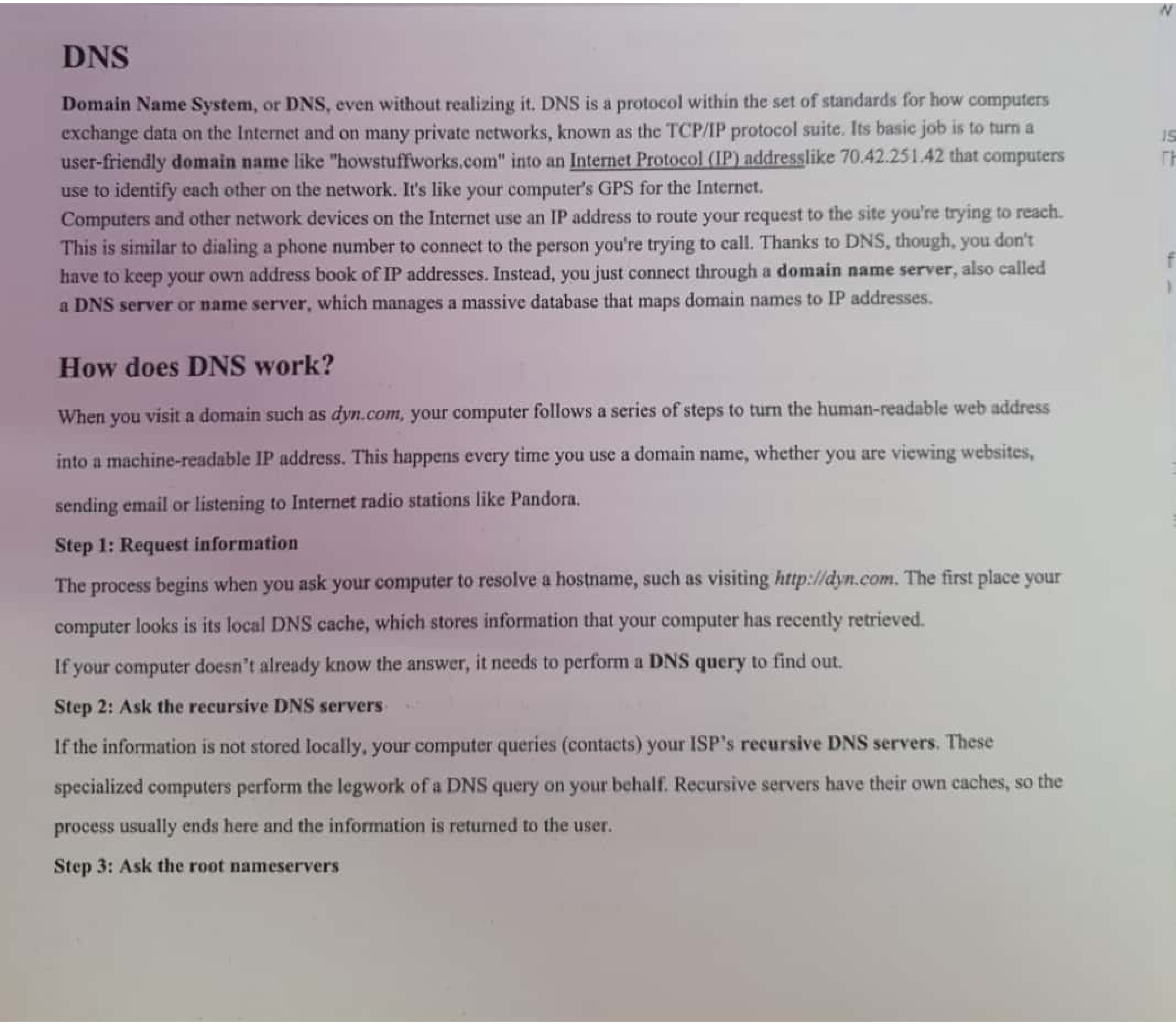


DNS

Domain Name System, or **DNS**, even without realizing it. DNS is a protocol within the set of standards for how computers exchange data on the Internet and on many private networks, known as the TCP/IP protocol suite. Its basic job is to turn a user-friendly **domain name** like "howstuffworks.com" into an Internet Protocol (IP) addresslike 70.42.251.42 that computers use to identify each other on the network. It's like your computer's GPS for the Internet.

Computers and other network devices on the Internet use an IP address to route your request to the site you're trying to reach. This is similar to dialing a phone number to connect to the person you're trying to call. Thanks to DNS, though, you don't have to keep your own address book of IP addresses. Instead, you just connect through a **domain name server**, also called a **DNS server** or **name server**, which manages a massive database that maps domain names to IP addresses.

How does DNS work?

When you visit a domain such as *dyn.com,* your computer follows a series of steps to turn the human-readable web address into a machine-readable IP address. This happens every time you use a domain name, whether you are viewing websites, sending email or listening to Internet radio stations like Pandora.

Step 1: Request information

The process begins when you ask your computer to resolve a hostname, such as visiting *http://dyn.com*. The first place your computer looks is its local DNS cache, which stores information that your computer has recently retrieved.

If your computer doesn't already know the answer, it needs to perform a **DNS query** to find out.

Step 2: Ask the recursive DNS servers

If the information is not stored locally, your computer queries (contacts) your ISP's **recursive DNS servers**. These specialized computers perform the legwork of a DNS query on your behalf. Recursive servers have their own caches, so the process usually ends here and the information is returned to the user.

Step 3: Ask the root nameservers

If the recursive servers don't have the answer, they query the **root nameservers**. A **nameserver** is a computer that answers questions about domain names, such as IP addresses. The thirteen root nameservers act as a kind of telephone switchboard for DNS. They don't know the answer, but they can direct our query to someone that knows where to find it.

Step 4: Ask the TLD nameservers

The root nameservers will look at the first part of our request, reading from right to left — *www.dyn.**com*** — and direct our query to the **Top-Level Domain (TLD) nameservers** for *.com*. Each TLD, such as *.com*, *.org*, and *.us*, have their own set of nameservers, which act like a receptionist for each TLD. These servers don't have the information we need, but they can refer us directly to the servers that *do* have the information.

Step 5: Ask the authoritative DNS servers

The TLD nameservers review the next part of our request — *www.**dyn**.com* — and direct our query to the nameservers responsible for this *specific* domain. These **authoritative nameservers** are responsible for knowing all the information about a specific domain, which are stored in **DNS records**. There are many types of records, which each contain a different kind of information. In this example, we want to know the IP address for *www.dyndns.com*, so we ask the authoritative nameserver for the **Address Record (A)**.

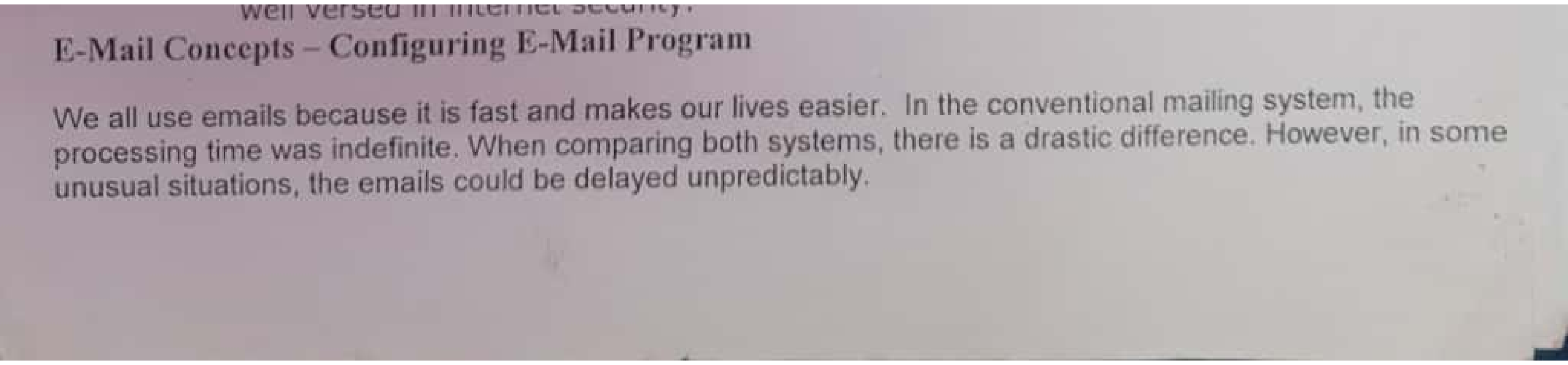
Step 6: Retrieve the record

The recursive server retrieves the A record for *dyn.com* from the authoritative nameservers and stores the record in its local cache. If anyone else requests the host record for *dyn.com*, the recursive servers will already have the answer and will not need to go through the lookup process again. All records have a **time-to-live** value, which is like an expiration date. After a while, the recursive server will need to ask for a new copy of the record to make sure the information doesn't become out-of-date.

Step 7: Receive the answer

Armed with the answer, recursive server returns the A record back to your computer. Your computer stores the record in its cache, reads the IP address from the record, then passes this information to your browser. The browser then opens a connection to the webserver and receives the website.

This entire process, from start to finish, takes only milliseconds to complete.



well versed in internet security.

E-Mail Concepts – Configuring E-Mail Program

We all use emails because it is fast and makes our lives easier. In the conventional mailing system, the processing time was indefinite. When comparing both systems, there is a drastic difference. However, in some unusual situations, the emails could be delayed unpredictably.

Knowing how the email system works is important to all the email users. However, most of us are unaware of what goes on behind the scenes. Learning this will help to figure out the status of your emails sent that ended up in an error or with a bounce back message.

What is email?

The term "email" stands for "electronic mail". The electronic mail is introduced first in the 1960s, however it became available in the current structure in the 1970s. Let us take a look at how email actually works.

Protocols used in email systems

The email communication is done via three protocols in general. They are listed below.

- IMAP
- POP
- SMTP

IMAP

The IMAP stands for Internet Mail Access Protocol. This protocol is used while receiving an email. When one uses IMAP, the emails will be present in the server and not get downloaded to the user's mail box and deleted from the server. This helps to have less memory used in the local computer and server memory is increased.

POP

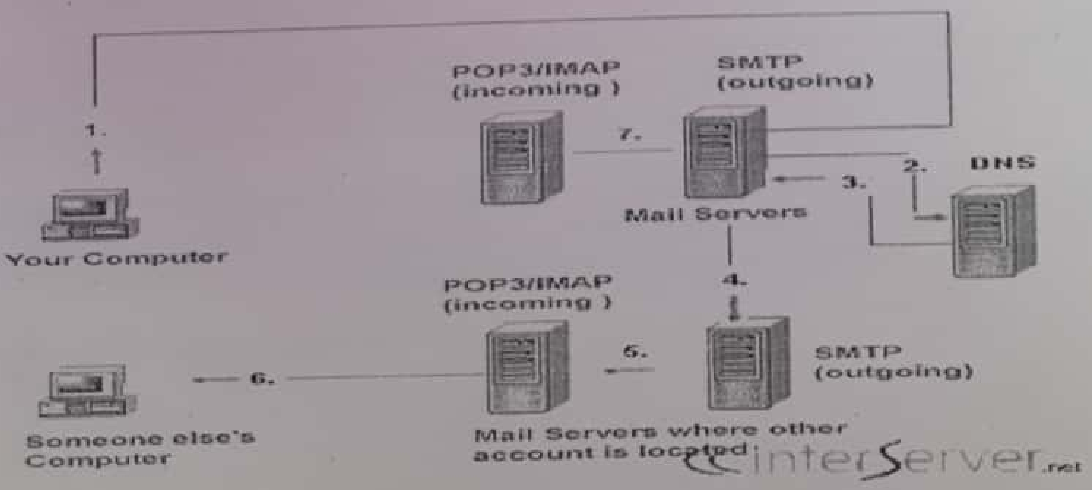
The POP stands for Post Office Protocol. This protocol is also used for incoming emails. The main difference with the both protocols is that POP downloads the entire email into the local computer and deletes the data on the server once it is downloaded. This is helpful in a server with less free memory. Current version of POP is POP3.

SMTP

The SMTP stands for Simple Mail Transfer Protocol. Email is sent using this protocol.

How does email work?

The diagram down below describes the path that email takes from your computer to the intended recipient . This shows the path of the email from sending to receiving ends. There are also many logical machines in the email delivery process. Please have a look at the diagram before proceeding.



onfiguring E-mail Program

Email is used for communicating by "mail" with other people on the Internet. There are many e-mail programs currently being used on the Internet, please note that our Customer Service Representatives are versed in using Outlook, Netscape Mail, and Internet Mail and may not have information on how to configure/use other E-mail programs.

When setting up your e-mail program(s), the following settings will most likely be used when configuring the program.

1. **Incoming (POP3) Server:** your domain.com (NOTE: DO NOT put 'www' or 'pop', etc. in front of the domain!)
2. **Outgoing (SMTP) Server:** your domain.com (NOTE: DO NOT put 'www' or 'smtp', etc. in front of the domain! The exception is with all Fully Managed Windows Servers, excluding the Business/Reseller III, you will need to use mail.yourdomain.com)***Some Internet Access Providers require you to use their SMTP server.*
3. **POP3 account/user name:**Fully Managed Linux - youraccountname / Fully Managed Windows - youraccountname@yourdomain.com (This would be the name of the email account you created)
4. **POP3 account/user password:**This is the password for the POP email account that you have created.

If you are hosting your email at a different company, you must consult their documentation for the settings mentioned below.

- **Step** 1 — Add the **email** address to the panel. ...
- **Step** 2 — Locate your username and password. ...
- **Step** 3 — Locate your **mail** servername. ...
- **Step** 4 — Set secure ports and choose IMAP or POP.

Fighting spam

Spam is unsolicited email that may be delivered your address. It may contain advertising, "chain letters", computer viruses, or even be a phishing attempt. Address databases are created by spammers using dedicated programs that pick addresses using a special dictionary or just collecting addresses posted publicly.

Unsolicited mail should be differentiated from honest mail. Proper mailing lists usually imply explicit consent. You can also unsubscribe to stop receiving such emails.

Yandex.Mail uses the "Anti-Spam" service, which learns from user-submitted spam complaints, to recognize spam. It puts suspicious email with spam features in the Spam folder.

Spamming refers to use of an electronic messaging system to send unsolicited messages especially advertising messages to a group of recipients. Unsolicited messages mean the recipient did not grant permission for those message to be sent.

Anti-spam refers to the use of any software, hardware or process to block spam from entering a system. The anti-spam software uses a set of protocols to determine unsolicited and unwanted messages and prevent those messages from getting to a user's inbox.

Most of the **Anti-spam** solutions that are available today can be customized as per your needs, allowing only the approved emails into your inbox. Such software always presumes that all the incoming emails are spam, and only allow those, from the people you know, to come in.

What is Anti-spam and Benefits of Using Antispam Software?

- Blocking Spam
- Quarantining Spam

- Automatic Filter Updates
- Monitoring Multiple Accounts
- Your Personal Whitelist
- Reporting Spam

some of the benefits and features of the anti-spam software:

Blocking Spam

Certain anti-spam solutions not only block specific email addresses but also search for subject lines and text in the email messages. You can customize it to block incoming emails based on senders, and even if your email address is not in the recipient field.

Quarantining Spam

Anti-spam filters automatically quarantine the spam emails, ensuring your inbox is spam free. Such quarantined emails are held for a fixed number of days, say 30 days or so, and then dumped. During that period, you can check and recover any legitimate email that may have been quarantined.

Automatic Filter Updates

Most of the anti-virus software comes with automatic filter update feature for timely detection of new types of Malware threats. Automatic updates not only helps the anti-spam software to stay up-to-date, it also helps secure your system from new kinds of Malware.

Monitoring Multiple Accounts

With this feature, you can monitor and filter spam from multiple accounts. You can filter your home email from work email, and vice versa.

Your Personal Whitelist

Some anti-spam software allows you to maintain a 'friendly' list of people whose emails you wish to accept. These emails will never be mistaken for spam as against the blacklist of spammers. You can also update the list in the future.

Reporting Spam

Some anti-spam programs allow you to report spam back to the company supplying the program. It helps that company to develop new type of filters based on the analysis of the reported spam.

DNS

DNS, which stands for Domain Name System, is used as the medium to translate domain names to their respective IP addresses when a client initiates a request query. DNS stores the database of all the domain names and their IP addresses which are registered on the network. It can be thought of as an attendance register for various websites present over the internet. In the case of DNS, it maintains the database of all the websites Domain Names and their IP (Internet Protocol) addresses that are operational all over the world.

The Domain Name System (DNS) is a central part of the Internet, providing a way to match names (a website that you are looking for) to numbers (the address for the website). Anything connected to the Internet - laptops, tablets, mobile phones, websites - has an Internet Protocol (IP) address made up of numbers. Your favourite website might have an IP address like 64.202.189.170, but this is obviously not easy to remember However a domain name such as bestdomainnameever.com is something people can recognise and remember. DNS syncs up domain names with IP addresses enabling humans to use memorable domain names while computers on the Internet can use IP addresses.

History of DNS

The origins of DNS date back to the time of ARPANET, when there were only a few computers to get an entry in the database. A HOSTS.TXT file was maintained by Stanford Research Institute, which constituted the data of all the machines, and was copied by all the host machines to remain updated.

Jon Postel from the Information Sciences Institute requested Paul Mockapetris to design the very first implementation of DNS, at the University of California, Irvine, in 1983. Then in 1984, BIND (Berkeley Internet Name Domain) was created by four students, Douglas Terry, Mark Painter, David Riggle, and Songnian Zhou, for Unix machines. After some revisions made in 1985 by Kevin Dunlap, it was later ported to Windows machines and is still the most widely used DNS on the planet.

How DNS works?

To understand the basic working of DNS, let me guide you with an example of a hotel. Let us assume, you need to visit your friend at some hotel. Now, what will you do? You'll reach the hotel reception and ask the receptionist for the room number of your friend. In order to do so, you'll need to tell the name of your friend to the receptionist, who'll check the same in her database and tell you the room number of your friend. She'll also call your friend to confirm whether he is available or not. Now, try to relate the example to working of DNS. In this case, you're the client sending a request to a DNS server, the receptionist, and your friend's name is the domain name and his room number is his IP address. The receptionist will type your friend's name on her computer containing the database of all the guests, called the Domain Name Space, if your friend is staying in the hotel she'll tell you the room number, otherwise not.

Similar thing happens in working of DNS: when you type the website name in your browser, the browser sends a request to the DNS server, if the website domain name is registered in the database with the DNS, then it'll reply you with the IP address of the website you are trying to access, which is something like 117.234.214.14

Understanding the Domain Name and IP Address

Take the domain name, www.google.com. The naming convention moves from right to left and vice-versa for IP address. In the domain name for Google, first, the DNS will check for **com** which stands for the commercial domain, and is a top-level domain. Proceeding further, Google is a sub-domain to com, and subsequently, **www** is a sub-

Storing help files as HTML pages on your

domain to Google domain. The dot (.) is used to separate the domains from their sub-domains. The full domain can only consist of 253 characters.

Now, if someone wants to know the domain name registered against an IP address, he will request the DNS server with the IP address of the website. Say, the IP address sent is 31.13.79.246, the DNS will first check the 31 then 13 then 79 and finally 246, concluding that the IP address belongs to www.fb.com. The DNS resembles the hierarchy strucure of a tree, not the biological one, there is a different tree in computer data structures, in which the address 31 belongs to the top position of the tree and is the primary domain in the hierarchy, addresses 13, 79, 246 are consecutive sub-domains. The number 246 refers to the server machine hosting the website www.fb.com.

How DNS Works in 6 Easy Steps:

1. The user logs onto their Internet Service Provider (ISP) to use the Internet.
2. The user opens up a web browser (Firefox, Chrome, Internet Explorer, Safari, etc.) and types a URL into the address bar. For example, perhaps the user types in https://www.atlantic.net/.
3. The computer then asks for the ISP's DNS servers for the specific IP address for www.atlantic.net.
4. Once the DNS server that holds this specific IP address for www.atlantic.net is found, the DNS server responds with the appropriate IP address and the user's computer then gives this address to the user's browser.
5. The browser opens a connection to the server using the IP address provided and retrieves the page from the site requested, in this case for www.atlantic.net.
6. The browser displays the requested page on the computer screen.

From a technical standpoint, the Domain Name System (DNS) is an organized naming system for computers, services or any other resource that is connected to the Internet or private network. To put it a bit more simply, the DNS translates easily memorized domain names, like www.atlantic.net, into numbers, like 209.208.84.170. The DNS is a critical component of the Internet, as it provides a worldwide, keyword-based redirection service.

To streamline efficiency, each DNS server contains only a small portion of host name to IP address mappings. Instead, the server contains special programming code that tells it where to look for the rest of the information necessary to connect to the website.Many applications and processes utilize DNS services available, including the World Wide Web, email software and other applications like Skype. On February 13th, Atlantic.Net announced the availability of Cloud Server DNS. This feature is designed to increase efficiency by offering self-service DNS record management backed by a geographically diverse, redundant and replication Cloud Server architecture. If you are interested in learning more about Atlantic.Net's Cloud Hosting DNS, contact us today at 1-800-422-2936.

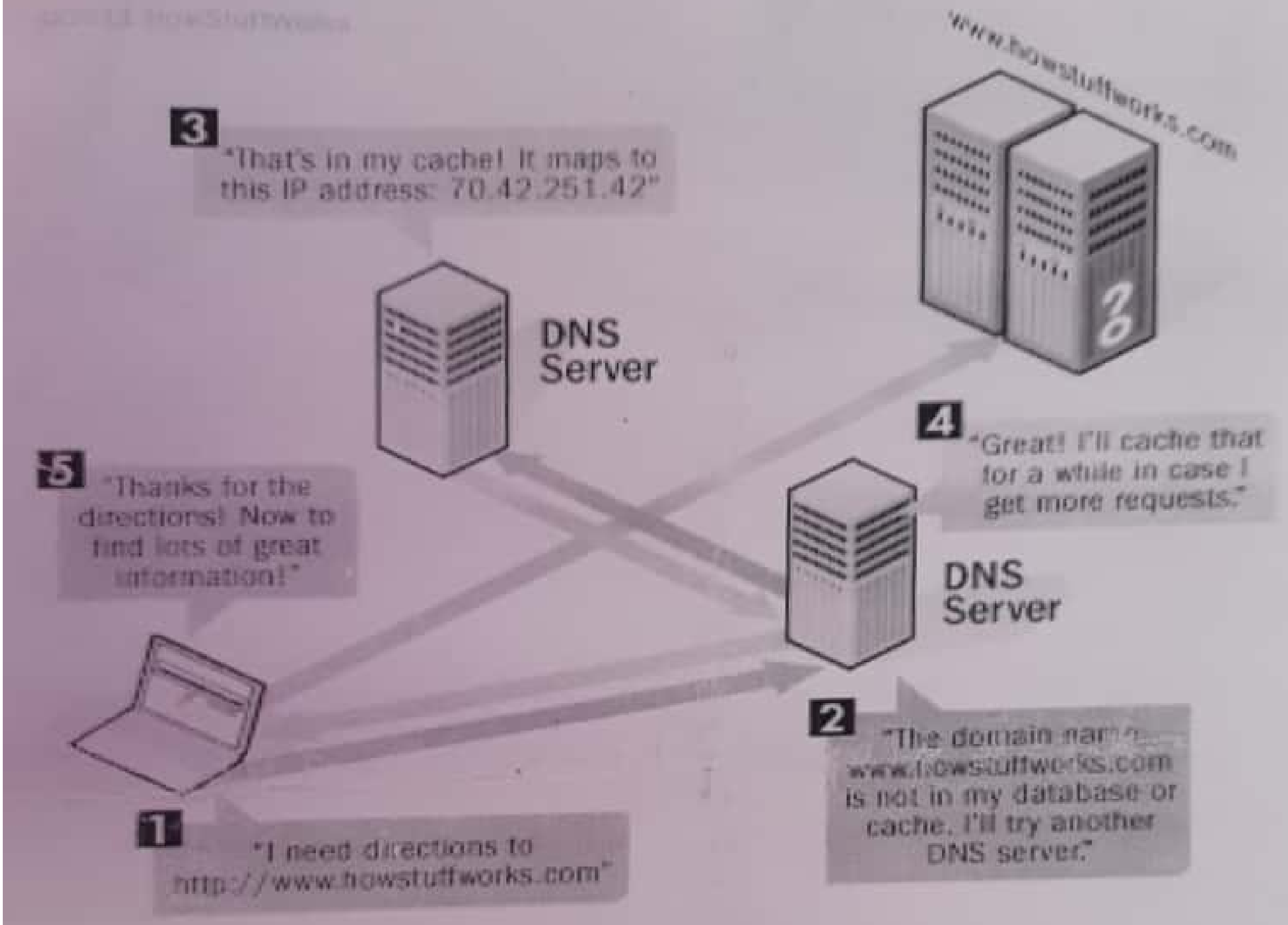
How DNS Works

The first step is to register you domain name. To put it simply, you buy it so that you are the one who will be able to make the association name <-> address. Once you are the proud owner of the domain name, you have access to what is called the "DNS zone", which is basically a list of pairs name <-> IP. It's a list because when you own a domain name, you can also use sub-domains (mail.google.com, plus.google.com, ...), each of them can be associated to a different address. Once you have configured the DNS zone, it is stored on your registar's server.

What is left to explain is how the queries are made. How does your computer knows the IP address for a given name ? When you type www.google.com in your browser, the browser starts by asking the translation to a root **nameserver.** There are 13 of them, and they are the entry points of the DNS protocol.

When they receive a request, they usually don't know the answer (as there are too many domain names) but they know **who to ask.** For example, they know that www.google.com depends on the "com" DNS server (which handles the domain names with the .com TLD). So the request in then sent to this server, which will in its turn know who to ask, and this continues until the request is sent to your registar's server, which will know the answer your computer is looking for.

Now in reality, this is not the real process. As you can imagine, the number of requests is overwhelmingly huge, and the 13 servers would create a bottleneck a lot too small for every request. There are complex caching systems used to he pressure on these servers and they are, in fact, involved in only a small part of the traffic. ing DNS servers will store the answer to DNS queries for a given time, and answer directly if the same query ives again in this time frame. In practice, your computer will ask these caching servers, and the caching server will k the authoritative servers for the answer (if it's not in the cache).


©2003 HowStuffWorks
www.howstuffworks.com
3
"That's in my cache! It maps to this IP address: 70.42.251.42"
DNS Server
4
"Great! I'll cache that for a while in case I get more requests."
5
"Thanks for the directions! Now to find lots of great information!"
DNS Server
2
"The domain name www.howstuffworks.com is not in my database or cache. I'll try another DNS server."
1
"I need directions to http://www.howstuffworks.com"

When you enter a URL into your Web browser, your DNS server uses its resources to resolve the ame into the IP address for the appropriate Web server.

How does DNS works?

1. You type a domain name such as google.com into your browser using client computer operating system such as Windows or Apple OS ("client").
2. The client needs to find the IP address where google.com search engine is located on the earth (typically all websites are hosted in the Internet data center).
3. Your browser will send this query to the operating system.
4. Each operating system is configured to query certain dns servers. Typically your ISP or network administrator configures such dns servers called Resolving Name Server.
5. The resolving name server does not aware of the location of the google.com, but it does know where the root servers are located.
6. Next, the resolving name server find the location of the top-level domain name server to send query for google.com. Each domain on the Internet has authoritative name server.
7. Finally, the authoritative name server will give you exact IP address of google.com. This information will come back to to the resolving name server, which caches the information and send backs an answer (answer to your query what is IP address of google.com) to the browser to the correct place. The end result you will see google search engine home page.