Application Layer: DNS and Network Applications

Study-Ready Notes

Compiled by Andrew Photinakis

October 17, 2025

Contents

1 DNS: Domain Name System

1.1 Fundamental Concepts

- **People identifiers**: SSN, name, passport number
- **Internet host identifiers**:
  - IP address (32-bit) - used for addressing datagrams
  - Domain name (e.g., cs.rit.edu) - used by humans

Key Question: How to map between IP addresses and domain names, and vice versa?

1.2 DNS Definition and Characteristics

- **Distributed database** implemented in hierarchy of many name servers
- **Application-layer protocol**: hosts and DNS servers communicate to resolve names
- Core Internet function implemented as application-layer protocol
- Complexity resides at network's "edge"

[Summary: DNS is a distributed hierarchical system that translates human-readable domain names into machine-readable IP addresses, functioning as a critical Internet infrastructure component.]

2 DNS Services and Structure

2.1 DNS Services

- **Hostname-to-IP-address translation**: Primary function
- **Host aliasing**: Canonical vs. alias names
- **Mail server aliasing**: Email domain resolution
- **Load distribution**: Multiple IP addresses for one name (server replication)

2.2 Why Not Centralize DNS?

- **Single point of failure**: Central server outage would break entire system
- **Traffic volume**: Immense query load would overwhelm single server
- **Distant centralized database**: High latency for remote users
- **Maintenance challenges**: Impossible to manage centrally at Internet scale

Conclusion: Centralization doesn't scale!
Real-world scale evidence:

- Comcast DNS servers: 600 billion queries/day
- Akamai DNS servers: 2.2 trillion queries/day

[Mnemonic: DNS Problems - Single Traffic Distance Maintenance (STDM)] [Summary: DNS provides multiple services beyond basic name resolution, and its distributed nature solves scalability and reliability issues that would plague a centralized system.]

3 DNS Architecture and Characteristics

3.1 DNS System Properties

- **Humongous distributed database**: Approximately 1 billion records, each relatively simple
- **Massive query volume**: Handles many trillions of queries daily
  - Many more reads than writes
  - Performance critical: Milliseconds matter for user experience
  - Almost every Internet transaction interacts with DNS
- **Decentralized organization**: Millions of organizations responsible for their own records
- **"Bulletproof" requirements**: High reliability and security essential

4 DNS Hierarchical Structure

4.1 Distributed Hierarchical Database

Resolution process example: Client wants IP address for www.amazon.com

1. Client queries root server to find .com DNS server
2. Client queries .com DNS server to get amazon.com DNS server
3. Client queries amazon.com DNS server to get IP address for www.amazon.com

[Summary: DNS uses a hierarchical distributed database with root servers at the top, TLD servers in the middle, and authoritative servers at the bottom for specific domains.]

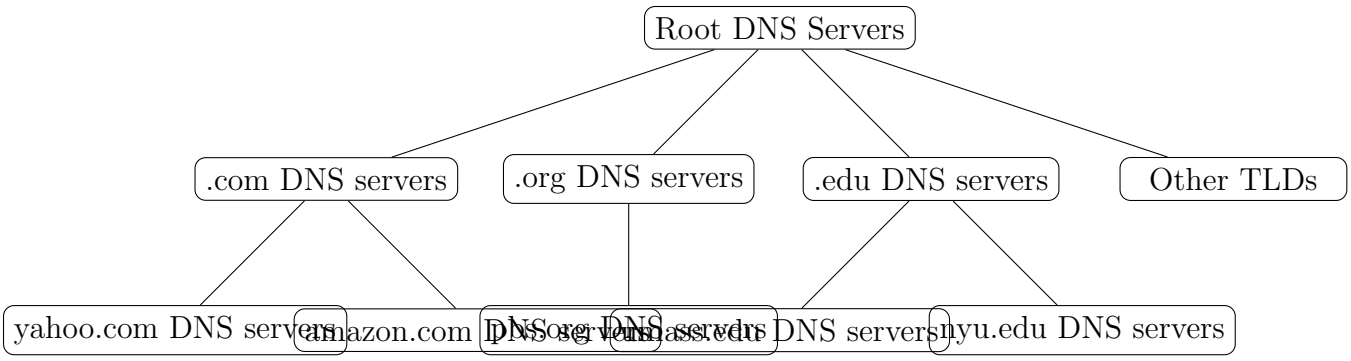

Figure 1: DNS Hierarchical Database Structure

5 DNS Server Types

5.1 Root Name Servers

- Official contact-of-last-resort for unresolved names
- Incredibly important Internet function - Internet couldn't function without it!
- DNSSEC provides security (authentication, message integrity)
- Managed by ICANN (Internet Corporation for Assigned Names and Numbers)

Deployment statistics:

- 13 logical root name "servers" worldwide
- Each "server" replicated many times (200 servers in US)
- Geographic distribution for reliability and performance

5.2 Top-Level Domain (TLD) Servers

- Responsible for top-level domains:
  - Generic TLDs: .com, .org, .net, .edu, .aero, .jobs, .museums
  - Country-code TLDs: .cn, .uk, .fr, .ca, .jp
- Managed by specific organizations:
  - Network Solutions: authoritative registry for .com, .net
  - Educause: .edu TLD

5.3 Authoritative DNS Servers

- Organization's own DNS server(s)
- Provide authoritative hostname-to-IP mappings for organization's named hosts
- Can be maintained by organization or service provider

5.4 Local DNS Name Servers

- First point of contact for host DNS queries
- Returns replies from:
  - Local cache of recent name-to-address translations
  - Forwarding requests into DNS hierarchy for resolution
- Each ISP has local DNS name server
- Doesn't strictly belong to hierarchy

Finding your local DNS server:

- MacOS: `% scutil --dns`
- Windows: `> ipconfig /all`

[Concept Map: DNS Hierarchy → Root → TLD → Authoritative → Local (caching)] [Summary: DNS uses four main server types: root servers for global coordination, TLD servers for domain categories, authoritative servers for specific domains, and local servers for client-side caching and resolution.]

6 DNS Name Resolution Methods

6.1 Iterated Query

Characteristics:

- Contacted server replies with name of server to contact
- "I don't know this name, but ask this server"
- Local server does most of the work

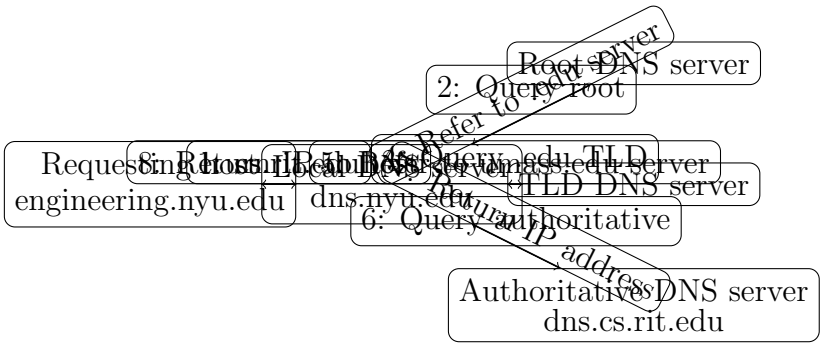

Figure 2: DNS Iterated Query Resolution

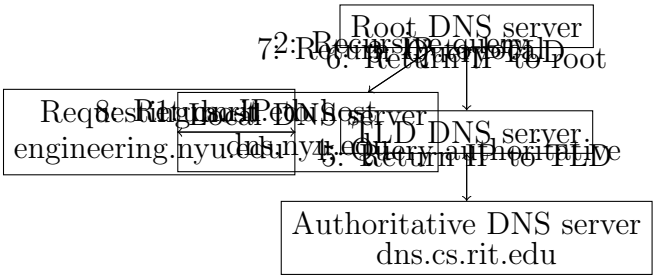

Figure 3: DNS Recursive Query Resolution

6.2 Recursive Query

Characteristics:

- Puts burden of name resolution on contacted name server
- Potential heavy load at upper levels of hierarchy
- Less common in practice due to load concerns

[Summary: DNS supports two resolution methods - iterated queries where servers refer clients to other servers, and recursive queries where servers do the complete resolution on behalf of clients.]

7 DNS Caching and Performance

7.1 Caching DNS Information

- Once any name server learns mapping, it **caches** the mapping
- Immediately returns cached mapping for subsequent queries
- **Benefits**:
  - Improves response time significantly

  - Reduces load on DNS infrastructure

- **Time-to-Live (TTL)**:
  - Cache entries timeout after TTL expires
  - TLD servers typically cached in local name servers

7.2 Cache Limitations

- Cached entries may be **out-of-date**
- If host changes IP address, may not be known Internet-wide until all TTLs expire
- DNS provides **best-effort name-to-address translation**

[Mnemonic: DNS Cache - Time To Live, Temporarily Local, Limited freshness] [Summary: DNS caching dramatically improves performance but introduces potential staleness issues, with TTL values controlling how long cached entries remain valid.]

8 DNS Records and Protocol

8.1 DNS Resource Records (RR)

DNS: distributed database storing resource records (RR)
RR format: {name, value, type, ttl}

- **type=A**
  - Name: hostname
  - Value: IP address
- **type=CNAME**
  - Name: alias name for canonical name
  - Value: canonical name
  - Example: www.ibm.com is really servereast.backup2.ibm.com
- **type=NS**
  - Name: domain (e.g., foo.com)
  - Value: hostname of authoritative name server for this domain
- **type=MX**
  - Value: name of SMTP mail server associated with name

Field	Description
Identification	16-bit number for query/reply matching
Flags	Query/reply, recursion desired/available, authoritative
# Questions	Number of questions in question section
# Answer RRs	Number of answer resource records
# Authority RRs	Number of authority resource records
# Additional RRs	Number of additional resource records
Questions	Variable number of questions (name, type fields)
Answers	Variable number of RRs in response to query
Authority	Records for authoritative servers
Additional Info	Additional "helpful" information

Table 1: DNS Message Format

8.2 DNS Protocol Messages

DNS query and reply messages share the same format:

[Summary: DNS uses standardized resource records to store different types of information and employs a consistent message format for both queries and replies with multiple sections for different purposes.]

9 DNS Administration and Security

9.1 Registering Domain Information

Example: New startup "Network Utopia"

1. Register name networkingtopia.com at DNS registrar (e.g., Network Solutions)
2. Provide names and IP addresses of authoritative name servers (primary and secondary)
3. Registrar inserts NS and A records into .com TLD server:
   - (networkingpia.com, dns1.networkingpia.com, NS)
   - (dns1.networkingpia.com, 212.212.212.1, A)
4. Create authoritative server locally with IP address 212.212.212.1
   - Type A record for www.networkingtopia.com
   - Type MX record for networkingpia.com

9.2 DNS Security Threats

9.2.1 DDoS Attacks

- Bombard root servers with traffic

- Not successful to date due to:
  - Traffic filtering
  - Local DNS servers cache TLD server IPs, allowing root server bypass
- Bombarding TLD servers potentially more dangerous

9.2.2 Spoofing Attacks

- Intercept DNS queries, returning bogus replies
- DNS cache poisoning
- Countermeasure: DNSSEC (RFC 4033) provides authentication services

[Summary: DNS administration involves registering domains with registrars who update TLD servers, while DNS security addresses DDoS and spoofing attacks through filtering, caching, and DNSSEC authentication.]