Network Basics

Ethernet Basics

Ethernet

Ethernet is the standard technology for connecting devices in a LAN or WAN and allowing them to communicate using a protocol, which is a collection of rules or a common network language. Ethernet explains how network devices format and transmit data in such a way that other devices on the same local area network segment can interpret, receive, and process it. An Ethernet cable is a physical connection between two computers.

Ethernet is popular because it's readily scalable, meaning that it's comparatively easy to integrate new technologies, such as *Fast Ethernet* and *Gigabit Ethernet*, into an existing network infrastructure. It's also relatively simple to implement in the first place, and with it, troubleshooting is reasonably straightforward.

Ethernet uses both **Data Link** and **Physical layer** specifications.

Ethernet is not a single protocol but an entire collection of different standards. These standards come from the IEEE and all of them start with **802.3** in their name. Ethernet is also pretty old, the first memo about Ethernet was written by Bob Metcalfe back in 1973.

Despite its age, Ethernet is the dominant choice for LANs. There are many different standards with speeds of 10 Mbps (megabits per second) up to 100 Gbps (gigabits per second). Here's an overview with some popular Ethernet standards:

Bandwidth	Common Name	Informal name	IEEE name	Cable Type
10 Mbps	Ethernet	10BASE-T	802.3	UTP 100m
100 Mbps	Fast Ethernet	100BASE-T	802.3u	UTP 100m
1000 Mbps	Gigabit Ethernet	1000BASE-LX	802.3z	Fiber 5000m
1000 Mbps	Gigabit Ethernet	1000BASE-T	802.3ab	UTP 100m
10 Gbps	10 Gigabit Ethernet	10GBASE-T	802.3an	UTP 100m

On the physical layer, there are different cable options and different speeds. One of the advantages of Ethernet, however, is that it uses the

same data link layer standard. You can mix different Ethernet standards in your network.

> Q: What is ethernet?
> A: Ethernet is a network technology used in LAN and WAN that connects devices using cables for the transmission of data. It provides services on the Physical and Data Link layers of the OSI Model.
>
> - Interview Q&A

Previous

Next

You have completed 0% of the lesson

0%

Jump to...

Network Basics

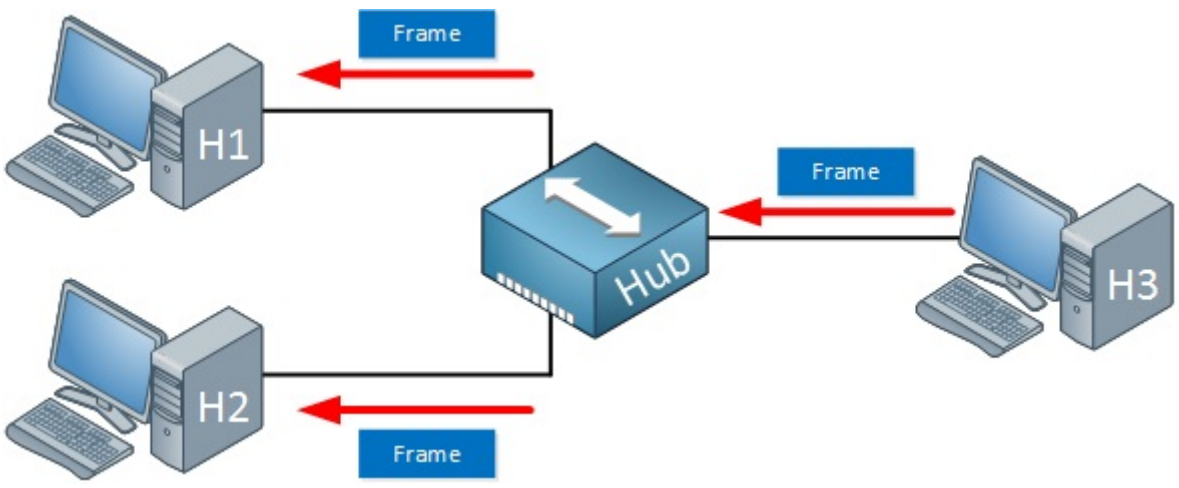
Ethernet Basics

Collision Domain

The term **collision domain** is an Ethernet term that refers to a particular network scenario wherein one device sends a packet out on a network segment and thereby forces every other device on that same physical network segment to pay attention to it. This is bad because if two devices on one physical segment transmit at the same time, a collision event—a situation where each device's digital signals interfere with another on the wire—occurs and forces the devices to retransmit later. Collisions have a dramatically negative effect on network performance.

The situation described is typically found in a **hub** environment where each host segment connects to a hub that represents only one collision domain and one broadcast domain.

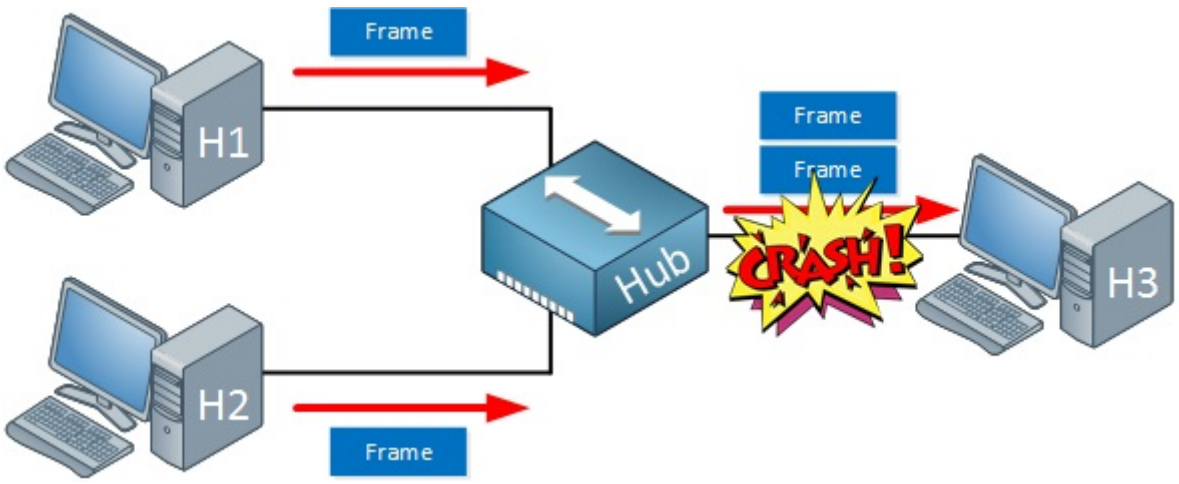
Hubs will be explained in detail in the further lessons but in order to understand the collision domain here is a brief definition of a hub: The hub is a simple device; it's basically nothing more but a repeater. When it receives an electrical signal, it will repeat it on all ports except the one where it received the signal on. This logic works fine in the following situation:



Above we see that H3 sends an Ethernet frame. Let's say that this frame is destined for H1. When the hub receives this frame, it will replicate it on the ports towards H1 and H2.

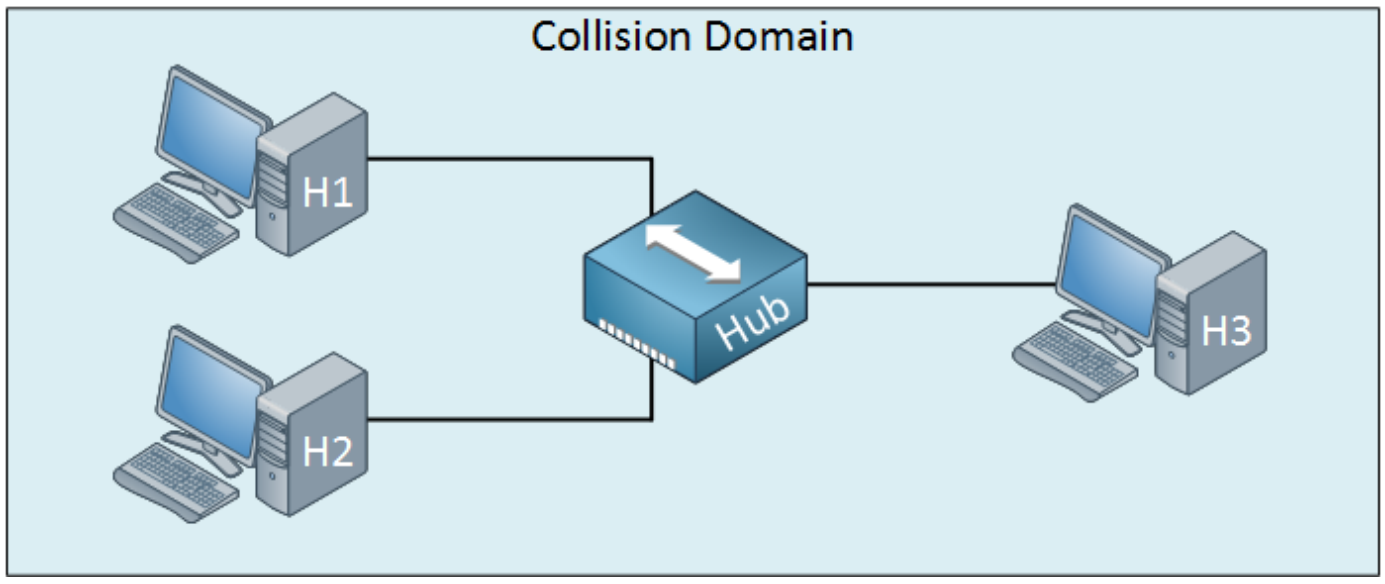
H1 wants to receive it, H2 doesn't care about so it will discard the frame. No problem! Our goal to send a frame from H3 to H1 is accomplished.

We do have a problem when H1 and H2 both send a frame at the same time, like the following situation:



When H1 and H2 send a frame at the same time, our hub will replicate them on the port that is connected to H3. In this case, a collision will occur and H3 will receive nothing.

Don't forget, we can get **collisions** on every port that is connected to the **hub**, this all belongs to the same collision domain.



In further lessons, we will cover how to minimize/avoid the collision domain.

Previous

Next

You have completed 14% of the lesson

14%

Jump to...

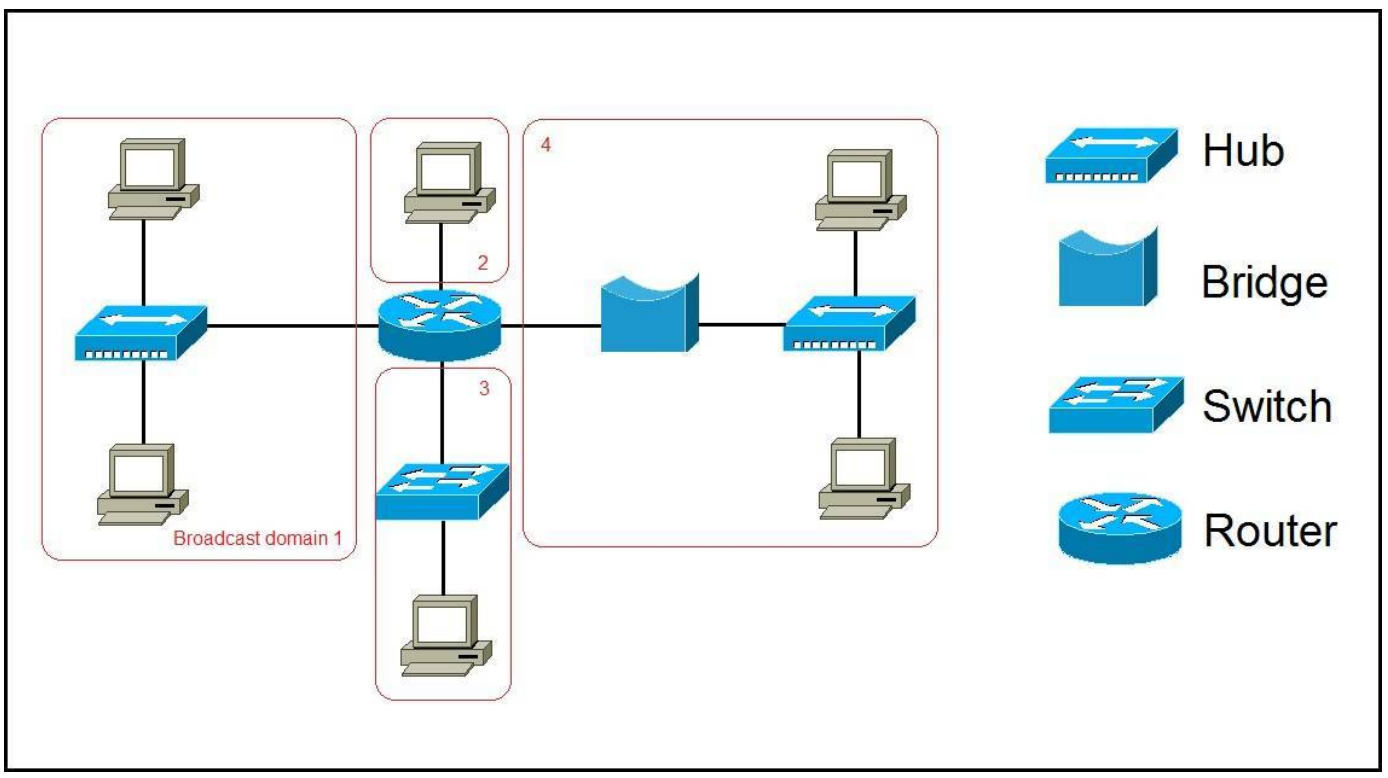
Network Basics

Ethernet Basics

Broadcast Domain

The term **broadcast domain** is used to describe a group of devices on a specific network segment that can reach each other with Ethernet broadcasts. Broadcasts sent by a device in one broadcast domain are not forwarded to devices in another broadcast domain. This improves the performance of the network because not all devices on a network will receive and process broadcasts.

Routers separate a LAN into multiple broadcast domains (every port on a router is in a different broadcast domain). Switches (by default) flood Ethernet broadcast frames out all ports, just like bridges and hubs. All ports on these devices are in the same broadcast domain. To better understand the concept of broadcast domains, consider the following example:



In the picture above we have a network of six computers, two hubs, a bridge, a switch, and a router. The broadcast domains are marked in red. Remember, all devices connected to a hub, a bridge, and a switch are in the same broadcast domain. Notice that all hosts can communicate with each other by Data Link layer (hardware address) broadcast. Only routers separate the LAN into multiple broadcast domains. That is why we have four broadcast domains in the network pictured above.

Previous Next

Jump to...

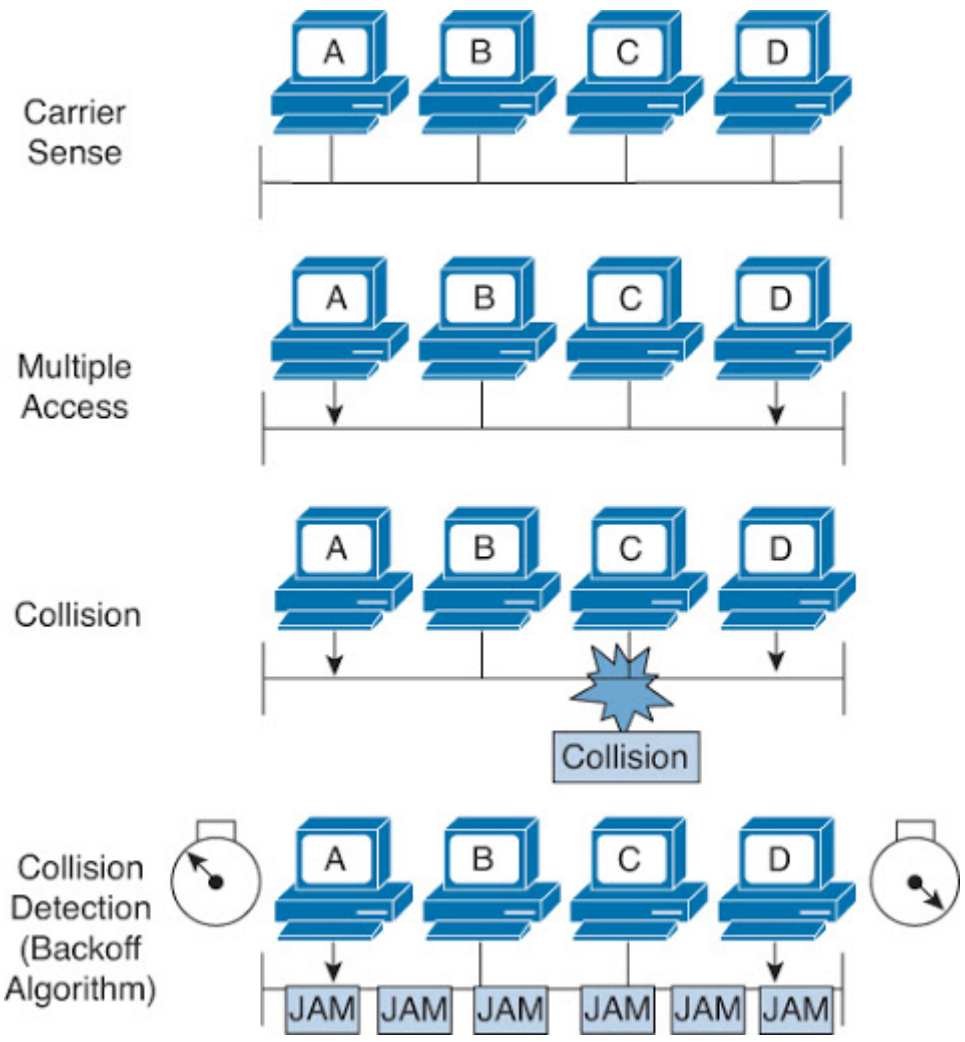
Network Basics

Ethernet Basics

CSMA/CD

Ethernet networking uses **Carrier Sense Multiple Access with Collision Detection (CSMA/CD)**, a media access control method that helps devices share the bandwidth evenly without having two devices transmit at the same time on the network medium. **CSMA/CD** was created to overcome the problem of those collisions that occur when packets are transmitted simultaneously from different hosts. Good collision management is crucial because when a host transmits in a CSMA/CD network, all the other hosts on the network receive and examine that transmission. Only **bridges, switches**, and **routers**, but not `hubs`, can effectively prevent a transmission from propagating throughout the entire network.

So, how does the CSMA/CD protocol work? Let's start by taking a look at the below figure, where a collision has occurred in the network.



When a host wants to transmit over the network, it first checks for the presence of a digital signal on the wire. If all is clear, meaning that no other host is transmitting, the host will then proceed with its transmission. But it doesn't stop there. The transmitting host constantly monitors the wire to make sure no other hosts begin transmitting. If the host detects another signal on the wire, it sends out an extended **jam signal** that causes all hosts on the segment to stop sending data (think busy signal). The hosts respond to that jam signal by waiting a while before attempting to transmit again. Backoff algorithms, represented by the clocks counting down on either

side of the jammed devices, determine when the colliding stations can retransmit. If collisions keep occurring after 15 tries, the hosts attempting to transmit will then time out.

When a collision occurs on an Ethernet LAN, the following things happen:

- A jam signal informs all devices that a collision occurred.
- The collision invokes a random backoff algorithm.
- Each device on the Ethernet segment stops transmitting for a short time until the timers expire.
- All hosts have equal priority to transmit after the timers have expired.

And following are the effects of having a CSMA/CD network that has sustained heavy collisions:

- Delay
- Low throughput
- Congestion

Previous

Next

Network Basics

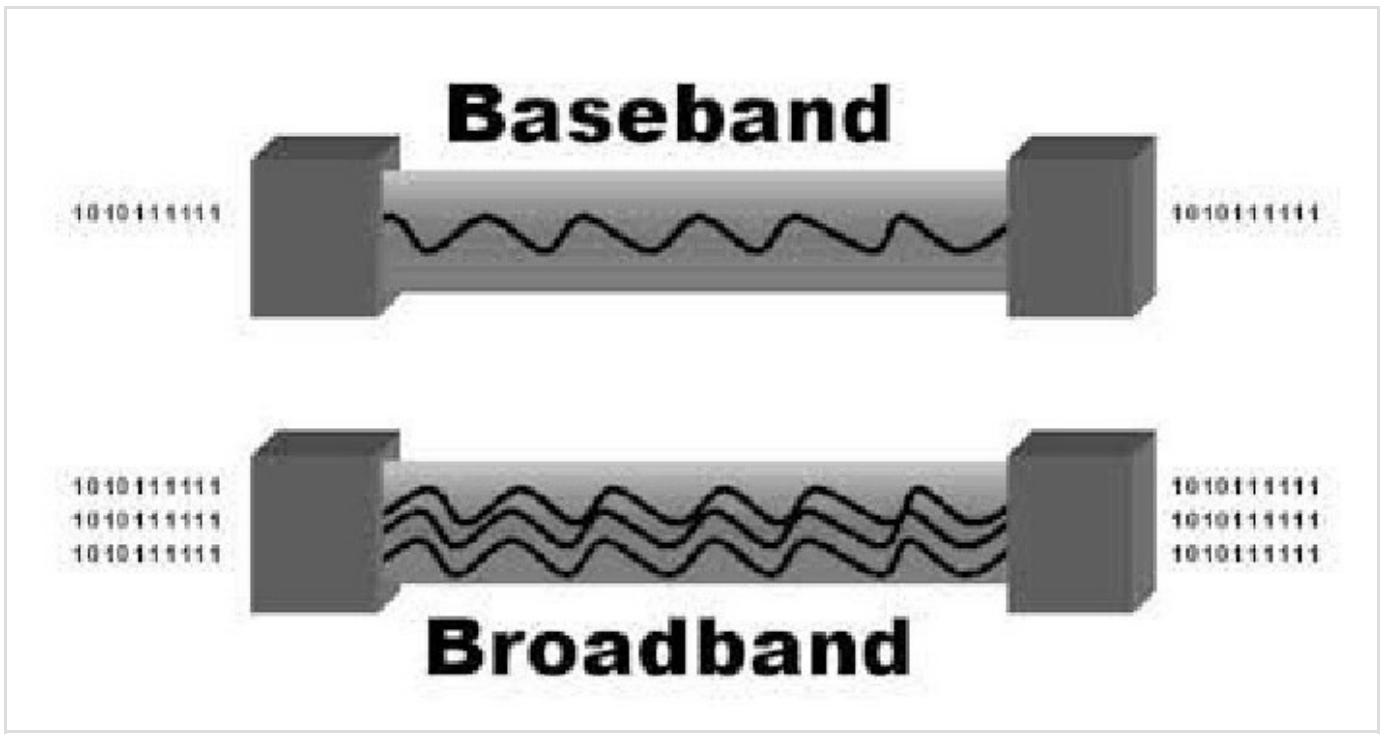
Ethernet Basics

Broadband/Baseband

We have two ways to send analog and digital signals down a wire: **broadband** and **baseband**.

Broadband allows us to have both our analog voice and digital data carried on the same network cable or physical medium. Broadband allows us to send *multiple frequencies of different signals* down the same wire at the same time (called *frequency-division multiplexing*) and to send both analog and digital signals.

Baseband is what all LANs use. This is where all the bandwidth of the physical media is used by *only one signal*. For example, Ethernet uses only one digital signal at a time, and it requires all the available bandwidth. If multiple signals are sent from different hosts at the same time, we get collisions; the same with wireless, except that, use only analog signaling.



Baseband vs. Broadband

Previous Next

You have completed 56% of the lesson

56%

Jump to...

Network Basics

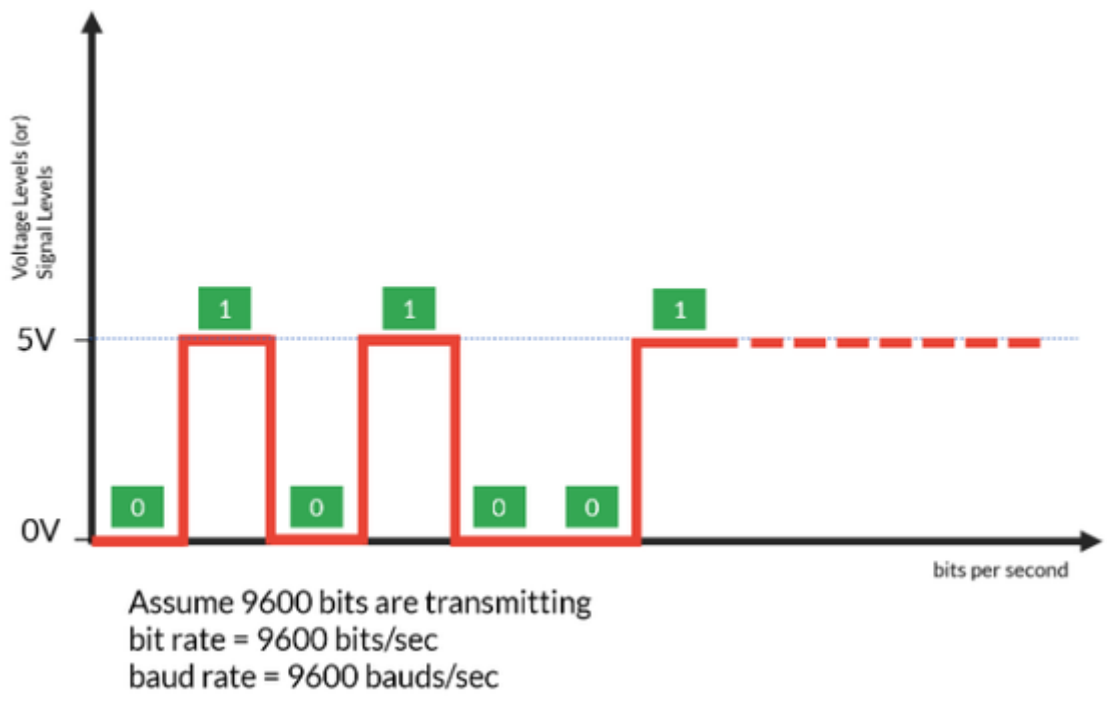
Ethernet Basics

Bit Rates vs. Baud Rate

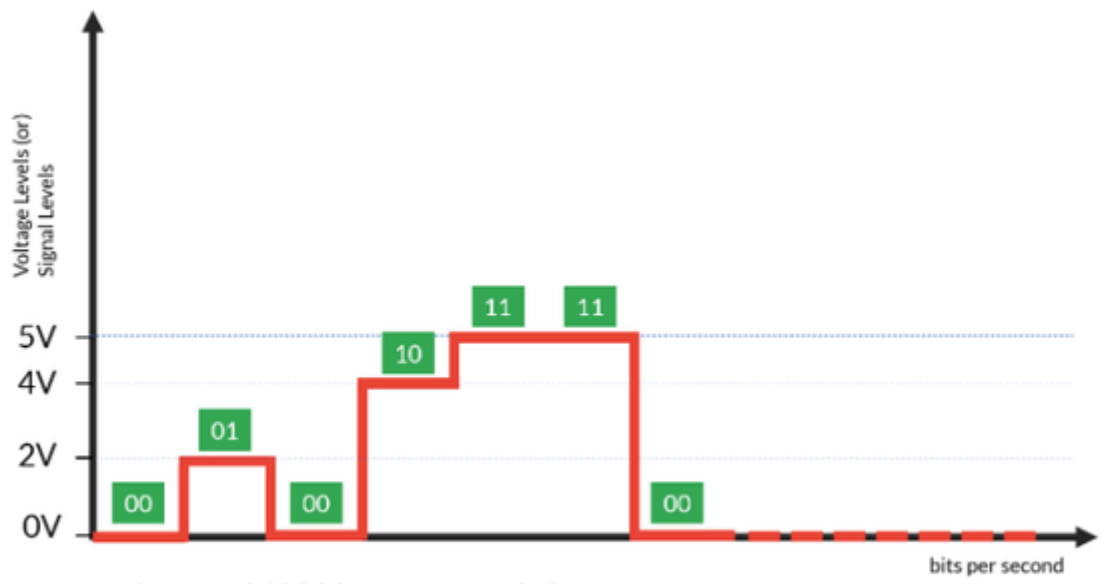
Bit rate is a measure of the number of data bits (0s and 1s) transmitted in one second in either a digital or analog signal. A figure of 56,000 bits per second (bps) means 56,000 0s or 1s can be transmitted in one second, which we simply refer to as bps.

In the 1970s and 1980s, we used the term **baud rate** a lot, but that was replaced by *bps* because it was more accurate. Baud was a term of measurement named after a French engineer, Jean-Maurice-Émile Baudot because he used it to measure the speed of telegraph transmissions.

One baud is one electronic state change per second—for example, from 0.2 volts to 3 volts or from binary 0 to 1. However, since a single state change can involve more than a single bit of data, the bps unit of measurement has replaced it as a more accurate definition of how much data you're transmitting or receiving.



Assume 9600 bits are transmitting
bit rate = 9600 bits/sec
baud rate = 9600 bauds/sec



Assume 9600 bits are transmitting
bit rate = 9600 bits/sec
baud rate = 4800 bauds/sec

In the figure, the number of data elements carried by each signal element in the bottom one is double of upper one, that is why the baud rate in the bottom one is half of the upper one.

Previous

Next

You have completed 71% of the lesson

71%

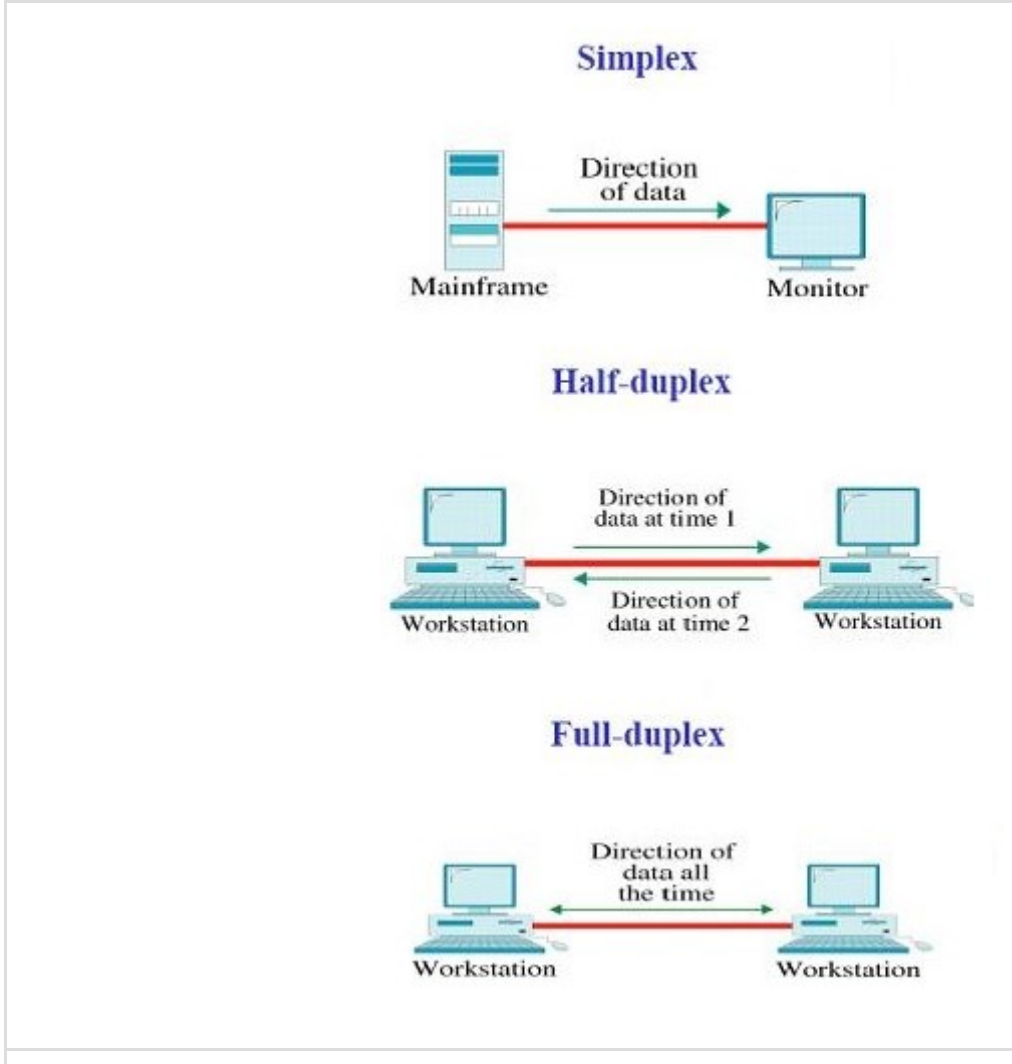
Jump to...

Network Basics

Ethernet Basics

Half- and Full-Duplex Ethernet

Half-duplex Ethernet is defined in the original *802.3 Ethernet specification*. Basically, when you run half-duplex, you're using only one wire pair with a digital signal either transmitting or receiving. This really isn't all that different from full-duplex because you can both transmit and receive—you just can't run half-duplex and full-duplex at the same time.



Ethernet communication

Half-duplex Ethernet—typically 10BaseT—is only about 30 to 40 percent efficient because it will usually provide only 3 Mbps to 4 Mbps at most. Although it's true that 100 Mbps Ethernet can and sometimes do run half-duplex, it's just not very common to find that happening anymore.

In contrast, full-duplex Ethernet uses two pairs of wires at the same time instead of one measly wire pair like half duplex employs. Plus, full duplex uses a point-to-point connection between the transmitter of the sending device and the receiver of the receiving device (in most cases the switch). This means that with full-duplex data transfer, you not only get faster data-transfer speeds, but you also get collision prevention too. Full-duplex Ethernet is supposed to offer 100 percent efficiency in both directions—for example, you can get 20 Mbps with a 10 Mbps Ethernet running full-duplex or 200 Mbps for Fast Ethernet. But this rate is something known as an **aggregate rate**, which translates as "you're supposed to get" 100 percent

efficiency.

Full-duplex Ethernet can be used in many situations; here are some examples:

- With a connection from a switch to a host
- With a connection from a switch to a switch
- With a connection from a host to a host

> **Note:** You can run full duplex with just about any device except a **hub**.

Previous

Next

You have completed 86% of the lesson

86%

Jump to...

Network Basics

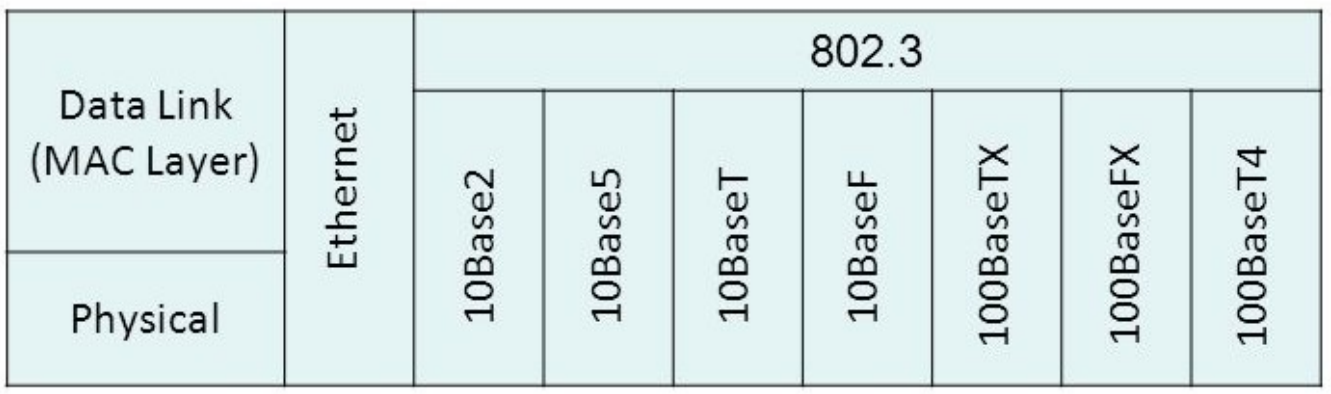
Ethernet at the Physical Layer

Ethernet at the Physical Layer

Ethernet was first implemented by a group called DIX (Digital, Intel, and Xerox). They created and implemented the first Ethernet LAN specification, which the IEEE used to create the IEEE 802.3 Committee. This was a 10 Mbps network that ran on coax, then on twisted-pair, and finally on fiber physical media.

The IEEE extended the 802.3 Committee to two new committees known as *802.3u (Fast Ethernet)*, *802.3ab (Gigabit Ethernet on Category 5+)*, and then finally to *802.3ae (10 Gbps over fiber and coax)*.

The below figure shows the IEEE 802.3 and the original Ethernet Physical layer specifications.



When designing your LAN, it's really important to understand the different types of Ethernet media available to you. Sure, it would be great to run *Gigabit Ethernet* to each desktop and *10 Gbps* between *switches*, as well as to *servers*. Although this is just starting to happen, justifying the cost of that network today for most companies would be a pretty hard sell. But if you mix and match the different types of Ethernet media methods currently available instead, you can come up with a *cost-effective* network solution that works great!

The Electronic Industries Association and the newer *Telecommunications Industry Alliance (EIA/TIA)* together form the standards body that creates the **Physical layer specifications** for Ethernet. The EIA/TIA specifies that Ethernet uses a registered jack (RJ) connector on unshielded twisted-pair (UTP) cabling (RJ-45). However, the industry is calling this just an 8-pin modular connector.

Each Ethernet cable type that is specified by the EIA/TIA has something known as inherent attenuation, which is defined as the loss of signal strength as it travels the length of a cable and is measured in decibels (dB). A higher-quality cable will have a higher-rated category and lower attenuation. For example, *Category 5* is better than *Category 3* because

Category 5 cables have more wire twists per foot and therefore *less crosstalk*. Crosstalk is the unwanted signal interference from adjacent pairs in the cable.

Here are the original IEEE 802.3 standards:

Ethernet Name	Cable Type	Maximum Speed	Maximum Transmission Distance	Notes
10Base5	Coax	10 Mbps	500 meters per segment	Also called thicknet, this cable type uses vampire taps to connect devices to cable.
10Base2	Coax	10 Mbps	185 meters per segment	Also called thinnet, a very popular implementatic of Ethernet over coax.
10BaseT	UTP	10 Mbps	100 meters per segment	One of the most popular network cabling schemes.
100BaseTX	UTP, STP	100 Mbps	100 meters per segment	Two pairs of Category 5 UTP.
10BaseFL	Fiber	10 Mbps	Varies (ranges from 500 meters to 2,000 meters)	Ethernet over fiber optics to the desktop.
100BaseFX	MMF	100 Mbps	2,000 meters	100 Mbps Ethernet over fiber optics.
1000BaseT	UTP	1000 Mbps	100 meters	Four pairs of Category 5 or higher.
1000BaseTX	UTP	1000 Mbps	100 meters	Two pairs of Category 6 or higher.

Ethernet Name	Cable Type	Maximum Speed	Maximum Transmission Distance	Notes
1000BaseSX	MMF	1000 Mbps	550 meters	Uses SC fiber connectors. Max length depends on fiber size.
1000BaseCX	Balanced, shielded copper	1000 Mbps	25 meters	Uses a specia connector, the HSSDC.
1000BaseLX	MMF and SMF	1000 Mbps	550 meters multimode/2,000 meters single mode	Uses longer wavelength laser than 1000BaseSX. Uses SC and LC connectors
10GBaseT	UTP	10 Gbps	100 meters	Connects to the network like a Fast Ethernet link using UTP.
10GBaseSR	MMF	10 Gbps	400 meters	850 nm laser. Max length depends on fiber size and quality.
10GBaseLR	SMF	10 Gbps	10 kilometers	1,310 nm laser Max length depends on fiber size and quality.
10GBaseER	SMF	10 Gbps	40 kilometers	1,550 nm lase Max length depends on fiber size and quality.
10GBaseSW	MMF	10 Gbps	400 meters	850 nm laser transceiver. 10GBaseLW SMF 10 Gbps 10 kilometers Typically used with SONET.

Ethernet Name	Cable Type	Maximum Speed	Maximum Transmission Distance	Notes
10GBaseEW	SMF	10 Gbps	40 kilometers	1,550 nm optical wavelength.

Previous

End of Lesson

You have completed 0% of the lesson

0%

Jump to...

Network Basics

Ethernet at the Data Link Layer

Binary to Decimal and Hexadecimal Conversion

Understanding the differences between *binary, decimal*, and *hexadecimal* numbers and how to convert one format into the other is very important before discussing the **TCP/IP protocol stack** and **IP addressing** in further. So let's get started with binary numbering. Each digit used is limited to being either a 1 (one) or a 0 (zero), and each digit is called 1 bit (short for binary digit). Typically, you count either 4 or 8 bits together, with these being referred to as a **nibble** and a **byte**, respectively.

The binary numbers are placed in a value spot, starting at the right and moving left, with each spot having double the value of the previous spot. The below table shows the decimal values of each bit location in a nibble and a byte. Remember, a nibble is four bits and a byte is eight bits. In network addressing, we often refer to a byte as an **octet**.

Nibble Values	Byte Values
8 4 2 1	128 64 32 16 8 4 2 1

What all this means is that if one digit (1) is placed in a value spot, then the nibble or byte takes on that decimal value and adds it to any other value spots that have a 1. And if zero (0) is placed in a bit spot, you don't count that value. For example, if we have a 1 placed in each spot of our nibble, we then add up `8 + 4 + 2 + 1` to give us a maximum value of 15. Another example of our nibble values is `1010`, which means that the 8 bit and the 2 bit are turned on and equal a decimal value of `10`. If we have a nibble binary value of `0110`, then our decimal value is `6` because the 4 and 2 bits are turned on.

But the byte values can add up to a value that's significantly higher than 15. This is how—if we count every bit as a one (1), then the byte binary value looks like this (remember, 8 bits equal a byte):

```
11111111
```

We then count up every bit spot because each is turned on. It looks like this, which demonstrates the maximum value of a byte:

```
128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255
```

It is strongly advised that you memorize the below table before braving the IP sections in further lessons since this lists all available subnet masks.

Binary Value Binary Value	Decimal Value Decimal Value

10000000	128
11000000	192
11100000	224
11110000	240
11111000	248
11111100	252
11111110	254
11111111	255

Hexadecimal addressing is completely different than binary or decimal—it's converted by reading nibbles, not bytes. By using a nibble, we can convert these bits to hex pretty simply. First, understand that the hexadecimal addressing scheme uses only the numbers 0 through 9. And because the numbers 10, 11, 12, and so on can't be used (because they are two-digit numbers), the letters A, B, C, D, E, and F are used to represent 10, 11, 12, 13, 14, and 15, respectively.

Hexadecimal Value	Binary Value	Decimal Value
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12

Hexadecimal Value	Binary Value	Decimal Value
D	1101	13
E	1110	14
F	1111	15

Suppose you have something like this: `0x6A`. (Some manufacturers put `0x` in front of characters so you know that they're a hex value, while others just give you an `h`. It doesn't have any other special meaning.) What are binary and decimal values? To correctly answer that question, all you have to remember is that each *hex* character is *one nibble* and *two hex* characters together make a *byte*. To figure out the binary value, first put the hex characters into two nibbles and then put them together into a byte. `6 = 0110` and `A (which is 10 in decimal) = 1010`, so the complete byte is `01101010`.

To convert from binary to hex, just take the byte and break it into nibbles. Here's how you do that: Say you have the binary number `01010101`. First, break it into nibbles—`0101` and `0101`—with the value of each nibble being 5 because the 1 and 4 bits are on. This makes the hex answer `0x55`. And in decimal format, the binary number is `01010101`, which converts to 64 + 16 + 4 + 1 = 85.

Previous

Next

Network Basics

Ethernet at the Data Link Layer

Ethernet Addressing

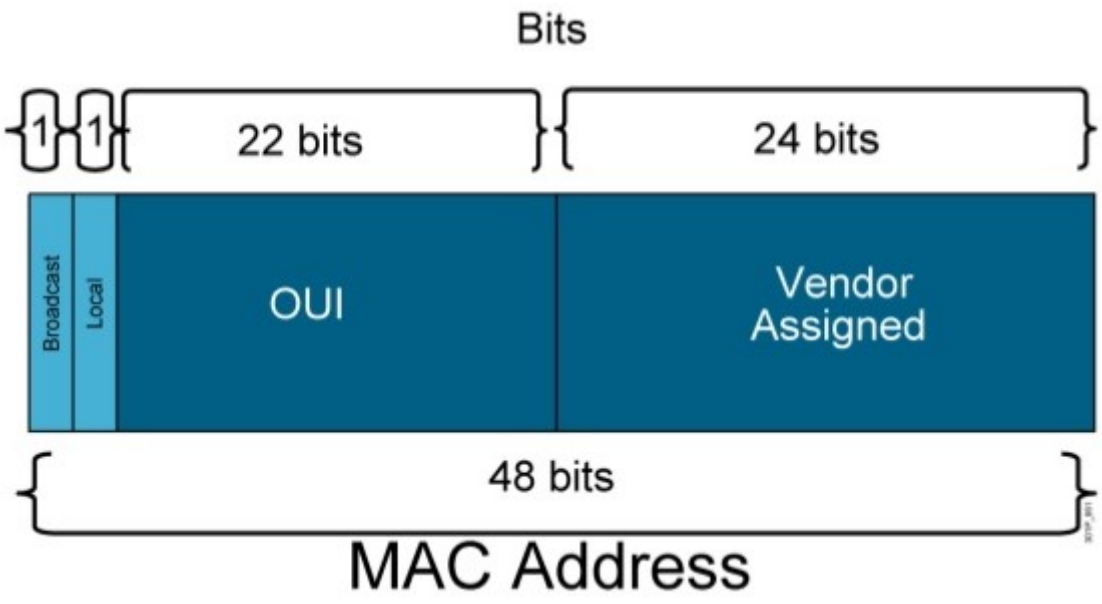
In order to communicate or transfer the data from one computer to another computer, we need some addresses. In Computer Network various types of addresses are introduced; each works at a different layer. **Ethernet addressing** uses the **Media Access Control (MAC)** address burned into each and every *Ethernet NIC*. **MAC Address** is a physical address which works at **Data Link Layer**.

MAC Addresses are unique 48-bits hardware number of a computer, which is embedded into the network card (known as Network Interface Card) during the time of manufacturing. MAC Address is also known as the Physical Address of a network device. In IEEE 802 standard, Data Link Layer is divided into two sublayers:

- Logical Link Control (LLC) Sublayer
- Media Access Control (MAC) Sublayer

MAC address is used by the **Media Access Control (MAC)** sublayer of the **Data Link Layer**. MAC Address is world wide unique since millions of network devices exist and we need to uniquely identify each.

MAC Address is a 12-digit hexadecimal number (6-Byte binary number), which is mostly represented by Colon-Hexadecimal notation. First 6-digits (say 00:40:96) of MAC Address identifies the manufacturer, called OUI (Organizational Unique Identifier). IEEE Registration Authority Committee assigns these MAC prefixes to its registered vendors. Some firms may have more than one MAC address.



MAC Address

Here are some OUI of well-known manufacturers:

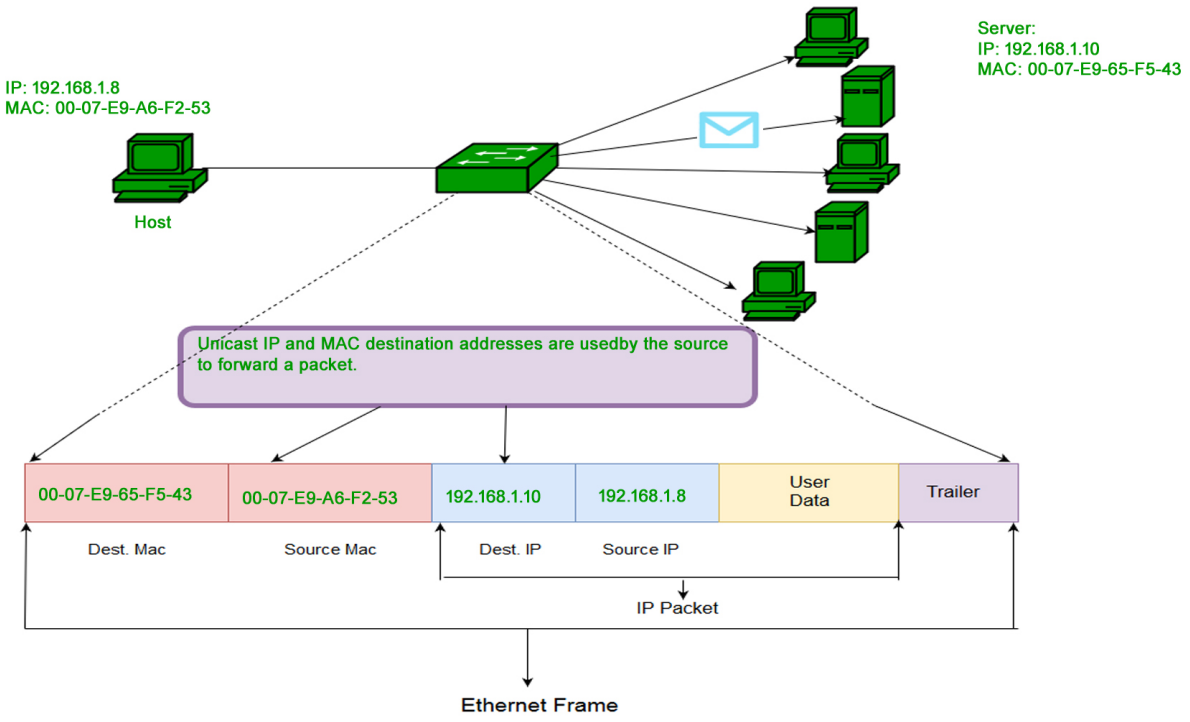
```
CC:46:D6 - Cisco
3C:5A:B4 - Google, Inc.
3C:D9:2B - Hewlett Packard
00:9A:CD - HUAWEI TECHNOLOGIES CO., LTD
```

The rightmost six digits represent the Network Interface Controller, which is assigned by the manufacturer.

Types of MAC Addresses

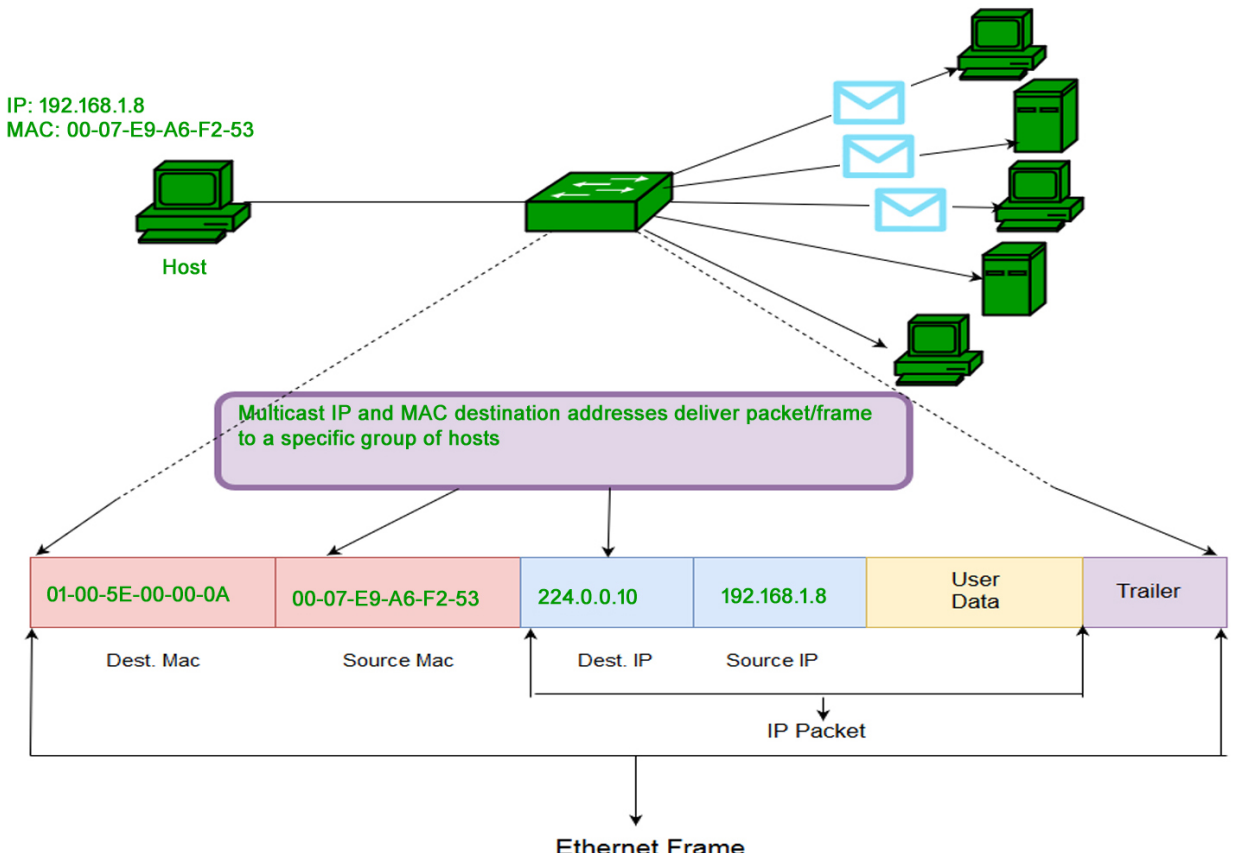
1. Unicast

A Unicast addressed frame is only sent out to the interface leading to specific NIC. If the LSB (least significant bit) of the first octet of an address is set to zero, the frame is meant to reach only one receiving NIC. MAC Address of source machine is always Unicast.



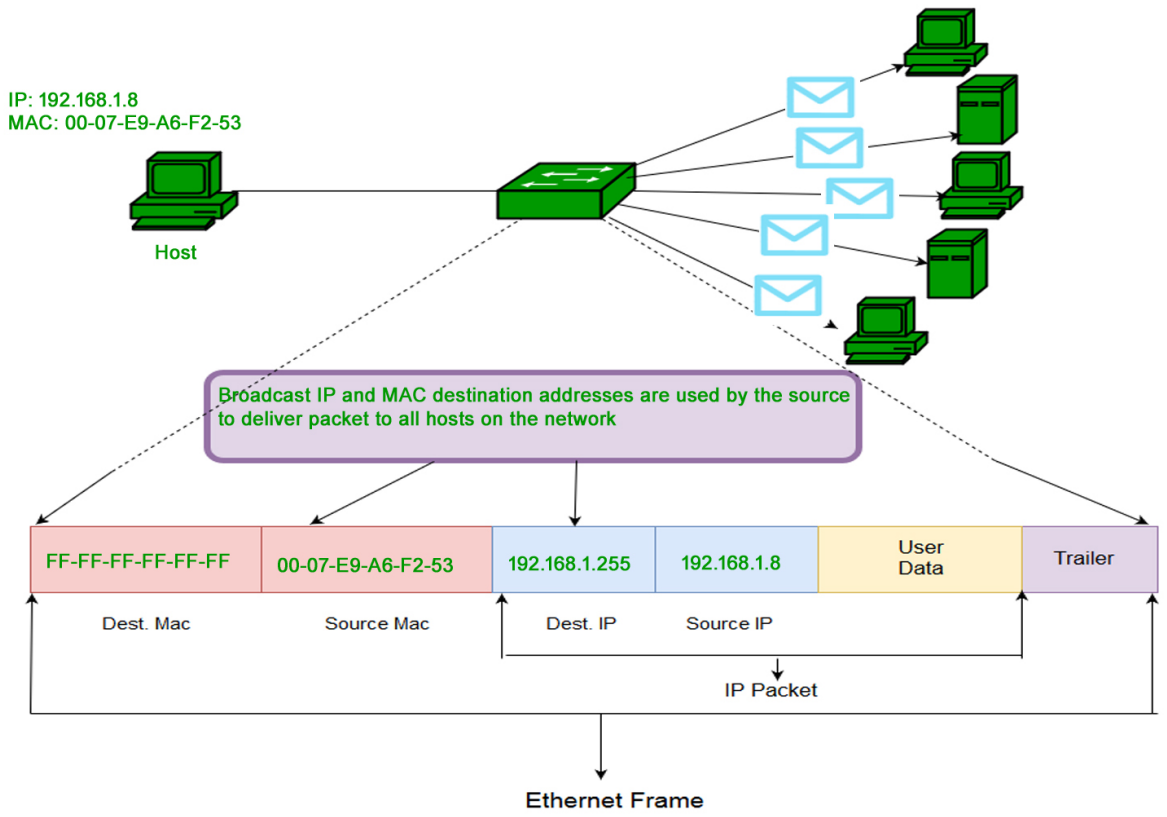
2. Multicast

The multicast address allows the source to send a frame to a group of devices. In Layer-2 (Ethernet) Multicast address, LSB (least significant bit) of the first octet of an address is set to one. IEEE has allocated the address block 01-80-C2-xx-xx-xx (01-80-C2-00-00-00 to 01-80-C2-FF-FF-FF) for group addresses for use by standard protocols.



3. Broadcast

Similar to Network Layer, Broadcast is also possible on the underlying layer (Data Link Layer). Ethernet frames with ones in all bits of the destination address (FF-FF-FF-FF-FF-FF) are referred to as the broadcast address. Frames which are destined with MAC address FF-FF-FF-FF-FF-FF will reach to every computer belong to that LAN segment.



Previous

Next

You have completed 25% of the lesson

25%

Jump to...

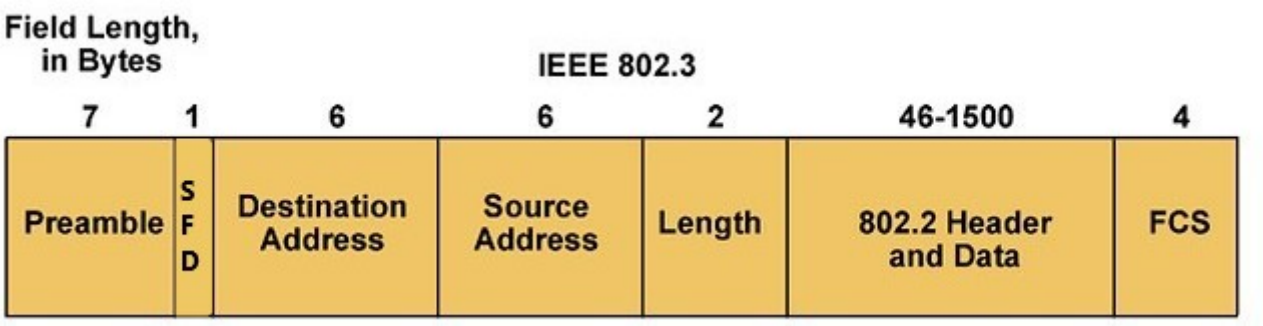
Network Basics

Ethernet at the Data Link Layer

Ethernet Frames

When transmitting data over Ethernet, the **Ethernet frame** is primarily responsible for the correct rulemaking and successful transmission of data packets. Essentially, **data** sent over Ethernet is **carried by the frame**. An Ethernet frame is between 64 bytes and 1,518 bytes big, depending on the size of the data to be transported.

The Ethernet frame structure is defined in the **IEEE 802.3** standard. Here is a graphical representation of an Ethernet frame and a description of each field in the frame:



The Ethernet frame

- **PREAMBLE** – Ethernet frame starts with 7-Bytes Preamble. This is a pattern of alternative 0's and 1's which indicates the starting of the frame and allows the sender and receiver to establish bit synchronization. Initially, **PRE (Preamble)** was introduced to allow for the loss of a few bits due to signal delays. But today's high-speed Ethernet doesn't need Preamble to protect the frame bits. **PRE (Preamble)** indicates the receiver that frame is coming and allow the receiver to lock onto the data stream before the actual frame begins.
- **Start of frame delimiter (SFD)** – This is a 1-Byte field which is always set to 10101011. SFD indicates that upcoming bits are starting of the frame, which is the destination address. Sometimes SFD is considered the part of PRE, this is the reason Preamble is described as 8 Bytes in many places. The SFD warns station or stations that this is the last chance for synchronization.
- **Destination MAC Address** – This is a 6-Byte field which contains the MAC address of the machine for which data is destined.
- **Source MAC Address** – This is a 6-Byte field which contains the MAC address of source machine. As Source Address is always an individual address (Unicast), the least significant bit of the first byte is always 0.
- **Length/Type** – Length is a 2-Byte field, which indicates the length of the entire Ethernet frame. This 16-bit field can hold the length value between 0 to 65534, but length cannot be larger than 1500 because of some own limitations of Ethernet.

- **Data** – This is the place where actual data is inserted, also known as Payload. Both IP header and data will be inserted here if the Internet Protocol is used over Ethernet. The maximum data present may be as long as 1500 Bytes. In case data length is less than minimum length i.e. 46 bytes, then padding 0's is added to meet the minimum possible length.
- **Frame Check Sequence (FCS)/Cyclic Redundancy Check (CRC)** – CRC is 4 Byte field. This field contains a 32-bits hash code of data, which is generated over the Destination Address, Source Address, Length, and Data field. If the checksum computed by destination is not the same as sent checksum value, data received is corrupted. But remember—this is error detection, not error correction.

Previous

Next

You have completed 50% of the lesson

50%

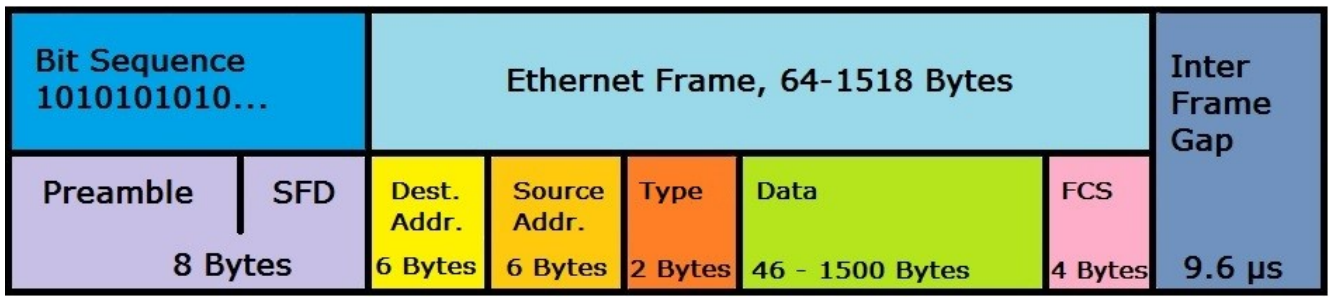
Jump to...

Network Basics

Ethernet at the Data Link Layer

Ethernet II Frames

Standard IEEE 802.3 basic frame format is discussed in detail in the preceding section. Now let's see the extended Ethernet frame (also called Ethernet II Frame) header:



The Ethernet II frame

An **Ethernet II frame** must be at least 64 bytes for collision detection to work, and can be a maximum of 1,518 bytes. The packet starts with a **preamble** that controls the synchronization between the sender and receiver and a **"Start Frame Delimiter (SFD)"** that defines the frame. Both values are bit sequences in the format "10101010 ..." in which the actual frame contains information about the source and destination addresses (MAC format), control information (in the case of Ethernet II the type field, later a length specification), followed by the transmitted data record. A **frame check sequence (FCS)** is an error-detecting code that closes the frame (except for the preamble and SFD). The packet is completed by an **"InterFrame Gap"**, which defines a 9.6 µs transmission pause.

Ethernet II uses the classic frame structure with a type field ("Type") which defines various protocols of the network layer. In the OSI model, the network layer is important for connecting and providing network addresses. The type field was replaced by a length specification in later frame formats.

Tip:

- In the type field, Ethernet II determines which switching protocols are used. This is important for segmenting the data stream and preventing data congestion.

The Ethernet II frame was defined in 1982 and has formed the foundation of all subsequent frame developments.

You have completed 75% of the lesson

75%

Jump to...