.
- There are two type of frames standard or extended.
- These frames can be differentiated on the basis of identifier fields.
- CAN frame with 11-bit identifier fields called Standard CAN and with 29-bit identifier field is called Extended frame
- Binary values in CAN protocol are termed as dominant (logic 0) and recessive (logic 1) bits.

CONTROL AREA NETWORK

STANDARD CAN - MESSAGE FRAMING

SOF	11 bit identifier	RTR	IDE	r0	DLC	0…8 byte data	CRC	ACK	EOF	IFS

Various fields in standard CAN

CONTROL AREA NETWORK

STANDARD CAN - MESSAGE FRAMING

- **SOF - Start of Frame bit.** It indicates start of message and used to synchronize the nodes on a bus. A dominant bit (logic 0) in the field marks the start of frame.
- **IDENTIFIER-**It serves dual purpose, one to determine which node has access to the bus and second to identify the type of message.
- **RTR - Remote Transmission Request.** It identifies whether it's a data frame or a remote frame. RTR is dominant when it is a data frame and recessive when it is a remote frame

CONTROL AREA NETWORK

STANDARD CAN - MESSAGE FRAMING

- IDE – Identifier Extension. It is used to specify the frame format. Dominant bit is for standard and recessive for extended frame.
- R0 - Reversed bit. Not used currently and kept for future use.
- DLC – Data Length Code. It is 4-bit data length code that contains the number of bytes being transmitted.
- DATA – Used to store up to 64 data bits of application data to be transmitted.

CONTROL AREA NETWORK

STANDARD CAN - MESSAGE FRAMING

- **CRC– Cyclic Redundancy Check.** The 16-bit (15 bits plus delimiter) cyclic redundancy check (CRC) contains the checksum of the preceding application data for error detection.

- **ACK – Acknowledge (ACK) field.** It compromises of the ACK slot and the ACK delimiter. When the data is received correctly the *recessive bit in ACK slot* is *overwritten* as *dominant bit* by the *receiver*.

CONTROL AREA NETWORK

STANDARD CAN - MESSAGE FRAMING

- **EOF– End of Frame (EOF).** The 7-bit field marks the end of a CAN frame (message) and disables
- **IFS - Inter Frame Space** that specifies minimum number of bits separating consecutive messages
  - IFS provides the intermission between two frames and consists of three recessive bits known as intermission bits. This time allows nodes for internal processing before the start of next frame.

CONTROL AREA NETWORK

EXTENDED CAN - MESSAGE FRAMING

- It is same as **11-bit identifier with some added fields**
- **SRR- Substitute Reverse Request-** always transmitted as a recessive bit to make sure standard data frame always have the same priority
- **R1**- It is another bit not used currently and kept for future use.

CONTROL AREA NETWORK

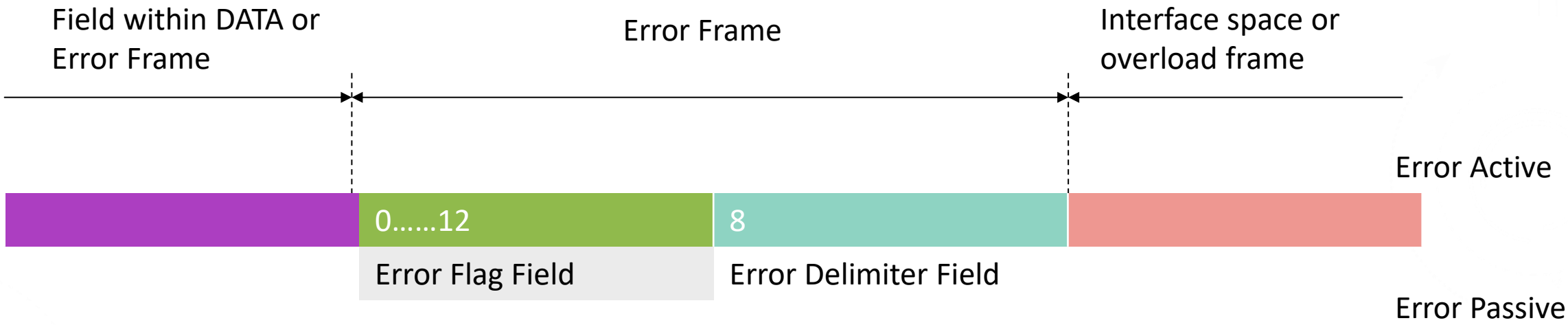
MESSAGE FRAME - TYPES

- There are four different frames which can be used on the bus.
- **Data frames-** most commonly used frame for data transmission from one node to another node
  - Data Frames contains Arbitration Fields, Control Fields, Data Fields, CRC Fields, a 2-bit Acknowledge Field and an End of Frame.
- **Remote frames –** seeks permission for data transmission from another node
  - Similar to data frame without data field & RTR bit is recessive.

CONTROL AREA NETWORK

MESSAGE FRAME - TYPES

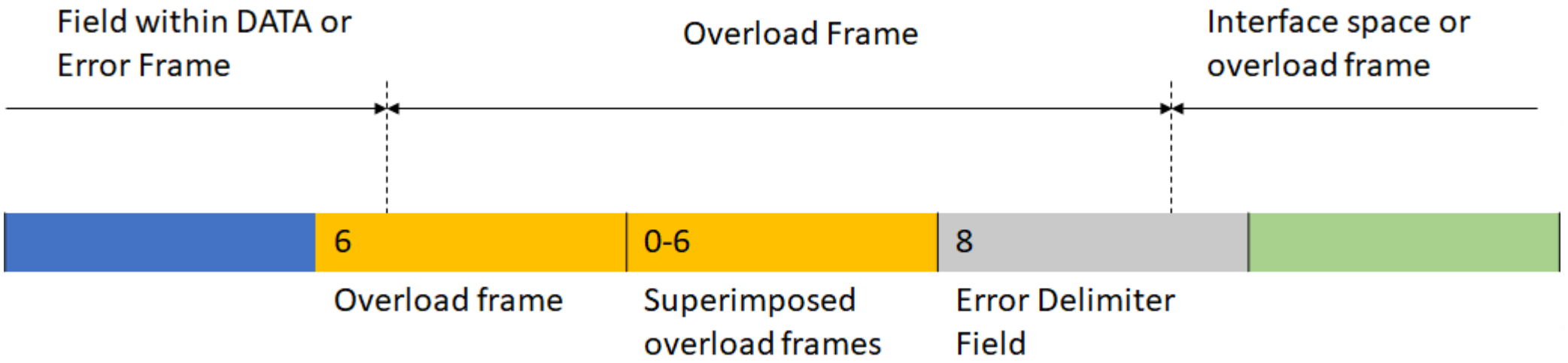
- **Error frames – If transmitting or receiving node detects an error, it will immediately abort transmission and send error frame consisting of an error flag made up of dominant bits and error flag delimiter made up of eight recessive bits.**



CONTROL AREA NETWORK

MESSAGE FRAME - TYPES

- **Overload frame-It is similar to error frame but used for providing extra delay between the messages. An Overload frame is generated by a node when it becomes too busy and is not ready to receive.**



CONTROL AREA NETWORK

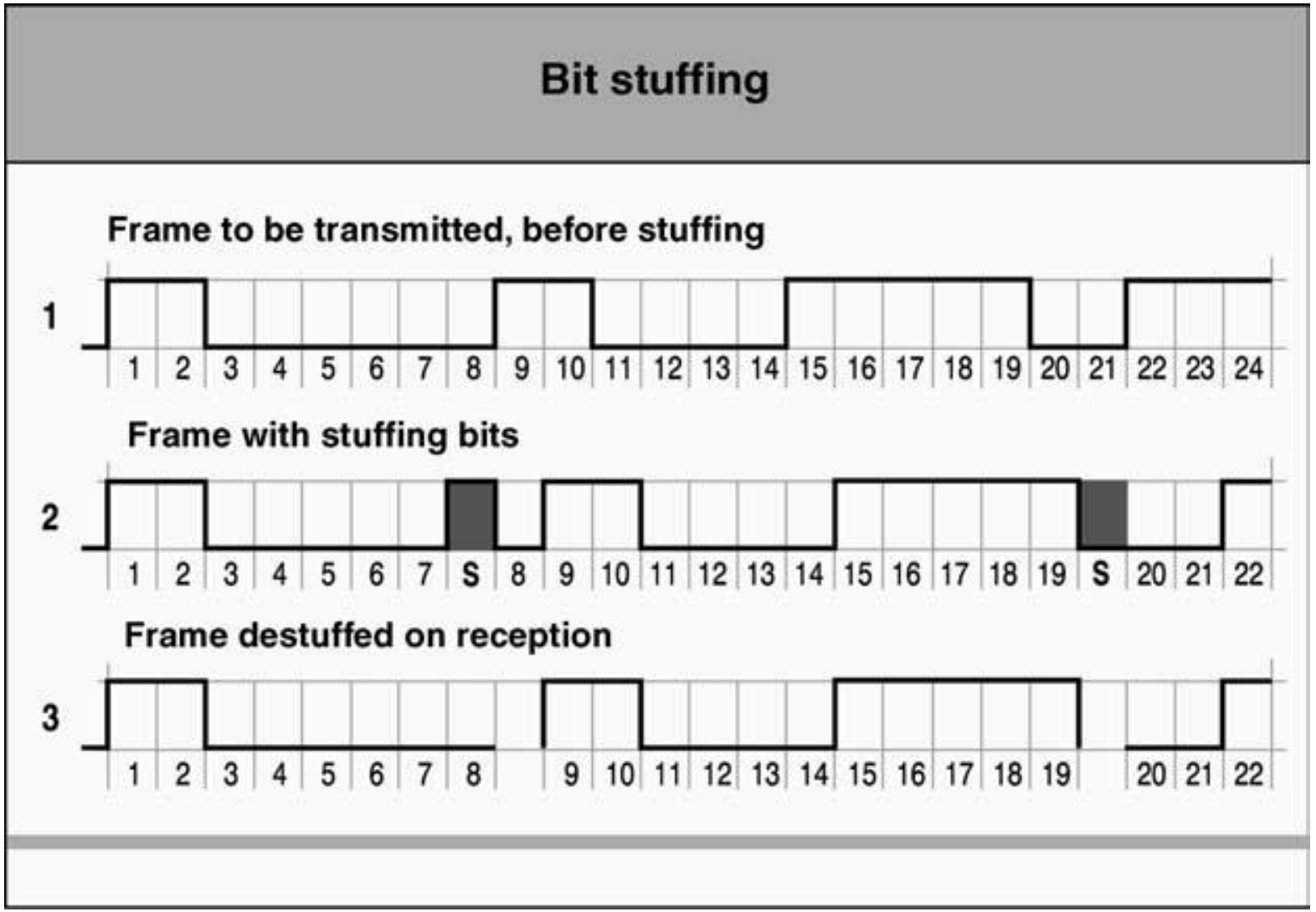
- Future of CAN bus
  - Need to accommodate increasing advanced vehicle
  - Role of Internet of Things in vehicles
  - Development of autonomous vehicles
  - The advancement of cloud computing
  - To address security issues
  - Vehicle telematics

CONTROL AREA NETWORK

- **BIT STUFFING METHOD**
  - After 5 bits of identical value (either dominant or recessive), a supplementary bit having an intentionally opposite value.
  - To 'break the rhythm', and thus indicate that everything is all right, despite appearances.
  - This technique slightly protracts the transmission time of a message.
  - But it, lays an active part in safeguarding its content during transport.

CONTROL AREA NETWORK

❑ BIT STUFFING METHOD

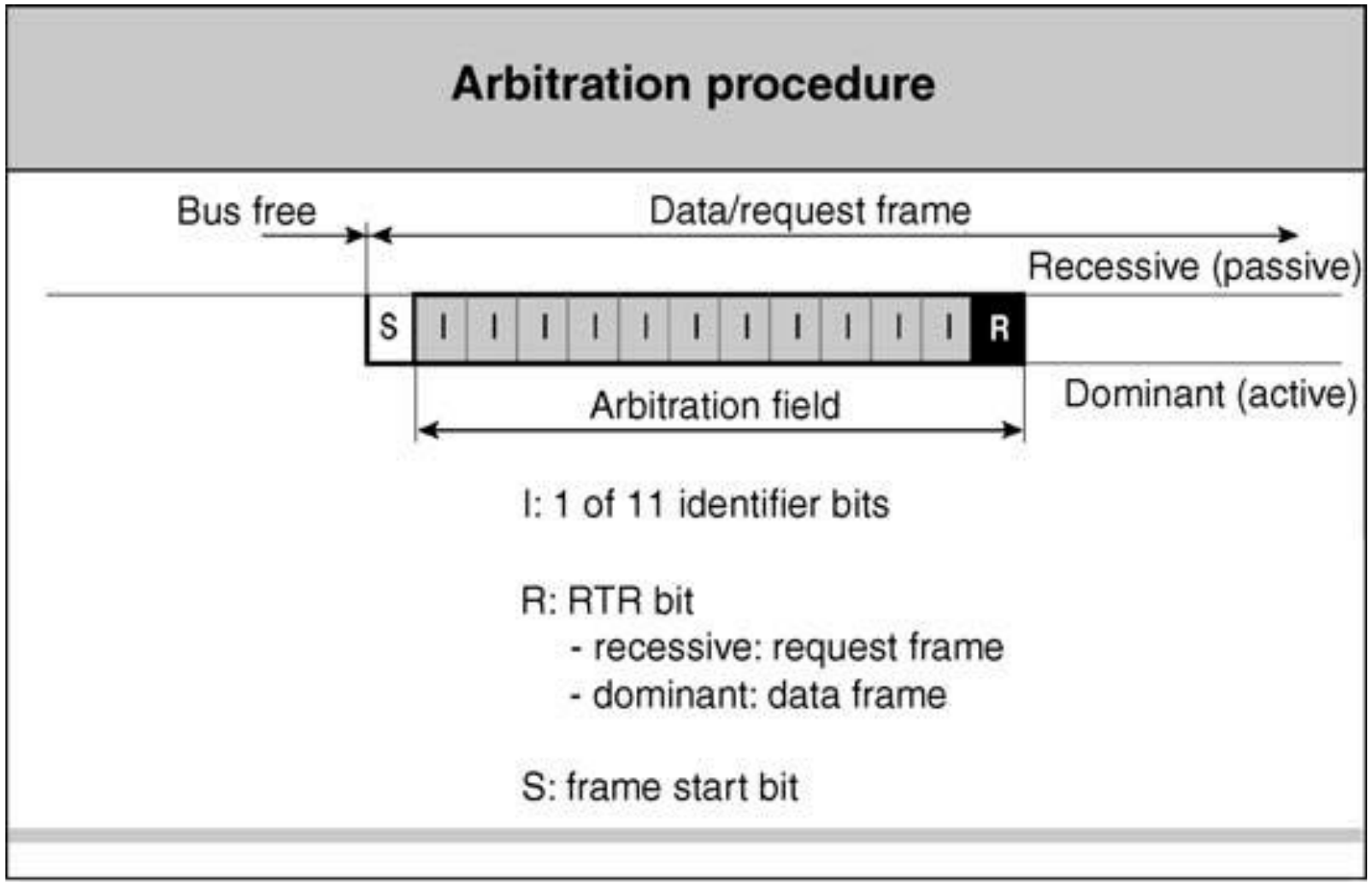


CONTROL AREA NETWORK

- **Arbitration field**
  - **Set of bits in the message frame assigned to each message to control arbitration.**
  - **Consists of the identifier bits and the bit immediately following, called the RTR (remote transmission request).**
  - **Identifier –**
    - **Length of the identifier is 11 bits,**
    - **The bits are transmitted in the order from ID_10 toID_0**
    - **the 7 most significant bits (from ID_10 to ID_4) must not all be recessive.**
  - **RTR – Must be dominant bit.**

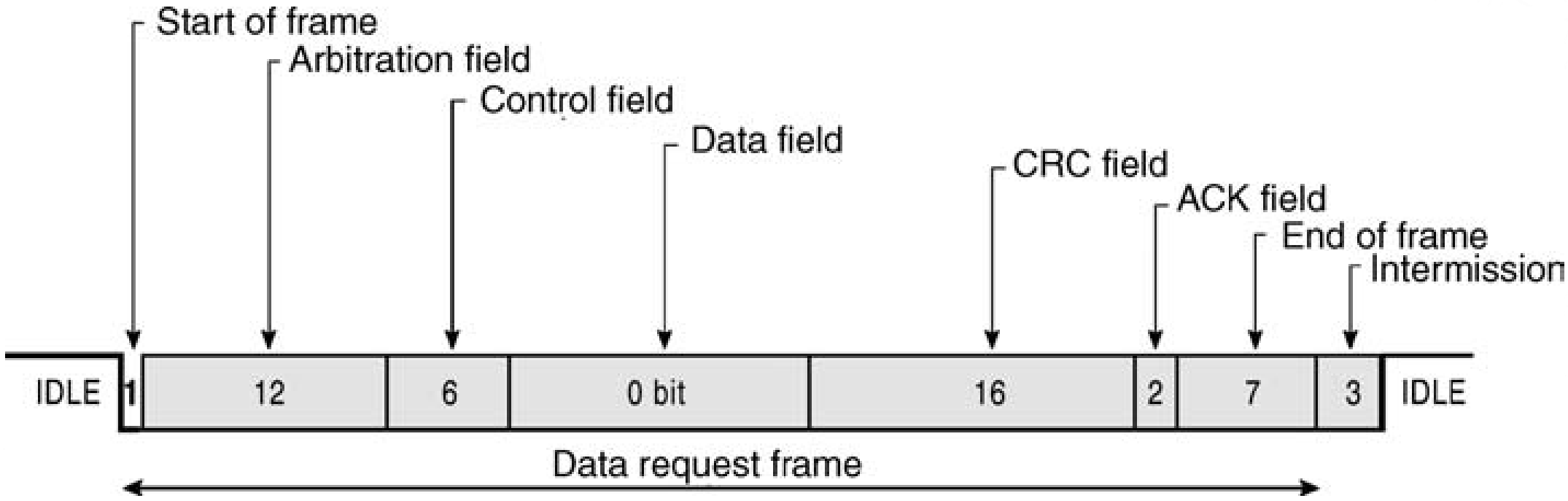
CONTROL AREA NETWORK

- **Arbitration field**



CONTROL AREA NETWORK

- **Remote Frame - RTR bit – Recessive**



CONTROL AREA NETWORK

Types of ERRORS in CAN

- In Physical Layer
  - Bit Error - bit itself is affected by errors
  - Bit Stuffing Error - involuntary (parasites, transmissions, forgotten elements) or sometimes deliberate.
- In Frame Layer
  - CRC delimiter error
  - ACKnowledgement delimiter error
  - end of frame error
  - error delimiter error
  - overload delimiter error
- The presence of errors will be signaled by an error frame which is generated on the bus to inform those entitled

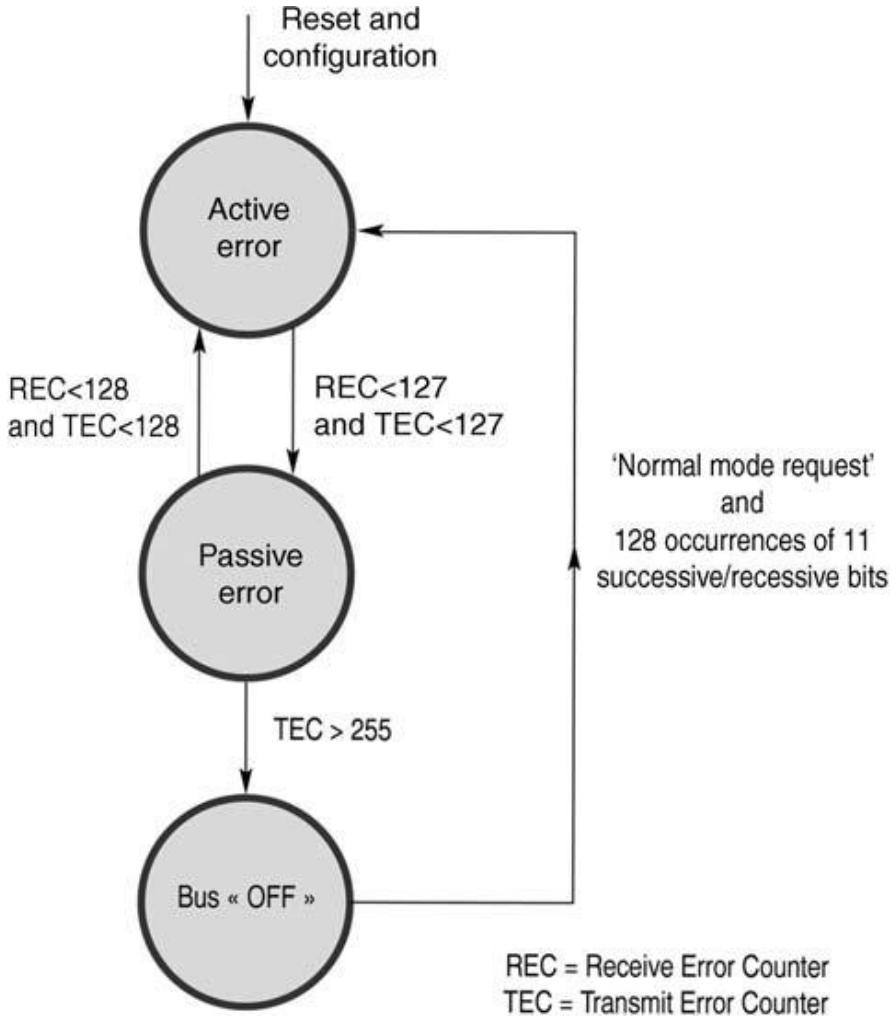
CONTROL AREA NETWORK

❑ CONFINEMENT ERRORS

- ❑ The term ‘confinement’ implies a vast mechanism having the purpose of determining whether a node is:
  - ❑ not disturbed at all,
  - ❑ slightly disturbed
  - ❑ rather more seriously disturbed by errors
  - ❑ too disturbed and must switch to bus off

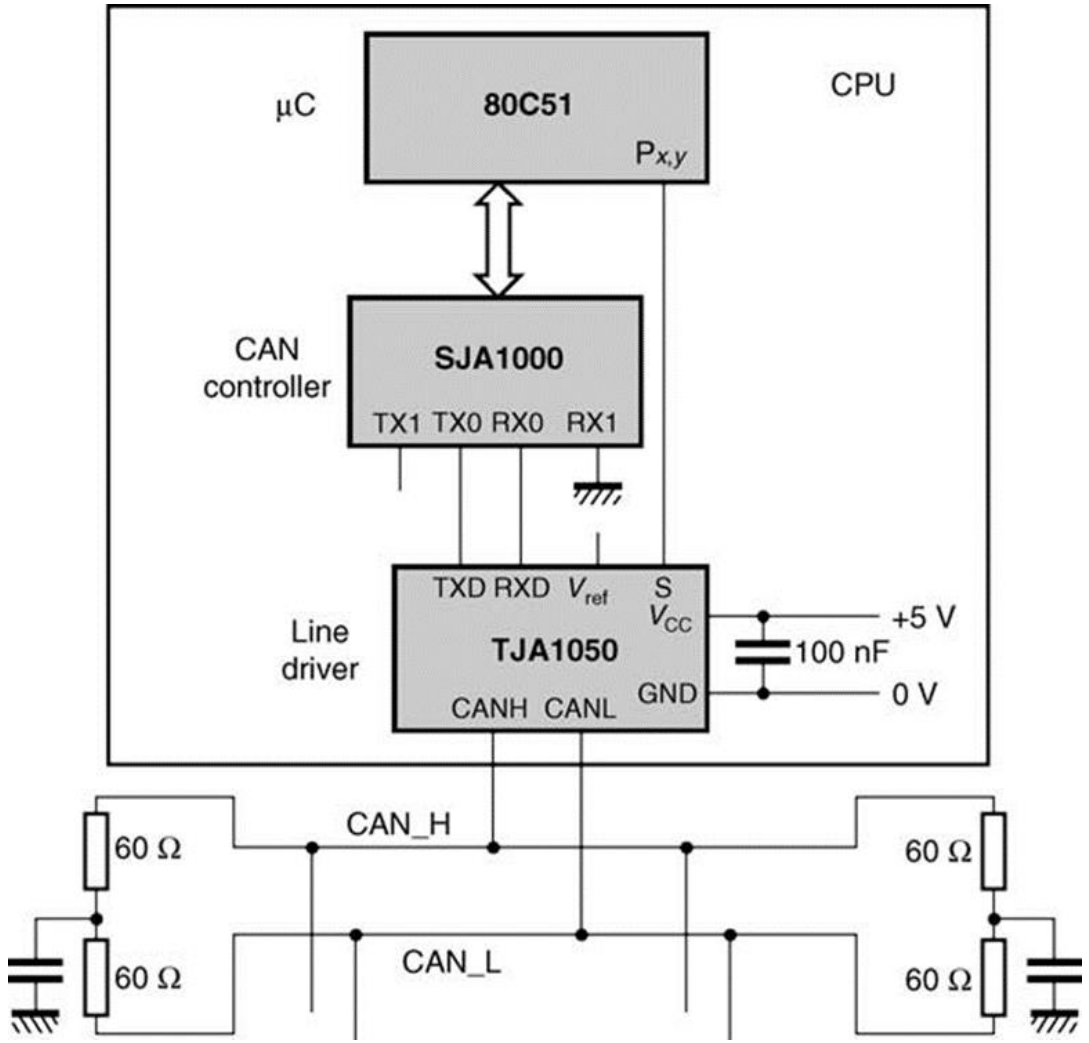
CONTROL AREA NETWORK

❑ Processing confinement errors



CONTROL AREA NETWORK

❑ Bus Access



ZIGBEE

ZIGBEE

INTRODUCTION

- ZigBee protocol is mostly used in the power constraint short distance communication

- Why another short-range communication protocol ?
  - Bluetooth
  - Wi-Fi

- ZigBee is specially built for control and sensor applications
- ZigBee is a standard with *very low cost*, *low power device*, *low data rate* and *short range communications*

ZIGBEE

INTRODUCTION

- Most commonly used standard in WSN, IoT
- Open-source standard developed by ZigBee Alliance
- It works on IEEE 802.15.4 standard for wireless personal area networks
- ZigBee operates at 868 MHz (Europe), 902-928MHz (US & Australia) and 2.4 GHz (worldwide) frequencies
- Data rate of 250 kbps for low data rate applications

ZIGBEE

- ZigBee devices can cover a distance range between 10-100 meters.
- It supports master-slave configuration or master-master configuration
- ZigBee architecture consist of coordinator, router and end device
- ZigBee supports star, tree and mesh topologies
- Modes of operation – Beacon and Beacon less

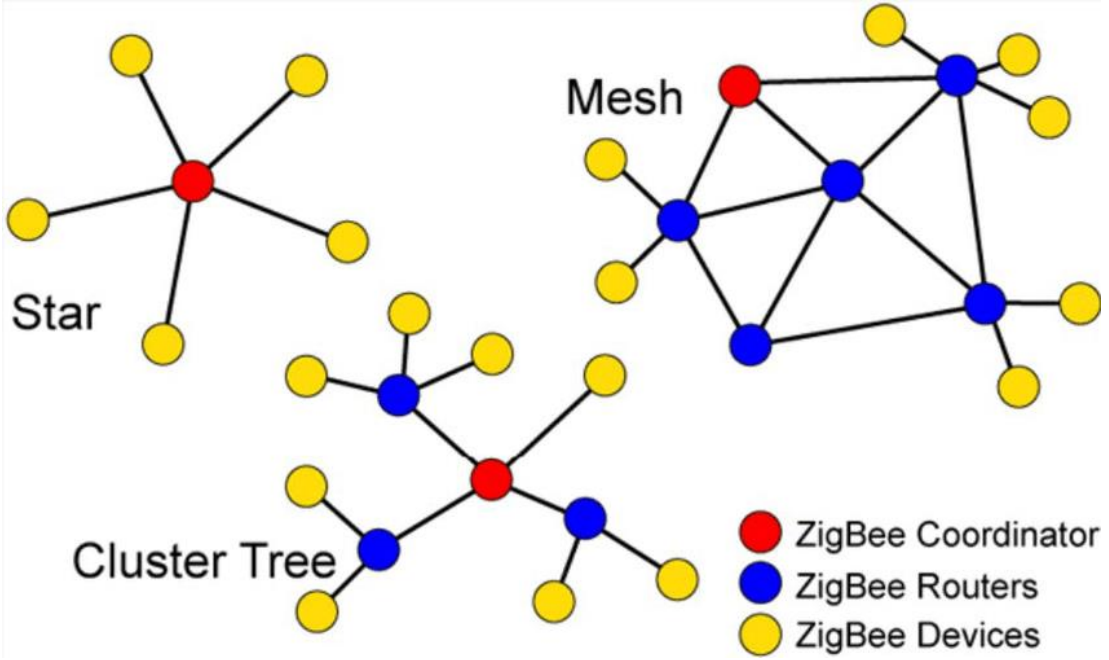
ZIGBEE

- Coordinator
  - Root of network
  - One coordinator in each network
  - Performs channel selection, assigning node ID, allocation of unique address
- Routers
  - Intermediate node between coordinator and end node
  - Route traffic between different nodes
  - Allows other end devices to join network

ZIGBEE

- **End Node**
  - All traffic to end device is first routed to parent node
  - sleep mode to increase battery efficiency

- **Architecture**



ZIGBEE

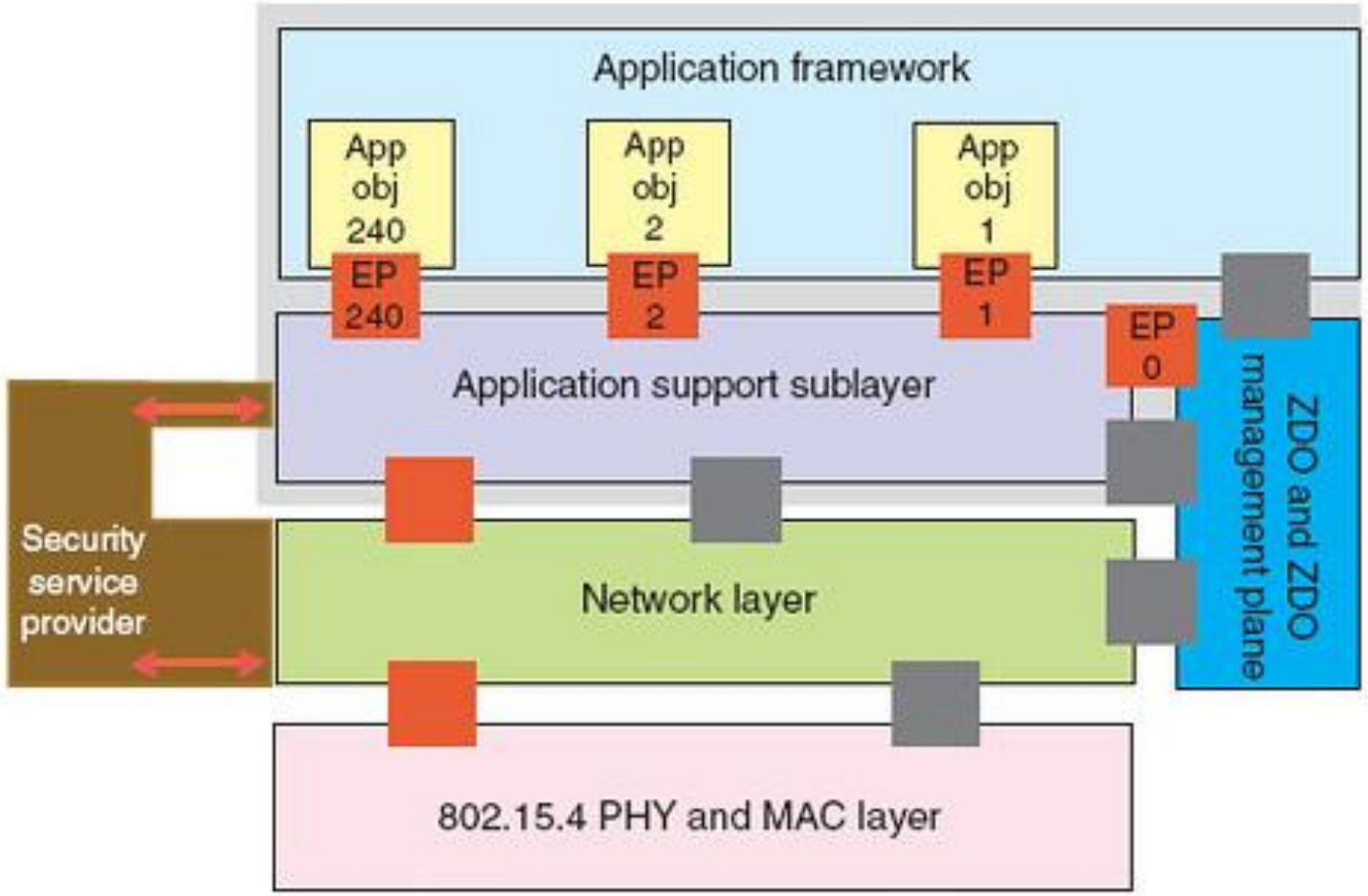
- Channel Access
  - The coordinator assigns only one channel to the network
- Methods of channel access
- Contention based - need not synchronized and used CSMA
  - CSMA: Channel goes to receive mode and check for any signal, if no signal is detected then it will start data transmission or else wait for a random period of time
- Contention free - Allocates specific time slot to each device
  - GTS- Guaranteed Time Slot

ZIGBEE

ZigBee characteristics

- Low power consumption – extended battery life
- Low data rate – 200-250 kbps
  - Wi-Fi- 11 Mbps, Bluetooth – 1 Mbps
- Short range
  - 75-100 meter indoor
  - Up to 300 meters outdoor
- Network join time- 30 sec
- Large and small networks – 65000 (theoretical)
- Low cost
- Uses AES encryption for security

ZIGBEE



Zigbee protocol stack

ZIGBEE

IEEE 802.15.4 PHY packet structure

Preamble (32)	Start of packet delimiter (8)	PHY header (8)	PHY Service Data Unit (PSDU) (0-1016 bits)

Preamble – synchronization
Start of packet delimiter
PHY header – PSDU length
PSDU – data field

ZIGBEE

IEEE 802.15.4 MAC frame format

- Data frame
- Beacon frame
- Acknowledgement frame
- Command frame

General MAC Frame Format

Octets : 2	1	0/2	0/2/8	0/2	0/2/8	Variable	2
Frame Control	Sequence number	Destination PAN identifier	Destination Address	Source PAN Identifier	Source Address	Frame payload	Frame Check Sequence
MAC header						MAC Payload	MAC footer

ZIGBEE

IEEE 802.15.4 MAC frame format - Beacon Frame Format

Octets : 2	1	4 or10	2	Variable	variable	Variable	2
Frame control	Beacon sequence number	Source address information	Super frame specification	GTS field	Pending address field	Beacon payload	Frame check sequence
MAC header			MAC payload				MAC Footer

- **The beacon frame is transmitted periodically by the PAN coordinator.**
- **It provides information about the network management through the super frame and GTS fields.**
- **It also synchronizes the network devices and indicates the proper communication period for them.**

ZIGBEE

IEEE 802.15.4 MAC frame format

Command Frame Format

Octets: 2	1	4 to 20	1	Variable	2
Frame control	Data Sequence number	Address information	Command type	Command payload	Frame Check sequence
	MAC header		MAC payload		MAC Footer

- **Useful for communication between the network devices.**
- **The command identifier specifies actions like association, disassociation, and data, GTS or beacon request.**

ZIGBEE

IEEE 802.15.4 MAC frame format

Data Frame Format

Octets : 2	1	4 to 20	Variable	2
Frame Control	Data Sequence number	Address information	Data payload	Frame check sequence
	MAC header		MAC payload	MAC footer

Acknowledgment Frame Format

Octets : 2	1	2
Frame control	Data Sequence Number	Frame Check sequence
	MAC header	MAC footer

ZIGBEE

Parameters	Description
Radio Technology	IEEE 802.15.4
Frequency band/Channels	2.4 GHz (ISM band) 16-channels (2 MHz wide)
Data rate	250 Kbits/sec
Encryption support	AES-128 at Network Layer
Communication range	75-100 meter indoor 300+ meters line of sight
Network size (Theoretical)	Up to 65,000

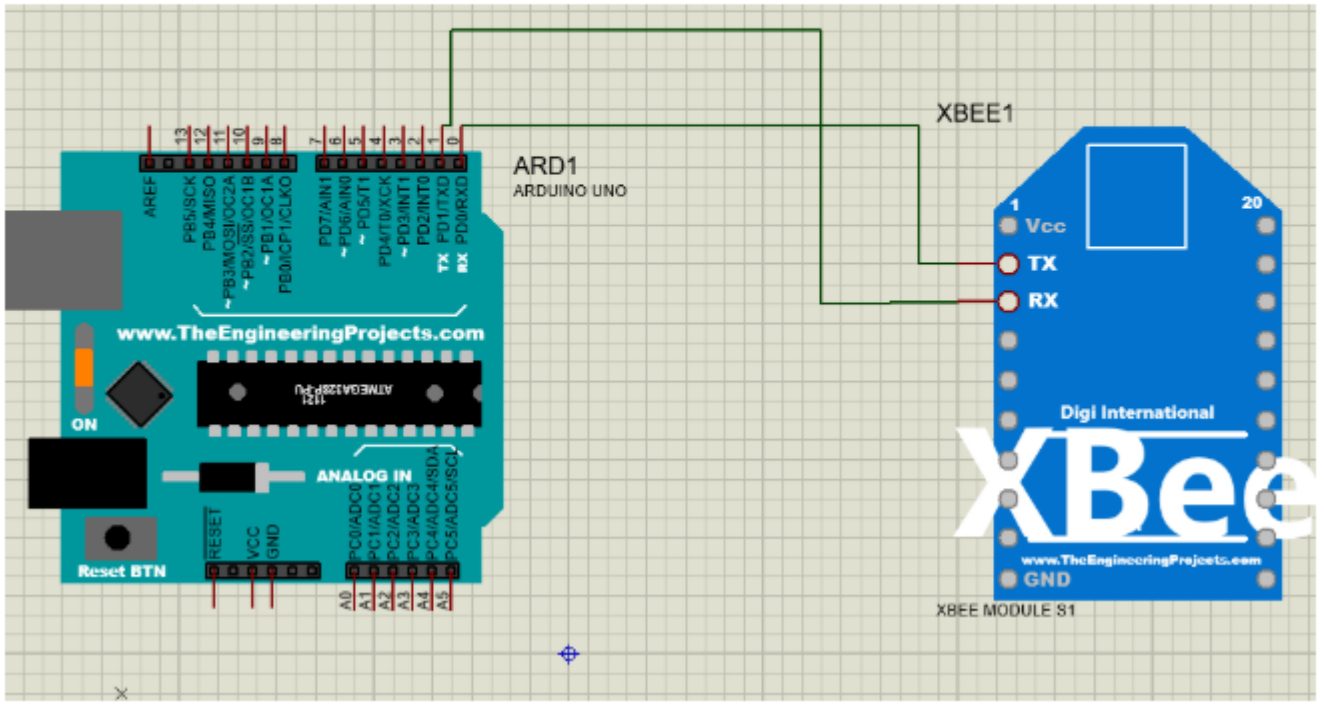
ZIGBEE

Advantages of ZigBee Protocol

- *Simple* setup and maintenance
- *Low power consumption*, ideal for *long lifetime devices*
- It can be used for lighting, security, appliance and home access
- *Low latency* and *low data rate*
- It offers *network flexibility* and *scalability*

ZigBee connection with Arduino

Gnd Xbee - Gnd Arduino
Vcc Xbee - 3.3V Arduino
Tx - Rx pin D0 Arduino
Rx - Tx pin D1 Arduino



ZIGBEE

Arduino Coding for Zigbee interface

```
#include <SoftwareSerial.h>
SoftwareSerial XBee(2, 3);
void setup()
{
  XBee.begin(9600);
  Serial.begin(9600);
}
void loop()
{
  if (Serial.available())
  {
XBee.write(Serial.read());
      }
  if (XBee.available())
  {

Serial.write(XBee.read());
      }
}
```

Bluetooth

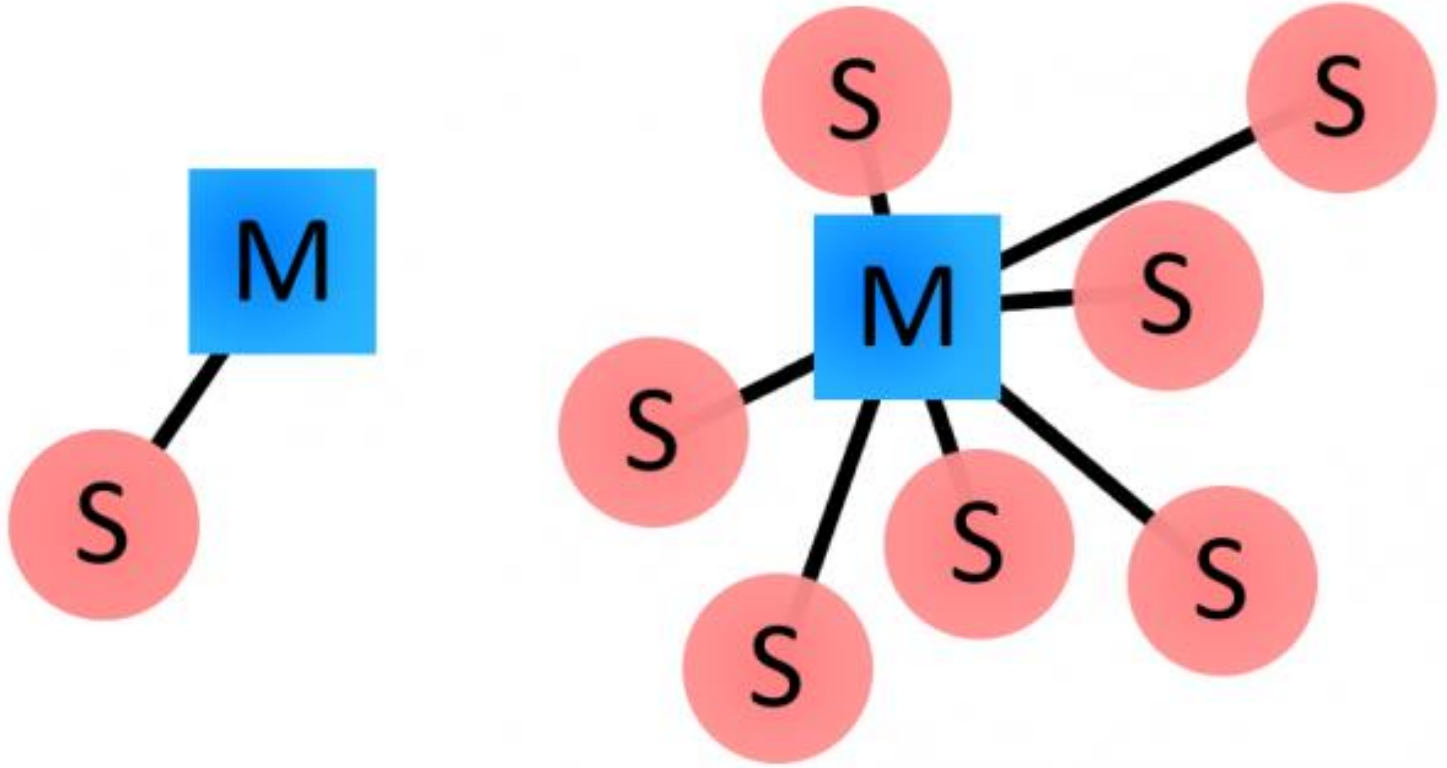
BLUETOOTH

- Bluetooth is a transceiver protocol works in 2.4GHz wireless link
- It's a secure protocol, and it's perfect for *short-range, low-power, low-cost, wireless transmissions*
- Bluetooth is a dynamic standard where devices can automatically find each other, establish connections, and discover what they can do for each other on an ad hoc basis.
- Bluetooth networks can be referred as piconet and scatternet
- It use a master/slave model

BLUETOOTH : INTRODUCTION

- A single master device can be connected to up to seven different slave devices
- The master coordinates communication throughout the piconet.
  - It can send data to any of its slaves and request data from them as well
- Slaves are only allowed to transmit to and receive from their master.
  - They can't talk to other slaves in the piconet
- Every single Bluetooth device has a unique 48-bit address, commonly abbreviated BD_ADDR. This will usually be presented in the form of a 12-digit hexadecimal value

BLUETOOTH



Examples of Bluetooth master/slave piconet topologies.

BLUETOOTH

VERSIONS

Bluetooth 1.0

1998.10 – 2003. 11
"Base Rate"
- 1Mbps data rate
- V1.0 - Draft
- V1.0A - published on 1999.7
- V1.0B Enhanced the Interoperability
- V1.1 - IEEE 802.15.1
- V1.2 Enhanced the compatibility

Bluetooth 2.0 + EDR

2004. 11 – 2007. 7
"Enhanced Data Rate"
- Higher ordered modulation for data payload
- 2Mbps or 3Mbps physical data rate

- V2.0
- V2.1

Bluetooth 3.0 + HS

2009. 4
"HS Mode"
- AMP Alternative MAC/PHY
- Implement high data rate by using 802.11 protocols.
- Facing the Challenge from Wi-Fi

- V3.0

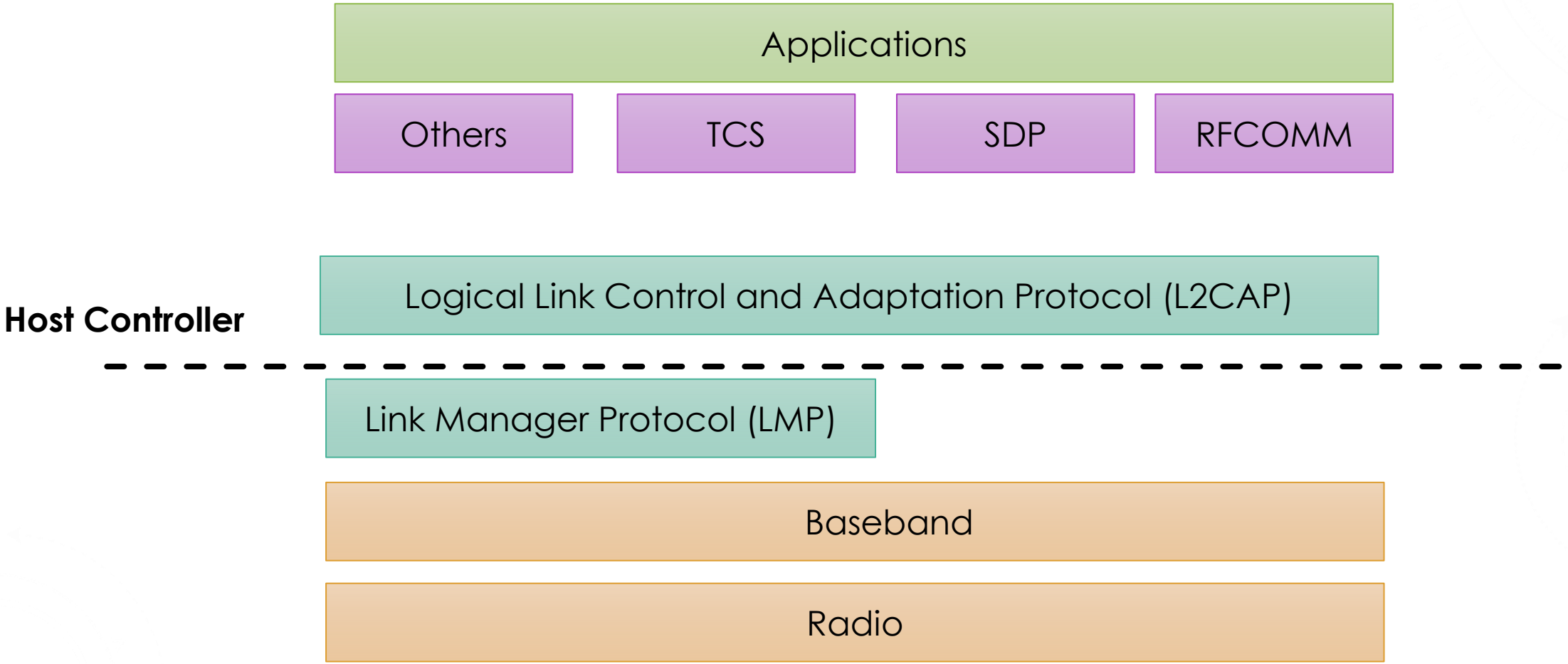
Bluetooth 4.0

2010. 6 – 2014. 12
"Low Energy"
- Facing the IoT application
- Changed the protocol greatly, almost a new technology

- V4.0
- V4.1
- V4.2

BLUETOOTH

ARCHITECTURE



BLUETOOTH : ARCHITECTURE

- **Radio:**
  - This layer defines the requirements for a transceiver to operate in the 2.4 GHz ISM band such as frequency bands, frequency hopping specifications, and modulation techniques.

- **Baseband:**
  - Baseband layer describes the specification of the Bluetooth Link Controller (LC), and carries baseband protocol.
  - Defines the addressing scheme, packet frame format, timing, and power control algorithms.
  - Two types of link can be created in baseband are ACL and SCO.

BLUETOOTH

Two types of link between master and slave

- **Asynchronous Connection Less (ACL)**
  - Packet switched data
  - Slave can have only one ACL link to master
  - Used for correct delivery over fast delivery
  - Maximum data rate of 721 kbps

- **Synchronous Connection Oriented (SCO)**
  - Real time data transmission
  - Damaged packet cannot be retransmitted
  - Data rate of 64 kbps

BLUETOOTH: ARCHITECTURE

- **LMP:**
  - Used by the Link managers for link set-up and control.
  - Other main functions of LMP are device authentication, message encryption, and negotiation of packet sizes.

- **Host Controller Interface (HCI):**
  - Provides a command interface to the Baseband LC and Link Manager, to access hardware status and control registers.

- **L2CAP:**
  - Logical Link Control and Adaptation Protocol (L2CAP) supports higher level protocol multiplexing, packet segmentation and reassembly, and also conveys the QoS.

BLUETOOTH: ARCHITECTURE

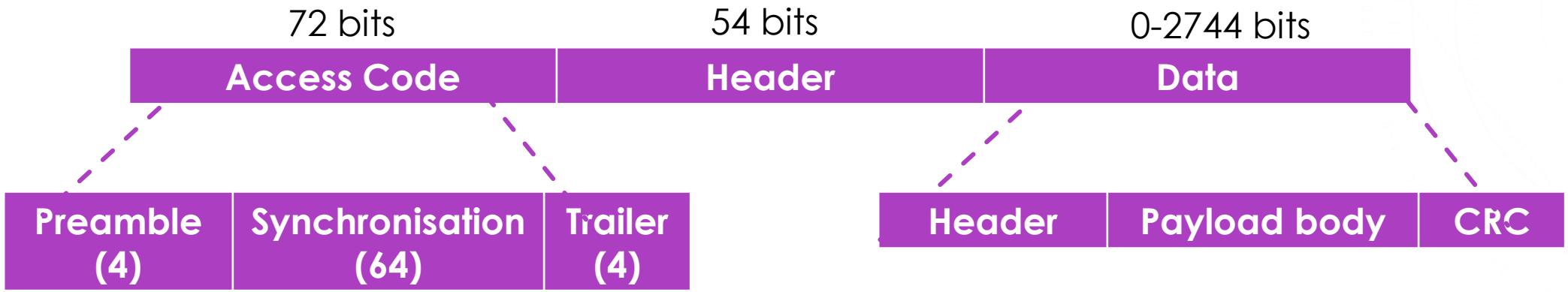
- **RFCOMM**:
  - RFCOMM protocol provides emulation of serial ports over the L2CAP protocol.

- **SDP**: Service Discovery Protocol (SDP)
  - Provides a means for applications to discover which services are provided by or available through a Bluetooth device

- **TCS**: telephony control protocol for telephony service

- **Applications**: This includes the application profiles that allow the user to interact with the Bluetooth applications.

BLUETOOTH: Frame Format



Access code (AC)

- Used for packet identification
- ***Once packet is received, receiver in piconet will compare the incoming signal with AC, if any mismatch is found packet will be ignored***
- **72 bit AC** is derived from master identity
- It is used for synchronization
- **Three types**
  - **Channel AC** – targets piconet ID
  - **Device AC** – individual devices
  - **Inquiry AC** – used during pairing

Data - contains data or control information from upper layer

Header – identifies logical channel

Payload body – contains user data

CRC- 16 bit checksum

BLUETOOTH: Frame Format

Header

- **Address** – it can address up to **7 slave device**s, if it is 0, then messages will broadcasted
- **Type** – defines type of **incoming data**
  - **ACL** – Data medium (DM) or Data High (DH)
  - **SCO** - Data Voice (DV) and High Quality Voice (HV)
- **12 types** of packet for each SCO and HV
- **4 common control packets**
  - **Flow-** 1 bit flow control
  - **ARQN** – 1 bit acknowledgement
  - **SEQN** – 1 bit sequential numbering scheme for packet ordering
  - **HEC** – Header error check
- **Header** = 3 X 18 bits = 54 bits

BLUETOOTH: Bonding & Pairing

- Pairing requires an **authentication process to establish** connection between devices.
- Sometimes pairing is a simple **"Just Works"** operation,
  - where the click of a button is all it takes to pair (this is common for devices with no UI, like headsets).
- Older, legacy **(v2.0 and earlier),** pairing processes involve the entering of a common **PIN code** on each device.
  - The PIN code can range in length and complexity from four numbers (e.g. "0000" or "1234") to a 16-character alphanumeric string.

BLUETOOTH: Connection Process

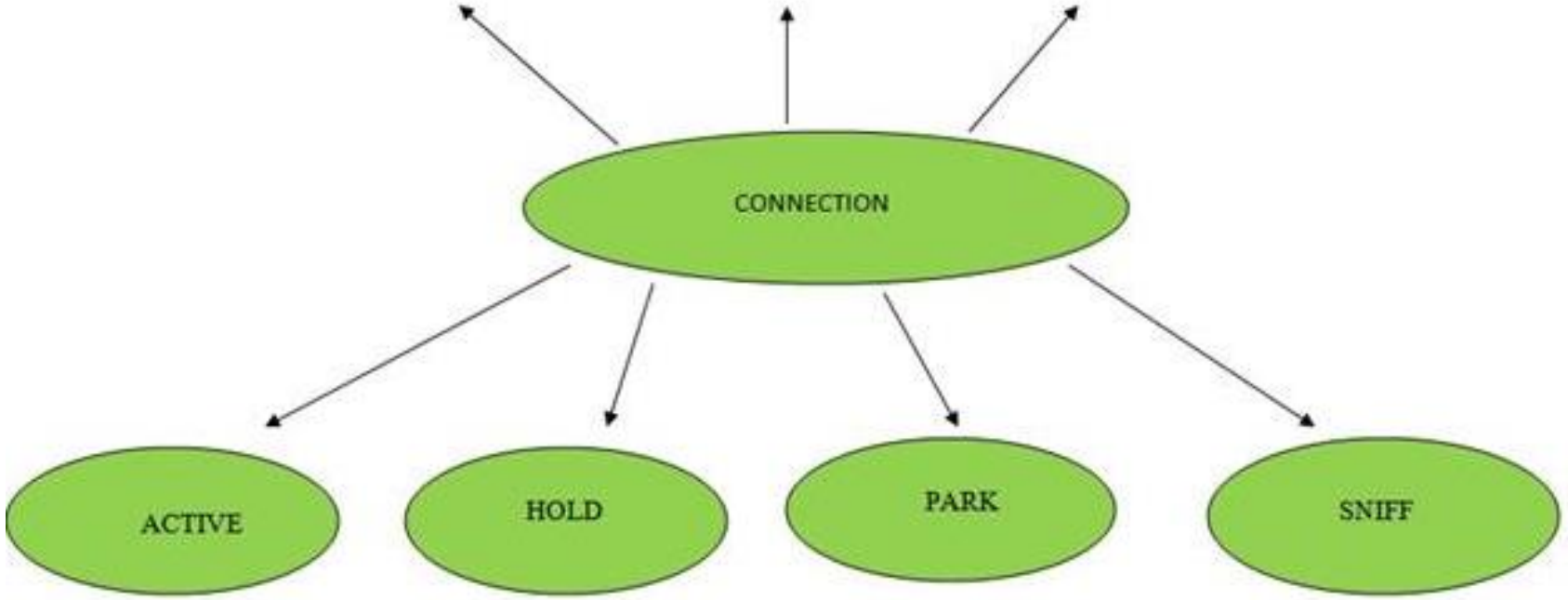
- Creating a Bluetooth connection between two devices is a multi-step process involving three progressive states.

- Step-1 Inquiry:
  - If two Bluetooth devices know absolutely nothing about each other
  - One must run an inquiry to try to discover the other
  - On receiving the inquiry, the device which listens to such a request will send back address, name and other information

BLUETOOTH: Connection Process

- **Step-2 Paging (Connecting) –**
  - Paging is the process of forming a connection between two Bluetooth devices.
- **Step-3 Connection –**
  - After a device has completed the paging process, it enters the connection state.
  - The mode of operation of device is decided in connection phase

BLUETOOTH: Connection Modes



BLUETOOTH: Connection Modes

- **Active Mode –**
  - Regular connected mode, where device send and receive data (regular connection up-to 7 devices)
- **Sniff Mode –**
  - Power saving mode, less active mode (frees slave for predetermined period of time, recurring fixed time slot)
- **Hold Mode –**
  - Temporary mode, device will be sleeping for a particular period of time (frees slave for predetermined period of time, one time slot)
- **Park Mode –**
  - Sleep mode with infinite time, slave will wake only if master tells it to wake up (maximum of 255 devices can be added)

BLUETOOTH: Power Class

- The transmit power determines the range of a Bluetooth module and it is defined by its power class.
- There are three defined classes of power:

Class Number	Max Output Power	Max Output Power	Max Range
Class 1	20 dBm	100 mW	100 m
Class 2	4 dBm	2.5 mW	10 m
Class 3	0 dBm	1 mW	10 cm

BLUETOOTH: HC05 – BLUETOOTH MODULE

- HC-05 module is an easy-to-use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup.

- The HC-05 Bluetooth Module can be used in a Master or Slave configuration, making it a great solution for wireless communication.

- This serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps modulation with complete 2.4GHz radio transceiver and baseband.

BLUETOOTH: HC05 – BLUETOOTH MODULE

- The Bluetooth module HC-05 is a MASTER/SLAVE module.
  - By default the factory setting is SLAVE.
- Role of the module (Master / Slave) can be configured only by AT COMMANDS.
- Master module can initiate a connection to other devices.
  - The user can use it simply for a serial port replacement to establish connection between MCU and GPS, PC to your embedded project, etc.

BLUETOOTH: HC05 – BLUETOOTH MODULE

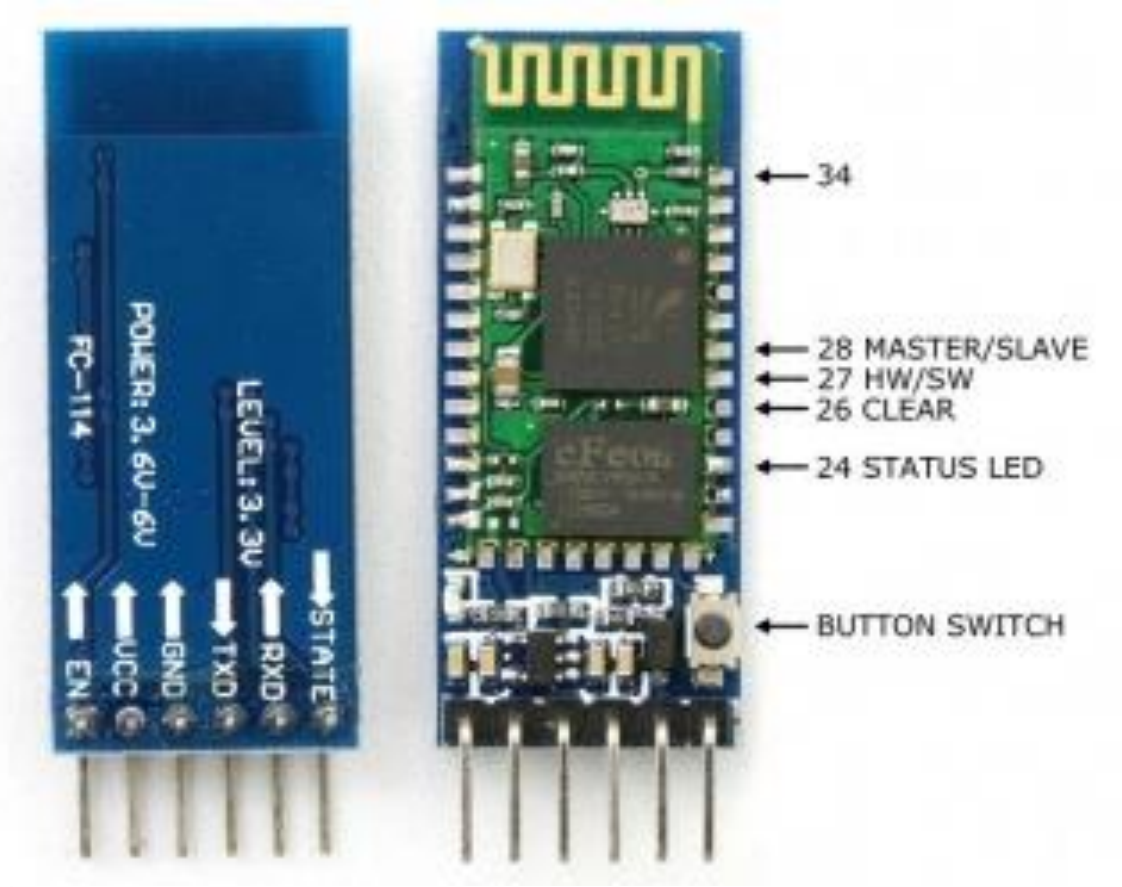
❑ Hardware Features

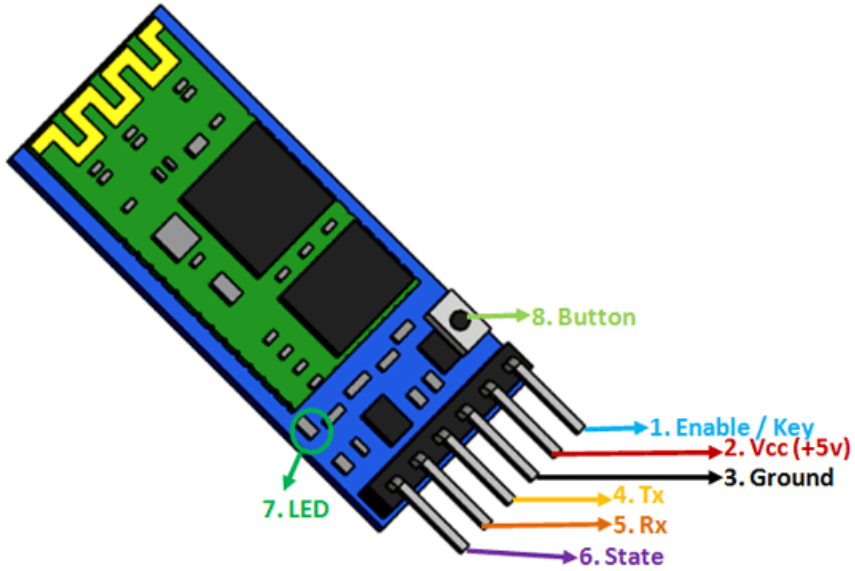
- Up to +4 dBm RF transmit power.
- 3.3 to 5 V I/O.
- PIO (Programmable Input/Output) control.
- UART interface with programmable baud rate.
- With integrated antenna.

❑ Software Features

- Slave default Baud rate: 9600,
- Data bits:8, Stop bit:1,Parity:No parity.
- Auto-connect to the last device on power as default.
- Permit pairing device to connect as default.
- Auto-pairing PINCODE:"1234" as default

BLUETOOTH: HC05 – BLUETOOTH MODULE





BLUETOOTH: HC05 – BLUETOOTH MODULE

Arduino

Write a program to control a LED connected at Pin 7 of Arduino through command send by Bluetooth supported mobile device.

```
#define ledPin 7
int state = 0;
void setup() {
    pinMode(ledPin, OUTPUT);
    digitalWrite(ledPin, LOW);
    Serial.begin(38400);
}
void loop() {
if(Serial.available() > 0){
// Checks whether data is comming from the serial port
        state = Serial.read();
// Reads the data from the serial port
}
if (state == '0') {
digitalWrite(ledPin, LOW); // Turn LED OFF
Serial.println("LED: OFF"); // Send back, to the phone, the String "LED: ON"
state = 0;
}
else if (state == '1') {
digitalWrite(ledPin, HIGH);
Serial.println("LED: ON");;
state = 0;
}
}
```

Wi-Fi

Wi-Fi

- WiFi is an alternative network to wired network which is commonly used for connecting devices in wireless mode.
- WiFi is stand for Wireless Fidelity is generic term that refers to IEEE802.11 standard for Wireless Local Networks or WLANs.
- WiFi connects computers to each other, to the internet and to the wired network.

Wi-Fi

Standards	Year of Release	Description
Wi-Fi-1 (802.11b)	1999	This version has link speed from 2Mb/s to 11 Mb/s over 2.4 Ghz frequency band
Wi-Fi-2 (802.11a)	1999	After a month of release previous version, 802.11a was released and it provide upto 54 Mb/s link speed over 5 Ghz band
Wi-Fi-3 (802.11g)	2003	In this version the speed was increased up to 54 to 108 Mb/s over 2.4 Ghz
802.11i	2004	This is same as 802.11g but only the security mechanism was increased in this version
802.11e	2004	This is also same as 802.11g, only Voice over Wireless LAN and multimedia streaming are involved
Wi-Fi-4 (802.11n)	2009	This version supports both 2.4 Ghz and 5 Ghz radio frequency and it offers up to 72 to 600 Mb/s speed
Wi-Fi-5 (802.11ac)	2014	It supports speed of 1733 Mb/s in 5 Ghz band

Wi-Fi: Elements of Network

- WIFI uses radio technology to transmit and receive data at high speed.
- Access Point (AP) –
  - AP is a wireless LAN transceiver or "base station" that can connect one or many wireless devices simultaneously to the Internet.
- Wi-Fi cards –
  - Accept the wireless signal and relay information. They can be internal and external.
- Safeguards –
  - Firewalls and anti-virus software protect networks from uninvited users and keep information secure.

Wi-Fi: Elements of Network

- **SSID (Service Set Identifier) :**
  - It is a 32-character name which identifies the Wi-Fi network and differentiate one Wi-Fi to another Wi-Fi.
  - All the devices are attempting to connect a particular SSID.
  - Simply, SSID is the name of the wireless network.

- **WPA-PSK (Wi-Fi Protected Access- Pre-Shared Key) :**
  - Program developed by the Wi-Fi Alliance Authority to secure wireless networks with the use of Pre-Shared Key(PSK) authentication.
  - WPA has 3 types, such as WPA. WPA2, WPA3.
  - It is a way of encrypting Wi-Fi signal to protect from unwanted users.

- Wi-Fi uses Ad-Hoc networks to transmit.
- It is a point-to-point network without any interface.

Wi-Fi: Topologies

- **Peer-to-peer topology** (Ad-hoc Mode)
- **AP-based topology** (Infrastructure Mode)

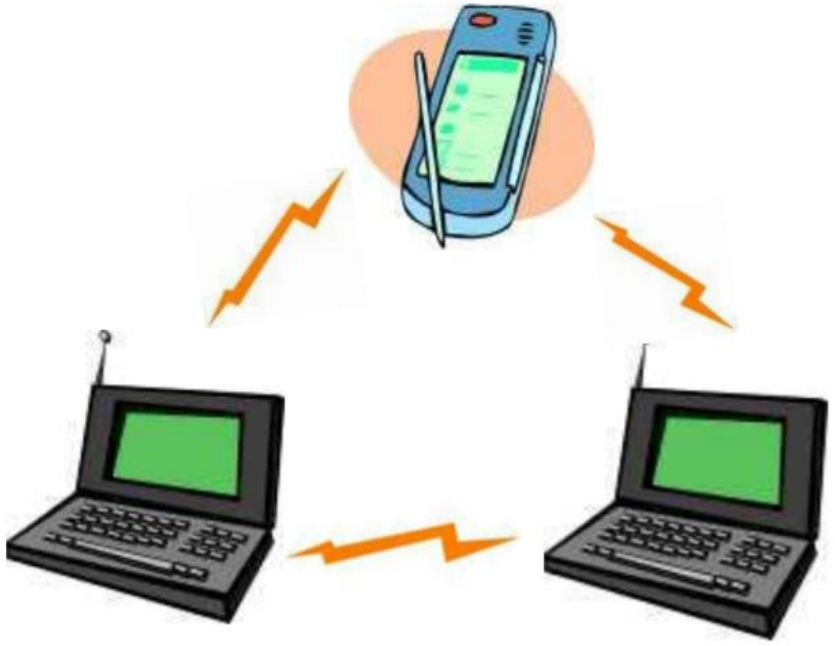
Wi-Fi

Peer-to-peer Topology

- AP is not required.
- Client devices within a cell can communicate with each other directly.
- It is useful for setting up a wireless network quickly and easily.

Wi-Fi

Peer-to-peer Topology



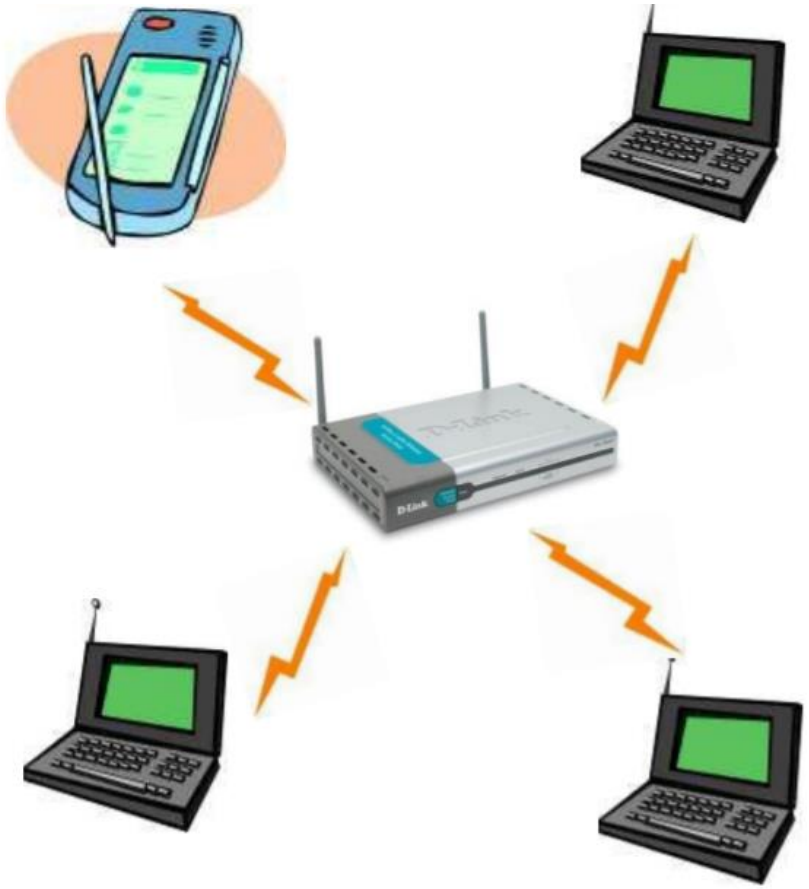
Wi-Fi

Infrastructure network

- The client communicate through Access Point.
- Any communication has to go through AP.
- If a Mobile Station (MS), like a computer, a PDA, or a phone, wants to communicate with another MS, it needs to send the information to AP first, then AP sends it to the destination MS.

Wi-Fi

Infrastructure network



Wi-Fi

Hotspots

- Hotspot is a geographical area that has a readily accessible wireless network.
- Hotspots are equipped with broad band Internet connection and one or more Access points that allow users to access the internet wirelessly.
- Hotspots can be setup in any public location that can support an Internet connection.

Wi-Fi

How a Wi-Fi Network Works

- Wi-Fi hotspot is created by installing an access point to an internet connection.
- An access point acts as a base station.
- When Wi-Fi enabled device encounters a hotspot the device can then connect to that network wirelessly.

Wi-Fi

How a Wi-Fi Network Works

- A single access point can support up to 30 users and can function within a range of 100 – 150 feet indoors and up to 300 feet outdoors.
- Many access points can be connected to each other via Ethernet cables to create a single large network.

Wi-Fi

Advantages

- Mobility
- Ease of Installation
- Flexibility
- Cost
- Reliability
- Security
- Use unlicensed part of the radio spectrum
- Roaming
- Speed

Wi-Fi: Topologies

Limitations

- Interference
- Degradation in performance
- High power consumption
- Limited range

Wi-Fi: Topologies

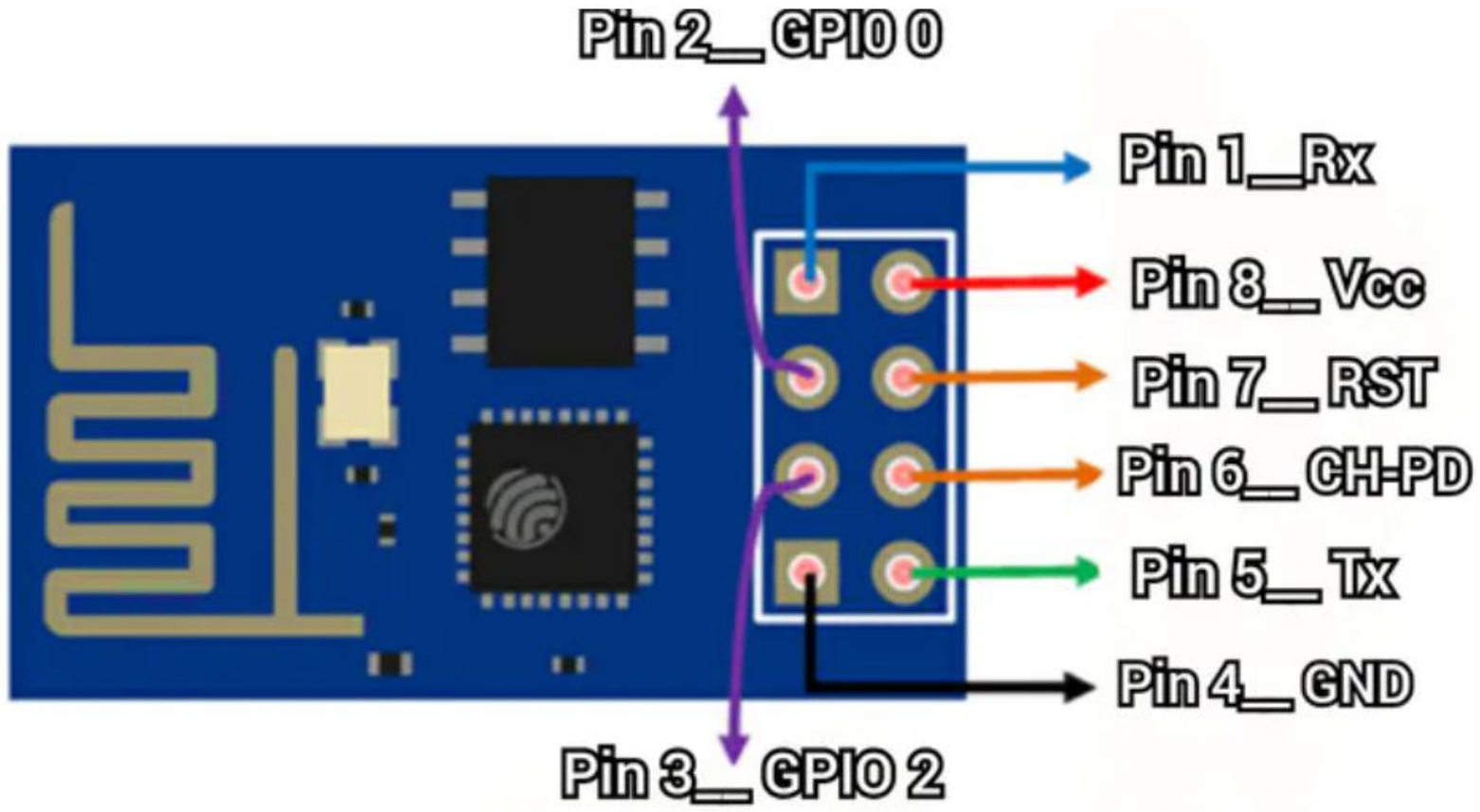
Limitations

- Interference
- Degradation in performance
- High power consumption
- Limited range

Wi-Fi

Interface with Arduino

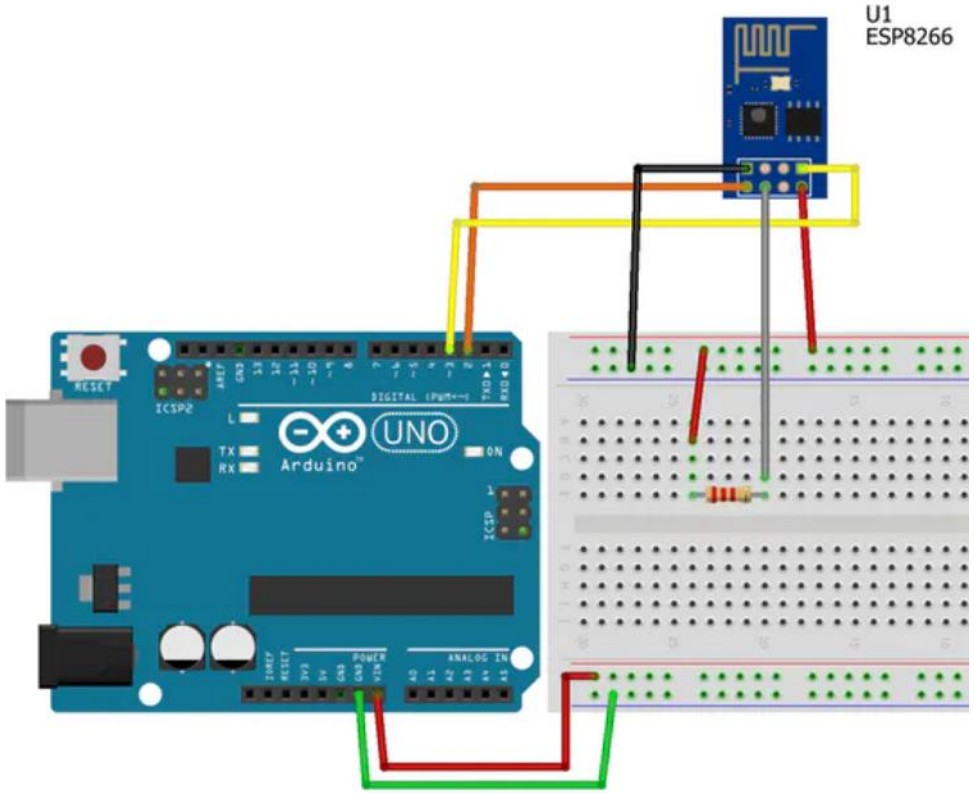
- ESP8266



Wi-Fi

Interface with Arduino

- **ESP8266**



Wi-Fi

Interface with Arduino

```
#include <SoftwareSerial.h>
SoftwareSerial ESPserial(2, 3); // RX | TX
void setup()
{
Serial.begin(115200); // communication with the host computer
//while (!Serial) { ; }
// Start the software serial for communication with the ESP8266
ESPserial.begin(115200);
Serial.println("");
Serial.println("Remember to set Both NL & CR in the serial monitor.");
Serial.println("Ready");
Serial.println("");
}
```

Wi-Fi

Interface with Arduino

```
void loop()
{
// listen for communication from the ESP8266 and then write it to the serial
monitor
if ( ESPserial.available() ) { Serial.write( ESPserial.read() ); }
// listen for user input and send it to the ESP8266
if ( Serial.available() ) { ESPserial.write( Serial.read() ); }
}
```

RS232

Introduction

- RS-232 (Recommended Standard 232)
- Interference Standard
- Commonly used in computer serial ports.
- Standard defines the electrical characteristics and timing of signals.
- Standard, TIA – 232 – F

History

- RS-232 was introduced in 1962
- RS-232 compatible port was a standard feature for serial communication.
- Standard was revised and updated by Electronic Industries Alliance and since 1988, by Telecommunications Industry Association.
- C version of standard issued in 1969

Standard

- In Rs232, user data is sent as a tine series of bits.
- Both synchronous and async. Transmissions are supported
- RS232 devices can be classified as Data Terminal Equipment (DTE) or Data Communication Terminal (DCE), defines that each device which wires will be sending and receiving each signal.
  - DTE : Refers to the terminals and computers that send and receive the data
  - DCE : Refers to the communication equipment, such a modems, responsible for transmitting the data.

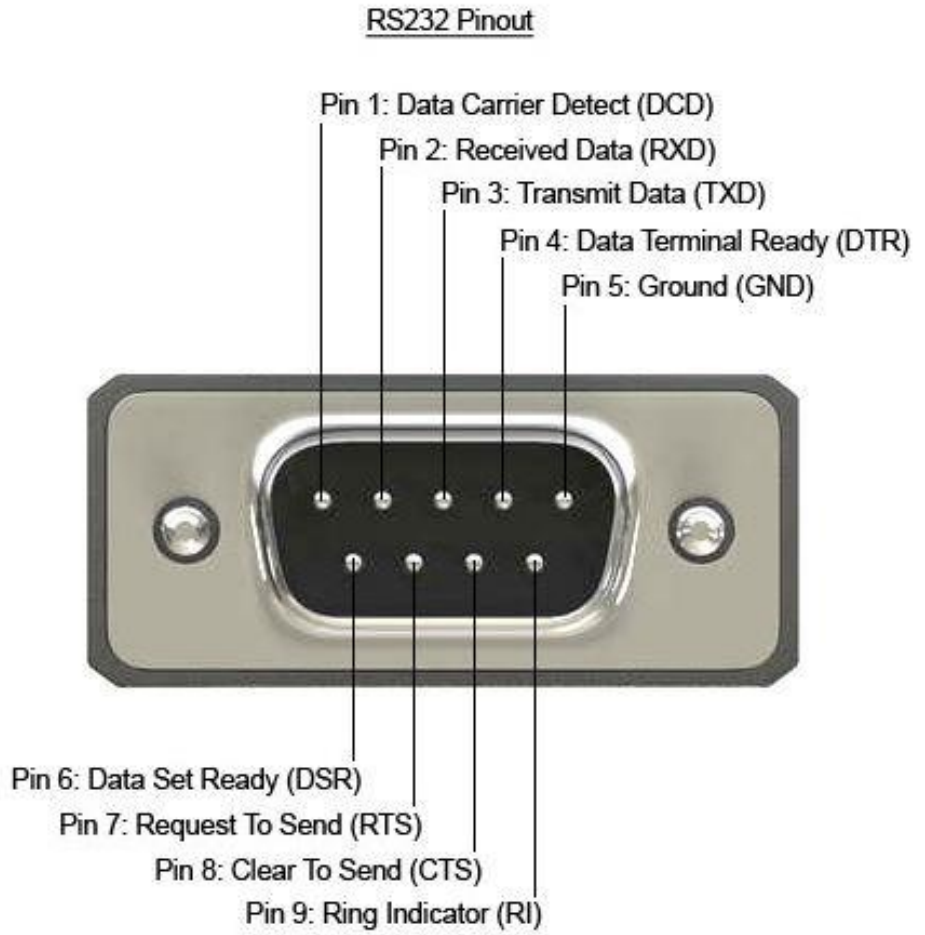
Limitations

- Limited transmission speed,
- Relatively large voltage swing,
- Big standard connectors
- Multi-drop connection amongst more than 2 devices is not defined.
- Also, multi-drop have limitations in speed and compatibility.

- This led to the development of Universal Serial Bus (USB)

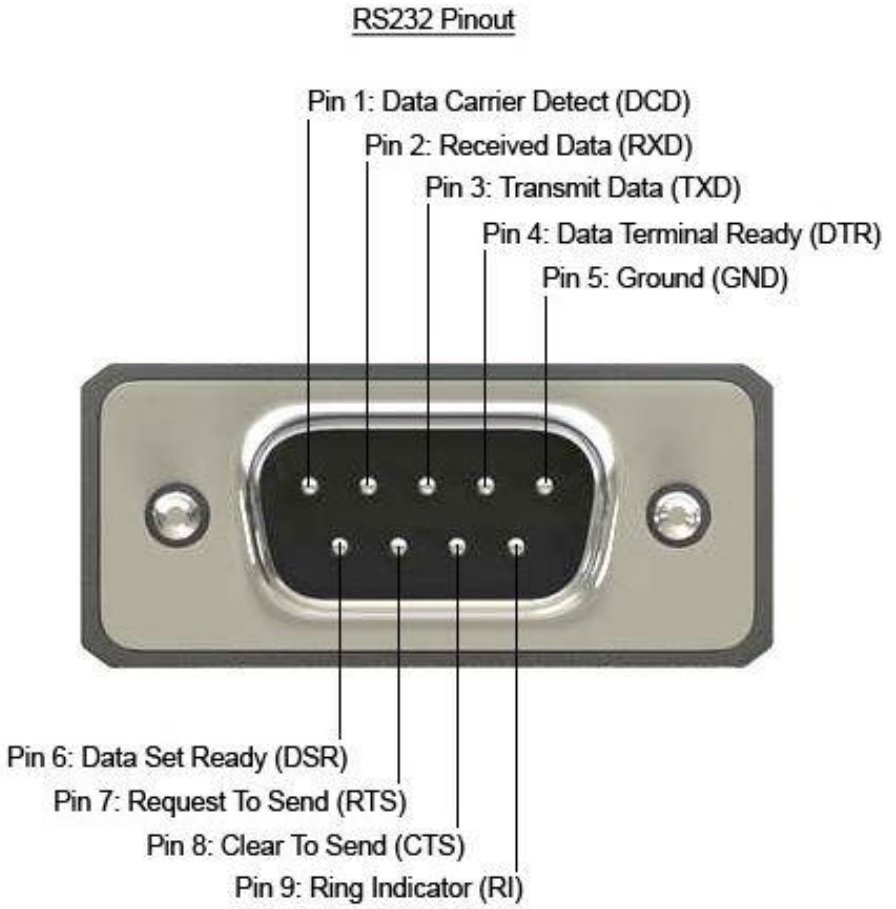
PINs

1. **Data Carrier Detect** – After a data terminal is detected, a **signal is sent** to the data set that is going to be transmitted to the terminal.
2. **Received Data** – The data set **receives** the **initial signal** via the **receive data line** (RxD).
3. **Transmitted Data** – The data terminal gets a **signal** from the data set, a **confirmation** that there is a connection between the data terminal and the data set.
4. **Data Terminal Ready** – A **positive voltage** is applied to the **data terminal ready** (DTR) **line**, a sign that the data terminal is prepared for the transmission of data.
5. **Signal Ground** – A return for all the signals on a single interface, the **signal ground** (SG) offers a return path for serial communications. Without SG, serial data cannot be transmitted between devices.



PINs

6. **Data Set Ready** – A **positive voltage** is applied to the **data set ready** (DSR) **line**, which ensures the **serial communications** between a data terminal and a data set can be completed.
7. **Request to Send** – A **positive voltage** indicates the **request to send** (RTS) can be performed, which means the data set is able to send information to the data terminal without interference.
8. **Clear to Send** – After a connection has been established between a data terminal and a distant modem, a clear to send (CS) signal ensures the data terminal recognizes that communications can be performed.
9. **Ring Indicator** – The ring indicator (RI) signal **will be activated** if a **modem** that **operates** as a data set **detects low frequency**. When this occurs, the **data terminal is alerted**, but the **RI will not stop the flow of serial data between devices**.



Handshaking in RS-232

- To ensure reliable data transmission between two devices, the data transmission must be coordinated.
- Many of the pins of RS-232 connector are used for handshaking.

13-1 IEEE STANDARDS

In 1985, the Computer Society of the IEEE started a project, called Project 802, to set standards to enable intercommunication among equipment from a variety of manufacturers. Project 802 is a way of specifying functions of the physical layer and the data link layer of major LAN protocols.

Topics discussed in this section:

Data Link Layer
Physical Layer

Figure 13.1 *IEEE standard for LANs*

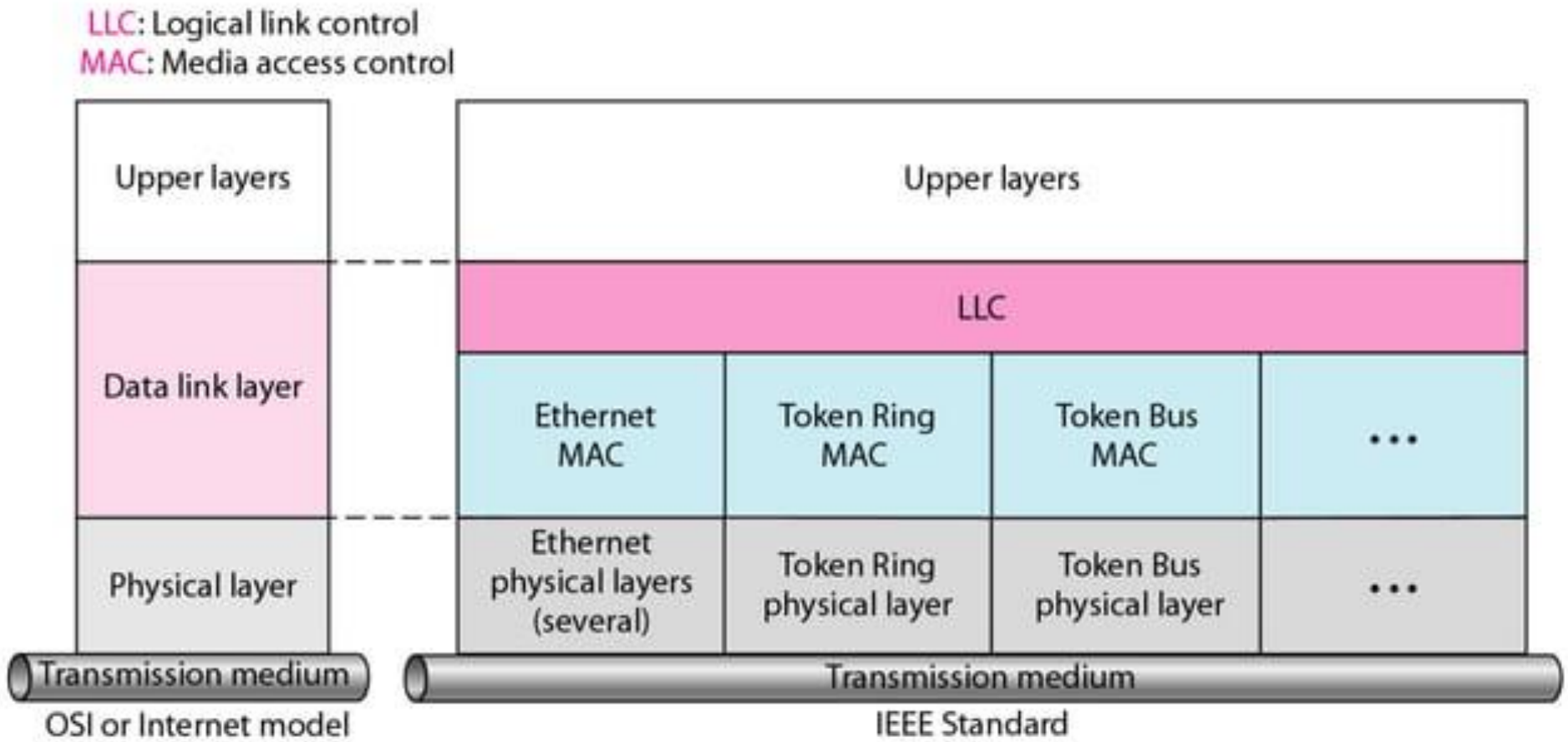
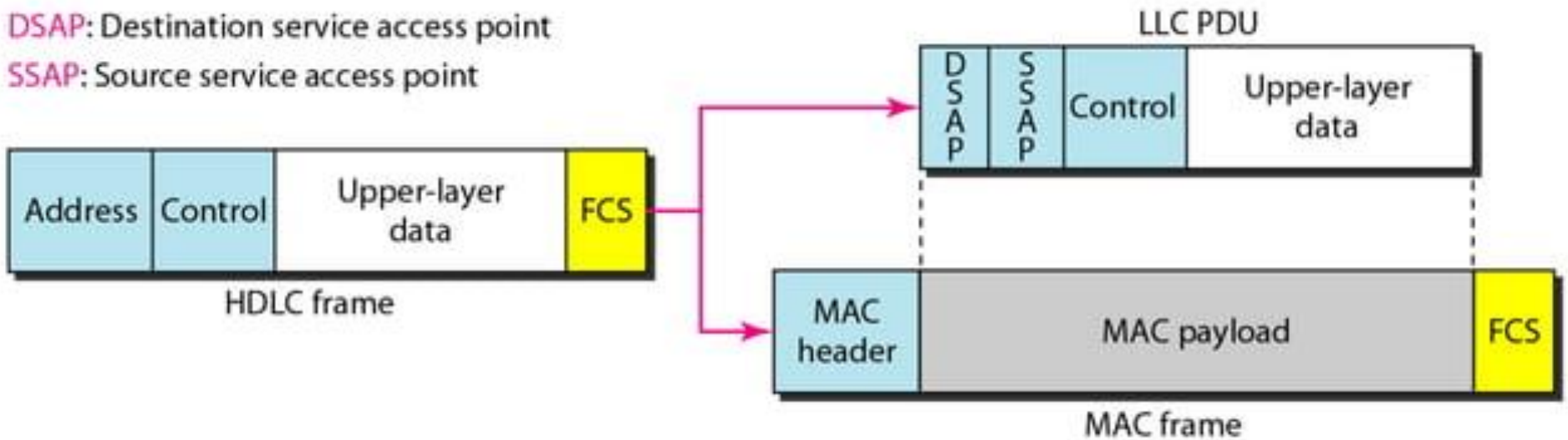


Figure 13.2 *HDLC frame compared with LLC and MAC frames*



13-2 STANDARD ETHERNET

The original Ethernet was created in 1976 at Xerox's Palo Alto Research Center (PARC). Since then, it has gone through four generations. We briefly discuss the ***Standard (or traditional) Ethernet*** *in this section.*

Topics discussed in this section:

MAC Sublayer
Physical Layer

Figure 13.3 *Ethernet evolution through four generations*

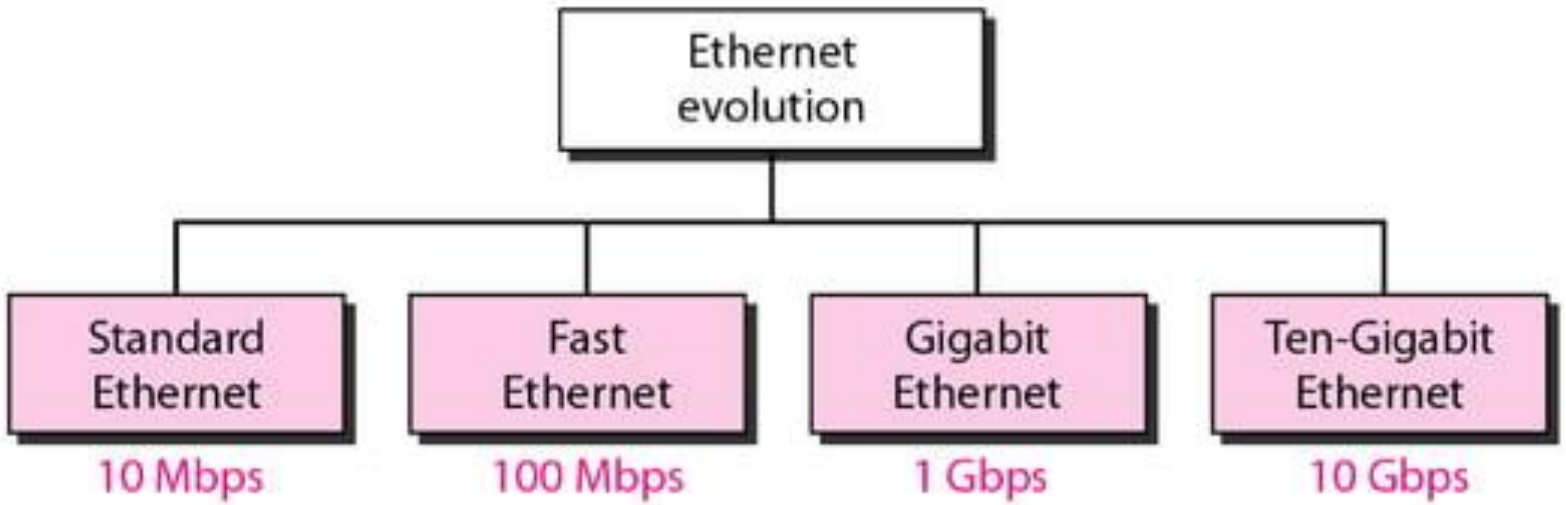


Figure 13.4 *802.3 MAC frame*

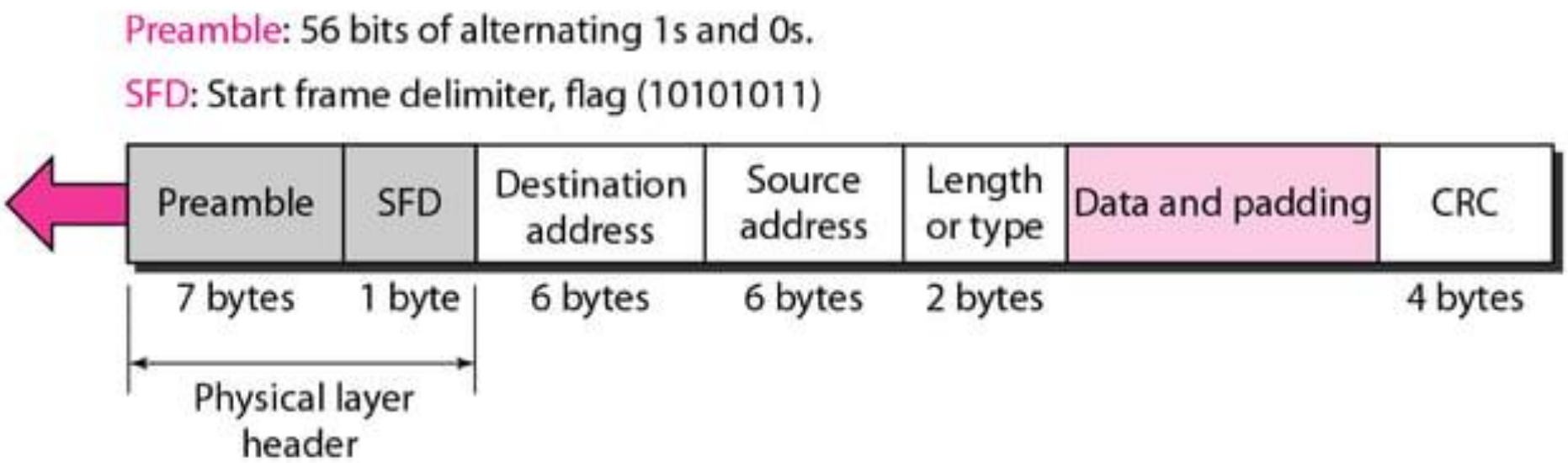
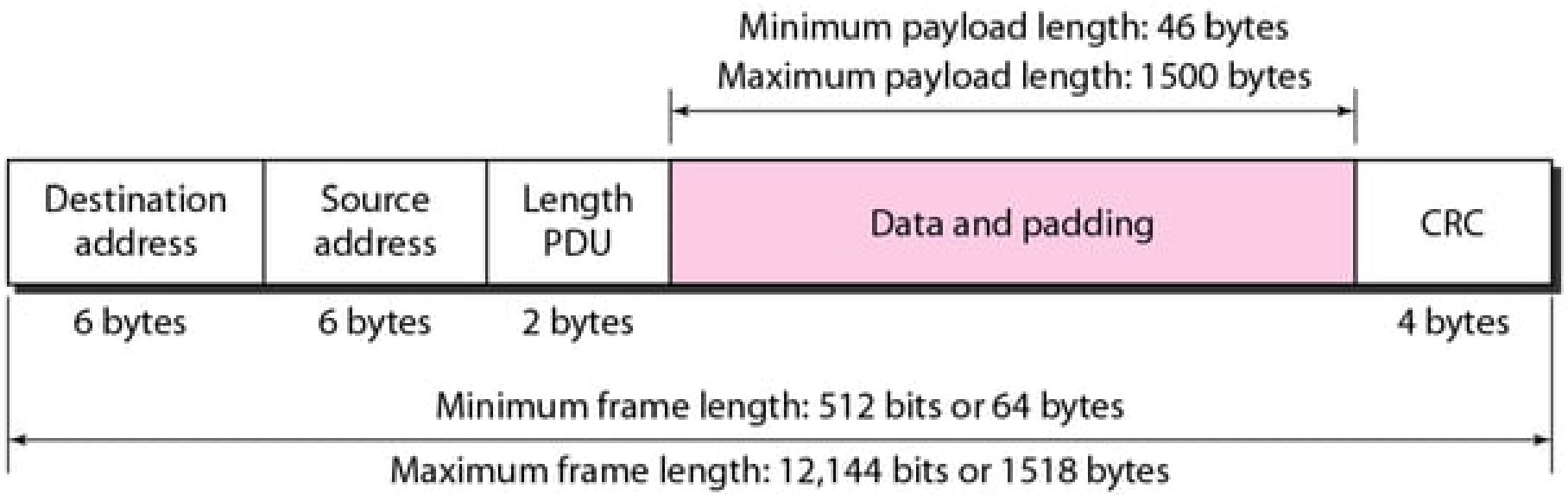


Figure 13.5 *Minimum and maximum lengths*



Note

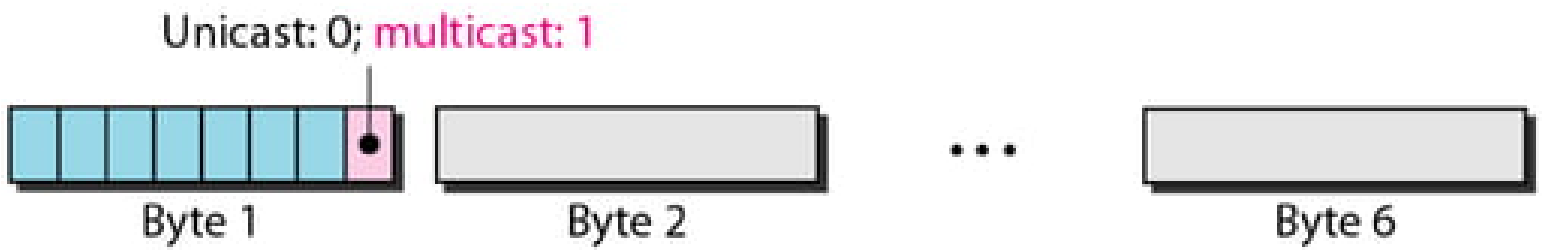
Frame length:

Minimum: 64 bytes (512 bits)

Maximum: 1518 bytes (12,144 bits)

Figure 13.6 *Example of an Ethernet address in hexadecimal notation*

Figure 13.7 *Unicast and multicast addresses*



Note

The least significant bit of the first byte defines the type of address. If the bit is 0, the address is unicast; otherwise, it is multicast.

Note

The broadcast destination address is a special case of the multicast address in which all bits are 1s.

Example 13.1

Define the type of the following destination addresses:

a. 4A:30:10:21:10:1A *b. 47:20:1B:2E:08:EE*

c. FF:FF:FF:FF:FF:FF

Solution

To find the type of the address, we need to look at the second hexadecimal digit from the left. If it is even, the address is unicast. If it is odd, the address is multicast. If all digits are F's, the address is broadcast. Therefore, we have the following:

a. This is a unicast address because A in binary is 1010.

b. This is a multicast address because 7 in binary is 0111.

c. This is a broadcast address because all digits are F's.

Example 13.2

Show how the address 47:20:1B:2E:08:EE is sent out on line.

Solution

The address is sent left-to-right, byte by byte; for each byte, it is sent right-to-left, bit by bit, as shown below:

Figure 13.8 *Categories of Standard Ethernet*

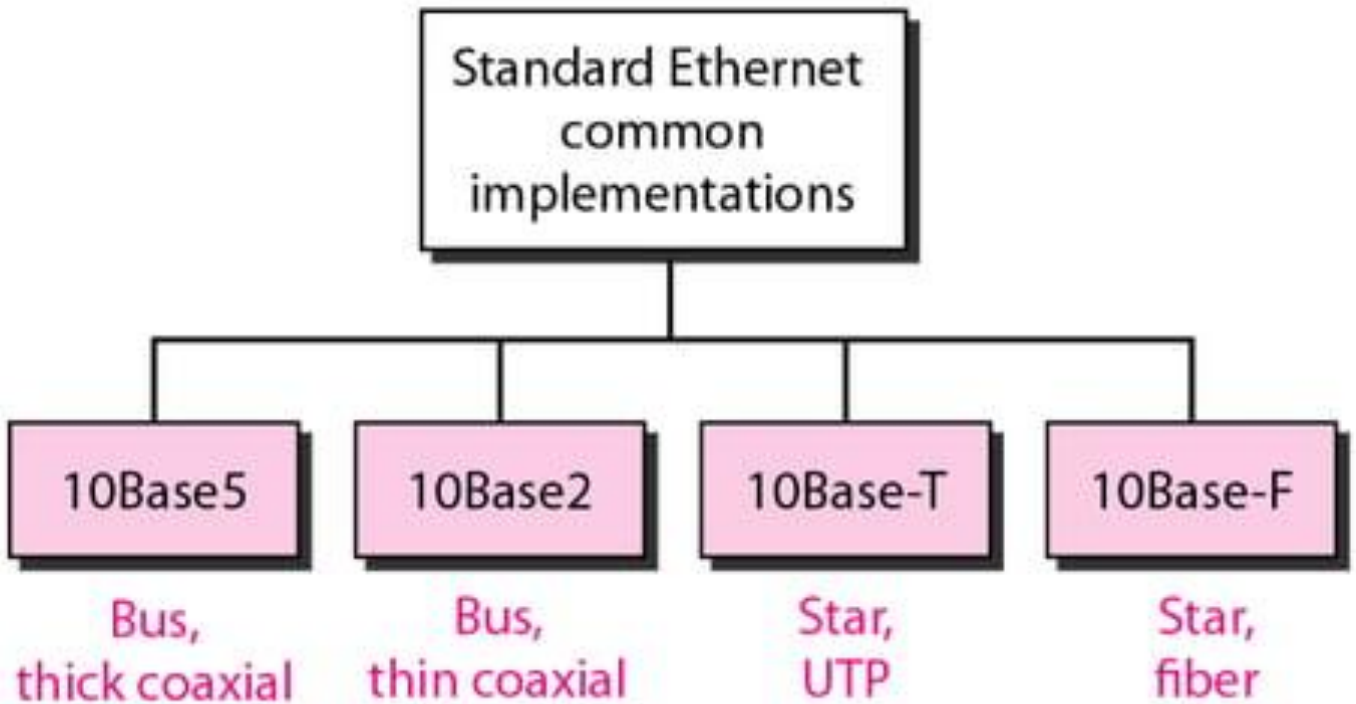


Figure 13.9 *Encoding in a Standard Ethernet implementation*

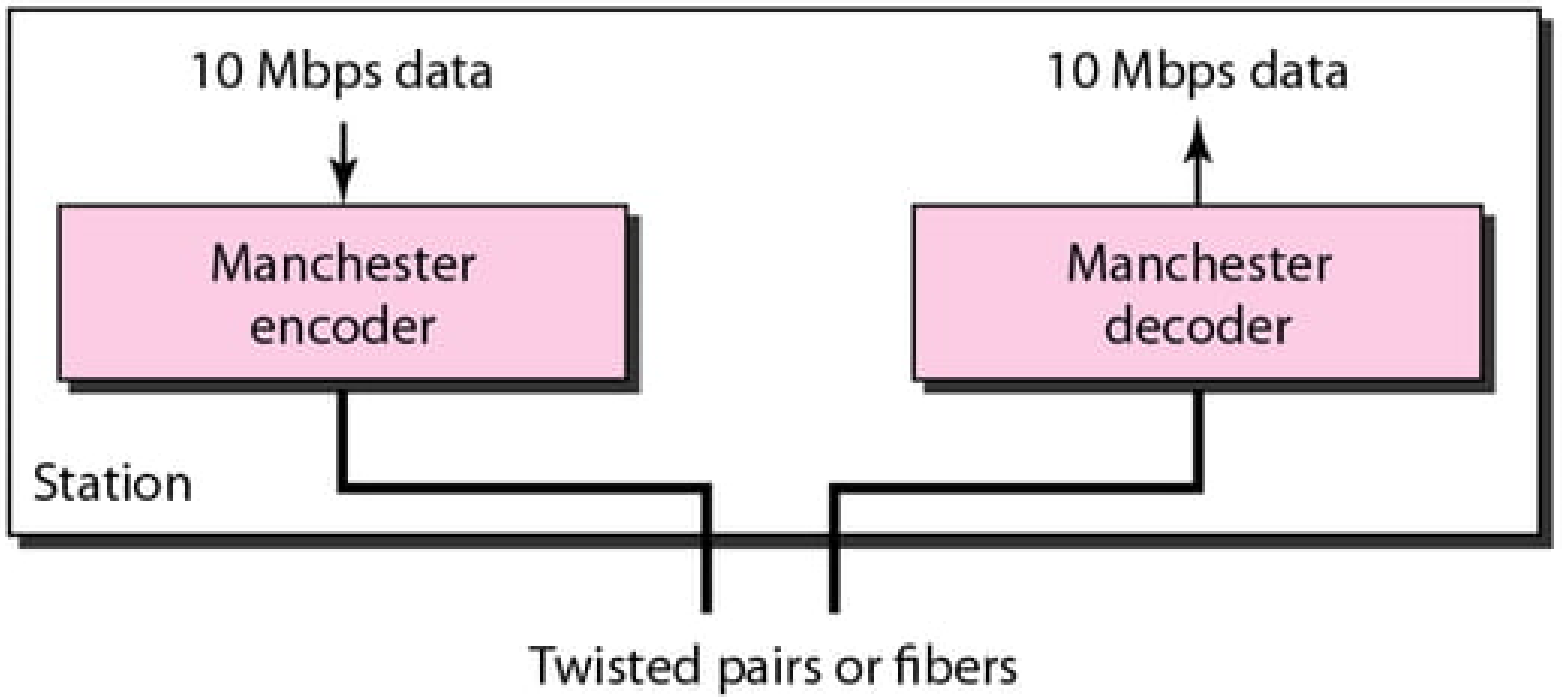


Figure 13.10 *10Base5 implementation*

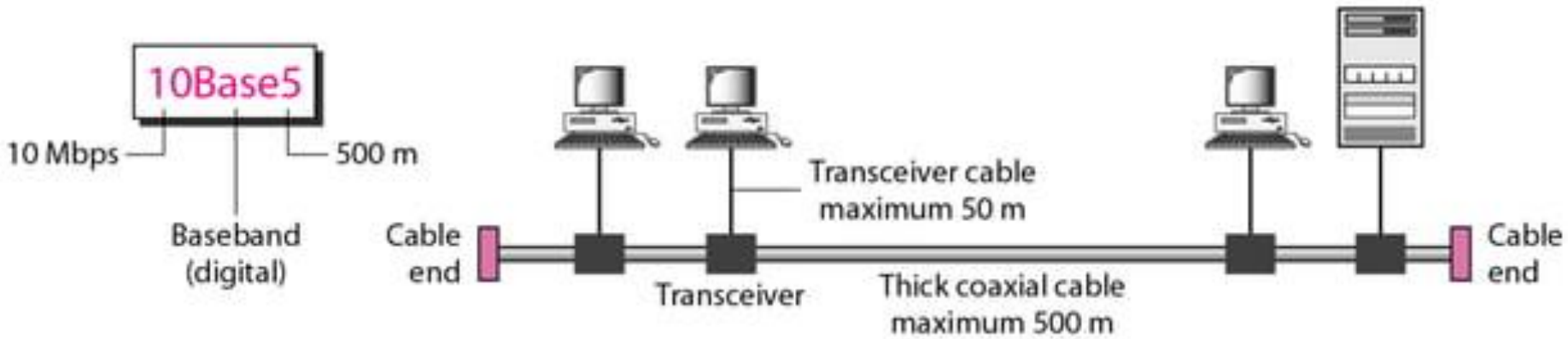


Figure 13.11 *10Base2 implementation*

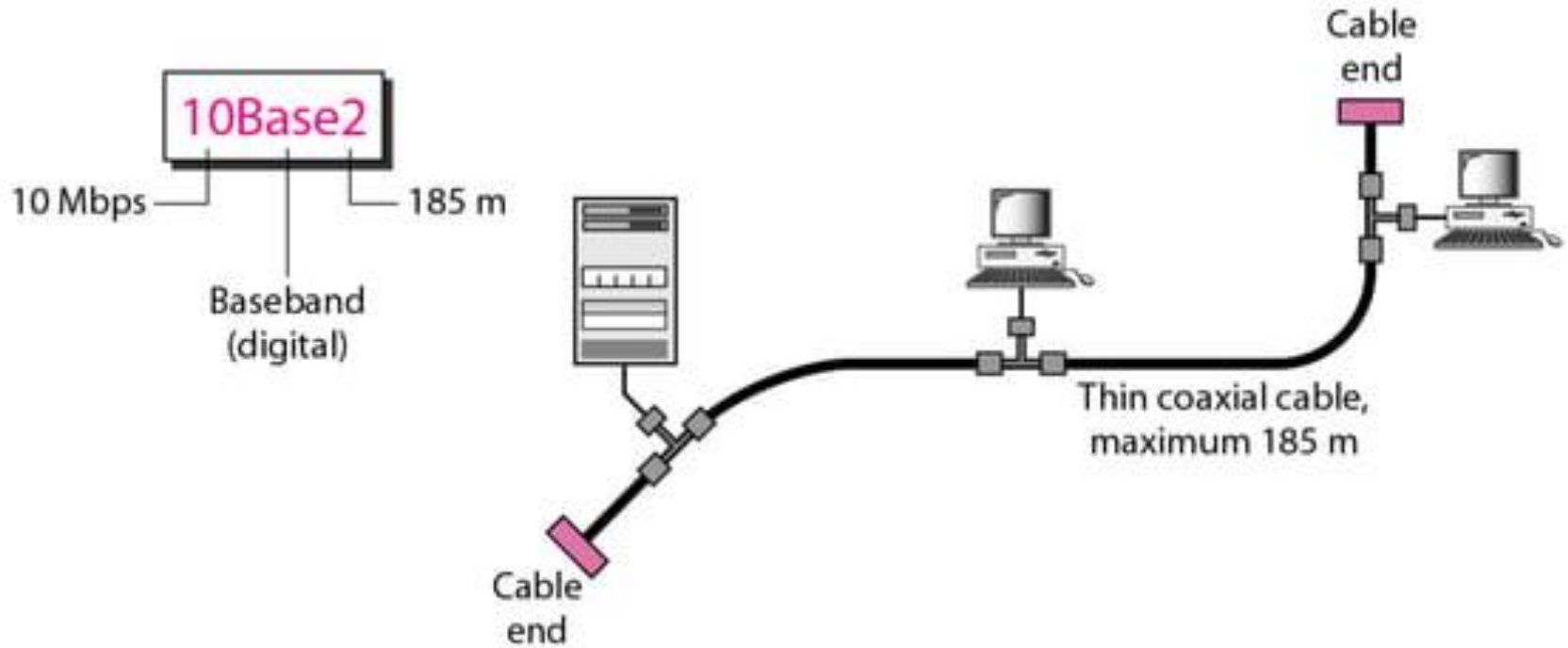


Figure 13.12 *10Base-T implementation*

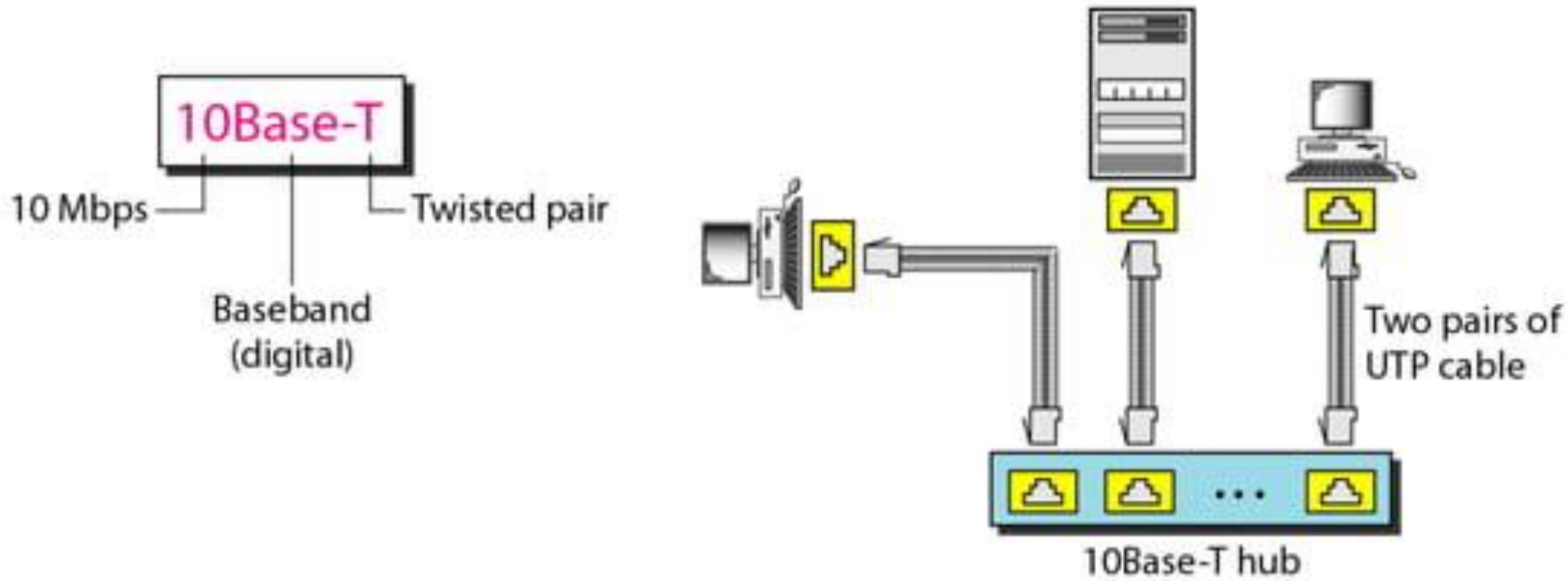


Figure 13.13 *10Base-F implementation*

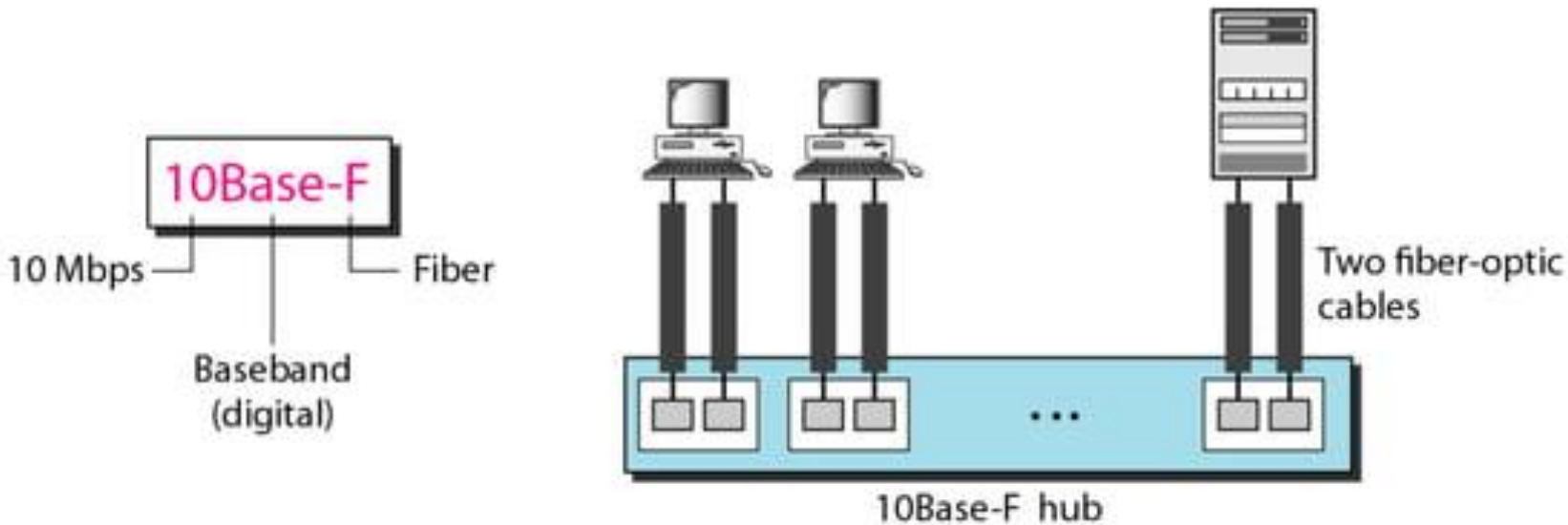


Table 13.1 ***Summary of Standard Ethernet implementations***

Characteristics	*10Base5*	*10Base2*	*10Base-T*	*10Base-F*
Media	Thick coaxial cable	Thin coaxial cable	2 UTP	2 Fiber
Maximum length	500 m	185 m	100 m	2000 m
Line encoding	Manchester	Manchester	Manchester	Manchester

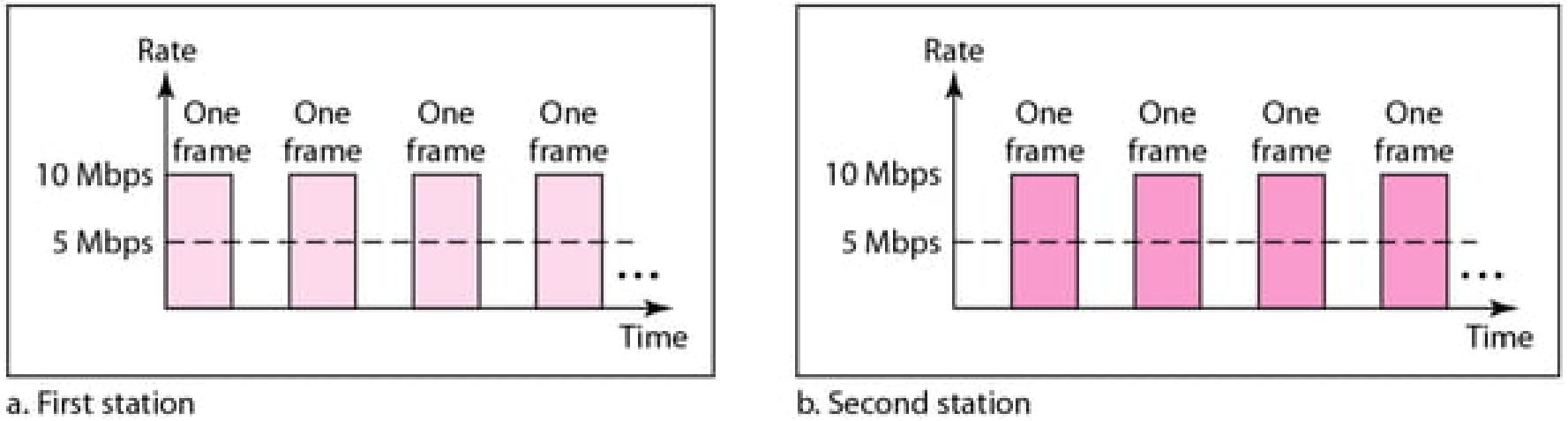
13-3 CHANGES IN THE STANDARD

The 10-Mbps Standard Ethernet has gone through several changes before moving to the higher data rates. These changes actually opened the road to the evolution of the Ethernet to become compatible with other high-data-rate LANs.

Topics discussed in this section:

Bridged Ethernet
Switched Ethernet
Full-Duplex Ethernet

Figure 13.14 *Sharing bandwidth*

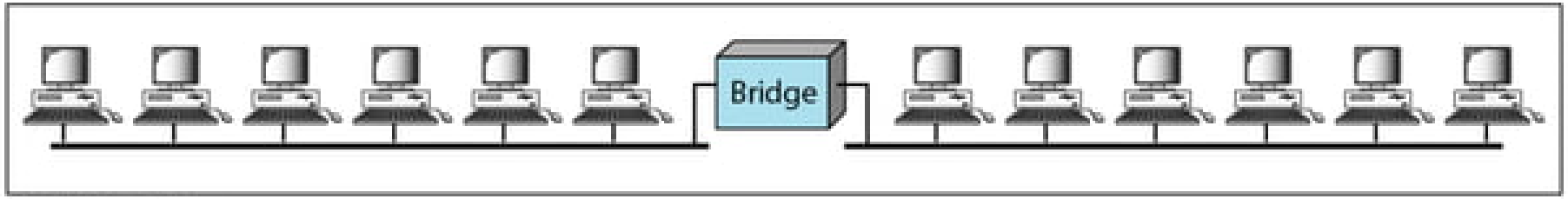


a. First station

b. Second station

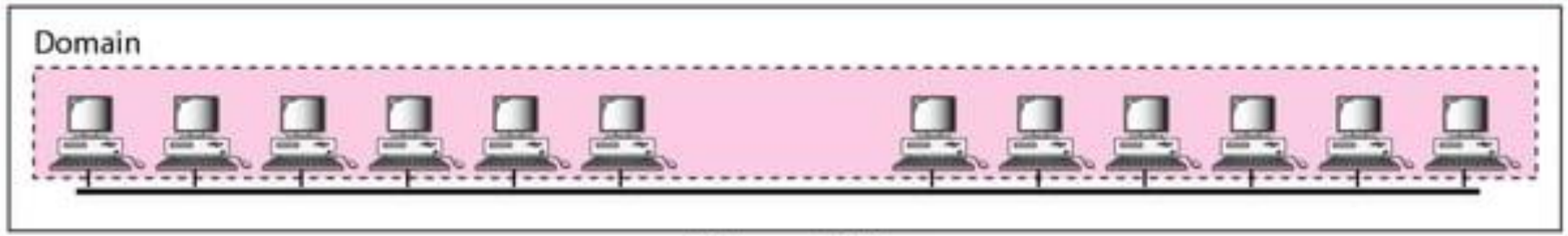
Figure 13.15 *A network with and without a bridge*

a. Without bridging

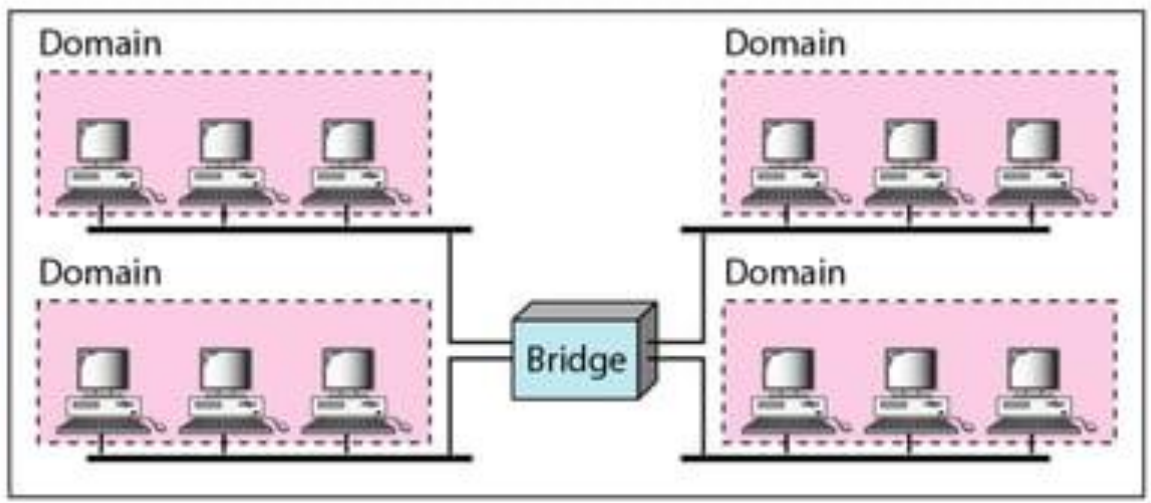


b. With bridging

Figure 13.16 *Collision domains in an unbridged network and a bridged network*



a. Without bridging



b. With bridging

Figure 13.17 *Switched Ethernet*

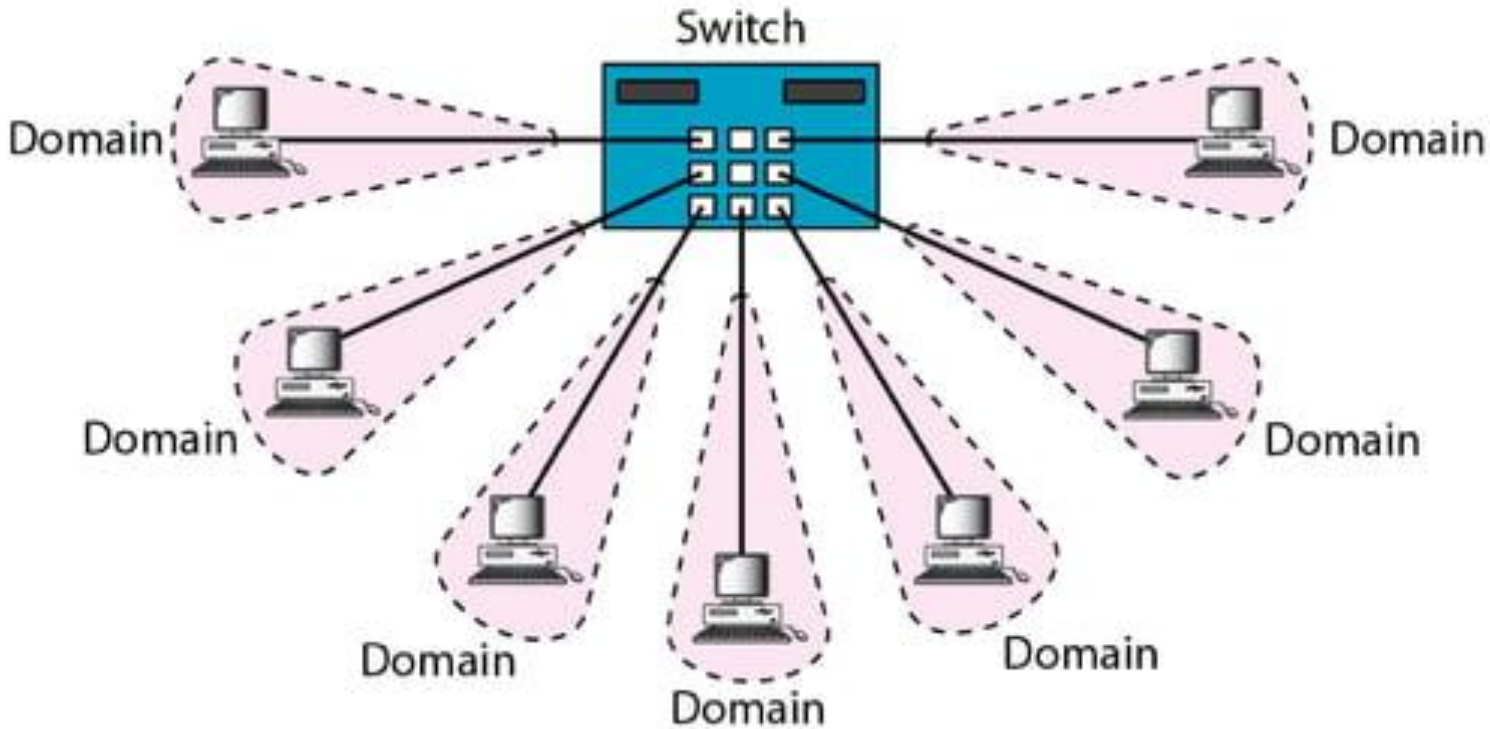
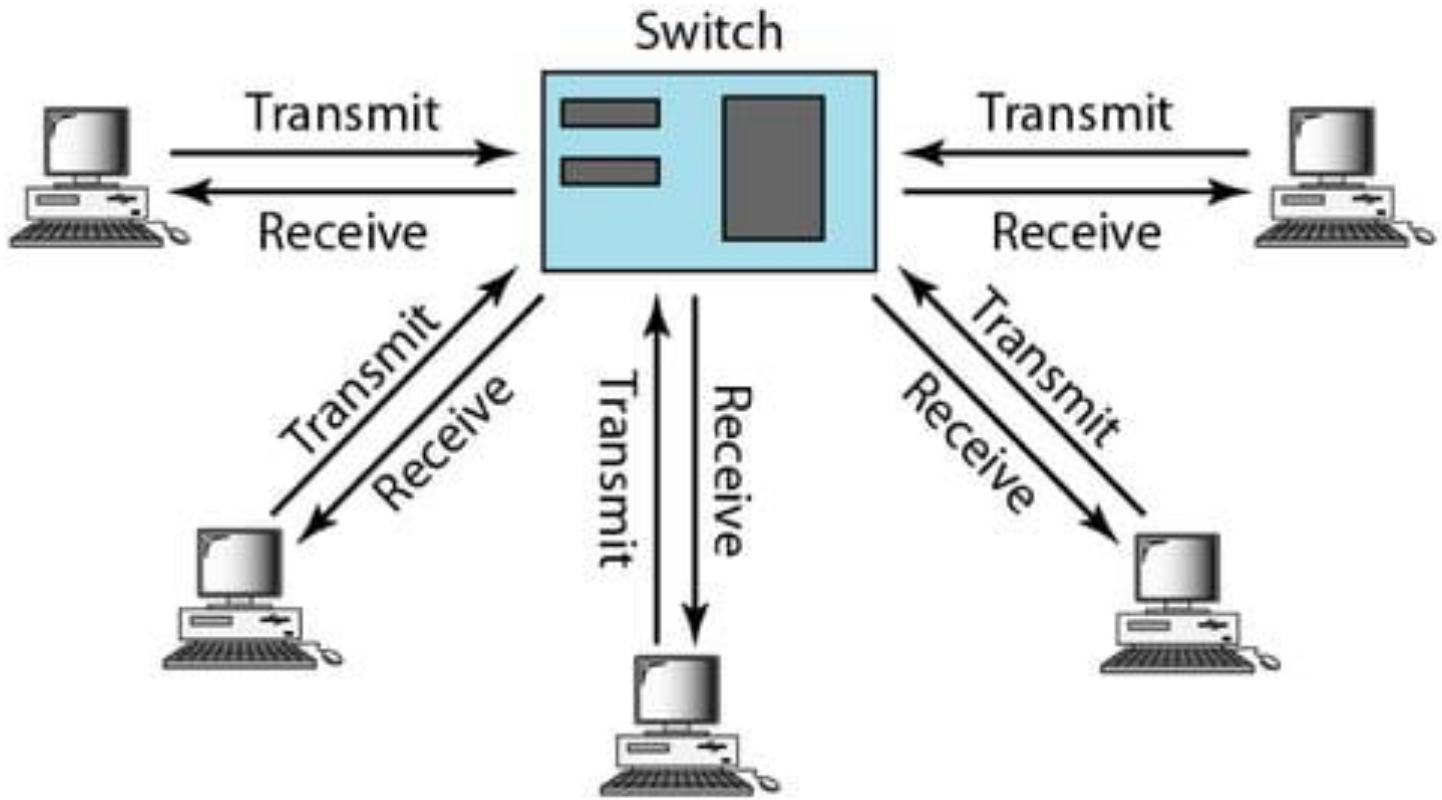


Figure 13.18 *Full-duplex switched Ethernet*



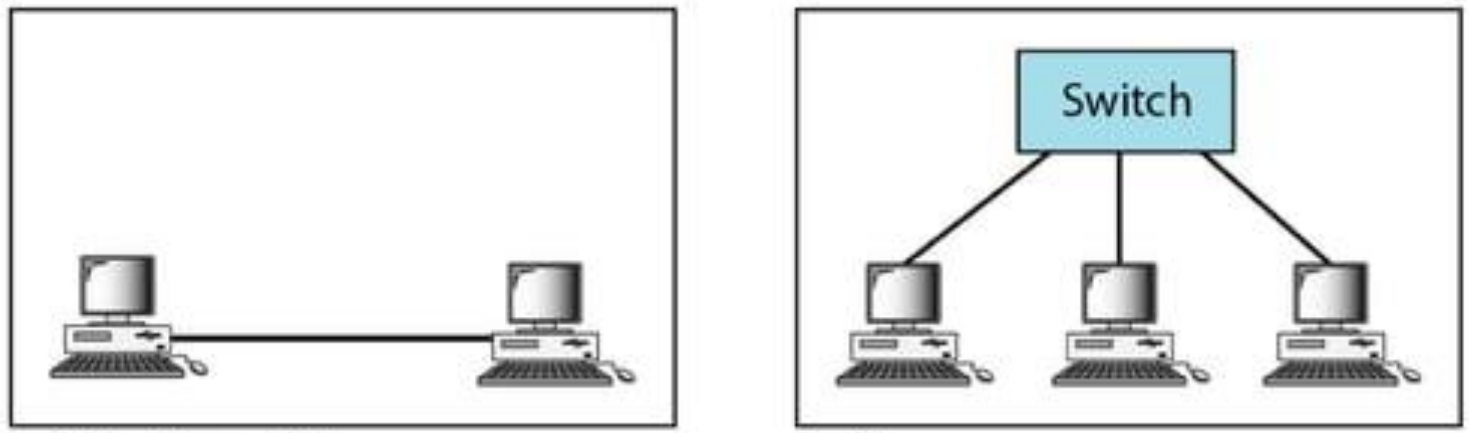
13-4 FAST ETHERNET

Fast Ethernet was designed to compete with LAN protocols such as FDDI or Fiber Channel. IEEE created Fast Ethernet under the name 802.3u. Fast Ethernet is backward-compatible with Standard Ethernet, but it can transmit data 10 times faster at a rate of 100 Mbps.

Topics discussed in this section:

MAC Sublayer
Physical Layer

Figure 13.19 *Fast Ethernet topology*



a. Point-to-point

b. Star

Figure 13.20 *Fast Ethernet implementations*

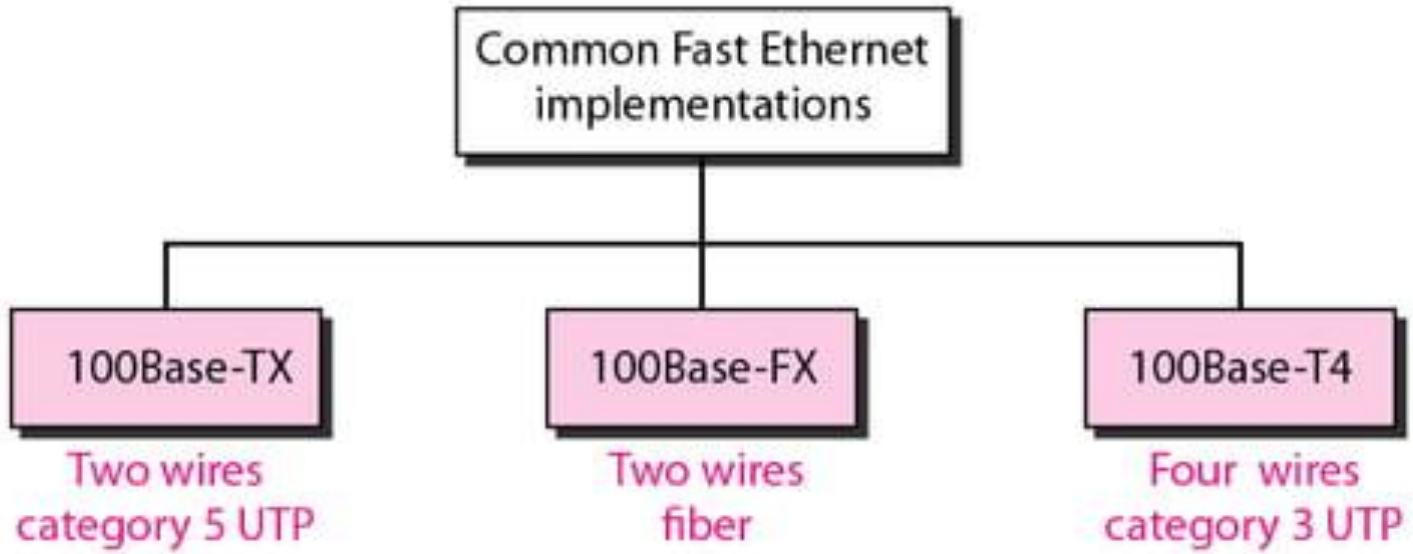


Figure 13.21 *Encoding for Fast Ethernet implementation*

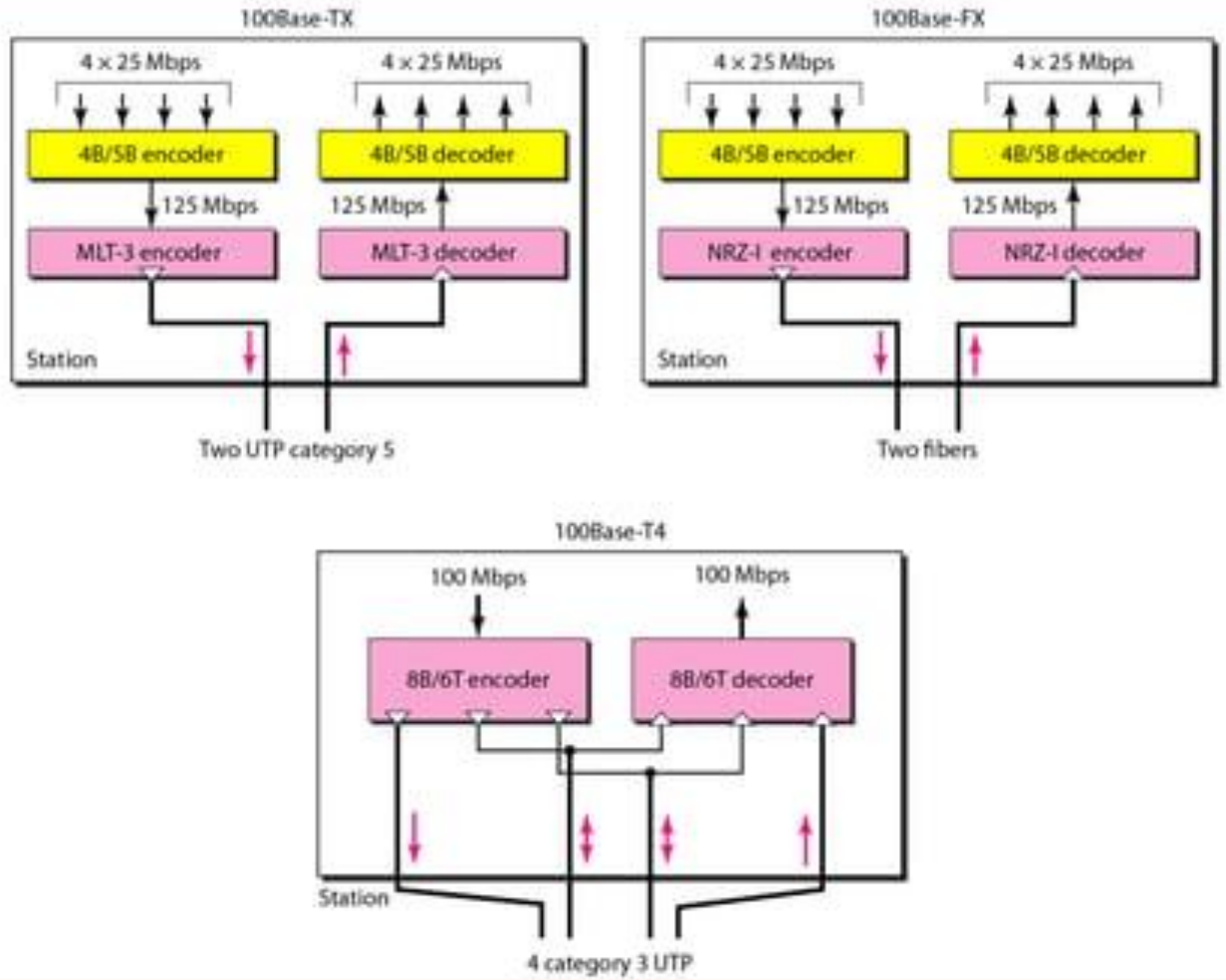


Table 13.2 ***Summary of Fast Ethernet implementations***

Characteristics	*100Base-TX*	*100Base-FX*	*100Base-T4*
Media	Cat 5 UTP or STP	Fiber	Cat 4 UTP
Number of wires	2	2	4
Maximum length	100 m	100 m	100 m
Block encoding	4B/5B	4B/5B	
Line encoding	MLT-3	NRZ-I	8B/6T

13-5 GIGABIT ETHERNET

The need for an even higher data rate resulted in the design of the Gigabit Ethernet protocol (1000 Mbps). The IEEE committee calls the standard 802.3z.

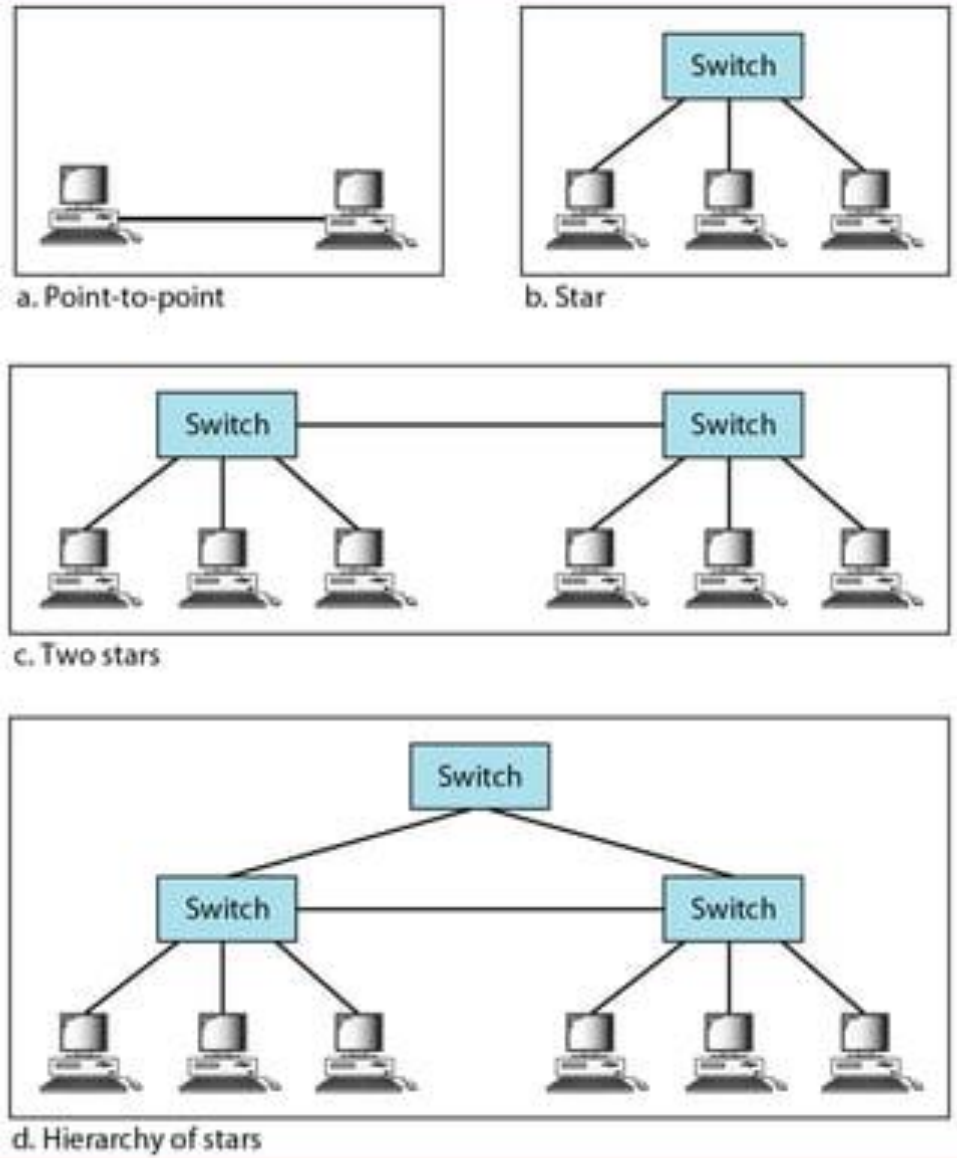
Topics discussed in this section:

MAC Sublayer
Physical Layer
Ten-Gigabit Ethernet

Note

In the full-duplex mode of Gigabit Ethernet, there is no collision; the maximum length of the cable is determined by the signal attenuation in the cable.

Figure 13.22 *Topologies of Gigabit Ethernet*



a. Point-to-point

b. Star

c. Two stars

d. Hierarchy of stars

Figure 13.23 *Gigabit Ethernet implementations*

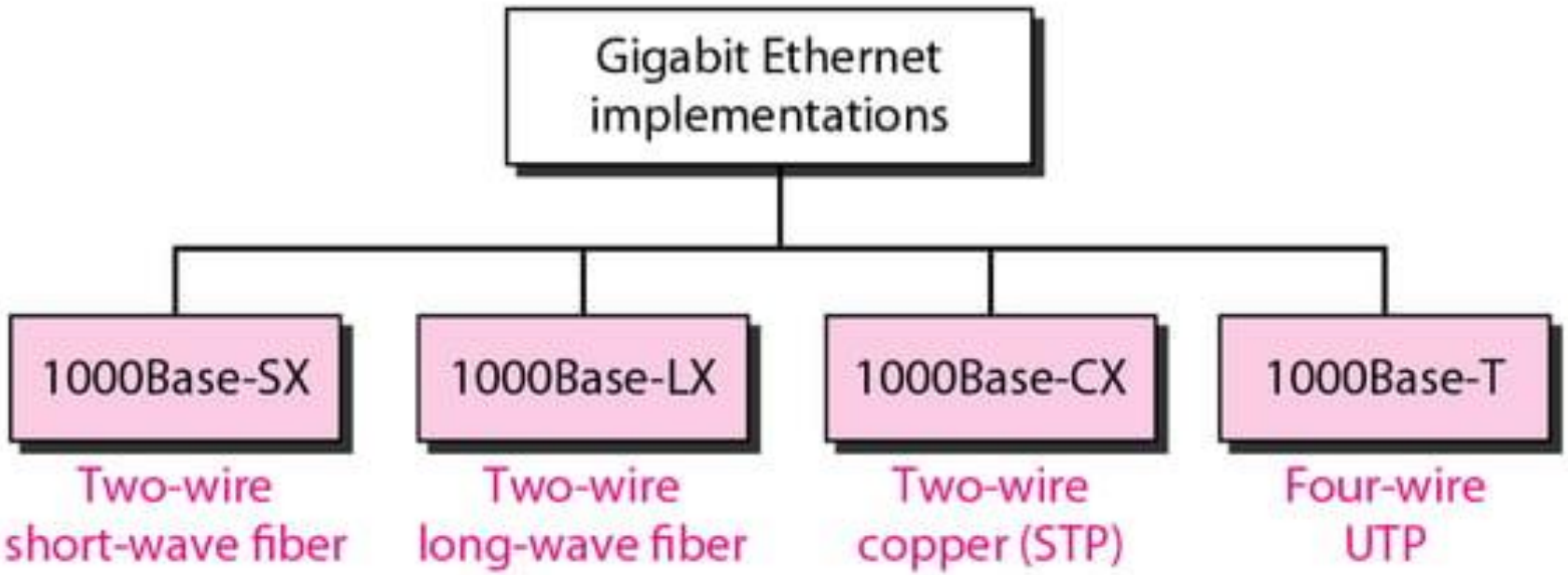


Figure 13.24 *Encoding in Gigabit Ethernet implementations*

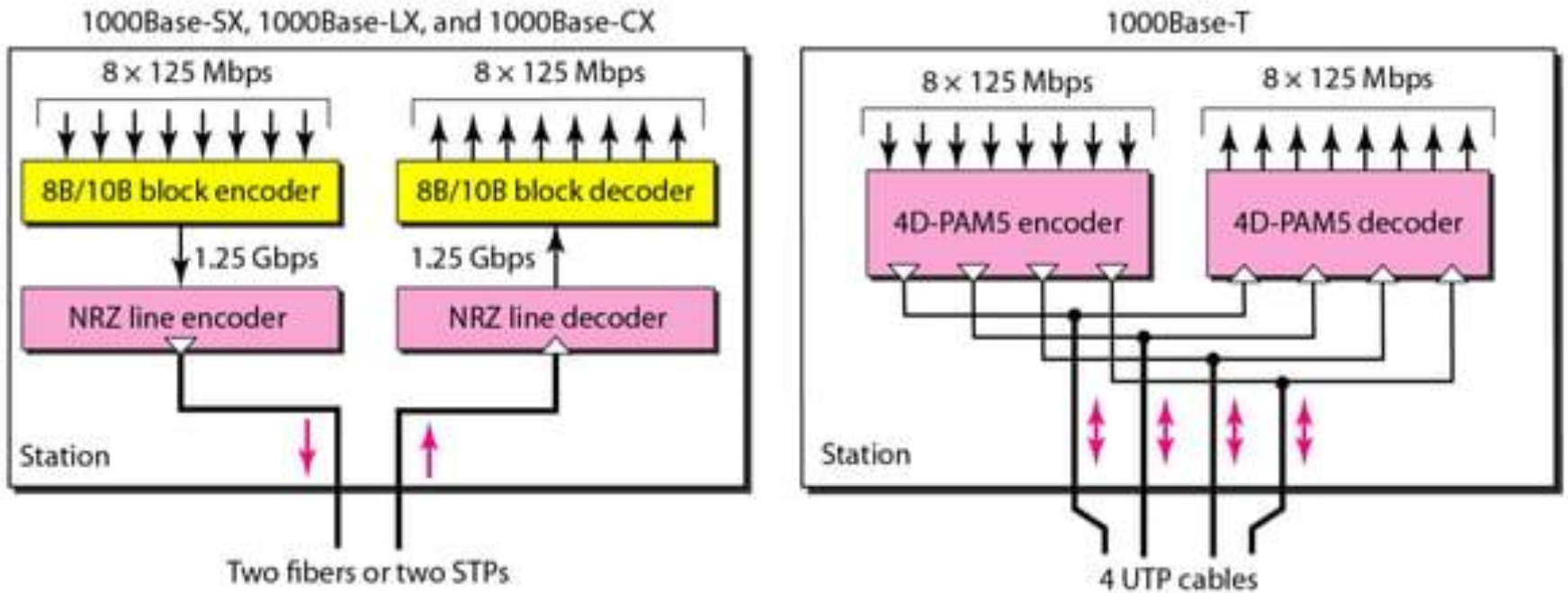


Table 13.3 *Summary of Gigabit Ethernet implementations*

Characteristics	*1000Base-SX*	*1000Base-LX*	*1000Base-CX*	*1000Base-T*
Media	Fiber short-wave	Fiber long-wave	STP	Cat 5 UTP
Number of wires	2	2	2	4
Maximum length	550 m	5000 m	25 m	100 m
Block encoding	8B/10B	8B/10B	8B/10B	
Line encoding	NRZ	NRZ	NRZ	4D-PAM5

Table 13.4 ***Summary of Ten-Gigabit Ethernet implementations***

Characteristics	*10GBase-S*	*10GBase-L*	*10GBase-E*
Media	Short-wave 850-nm multimode	Long-wave 1310-nm single mode	Extended 1550-mm single mode
Maximum length	300 m	10 km	40 km