**Chapter 14: Wireless LANs**

Table of Contents

[14.1 IEEE 802.11 2](#_Toc86834899)

[Wireless LAN Architecture 2](#_Toc86834900)

[MAC Layers 4](#_Toc86834901)

[CSMA-CA 5](#_Toc86834902)

[Frame Format 8](#_Toc86834903)

[Hidden Station Problem 10](#_Toc86834904)

[Exposed Station Problem 11](#_Toc86834905)

[14.2 Bluetooth 12](#_Toc86834906)

[Piconet 12](#_Toc86834907)

[Scatternet 13](#_Toc86834908)

[Bluetooth Layers 13](#_Toc86834909)

[Frame Format 17](#_Toc86834910)

[Pairing 19](#_Toc86834911)

[Bluetooth Low Energy 20](#_Toc86834912)

## 14.1 IEEE 802.11

The IEEE specifications for **wireless LAN** are defined in **IEEE 802.11**, which covers the physical and data-link layers.

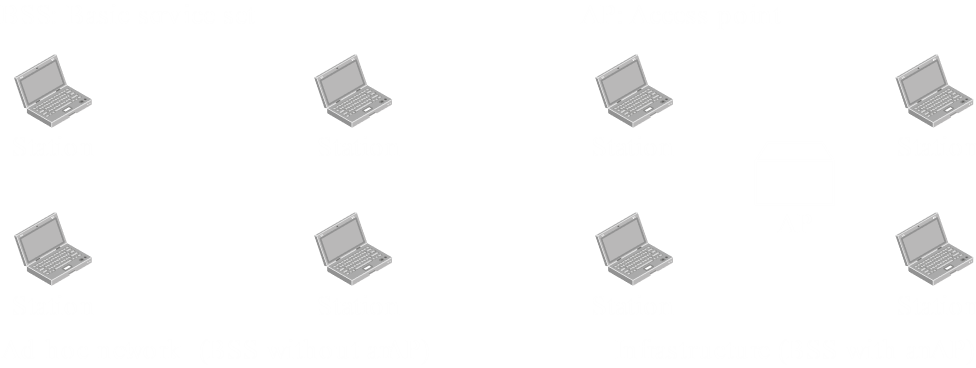
### Wireless LAN Architecture

The **architecture** of wireless LAN is of two types, **ad hoc** networks and **infrastructure** networks.

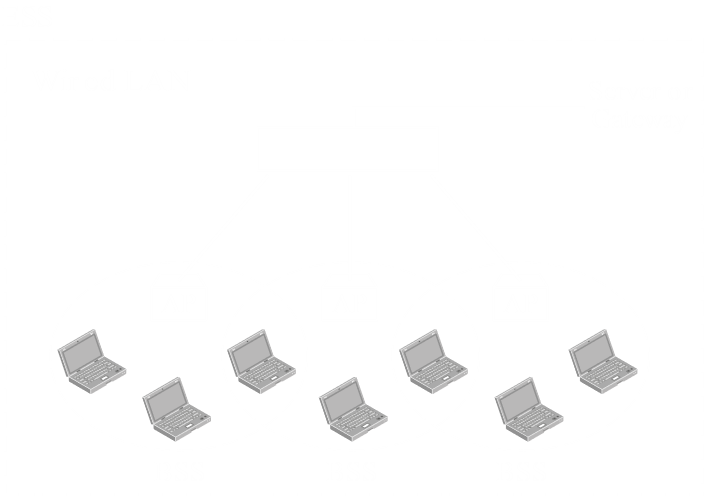
An **ad hoc** network is a **basic service set** (BSS) that has no central control point, or **access point** (AP). A BSS is just a smaller operational unit. The devices in an ad hoc network are thus going to transmit data amongst themselves through a wireless medium.

An **infrastructure** network is a BSS that does have an AP. All the devices on the network are connected through the AP. This is similar to how base stations work in GSM, though APs are not nearly as powerful. We generally use infrastructure networks. For example, when we connect to a home WIFI network, the actual router we are connecting to has an AP built into it which is how we can connect wirelessly. Otherwise, we would have to connect to the router using LAN cables.

For both ad hoc networks and infrastructure networks, **CSMA-CA** is being used.

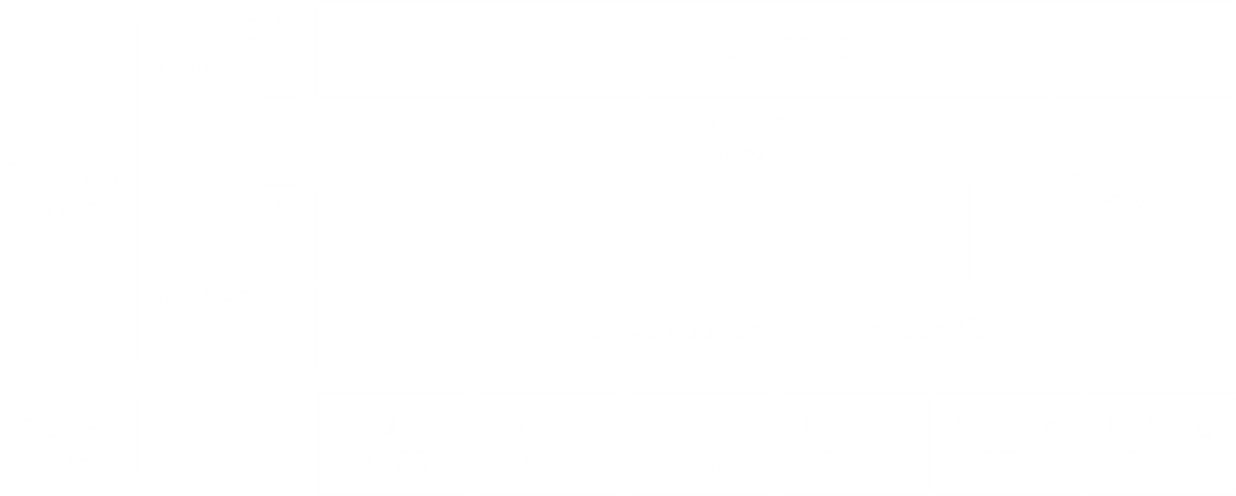


We can also have **extended service sets** (ESS), which is a group of infrastructure networks that have their APs connected through a distribution system. This allows devices on the same ESS to communicate with each other. They can also use the ESS to connect to a device on another ESS.



The connection between an AP and the distribution system of an ESS could be wired, or wireless, but it is preferable that it be wired. If it is, **CSMA-CD** can be used for those connections (not for the wireless connections within the BSS).

### MAC Layers



In the image above, we can see how the physical and data-link layers are implemented for wireless LAN.

In the **physical layer**, we can use one of multiple protocols:

* Frequency Hopping Spread Spectrum (FHSS)
* Direct Sequence Spread Spectrum (DSSS)
* Infrared
* Orthogonal Frequency Division Multiplexing (OFDM)

In the **data-link layer**, we firstly have the two general layers, the MAC layer and LLC layer. The LLC layer will not be discussed here.

Unlike wired LAN, the MAC layer for wireless LAN is divided into two portions, **Point Coordination Function** (PCF) and **Distributed Coordination Function** (DCF). PCF is a controlled access technique, while DCF deals with CSMA-CA. Note that the PCF protocol is not used in ad hoc networks. It only exists in infrastructure networks because both wired and wireless LANs are used together.

PCF actually has priority over DCF for a variety of reasons. This means that if a certain connection is using PCF, then it has a higher priority than DCF. On a normal network, this can cause issues, since the connection that uses DCF will never get the opportunity to transfer data. To avoid this, a **repetition interval** is defined. Essentially, there are two time periods, **contention-free** and **contention**. PCF connections use the network during the contention-free interval. After this, there is a contention interval during which DCF connections use the network.

### CSMA-CA

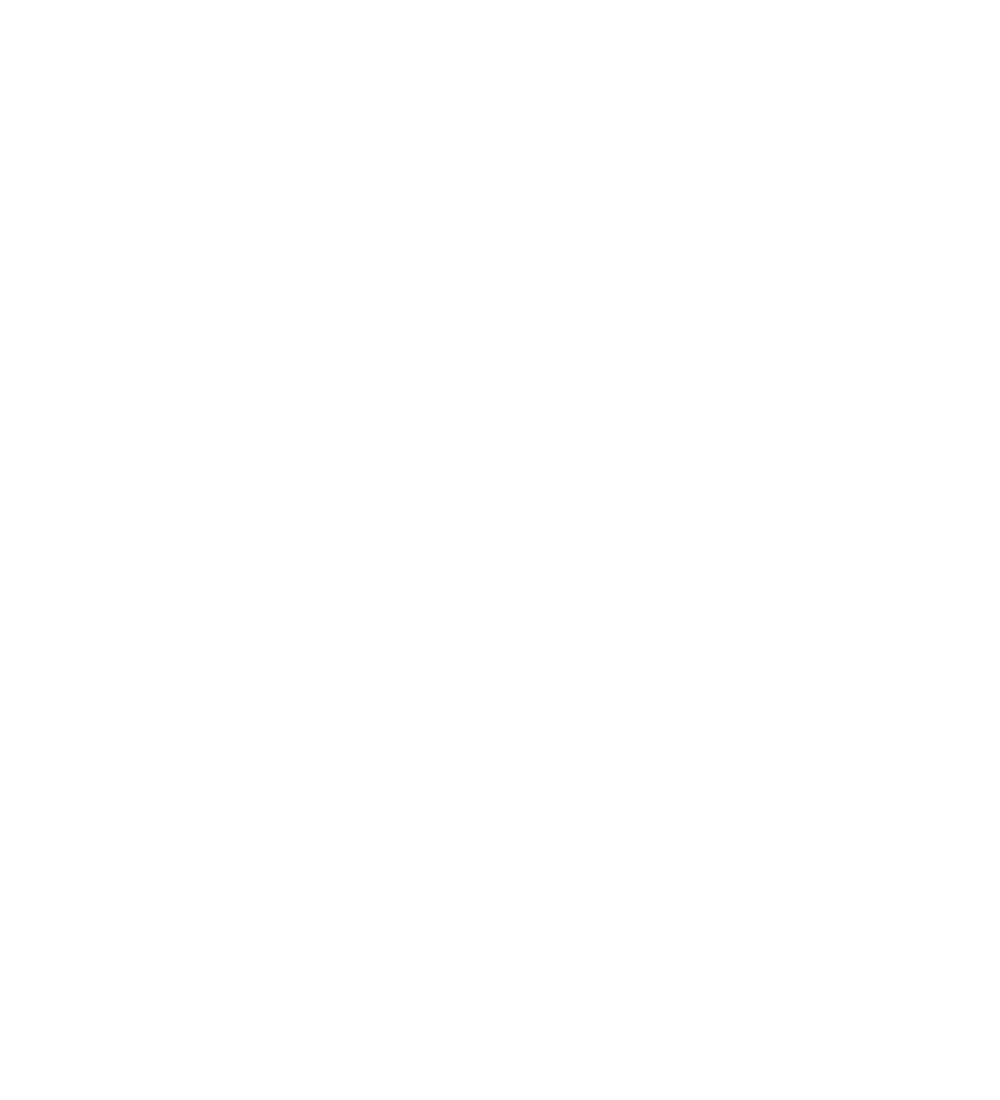
There were two issues with using CSMA-CD when it came to wireless communication:

1. **Attenuation** and **noise** make it difficult to differentiate between voltage levels.
2. The **hidden station** problem, which will be discussed later.

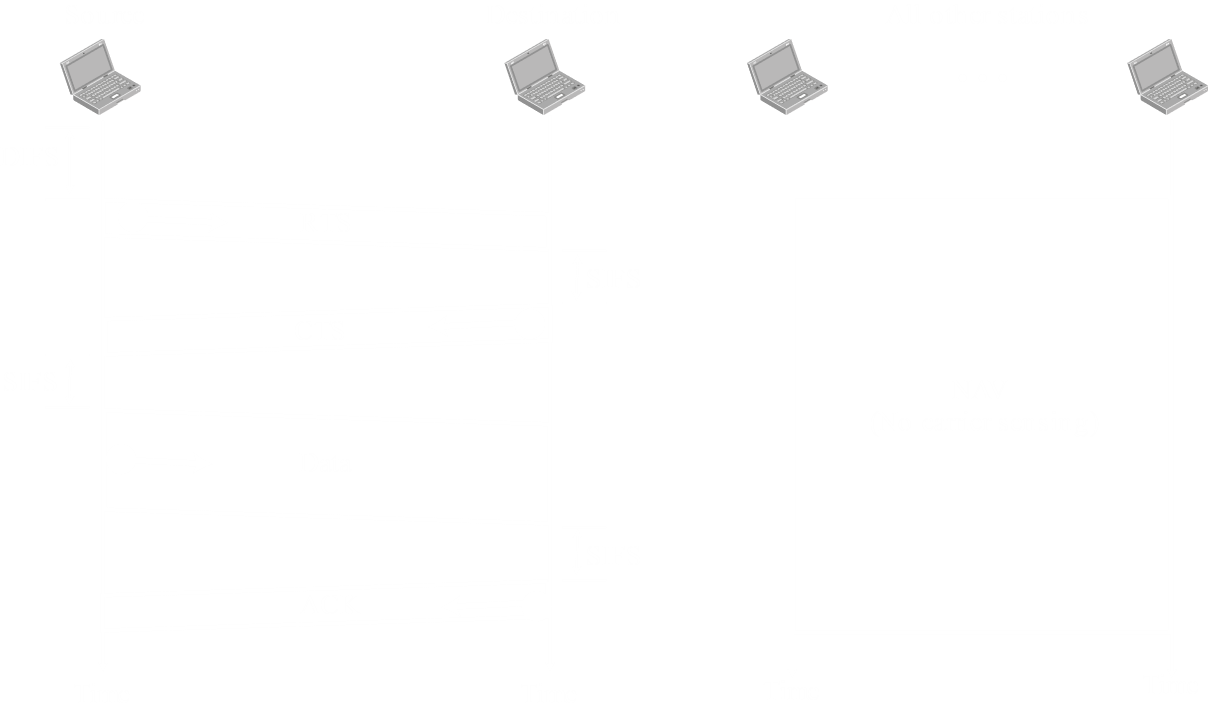
Due to these issues, CSMA-CA was developed.

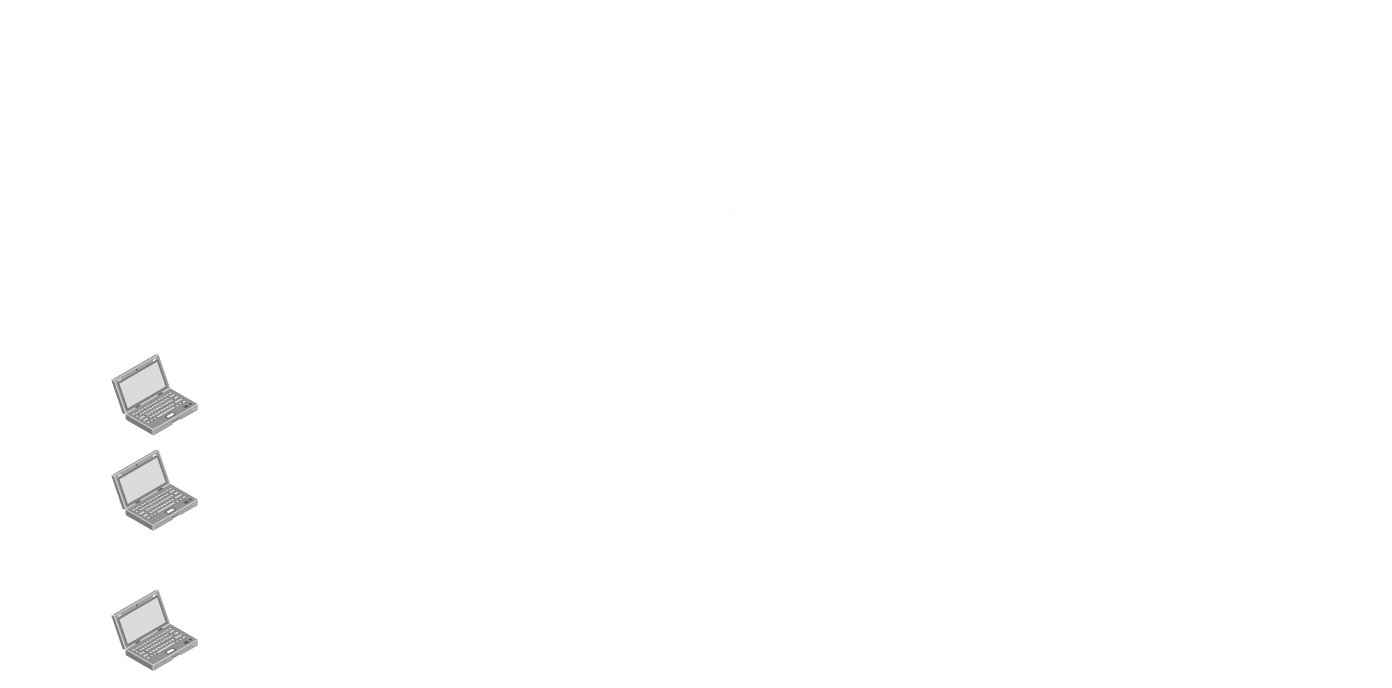
CSMA-CA is exceedingly conservative. We do not detect collisions, but by being careful, we can avoid them altogether.

1. After the **persistence strategy** stage, CSMA-CD would immediately start transmission. In CSMA-CA, we wait for the **distributed inter-frame space** (DIFS). We are also waiting for the **contention window** at this stage (not shown in diagram).
2. Once this time has passed, a **request to send** (RTS) frame is sent to the receiver. In response to this, the node is expecting a **clear to send** (CTS) frame. If this frame is not received withing a fixed amount of time, we go back to the back-off strategy. If we do receive it, we can keep going.
3. Once the CTS is received, we wait for the **short inter-frame space** (SIFS). The actual frame is sent after this time has passed. If an ACK is received within the expected time period, the frame is successfully sent. If the ACK is not received, we go back to the back-off strategy.

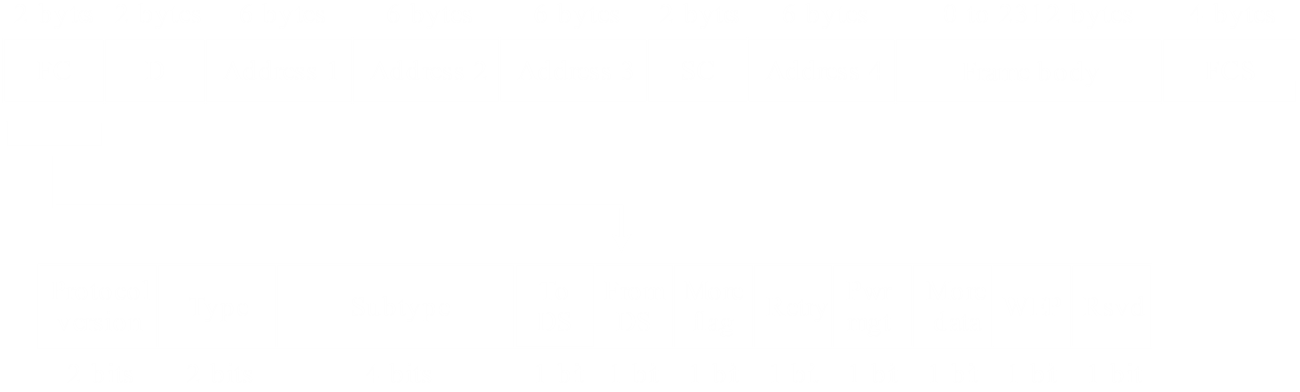


While all of this is happening, all other devices on the network are on a timer called the **network allocation vector** (NAV). During this time, they do not even check the medium. They are informed about how long this may be by the RTS frame, which has a special **duration period** value.





### Frame Format



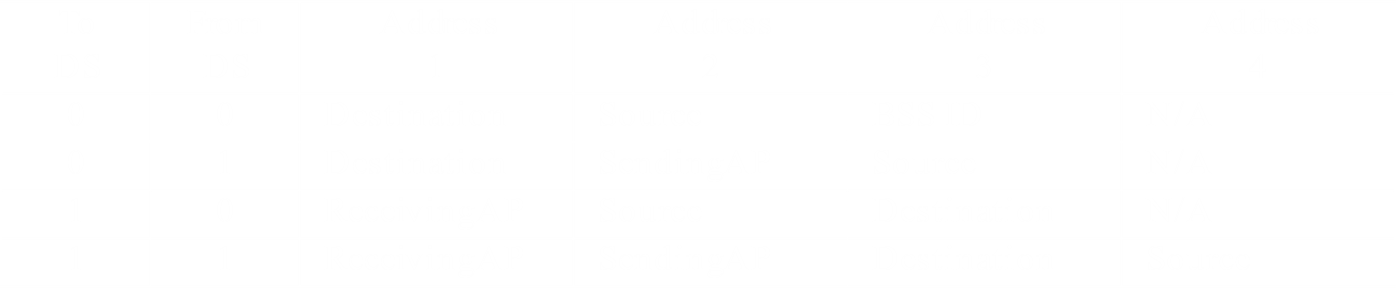
The parts of a wireless MAC frame are:

1. **Frame Control** (FC) – This is of 2 bytes or 16 bits and is discussed in depth below.
2. **Duration Field** (D) – These 2 bytes inform all devices on the network about how long their **NAV** should be.
3. **Address Fields** – There are four address fields, each of 6 bytes. Two of these fields are for the actual **devices**, while the other two are for **AP**s that may or may not be required depending on whether we are communicating within the same **BSS** or not.
4. **Frame Body** – Ranging from 0 to 2312 bytes, this contains the actual data sent from the upper layers.
5. **Forward Correction Sequence** (FCS) – These 4 bytes are used for error correction mechanisms.
6. **Sequence Control** (SC) – These 2 bytes are used for flow control.

In the 16 bits of **frame control**:

1. The first 2 bits contain information about the **protocol version** being used.
2. The second 2 bits inform us about the **type** of frame being sent. The frame could be a **control** frame, a **management** frame or a **data** frame. There are only 3 types of frames, but we are forced to use 2 bits nonetheless. One combination goes unused.
3. The next 4 bits contain the **subtype**, since every type of frame can themselves be of several types. **Control** frames can be of three types: RTS, CTS or ACK.
4. The **To DS** and **From DS** bits define whether the frame is coming from a **distribution system** and going to a distribution system. Essentially, we know whether or not the sender and receiver are from the same BSS.

The different combinations of these two bits give us addresses like this:

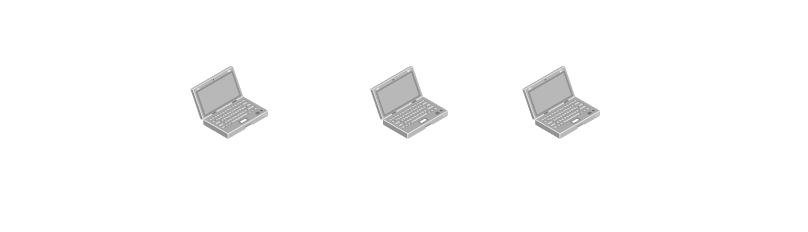


1. The **More Flag** tells us whether or not there are more fragments of this data coming after this.
2. The **Retry** bit tells us if the current frame is a retransmitted frame.
3. The **Power Management** bit tells us about whether the station is in power management mode. We do not need to know what that is.
4. The **More Data** bit is similar to the More Flag, telling us if the station has more data to send.
5. The **Wired Equivalent Privacy** (WEP) bit tells us if encryption is being used.
6. The final bit is the **Reserved** bit.

### Hidden Station Problem

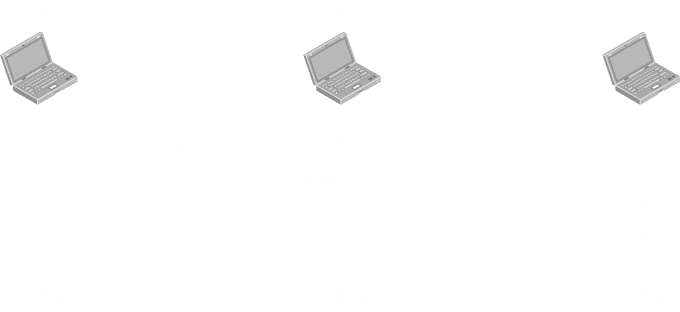
The **hidden station** problem was mentioned as one of the reasons CSMA-CD could not be used.

Every station has a specified **range** within which it can transmit data. Any device within that range can be transmitted to. With this information, consider the following diagram:



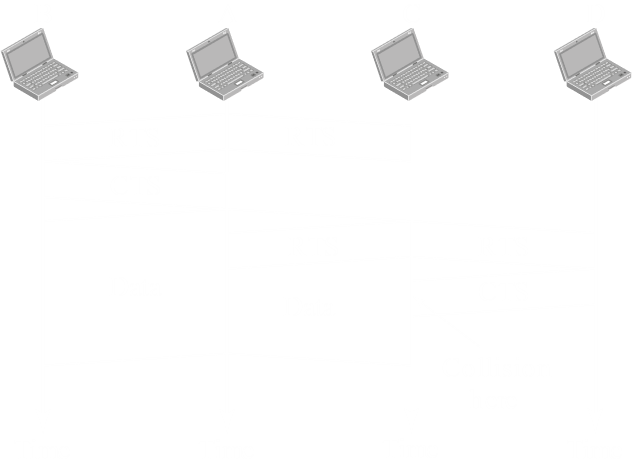
The station A is within the range of both the stations B and C, but the stations B and C are not within each-other’s ranges. This means that station B will be **unaware** of the existence of station C and vice versa. If stations B and C both want to transmit information to station A, both stations will believe they are the only ones trying to transmit, which will result in a collision.

The solution to this is the use of **RTS** and **CTS**. Stations B and C are both within the range of station A, so the CTS frame it sends out in response to any RTS frame will be received by both of them. In the diagram below, the RTS from station B is being responded to, but the CTS also goes to station C. Station C becomes aware of the existence of station B and the fact that that station is transmitting data with station A right now. The CTS frame has a **duration bit**, similar to the RTS, which allows station C to know how long it needs to wait using NAV.



### Exposed Station Problem

The **Exposed Station Problem** is the opposite of the Hidden Station Problem, where a device behaves too conservatively.



In the figure above, if station A wants to send data to station B and station C wants to send data to station D, both can do so without any issues. However, station C is exposed to the transmission by station A, and cannot send data to station D at this time. If station C sends an RTS to station D, the CTS it gets in return collides with the transmission from A, thus never reaching C. This means the full capacity of the channel is not being used and the use of RTS and CTS cannot help in this case.

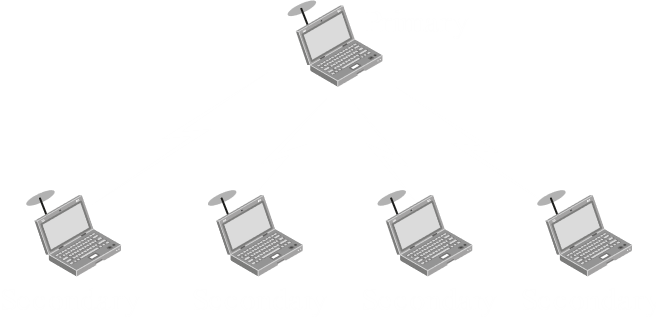
## 14.2 Bluetooth

Bluetooth allows data to be transferred **wirelessly** within a **short-range**. Bluetooth falls into the category of **Personal Area Network** (PAN). It is an **ad-hoc network**, meaning there is no controlling device. It operates in the Industry, Scientific and Medical (**ISM**) radio band, which does not require any licensing.

Bluetooth networks can be of two types, **Piconet** and **Scatternet**.

### Piconet

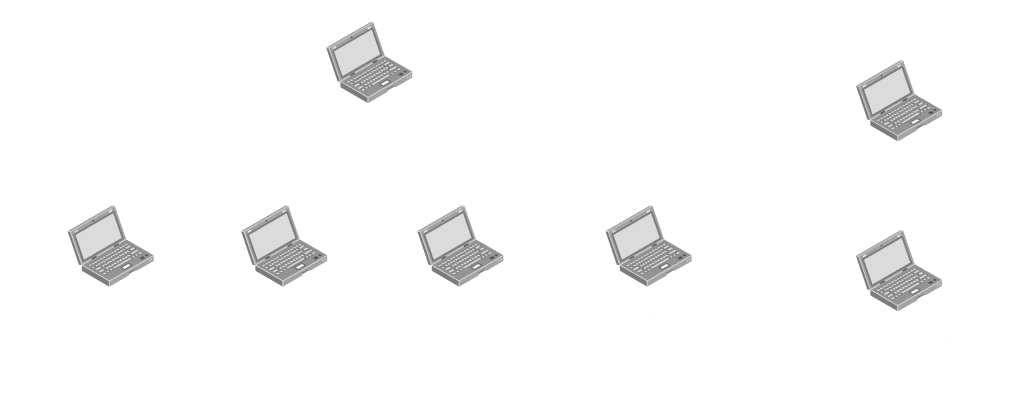
In **piconet**, a controller exists. A maximum of **8 secondary devices** can connect to the **controlling device**. However, the secondary devices cannot communicate in between themselves. They can only communicate with the controller.



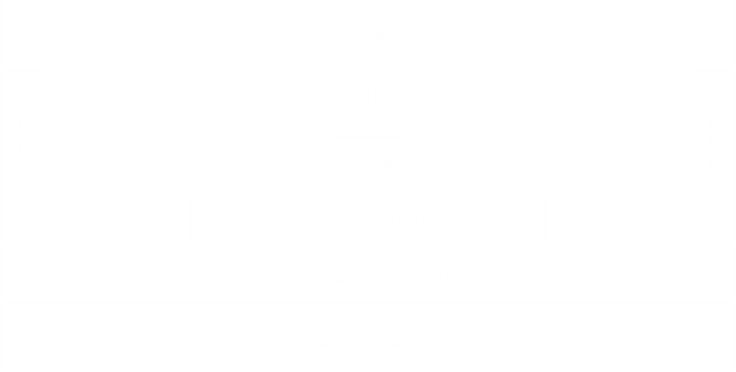
One example is a computer. We can connect multiple devices to a computer via Bluetooth, but those devices can only communicate with the computer, not between themselves.

### Scatternet

A **scatternet** is like multiple piconets joint together. One controlling device connects to multiple secondary devices, and some secondary device is itself a controlling device, and is thus connected to other secondary devices.



### Bluetooth Layers



Similar to how TCP/IP is divided into layers, Bluetooth can also be divided into layers.

#### Radio Layer

First comes the **Radio Layer**. This is similar to the Physical layer. The 2.4 GHz ISM band is divided into 79 channels of 1 MHz each. The first channel starts at 2402 MHz.

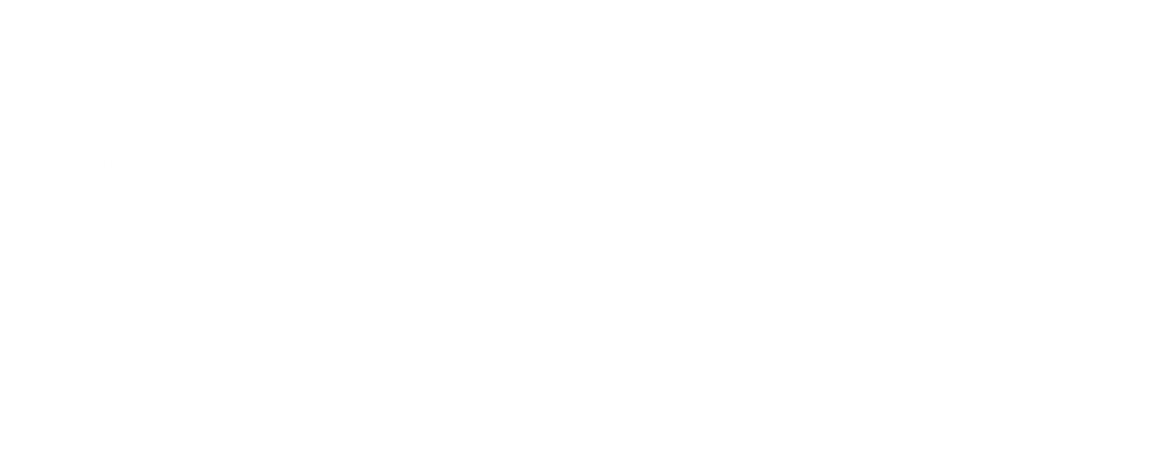
One would imagine that having such a small number of available channels would cause interreference, and it would. To avoid this, a **Frequency Hopping Spread Spectrum** (FHSS) is used. Essentially, carrier frequencies are changed 1600 times per **second**.

To identify bits of data, **Gaussian Frequency-Shift Keying** (GFSK) is used. The easiest explanation of this is a 1-bit is indicated if the actual frequency is a little higher than the carrier frequency and a 0-bit is indicated if the actual frequency is a little lower that the carrier frequency.

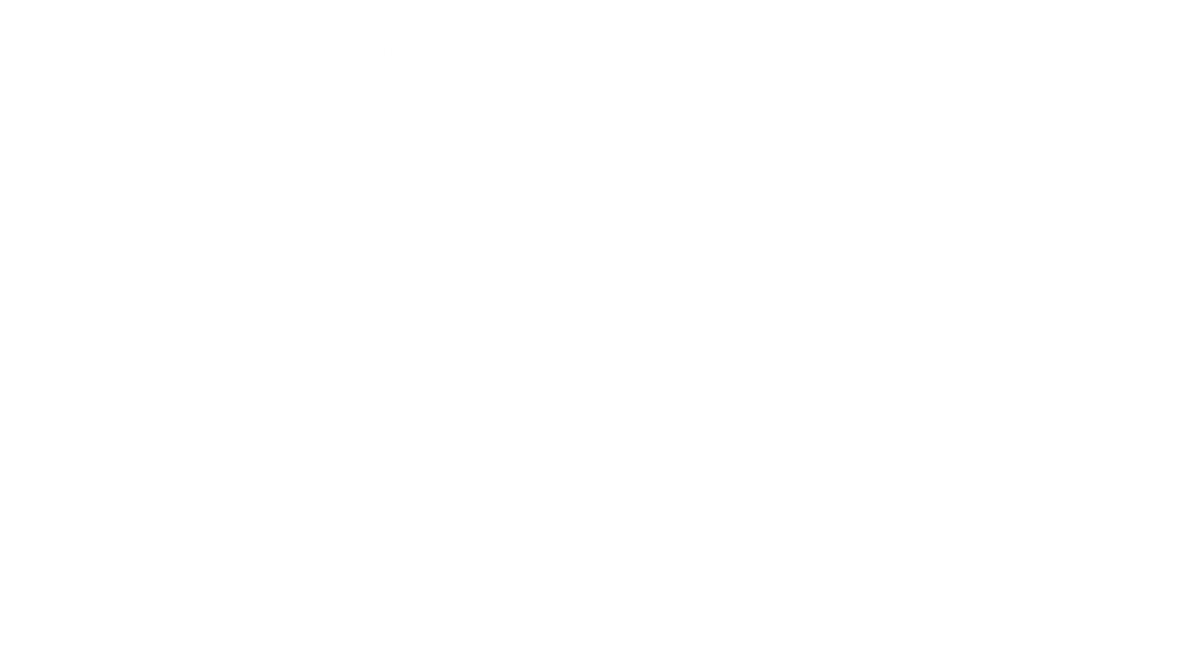
#### Baseband Layer

Next comes the **baseband layer**, which is similar to the MAC section of the data-link layer. This layer handles **time-division multiple access** (TDMA). In Bluetooth, half-duplex communication is used. This means only one party can send data at a time. We know that frequencies are changed 1600 times per second, meaning one frequency is used for . For this time, one of the devices will be able to send data.

Imagine we have one controller and one secondary device. These two devices will alternate. This results in the **controller** getting every **even division**, and the **secondary device** getting every **odd division**.



If we have multiple secondary devices, the controller still gets all the even divisions, but the odd divisions are divided amongst the secondary devices. Only the secondary device with which the controller decided to communicate in an even division will be able to respond in the next odd division. Essentially, this is **Polling**.

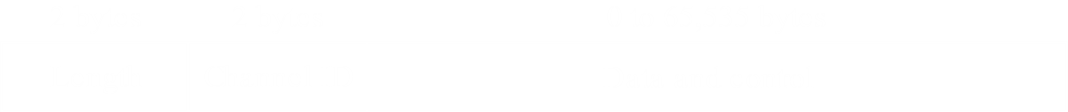


#### L2CAP Layer

The L2CAP layer provides several services:

* **Protocol Multiplexing** – The data from the upper layer cannot be understood directly by the baseband layer. The L2CAP layer multiplexes the data from the upper layer in a way so as to allow the baseband layer to understand it.
* **Segmentation and Reassembly** – The upper layers can send a huge amount of data. The L2CAP layer breaks this data into 1-slot frames or 3-slot frames or 5-slot frames so that the baseband layer can send it. The breakdown process is called **segmentation**. On the receiver’s end, the L2CAP layer **reassembles** the fragments.
* **Quality of Service** – Users can set a specific quality of service in Bluetooth. If nothing is set, the devices will simply try their best to transmit data.
* **Group Management** – This refers to multicast data transfer.
* **Frame Format**

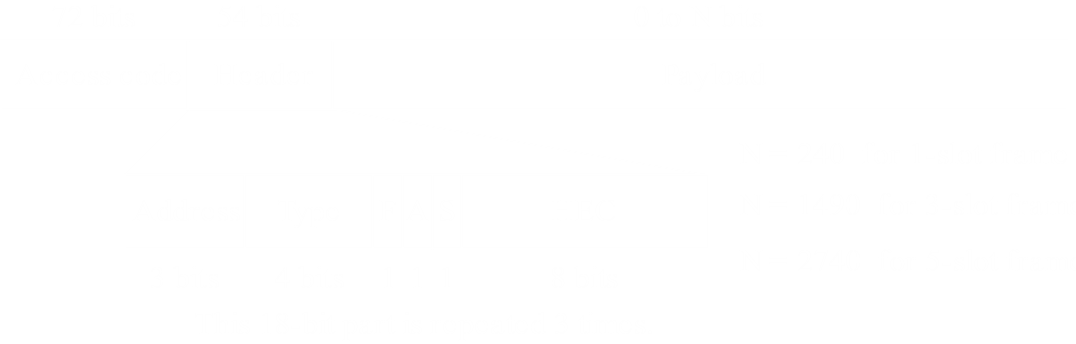
Regarding the last feature, the frame format is as follows:



This is simple enough to understand. The only thing worth discussing is the **Channel ID**.

The Channel ID identifies which device is being communicated with. If the channel ID used by secondary devices to communicate with the primary device is the same, it does not matter. However, the channel ID used by the primary device to communicate with each of the secondary devices needs to be different.

### Frame Format



A **Bluetooth frame**, the thing being transmitted by the baseband layer, essentially has 3 parts:

1. A 72-bit **access code**, which allows us to identify the **controlling device**
2. A 54-bit **header**
3. The **data**

For the **header** section, the 54-bits are further divided into sections.

1. 3-bits for the **address**, which identifies one of the 8 **secondary devices**.
2. 4-bits for the **type**, specifying the data type coming from the upper layer.
3. A 1-bit flag, **F**, for **flow-control**. If a device cannot receive more frames (its buffer is full), this flag is set to 0.
4. A 1-bit flag, **A**, for **acknowledgement**, which is set to 1 to indicate an ACK.
5. A 1-bit flag, **S**, for **sequence**, which denotes the sequence number of the frame to allow detection of retransmissions. Only 1 bit is needed since Bluetooth uses **Stop-and-Wait ARQ**.
6. An 8-bit **HEC**, which is used for error-detection using a checksum.

This makes up a total of **18 bits**, which are **repeated 3 times** to form the 54-bit header. This is to compensate for possible **errors**. For example, if one bit is set to 1 for 2 of the sets and is set to 0 in the third set, then the receiver will assume that the bit is supposed to be 1. This odd technique needs to be used since the environment for Bluetooth connections tends to be very noisy.

The **data** section can be of three types:

1. 1-slot frame
2. 3-slot frame
3. 5-slot frame

Each of these essentially tells us how many **time divisions** are being used to send the data.

For a **1-slot frame**, the data section can be **240 bits**. Looking back at the baseband layer frames, we will notice that the actual frame transmission only occurs for . The rest of the time is occupied by synchronization and the actual hopping. This is why the frame size can only be bits, since we use a bandwidth, meaning 1 bit is transferred every micro-second.

For a **3-slot frame**, the data section can be **1490 bits**. The same frequency is used for thrice the normal time, so is available for the frame itself. This leaves bits for the data.

Similarly, for a **5-slot frame**, the data section can be **2740 bits**, since and .

### Pairing

The steps of pairing between Bluetooth devices are:

1. **Discovery** - The **controlling device** begins searching for Bluetooth devices available within a **10-meter** radius. Any available **secondary devices** make themselves visible to the controlling device.
2. **Pairing** – To connect to a specific secondary device, the secondary device must send the controlling device its **authentication code**.
3. **Detection of Profiles** – Every Bluetooth device supports some **profiles**, which specifies some services and functions. The pairing device needs to ensure that they support the same profiles.
4. **Connection** – The pairing devices are connected based on one of the profiles that both devices support.
5. **Data Synchronization** – Data is transferred between the connected devices.

### Bluetooth Low Energy

**Bluetooth Low Energy** (BLE) is a new technology that is currently being used in place of classical Bluetooth.

Features:

* As the name suggests, it uses very **low energy**. In fact, if we turn on a Bluetooth device and leave it on for several days, an insignificant amount of charge will be consumed. For this reason, it is used in medical equipment.
* BLE has a **higher data-rate**, 3 Mb/s compared to the 1 Mb/s of classical Bluetooth. However, this is still very low compared to other forms of communication.
* An **unlimited number** of secondary devices can be connected to the controlling device.
* BLE has **better security** than classical Bluetooth. However, it is still less secure than other forms of communication. One example is **Bluejacking**, in which data is sniped from a Bluetooth connection.