**Chapter 13: Wired LANs – Ethernet**

Table of Contents

[13.1 IEEE Standards 2](#_Toc86834857)

[13.2 Standard Ethernet 3](#_Toc86834858)

[MAC Frames 3](#_Toc86834859)

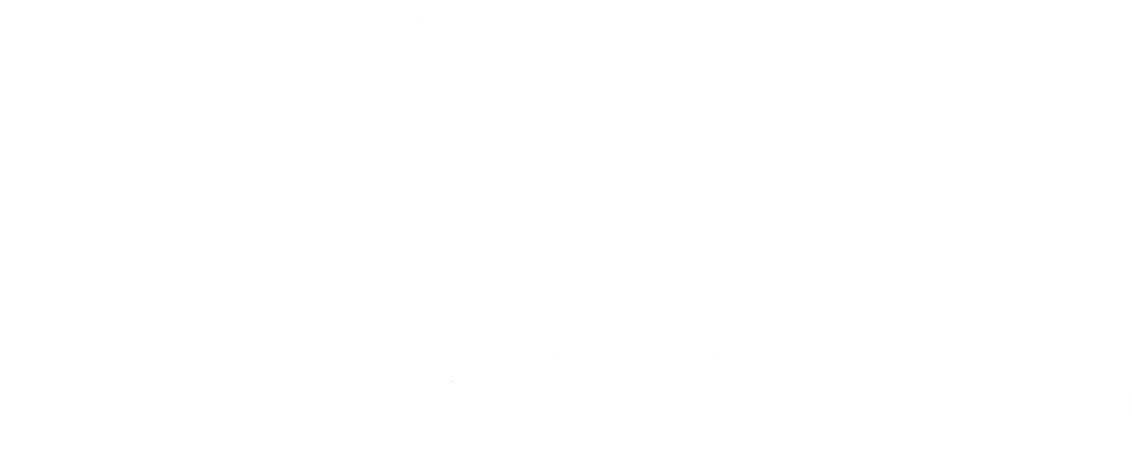
[Frame Sizes 5](#_Toc86834860)

[MAC Addresses 6](#_Toc86834861)

[Collision Detection 7](#_Toc86834862)

## 13.1 IEEE Standards

The standards related to Wired LAN are from **IEEE 802.3**. The most commonly used standard is called **Ethernet**.



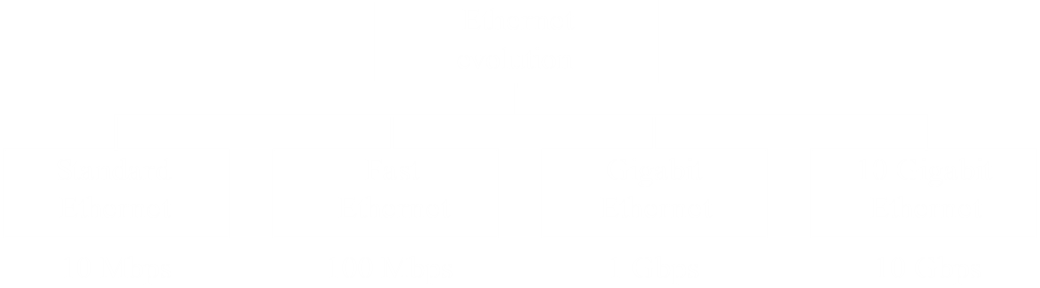
As can be seen in the diagram above, the Ethernet protocol is divided between the **MAC Layer** in the Data Link Layer and the **Physical Layer**.

In reality, what protocol will be used in these two layers depends entirely on what standard is being used in the **transmission medium**. If the transmission medium is using Ethernet, we use Ethernet protocols. If it is using a Token Ring, we use Token Ring protocols and so one.

However, the layers above these will be unaffected by these changes. This allows networks with different physical implementations to connect with each other.

## 13.2 Standard Ethernet

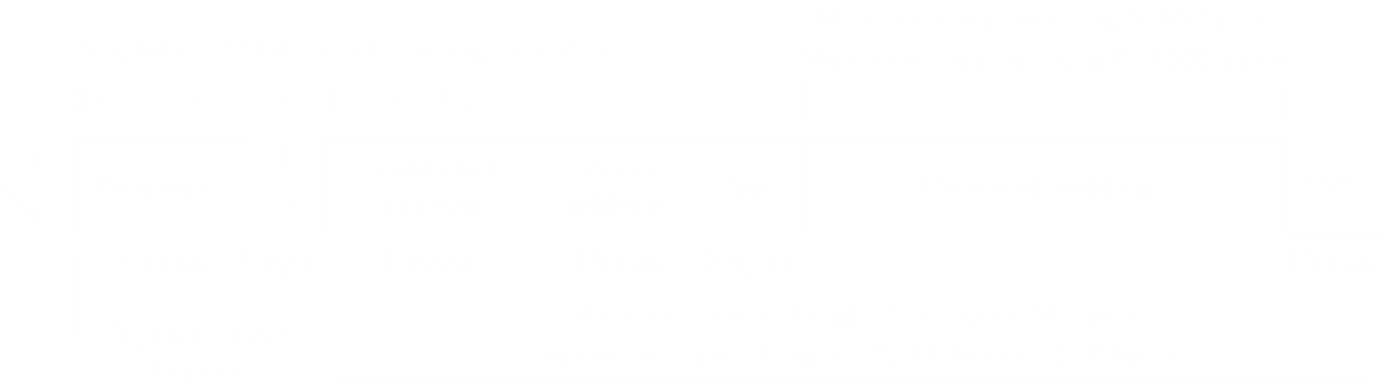
There are actually several versions of Ethernet.



Currently, **Fast Ethernet** is commonly used, but higher speed versions are available.

### MAC Frames

In Wired LANs, we transfer **frames**. These frames are also called **802.3 MAC Frames**. The format for these frames is like this:



The first two sections, **Preamble** and **SFD**, are actually part of the **Physical Layer Header**, meaning they are attached when the frame gets to the Physical Layer, not in the MAC Layer. The **Preamble** indicates that a new frame is about to start. It has **56 bits of alternative 1s and 0s**. The **SFD** indicates that the actual frame starts just after this byte. It acts as a **delimiter** and has the format **10101011**.

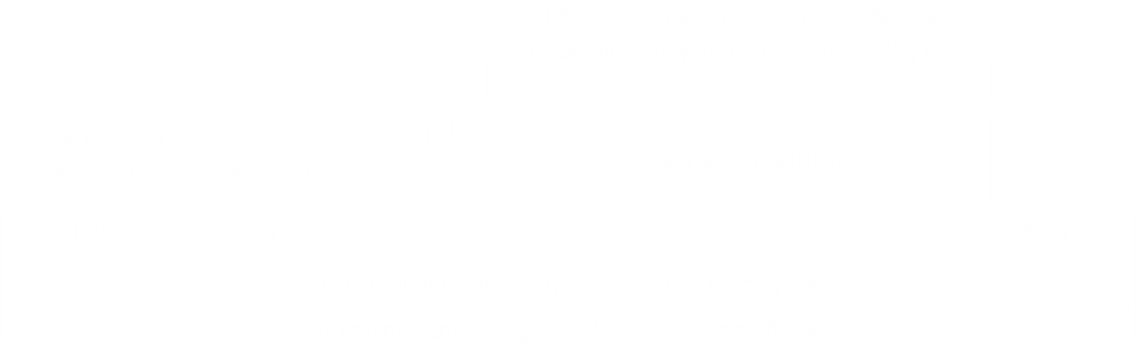
The **Data and Padding** comes from the **Upper Layers**, so that part can also stay vague for now.

The **Destination Address** and **Source Address** are being added in the **MAC Layer**. These are **MAC Addresses**, or Physical Addresses. The Source Address is simple enough to find, since it is the MAC Address of the current machine, but the Destination Address is the MAC Address of the next node. This needs to be retrieved using the **ARP Protocol**. The ARP Protocol in turn needs to know the **IP Address** of the next node in order to retrieve its MAC Address. The IP Address is provided by the **Network Layer** from a **Routing Table**. However, the ARP Protocol is handled by the Network Layer and is not being executed by the Data Link Layer right now. Essentially, by the time we are creating the MAC Frame, we already know the Destination Address.

The **Length or Type** represents either the length of the packet or what type of packet is being sent.

### Frame Sizes

The MAC Frame has a minimum and a maximum size. The minimum size is **64** bytes and the maximum size is **1518 bytes**.



As can be seen in the diagram above, the difference in size is due to the size variations in the actual **Data and Padding** section, which can vary from **46** bytes to **1500 bytes**.

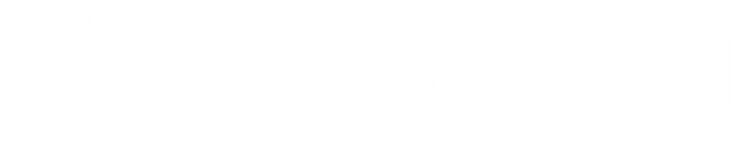
The minimum frame size restriction is because having frames smaller than that will prevent **CSMA-CD** from detecting **frame collision**. The frame needs to be transmitting for at least **double the propagation time**.

Of course, this depends on the **speed** of the connection. The scenario we are looking at here is for **standard ethernet**, which works at **10 MBps**. Essentially, the maximum propagation time is and since we are transmitting at , the size needs to be or . Faster connections will require a larger minimum size, so that the frame is transmitting for enough time.

The maximum frame size restriction is to **prevent monopolization** of the network and because having frames that are larger would require **larger buffers**, which is expensive.

### MAC Addresses

The source and destination addresses are of the following format:



There are **6 divisions**, each with **2 hex digits**, giving a total of **12 hex digits**. Each hex digit is **4 bits**, giving a total of **48 bits** or **6 bytes**.

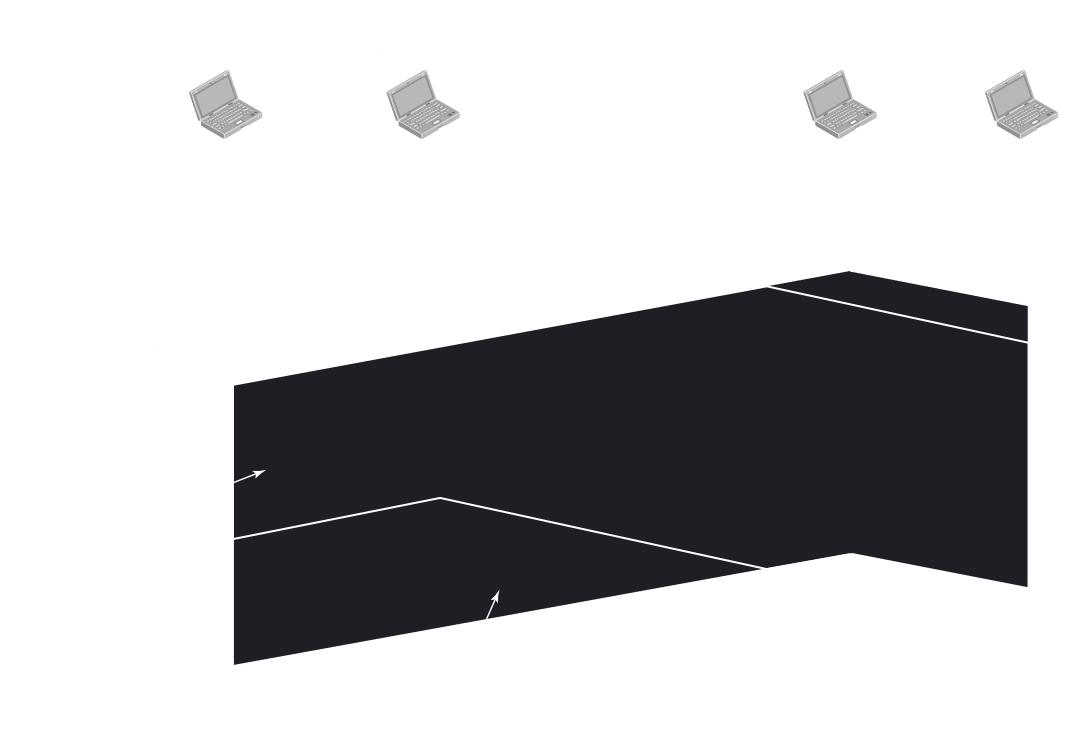
A **Unicast** MAC Address has an **even LSB** in the **first division**, for example , since . This indicates that there is only **one destination**.

A **Multicast** MAC Address has an **odd LSB** in the **first division**, for example , since . This indicates that there are **multiple destinations**.

A **Broadcast** MAC Address just has all the digits set to , as in . This indicates that the message is meant for **all nodes** in the network.

On the transmission medium, **each byte** of the address is sent out from **left-to-right**. For each byte, the **bits** are sent out **right-to-left**. Thus, is sent as . Notice how , which is in binary, is on the left but the bits are flipped.

### Collision Detection



We have previously seen how collision detection in CSMA-CD works. The diagram above should serve as a reminder.