# CECS 572 - EXAM 2

**Student Name – Aishwarya Bhavsar**

**Student ID – 029371509** ***I certify that this is my original work - AB***

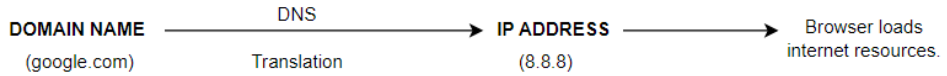
**CATEGORY THREE:** DNS is a messy, horrible solution to a problem of domain name linking to actual IP addresses. It also has some issues with IPv6, which have been partly addressed. Propose an alternate solution which does not rely on distributed name servers to match names. What will it look like? How will it function? Are there flaws? Benefits? Concerns?

**ESSAY:**

## DNS:

"Names" plays a crucial role in computer systems. It is due to "names" that we identify the unique entities, refer locations, etc. Names can be resolved to entities. Name Resolution is the one that allows process to access and retrieve the named entity. To resolve the names, naming issue, naming conflicts, a naming system should be implemented. DNS is **Domain Naming System** which was invented in **1983**. The naming systems are further implemented in **distributed and non-distributed** systems. Now the question is how they are implemented. **Efficiency and scalability** of the naming system are greatly affected.

1. The **network only understands numerical addresses**.
2. However, humans cannot memorize the numerical addresses to locate a website. Humans prefers names instead of numbers.
3. DNS is a mechanism that acts as a **translator** which converts the names to the network addresses.




for example : 127.0.0 - anything is your local computer

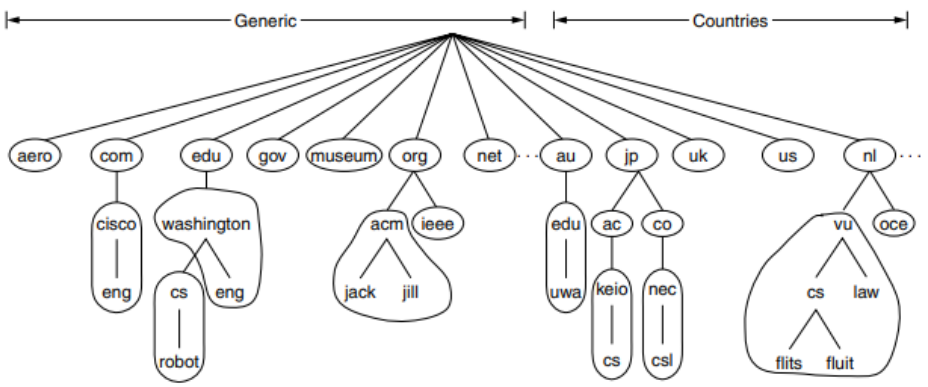



**Figure 7-5.** Part of the DNS name space divided into zones (which are circled).

4. So basically, DNS is like a **database system** or more casually a **phonebook of the internet**. This is how we refer the internet.
5. DNS is used for **mapping host names to IP addresses.**
6. It is a **hierarchical, domain-based naming scheme**. To implement this naming scheme, distributed database systems originated.
7. **Names should not contain addresses, routes, and similar information.**
8. **One DNS only handles one authorized zone. [RFC 882], [RFC 883]** gives more in depth information related to this**.**
9. **RESOLVERS:** Resolver is a **library procedure** which is called by an application program call for mapping name to an IP address. Name as a parameter is passed. Resolver queries the name to DNS server; it looks for the name and returns response in the form of IP address to the resolver and the resolver returns it to the caller. The messages are sent in the form of UDP packets. The Program establishes a TCP connection. **Resolvers extract information from the name servers.** They are **Complex regular expressions.**

Some **IRC networks have multiple DNS Names**.

Now, IRC is the **Internet Relay Chat Protocol,** or we can say a **multi-chat protocol** that needs **socket.** It is very popular among **hackers**, it's **free** and **open source**. It is a **real time asynchronous** communication. Any client has reference to the channel. The Problems that one faces using **IRC vs DISCORD** is if we connect on desktop and on mobile, we need to disconnect one of them.

**"HOSTS.TXT"** - was a single point container that stored all the hostnames and their respective numerical addresses in the 1970's which later became inefficient to handle.

**DNS Server Types:**

1. Recursive Servers,
2. Root Name Servers,
3. TLD Servers: top-level domain -- the .com, .edu or .org in the URL,
4. Authoritative name server

**Static Hosts** – These are provided by ISP, IP's

**Dynamic DNS(DDNS): Automatically** updates the records, updates the name server in the DNS **real-time.**

- **A Records –** It holds the *IP address* of the domain. Helps in ***Load balancing*** and ***prevent hacking***. **IP address of the source and the name.**
- **A Name** – Master record of its **sub-domain. Returns IP address. High TTL.**
- **Dynamic A record –** more frequently change/update, permanent.
- **SRV Record -**
- **CName Record –** C stands for ***"Canonical name records".*** These are used instead of an A record. In simple terms it is an **alias.**

## Challenges of DNS:

The shortcomings of DNS are **security and privacy concerns.** It was designed and implemented **without security**. One can easily **hijack DNS traffic**. Due to lack of security, DNS is vulnerable to active **DOS attack**s which can further be prevented by transmitting DNS messages over secure sessions and encrypting the DNS. **DNS Cache Poisoning** is one such vulnerability of DNS.

## Why DNS is messy?

Very much vulnerable to:

1. **DNS Attacks**
2. **Server Breakdowns**
3. Can be **hacked**
4. Controlled by a non-profit organisation named **ICANN75**.
5. **Fool people** – some hackers own similar domain names and creates the exact same website, if by mistake the user goes to wrong website, he gets scammed.
6. **Poor Error Handling:** Since all DNS servers are connected, troubleshooting the errors becomes difficult
7. **MASTER-SLAVE RELATIONSHIP:** One DNS Server acts as a master. If the master server fails, all other connected servers become unresponsive.

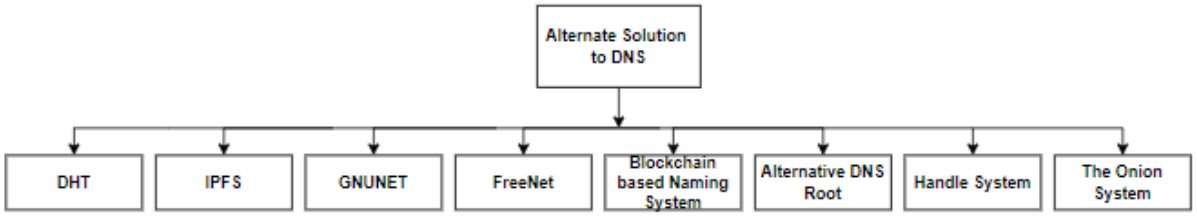
## Issues of DNS with IPv6? [DNS Server Not Responding Issue – DNS fails to map]

If we enable IPv6 host, it will start querying **'AAAA' DNS records**. Routers simply ignore the question; they don't answer the 'AAAA' DNS record. Computer waits for the **timeout** and tries to ask the question again.

## What are distributed name servers?

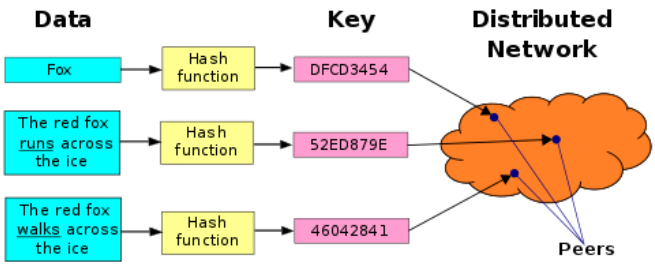
Name servers are **dedicated server on the web** designed to find websites by a domain name. It is like a contacts list. Instead of memorizing tedious, long and unnecessary phone numbers, you just assign or give name to a number. Name servers functions in the exact same way. They assign an IP address to a corresponding convenient domain name. So, it helps you in a way to only remember the domain name and not the brain-wrecking numbers!

## Alternate Solution:

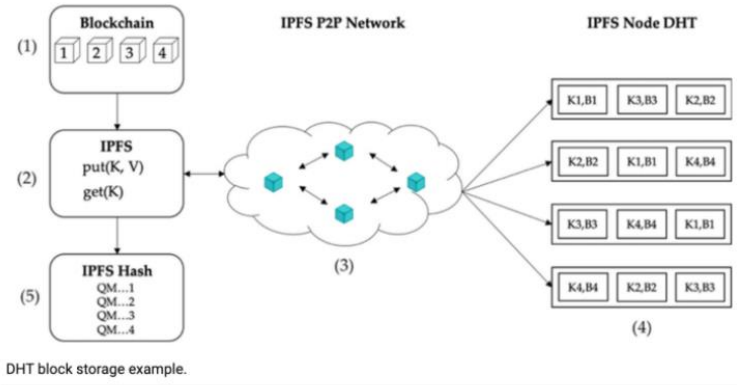



## Functioning & Working of the Alternate Solution:

**1. DHT (Distributed Hash Tables):** For domain lookups we can easily implement a **decentralized system.** DHT's are used in **cryptocurrencies** and other decentralised projects. DHT's are like a **regular hash-table**. A **key-value pair**. Key is provided and the values are returned. There are **multiple nodes,** and each node holds a key-value pairs. They are distributed across the entire network. The key-value pairs are **encoded using a hashing algorithm**.

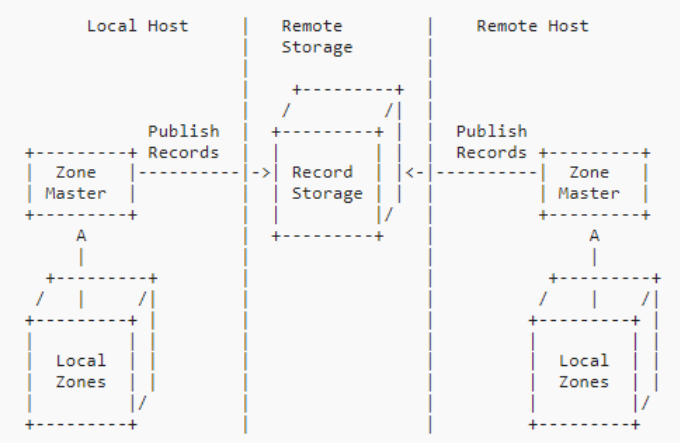



**2. IPFS (Inter-Planetary File System):** IPFS is a **Point-To-Point(P2P) file storage** system. It is like torrent that we use to download files using the BitTorrent protocol. It makes use of DHT to store data in various nodes across the network. Objects are not allowed to be altered, instead **versioning** can be used to alter .txt, .htm file for a website.

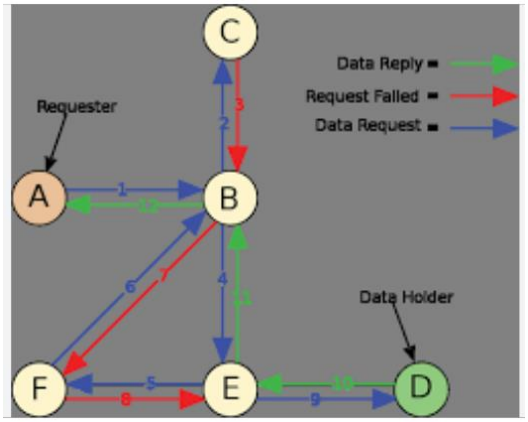



IPFS DHT Key-to-Block Storage example

**3. GnuNet:** GnuNet is created by the GNU foundation who made the Linux OS possible. It is fully **decentralised** and makes use of **directed graphs** instead of hierarchy. To achieve the integrity, the records in the GnuNet are **cryptographically secured.** Queries and replies are confidential. It means they are private to the entities. GNU Naming System bind names to the tokens that are cryptographically secured. As we saw in RFC 1035, GNS follows **local root zone deployments.** Usage of specific root zone is not at all expected. Users have the power to delegate the control of names to their zones via the local configurations. It's a kind of pet name system where you can name your zones.



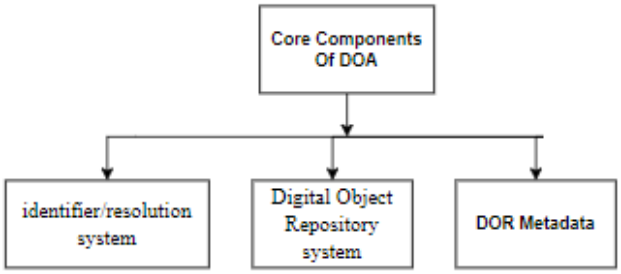

**4. FreeNet:** FreeNet is also a P2P based file sharing system. It is a **Peer-To-Peer** platform. It has a free software for publishing, uploading and **communicating** on the internet **without fear of censorship**. However, FreeNet is not suitable for applications that require high bandwidth. For eg: video streaming.



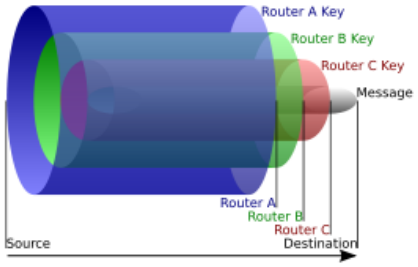

**5. Blockchain Based Naming System:** Block-chains are appended only **hash functions.** These systems are distributed between the peers of the name space, where **records are kept locally** with each peer. This is not a hierarchical naming system. Peers are free to add new names but only the owner is supposed to release it again. Famous implementations are **Namecoin, Emercoin or Blockstock** but in terms of scalability they still run behind the DNS. Clients that retrieve information from such naming system are allowed to keep a copy of the complete name space locally. This further tells us that queries are confidential. Just like we have resolvers in DNS, the clients rely on a **system that accesses information** on Blockchain. This connection between the client and the system needs a separately provided protection. In Blockchain, the Names are not confidential by design.

**6. Alternative DNS Root:** These are based on **DNS Protocol** but uses alternative root. Alternic 2 dating back to 1996 was being experimented. Its operates or works just like the regular DNS. However, the top-level domains **(TLDs)** could be same or different or slightly different. The deployment of Domain Name System Security Extensions **(DNSSEC)** using the hashing or crypto-algorithms is may or may not happen. In such case we are unsure whether the ICANN policies refrain or allow to apply to systems.

**7. The HANDLE System:** Handle System is a crucial and fundamental part of the **Digital Object Architecture.** The **identifier/resolution** system of the DOA is the Handle System. It is consistent with the DNS.




**8. The ONION System:** Also known as **"Hidden Service"** or also called as **"anonymous service"** is used by the TOR (Onion Routing Project) Project. Onion routers are network nodes, and the data or messages are encrypted layer-wise just like the peels of onion. The Onion names use the *".onion"* TLD using a process to reserve *"Special Use Domain Names"*.




**Flaws Of the Alternate Solution:** The above-mentioned alternatives are fabulous, but the main hurdle is **mass adoption.** You can't access web through them. Services like **Unstoppable Domains** directly maps the domain to hash thus allowing users to navigate to IPFS files using regular DNS. Speaking further, the incapability of name space coordination between the alternate name systems and the DNS results in unworkable and tremendous **name collisions.** We have a fear that the Internet would get fragmented as a separate ecosystem would get created altogether. And we do not wish that the internet gets fragmented as the main aim of internet was to have a one internet in the whole world.

**Benefits Of the Alternate Solution: Security, Availability, Privacy & Anonymity**

1. Censorship-Resistant.

2. Privacy- Preserving.

3. Decentralised Domain Name Resolution Protocol.

4. Helps in loading and speeding up the webpage.

5. Blockchain – integrity is guaranteed. Verification whether the information is correct or not can be done by peers.

6. DNS Response Time Speed Up.

**Concerns Of the Alternate Solution:** Any alternative protocol to the DNS can be easily designed and implemented. And it could be of much superior quality than the existing ones. But the question that lies here is: **Is our current infrastructure or the current technology capable of implementing it? Are they designed to adapt and accommodate new changes?** Especially the large regulatory bodies like

**ICANN75 (Internet Corporation for Assigned Names and Numbers)** who are responsible for standardising, defining, making a change, updating every aspect of protocols, **mass adoption is the most difficult task** that developing the infrastructure to fit in new protocols.

**Conclusion:** We have achieved a good decent kind of internet so far, but the path to achieve a better internet is still long and with full of hurdles. If we consider practically, it will take a **lot of time** and **huge deployment challenge** to come into effect. Convincing the community to embrace the alternatives is still a cumbersome task, but this is how it will accelerate and popularise. **Transition mechanisms** are not viable for long-term. Alternate solutions to DNS cannot be assumed or expected to work flawlessly. To bridge DNS can lead to serious annoying results like increase in the **support costs, user dismay and frustration, less security, less privacy, instability of the internet.** The alternative naming system though appear to be useful, but they possess **serious risks. Unexpected clashing, crashing of the internet, user frustration, numerous name collisions** and the list goes on.