**What are Email Protocols (POP3, SMTP and IMAP) and their default ports?**

Email is an essential part of business and personal communication online. The email protocols define the mechanism of the email exchange between servers and clients. This way, they allow us to send and receive messages over the network correctly.

An email protocol is a group of rules which ensure that emails are properly transmitted over the Internet. In fact, there is a list of email protocols that handle email transactions

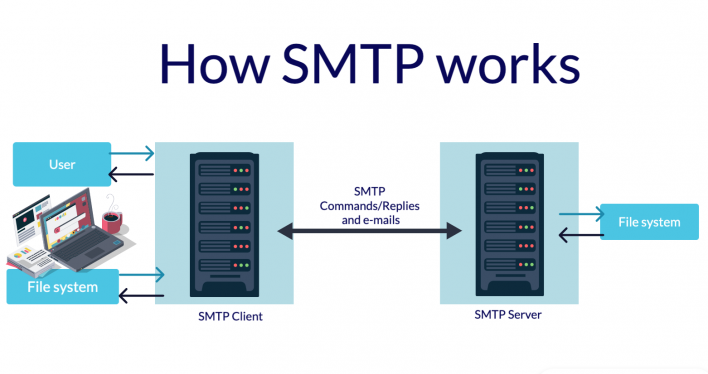
The standard email protocol list includes:

* SMTP
* POP3
* IMAP

Each of them operates differently and provides a different service for managing your email account.

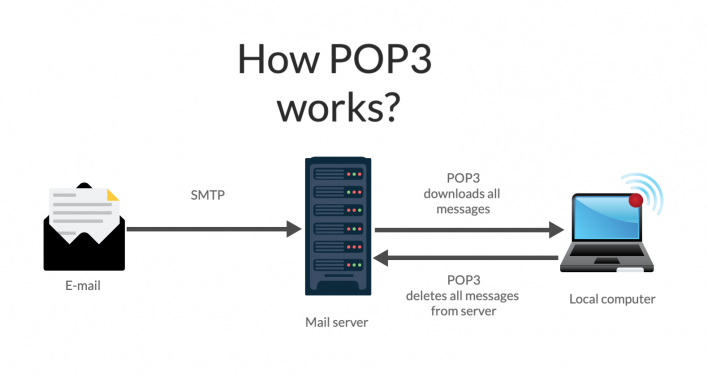
### **What is SMTP?**

[**SMTP**](https://world.siteground.com/kb/what-is-smtp/)**stands for Simple Mail Transfer Protocol**, and it is responsible for sending email messages. This protocol is used by email clients and mail servers to exchange emails between computers.



A mail client and the SMTP server communicate with each other over a connection established through a particular email port. Both entities are using SMTP commands and replies to process your outgoing emails. Thanks to the Simple Mail Transfer Protocol, messages can be sent from the same account on different email applications.

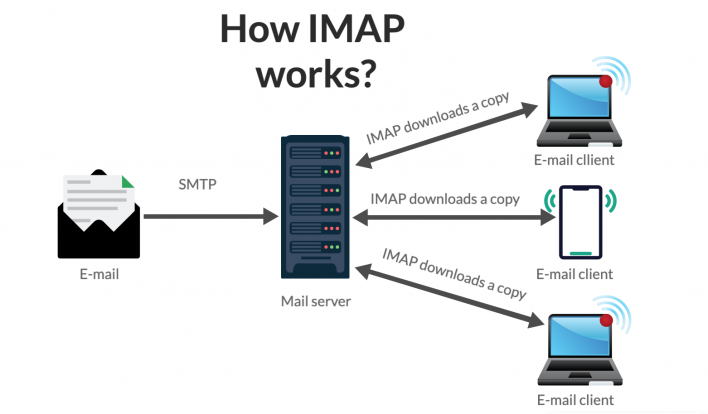
### **What is POP3?**



The [POP3](https://world.siteground.com/kb/what-is-pop3/) abbreviation stands for **Post Office Protocol version 3**, which provides access to an inbox stored in an email server. It executes the download and deletes operations for messages. Thus, when a POP3 client connects to the mail server, it retrieves all messages from the mailbox. Then it stores them on your local computer and deletes them from the remote server.

Modern POP3 clients allow you to keep a copy of your messages on the server if you explicitly select this option.

### **What is IMAP?**



**The Internet Message Access Protocol (**[**IMAP**](https://world.siteground.com/kb/what-is-imap/)**) allows you to access and manage your email messages on the email server.** This protocol permits you to manipulate folders, permanently delete and efficiently search through messages. It also gives you the option to set or remove email flags, or fetch email attributes selectively. By default, all messages remain on the server until the user specifically deletes them.

IMAP supports the connection of multiple users to a single mail server.

## **Default email ports**

Email ports are communication endpoints that define how a message should be transmitted. That includes whether a message should be encrypted and exchanged securely.

To establish a connection between your email client and your mail server, you need the latter’s [IP address](https://en.wikipedia.org/wiki/IP_address) and a port number. These attributes are assigned by [**IANA**](https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml) (**Internet Assigned Numbers Authority**).

Each protocol has its own port numbers to connect through and each port supports a different type of encryption.

### **SMTP Ports**

The available SMTP ports are four and each of them underlies a different type of encryption for email sending.

* **25** – This port serves to send messages in plain text, although if the mail server supports it, it can be encrypted with [TLS](https://en.wikipedia.org/wiki/Transport_Layer_Security). Therefore, many Internet service providers block it, as it represents a security risk.
* **Port 2525**is an alternative to the SMTP port 25 and can be encrypted over TLS.
* **587 –**This is the port IANA registered as the secure SMTP port, and it requires an explicit TLS connection. However, if the email server does not support TLS, the message will be sent in plain text.
* **Port 465**works over an implicit SSL connection and if the server does not support it, the operation will be aborted.

### **POP3 ports**

* **Port** **110** is the default POP3 port and it is not encrypted.
* The encrypted port for POP3 is **995** and works over TLS/SSL.

### **IMAP ports**

By default IMAP works on two ports like POP3:

* **143** – this is the default port which does not provide any encryption.
* **Port 993** is the secure port for IMAP and it works over TLS/SSL encryption.

## **What is the difference between SMTP, POP3, and IMAP?**

### **Incoming vs. outgoing protocols**

POP3 and IMAP are handling the incoming emails and they operate in different ways to retrieve or access your email messages. Thus, they are considered mail access protocols.

On the other hand, the Simple Mail Transfer Protocol is behind the message transfer from server to server, or mail client to server. As this is the protocol handling the email sending from an email account, it is labeled as the outgoing protocol.

In short, thanks to IMAP and POP3, you are able to receive emails, and SMTP allows you to send messages.

### **IMAP vs. POP3**

As we already mentioned, both of these protocols relate to email retrieval. All modern servers support both protocols, although they function in different manners.

While the POP3 protocol assumes that your email is being accessed only from one application, IMAP allows simultaneous access by multiple clients. This is why IMAP is more suitable for you, if you’re going to access your email from different locations or if multiple users manage your messages.

On the other hand, POP3 downloads your emails to your local computer, deleting them from the server. Thus, it reduces the space your email account uses on your web server.

To sum it up, there are 3 email protocols – SMTP, POP3, and IMAP. Each of them works on specific port numbers and operates differently. If you are having trouble connecting to an incoming or outgoing server, try using an alternative port number.

## What is DNS?

The Domain Name System (DNS) is the phonebook of the Internet. Humans access information online through [domain names](https://www.cloudflare.com/learning/dns/glossary/what-is-a-domain-name/), like nytimes.com or espn.com. Web browsers interact through [Internet Protocol (IP)](https://www.cloudflare.com/learning/network-layer/internet-protocol/) addresses. DNS translates domain names to [IP addresses](https://www.cloudflare.com/learning/dns/glossary/what-is-my-ip-address/) so browsers can load Internet resources.

Each device connected to the Internet has a unique IP address which other machines use to find the device. DNS servers eliminate the need for humans to memorize IP addresses such as 192.168.1.1 (in IPv4), or more complex newer alphanumeric IP addresses such as 2400:cb00:2048:1::c629:d7a2 (in IPv6).

## How does DNS work?

The process of DNS resolution involves converting a hostname (such as www.example.com) into a computer-friendly IP address (such as 192.168.1.1). An IP address is given to each device on the Internet, and that address is necessary to find the appropriate Internet device - like a street address is used to find a particular home. When a user wants to load a webpage, a translation must occur between what a user types into their web browser (example.com) and the machine-friendly address necessary to locate the example.com webpage.

In order to understand the process behind the DNS resolution, it’s important to learn about the different hardware components a DNS query must pass between.

## There are 4 DNS servers involved in loading a webpage:

* [DNS recursor](https://www.cloudflare.com/learning/dns/dns-server-types/) - The recursor can be thought of as a librarian who is asked to go find a particular book somewhere in a library. The DNS recursor is a server designed to receive queries from client machines through applications such as web browsers. Typically the recursor is then responsible for making additional requests in order to satisfy the client’s DNS query.
* **Root nameserver** - The [root server](https://www.cloudflare.com/learning/dns/glossary/dns-root-server/) is the first step in translating (resolving) human readable host names into IP addresses. It can be thought of like an index in a library that points to different racks of books - typically it serves as a reference to other more specific locations.
* [TLD nameserver](https://www.cloudflare.com/learning/dns/dns-server-types/) - The top level domain server ([TLD](https://www.cloudflare.com/learning/dns/top-level-domain/)) can be thought of as a specific rack of books in a library. This nameserver is the next step in the search for a specific IP address, and it hosts the last portion of a hostname (In example.com, the TLD server is “com”).
* [Authoritative nameserver](https://www.cloudflare.com/learning/dns/dns-server-types/) - This final nameserver can be thought of as a dictionary on a rack of books, in which a specific name can be translated into its definition. The authoritative nameserver is the last stop in the nameserver query. If the authoritative name server has access to the requested record, it will return the IP address for the requested hostname back to the DNS Recursor (the librarian) that made the initial request.

#### The 8 steps in a DNS lookup:

1. A user types ‘example.com’ into a web browser and the query travels into the Internet and is received by a DNS recursive resolver.
2. The resolver then queries a DNS root nameserver (.).
3. The root server then responds to the resolver with the address of a Top Level Domain (TLD) DNS server (such as .com or .net), which stores the information for its domains. When searching for example.com, our request is pointed toward the .com TLD.
4. The resolver then makes a request to the .com TLD.
5. The TLD server then responds with the IP address of the domain’s nameserver, example.com.
6. Lastly, the recursive resolver sends a query to the domain’s nameserver.
7. The IP address for example.com is then returned to the resolver from the nameserver.
8. The DNS resolver then responds to the web browser with the IP address of the domain requested initially.

Once the 8 steps of the DNS lookup have returned the IP address for example.com, the browser is able to make the request for the web page:

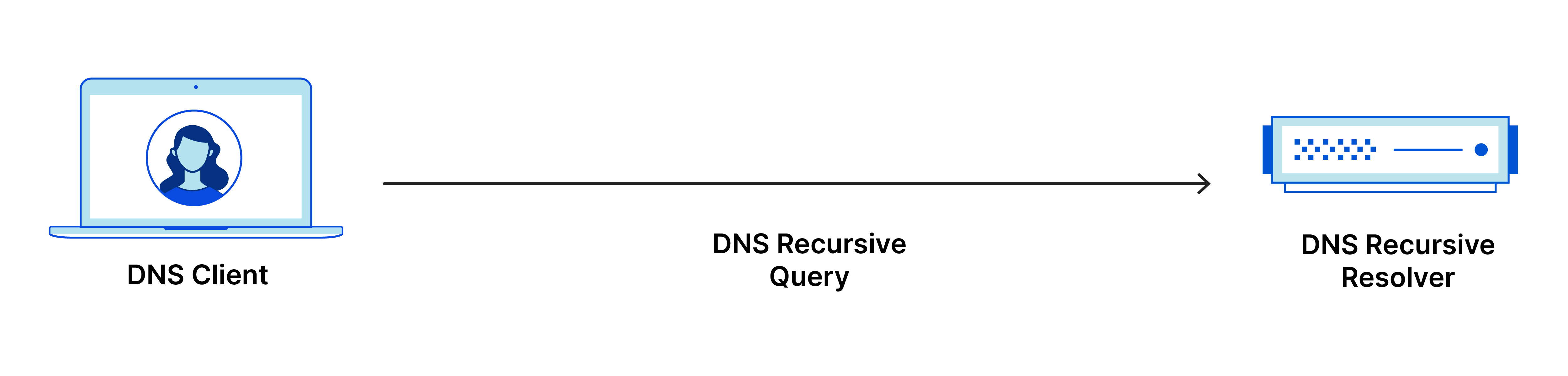
1. The browser makes a [HTTP](https://www.cloudflare.com/learning/ddos/glossary/hypertext-transfer-protocol-http/) request to the IP address.
2. The server at that IP returns the webpage to be rendered in the browser (step 10).

## What is a DNS resolver?

The DNS resolver is the first stop in the DNS lookup, and it is responsible for dealing with the client that made the initial request. The resolver starts the sequence of queries that ultimately leads to a URL being translated into the necessary IP address.

Note: A typical uncached DNS lookup will involve both recursive and iterative queries.

It's important to differentiate between a [recursive DNS](https://www.cloudflare.com/learning/dns/what-is-recursive-dns/) query and a recursive DNS resolver. The query refers to the request made to a DNS resolver requiring the resolution of the query. A DNS recursive resolver is the computer that accepts a recursive query and processes the response by making the necessary requests.



## What are the types of DNS queries?

In a typical DNS lookup three types of queries occur. By using a combination of these queries, an optimized process for DNS resolution can result in a reduction of distance traveled. In an ideal situation cached record data will be available, allowing a DNS name server to return a non-recursive query.

#### 3 types of DNS queries:

1. **Recursive query** - In a recursive query, a DNS client requires that a DNS server (typically a DNS recursive resolver) will respond to the client with either the requested resource record or an error message if the resolver can't find the record.
2. **Iterative query** - in this situation the DNS client will allow a DNS server to return the best answer it can. If the queried DNS server does not have a match for the query name, it will return a referral to a DNS server authoritative for a lower level of the domain namespace. The DNS client will then make a query to the referral address. This process continues with additional DNS servers down the query chain until either an error or timeout occurs.
3. **Non-recursive query** - typically this will occur when a DNS resolver client queries a DNS server for a record that it has access to either because it's authoritative for the record or the record exists inside of its cache. Typically, a DNS server will cache DNS records to prevent additional bandwidth consumption and load on upstream servers.

## What is DNS caching? Where does DNS caching occur?

The purpose of caching is to temporarily stored data in a location that results in improvements in performance and reliability for data requests. DNS caching involves storing data closer to the requesting client so that the DNS query can be resolved earlier and additional queries further down the DNS lookup chain can be avoided, thereby improving load times and reducing bandwidth/CPU consumption. DNS data can be cached in a variety of locations, each of which will store DNS records for a set amount of time determined by a [time-to-live (TTL)](https://www.cloudflare.com/learning/cdn/glossary/time-to-live-ttl/).

#### Browser DNS caching

Modern web browsers are designed by default to cache DNS records for a set amount of time. The purpose here is obvious; the closer the DNS caching occurs to the web browser, the fewer processing steps must be taken in order to check the cache and make the correct requests to an IP address. When a request is made for a DNS record, the browser cache is the first location checked for the requested record.

In Chrome, you can see the status of your DNS cache by going to chrome://net-internals/#dns.

#### Operating system (OS) level DNS caching

The operating system level DNS resolver is the second and last local stop before a DNS query leaves your machine. The process inside your operating system that is designed to handle this query is commonly called a “stub resolver” or DNS client. When a stub resolver gets a request from an application, it first checks its own cache to see if it has the record. If it does not, it then sends a DNS query (with a recursive flag set), outside the local network to a DNS recursive resolver inside the Internet service provider (ISP).

When the recursive resolver inside the ISP receives a DNS query, like all previous steps, it will also check to see if the requested host-to-IP-address translation is already stored inside its local persistence layer.

The recursive resolver also has additional functionality depending on the types of records it has in its cache:

1. If the resolver does not have the [A records](https://www.cloudflare.com/learning/dns/dns-records/dns-a-record/), but does have the [NS records](https://www.cloudflare.com/learning/dns/dns-records/dns-ns-record/) for the authoritative nameservers, it will query those name servers directly, bypassing several steps in the DNS query. This shortcut prevents lookups from the root and .com nameservers (in our search for example.com) and helps the resolution of the DNS query occur more quickly.
2. If the resolver does not have the NS records, it will send a query to the TLD servers (.com in our case), skipping the root server.
3. In the unlikely event that the resolver does not have records pointing to the TLD servers, it will then query the root servers. This event typically occurs after a DNS cache has been purged.