Ethernet Basics

Ethernet

Ethernet is the traditional technology for connecting devices in a LAN or WAN, enabling them to communicate with each other via a protocol -- a set of rules or common network language. Ethernet describes how network devices can format and transmit data so other devices on the same local area network segment can recognize, receive, and process the information. An Ethernet cable is the physical, encased wiring over which the data travels.

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.

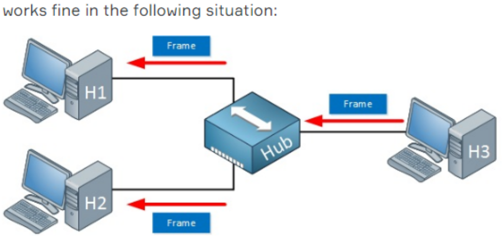
> Q: What is ethernet?
> A: Ethernet is a network technology used in LAN and WAN that connects devices using cables for the transmission of data. It provides services on the Physical and Data Link layers of the OSI Model.
>
> - Interview Q&A

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.

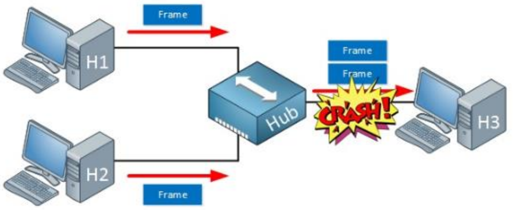
Hubs will be explained in detail in the further lessons but in order to understand the collision domain here is a brief definition of a hub: The hub is a simple device; it's basically nothing more but a repeater. When it receives an electrical signal, it will repeat it on all ports except the one where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

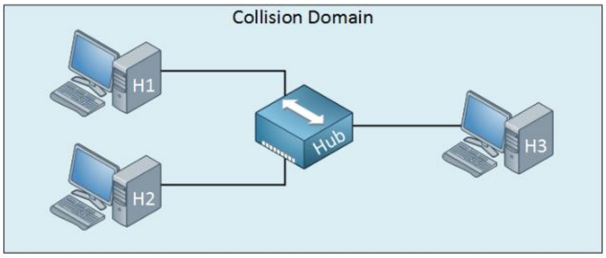
H1 wants to receive it, H2 doesn't care about so it will discard the frame. No problem! Our goal to send a frame from H3 to H1 is accomplished.

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.

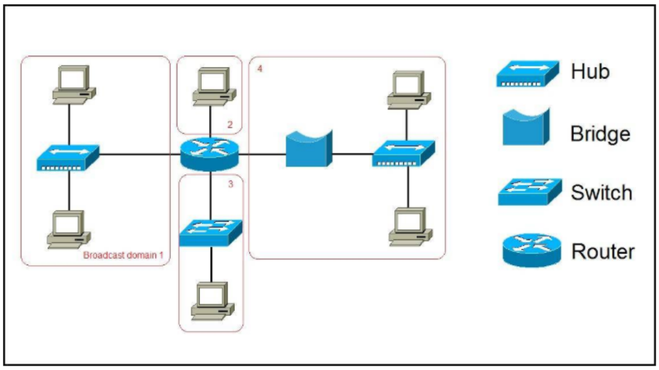
Don't forget, we can get **collisions** on every port that is connected to the **hub**, this all belongs to the same collision domain.



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:

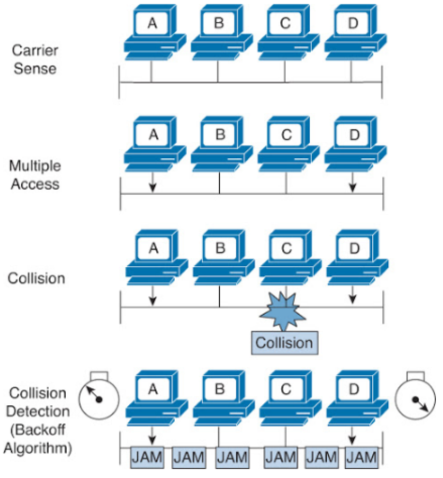


In the picture above we have a network of six computers, two hubs, a bridge, a switch, and a router. The broadcast domains are marked in red. Remember, all devices connected to a hub, a bridge, and a switch are in the same broadcast domain. Notice that all hosts can communicate with each other by Data 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.

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:

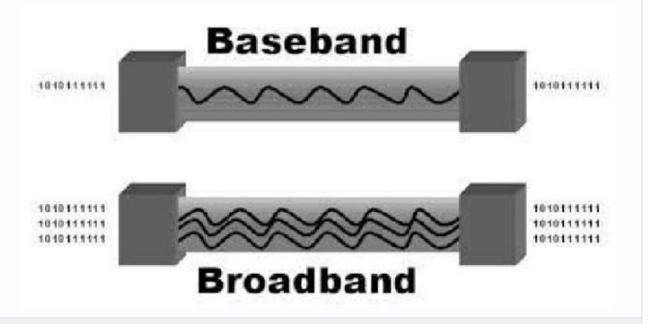
- Delay
- Low throughput
- Congestion

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.



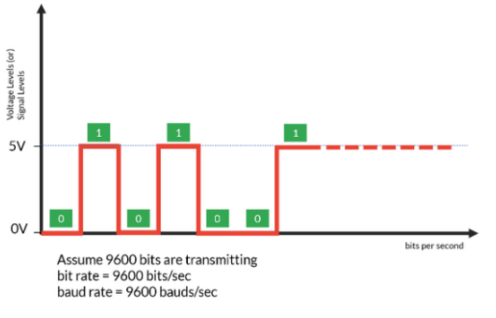
Baseband vs. Broadband

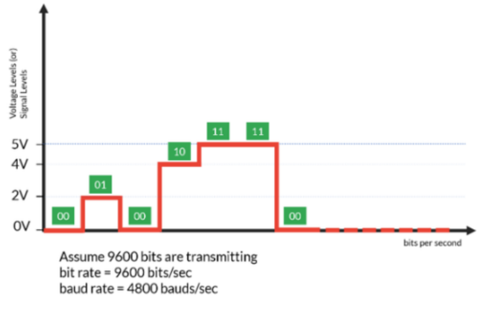
Bit Rates vs. Baud Rate

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.

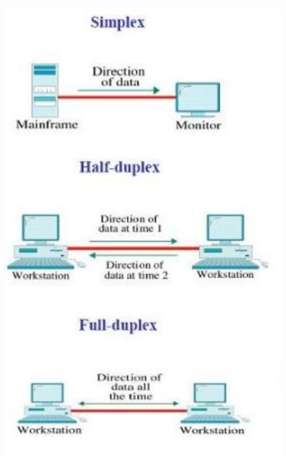




In the figure, the number of data elements carried by each signal element in the bottom one is double of upper one, that is why the baud rate in the bottom one is half of the upper one.

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

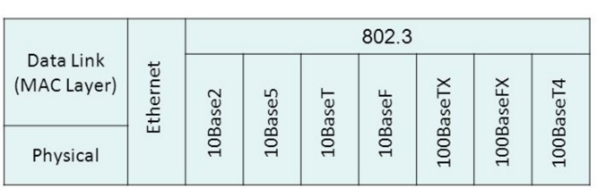
Note: You can run full duplex with just about any device except a **hub**.

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!

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 implementation 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.
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 special 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 laser. 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.
10GBaseEW	SMF	10 Gbps	40 kilometers	1,550 nm optical wavelength.