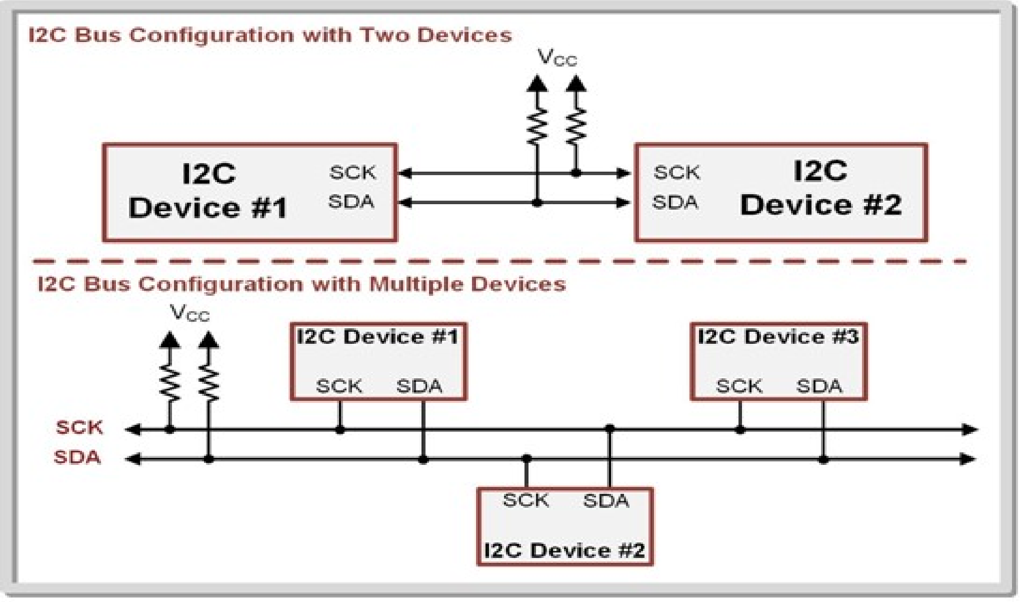
Module-6: Embedded Networking Protocols

The I2C Protocol

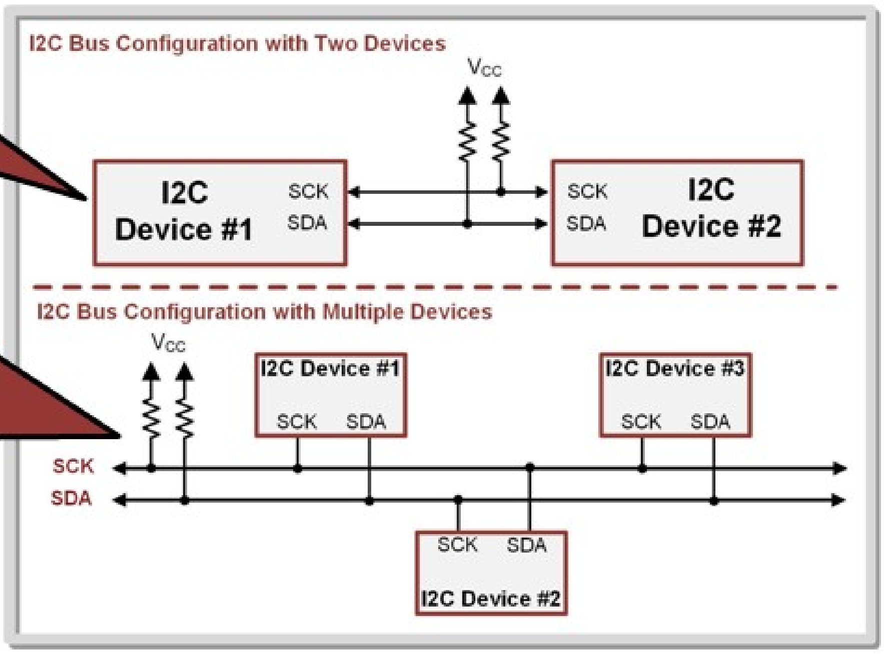
- The **Inter-Integrated Circuit (I2C)** standard is a serial interface implement with a two-wire link that can support multiple masters and multiple slaves.



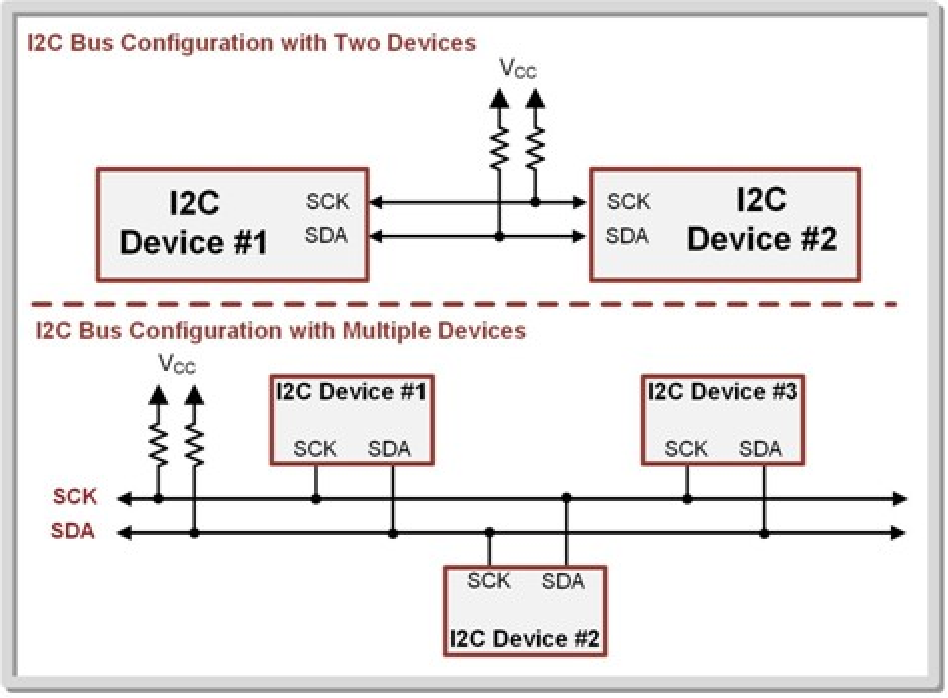
- The **Inter-Integrated Circuit (I2C)** standard is a serial interface implement with a two-wire link that can support multiple masters and multiple slaves.

Providing a clock allows higher data rates over UART.

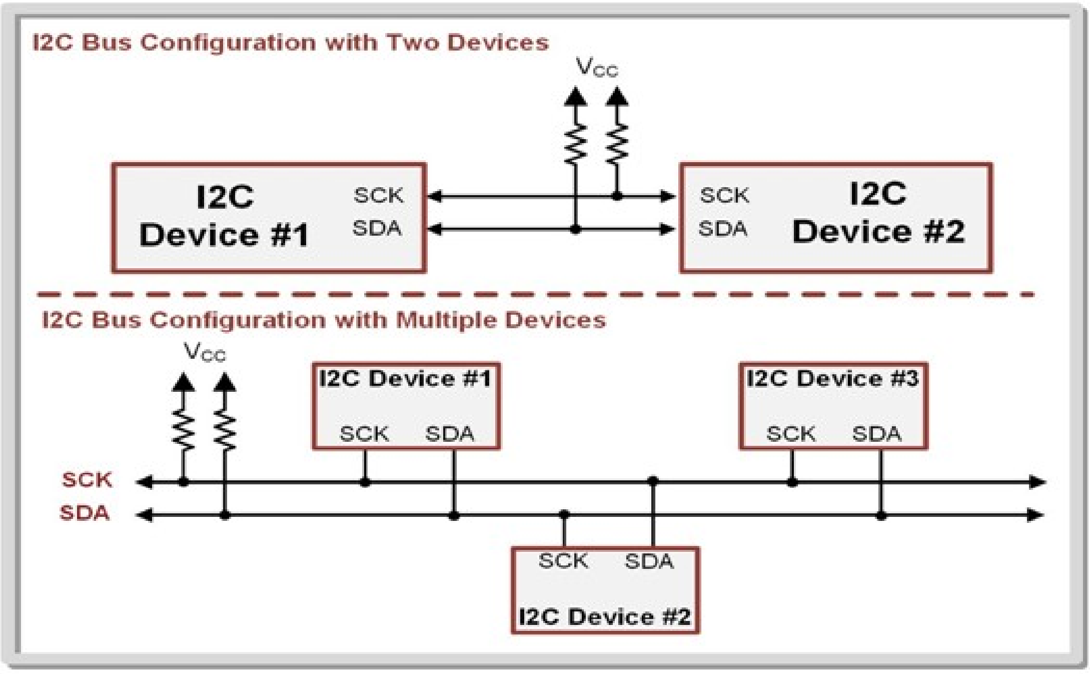
The two-wire interface allows devices to be added without increasing the number of lines like on SPI.



- An I2C bus contains a clock line (SCK) and data line (SDA).



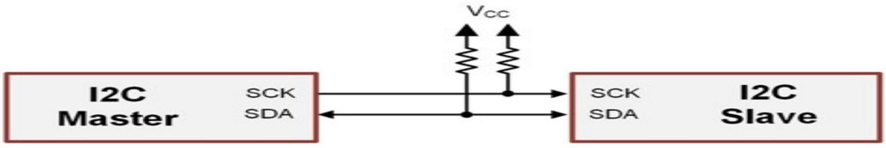
- An I2C link is always **half-duplex**, meaning that all devices share the data line with only one device transmitting at any given time.



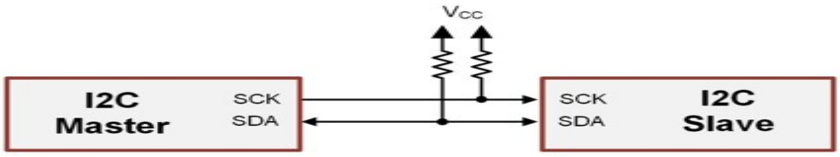
The I2C Protocol

- **Master** – the device that initiates communication and controls the clock.

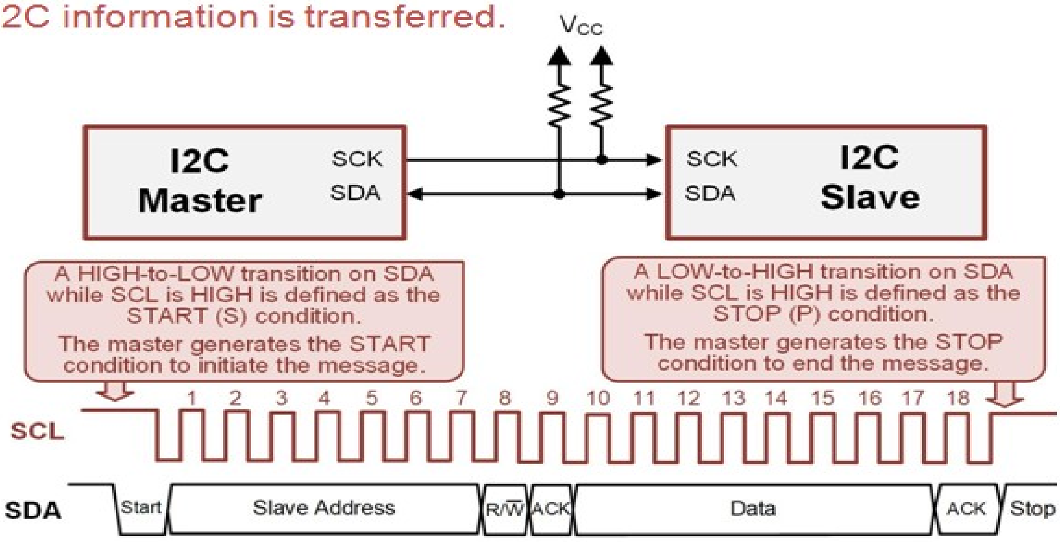
- Multiple masters are also supported on an I2C bus.



- **Slave** – a device on the bus that is read or written to, but does not initiate transmission or provide a clock.
- **Slave address** – a unique and <u>predetermined</u> address for each slave on the bus.
- This address is used by the master to indicate which slave it wants to communicate with.

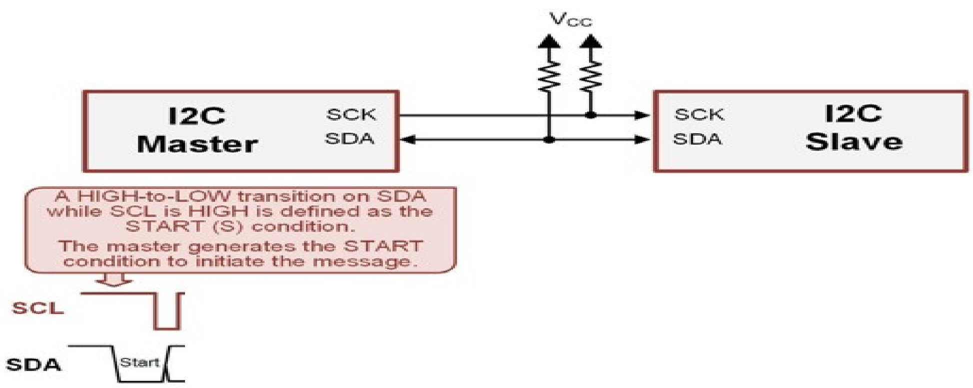


- **Idle** – when both SDA and SCL are held high by the pull-up resistors and no I2C device is attempting to communicate.
- **Busy** – when devices are driving the bus.
- **Messages** – how I2C information is transferred.



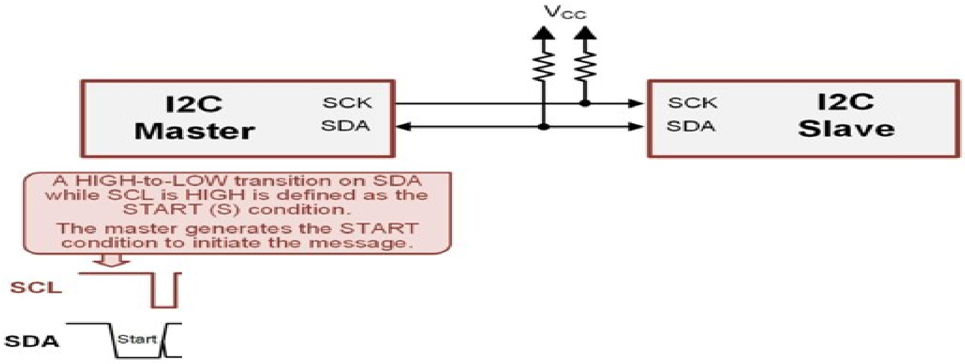
- A master initiates a new message by generating a START (S) condition by pulling SDA LOW while SCL is still HIGH.

- As soon as the START condition is generated, the SCL will be pulled LOW and start pulsing to provide the clock for the message.

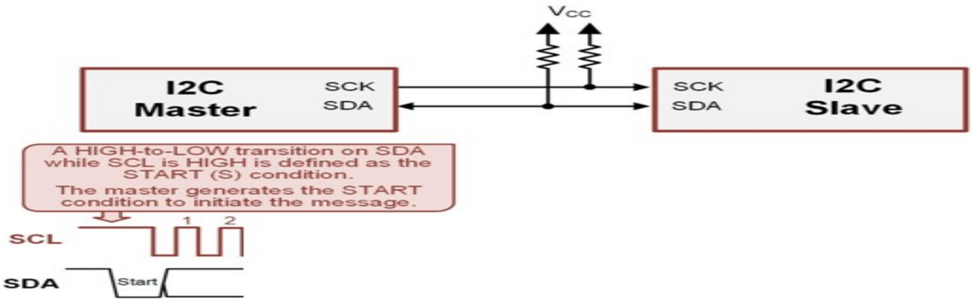


The I2C Protocol

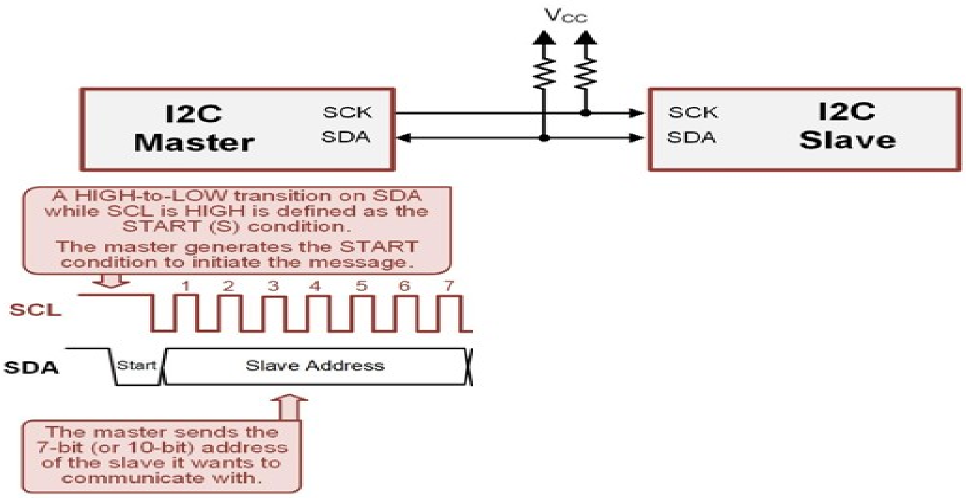
- The master is responsible for pulsing the clock.



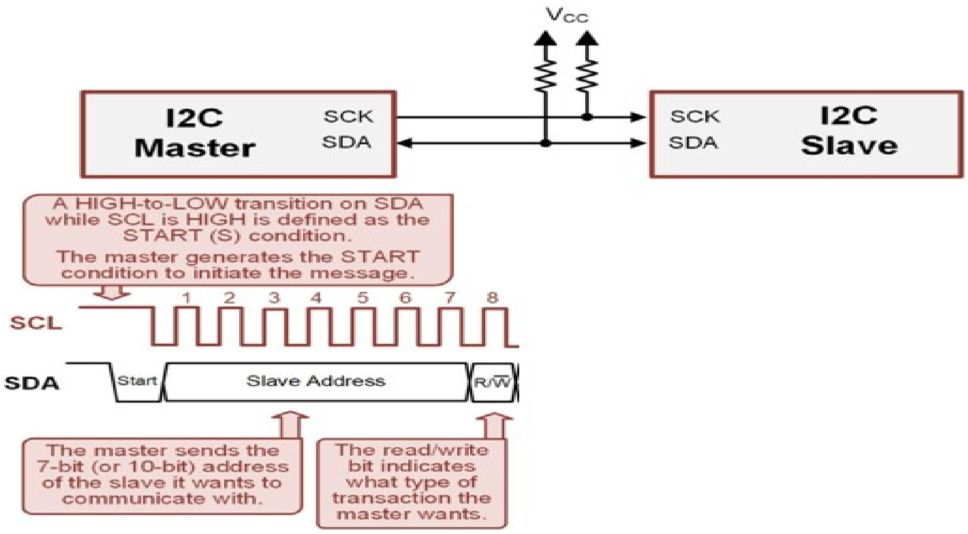
- Each clock pulse within the I2C message is numbered by **periods**.
- Both the master and the slaves count the number of periods that have occurred since the message started in order to know when certain frames and signals should be present.



- After the master generates the START condition, it first sends the slave address that it wishes to communicate with.
- I2C slave addresses can either be 7-bit (default) or 10-bit.

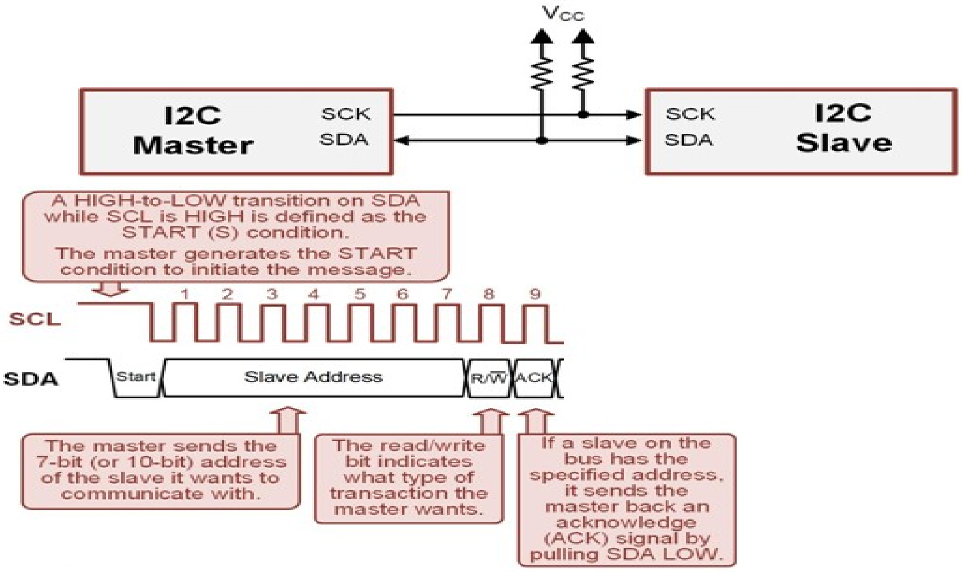


- The slave address is followed by the read/write signal indicating which type of transaction is being requested in the message.
- The START condition, slave address, and read/write signal constitute periods 1 ⟶ 8.

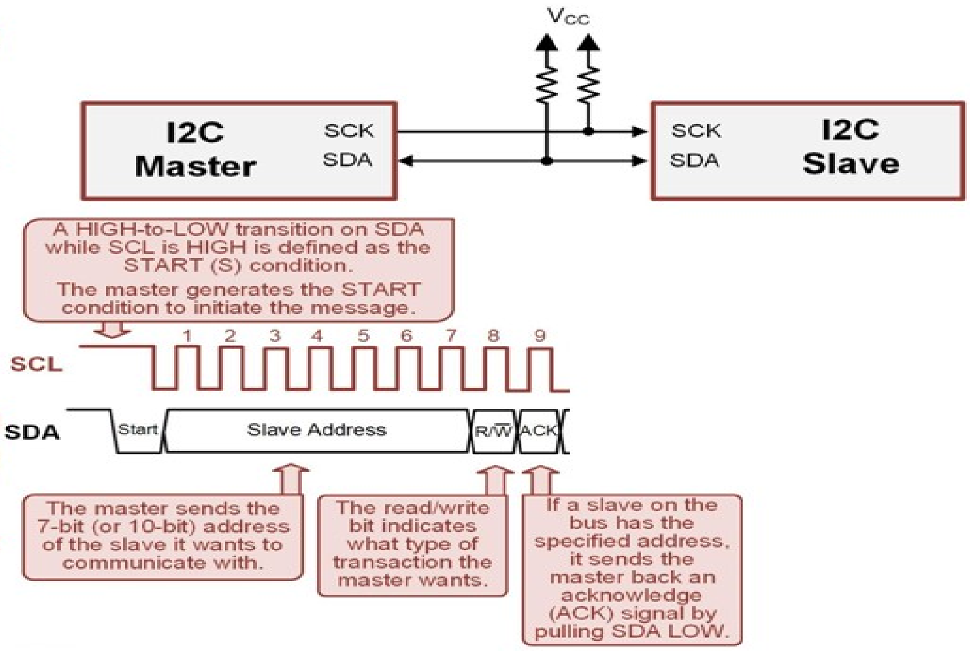


The I2C Protocol

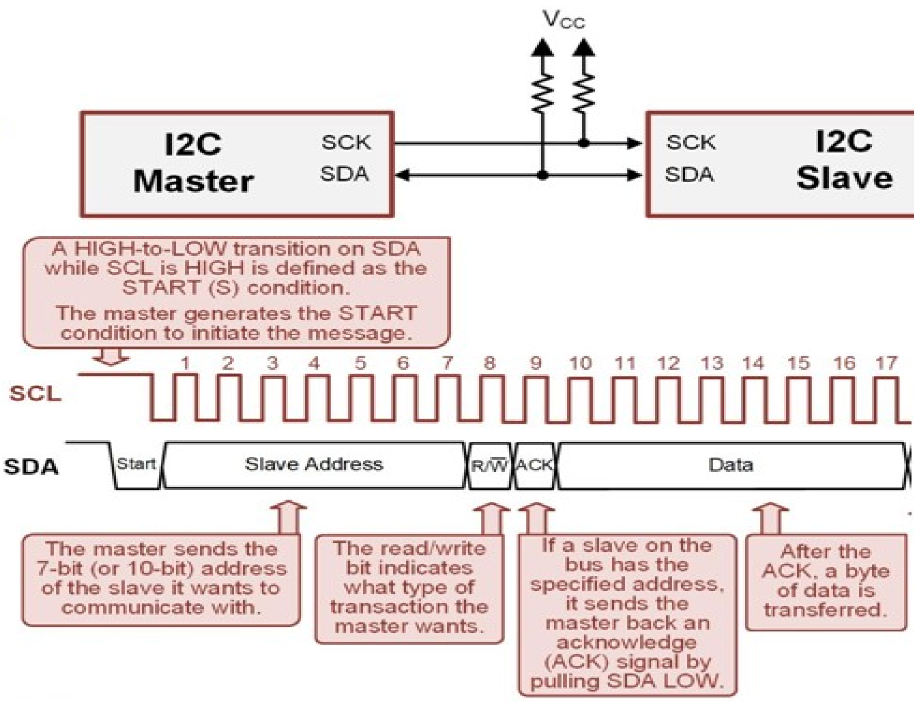
- Period 9 of the message is reserved for the slave acknowledge (ACK) or no-acknowledge (NACK) signal.
- After the slave address and read/write signal are sent by the master, each slave on the bus checks whether it is being addressed.



- If a slave exists with the specified slave address, it will send an **ACK** signal back to the master by pulling SDA LOW.
- If no device exists with the specified slave address, no device will pull down SDA. This will result in period 9 remaining HIGH and will be interpreted as a **NACK**.

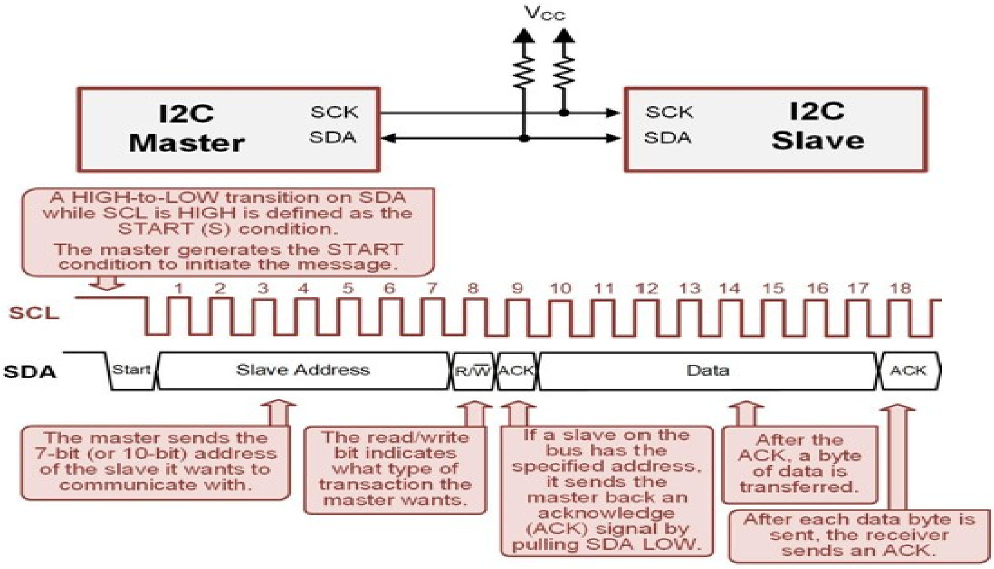


- If the master sees the ACK signal, it knows a slave exists with the specified address and proceeds with the message.

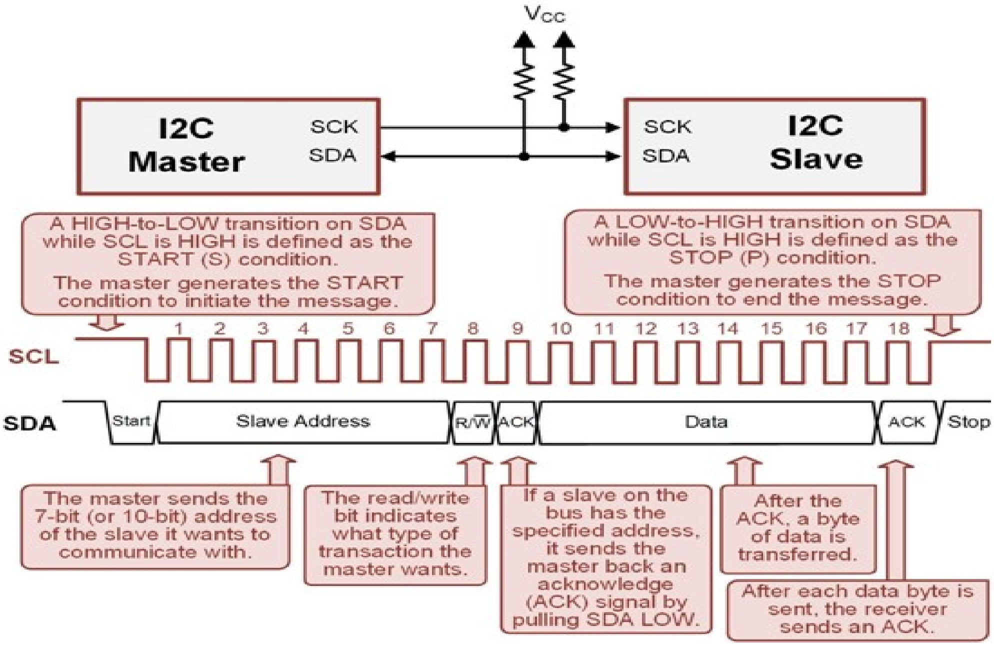


The I2C Protocol

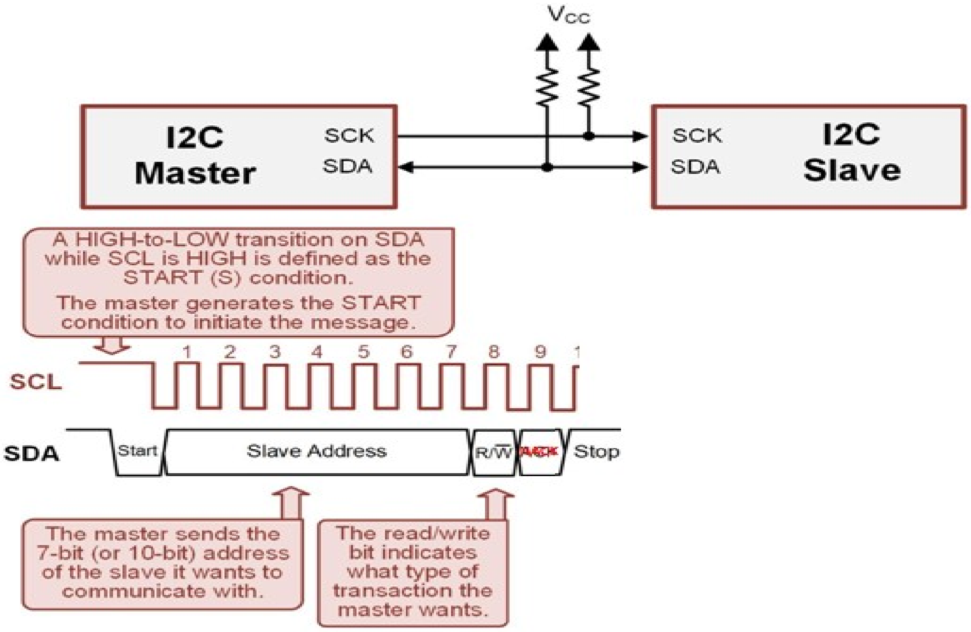
- After each byte is sent, the receiving device sends an ACK signal indicating that it successfully received the data.



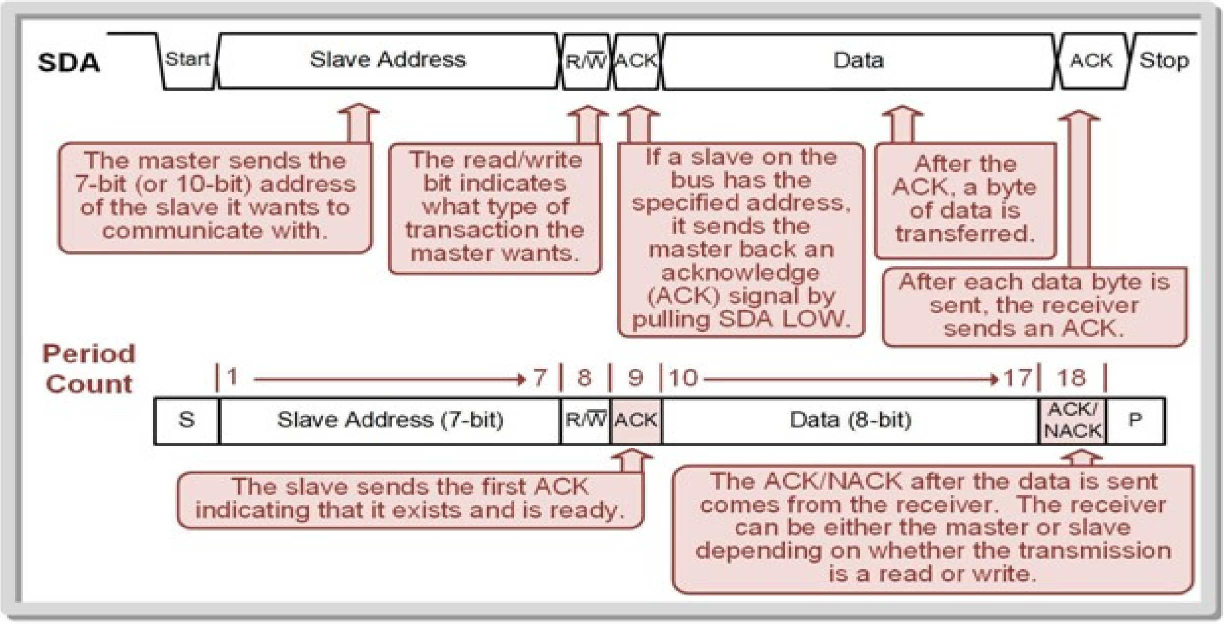
- A STOP condition occurs when there is a LOW-to-HIGH transition on SDA while SCL is HIGH.
- Once, SDA goes HIGH, SCL also remains HIGH indicating that the bus is idle again.



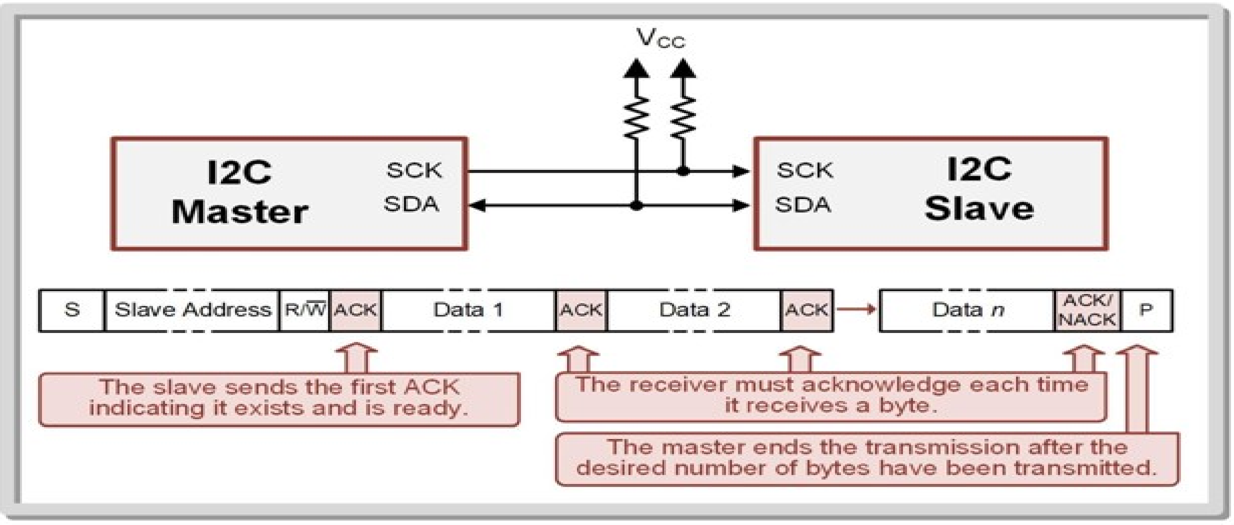
- A NACK in period 9 tells the master that no slave exists with the specified address.
- The master then generates a STOP condition and ends the message.



The I2C Protocol



- When the master is **writing** to a slave, the master sends the 8-bits of data and the slave produces the ACK/NACK signal.
- When the master is **reading** from a slave, the slave sends the 8-bits of data and the master produces the ACK/NACK signal.
- After the data has been sent and acknowledged, the master can end the message by generating the STOP condition anytime.
- The Master can send multiple data bytes in a single message.



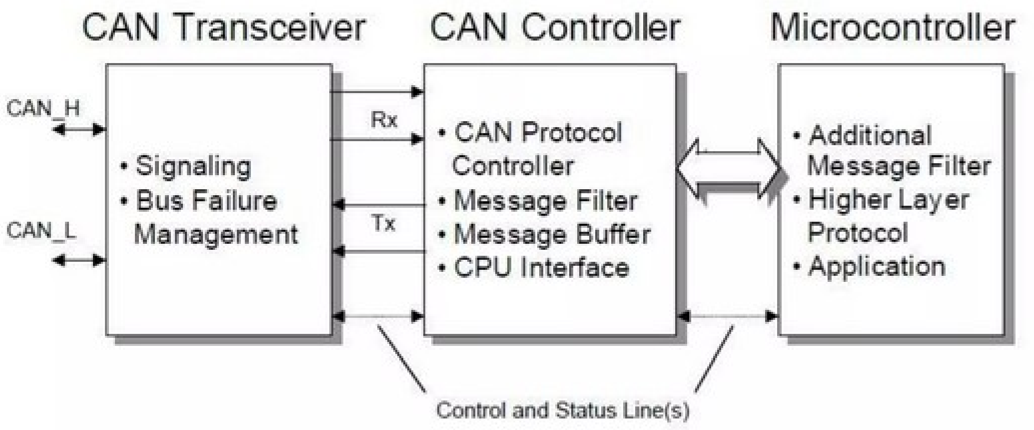
Controller Area Network

INTRODUCTION

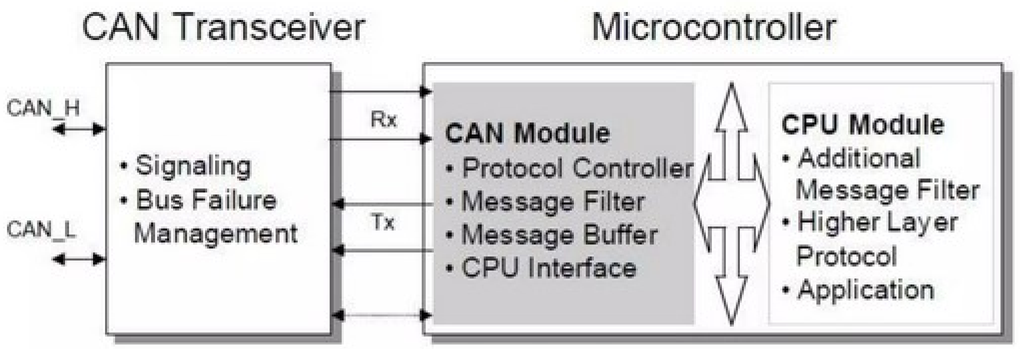
- CAN bus is a vehicle bus standard designed to allow microcontroller and devices to communicate with each other in vehicle.
- It is a message based protocol.
- It is present in both physical and data link layer of OSI layers.
- It is a multi-master serial bus .

CAN implementation

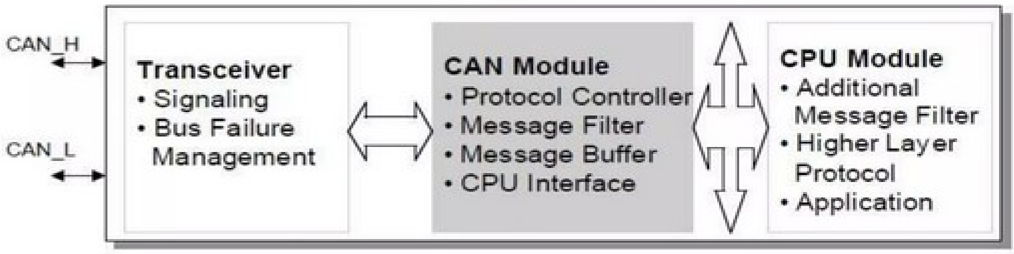
- Stand alone CAN Controller.
- Integrated CAN Controller.
- Single-Chip CAN Controller.



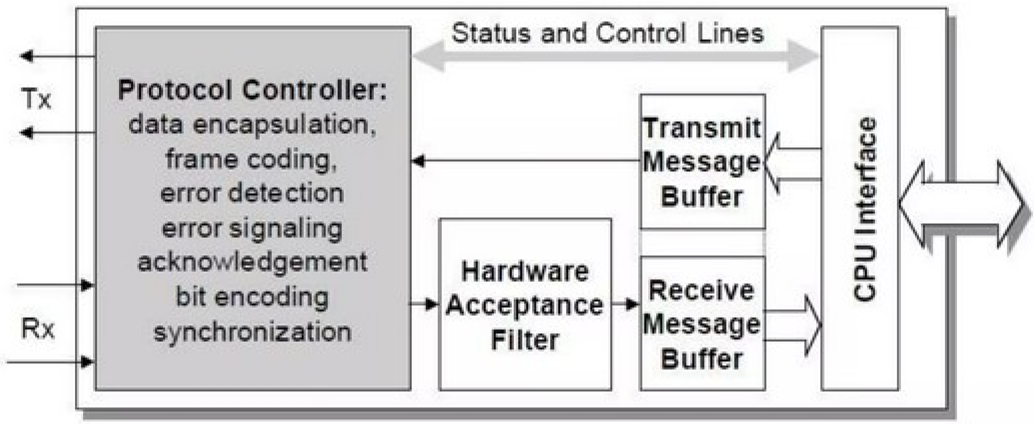
Integrated CAN Controller



Single-Chip CAN Controller



CAN Controller Architecture



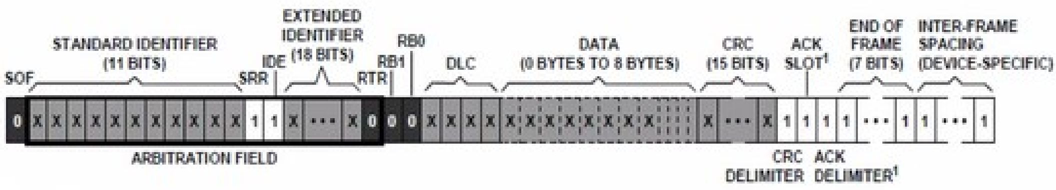
Message Frame

- Start of Frame.
- Arbitration Field.
- Control Field.
- Data Field.
- CRC Field.
- Acknowledgement bits.
- End of Frame.

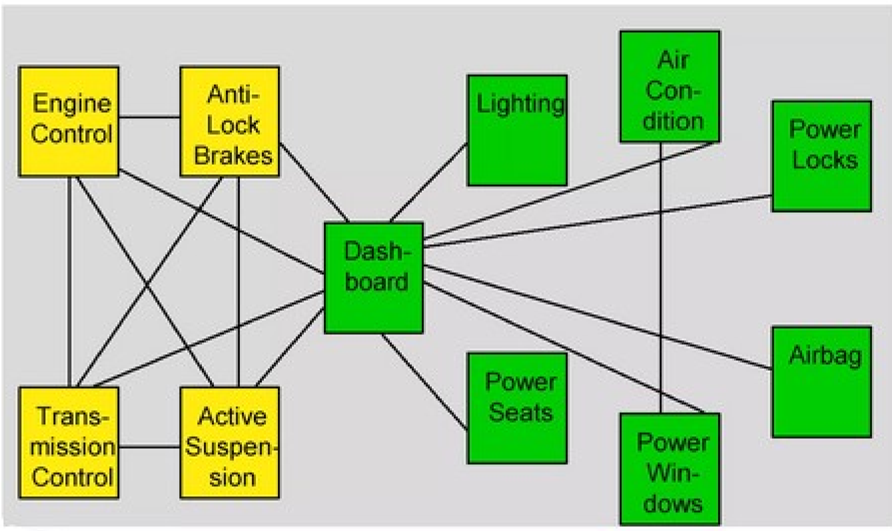
Message Frame Structure

- Standard Data Frame

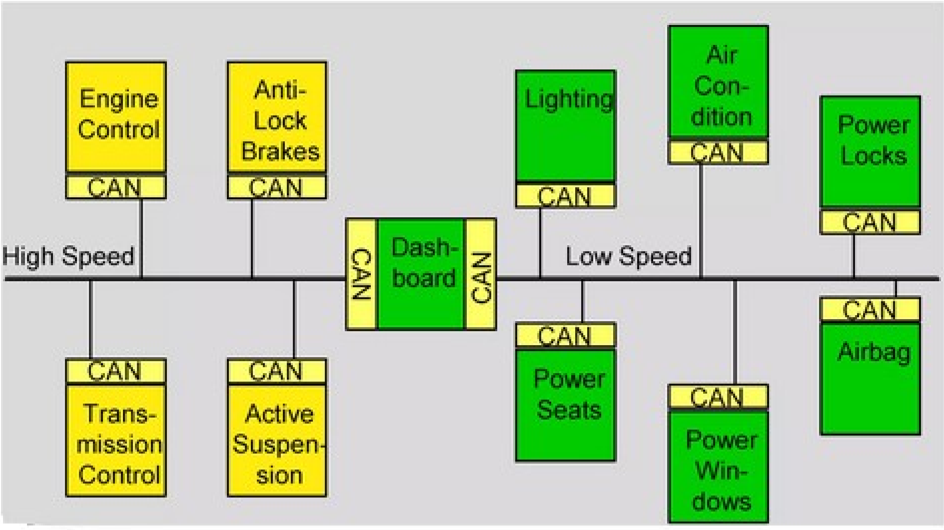
- Extended Data Frame



Before CAN



After CAN



Applications

- Automobiles
- Aerospace
- Maritime
- Industrial automachine
- Medical equipments

Advantages

- High throughput under light loads
- Local and global prioritization possible

Limitations

- Unfair access - node with a high priority can "hog" the network.
- Poor latency for low priority nodes.

Conclusion

- CAN is ideally suited in applications requiring a large number of short messages with high reliability in rugged operating environments. Because CAN is message based and not address based, it is especially well suited when data is needed by more than one location and system-wide data consistency is mandatory.

Bluetooth Introduction

Bluetooth Communications. 802.15

Bluetooth is considered as a secure short-range wireless network.

- A cable replacement technology
- 1 Mb/s symbol rate
- Range 10+ meters
- Single chip radio
  - at low power & low price ($5)

Why not use Wireless LANs?
- power
- cost

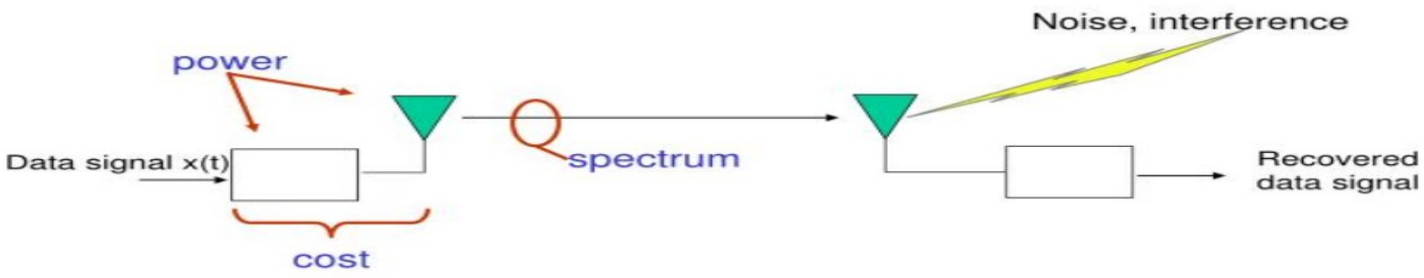
Bluetooth working group history

- February 1998: The **B**luetooth **S**pecial **I**nterest **G**roup promoter: **Ericsson, IBM, Intel, Nokia, Toshiba.**
- May 1998:
- July 1999: version **1.0A** is released.
- December 1999: version **1.0B** is released.
- March 2001: version **1.1** is released
- ***Where Did the Name Come From?***
- Herald Blatant "Bluetooth II "–King of Denmark 940-981 AC.
- Noted for unifying Denmark and Sweden.

Applications

User benefits
- **Multiple device access (phone, music)**
- **Cordless phone benefits**
- **Hands free operation**
- **Conference Table**
- **Cordless Computer**
- **Business Card Exchange**
- **Instant Postcard**
- **Computer Speakerphone**

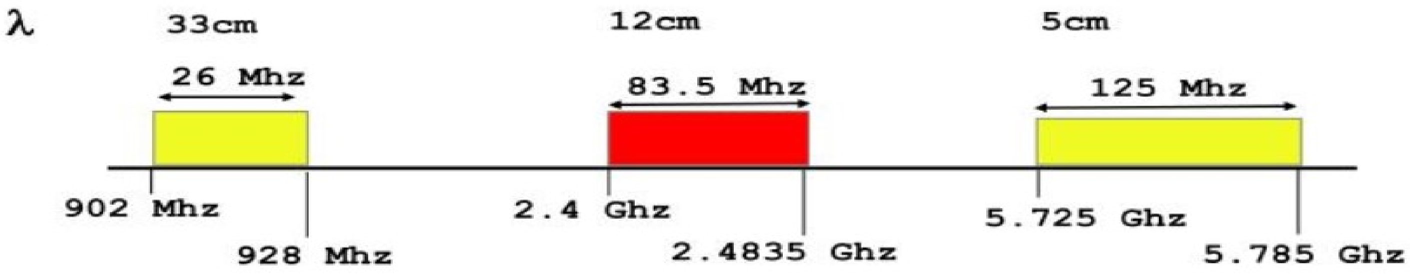
Design Considerations



Goal:

- high bandwidth
- conserve battery power
- cost < $10

Unlicensed Radio Spectrum



cordless phones,
baby monitors,
Wireless LANs

802.11 Cell.
802.15 Bluetooth,
Microwave oven

802.11a
HyperLan

Bluetooth Radio

- **Low Power**
  - Standby modes: **Sniff, Hold, Park.**
  - Low voltage RF

Power Class	Transmit Power	Nominal Range
1	100 mW (20 dBm)	100 m
2	2.5 mW (4 dBm)	20 m
3	1 m W (0 dBm)	10 m

Low Cost

Single chip radio (minimize external components)
Today's technology
Time division duplex

Bluetooth Architecture, Piconets and Scatternets

A Piconet is collection of devices connected to the Master.

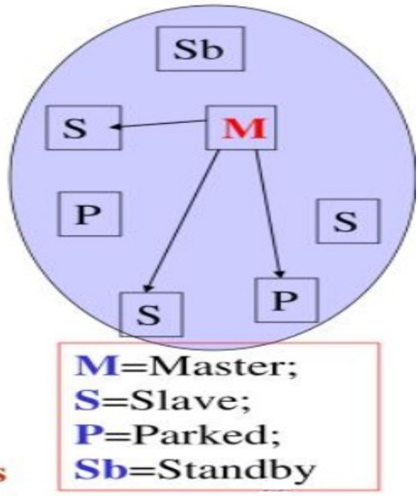
• One unit will act as a **Master** (the device, which initiates an exchange of data) and the others as **Slaves** (the device, which responds to the Master)

• Master sets the **clock**, **dwell time**, **hopping pattern**.

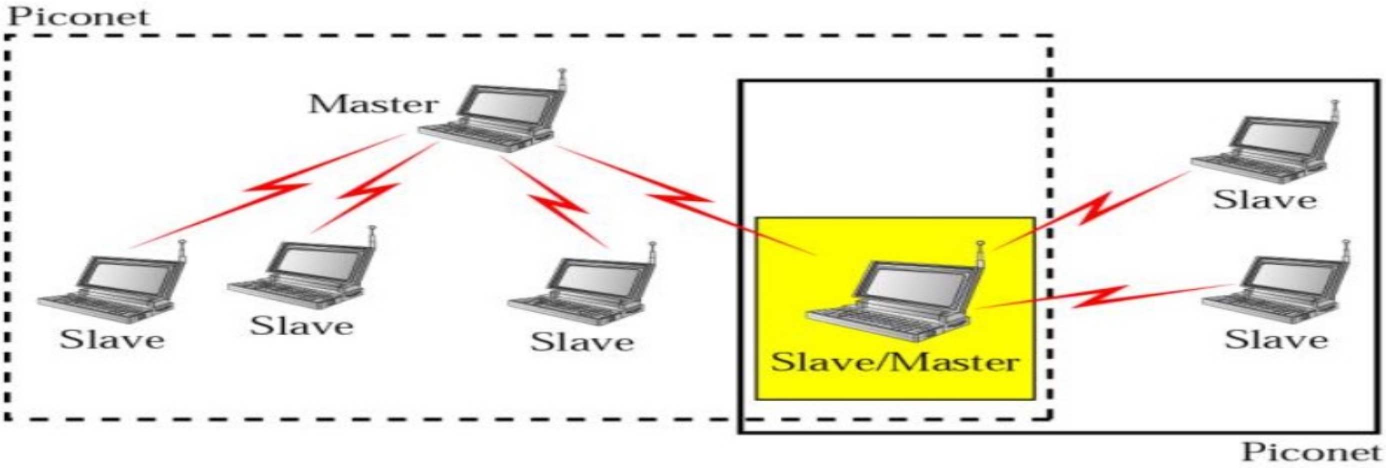
• **Each Piconet** has a unique **hopping pattern/ID**

• Each master can connect to 7 (specification limits) simultaneous or 255 inactive (parked) slaves per **Piconet**

A Scatternet is collection of the Piconets connected in an Ad Hoc fashion.

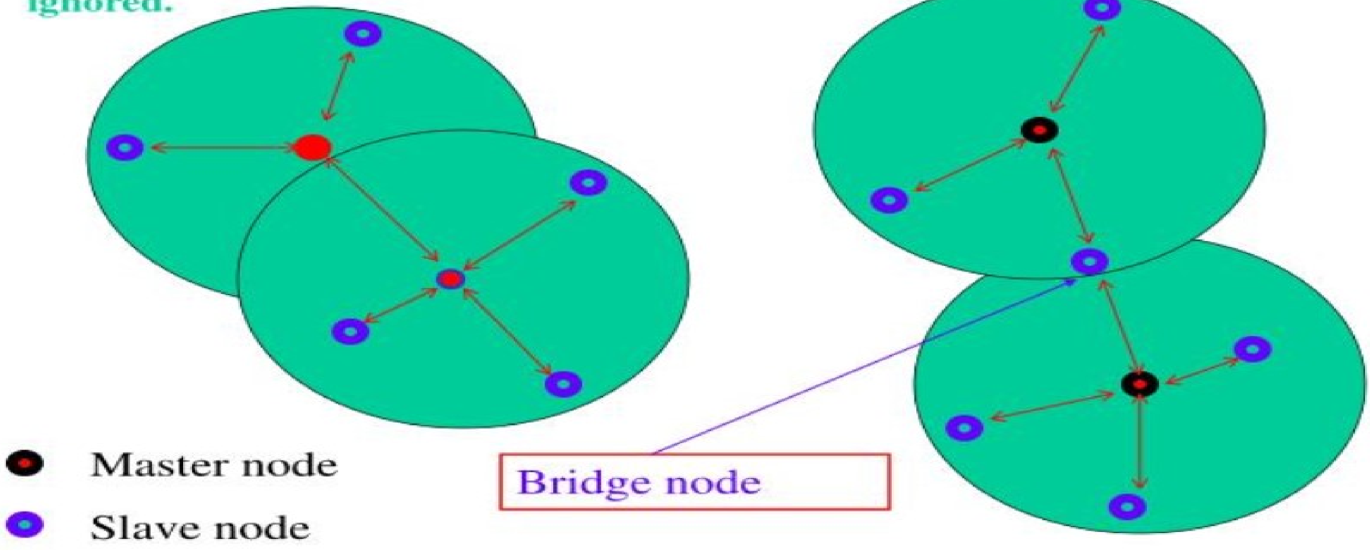


Scatternet



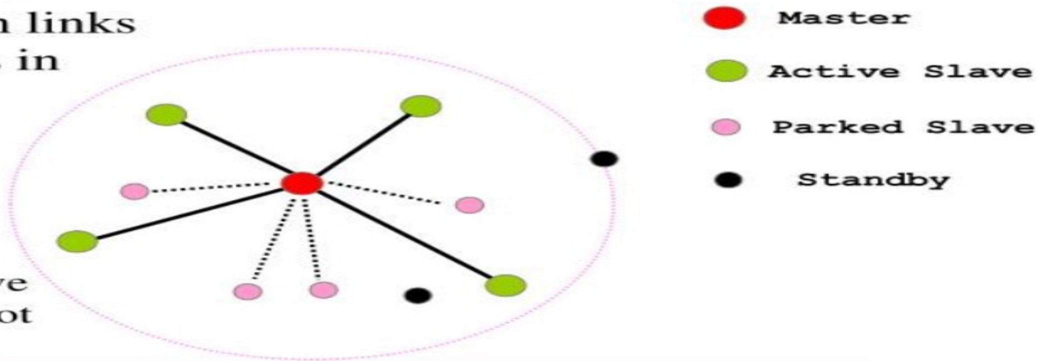
Communication in a Scatternet

If there are many independent piconets: there could be a collision on a particular channel, these **packets** will be lost and **retransmitted,** or if **voice** signals, it will be **ignored.**



Piconet formation

- Page - scan protocol
  - to establish links with nodes in proximity



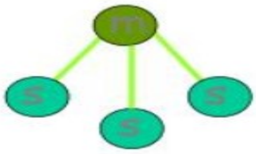
Direct, slave-to-slave communication is not possible.

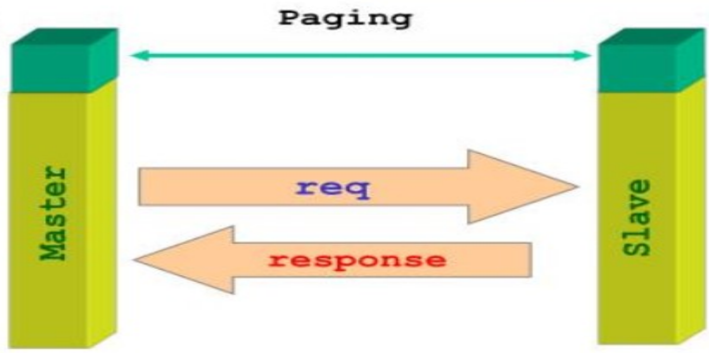
Piconet Addressing:
Active Member Address (**AMA**, 3-bits);
Parked Members Address (**PMA**, 8-bits)

Piconet Management

- Attach and detach slaves
- Master-slave switch
- Establishing SCO links
- Handling of low power modes
- (Sniff, Hold, Park)

Transmit power in Bluetooth applications for short-range connectivity is 0 dBm. allow up to 20 dBm

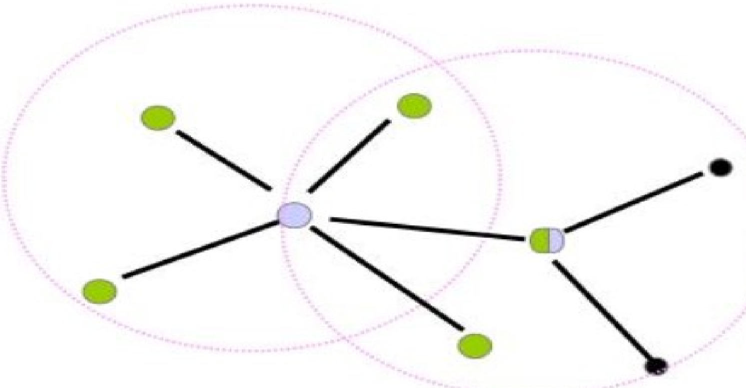




Operating modes

- Two modes: 1. As a **Master**, or 2. As a **Slave**. If it is Master that sets the frequency hopping sequences. Slaves synchronize to the Master in time and frequency by following the Master's hopping sequence.
- Every Bluetooth device has a **unique address**, and a **clock**. The baseband part of the Bluetooth specification describes an **algorithm** which can calculate a **frequency hop sequence** from a **Bluetooth device address and a Bluetooth clock.**
- When Slaves connect to a Master, they are told the Bluetooth device **address and clock** of the Master. They then use this to calculate the **frequency hop sequence**. Because all Slaves use the Master's clock and address, **all are synchronized** to the Master's frequency hop sequence.
- In addition to controlling the frequency hop sequence, the **Master controls when devices are allowed to transmit.**
- The Master allows Slaves to transmit by allocating slots for **voice traffic or data traffic**. In data traffic slots, Slaves are only allowed to transmit when replying to a transmission to the by the Master

Scatternet scenario

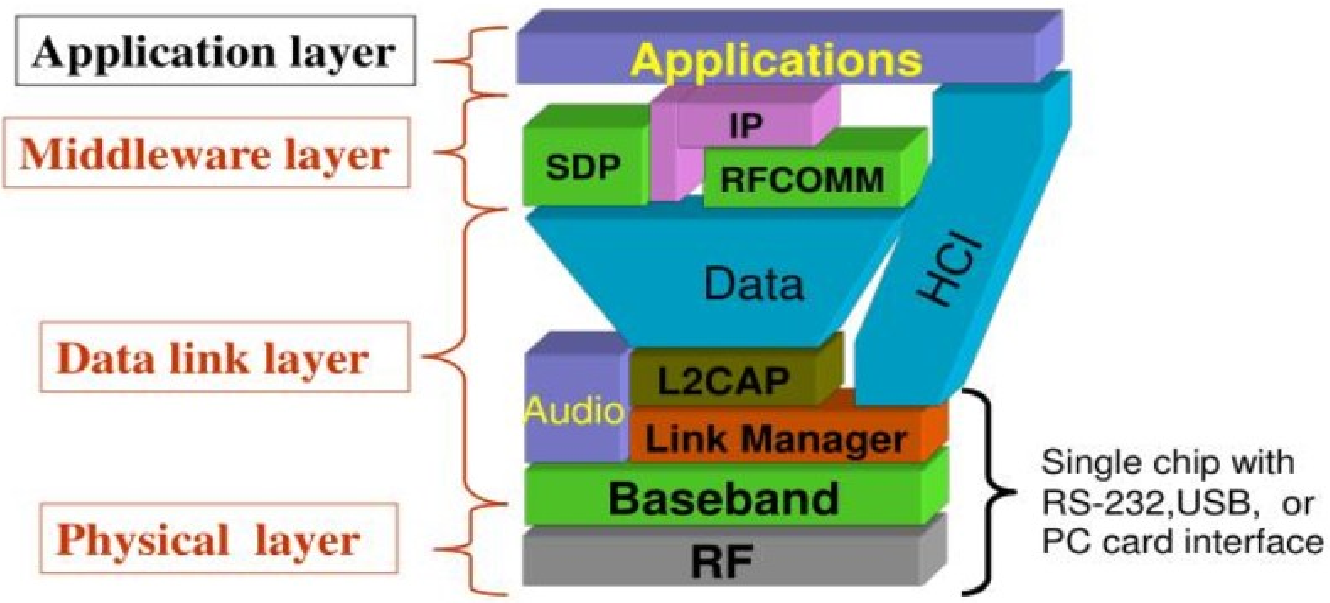


M in an existing Piconet might allow itself to be paged and connected to a new device and then **switch between S/M .**

This **M/S** switch is useful in situation where a connection has just been established by a device which normally wishes to be a **S**.

Mechanism involves the **S sending its FHS packet** to the **M**; **M** takes on a **CLK** offset to match the **S**'s CLK, while the **S** switches to using its own CLK.

Bluetooth protocol stack



Allocations of The Bluetooth protocol stack is a series of layers, through there are some features which cross several layers.

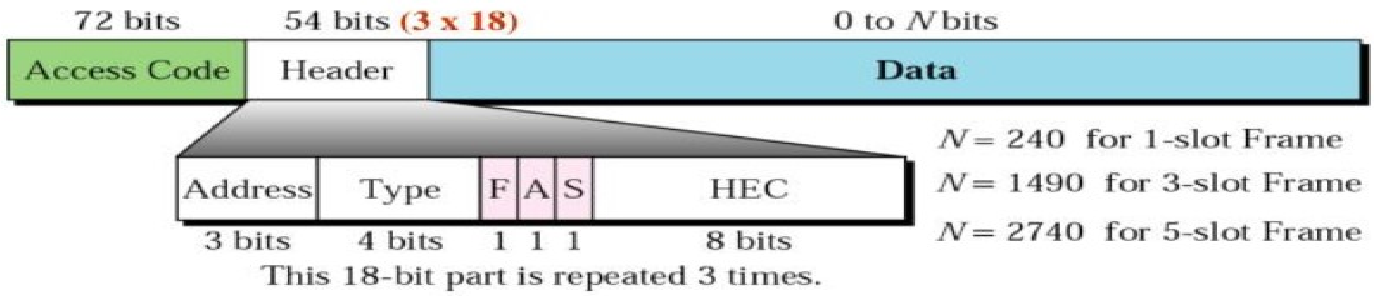
The Bluetooth Protocols (Cont)

e. **HCI** (The Host Controller Interface) **handles communications between a separate hosts and Bluetooth module.**

f. **LM** (The Link Manager) **controls & configures links to other devices. The LM translates commands into operations at the baseband level. following operations:**

- **Attaching Slaves to a piconet, and allocating their active member addresses**
- **Breaking connection to detach Slaves from a piconet.**
- **Configuring the link including controlling Master/Slave switches.**
- **Establishing ACL (data) and SCO (voice) links.**
- **Putting connections into low-power modes: Hold, Sniff, and Park.**

Frame format types



The Address **- identifies which of the 8 active devices the frame is intended for.**

The Type **- frame type (ACL, SCO)**

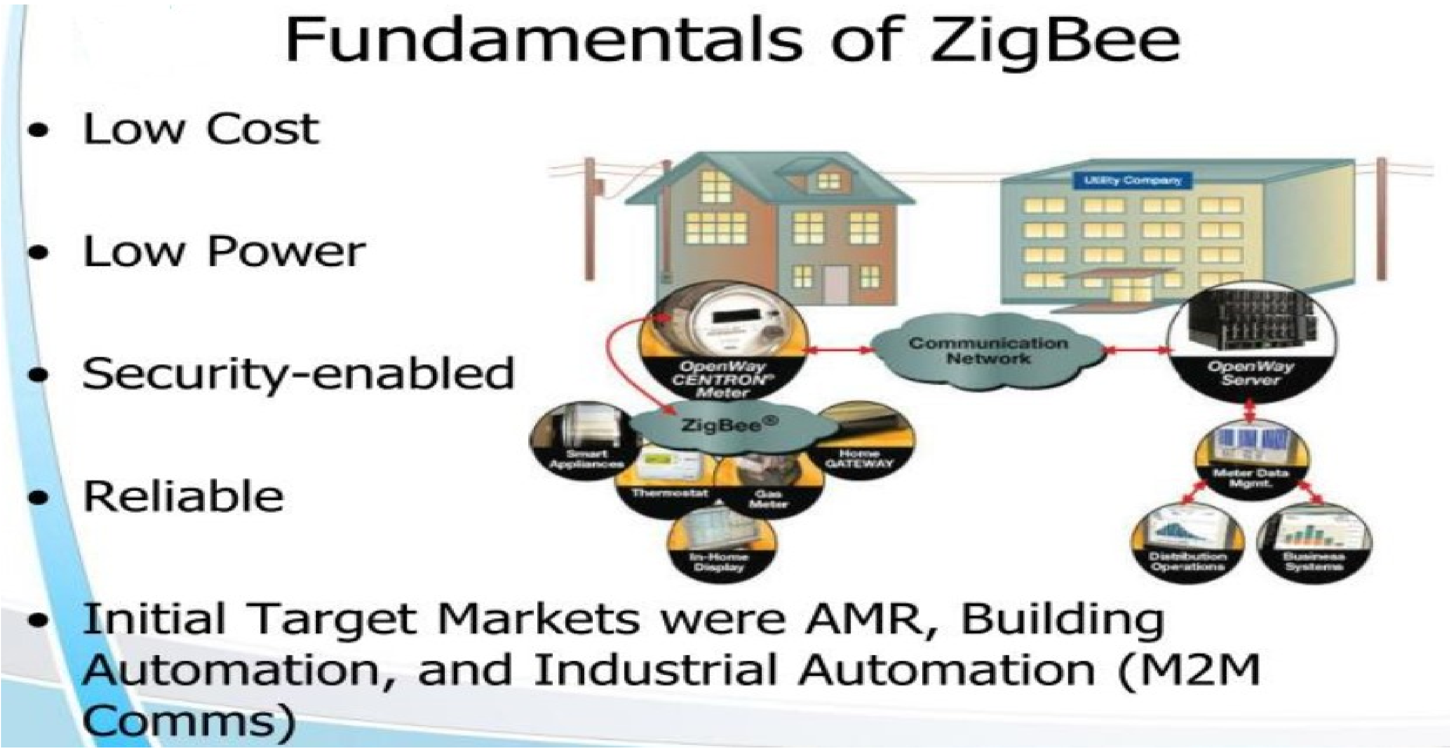
The Flow **- is asserted by a slave when its buffer is full and cannot receive any more data**

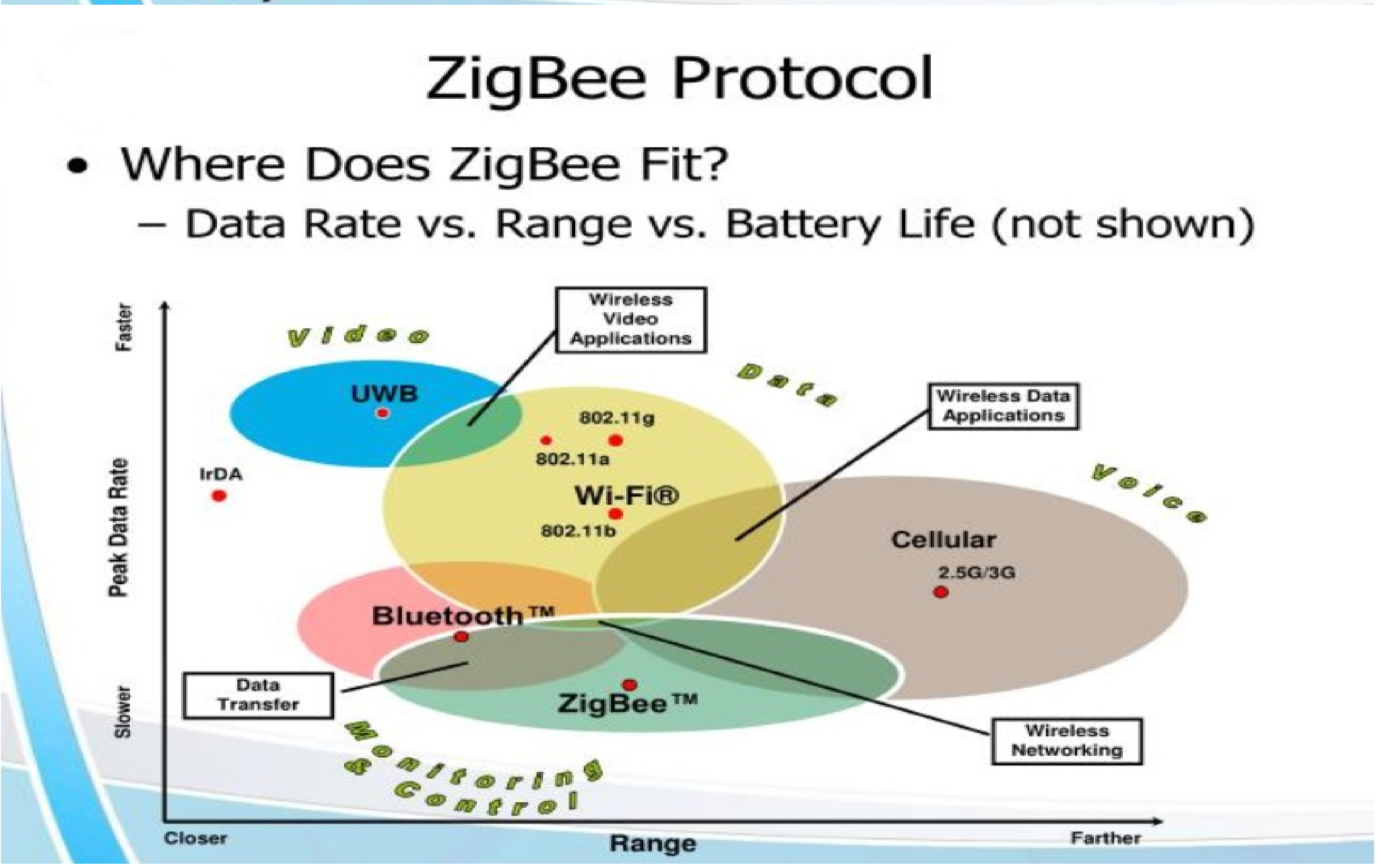
The Ack-ment **bit is used for ACK**

The Sequence **- is used to number the frames for retransmissions. The protocol is stop-and-wait.**

Checksum **The 18-bit header is repeated 3 times for a total of 54 bits *Header***

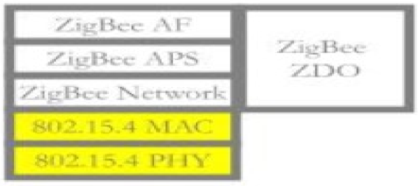
ZigBee Introduction





802.15.4 Protocol

- 802.15.4 Specifications
  - Supported Networks
    - Point-Point
    - Point-Multipoint/Star
  - Types of Nodes
    - Coordinator
    - End Node
  - Reliable Delivery
    - CSMA/CA
    - MAC-level (pt-pt) Retries/Acknowledgments
  - 64-bit IEEE and 16-bit short Addressing
  - 16 DSSS RF Channels

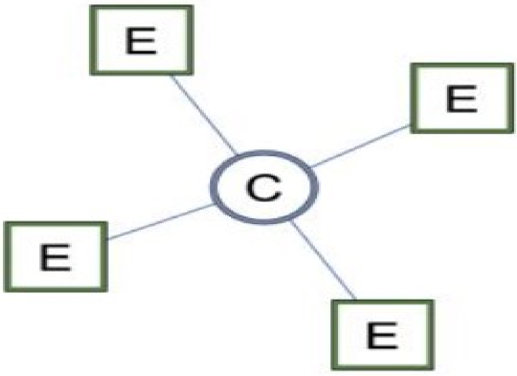


802.15.4 Protocol

- 802.15.4 Nodes in a PAN (Personal Area Network)

Coordinator
One per PAN
Establishes/Organizes a PAN
Mains-powered

End Device
Several can be in a PAN
Low power

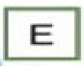


ZigBee Protocol

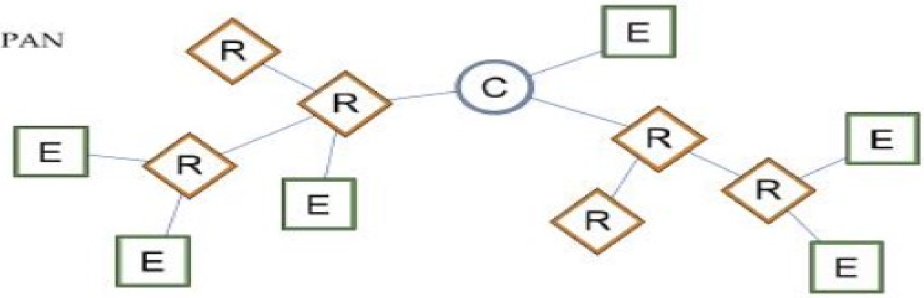
- ZigBee Nodes in a PAN (Personal Area Network)

- **Coordinator**
- One per PAN
- Establishes/Organizes a PAN
- Mains-powered

- **Router**
- Optional
- Several can be in a PAN
- Mains-powered



- **End Device**
- Several can be in a PAN
- Low power



ZigBee Protocol

PAN Network Formation

- **Coordinator** must select an unused operating channel and PAN ID
  - Energy scan on all channels
  - Sends Beacon request (**Broadcast** PAN ID)
  - Listens to all responses and logs the results
- After the Coordinator has started, it will allow nodes to join to it for a time based on the specified Node Join Time

C

ZigBee Protocol

- Router Startup
  - A new Router must locate a Router that has already joined a PAN or a Coordinator
    - Sends a **Broadcast** PAN ID on each channel
    - Returns sent via **Unicast**
  - Router will then try to join to a Router or Coordinator that is allowing joining

R

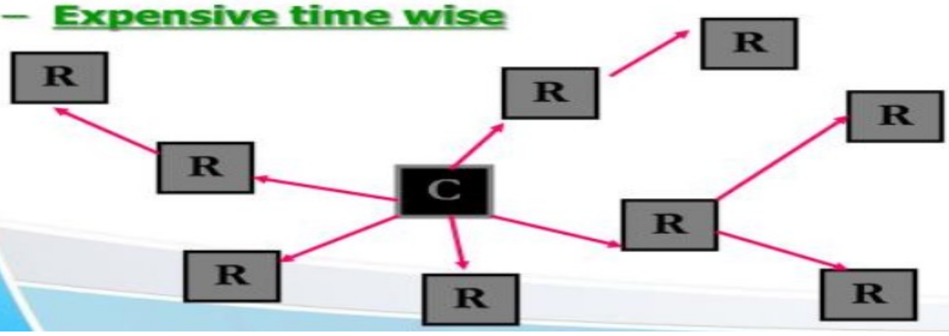
ZigBee Protocol

- End node: Low-power Sleep Modes
- End Node Startup
  - A new End node must locate a Router that has already joined a PAN or a Coordinator
    - Sends a **Broadcast** PAN ID on each channel
    - Returns sent via **Unicast**
  - End node will then try to join to a parent (Router or Coordinator) that is allowing joining

E

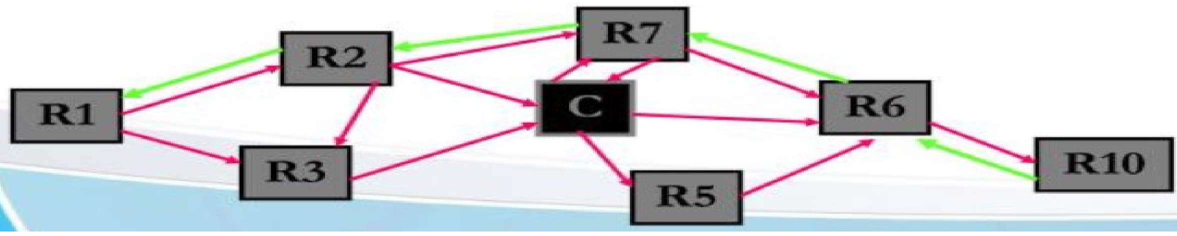
ZigBee Protocol

- Broadcast Transmissions - Relayed to All Nodes
  - No Acks are transmitted – Routers listen to neighboring Routers to know if message was retransmitted
  - Retransmit if neighbors are not heard (up to 2 times)
  - Broadcast Transaction Table used to ensure Routers do not repeat a message they have already repeated
  - **Expensive time wise**



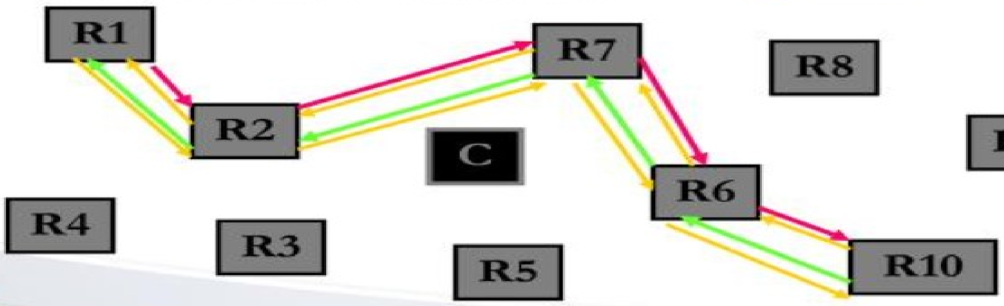
ZigBee Protocol

- Route Discovery consists of the following commands:
  - Route Request (**broadcast**)
    - 64-bit address used to find the local 16-bit address (Network address discovery)
    - Routing tables based on 16-bit address
  - Route Reply (**unicast**)
    - Positive acknowledgement returned
    - If node is gone- Network address discovery fails

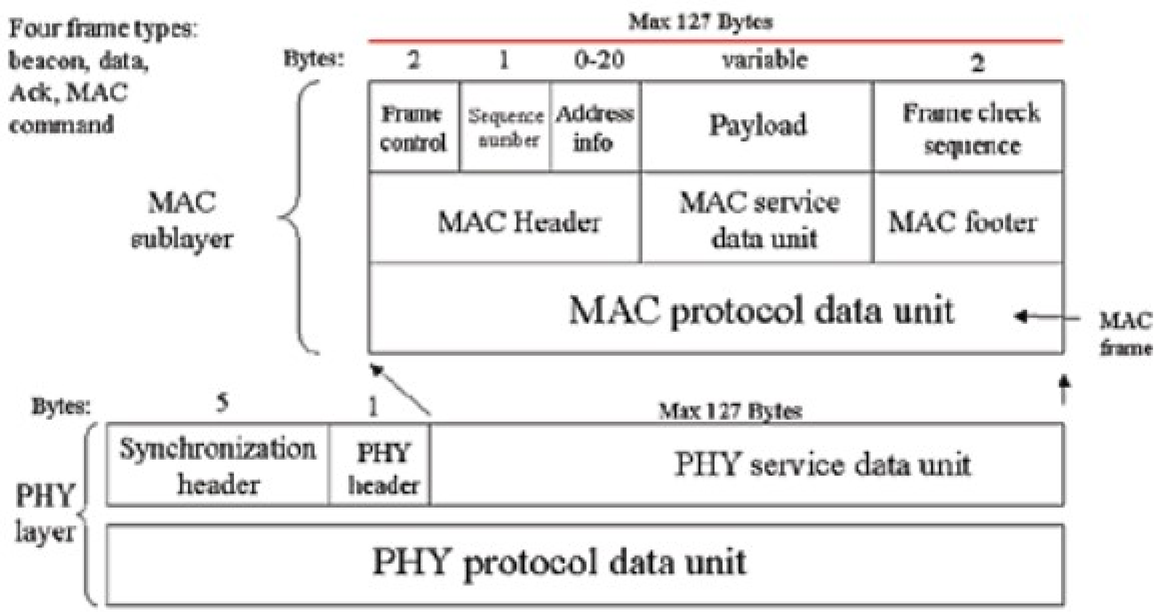


ZigBee Protocol

- Normal Data Transmissions (Unicast - established Network)
  - R1 must transmit data to R10.
  - MAC ACKs are transmitted for each hop.
  - One Network ACK is transmitted from the Destination node back to the Source



ZigBee Data Packet Format



Wireless Standards Comparison

Feature(s)	IEEE 802.11b	Bluetooth	ZigBee
Battery Life	Hours	Days	Years
Complexity	Very Complex	Complex	Simple
Nodes/Master	32	7	64000
Latency	Enumeration up to 3 seconds	Enumeration up to 10 seconds	Enumeration up to 30 milliseconds
Range	100m-1000m	10m	70m-300m (ETSI), 1600m (FCC)
Extendability	Roaming possible	No	Yes
RF Data Rate	11Mbps	1Mbps	250Kbps
Security	Authentication Service Set ID (SSID)	64-bit, 128-bit	128-bit AES and Application Layer user defined