DOI: 10.24412/2181-144X-2025-1-53-59

DOMAIN NAME SYSTEM AND ITS WORKING PRINCIPLE

Farmonov Bobur [1]✉[0009-0009-9389-4296], ***Berdimurodov Mansur*** [2]✉[0000-0002-3031-5009]

[1]*Senior Lecturer of the Department of Information Security, National University of Uzbekistan, E-mail: boburfarmonov93@mail.ru*
[2]*PhD, Senior Lecturer, Department of Modern Information and Communication Technologies, International Islamic Academy of Uzbekistan, E-mail: m.berdimurodov@iiau.uz*

Annotatsiya. Ushbu maqolada Internetning boshlang‘ich IP manzillarini aniq kompyuterlarga bog‘lash, tarmoqga ko‘proq qurilmalar va odamlar qo‘shish tahlili. Hozirgi kunda muayyan IP manzilini brauzerga kiritish orqali veb-saytga kirish mumkin bo‘lsada, foydalanuvchilar yodda saqlash oson bo‘lgan veb-sayt nomlariga bog‘lashni amalga oshirish jarayoni hamda domen nomi tizimi (DNS) va uning qanday ishlashi haqida ma’lumot beradi. DNS tizimining tarkibiy qismlari va bajarilish ketma-ketligi ko‘rsatilgan. Domen nomi tizimi (DNS) domen nomlarini IP manzillariga aylantiradi, bu esa brauzerlar internet sahifalarini yuklash uchun foydalanadigan ma’lumotlardir. Internetga ulangan har bir qurilmaning o‘z IP manzili bor, bu boshqa qurilmalarga uni topish uchun ishlatiladi. Foydalanuvchilar brauzerlarining URL satriga domen nomlarini kiritganda, DNS serverlari ushbu domen nomlarini raqamli IP manzillariga tarjima qilish uchun javobgardir, bu esa ularni to‘g‘ri veb-saytgacha olib boradi. DNS serverlari internet foydalanuvchilarning brauzerlariga iiau.uz kabi oddiy qisqartma so‘zlarni kiritishiga imkon beradi, bu esa har bir veb-saytning IP manzilini eslab qolmaslik uchun juda muxum.

Kalit so‘zlar: DNS, DNS yozuv turlari, DNS server turlari, Rekursiv server, Asosiy nom serveri, Yuqori darajadagi domen serveri (TLD), Avtorizatsiya qilingan nom serveri, Rekursiv so‘rov, Iterativ so‘rov, Subdomenlar.

Аннотация. В данной статье рассматривается привязка начальных IP-адресов Интернета к конкретным компьютерам, а также анализ добавления большего количества устройств и пользователей в сеть. В настоящее время, хотя можно получить доступ к веб-сайту, введя определённый IP-адрес в браузере, пользователи стремятся к тому, чтобы связывать веб-сайты с запоминаемыми названиями. Также рассматривается система доменных имен (DNS) и то, как она работает. Показаны составные части системы DNS и последовательность выполнения. Система доменных имен (DNS) преобразует доменные имена в IP-адреса, которые используются браузерами для загрузки веб-страниц. У каждого устройства, подключенного к Интернету, есть свой IP-адрес, который используется другими устройствами для его нахождения. Когда пользователи вводят доменные имена в адресную строку браузера, DNS-серверы отвечают за перевод этих доменных имен в цифровые IP-адреса, что приводит их к нужному веб-сайту. DNS-серверы позволяют пользователям Интернета вводить простые сокращенные слова, такие как iiau.uz, в их браузеры, что очень важно, чтобы не запоминать IP-адреса каждого веб-сайта.

Ключевые слова: Система доменных имен, Типы записей DNS, Типы серверов DNS, Рекурсивный сервер, Корневой сервер имен, Сервер верхнего уровня (TLD), Авторитетный сервер имен, Рекурсивный запрос, Итеративный запрос, Субдомены.

Annotation. This article discusses the linkage of initial IP addresses of the Internet to specific computers, as well as the analysis of adding more devices and users to the network. Currently, while it is possible to access a website by entering a specific IP address in the browser, users strive to associate websites with memorable names. The article also examines the Domain Name System (DNS) and how it works. The components of the DNS system and the sequence of execution are presented. The Domain Name System (DNS) converts domain names into IP addresses, which are used by browsers to load web pages. Every device connected to the Internet has its own IP address, which is used by other devices to locate it. When users enter domain names into the browser's address bar, DNS servers are responsible for translating these domain names into digital IP addresses, guiding them to the correct website. DNS servers allow Internet users to enter simple abbreviated words, such as iiau.uz, in their browsers, which is very important to avoid remembering the IP addresses of every website.

Key words: DNS, DNS record types, DNS server types, Recursive server, Root name server, TLD server, Authoritative name server, Recursive query, Iterative query, Subdomains.

Introduction

Any time being online, all internet users using the Domain Name System (DNS) whether they realize it or not! When we're online, we typically rely on website names, E_mail addresses, or search engines to find what we need and communicate successfully. However, computers operate differently, communicating with each other using a system of numbers known as IP addresses. Finding and remembering a string of random numbers for every website we want to visit would be near-impossible for us. That's where DNS comes in handy. DNS translates human-readable website names into IP addresses, allowing us to visit websites, send E_mails, and book flights by remembering just a catchy address, like gcore.com, instead of a string of numbers, like 92.223.84.84. DNS translates user-friendly website names, like www.gcore.com, into numerical IP addresses that computers use to communicate with each other, like 92.223.84.84 or 2a03:90c0:9994::9994. Both your device and the website you want to view have numbers that need to connect [1-3].

When the Internet started, it was easier for people to correspond to specific IP addresses with specific computers, but that didn't last long as more devices and people joined the growing network. While it's still possible to type a particular IP address into a browser and reach a website, users wanted website names that would be easier to remember. When the Internet started, Stanford's Elizabeth Feinler personally assigned those names and addresses in a master list of every Internet-connected computer. This text file was called "hosts.txt". As the Internet grew to millions of domains, this was not sustainable. In 1983, Paul Mockapetris, a USC researcher, was tasked with developing a solution. His solution was a new system that he named DNS, which remains based on Mockapetris' fundamental principles. Today, the standards for DNS are maintained by the Internet Engineering Task Force (IETF) in RFC 1035[4,5].

There are many DNS record types, each with their own purpose in denoting how a query should be treated. Common DNS records are the following:

SOA Record: A start of authority record (abbreviated as SOA record) is a type of resource record in the Domain Name System (DNS) containing administrative information about the zone, especially regarding zone transfers.

A Record: The A record is the most basic type of DNS record. It's used to map a domain name to an IPv4 address, and it's the only record type that's required for a website to work.

AAAA Record: The A record is the most basic type of DNS record. It's used to map a domain name to an IPv6 address.

NS record: These name server records denote which authoritative server is responsible for having all the information about a given domain. Often, domains have both primary and backup name servers to increase reliability, and multiple NS records are used to direct queries to them.

TXT record: TXT records enable administrators to enter text into DNS. The original purpose was to put human-readable notes in DNS, but today, machine-readable notes are often put there. TXT records are used to confirm domain ownership, secure E_mail and counter E_mail spam.

CNAME record: Canonical name records are used instead of an A record when there is an alias. They are used to retry the query of the same IP address with two different domains. An example would be in the URL searchsecurity.techtarget.com, where the CNAME would query techtarget.com.

MX Record: An MX record is used to route E_mail messages for a domain. When someone sends an E_mail to "example@example.com", their E_mail server will look up the MX records for the "example.com" domain and route the message accordingly.

PTR Record: A PTR record is used to map an IP address to a domain name. This type of record is mostly used for reverse DNS lookups.

There are several server types involved in completing a DNS resolution. The following list describes the four name servers in the order a query passes through them. They provide the domain name being sought or referrals to other name servers [6-8].

Recursive server. The recursive server takes DNS queries from an application, such as a web browser. It is the first resource the user accesses and either provides the answer to the query if it has it cached or accesses the next-level server if it doesn't. This server may go through several iterations of querying before returning an answer to the client.

Root name server. This server is the first place the recursive server sends a query if it doesn't have the answer cached. The root name server is an index of all the servers that will have the information being queried. These servers are overseen by the Internet Corporation for Assigned Names and Numbers, specifically a branch of ICANN called the Internet Assigned Numbers Authority.

TLD server. The Top-Level Domain nameserver is a DNS server function that is responsible for storing and managing information about domain names that use a specific top-level domain (TLD). A TLD is the far end of a domain name, such as.com, org, online, and .net. If your query is to find the IP address of hostinger.com, the root nameserver will redirect the DNS recursive resolver to the .com TLD nameserver. Next, the TLD nameserver will inform the resolver about the location of the matching IP address at a specific authoritative nameserver

Authoritative name server. The authoritative name server is the final checkpoint for the DNS query. These servers know everything about a given domain and deal with the subdomain part of the domain name. These servers contain DNS resource records with specific information about a domain, such as the A record. They return the necessary record to the recursive server to send back to the client and cache it closer to the client for future lookups.

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.

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.

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.

It is essential to know about Fully Qualified Domain Name (FQDN), to understand the DNS hierarchy. A FQDN is the domain name that specifies its exact location in the DNS hierarchy. It specifies all domain levels including the top-level domain and the root zone. It consists of two parts, the host name and the domain name. An example of FQDN in a mail server is “mail.mydomain.com” where “mail” is the host name and the “mydomain.com” is the domain name. A fully qualified domain name is supposed to have little ambiguity. FQDN is otherwise called an absolute domain name [9-12].

A domain is structured into different parts, separated by dots. Each part has a specific purpose and contributes to the overall hierarchical structure of the domain name. DNS hierarchy is comprised of the following four levels (Fig. 1):

1. Root Level Domain
2. Top Level Domains (TLD)
3. Second Level Domains (SLD)
4. Subdomains

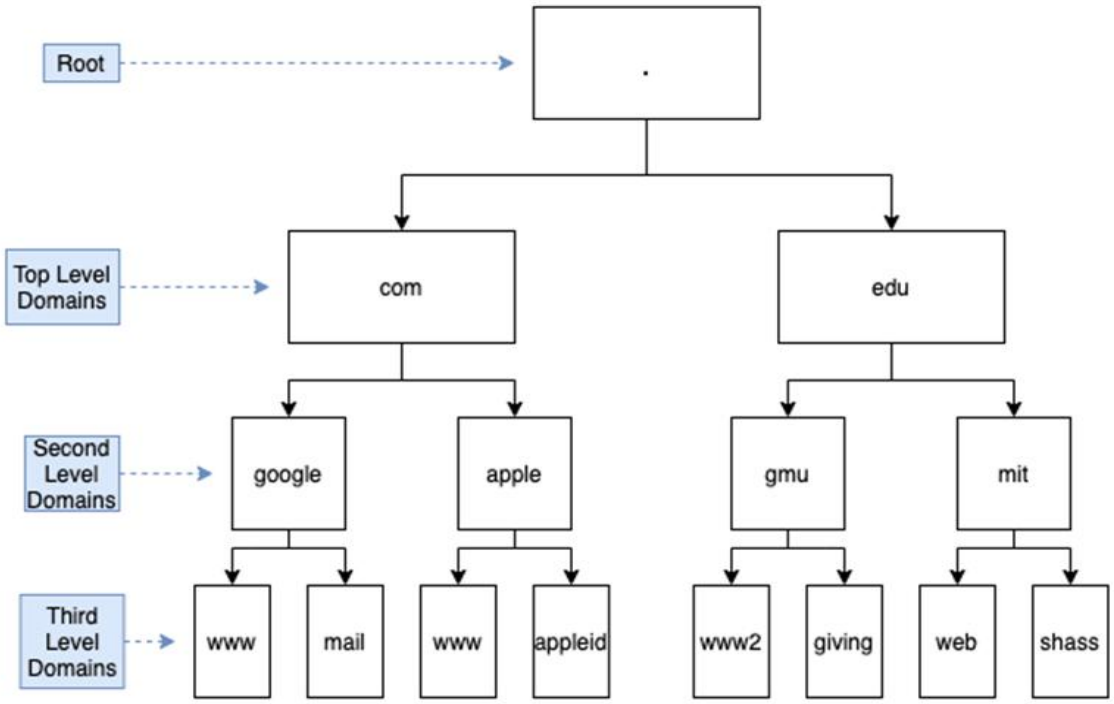

Fig.1. Domain name levels.

The DNS hierarchy originates at the Root Level, housing the DNS root zone managed by authoritative root name servers. These servers are pivotal in redirecting requests to the appropriate Top-Level Domain (top level domains) name servers, marking the commencement of the translation process from human-readable domain names to IP addresses. The Root Level represents the apex of the domain naming system, playing a foundational role in facilitating the initial step in the domain resolution process. The DNS is a hierarchical naming system for computers, services, or any resource participating in the Internet. The top of that hierarchy is the root domain. The root domain does not have a formal name and its label in the DNS hierarchy is an empty string. All fully qualified domain names (FQDNs) on the Internet can be regarded as ending with this empty string for the root domain, and therefore ending in a full stop character (the label delimiter), e.g., "www.example.com.". This is generally implied rather than explicit, as modern DNS software does not actually require that the terminating dot be included when attempting to translate a domain name to an IP address.

The root domain contains all top-level domains of the Internet. As of July 2015, it contained 1058 TLDs, including 730 generic top-level domains (gTLDs) and 301 country code top-level domains (ccTLDs) in the root domain. In addition, the ARPA domain is used for technical name spaces in the management of Internet addressing and other resources. A TEST domain is used for testing internationalized domain names.

Situated just below the Root Level, Top-Level Domains (top level domains) constitute the subsequent tier in the DNS hierarchy. Examples include widely recognized extensions such as .com, .net, and .org, each reflecting organizational hierarchy or geographic distinctions. Top level domain name servers exert authority over these domain extensions, offering critical information regarding Second Level Domains within their respective realms. Top level domains are integral components that contribute to the hierarchical structure of domain names, signifying organizational or geographic affiliations.

- “com” for commercial websites.
- “org” for organizational websites.

• “edu” for educational websites.
• “net” for network organizations.
• “gov” for governmental websites.
• “mil” for military websites.

Second Level Domain. Directly beneath top level domains, Second Level Domains (second level domains) form the subsequent layer in the DNS hierarchy. These domains are specific to organizations or entities and serve as primary identifiers within web addresses.

Sub Domain Within the structure of Second Level Domains, the DNS hierarchy further extends to Sub-Domains. These Sub-Domains allow for additional organizational structuring of a website, enhancing flexibility in design and content management. An illustrative example is found in blog.example.com, where “blog” serves as a Sub-Domain of “example.com.” Sub-Domains provide a means of categorizing content under broader domains, contributing to effective organization and management. Total number of subdomains may be up to 127, Each subdomain may contain from 0 to 63 bytes. The full domain name may not exceed a total length of 253 ASCII characters in its textual representation.

Working principle of dns lookup. DNS lookup is when a DNS resolver asks DNS servers to find the IP address or related information of a domain name. When you enter a domain name in your web browser (or any other internet application,) the DNS resolver starts a DNS lookup to query the domain name into its matching IP address, giving you access to the desired content (Fig.2).

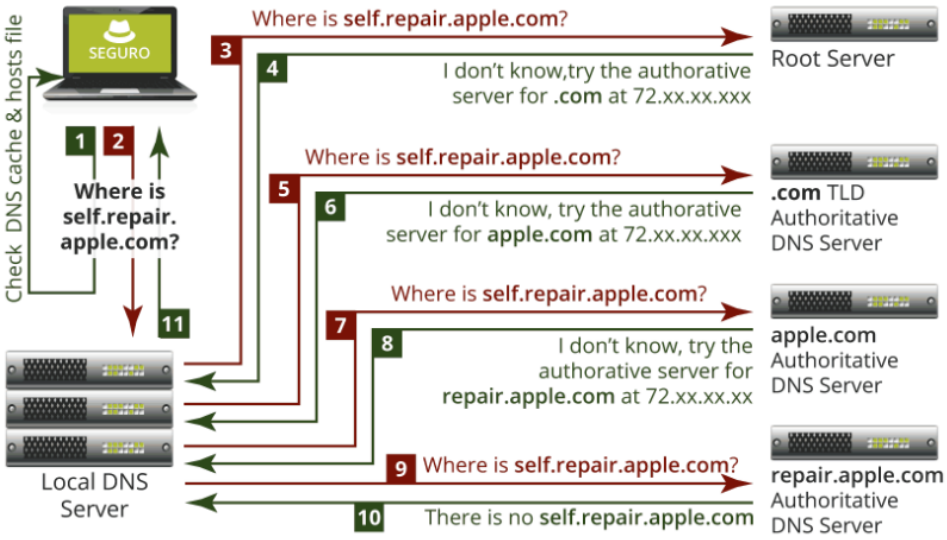

Fig.2. DNS lookup queries sequence.

1. DNS query initiated: When you enter a domain name in your web browser (or any application requiring internet access,) your device initiates a DNS query to find the corresponding IP address.

2. DNS resolver cache check: The DNS resolver first checks its local cache—the DNS cache—to see if it recently resolved the same domain name. If the information is found in the cache, it can provide the IP address directly without the need for further queries to the name servers, we can skip to step 6.

3. Query to hosts file: If the domain information is not found in the DNS cache. The DNS resolver checks hosts file in the system, which is located in C:\Windows\system32\drivers\etc. If the information is found in the cache, it can provide the IP address directly without the need for further queries to the name servers, we can skip to step 6.

4. Query root name servers: If the domain information is not found in the DNS cache and hosts.txt file, the DNS resolver queries the root server. The root name server then responds to the resolver's query with the TLD name server responsible for the specific domain extension. For this one it is "example.com," so the TLD is "com."

5. Contact TLD name servers: The DNS resolver then queries the TLD name servers to obtain the authoritative name servers addresses responsible for the queried domain (e.g., "example.com.")

6. Query authoritative name server: The DNS resolver sends a query to one of the authoritative name servers to obtain the IP address associated with the domain name. The authoritative name servers respond to the DNS resolver with the IP address.

7. Establish connection: Now that the DNS resolver has obtained the IP address "192.0.2.1", it sends it back to the user's browser. The resolver will also store this information in the DNS cache respecting the TTL (time to live), which was provided as a part of the authoritative answer. With the IP address, the computer/device can connect to the appropriate server. The web content is then delivered to the device, allowing the user to access the website.

DNS propagation is the process of updating DNS records across the Internet. When you make a change to a DNS record, that change needs to propagate to all of the DNS servers around the world so that everyone can see the new record.

The time it takes for DNS changes to propagate can vary depending on a few factors, but it's generally pretty quick. For most people, DNS changes will propagate within an hour or two. In some cases, it may take up to 24 hours for the changes to fully propagate. If you're making a major change to your website (like changing your web hosting provider), you should plan ahead and make the DNS changes at least a day in advance. This way, you can be sure that everyone will be able to see your new website as soon as the DNS changes have propagated.

The TTL is a value that's set in a DNS record. It tells DNS servers how long they should cache the record. For example, if a DNS record has a TTL of 24 hours, that means that any DNS server that looks up that record can cache it for up to 24 hours. The TTL is important because it controls how often DNS servers need to check for changes to DNS records. If you make a change to a DNS record, you need to wait for the TTL to expire before that change will propagate to all of the DNS servers around the world.

Conclusion

In conclusion, in this article we have considered the components and the principle of operation of the DNS system. DNS plays a critical role in the functioning of the internet, translating human-readable domain names into numerical IP addresses, allowing seamless communication between devices and access to online services. However, DNS management comes with risks, such as DNS attacks that can disrupt services and compromise data security. And we are showed how DNS system is essential for our daily internet usage.

List of references:

[1]. Kabulov A. V., Berdimurodov M. A., Saymanov I. M. Kriptografik algoritm mikrobuyruqlarining mantiqiy bul funksiya shakli (AES, EL-GAMAL) //2021, No. 3 (127/1). – 2023. – T. 127. – №. 101. – C. 5-16.

[2]. Berdimurodov M., Baizhumanov A. Algorithms for minimizing functions of the algebra of logic in the class of disjunctive normal forms and estimating their complexity

//2022 International Conference on Information Science and Communications Technologies (ICISCT). – IEEE, 2022. – С. 1-5.

[3]. Ayodeji, A., Di Buono, A., Pierce, I., & Ahmed, H. (2024). Wavy-attention network for real-time cyber-attack detection in a small modular pressurized water reactor digital control system. Nuclear Engineering and Design, 424.

[4]. Liang, Z., Zheng, Z., Chen, W., Pei, Z., Wang, J., & Chen, J. (2024). A novel deep transfer learning framework integrating general and domain-specific features for EEG-based brain–computer interface. Biomedical Signal Processing and Control, 95.

[5]. Saini, H., Mehra, H., Rani, R., Jaiswal, G., Sharma, A., & Dev, A. (2024). Enhancing cyberbullying detection: a comparative study of ensemble CNN–SVM and BERT models. Social Network Analysis and Mining, 14(1).

[6]. Wu, J., Yang, R., Zhao, P., & Yang, L. (2024). Computer-aided mobility solutions: Machine learning innovations to secure smart urban transportation. Sustainable Cities and Society, 107.

[7]. Shuhan, W., Chengzhi, Y., Xiaoxiao, L., & Siyu, W. (2024). Smart infrastructure design: Machine learning solutions for securing modern cities. Sustainable Cities and Society, 107.

[8]. Mendoza-Bernal, J., González-Vidal, A., & Skarmeta, A. (2024). A Convolutional Neural Network approach for image-based anomaly detection in smart agriculture. Expert Systems with Applications, 247.

[9]. Jain, Research trends, themes, and insights on artificial neural networks for smart cities towards SDG-11, Journal of Cleaner Production https://doi.org/10.1016/j.jclepro.2023.137300

[10]. Yao, Differential privacy in edge computing-based smart city Applications: Security issues, solutions and future directions, Array https://doi.org/10.1016/j.array.2023.100293

[11]. Jia, Artificial intelligence enabled cyber security defense for smart cities: A novel attack detection framework based on the MDATA model, Knowledge-Based Systems https://doi.org/10.1016/j.knosys.2023.110781

[12]. Heidari, Applications of ML/DL in the management of smart cities and societies based on new trends in information technologies: A systematic literature review, Sustainable Cities and Society https://doi.org/10.1016/j.scs.2022.104089