Лабораторна робота № 1

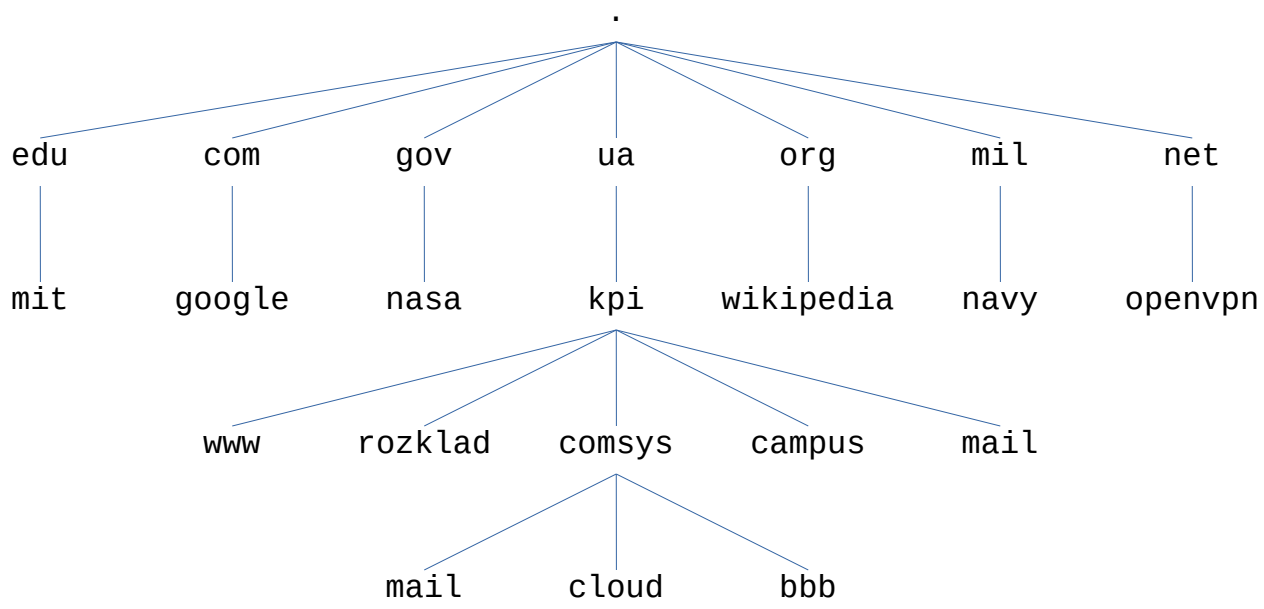
Служба доменних імен (DNS)

1. Короткі теоретичні відомості.

1.1. Ієрархія доменів.

DNS реалізує ієрархічний простір імен для Інтернет-об’єктів. На відміну від імен файлів Unix, які обробляються зліва направо (компоненти імен відокремлюються символом “/”), імена DNS обробляються зправо наліво і використовують символ “.” в якості роздільника. Хоча вони обробляються зправо наліво, користувачі все ще читають доменні імена зліва направо. Прикладом доменного імені для вузла є comsys.kpi.ua.

Доменні імена використовуються для іменування Інтернет-об’єктів. Під цим мається на увазі те, що DNS не використовується строго для перетворення імен вузлів у адреси вузлів. Точніше сказати, що DNS відображає доменні імена у значення. На даний момент ці значення є IP-адресами.



Малюнок 1. Приклад ієрархії доменів.

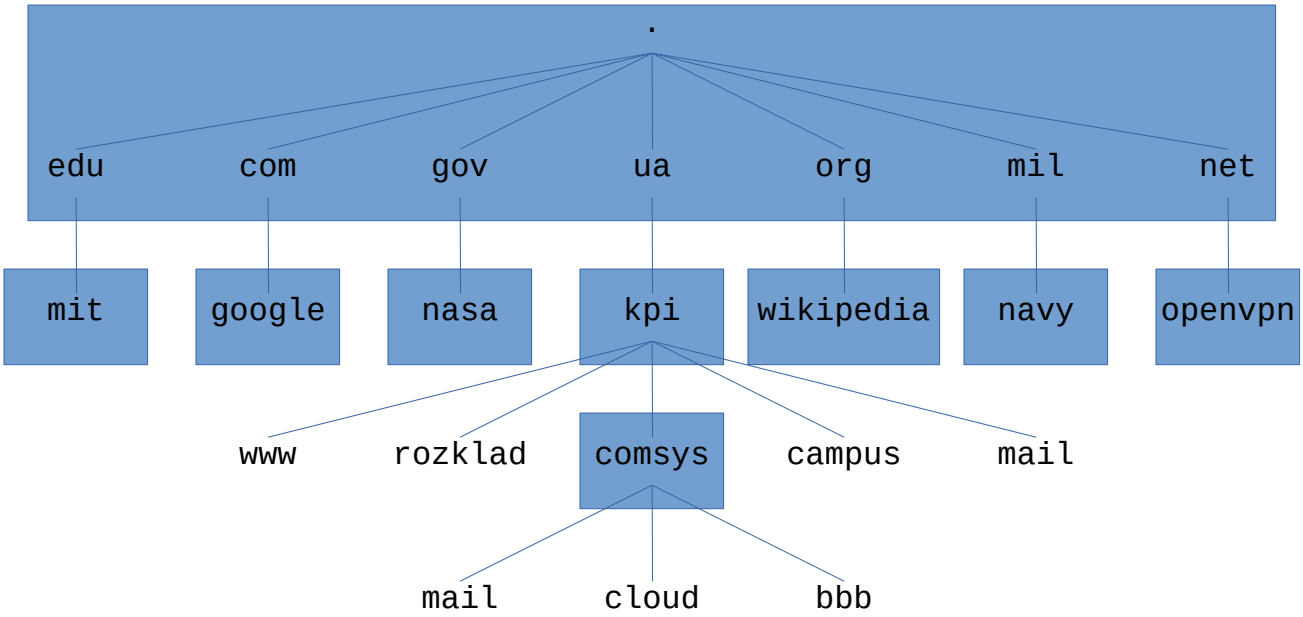
Ієрархія DNS, так само як ієрархія файлів Unix, може бути візуалізована у вигляді дерева, де кожен вузол у дереві відповідає домену, а листя у дереві відповідають іменованим вузлам. На малюнку 1 наведено приклад ієрархії доменів.

Під час розробки ієрархії доменних імен відбулася значна кількість дискусій щодо того, які конвенції регулюватимуть імена, які повинні видаватися вгорі ієрархії. На першому рівні ієрархія не дуже широка. Для кожної країни світу виділене одне доменне ім'я, а також “велика

шістка” доменів: .edu, .com, .gov, .mil, .org та .net. Усі ці шість доменів спочатку базувались у США, наприклад: лише акредитовані в США навчальні заклади можуть зареєструвати доменне ім’я .edu. За останні роки кількість доменів верхнього рівня була розширена для вирішення великого попиту на імена доменів .com. До нових доменів верхнього рівня належать .biz, .coop, .info... Зараз існує понад 1200 доменів верхнього рівня.

1.2. Сервери імен.

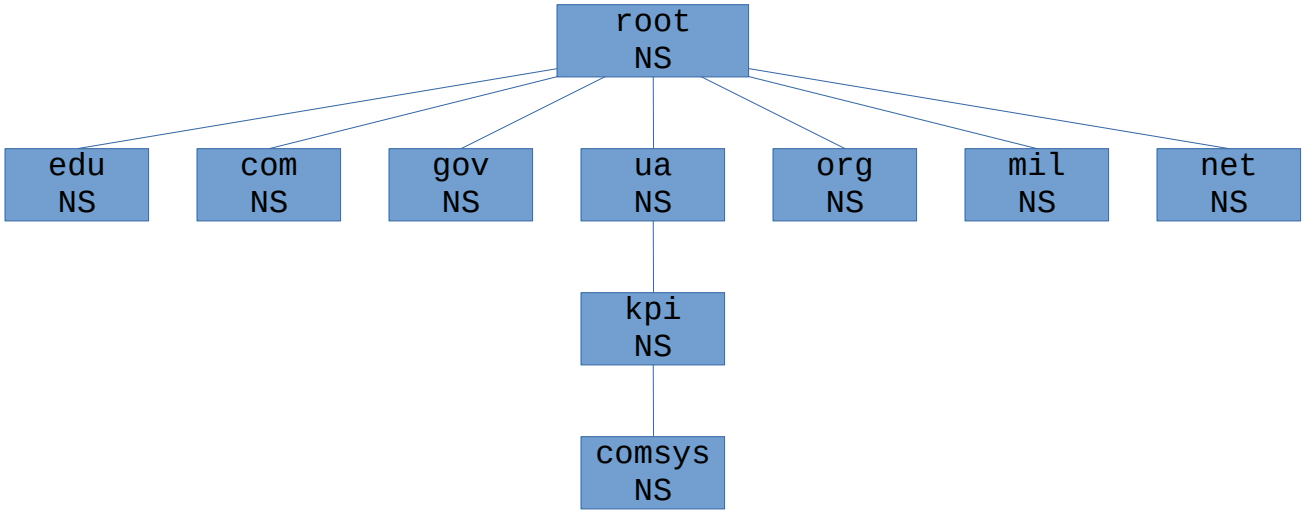
Повна ієрархія доменних імен існує лише абстрактно. Першим кроком є розділення ієрархії на піддерева, які називаються зонами. На малюнку 2 показано, як ієрархію, подану на малюнку 1, можна розділити на зони. Кожна зона може розглядатися як відповідна якомусь адміністративному органу, який відповідає за цю частину ієрархії. Наприклад, верхній рівень ієрархії утворює зону, якою керує Internet Corporation for Assigned Names and Numbers (ICANN). У центрі на малюнку знаходиться зона, яка відповідає Національному технічному університету України “Київський політехнічний інститут імені Ігоря Сікорського”. У межах цієї зони деякі кафедри не хочуть керувати власною частиною ієрархії (і тому вони залишаються в зоні університетського рівня), тоді як інші, як кафедра обчислювальної техніки, керує власною зоною.



Малюнок 2. Ієрархія доменів, розділена на зони.

Важливість зони полягає в тому, що вона відповідає основній одиниці реалізації в DNS - серверу імен. Зокрема, інформація, що міститься в кожній зоні, реалізована на двох або більше серверах імен. Кожен сервер імен, у свою чергу, є програмою, до якої можна отримати доступ через мережу Інтернет. Клієнти надсилають запити на сервери імен, а сервери імен їм відповідають. Іноді відповідь містить повну інформацію, яку бажає клієнт, а іноді відповідь містить вказівник на інший сервер, якому клієнт повинен відправити запит. Таким чином, з точки зору реалізації, точніше вважати, що DNS представлений ієрархією серверів імен, а не ієрархією доменів, як показано на малюнку 3.

Кожна зона реалізована на двох або більше серверах імен для забезпечення відмовостійкості. Тобто інформація з домену буде доступна, навіть якщо один сервер імен виходить з ладу. З іншого боку, сервер імен може обслуговувати більше однієї зони.



Малюнок 3. Ієрархія серверів імен.

Кожен сервер імен містить інформацію про зону як сукупність записів ресурсів. По суті, запис ресурсу - це прив’язка імені до значення, яке складається з п'яти полів:

```
(Name, Value, Type, Class, TTL)
```

Поля Name та Value – містять інформацію, яка цікавить кінцевих користувачів, тоді як поле Type вказує, як значення слід інтерпретувати. Наприклад, Type = A вказує, що Value є IP-адресою. Таким чином, записи A реалізують відображення імені та адреси. Інші типи записів включають:

- `NS` – поле Value містить доменне ім'я для вузла, на якому запущений сервер імен, який знає, як перетворювати імена в межах зазначеного домену;
- `CNAME` – поле Value містить канонічну назву для конкретного вузла; воно використовується для визначення псевдонімів;
- `MX` – поле Value містить доменне ім'я вузла, на якому знаходиться поштовий сервер, який оброблює повідомлення для вказаного домену.

Поле Class було включено, щоб дозволити сутностям, крім NIC, визначати корисні типи записів. На сьогоднішній день єдиним широко використовуваним Class є Internet, його позначають IN. Поле “Час життя” (TTL) показує, як довго цей ресурсний запис є дійсним. Він використовується серверами, які кешують записи ресурсів з інших серверів. Коли час життя TTL закінчується, сервер повинен вилучити запис із кешу.

Приклад.

Кореневий сервер імен містить запис NS для кожного сервера імен домену верхнього рівня (TLD). Це визначає сервер, який може відповідати на запити для цієї частини ієрархії DNS (.ua

та .com у прикладі). Він також має записи A, які перетворюють ці імена у відповідні IP-адреси. У сукупності ці два записи ефективно реалізують покажчик з кореневого сервера імен на один із серверів TLD.

```
(ua, in1.ns.ua, NS, IN)
(in1.ns.ua, 74.123.224.40, A, IN)
(com, a.gtld-servers.net, NS, IN)
(a.gtld-servers.net, 192.5.6.30, A, IN)
...
```

Просуваючись вниз по ієрархії на один рівень, сервер має записи для таких доменів, як цей:

```
(kpi.ua, ns.kpi.ua, NS, IN)
(ns.kpi.ua, 77.47.128.130, A, IN)
...
```

Наведені запис NS та запис A для сервера імен, який відповідає за частину ієрархії kpi.ua. Цей сервер може мати можливість безпосередньо відповідати на деякі запити (наприклад, comsys.kpi.ua), тоді як він перенаправлятиме інші запити на сервер ще на один рівень в ієрархії нижче (наприклад, для запиту про cloud.comsys.kpi.ua).

```
(cloud.comsys.kpi.ua, 10.18.49.190, A, IN)
(comsys.kpi.ua, ns.comsys.kpi.ua, NS, IN)
(ns.comsys.kpi.ua, 10.18.49.2, A, IN)
...
```

Нарешті, сервер імен третього рівня, який керує доменом comsys.kpi.ua, містить записи A для всіх своїх вузлів. Він також може визначити набір псевдонімів (записів CNAME) для кожного з цих вузлів. Псевдоніми іноді є просто зручними (наприклад, коротшими) назвами комп'ютерів. Наприклад, www.comsys.kpi.ua є псевдонімом для вузла з іменем comsys.kpi.ua. Це дозволяє перенести веб-сайту на інший сервер, не змінюючи налаштування у користувачів; вони просто продовжують використовувати псевдонім, не зважаючи на те, на якому сервері зараз знаходиться веб-сайт домену. Записи обміну поштою (MX) служать тій самій меті для програм електронної пошти - вони дозволяють адміністратору змінювати вузол, який отримує пошту домену, не змінюючи електронної адреси.

```
(www.comsys.kpi.ua, comsys.kpi.ua, CNAME, IN)
(comsys.kpi.ua, 77.47.192.42, A, IN)
(comsys.kpi.ua, mail.comsys.kpi.ua, MX, IN)
(mail.comsys.kpi.ua, 77.47.193.180, A, IN)
...
```

Система імен DNS зазвичай використовується для іменування вузлів (включаючи сервери) та сайтів, хоча записи ресурсів можна визначити практично для будь-якого типу об’єкта. DNS не використовується для іменування окремих людей або інших об’єктів, таких як файли або каталоги; інші системи імен зазвичай використовуються для ідентифікації таких об'єктів. Наприклад, X.500 - це система імен ISO, призначена для спрощення ідентифікації людей. Це дозволяє назвати особу, вказавши набір атрибутів: ім’я, прізвище, номер телефону, поштову адресу тощо. X.500 виявився занадто громіздким - і, в якомусь сенсі, був узурпований потужними пошуковими системами, які тепер доступні в Інтернеті, - але з часом він перетворився на Легкий протокол доступу до каталогів (LDAP). LDAP - це підмножина X.500, спочатку розроблена як інтерфейс для персональних комп'ютерів до X.500. Сьогодні широко використовується, переважно на рівні підприємств, як система збереження інформації про користувачів.

1.3. Перетворення імені.

Враховуючи ієрархію серверів імен, необхідно розглянути питання про те, як клієнт використовує ці сервери для перетворення доменного імені в IP-адресу. Для ілюстрації основної ідеї, припустимо, клієнт хоче перетворити ім'я comsys.kpi.ua. Клієнт міг спочатку надіслати запит, що містить це ім’я, на один із кореневих серверів (як ми побачимо нижче, це рідко трапляється на практиці, але на сьогодні достатньо для ілюстрації базової операції). Кореневий сервер, який не може збігатися з цілим іменем, повертає найкращий збіг, який у нього є - запис NS для ua, який вказує на сервер TLD in1.ns.ua. Сервер також повертає всі записи, пов’язані з цим записом, у цьому випадку запис A для in1.ns.ua. Клієнт, не отримавши відповіді, далі відправляє той самий запит на сервер імен на IP-адресу 74.123.224.40. Цей сервер також не може збігатися з цілим ім'ям, тому повертає NS та відповідні записи A для домену kpi.ua. Ще раз клієнт надсилає той самий запит, що й раніше, на сервер на вузлі з IP 77.47.128.130. Цього разу досягнуто сервера, який може повністю обробити запит. Остаточний запит до сервера з адресою 77.47.128.130 видає запис A для comsys.kpi.ua, і клієнт дізнається, що відповідною IP-адресою є 77.47.192.42.

Цей приклад все ще залишає без відповіді кілька питань щодо процесу перетворення. Перше питання полягає в тому, як спочатку клієнт знайшов кореневий сервер, або, по-іншому, як він шукає ім’я сервера, який знає, як перетворити потрібне ім'я? Це фундаментальна проблема будь-якої системи іменування, і відповідь полягає в тому, що систему потрібно якось завантажити. У цьому випадку відображення імені та адреси для одного або декількох кореневих серверів добре відомо; тобто він публікується якимось чином поза самою системою імен.

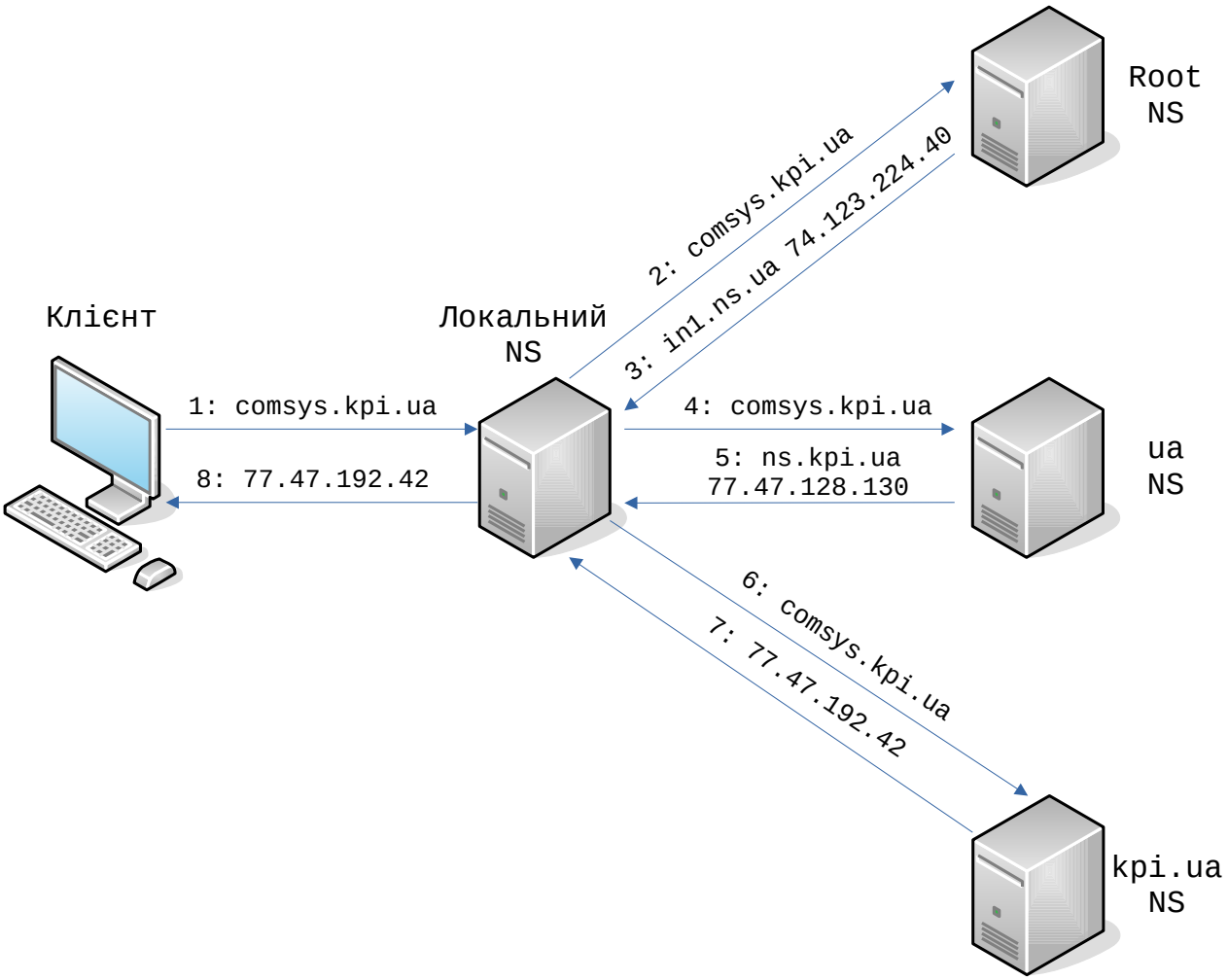
Однак на практиці не всі клієнти знають про кореневі сервери. Натомість клієнтська програма, що працює на кожному вузлі Інтернету, ініціалізується адресою локального сервера імен. Наприклад, усі вузли кафедри обчислювальної техніки знають про сервер ns.comsys.kpi.ua. Цей локальний сервер імен, у свою чергу, має записи ресурсів для одного або декількох кореневих серверів, наприклад:

```
('root', a.root-servers.net, NS, IN)
(a.root-servers.net, 198.41.0.4, A, IN)
```

Таким чином, перетворення імені насправді передбачає запит клієнта на локальний сервер, який, у свою чергу, діє як клієнт, який запитує віддалені сервери від імені клієнта. Це призводить до взаємодії клієнт/сервер, проілюстровану на малюнку 4. Однією з переваг цієї

моделі є те, що всі вузли в Інтернеті не повинні постійно оновлюватися. Де розташовані поточні кореневі сервери повинні знати лише DNS-сервери. Друга перевага полягає в тому, що локальний сервер бачить відповіді, які повертаються із запитів, відправлених усіма локальними клієнтами. Локальний сервер кешує ці відповіді і іноді може перетворювати майбутні запити, не виходячи з мережі. Поле TTL у записах ресурсів, що повертаються віддаленими серверами, вказує, як довго кожен запис можна безпечно кешувати. Цей механізм кешування може бути використаний і далі вгору по ієрархії, зменшуючи навантаження на кореневий і TLD-сервери.

Друге питання полягає в тому, як працює система, коли користувач подає часткове ім’я (наприклад, comsys), а не повне доменне ім’я (наприклад, comsys.kpi.ua). Відповідь полягає в тому, що клієнтська програма налаштована на локальний домен, в якому знаходиться вузол (наприклад, kpi.ua), і вона додає цей рядок до будь-яких простих імен перед надсиланням запиту.



Малюнок 4. Перетворення імен на практиці.

1.4. Доменна термінологія.

1.4.1. Система доменних імен.

Система доменних імен, більш відома як "DNS" - це мережна система, яка дозволяє перетворити зручні для людини (символьні) імена в унікальні IP-адреси.

1.4.2. Доменне ім'я.

Доменне ім’я - це дружнє для людини ім’я, яке ми звикли пов’язувати з Інтернет-ресурсом. Наприклад, "google.com" - це доменне ім'я. Деякі люди скажуть, що частина "google" - це домен, але загалом ми можемо називати комбіновану форму іменем домену.

URL-адреса “google.com” пов’язана із серверами, що належать Google Inc. Система доменних імен дозволяє нам отримувати доступ до серверів Google, коли ми вводимо “google.com” у веббраузері.

1.4.3. IP-адреса.

IP-адреса - це те, що ми називаємо адресою мережі. Кожна IP-адреса повинна бути унікальною в своїй мережі. Коли ми говоримо про вебсайти, ця мережа - це весь Інтернет.

IPv4, найпоширеніша форма адрес, записується як чотири набори чисел, кожен набір має до трьох цифр, причому кожен набір відокремлений крапкою. Наприклад, “111.222.111.222” може бути дійсною IP-адресою IPv4. За допомогою DNS ми прив'язуємо ім'я до цієї адреси, щоб вам не потрібно було запам'ятовувати складний набір чисел для кожного місця, яке ви хочете відвідати в мережі.

1.4.4. Домен верхнього рівня.

Домен верхнього рівня (TLD) є найбільш загальною частиною домену. Домен верхнього рівня - це найвіддаленіша частина праворуч (відокремлена крапкою). Загальними доменами верхнього рівня є "com", "net", "org", "gov", "edu" та "io".

Домени верхнього рівня знаходяться на вершині ієрархії з точки зору доменних імен. ICANN (Інтернет-корпорація з присвоєними іменами та номерами) певним сторонам надає управлінський контроль над доменами верхнього рівня. Потім ці сторони можуть розповсюджувати доменні імена згідно TLD, як правило, через реєстратора доменів.

1.4.5. Вузли.

В межах домену власник домену може визначити окремі вузли, які посилаються на окремі комп’ютери або служби, доступні через домен. Наприклад, більшість власників доменів роблять свої веб-сервери доступними через відкритий домен (example.com), а також через визначення “вузла” “www” (www.example.com).

Ви можете мати інші визначення вузлів під загальним доменом. Ви можете отримати доступ до API через вузол “api” (api.example.com), або можете отримати доступ до ftp, визначивши вузол, який називається “ftp” або “files” (ftp.example.com або files.example.com). Імена вузлів можуть бути довільними, якщо вони унікальні для домену.

1.4.6. Піддомен.

Предметом, що стосується вузлів, є піддомени.

DNS працює в ієрархії. Домени верхнього рівня можуть мати багато доменів. Наприклад, "com" TLD має як "google.com", так і "ubuntu.com" під ним. "піддомен" означає будь-який домен, який є частиною більшого домену. У цьому випадку ”ubuntu.com” можна сказати як піддомен “com”. Зазвичай це просто називається доменом, або частина “ubuntu” називається SLD, що означає домен другого рівня.

Так само кожен домен може контролювати "піддомени", які знаходяться під ним. Зазвичай це те, що ми маємо на увазі під піддоменами. Наприклад, може бути піддомен для історичного відділу школи за адресою “www.history.school.edu”. Частина "history" є піддоменом.

Різниця між іменем вузла та піддоменом полягає в тому, що вузол визначає комп’ютер або ресурс, тоді як піддомен розширює батьківський домен. Це метод поділу самого домену.

Незалежно від того, чи йдеться про піддомени чи вузли, можна побачити, що самі ліві частини домену є найбільш конкретними. Ось як працює DNS: від найбільш до найменш конкретного, коли читаєте зліва направо.

1.4.7. Повне доменне ім'я.

Повне доменне ім’я, яке часто називають FQDN, - це те, що ми називаємо абсолютним доменним ім’ям. Домени в системі DNS можуть бути надані відносно один одного, і як такі, можуть бути дещо неоднозначними. Ідентифікаційне доменне ім’я (FQDN) - це абсолютне ім’я, яке вказує його розташування щодо абсолютного кореня системи доменних імен.

Це означає, що він визначає кожен батьківський домен, включаючи TLD. Правильне повне доменне ім’я закінчується крапкою, що вказує на корінь ієрархії DNS. Прикладом повного доменного імені є "mail.comsys.kpi.ua.". Іноді програмне забезпечення, яке вимагає повного доменного імені, не вимагає кінцевої крапки, але кінцева крапка потрібна, щоб відповідати стандартам ICANN.

1.4.8. Сервер імен.

Сервер імен - це комп'ютер, призначений для перетворення доменних імен в IP-адреси. Ці сервери виконують більшу частину роботи в системі DNS. Оскільки загальна кількість перетворень доменів є занадто великою для будь-якого одного сервера, кожен сервер може перенаправити запит на інші сервери імен або делегувати відповідальність за підмножину піддоменів, за які вони відповідають.

Сервери імен можуть бути "авторитетними", тобто вони дають відповіді на запити про домени, що знаходяться під їх контролем. В іншому випадку вони можуть вказувати на інші сервери або подавати кешовані копії даних інших серверів імен.

1.4.9. Файл зони.

Файл зони - це простий текстовий файл, який містить зіставлення між іменами доменів та IP-адресами. Ось як система DNS нарешті з’ясовує, з якою IP-адресою слід зв’язатися, коли користувач запитує певне доменне ім’я.

Файли зон знаходяться на серверах імен і, як правило, визначають ресурси, доступні під певним доменом, або місце, куди можна звернутись, щоб отримати цю інформацію.

1.4.10. Записи.

У файлі зони зберігаються записи. У своїй найпростішій формі запис в основному являє собою єдине відображення між ресурсом та іменем. Вони можуть зіставити доменне ім’я з IP-адресою, визначити сервери імен для домену, визначити поштові сервери для домену тощо.

1.5. Як працює DNS?

1.5.1. Кореневі сервери.

Як ми вже говорили вище, DNS за своєю суттю є ієрархічною системою. На вершині цієї системи знаходиться так звані «кореневі сервери». Ці сервери контролюються різними організаціями та делегуються повноваженнями ICANN (Інтернет-корпорація з присвоєння імен та номерів).

В даний час працює 13 кореневих серверів. Однак, оскільки існує неймовірна кількість імен, які потрібно перетворювати щохвилини, кожен із цих серверів насправді відображається дзеркально. Цікавим у цій установці є те, що кожне дзеркало для одного кореневого сервера має

однакову IP-адресу. Коли надсилаються запити на певний кореневий сервер, запит буде перенаправлено до найближчого дзеркала цього кореневого сервера.

Що роблять ці кореневі сервери? Кореневі сервери обробляють запити на інформацію про домени верхнього рівня. Отже, якщо надходить запит на щось, що сервер імен нижчого рівня не може вирішити, запит надходить до кореневого сервера домену.

Кореневі сервери насправді не знатимуть, де розміщений домен. Однак вони зможуть направити запитувача на сервери імен, які обробляють спеціально запитаний домен верхнього рівня.

Отже, якщо на кореневий сервер подається запит на “www.wikipedia.org”, кореневий сервер не знайде результат у своїх записах. Він перевірить свої файли зон на наявність списку, який відповідає “www.wikipedia.org”. Не знайде.

Натомість він знайде запис для домену “org” верхнього рівня і дасть клієнту, який запитував, адресу сервера імен, відповідального за “org” адреси.

1.5.2. Сервери TLD.

Потім запитувач надсилає новий запит на IP-адресу (надану йому кореневим сервером), яка відповідає за домен верхнього рівня.

Отже, щоб продовжити наш приклад, він надішле запит на сервер імен, відповідальний за знання про “org” домени, щоб перевірити, чи знає він, де знаходиться “www.wikipedia.org”.

Ще раз запитувач шукатиме “www.wikipdia.org” у своїх файлах зони. Він не знайде цей запис у своїх файлах.

Однак він знайде запис із переліком IP-адреси сервера імен, відповідального за “wikipedia.org”. Це наближає клієнта до потрібної відповіді.

1.5.3. Сервери імен на рівні домену.

На даний момент запитувач має IP-адресу сервера імен, який відповідає за знання фактичної IP-адреси ресурсу. Він надсилає новий запит на сервер імен, ще раз запитуючи, чи може він перетворити “www.wikipedia.org”.

Сервер імен перевіряє свої файли зон і виявляє, що у нього є файл зони, пов'язаний з “wikipedia.org”. Усередині цього файлу є запис для вузла “www”. Цей запис повідомляє IP-адресу, де знаходиться цей вузол. Сервер імен повертає остаточну відповідь запитувачу.

1.5.4. Що таке сервер перетворення імен?

У наведеному вище сценарії ми називали "запитувача".

Майже у всіх випадках запитувачем буде вузол, який ми називаємо "сервером перетворення імен". Сервер перетворення імен налаштований на запитання інших серверів. В основному це посередник для користувача, який кешує попередні результати запитів для збільшення швидкості та знає адреси кореневих серверів, щоб мати змогу перетворити запити, зроблені для вузлів, про які він ще не знає.

В основному, користувач, як правило, має кілька серверів перетворення імен, налаштованих у своїй комп'ютерній системі. Сервери перетворення імен зазвичай надаються провайдером або іншими організаціями. Наприклад, Google пропонує свої сервери перетворення імен, які ви можете запитувати. Вони можуть бути налаштовані на вашому комп'ютері автоматично або вручну.

Коли ви вводите URL-адресу в адресний рядок веббраузера, комп’ютер спочатку перевіряє, чи зможе він локально з’ясувати, де знаходиться ресурс. Він перевіряє файл “hosts” на комп’ютері. Потім він надсилає запит на сервер перетворення імен і чекає відповідь, щоб з'ясувати IP-адресу ресурсу.

Потім сервер перетворення імен перевіряє свій кеш на відповідь. Якщо він не знаходить його, він виконує описані вище дії.

1.6. Файли зони.

Файли зон - це спосіб, яким сервери імен зберігають інформацію про домени, про які вони знають. Кожен домен, про який знає сервер імен, зберігається у файлі зони. Більшість запитів, що надходять на середній сервер імен, не є тим, для чого сервер буде мати файли зон.

Якщо він налаштований на обробку рекурсивних запитів, таких як сервер перетворення імен, він знайде відповідь і поверне її. В іншому випадку він повідомить запитуючу сторону, де шукати далі.

Чим більше файлів зон має сервер імен, тим на більше запитів він зможе авторитетно відповісти.

Файл зони описує "зону" DNS, яка в основному є підмножиною всієї системи імен DNS. Зазвичай він використовується для налаштування лише одного домену. Він може містити кілька записів, які визначають, де знаходяться ресурси для відповідного домену.

$ORIGIN зони - параметр, який за замовчуванням дорівнює найвищому рівню повноважень зони.

Отже, якщо для налаштування «example.com» використовується файл зони домену, для $ORIGIN буде встановлено example.com .

Це або налаштовано у верхній частині файлу зони, або його можна визначити у файлі конфігурації сервера DNS, який посилається на файл зони. У будь-якому випадку, цей параметр описує, для чого зона буде повноважною.

Аналогічно, $TTL налаштовує "час життя" інформації, яку він надає. В основному це таймер. Кешуючий сервер імен може використовувати попередньо запрошені результати для відповіді на запитання, поки значення TTL не закінчиться.

1.7. Типи записів.

1.7.1. SOA запис.

Запис авторизації, або SOA, є обов’язковим записом у всіх файлах зони. Це має бути перший явний запис у файлі (хоча вище можуть бути вказані специфікації $ORIGIN або $TTL). Початок авторитетного запису виглядає приблизно так:

```
domain.com.  IN SOA   ns1.domain.com. admin.domain.com. (
                          12083           ; serial number
                          3h              ; refresh interval
                          30m             ; retry interval
                          3w              ; expiry period
                          1h              ; negative TTL
                          )
```

Пояснимо, для чого призначена кожна частина:

- **domain.com.**: Це корінь зони. Вказує, що файл зони призначений для домену domain.com. Часто ви бачите, що це замінено на @, що є просто заповнювачем, який замінює вміст змінної $ORIGIN.

- **IN SOA**: Частина “IN” означає Інтернет (і вона буде присутня у багатьох записах). SOA є показником того, що це запис Start of Authority.
- **ns1.domain.com.**: визначає основний сервер імен для цього домену. Сервери імен можуть бути як основними, так і вторинними, і якщо налаштовано динамічний DNS, один сервер повинен бути “основним”. Якщо ви не налаштували динамічний DNS, то це лише один із ваших основних серверів імен.
- **admin.domain.com.**: Це електронна адреса адміністратора для цієї зони. "@" Замінено крапкою в електронній адресі. Якщо в іменній частині адреси електронної пошти, як правило, є крапка, вона замінюється на "\" в цій частині (your.name@domain.com стає your\name.domain.com).
- **12083**: Це серійний номер файлу зони. Кожного разу, коли ви редагуєте файл зони, ви повинні збільшувати це число, щоб файл зони розповсюджувався правильно. Вторинні сервери перевірять, чи не є серійний номер первинного сервера для зони більшим, ніж той, який вони мають у своїй системі. Якщо це так, вони запитує новий файл зони, якщо ні, продовжують обслуговувати оригінальний файл.
- **3h**: Інтервал оновлення для зони. Це кількість часу, який вторинний сервер буде чекати перед опитуванням первинного для зміни файлу зони.
- **30m**: Це інтервал повторної спроби для цієї зони. Якщо вторинний сервер не може підключитися до основного, коли закінчився період оновлення, він зачекає стільки часу і спробує опитати основний.
- **3w**: Це термін придатності. Якщо вторинний сервер імен не зміг зв’язатися з основним протягом такого періоду часу, він більше не повертає відповіді як авторитетне джерело для цієї зони.
- **1h**: Стільки часу сервер імен буде кешувати помилку імені, якщо він не може знайти запитане ім’я у цьому файлі.

1.7.2. A та AAAA записи.

Обидва ці записи відображають вузол на IP-адресу. Запис "A" використовується для зіставлення вузла з IP-адресою IPv4, тоді як записи "AAAA" використовуються для зіставлення вузла з адресою IPv6.

Загальний формат цих записів такий:

```
host     IN      A       IPv4_address
host     IN      AAAA    IPv6_address
```

Отож, оскільки наш запис SOA вказав основним сервером вузол “ns1.domain.com”, нам необхідно вказати його адресу, оскільки “ns1.domain.com” знаходиться в зоні “domain.com”, що цей файл є визначальним.

Запис може виглядати приблизно так:

```
ns1     IN  A       111.222.111.222
```

Зверніть увагу, що ми не повинні вказувати повне ім’я. Ми можемо просто вказати вузол без повного доменного імені, а сервер DNS заповнить решту значенням $ORIGIN. Однак ми могли б так само легко використовувати повне доменне ім’я:

```
ns1.domain.com.     IN  A       111.222.111.222
```

У більшості випадків тут ви визначаєте свій вебсервер як “www”:

```
www     IN  A       222.222.222.222
```

Ми також повинні сказати, як перетворюється базовий домен. Ми можемо зробити це так:

```
domain.com.     IN  A       222.222.222.222
```

Замість цього ми могли використати “@” для посилання на базовий домен:

```
@       IN  A       222.222.222.222
```

Ми також маємо можливість перетворити все, що в цьому домені не визначено явно для цього сервера. Ми можемо зробити це за допомогою символу "*":

```
*       IN  A       222.222.222.222
```

Усі вони так само добре працюють із записами AAAA для адрес IPv6.

1.7.3. CNAME записи.

Записи CNAME визначають псевдонім канонічного імені сервера (такий, який визначається записом A або AAAA).

Наприклад, можемо мати запис імені A, що визначає вузол “server1”, а потім використовувати “www” як псевдонім для цього вузла:

```
server1     IN  A       111.111.111.111
www         IN  CNAME   server1
```

Майте на увазі, що ці псевдоніми мають певні втрати продуктивності, оскільки вони потребують додаткового запиту до сервера. Здебільшого того самого результату можна було досягти, використовуючи додаткові записи A або AAAA.

Одним із випадків, коли рекомендується CNAME, є надання псевдоніма для ресурсу поза поточною зоною.

1.7.4. MX записи.

Записи MX використовуються для визначення поштових серверів, які використовуються для домену. Це допомагає електронним повідомленням правильно надходити на ваш поштовий сервер.

На відміну від багатьох інших типів записів, поштові записи зазвичай не відображають вузол до чогось, оскільки вони застосовуються до всієї зони. Вони зазвичай виглядають так:

```
        IN  MX  10   mail.domain.com.
```

Зверніть увагу, що на початку немає імені вузла.

Також зверніть увагу, що там є додатковий номер. Це номер налаштування, який допомагає комп’ютерам вирішити, на який сервер надсилати пошту, якщо визначено кілька поштових серверів. Менші цифри мають вищий пріоритет.

Запис MX, як правило, повинен вказувати на вузол, визначений записом A або AAAA, а не на такий, який визначений CNAME.

Отже, скажімо, у нас є два поштових сервери. Повинні бути записи, які виглядають приблизно так:

```
        IN  MX  10  mail1.domain.com.
        IN  MX  50  mail2.domain.com.
mail1   IN  A       111.111.111.111
mail2   IN  A       222.222.222.222
```

У цьому прикладі вузол “mail1” є найкращим сервером обміну електронною поштою.

Ми також могли б написати це так:

```
        IN  MX  10  mail1
        IN  MX  50  mail2
mail1   IN  A       111.111.111.111
mail2   IN  A       222.222.222.222
```

1.7.5. NS записи.

Цей тип запису визначає сервери імен, які використовуються для цієї зони.

Можливо, вам цікаво, "якщо файл зони знаходиться на сервері імен, чому йому потрібно посилатися на себе?". Частиною того, що робить DNS таким успішним, є його багаторівневе кешування. Однією з причин визначення серверів імен у файлі зони є те, що файл зони може фактично обслуговуватися з кешованої копії на іншому сервері імен.

Як і записи MX, це загальнозонові параметри, тому вони також не приймають вузли. Загалом вони виглядають так:

```
        IN  NS      ns1.domain.com.
        IN  NS      ns2.domain.com.
```

Ви повинні мати принаймні два сервери імен, визначені у кожному файлі зони, для забезпечення відмовостійкості. Більшість програм DNS-сервера вважає файл зони недійсним, якщо існує лише один сервер імен.

Як завжди, включіть відображення для вузлів із записами A або AAAA:

```
        IN  NS      ns1.domain.com.
        IN  NS      ns2.domain.com.
ns1     IN  A       111.222.111.111
ns2     IN  A       123.211.111.233
```

1.7.6. PTR записи.

Використовувані записи PTR визначають ім'я, пов'язане з IP-адресою. Записи PTR є оберненими до записів A або AAAA. Записи PTR унікальні тим, що починаються з кореня .arpa і

делегуються власникам IP-адрес. Регіональні реєстри Інтернету (RIR) керують делегуванням IP-адрес організаціям та постачальникам послуг. До регіональних Інтернет-реєстрів входять APNIC, ARIN, RIPE NCC, LACNIC та AFRINIC.

Ось приклад запису PTR для 11.22.33.44:

```
44.33.22.11.in-addr.arpa.   33692   IN  PTR host.example.com.
```

Цей приклад запису PTR для адреси IPv6 показує формат реверсного запису DNS-сервера Google IPv6 2001: 4860: 4860 :: 8888.

```
8.8.8.8.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.6.8.4.0.6.8.4.1.0.0.2.ip6.arpa. 86400IN PTR google-public-dns-a.google.com.
```

2. Завдання на роботу.

2.1. Встановити та налаштувати первинний та резервний сервери для двох прямих та зворотної зон, які відповідають наступним вимогам:

- в прямих зонах обов'язково мають бути записи типів: SOA, NS, A, CNAME, які відповідають варіанту завдання;
- в зворотній зоні обов'язково мають бути записи типів: SOA, NS, PTR, які відповідають варіанту завдання.

2.2. Перевірити результат передачі (transfer) зон на резервному сервері.

2.3. Перевірити працездатність первинного та резервного серверів для всіх зон за допомогою діагностичних утиліт, наприклад: nslookup, dig, host.

2.4. Рекомендується використовувати наступне програмне забезпечення:

- DNS-сервер: bind9.

Варіант	**Ім'я зон**	**Ресурси зони**		
		Ім'я	**Адреса**	**Псевдонім**
1	letter.net zone01.com	alpha	172.20.1.10	ws1
		beta	172.20.1.20	ws2
		gamma	172.20.1.30	ws3
		delta	172.20.1.40	ws4
		omega	172.20.1.50	ws5
2	planet.edu zone02.org	mercury	192.168.11.21	srv1
		venus	192.168.11.22	srv2
		earth	192.168.11.23	srv3
		saturn	192.168.11.24	srv4
		jupiter	192.168.11.25	srv5

Варіант	Ім'я зон	Ресурси зони		
		Ім'я	Адреса	Псевдонім
3	cat.com zone03.net	tiger	192.168.1.11	www
		lion	192.168.1.12	ftp
		lynx	192.168.1.13	ssh
		leopard	192.168.1.14	pop3
		jaguar	192.168.1.15	imap
4	flower.org zone04.edu	rose	172.20.1.31	pc1
		gerbera	172.20.2.32	pc2
		tulip	172.20.3.33	pc3
		aster	172.20.4.34	pc4
		peony	172.20.5.35	pc5
5	linux.net zone05.com	ubuntu	172.25.11.10	node1
		debian	172.25.11.20	node2
		centos	172.25.11.30	node3
		gentoo	172.25.11.40	node4
		fedora	172.25.11.50	node5
6	color.edu zone06.net	red	192.168.22.10	ws1
		green	192.168.22.20	ws2
		blue	192.168.22.30	ws3
		black	192.168.22.40	ws4
		white	192.168.22.50	ws5
7	metal.com zone07.org	gold	172.30.10.31	srv1
		silver	172.30.10.32	srv2
		iron	172.30.10.33	srv3
		copper	172.30.10.34	srv4
		zinc	172.30.10.35	srv5
8	capital.org zone08.edu	london	192.168.33.1	www
		tokyo	192.168.33.2	ftp
		paris	192.168.33.3	ssh
		rome	192.168.33.4	pop3
		berlin	192.168.33.5	imap

Варіант	Ім'я зон	Ресурси зони		
		Ім'я	Адреса	Псевдонім
9	currency.net zone09.com	dollar	192.168.55.10	pc1
		dinar	192.168.55.20	pc2
		lira	192.168.55.30	pc3
		peso	192.168.55.40	pc4
		real	192.168.55.50	pc5
10	river.edu zone10.net	nile	172.21.11.10	node1
		amazon	172.21.11.20	node2
		congo	172.21.11.30	node3
		amur	172.21.11.40	node4
		mekong	172.21.11.50	node5
11	fruit.com zone11.org	apple	172.31.50.1	ws1
		orange	172.31.50.2	ws2
		grape	172.31.50.3	ws3
		banana	172.31.50.4	ws4
		lemon	172.31.50.5	ws5
12	digit.org zone12.edu	one	192.168.77.11	srv1
		two	192.168.77.22	srv2
		three	192.168.77.33	srv3
		four	192.168.77.44	srv4
		five	192.168.77.55	srv5
13	month.net zone13.com	march	192.168.99.10	www
		april	192.168.99.20	ftp
		may	192.168.99.30	ssh
		june	192.168.99.40	pop3
		july	192.168.99.50	imap
14	name.edu zone14.org	maria	172.16.70.1	pc1
		tomas	172.16.70.2	pc2
		tereza	172.16.70.3	pc3
		stefan	172.16.70.4	pc4
		sara	172.16.70.5	pc5

Варіант	Ім'я зон	Ресурси зони		
		Ім'я	Адреса	Псевдонім
15	country.com zone15.net	france	192.168.88.11	node1
		china	192.168.88.12	node2
		spain	192.168.88.13	node3
		italy	192.168.88.14	node4
		germany	192.168.88.15	node5

3. Контрольні запитання.

3.1. Призначення та принципи роботи DNS.
3.2. Типи DNS-серверів та їх призначення.
3.3. Типи ресурсних записів в DNS та їх призначення.
3.4. Робота сервера в ітеративному та рекурсивному режимах.
3.5. Загальна процедура перетворення доменного імені в IP-адресу.
3.6. Кешування в DNS.

4. Література.

https://web.archive.org/web/20210412192956/https://book.systemsapproach.org/applications/infrastructure.html

https://web.archive.org/web/20210330175027/https://www.digitalocean.com/community/tutorials/an-introduction-to-dns-terminology-components-and-concepts

RFC1034 https://tools.ietf.org/html/rfc1034

RFC1035 https://tools.ietf.org/html/rfc1035