# Differences Between A and CNAME Records

The A and CNAME records are the two common ways to map a host name (“name”) to one or more IP addresses. There are important differences between these two records.

The [A record](https://support.dnsimple.com/articles/a-record) points a name to a specific IP. If you want blog.dnsimple.com to point to the server 185.31.17.133 you’ll configure:

blog.dnsimple.com. A 185.31.17.133

The [CNAME record](https://support.dnsimple.com/articles/cname-record) points a name to another name instead of to an IP. The CNAME source represents an alias for the target name and inherits its entire resolution chain.

Let’s use [our blog](https://blog.dnsimple.com/) as an example:

blog.dnsimple.com. CNAME aetrion.github.io.

aetrion.github.io. CNAME github.map.fastly.net.

github.map.fastly.net. A 185.31.17.133

We use [GitHub Pages](http://pages.github.com/) and we set blog.dnsimple.com as a CNAME of aetrion.github.io, which is a CNAME of github.map.fastly.net, which is an A record pointing to 185.31.17.133. This means blog.dnsimple.com resolves to 185.31.17.133.

An A record points a name to an IP. A CNAME record can point a name to another CNAME or to an A record.

[**COMPUTER**](https://www.lifewire.com/learn-how-internet-network-4102756) **NETWORK**

A **computer network** is a set of **computers** connected together for the purpose of sharing resources. The most common resource shared today is connection to the Internet. Other shared resources can include a printer or a file server.

One way to categorize the different types of [computer network](https://www.lifewire.com/what-is-computer-networking-816249) designs is by their scope or scale. For historical reasons, the networking industry refers to nearly every type of design as some kind of *area network*. Common types of area networks are:

* **LAN**: Local Area Network
* **WAN**: Wide Area Network
* **WLAN**: [Wireless Local Area Network](https://www.lifewire.com/wlan-816565)
* **MAN**: Metropolitan Area Network
* **SAN**: [Storage Area Network](https://www.lifewire.com/definition-of-san-818007), System Area Network, Server Area Network, or sometimes Small Area Network
* **CAN**: Campus Area Network, Controller Area Network, or sometimes Cluster Area Network
* **PAN**: [Personal Area Network](https://www.lifewire.com/definition-of-pan-817889)

LAN and WAN are the two primary and best-known categories of area networks, while the others have emerged with technology advances.

Network types differ from [network topologies](https://www.lifewire.com/computer-network-topology-817884) (such as bus, ring, and star).

**LAN: Local Area Network**

A [LAN](https://www.lifewire.com/what-is-lan-4684071) connects network devices over a relatively short distance. A networked office building, school, or home usually contains a single LAN, though sometimes one building will contain a few small LANs (perhaps one per room), and occasionally a LAN will span a group of nearby buildings. In [TCP/IP networking](https://www.lifewire.com/transmission-control-protocol-and-internet-protocol-816255), a LAN is often but not always implemented as a single IP [subnet](https://www.lifewire.com/what-is-subnet-818392).

In addition to operating in a limited space, LANs are also typically owned, controlled, and managed by a single person or organization. They also tend to use certain connectivity technologies, primarily [Ethernet](https://www.lifewire.com/what-is-ethernet-3426740) and [Token Ring](https://www.lifewire.com/what-is-token-ring-817952).

**WAN: Wide Area Network**

As the term implies, a [WAN](https://www.lifewire.com/wide-area-network-816383) spans a large physical distance. The [internet](https://www.lifewire.com/difference-between-the-internet-and-the-web-2483335) is the largest WAN, spanning the Earth.

A WAN is a geographically-dispersed collection of LANs. A network device called a [router](https://www.lifewire.com/what-is-a-router-2618162) connects LANs to a WAN. In IP networking, the router maintains both a LAN address and a WAN address.

A WAN differs from a LAN in several important ways. Most WANs (like the internet) aren't owned by any one organization but rather exist under collective or distributed ownership and management.

WANs tend to use technology like [ATM](https://www.lifewire.com/asynchronous-transfer-mode-817942), [Frame Relay](https://www.lifewire.com/definition-of-frame-relay-817947), and [X.25](https://www.lifewire.com/x-25-816286) for connectivity over the longer distances.

**LAN, WAN, and Home Networking**

Residences typically employ one LAN and connect to the internet WAN via an [internet service provider (ISP)](https://www.lifewire.com/internet-service-provider-isp-2625924) using a [broadband modem](https://www.lifewire.com/definition-of-broadband-modem-817451). The ISP provides a WAN IP address to the modem, and all of the computers on the home network use LAN IP addresses (also called [private IP addresses](https://www.lifewire.com/what-is-a-private-ip-address-2625970)).

All computers on the home LAN can communicate directly with each other but must go through a central [network gateway](https://www.lifewire.com/definition-of-gateway-817891), typically a [broadband router](https://www.lifewire.com/what-is-a-broadband-router-816301), to reach the ISP and beyond.

**Other Types of Area Networks**

While [LAN](https://www.lifewire.com/virtual-local-area-network-817357) and WAN are by far the most popular network types mentioned, you may also see references to these others:

* **Wireless Local Area Network**: A LAN based on [Wi-Fi](https://www.lifewire.com/what-is-wi-fi-2377430) wireless network technology.
* **Metropolitan Area Network**: A network spanning a physical area larger than a LAN but smaller than a WAN, such as a city. A MAN is typically owned and operated by a single entity such as a government body or large corporation.
* **Campus Area Network**: A network spanning multiple LANs but smaller than a MAN, such as on a university or local business campus.
* **Personal Area Network**: A network that surrounds an individual. A wireless PAN (WPAN) might be created between [Bluetooth](https://www.lifewire.com/definition-of-bluetooth-816260) devices.
* **Storage Area Network**: [Connects servers to data storage devices](https://www.lifewire.com/definition-of-san-818007) through technology like [Fibre Channel](https://www.lifewire.com/definition-of-fibre-channel-816326).
* **System Area Network**(also called Cluster Area Network, or CAN): Links high-performance computers with high-speed connections in a cluster configuration.

**IP ADDRESS**

An Internet Protocol **address** (**IP address**) is a numerical label assigned to each device connected to a computer network that uses the Internet Protocol for communication. An **IP address** serves two main functions: host or network interface identification and location addressing.

## STATIC IP ADDRESS

As the name speaks, the static IP addresses are those types of IP address that never change once they are assigned to a device on a network. No doubt this type of addressing is cost effective but could have a high security risk. Static IP addresses are mostly used by web, email and gaming servers who don’t care much about hiding their locations.

## DYNAMIC IP ADDRESS

On the other hand, a Dynamic IP address changes each time the device logs in to a network. This kind of IP address is very tough to trace and are thus used by companies and business firms.

You must be thinking as to who or what allocates this Dynamic IP address every time the device logs in. Well, these IP address are assigned using DHCP (Dynamic Host Configuration Protocol). Talking about DHCP in detail is beyond the scope of this article and we will take it up in a future post.

**PING**

Command used to check connectivity between two devices. The **ping command** is usually used as a simple way to verify that a computer can communicate over the network with another computer or network device.

**Domain Name System**

Is responsible in a network to translate domain name to their correspondent IP address. (**Domain Name Service**) The Internet's system for converting alphabetic names into numeric IP addresses. For example, when a Web address (URL) is typed into a browser, DNS servers return the IP address of the Web server associated with that name.

NAT

Is a protocol used like a router to manage traffic coming from all devices in a network.

# What computer networks are and how to actually understand them

by Sumedh Nimkarde

Whether you are new to the world of development, or have been building things for a long time — or even if you’re a person who just likes computers and uses the internet daily — you’ve got to know the basics of networking and specifically Computer Networks.

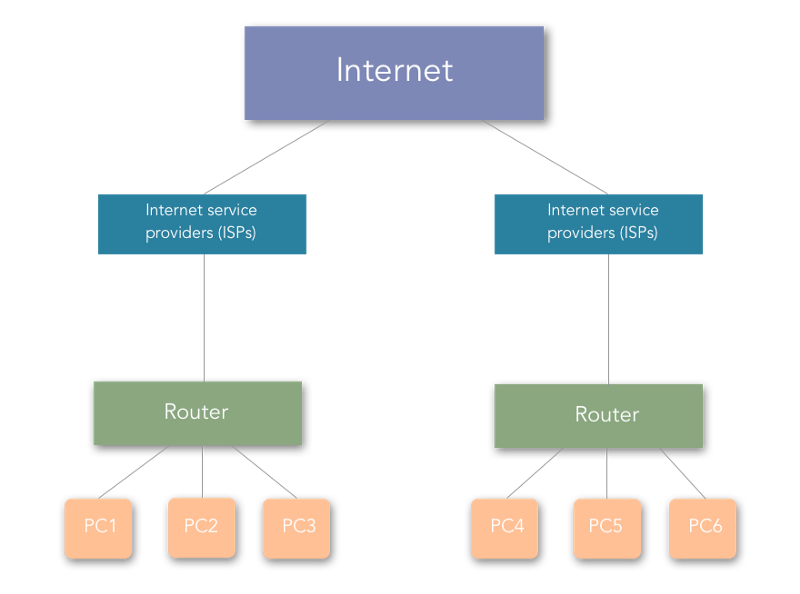
If you like digging more into servers, their security, and how you connect to your servers from a remote client, all of this requires some knowledge of computer networks and their components. I have tried to cover most of the topics concerning computer networks in this article.

Also, from here, I will refer to “computer networks” simply as “networks”.

Let us first look at my working definition of computer networks:

Computer networks can be defined as the exchange of network packets between computing machines across the world with the help of data lines like wire cables, optical fibers, etc.

The**Internet** is a kind of computer network. Sorta.



We will take a look at some commonly used terms and components and how they function in a computer network, some of which are in the above diagram.

### Commonly used terms in Computer Networks

#### Nodes

Nodes in computer networks mean any computing device such as computers, mobile phones, tablets, etc which try to send and receive network packets across the network to another similar device.

#### Network Packets

Network packets are nothing but the information or units of data that a source node wants to send/receive to/from the destination node. In this article, network packets/data packets all convey the same meaning.

#### Internet Protocol (IPs)

Consider you want to send a birthday gift to your friend on their birthday, where will you send it? To their street address right?

Same is the case here. The early computer scientists wanted to identify computers on the internet with a unique number, something like a telephone numbers today. So, they came up with the concept of TCP/IP.

An IP of a computer device is the address of that device in a computer network. Technically, it is a 32-bit number used which identifies devices in a network. All the communication to and fro from the device in that network will be done in terms of its IP address.

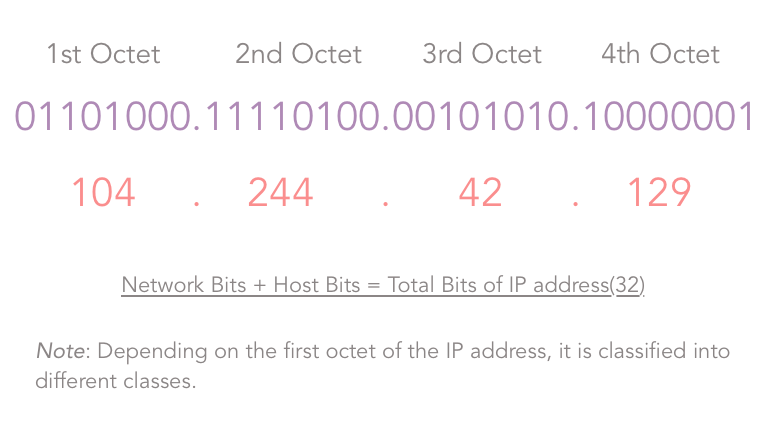
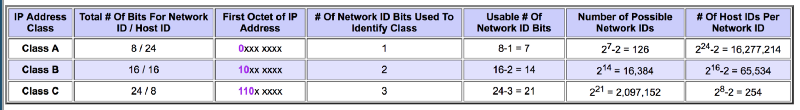
Consider that you are uploading a file to any site or say to Google drive.

Talking at the lowest level of network communication, your file is converted to packets and each packet has the destination node address with it which is nothing but the IP address.

On a higher level, IP addresses are classified into two types:

* **IPv4**: IPv4 addresses are 32 bits (four bytes) as explained in the definition. An example of the IPv4 address would be **104.244.42.129**which is the IPv4 address of**twitter.com**.They are stable to use and hence are used today to identify machines in the world.
* **IPv6**: IPv6 addresses are pretty new to the world and are basically eight hexadecimal numbers separated by “:”. An example of IPv6 address would be **2001:0cb8:85a3:0000:0000:8a2e:0370:7334**. They are unstable and hence not used widely yet. The web is still using IPv4 due to its stability and there is no estimate when we will start to use IPv6 since it is not stable for now.

IPv4 is classified into five classes named Class A, B, C, D, E.

Octets in IP address.Source: tcpipguide.com

**Class A**: As shown in the third column of the above image, for a Class A IP addresses, the first bit of the first octet of the IP address is constant and is “0”.

The Second column indicates the Network bits and the host bits of the corresponding class of IP address. Consider in case of a Class A IP address, we have the following formula:

**Number of networks/subnets = 2^(# of network bits) .**

**Number of valid hosts in each subnet = 2^(# of host bits) — 2 .**

The number of network bits and host bits are decided by the default subnet mask of the class of IP address.

The default subnet mask for a class A IP addresses is **255.0.0.0,**that is **11111111.00000000.0000000.00000000`.**Thus, for class A:

**Network bits = 8, and Host bits = 24.**

Since Network bits = 8, Host bits = 24, their sum has to be 32, since IPv4 addresses are of 32 bits. But, since we are using the one bit (first bit in the first octet) to identify the class:

**Number of usable network bits = Number of network bits — Number of constant bits = 8–1 = 7**

Thus, the ***Number of possible networks in Class A* = 2^7 — 2 = 126** and,

**Number of possible hosts (that is devices that can be connected to the network) per network in Class A = 2^24-2 = 16277214 .**

Now, here, for class A, you may wonder why I subtracted an extra 2 from the number of possible networks. It is because, for class A, 127.x.y.z was kept reserved. For other classes, the usual formula is used.

Thus, IP addresses in class A range from 1.x.x.x to 126.x.x.x.

**Class B:**the case is similar with Class B. The only difference is 2 bits of the first octet are constant (10) and they identify the class of IP address that is class B. All other calculations are same, and I am not mentioning them here since they are easy to grab from the table above. They range from 128.0.x.x to 191.255.x.x .

**Class C**: 3 bits of the first octet are constant (110) and they identify the class as class C. They range from 192.0.0.x to 223.255.255.x .

**Class D and Class E**: Class D and Class E are used for experimental purposes.

IPv4 addresses are mainly of two types:

* **Static**: These IP addresses are the ones which remain constant for a device over time. Examples of these are the remote servers that we use to host our apps, websites, etc. where we use the ssh client to ssh to our server.
* **Dynamic**: Generally, these are the IP addresses that a common computer in an Internet network is assigned. Try switching your router off and you will see a change in the IP address of your computer! (But only after reading this article ?). Now, you may be thinking who allocates these IP addresses? It is the DHCP (Dynamic Host Configuration Protocol) server which is explained briefly further in this article.

**Note**: A device can have multiple IP addresses at the same time. Consider a device connected to two networks, wifi as well as any LAN network — it will have two IP addresses. This implies that the IP addresses are assigned to the interfaces and not directly to the computer.

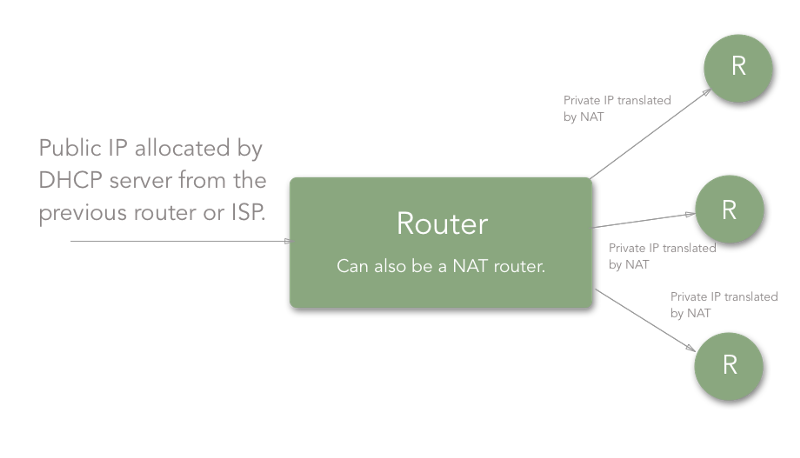
Okay, so far so good. Let’s continue.

### Routers

As its name suggests, a Router is a hardware component that takes care of routing packets. It determines which node the packet came from and which destination node the sender node want to send it to. No computer knows where other computers are located, and packets are not sent to every computer. A Router identifies the destination node address to which a network packet has to be sent and it forwards it to the desired address.

Routers have a specific **“Routing Protocol”** which defines the format in which they exchange data with another router or networking nodes. In other words, routing protocol defines how routers communicate with each other.

Routers build up a **“Routing Table”** which identifies the most optimized paths to be taken in the network while sending packets.

A Router.

Technically, a routing table is just a table with the list of “routes” from one router to other. Each route consists of the address of the other routers/nodes in the network and how to reach them.

Routing table:

Destination Gateway Genmask Flags Metric Refs Ifacedefault 192.168.0.1 0.0.0.0 UG 1024 233 eth0192.168.0.0 \* 255.255.255.0 UC 0 0 wlan0192.168.0.0 \* 255.255.255.0 UH 0 2 eth0

Above is an example of a routing table. The key points to take a note of here are:

* **Destination:**This is the IP address of the destination node. It indicates where the network data packet should end up.
* **Gateway:**Gateway is the component which connects two networks. Consider that you have a router connected to another router. Each of the routers has devices connected to it. So, the address of the last router (say R1 here) after which the network packet enters the other network (say R2’s network) is called the gateway. Usually, the gateways are nothing but the routers. Let me give one more example: say that your room is one network and your sibling’s room next to yours is another network, then the “door” between the two rooms can be considered the gateway. People sometimes refer to the “**routers**” as the gateway, because, that’s what they are, “**a gateway to another network**”.
* **Genmask/Subnet mask:** It is nothing but the net/subnet mask. A subnet mask is a number which when combined with an IP address allows you to divide the IP space into smaller and smaller chunks for use in both physical and logical networks. The explanation of how subnet mask calculations happen is beyond the scope of this article.
* **Flags:**Different flags have a different meaning. For example, in the first route, “U” in “UG” means the route is UP, whereas “G” in “UG” means GATEWAY. Since the route signifies a GATEWAY, it is a door to the other network. Whenever we send any data through this route, it gets sent to another network.
* **Iface (Network interface):**Network interface refers to the network that the route defined in the routing table is having the destination computer in. That is if you are connected to Wifi, then it would be “wlan” and when you are connected to a LAN, then it would be “eth”.

So this is the way a router works, with the help of **Routing Protocol** and **Routing Table**.

All good up to now. But, you must be thinking —

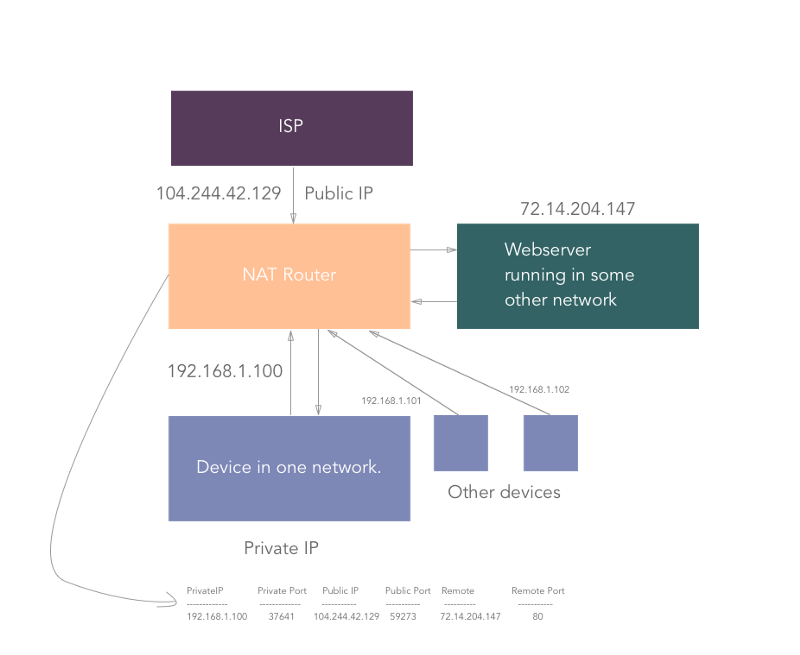
“Okay! But hey, we are learning about components here. I need to stitch them together and get to know how the internet works.”

Cool! Some more terms and you will have a proper understanding of how everything goes.

### Network Address Translation (NAT)

Network address translation is a technique used by routers to provide internet service to more devices with less usage of public IPs. Thus, a router is assigned a single IP address by the ISP and it assigns the private IPs to all the devices connected to it. NAT helps the ISPs provide internet access to more consumers.

Thus, if you are connected to the router of your house, your public IP will be visible to the world, but the private one will not. Whatever network packets are communicated will be addressed by your public IP (that is the public IP assigned to the router).

Network address translation (NAT)

Consider the above figure. Let’s say that in your home network, you are trying to access **medium.com (remote static IP:**72.14.204.147**)**, from your computer (private IP: 192.168.1.100).

So, for your computer, the connection looks like:

192.168.1.100:37641 → 72.14.204.147:80 .

“37641” is the random port number assigned by NAT router to your device/computer. (When there is network communication between daemons running on different ports on a computer, the respective port is used by NAT). Each outbound connection gets an assigned port by the NAT router.

The connection is established in NAT like:

Private IP |PrivatePort |PublicIP |PublicPort |Remote |RemotePort

------------- ------------ --------- ----------- ------- -----------

192.168.1.100 | 37641 | 104.244.42.129 | 59273 | 72.14.204.147 | 80

But, since the outside world of the network doesn’t know about your private address, the connection looks like the following to **medium.com**:

104.244.42.129:59273 → 72.14.204.147:80 .

That way, we achieve assigning a higher number of IP addresses without wasting many public IPs.

Now, when medium.com sends the response back to 104.244.42.129:59273 , it travels all the way to your home router which then looks up for the respective private IP and private port and redirects the packet to your device/computer.

**Note**: NAT is a generalized concept. NAT can be achieved as 1:1, 1:N where 1, N are the number of IP addresses in the network. A technique called as “IP Masquerading” is a 1:N NAT.

### Dynamic Host Configuration Protocol (DHCP)

**Dynamic Host Configuration Protocol** or **DHCP**is responsible for assigning dynamic IP addresses to the hosts. The DHCP server is maintained by the ISP or previous router if there is a chain of routers to reach the host.

Thus, allocation of IP addresses is carried out by the DHCP server. Generally, ISP maintains a DHCP server and the routers in our houses get assigned a public IP from the DHCP server.

**Note**: Whenever a router or say a DHCP server maintained by an ISP or router restarts, the IP address allocation starts again and devices are allocated IPs which are different than the previous ones.

### Domain Name System/Server

We have already discussed that any machine is identified by the IP address.

Okay, so you are running a web server on your localhost on your machine. If you have dug around in the hosts on any Linux machine, you would have encountered something like this:

127.0.0.1 localhost255.255.255.255 broadcasthost::1 localhost

which means that even if you type 127.0.0.1 in your browser’s URL bar, it would mean the same as localhost .

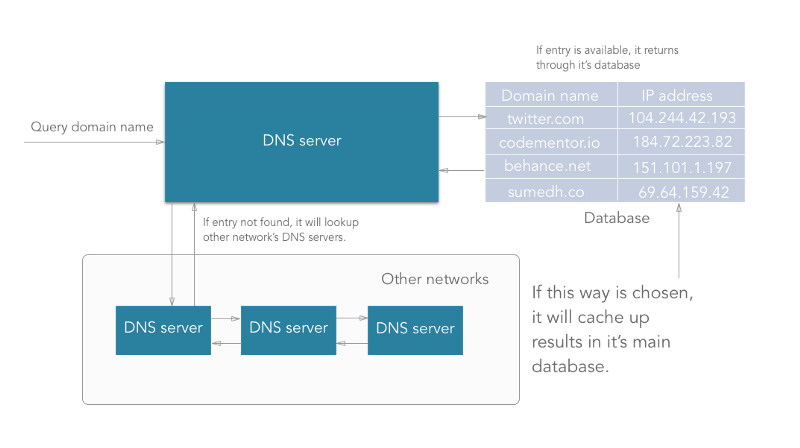
Similar to the above, the websites you use daily are web servers running on some remote instance/node having a static IP address. So, typing that IP address in your browser’s URL bar will take you to the website?

Yes, surely it will. But, are you a superhuman to remember the IP addresses of thousands of sites?

**NO.**

Thus, there come the domains that we use, say medium.com, twitter.com, behance.net, codementor.io, etc.

A Domain Name Server is a server having huge records of domain name mapping IP addresses which searches for the domain input and returns the respective IP address of the machine on which the website you want to access is hosted.

Domain Name System (DNS)

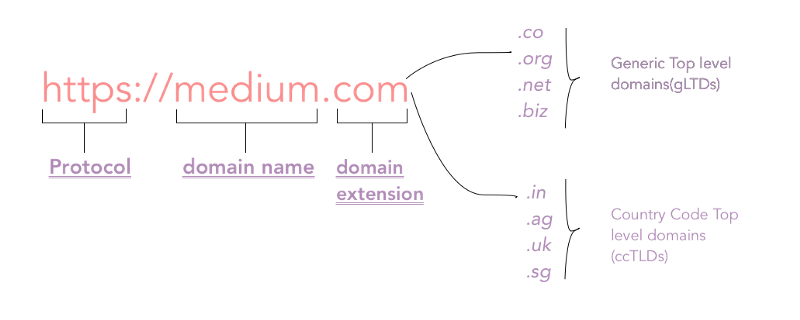
**How does DNS work actually?**

1. DNS is managed by your ISP (internet service provider).
2. When we type an URL in the address bar, the data packets travel through your router, maybe multiple routers to your ISP where your DNS server is present.
3. DNS server present at the ISP looks up for the domain in its database. If an entry is found, then it returns it.
4. If any entry is not found in its primary database that it maintains, the DNS server will travel through the internet to another DNS server maintained by another ISP and check if the entry is available in that another DNS server’s database. Along with returning the IP address taken from another DNS, it will update the primary database with this new entry also.
5. Thus, sometimes (very rarely) a DNS server may have to traverse to multiple DNS servers to get a matching entry.
6. If after traversing a lot of DNS servers across the internet, it doesn’t get a matching entry, then the DNS server throws an error indicating that the “domain name is invalid or doesn’t exist”.

**Note:**

**The Internet Corporation for Assigned Names and Numbers (ICANN)**is a consortium (a non-profit corporation) that manages the assignment of domain names and IP address ranges on behalf of the community.

A domain is divided into three parts as shown in the following figure.



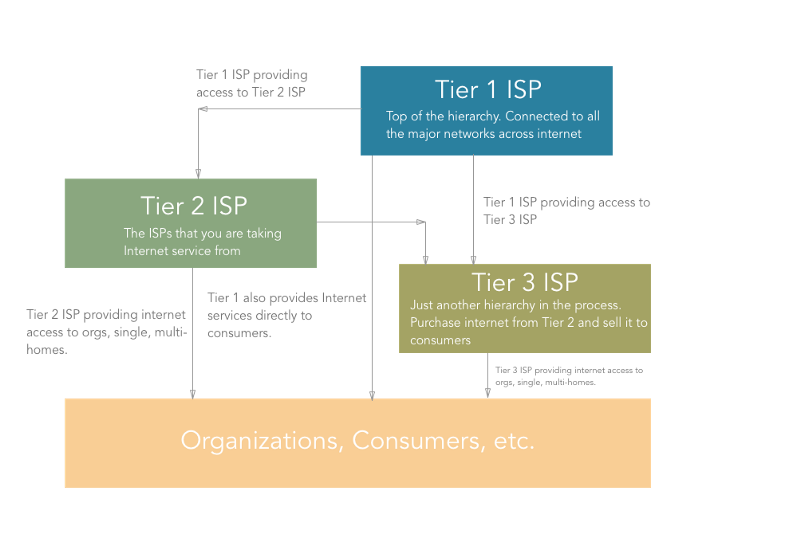
1. **Protocol**: The protocol used to access the website, for example, HTTP, HTTPS, etc.
2. **Domain name**: The main domain name in our domain. This can be anything that is available as per the ICANN registry.
3. **Domain extension**: This is one which is considered important while buying a domain. Generally, it is classified into two types:

* **Generic Top-level Domains (gTLDs)**: This includes most popular domain extensions like .com, .org, .net, .edu, .co, etc.
* **Country Code Top-level Domains(ccTLDs)**: These indicate that the domain is related to the country code specified in the domain extension. For example, “.in” indicates that the website is originated from India. Also, some of the ccTLDs require that the person purchasing the domain should be from the same country. Most of the small country code extensions are not searchable from outside that country.

### Internet Service Providers (ISPs)

**Internet Service Providers** are the companies that provide everyone Internet. The article you are reading now is because of the internet that your ISP provides you.

ISPs provide internet, handle routing your requests to the correct destination, resolve domain names with the help of DNS cache that they maintain, and handle all this network infrastructure which enables us to use the internet.

Internet Service Providers (ISP)

ISP is a hierarchical thing working across the internet. There are certain types of ISPs namely Tier 1, Tier 2, Tier 3 ISPs.

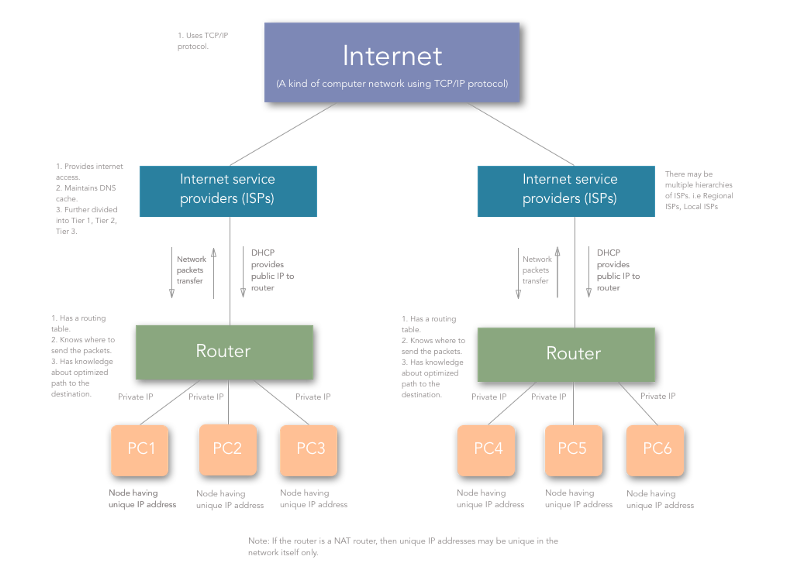
* **Tier 1** ISPs are the ones which connect major networks on the internet. Consider them as the major highways of the internet. They are connected to almost every network on the internet. Also, they provide internet access to the Tier 2 ISPs. ex. CERFNet, UUNet, PSINet. They are also called Network Service Providers. These ISPs are connected to each other by means of [large cables](https://www.google.co.in/search?q=ISP+cables+in+sea&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjO6aPV57XdAhUMyrwKHaZvAJAQ_AUICigB&biw=1920&bih=1006#imgrc=gzXVOSE_UDEmAM:) going beneath the sea.
* The **Tier 2 (Regional)**ISPs are the ones who primarily provide Internet services to organizations, consumers (that is “us”) or the Tier 3 ISPs. The internet connection you are using is from a Tier 2 ISP. However, organizations can also get Internet access from Tier 1 ISPs.
* **Tier 3 (Local)**ISPs are just like Tier 2. It’s just one more level of hierarchy out there that purchases bandwidth from Tier 2 ISP and sells it to consumers.

The traffic that goes through your router also goes through Tier 3 (if present), Tier 2, and ultimately through Tier 1 ISPs all the way to another network.

Woot Woot! I am happy that you are still with me. We will put all the things together now.

### Putting all of the above things together

Up until now, we have learned about all the components needed to make everything work. Now, we will glue them together.

A Detailed diagram of a general Computer Network

Let’s summarize all the things we’ve learned:

* When a computer/device comes online, it gets a private IP assigned by the router. The router gets a public IP from the ISP.
* Other devices in the network are allocated unique private IPs.
* ISPs are the ones who are present across the world and are connected to each other. They sell Internet services to the regional and local ISPs, from whom we, the consumers, purchase Internet.
* Thus, when a device tries to establish a network connection with some other device on some other network, it does it with the identity of its gateway (the router). The router then maps the private IP and private port number with the public IP and random high integer public port number.
* The router then sends the packets to the desired destination where some other router or gateway does the same thing as the previous router and analyses which computer/device that packet came from.
* The remote computer/device responds by sending the destination as the public IP and public port of the router.
* The router then again checks for the private IP and private port and forwards the network packets.

# The Layers of the OSI Model Illustrated

The Open Systems Interconnection (OSI) model defines a networking framework to implement protocols in layers, with control passed from one layer to the next. It is primarily used today as a teaching tool. It conceptually divides [computer network](https://www.lifewire.com/what-is-computer-networking-816249) architecture into 7 layers in a logical progression. The lower layers deal with electrical signals, chunks of [binary data](https://www.lifewire.com/working-with-binary-and-hexadecimal-numbers-816247), and routing of these data across networks. Higher levels cover network requests and responses, representation of data, and network protocols as seen from a user's point of view.

The OSI model was originally conceived as a standard architecture for building network systems and indeed, many popular network technologies today reflect the layered design of OSI.

#### PHYSICAL LAYER

 Lifewire / Colleen Tighe

At Layer 1, the Physical layer of the OSI model is responsible for ultimate transmission of digital data [bits](https://www.lifewire.com/definition-of-bit-816250) from the Physical layer of the sending (source) device over network communications media to the Physical layer of the receiving (destination) device. Examples of layer 1 technologies include [Ethernet cables](https://www.lifewire.com/what-is-an-ethernet-cable-817548) and [hubs](https://www.lifewire.com/ethernet-and-network-hubs-816358). Additionally, hubs and other [repeaters](https://www.lifewire.com/definition-of-repeater-816359) are standard network devices that function at the Physical layer, as are cable connectors.

At the Physical layer, data are transmitted using the type of signaling supported by the physical medium: electric voltages, radio frequencies, or pulses of infrared or ordinary light.



**SPONSORED BY VERIZON 5G**

5G Changes Everything

Will 5G change your life? There are lots of real world 5G use cases, like device-to-device communication, smart cars, and faster access to healthcare.

**LEARN MORE**

#### DATA LINK LAYER

Lifewire / Colleen Tighe

When obtaining data from the Physical layer, the Data Link layer checks for physical transmission errors and packages bits into data "frames". The Data Link layer also manages physical addressing schemes such as [MAC](https://www.lifewire.com/media-access-control-mac-817973) addresses for Ethernet networks, controlling access of any various network devices to the physical medium. Because the Data Link layer is the single most complex layer in the OSI model, it is often divided into two parts, the **Media Access Control** sub-layer and the **Logical Link Control** sub-layer.

#### NETWORK LAYER

Lifewire / Colleen Tighe

The Network layer adds the concept of routing above the Data Link layer. When data arrives at the Network layer, the source and destination addresses contained inside each frame are examined to determine if the data has reached its final destination. If the data has reached the final destination, this layer 3 formats the data into packets delivered up to the Transport layer. Otherwise, the Network layer updates the destination address and pushes the frame back down to the lower layers.

To support routing, the Network layer maintains logical addresses such as [IP addresses](https://www.lifewire.com/what-is-an-ip-address-2625920) for devices on the network. The Network layer also manages the mapping between these logical addresses and physical addresses. In IP networking, this mapping is accomplished through the [Address Resolution Protocol (ARP)](https://www.lifewire.com/address-resolution-protocol-817941).

#### TRANSPORT LAYER

Lifewire / Colleen Tighe

The Transport Layer delivers data across network connections. [TCP](https://www.lifewire.com/transmission-control-protocol-and-internet-protocol-816255) is the most common example of a Transport Layer 4 [network protocol.](https://www.lifewire.com/definition-of-protocol-network-817949) Different transport protocols may support a range of optional capabilities including error recovery, flow control, and support for re-transmission.

#### SESSION LAYER

Lifewire / Colleen Tighe

The Session Layer manages the sequence and flow of events that initiate and tear down network connections. At layer 5, it is built to support multiple types of connections that can be created dynamically and run over individual networks.

#### PRESENTATION LAYER

Lifewire / Colleen Tighe

The Presentation layer is the simplest in function of any piece of the OSI model. At layer 6, it handles syntax processing of message data such as format conversions and encryption/decryption needed to support the Application layer above it.

#### APPLICATION LAYER

Lifewire / Colleen Tighe

The Application layer supplies network services to end-user applications. Network services are typically protocols that work with user's data. For example, in a web browser application, the Application layer protocol [HTTP](https://www.lifewire.com/hypertext-transfer-protocol-817944) packages the data needed to send and receive webpage content. This layer 7 provides data to (and obtains data from) the Presentation layer.

**Was this page helpful?**

<https://www.lifewire.com/layers-of-the-osi-model-illustrated-818017>

### Directories

From top to bottom, the directories you are seeing are as follows.

#### */bin*

*/bin* is the directory that contains *bin*aries, that is, some of the applications and programs you can run. You will find the *ls* program mentioned above in this directory, as well as other basic tools for making and removing files and directories, moving them around, and so on. There are more *bin* directories in other parts of the file system tree, but we’ll be talking about those in a minute.

#### */boot*

The */boot* directory contains files required for starting your system. Do I have to say this? Okay, I’ll say it: **DO NOT TOUCH!**. If you mess up one of the files in here, you may not be able to run your Linux and it is a pain to repair. On the other hand, don’t worry too much about destroying your system by accident: you have to have superuser privileges to do that.

#### */dev*

*/dev* contains *dev*ice files. Many of these are generated at boot time or even on the fly. For example, if you plug in a new webcam or a USB pendrive into your machine, a new device entry will automagically pop up here.

#### */etc*

*/etc* is the directory where names start to get confusing. */etc* gets its name from the earliest Unixes and it was literally “et cetera” because it was the dumping ground for system files administrators were not sure where else to put.

Nowadays, it would be more appropriate to say that *etc* stands for “Everything to configure,” as it contains most, if not all system-wide configuration files. For example, the files that contain the name of your system, the users and their passwords, the names of machines on your network and when and where the partitions on your hard disks should be mounted are all in here. Again, if you are new to Linux, it may be best if you don’t touch too much in here until you have a better understanding of how things work.

#### */home*

*/home* is where you will find your users’ personal directories. In my case, under */home* there are two directories: */home/paul*, which contains all my stuff; and */home/guest*, in case anybody needs to borrow my computer.

#### */lib*

*/lib* is where *lib*raries live. Libraries are files containing code that your applications can use. They contain snippets of code that applications use to draw windows on your desktop, control peripherals, or send files to your hard disk.

There are more *lib* directories scattered around the file system, but this one, the one hanging directly off of */* is special in that, among other things, it contains the all-important kernel modules. The kernel modules are drivers that make things like your video card, sound card, WiFi, printer, and so on, work.

#### */media*

The */media* directory is where external storage will be automatically mounted when you plug it in and try to access it. As opposed to most of the other items on this list, */media* does not hail back to 1970s, mainly because inserting and detecting storage (pendrives, USB hard disks, SD cards, external SSDs, etc) on the fly, while a computer is running, is a relatively new thing.

#### */mnt*

The */mnt* directory, however, is a bit of remnant from days gone by. This is where you would manually mount storage devices or partitions. It is not used very often nowadays.

#### */opt*

The */opt* directory is often where software you compile (that is, you build yourself from source code and do not install from your distribution repositories) sometimes lands. Applications will end up in the */opt/bin* directory and libraries in the */opt/lib* directory.

A slight digression: another place where applications and libraries end up in is */usr/local*, When software gets installed here, there will also be */usr/local/bin* and */usr/local/lib* directories. What determines which software goes where is how the developers have configured the files that control the compilation and installation process.

#### */proc*

*/proc*, like */dev* is a virtual directory. It contains information about your computer, such as information about your CPU and the kernel your Linux system is running. As with */dev*, the files and directories are generated when your computer starts, or on the fly, as your system is running and things change.

#### */root*

*/root* is the home directory of the superuser (also known as the “Administrator”) of the system. It is separate from the rest of the users’ home directories BECAUSE YOU ARE NOT MEANT TO TOUCH IT. Keep your own stuff in you own directories, people.

#### */run*

*/run* is another new directory. System processes use it to store temporary data for their own nefarious reasons. This is another one of those DO NOT TOUCH folders.

#### */sbin*

*/sbin* is similar to */bin*, but it contains applications that only the superuser (hence the initial *s*) will need. You can use these applications with the sudo command that temporarily concedes you superuser powers on many distributions. */sbin* typically contains tools that can install stuff, delete stuff and format stuff. As you can imagine, some of these instructions are lethal if you use them improperly, so handle with care.

#### */usr*

The */usr* directory was where users’ home directories were originally kept back in the early days of UNIX. However, now */home* is where users kept their stuff as we saw above. These days, */usr* contains a mish-mash of directories which in turn contain applications, libraries, documentation, wallpapers, icons and a long list of other stuff that need to be shared by applications and services.

You will also find *bin*, *sbin* and *lib* directories in */usr*. What is the difference with their root-hanging cousins? Not much nowadays. Originally, the */bin* directory (hanging off of root) would contain very basic commands, like ls, mv and rm; the kind of commands that would come pre-installed in all UNIX/Linux installations, the bare minimum to run and maintain a system. */usr/bin* on the other hand would contain stuff the users would install and run to use the system as a work station, things like word processors, web browsers, and other apps.

But many modern Linux distributions just put everything into */usr/bin* and have */bin* point to */usr/bin* just in case erasing it completely would break something. So, while Debian, Ubuntu and Mint still keep */bin* and */usr/bin* (and */sbin* and */usr/sbin*) separate; others, like Arch and its derivatives just have one “real” directory for binaries, */usr/bin*, and the rest or *\*bin*s are “fake” directories that point to */usr/bin*.

#### */srv*

The */srv* directory contains data for servers. If you are running a web server from your Linux box, your HTML files for your sites would go into */srv/http* (or */srv/www*). If you were running an FTP server, your files would go into */srv/ftp*.

#### */sys*

*/sys* is another virtual directory like */proc* and */dev* and also contains information from devices connected to your computer.

In some cases you can also manipulate those devices. I can, for example, change the brightness of the screen of my laptop by modifying the value stored in the */sys/devices/pci0000:00/0000:00:02.0/drm/card1/card1-eDP-1/intel\_backlight/brightness* file (on your machine you will probably have a different file). But to do that you have to become superuser. The reason for that is, as with so many other virtual directories, messing with the contents and files in */sys* can be dangerous and you can trash your system. DO NOT TOUCH until you are sure you know what you are doing.

#### */tmp*

/tmp contains temporary files, usually placed there by applications that you are running. The files and directories often (not always) contain data that an application doesn’t need right now, but may need later on.

You can also use /tmp to store your own temporary files — /tmp is one of the few directories hanging off / that you can actually interact with without becoming superuser.

#### */var*

*/var* was originally given its name because its contents was deemed *variable*, in that it changed frequently. Today it is a bit of a misnomer because there are many other directories that also contain data that changes frequently, especially the virtual directories we saw above.

Be that as it may, */var* contains things like logs in the */var/log* subdirectories. Logs are files that register events that happen on the system. If something fails in the kernel, it will be logged in a file in */var/log*; if someone tries to break into your computer from outside, your firewall will also log the attempt here. It also contains *spools* for tasks. These “tasks” can be the jobs you send to a shared printer when you have to wait because another user is printing a long document, or mail that is waiting to be delivered to users on the system.

Your system may have some more directories we haven’t mentioned above. In the screenshot, for example, there is a */snap* directory. That’s because the shot was captured on an Ubuntu system. Ubuntu has recently incorporated [snap](https://www.ubuntu.com/desktop/snappy) packages as a way of distributing software. The */snap* directory contains all the files and the software installed from snaps.

### Digging Deeper

That is the root directory covered, but many of the subdirectories lead to their own set of files and subdirectories. Figure 2 gives you an overall idea of what the basic file system tree looks like (the image is kindly supplied under a CC By-SA license by Paul Gardner) and [Wikipedia has a break down with a summary of what each directory is used for](https://en.wikipedia.org/wiki/Unix_filesystem#Conventional_directory_layout).

