Лабораторна робота №3. Служба імен доменів DNS

Теоретичні відомості

Хоча програми теоретично можуть звертатися до веб-сторінок, поштових скриньок та інших ресурсів по мережевих адресах комп'ютерів (наприклад, IP), на яких зберігається дана інформація, користувачам важко запам'ятовувати такі адреси. Крім того, розміщення веб-сторінки компанії за адресою 128.111.24.41 означатиме, що в разі переїзду сервера компанії на нову машину, новий IP буде необхідно повідомити всім зацікавленим особам. Для відділення імен машин від їх адрес було вирішено використовувати зрозумілі імена високого рівня. Тому звернутися до веб-серверу компанії можна за адресою www.cs.washington.edu. Проте, так як мережа сама по собі розуміє тільки числові адреси, потрібен механізм перетворення імен в мережеві адреси. У наступних відображено, як проводиться це відображення в Інтернеті.

Колись давно в часи мережі ARPANET відповідність між текстовими та числовими адресами просто записувалося в файлі hosts.txt, в якому перераховувалися всі імена комп'ютерів і їх IP-адреси. Щоночі всі хости отримували цей файл з сайту, на якому він зберігався. В мережі, що складається з декількох сотень великих машин, що працюють під управлінням системи з поділом часу, такий підхід виправдовував себе.

Однак ще за довго до того, як до мережі були підключені мільйони комп'ютерів, стало зрозуміло, що цей спосіб не зможе працювати вічно. По-перше, розмір файлу рано чи пізно став би занадто великим. Однак, що ще важливіше, якщо управління іменами хостів не здійснюється централізовано, неминуче виникнення конфліктів імен. Водночас уявити собі централізоване управління іменами всіх хостів гігантської міжнародної мережі досить складно. Для вирішення вищезгаданих проблем в 1983 році і була розроблена служба імен доменів (DNS, Domain Name System). Відтоді вона стала найважливішою частиною Інтернету.

Суть системи DNS полягає в ієрархічній схемі імен, заснованої на доменах, і розподіленій базі даних, що реалізує цю схему імен. В першу чергу ця система використовується для перетворення імен хостів в IP-адреси, але також може використовуватися і в інших цілях. Визначення системи DNS дано в RFC 1034, 1035, 2181 і далі розроблено в багатьох інших.

В загальних рисах система DNS застосовується в такий спосіб. Для перетворення імені в IP-адресу прикладна програма звертається до бібліотечної процедурі, яка називається розпізнавачем (resolver), передаючи їй ім'я як параметр. Розпізнавач посилає запит, що містить ім'я, локальному DNS-серверу, який шукає ім'я і повертає відповідний IP-адресу розпізнавачів, який, в свою чергу, передає цю адресу прикладнії програмі. Запит і відповідь передають-ся як UDP-пакети. Маючи IP-адресу, програма може встановити TCP-з'єднання з адресатом або послати йому UDP-пакети.

Простір імен DNS

Управління великим і постійно змінюючимся набором імен являє собою нетривіальну задачу. В поштовій системі на листах потрібно вказувати (явно або неявно) країну, штат або область, місто, вулицю, номер будинку, квартиру і прізвище одержувача. Завдяки використанню такої ієрархічної схеми не виникає плутанини між Марвіном Андерсоном, що живуть на Мейн-стріт в Уайт-Плейнс, штат Нью-Йорк, і Марвіном Андерсоном з Мейн-стріт в Остіні, штат Техас. Система DNS працює аналогічно.

Для Інтернету основа ієрархії іменування розроблена організацією під на-званням ICANN (Internet Corporation for Assigned Names and Numbers - інтернет-корпорація з присвоєння імен та адрес). ICANN була створена для цих цілей в 1998 році, так як Інтернет розвинувся у всесвітній економічний концерн. Інтернет концептуально розділений на більш ніж 250 доменів верхнього рівня (top-level domains). Доменами називають в Інтернеті безліч хостів, об'єднаних в логічну групу. Кожен домен верхнього рівня підрозділяється на піддомени (subdomains), які, в свою чергу, також можуть складатися з інших доменів і т. Д. Всі ці домени можна розглядати у вигляді дерева, показаного на рис.1. Листям дерева є домени, не розділяються на піддомени (але складаються з хостів, звичайно). Такий кінцевий домен може складатися з одного хоста або може пред-ставлять компанію і містити в собі тисячі хостів.

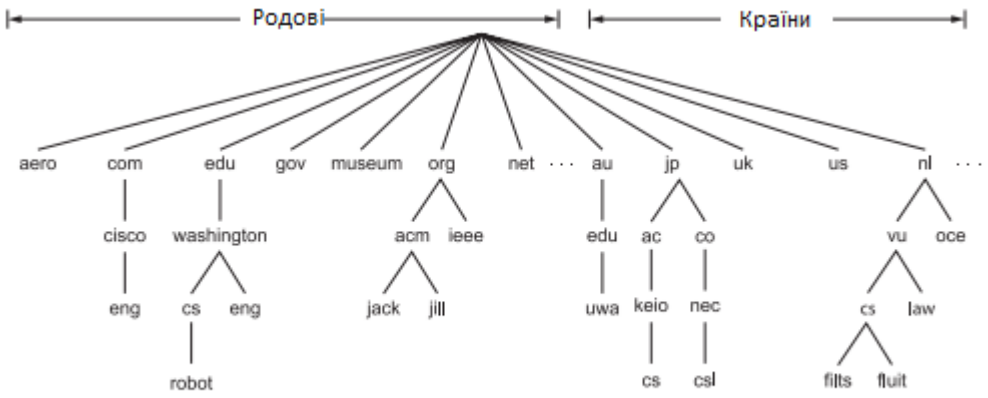


Рис.1 Частина доменного простіру імен

Домени верхнього рівня розділяються на дві групи: родові домени і домени держав. В майбутньому будуть додаватися нові базові домени вищого рівня.
За кожною державою відповідно до міжнародного стандарту ISO3166 закріплений один домен держави. Інтернаціоналізовані доменні імена країн, в яких використовується алфавіт, відмінний від латинського, були введені в 2010 році. Ці домени дозволяють іменувати хости, використовуючи арабські, кириличні, китайські та інші писемності.

Зарезервувати домен другого рівня, такий як імя_компанії.com, просто. Домени вищого рівня управляються реєстраторами (registrars), призначеними ICANN. Для того щоб отримати ім'я, потрібно просто звернутися до відповідного (в даному випадку com) і перевірити, чи доступне бажане ім'я і не є воно чиєїсь торговою маркою. Якщо все гаразд, замовник реєструється і за невелику щорічну абонентську плату отримує домен другого рівня.

Однак у міру комерціалізації та інтернаціоналізації Інтернету з'являється все більше спірних питань, особливо щодо іменування доменів. Ці суперечки захоплюють і саму ICANN. Деякі домени є самоорганизующимися, на інших існують обмеження і не всякий може отримати ім'я (як показано в табл.1). Але які обмеження доречні? Взяти хоча б домен pro. Він призначений для кваліфікованих фахівців. Але хто є фахівцем, а хто ні? Зрозуміло, що доктори і адвокати - це професіонали, суперечці немає. А що робити з вільними фотографами, вчителями музики, заклинателями, водопровідниками, перукарями, татуювальник, найманцями і повіями? Чи мають право кваліфіковані представники всіх цих та багатьох інших професій отримувати домени pro? Хто повинен це визначати?

Крім того, на іменах можна заробляти. Так країна Тувалу здала в оренду права на свій домен tv за $ 50 млн завдяки тому, що код країни відмінно підходить для реклами телевізійних сайтів. Практично всі загальновживані англійські слова використовуються як імена піддоменів com, разом з найбільш частими помилками. Спробуйте набрати якесь слово, яке стосується домашнього господарства, тварин, рослин, частин тіла т. д. У самої практики реєстрації доменних імен з метою їх подальшого продажу зацікавленій стороні навіть є назва - кіберсквоттінг (cybersquatting). Багато компаній, які виявилися не достатньо спритними в цьому питанні, виявили, що найочевидніші доменні імена вже зайняті, коли почалася ера Інтернету і вони спробували зареєструватися. У загальному і цілому, якщо не були порушені права на товарний знак і не було почато шахрайських дій, щодо імен працює правило «першим Запитив - першим отримав». Проте політика щодо вирішення спорів з приводу імен все ще не до кінця розроблена.

Ім'я кожного домена, подібно повному шляху до файлу в файлової системі, складається з шляху від цього домену до (безіменній) вершини дерева. Компоненти шляху поділуляють точками. Так, домен технічного відділу корпорації Cisco може виглядати як eng.cisco.com, а не так, як це прийнято в стилі UNIX (/ com / cisco / eng). Слід відзначити, що через таку ієрархічної системи найменування eng.cisco.com не конфліктує з потенційним використанням імені eng в домені eng.washington.edu, де він може позначати факультет англійської мови Вашингтонського університету. Імена доменів можуть бути абсолютними і відносними. Абсолютна ім'я домена завжди закінчується крапкою (наприклад, eng.cisco.com.), Тоді як відносне ім'я - ні. Для того щоб можна було єдиним чином визначити істинні значення відносних імен, вони повинні інтерпретуватися в деякому контек-сте. У кожному разі іменований домен означає певний вузол дерева і всі вузли під ним.

Імена доменів нечутливі до зміни регістра символів. Так, наприклад, edu, Edu і EDU означають одне і те ж. Довжина імен компонентів може досягати 63 символів, а довжина повного шляху не повинна перевершувати 255 символів.

В принципі, нові домени можуть додаватися в дерево з використанням як родового домена, так і домена, що позначає країну. Наприклад, cs.wasington.edu можна без проблем помістити в домен us під ім'ям cs.wasington.wa.us. На практиці, однак, майже всі організації в США поміщаються під родовими доменами, тоді як майже всі організації за межами США розташовуються під доменами своїх держав. Не існує будь-яких правил, що

забороняють реєстрацію під кількома до-менами верхнього рівня. Великі компанії часто саме так і чинять (на-приклад, sony.com, sony.net і sony.nl).

Кожен домен управляє розподілом доменів, розташованих під ним. Наприклад, в Японії домени ac.jp і co.jp відповідають американським доменах edu і com. В Голландії подібне відмінність не використовується, і всі домени організацій поміщаються прямо під доменом nl. Як приклад наведемо імена доменів факультетів обчислювальної техніки («комп'ютерних наук» - comuter science) трьох університетів.

1. cs.wasington.edu (Вашингтонський університет, США)

2. cs.vu.nl (університет Врійе, Нідерланди)

3. cs.keio.ac.jp (університет Кейо, Японія)

Для створення нового домену потрібен дозвіл домену, в який він буде включений. Наприклад, якщо у Вашингтонському університеті утворилася група VLSI, яка хоче зареєструвати домен vlsi.cs.wasington.edu, їй потрібно дозвіл від того, хто керує доменом cs.wasington.edu. Аналогічно, якщо створюється новий університет, наприклад, університет Північної Дакоти, він повинен попросити менеджера домена edu привласнити їх домену ім'я unsd.edu (якщо воно ще не зайнято). Таким чином, вдається уникнути конфлікту імен, а кожен домен відстежує стан всіх своїх піддоменів. Після того як домен створений і зареєстрований, в ньому можуть створюватися піддомени, наприклад cs.unsd.edu, для чого вже не потрібно дозволу вищестоящих доменів.

Структура доменів відображає не фізична будова мережі, а логічне поділ між організаціями та їх внутрішніми підрозділами. Так, якщо факультети обчислювальної техніки та електротехніки розташовуються в одній будівлі і користуються однією загальною локальною мережею, вони проте можуть мати різні домени. І навпаки, якщо, скажімо, факультет обчислювальної техніки розташовується в двох різних корпусах університету з різними локальними мережами, логічно всі хости обох будівель зазвичай належать до одного й того ж домену.

У кожного домена, незалежно від того, чи є він одиноким хостом або доменом верхнього рівня, може бути набір асоційованих з ним записів ресурсів (resource records). Ці записи є базою даних DNS. Для одинокого хоста запис ресурсів найчастіше представляє собою просто його IP-адресу, але існує також багато інших записів ресурсів. Коли распознаватель передає ім'я домена DNS-серверу, те, що він отримує назад, являє собою записи ресурсів, асоційоване з його ім'ям. Таким чином, істинне призначення системи DNS полягає в перетворенні доменних імен в записи ресурсів.

Запис ресурсу складається з п'яти частин. Хоча для ефективності вони часто пере-кодуються в двійкову форму, в більшості описів записи ресурсів представ-лени у вигляді ASCII-тексту, по одному рядку на запис ресурсу. Ми будемо використовувати наступний формат:

```
Domain_name  Time_to_live Class Type Value
```

Поле Domain_name (ім'я домена) позначає домен, до якого відноситься поточна запис. Зазвичай для кожного домена існує декілька записів ресурсів, і кожна копія бази даних зберігає інформацію про декілька доменах. Поле імені домена є первинним ключем пошуку, використовуваним для виконання запитів. Порядок записів в базі даних значення не має. У відповідь на запит про домен повертаються всі записи необхідного класу.

Поле Time_to_live (час життя) вказує, наскільки стабільно стан записі. Рідко мінливим даними присвоюється високе значення цього поля, наприклад, 86 400 (число секунд в добі). Непостійна інформація позначається невеликим значенням, наприклад, 60 (1 хвилина).

Третім полем кожного запису є поле Class (клас). Для інформації Інтернета значення цього поля завжди дорівнює IN. Для іншої інформації застосовуються інші коди, однак на практиці вони зустрічаються рідко.

Поле Type (тип) означає тип DNS-записи. Їх існує досить багато. Важливі типи записів перераховані в табл. 2.

Запис SOA (Start Of Authority - початкова точка повноважень) повідомляє ім'я первинного джерела інформації про зону сервера імен (описаного нижче), адреса електронної пошти його адміністратора, унікальний порядковий номер, різні прапори і тайм-аути.

Найважливішою є запис A (Address - адреса). Вона містить 32-розрядну IPv4-адрес уінтерфейсу для хоста. У відповідного запису AAAA («quad A» - «чотири A») є 128-розрядна IPv6-адреса. У кожного хоста в Інтернеті повинен бути щонайменше одна IP-адреса, щоб інші машини могли з ним спілкуватися. На деяких хостах може бути одночасно встановлено декілька мережевих з*єднань. В цьому випадку їм потрібно за дві або більше записи типу A або AAAA. Відповідно, DNS може видавати кілька адрес на одне ім'я.

Запис MX є стандартною. У ній вказується ім'я хоста, готового приймати пошту для зазначеного домену. Справа в тому, що не кожна машина може займатися прийомом пошти. Якщо хто-небудь хоче надіслати листа на адресу, наприклад bill @microsoft.com, то що відправляє хосту потрібно буде спочатку знайти поштовий сервер на microsoft.com. Запис MX може допомогти в цих пошуках.

Таблиця 2. Основні типи записів ресурсів DNS

Тип	Зміст	Значення
SOA	Початковий запис зони	Параметри для цієї зони
A	IPv4 адреса хосту	Ціле число, 32 десятичних розрядів
AAAA	IPv6 адреса хосту	Ціле число, 128 двійкових розрядів
MX	Обмін поштою	Приорітет, з яким домен буде приймати пошту
NS	Сервер імен	Ім'я серверу, для цього домену
CNAME	Канонічне ім'я	Ім'я домену
PTR	Вказівник	Псевдонім IP адреси
SPF	Правила відправки пошти	Правила відправки почти, закодованної у текстовому вигляді
SRV	Сервіс	Хост, доступний у даному вигляді
TXT	Текст	Не інтерпретований ASCII код

Ще один важливий тип запису - це NS. Запис NS містить інформацію про сервер імені для домену або піддомена. Це хост, на якому міститься копія бази даних для домена. Він використовується в процесі пошуку імені, тому ми коротенько опишемо цей процес.

Запіс CNAME дозволяють створювати псевдоніми. Уявімо собі, що людина, знайомий в загальних рисах з формуванням імен в Інтернеті, хоче послати повідомлення користувачеві paul на відділенні обчислювальної техніки Массачусетського технологічного інституту (MIT). Він може спробувати вгадати потрібний йому адресу, склавши рядок paul@cs.mit.edu. Однак ця адреса працювати не буде, так як домен відділення обчислювальної техніки Массачусетського технологічного інституту насправді називається csail.mit.edu. Таким чином, для зручності тих, хто цього не знає, MIT може створити запис CNAME, що дозволяє звертатися до потрібного домену за обома іменах. Такий запис буде мати наступний вигляд:

```
cs.mit.edu        86400        IN CNAME csail.mit.edu
```

Як і CNAME, запис PTR вказує на інше ім'я. Однак на відміну від запису CNAME, що є, власне, Макровизначення (тоесть механізмом заміни одного рядка інший), PTR являє собою звичайний тип даних DNS, інтерпретація якого залежить від контексту. На практиці запис PTR майже завжди використовується для асоціації імені з IP-адресою, що дозволяє за IP-адресою знаходити ім'я відповідної машини. Це називається зворотним пошуком (reverse lookups).

Запис SRV - це новий тип, що дозволяє визначати хост для шуканого сервісу в домені. Наприклад, веб-сервер для cs.wasington.edu може бути визначений як cockatoo.cs.wasington.edu. Даний запис є розширеним варіантом записи MX, яка виконує ту ж задачу в рамках поштових сервісів.

SPF - також новий тип запису. Він дозволяє домену закодувати інформацію про те, які машини будуть відсилати з нього листи в іншу частину Інтернету. Це допомагає приймаючим машинам перевіряти, чи припустима дана пошта. Якщо пошта приходить з машини, яка називається dodgy, а доменні записи говорять про те, що пошта з домена буде

надсилатися тільки машиною під назвою smtp, великі шанси того, що дані повідомлення є спамом.

Останні в списку, TXT-записи спочатку призначалися для того, щоб дозволити доменах ідентифікувати себе довільним чином. Сьогодні з їх допомогою зазвичай кодується інформація, призначена для зчитування машиною, звичайно це SPF-інформація.

Нарешті, останнє поле записи ресурса- це полеValue (значення) - може бути числом, ім'ям домена або текстової ASCII-рядком. Сенс поля залежить від типу запису. Короткий опис поля Value для кожного з основних типів записів дано в табл. 2.

Приклад інформації, що зберігається в базі даних DNS домену, наведено в лістингу 1. У ньому показана частина (гіпотетичної) бази даних домену cs.vu.nl, представленого також у вигляді вузла дерева доменів на рис.2 У базі даних міститься сім типів записів ресурсів

```
: Официальная информация для cs.vu.nl
cs.vu.nl.      86400  IN  SOA  star boss (9527,7200,7200,241920,86400)
cs.vu.nl.      86400  IN  MX   1 zephyr
cs.vu.nl.      86400  IN  MX   2 top
cs.vu.nl.      86400  IN  NS   star

star           86400  IN  A    130.37.56.205
zephyr         86400  IN  A    130.37.20.10
top            86400  IN  A    130.37.20.11
www            86400  IN  CNAME  star.cs.vu.nl
ftp            86400  IN  CNAME  zephyr.cs.vu.nl

flits          86400  IN  A    130.37.16.112
flits          86400  IN  A    192.31.231.165
flits          86400  IN  MX   1 flits
flits          86400  IN  MX   2 zephyr
flits          86400  IN  MX   3 top

rowboat           IN  A    130.37.56.201
                  IN  MX   1 rowboat
                  IN  MX   2 zephyr

little-sister     IN  A    130.37.62.23

laserjet          IN  A    192.31.231.216
```

Лістинг 1. Частина можливої бази даних домену cs.vu.nl

В першому не закоментувавши рядки лістінга 1 дається основна інформація про домен, яка в подальшому нас цікавити не буде. Наступні два рядки визначають два хоста, з якими слід зв'язатися в першу чергу при спробі доставити електронну пошту, надіслану за адресою person@cs.vu.nl. Хост на ім'я zephyr (спеціальна машина) слід опитати першим. У разі невдачі слід спробувати доставити лист машині по імені top. У наступному рядку визначений сервер імен для домену star.

Після порожніх рядків, доданих для зручності читання, йдуть рядки, що повідомляють IP-адреси для star, zephyr і top. Далі слід псевдонім www.cs.vu.nl, дозволяючий не звертатися до якоїсь конкретної машині. Створення цього псевдоніма дозволяє домену cs.vu.nl змінювати свій WWW-сервер, не змінюючи адреси, за якою користувачі зможуть продовжувати до нього звертатися. Те ж справедливо і для домену ftp.cs.vu.nl - FTP-сервера.

У секції, призначеної для машини flits, перераховані два IP-адреси і три можливі варіанти адреси для обробки пошти, надісланої на flits.cs.vu.nl. В першу чергу, природно, слід намагатися доставити лист самому комп'ютеру flits. Але якщо цей хост вимкнений, слід продовжувати спроби, звертаючись до хостів zephyr itop. Наступні три рядки містять типові записи для комп'ютера, в даному випадку для rowboat.cs.vu.nl. Зберігається в базі даних інформація містить IP-адресу, а також імена першого і другого хостів для доставки пошти. Слідом йде запис про машину, яка сама не здатна отримувати пошту. Останній рядок, ймовірно, описує лазерний принтер, підключений до Інтернету.

Сервери імен

Теоретично один сервер міг би містити всю базу даних DNS і відповідати на всі запити до неї. На практиці цей сервер виявився б настільки перевантаженим, що був би просто марним. Більш того, якби з ним коли-небудь що-небудь сталось, то весь Інтернет не працював би.

Щоб уникнути проблем, пов'язаних із зберіганням всієї інформації в одному місці, простір імен DNS розділене на не перехресні зони (zones). Один можливий спосіб поділу простору імен, показаного на рис.1, на зони зображений на рис. 3. Кожна окреслена зона містить частину загального дерева доменів.

Розстановка меж зон цілком залежить від адміністратора зони. Це рішення ґрунтується на тому, скільки серверів імен вимагається в тій чи іншій зоні. Напри заходів, на рис. 2 у Вашингтонського університету є зона для washington.edu, яка керує доменом eng.washington.edu, але не доменом cs.washington.edu, розташованим в окремій зоні зі своїми серверами імен. Подібне рішення може бути ухвалене, коли факультет англійської мови не хоче управляти власним сервером імен, але цього хоче факультет обчислювальної техніки.

Кожна зона також асоціюється з одним або більше сервером імен. Це хости, на яких знаходиться база даних для зони. Зазвичай у зони є один основний сервер імен, який отримує інформацію з файлу на своєму диску, і один або більше другорядних серверів імен, які отримують інформацію з основного сервера імен. Для підвищення надійності деякі сервери імен можуть бути розташовані поза зоною.

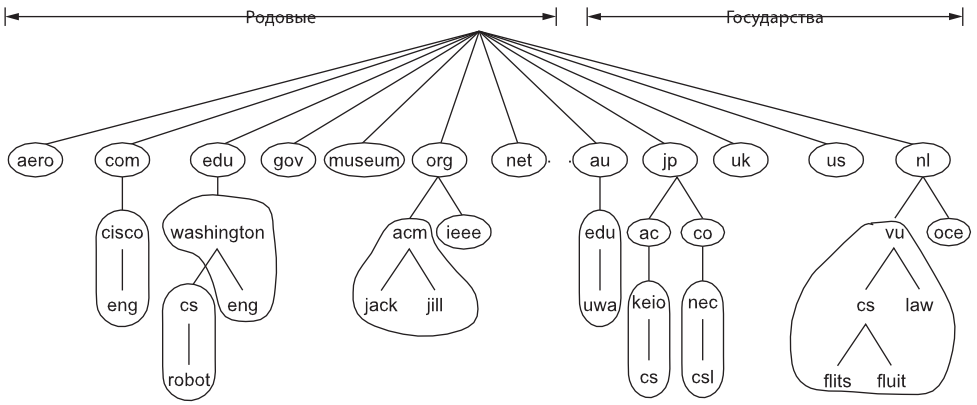
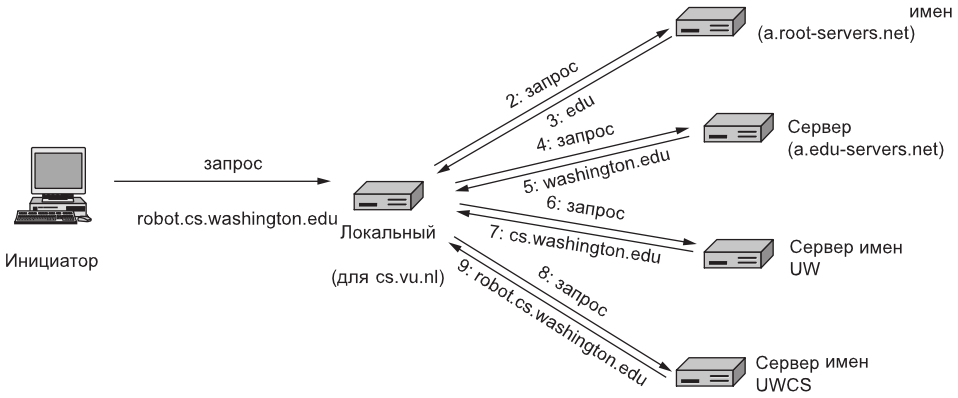


Рис. 3. Приклад пошуку розпізнувачем ім’я та віддаленого хосту в десяти кроках

Процес пошуку адреси по імені називається дозволом імен (name resolution). Розпізнувач звертається із запитом дозволу імені домена до локального серверу імен. Якщо шуканий домен належить до сфери відповідальності даного сервера імен, як, наприклад, домен top.cs.vu.nl підпадає під юрисдикцію домена cs.vu.nl, тоді даний DNS-сервер сам відповідає розпізнавачів на його запит, передаючи йому авторитетну запис (authoritative record) ресурсу. Авторитетної називають запис, одержувану від офіційного джерела, котра береже даний запис і керуючого її станом. Тому такий запис завжди вважається вірним, на відміну від кешованих записів (cached records), які можуть застарівати.

Однак що відбувається, якщо домен віддалений, як, наприклад, у випадку, коли flits.cs.vu.nl намагається знайти IP-адресу для robot.cs.washington.edu у Вашингтонському університеті? В цьому випадку, якщо в кеші немає інформації про запитуваній домені, доступному локально, сервер імен посилає віддалений запит. Пояснимо даний процес на прикладі, показаному на рис. 4. На першому кроці (позначений «1») надсилається запит локального сервера імен. Цей запит містить ім'я шуканого домену, тип (A) і клас (IN).



На наступному кроці посилається запит на один з кореневих серверів імен (root name servers), що знаходяться на вершині ієрархії. На цих серверах імен зберігається інформація про кожного домені вищого рівня. Цей запит показаний як крок 2 на рис. 3. Щоб

зв'язатися з кореневим сервером, на кожному сервері імен повинна бути інформація про один або більше кореневих серверах імен. Зазвичай ця інформація представлена в файлі системної конфігурації, який завантажується в кеш DNS, коли запускається сервер DNS. Він є просто списком записів NS і відповідних записів A.

Існує 13 кореневих серверів DNS, які називаються нехитро - від a-root-servers.net до m.root-servers.net. Кожен кореневий сервер логічно міг би бути окремим комп'ютером. Однак так як весь Інтернет залежить від кореневих серверів, вони є потужними машинами, а інформація, що зберігається на них, неодноразово дублюється. Більшість серверів розташоване в різних географічних точках, і доступ до них здійснюється за допомогою адресації будь-якого пристрою з групи, при цьому пакет доставляється на найближчий адресу (ми описали адресацію будь-якого пристрою групи у п'ятому розділі). Дублювання інформації підвищує надійність і продуктивність.

Малоймовірно, щоб цей кореневий сервер імен знав адресу машини в Вашингтонськім університеті. Швидше за все, він навіть не знає адреси сервера імен самого університету, однак він повинен знати сервер імен домену edu, на якому розташований cs.washington.edu. Він повертає ім'я та IP-адреса для частини відповіді на третьому кроці. Далі локальний сервер імен продовжує цей складний шлях. Він направляє запит серверу імен edu (a.edu-servers.net), який видає ім'я сервера Вашингтонського університету. Цей процес проілюстрований шагамі4 і 5. Тепер ми вже підійшли ближче. Локальний сервер імен відсилає запит на сервер імен Вашингтонського уні-верситету (крок 6). Якщо шукане ім'я домена знаходиться на факультеті англійської мови, буде отримана відповідь, так як зона університету цей факультет охоплює. Але факультет обчислювальної техніки вирішив запустити власний сервер імен. Запит повертає ім'я та IP-адреса сервера імен факультету обчислювальної техніки Вашингтонського університету (крок 7).

Нарешті, локальний сервер імен запитує сервер імен факультету обчисли-котельної техніки Вашингтонського університету (крок 8). Цей сервер відповідає за до-мен cs.washington.edu, так що він повинен видати відповідь. У підсумку остаточний відповідь повертається (крок 9), і локальний сервер імен передає його на flits.cs.vu.nl (крок 10). Ім'я отримано.

Ви можете вивчити цей процес, використовуючи стандартні програми типу dig, які встановлені на більшості UNIX-систем. Наприклад, надрукувавши ви відправите запит robot.cs.washington.edu на сервер імен a.edu-servers.net і отримаєте роздруківку результату. Так ви побачите інформацію, яку ми отримали на четвертому кроці в нашому прикладі, і дізнаєтеся ім'я та IP-адреси серверів імен Вашингтонського університету.

У цьому довгому сценарії є три технічні моменту, що вимагають пояснень. По-перше, на рис. 3 використовується два різних механізму запиту. Коли хост flits. cs.vu.nl відсилає запит на локальний сервер імен, цей сервер виконує запит від імені flits, поки не отримає відповідь, яку можна буде повернути. Він не повертає часткових відповідей. Вони можуть бути корисними, але в запиті про них немає ні слова. Цей механізм називається рекурсивним запитом (recursive query).

З іншого боку, кореневий сервер імен (і кожний наступний) не продовжує рекурсивно запит локального сервера імен. Він повертає лише часткову відповідь і переходить до

наступного запиту. Локальний сервер імен відповідає за продовження пошуку відповіді, спрямовуючи наступні запити. Цей механізм називається ітеративним запитом (iterative query).

В одному процесі пошуку імені можуть бути задіяні обидва механізму, як показано в цьому прикладі. Рекурсивні запити практично завжди здаються кращими, але багато серверів імен (особливо кореневі) їх не обробляють. Вони занадто завантажені. Ітеративні запити накладають вантаж обробки запиту на ту машину, яка їх породжує. Для локального сервера імен розумно підтримувати рекурсивні запити, щоб надавати сервіс хостам на своєму домені. Ці хости не обов'язково повинні бути налаштовані таким чином, щоб оббігати всі сервери імен, їм потрібна лише можливість звернутися до локального.

Друге, на чому варто загострити увагу, - це кешування. Всі відповіді, в тому числі всі повернуті часткові відповіді, зберігаються в кеші. Таким чином, якщо інший хост cs.vu.nl запрошувати robot.cs.washington.edu, відповідь буде вже відомий. Більш того, якщо хост запрошувати інший хост на тому ж домені, наприклад galah. cs.washington.edu, відповідь може бути відісланий безпосередньо на сервер імен, який відповідає за це ім'я. Подібним чином запити на інші домени на washington.edu можуть починатися безпосередньо з сервера імен washington.edu. Використання відповідей, збережених в кеші, серйозно скорочує кількість кроків в запиті і підвищує продуктивність. Сценарій, який ми накидали, насправді, є гіршим з можливих варіантів, так як в кеші немає корисної інформації.

Однак відповіді, збережені в кеші, не є авторитетними, так як зміни в домені cs.washington.edu не розповсюджуватимуться автоматично на всі кеші, в яких може зберігатися копія цієї інформації. З цієї причини записи кеша зазвичай довго не живуть. В кожного запису ресурсу присутня поле Time_to_live. Воно повідомляє віддалених серверів, наскільки довго слід зберігати цей запис в кеші. Якщо яка-небудь машина зберігає постійна адреса роками, можливо, буде достатньо надійно зберігати цю інформацію в кеші протягом одного дня. Для більш непостійній інформації, ймовірно, більш обачно видаляти всі записи через кілька секунд або одну хвилину.

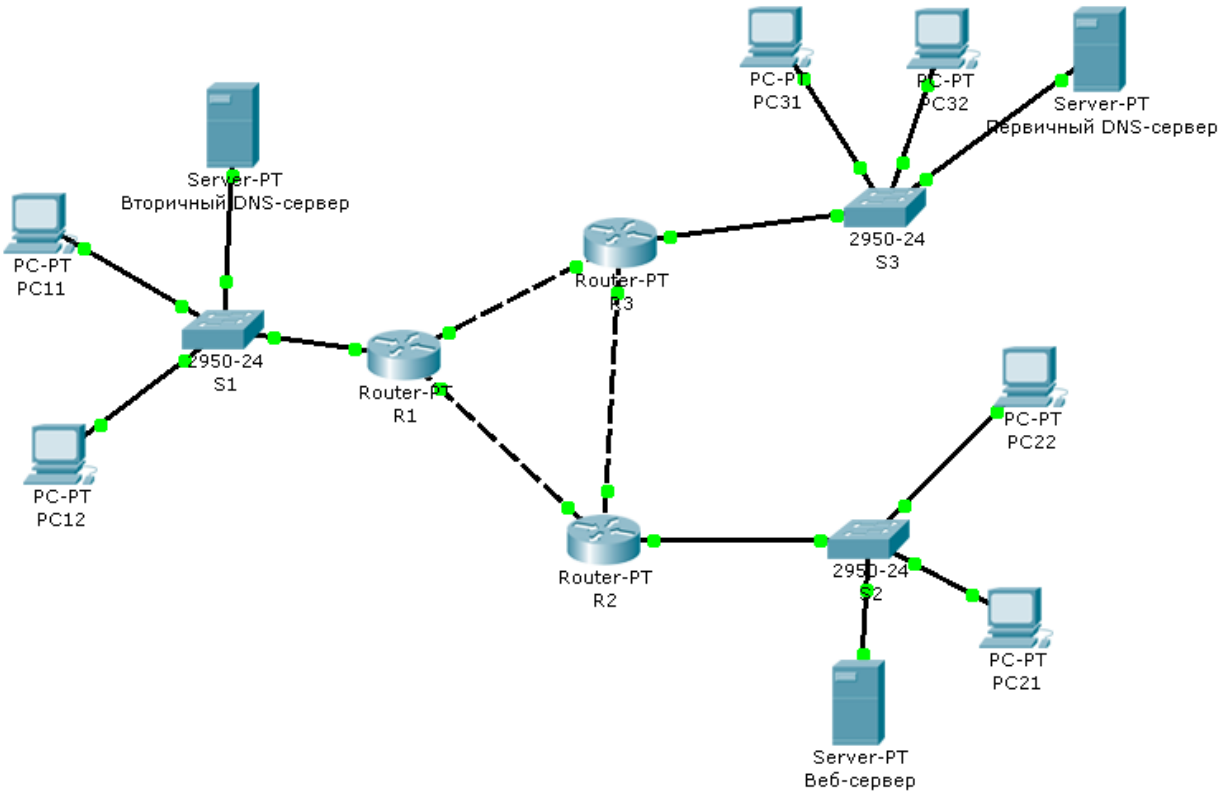
Завдання на лабораторну роботу

Налаштування DNS-сервера в середовищі Cisco Packet Tracer.

У лабораторнsq роботі необхідно забезпечити доступ до сайту з будь-якого з комп'ютерів мережі. Для роботи з сайтом не по IP, а по імені використовується служба DNS (яка знаходить відповідність між IP-адресою і DNS-ім'ям (якщо таке є) і навпаки).

В цілях надійності використовується не тільки первинний, а й вторинний DNS-сервер, який зберігає копію бази даних первинного DNS-сервера.

№1. Створити мережу за зразком.



№2. Налаштувати адресацію у відповідності з таблицею

	Роутер в мережі	**Адреса мережі**	**Маска**
Мережа №1	R1	192.168.1.0	255.255.255.128
Мережа №2	R2	192.168.2.0	255.255.255.128
Мережа №3	R3	192.168.3.0	255.255.255.128
Мережа №4	R1,R2	10.0.0.0	255.0.0.0
Мережа №5	R1,R3	20.0.0.0	255.0.0.0
Мережа №6	R2,R3	30.0.0.0	255.0.0.0

(визначте клас, кожній з мереж і маску в префиксной формі).

№3. Налаштуйте в мережі маршрутизацію по OSPF..

Налаштування DNS.

№4. В налаштуваннях DNS-сервера створіть A-запис, в якій вказати відповідність між DNS-ім'ям та IP-адресою веб-сервера.

Наприклад: 192.168.1.1 □ test.local

З комп'ютера зайдіть на вказаний сайт по DNS-імені.

Сайт повинен бути доступний з будь-якого комп'ютера мережі.

№5. Налаштуйте для створеного домена аліас (друге ім'я) (test1.ru).

Для цього налаштуйте запис CNAME. Вкажіть два імені - ім'я вузла та ім'я вузла, яке також йому відповідає.

Часто зустрічаються ситуації, коли це потрібно:

Наприклад, будь-який сайт має адресу як з www, так і без www (yandex.ru і www.yandex.ru)

№6. Налаштуйте запис SOA.

SOA-запис (Start Of Authority) - запис SOA містить ім'я первинного DNS-сервера (Primary Name Server), адреса, необхідний для встановлення технічних контактів (Hostmaster), серійний номер (Serial number) різні значення таймерів (Refresh, Retry, Expire, Minimum TTL)

Serial number

Serial number (серійний номер) - це номер версії файлу зони. Цей номер повинен бути позитивним цілим числом і збільшуватися щоразу, коли в файл зони вносяться зміни. Збільшення серійного номера показує вторинним серверам, що зона змінена, і що їм необхідно оновити у себе зону.

В програмі Cisco Packet Tracer дана записуються такі поля:

• DNS-ім'я ресурсу.

• Primary Name Server - адреса первинного DNS-сервера для даного домена.

• Minimum TTL - визначає "час життя" негативних відповідей на запити про ресурси, що не існують в DNS. Допустимі значення: не менше 5 хвилин.

• Retry Time - показує, як довго вторинний сервер імен повинен чекати, перед тим як повторити спробу запиту первинного сервера (на предмет змін серійного номера даної зони), якщо попередня спроба виявилася невдалою.

• Expire Time - вказує верхнє обмеження за часом, протягом якого вторинний сервер може використовувати раніше отримані дані про зону до того як вони втратять силу через відсутність оновлення (наприклад, внаслідок відключення первинного сервера імен на тривалий час).

• Refresh Time - часовий параметр Refresh показує як часто вторинні сервери повинні запитувати первинний сервер, щоб дізнатися, чи не збільшився чи Serial number (серійний номер) зони і, отже, чи не потрібно оновити її у себе.

• Mail box - поштову адресу відповідальної особи.

Для домена test1.ru налаштуйте всі зазначені вище параметри, у відповідність з таблицею:

Тип запису:	**Задаємо в Cisco Packet Tracer:**
Primary NameServer	IP-адреса вашого серверу
Minimum TTL	3600
Refresh Time	3600
Expire Time	86400
Mailbox	Ваш e-mail.
Minimum TTL	300

Примітка: з метою наочності - задані досить короткі інтервали часу. В реальності задають не менше години.

№7. Налаштувати запис NS.

Запис NS (name server) вказує на DNS-сервер для даного домена.

Для стабільної роботи домену вказується не менше двох NS-записів.

У разі недоступності одного з DNS-серверів відбувається запит на інший DNS-сервер.

Приклад:

Інформація про домен MAIL.RU

domain: MAIL.RU

nserver: ns1.mail.ru. 94.100.179.159

nserver: ns2.mail.ru. 94.100.186.189

nserver: ns3.mail.ru. 94.100.179.93

nserver: ns4.mail.ru. 94.100.178.100

nserver: ns5.mail.ru. 217.69.129.241

nserver: ns.mail.ru. 217.69.129.230

№8. В попередніх завданнях були створені і налаштовані два DNS-сервера. Вимкніть один DNS-серверів і зайдіть після цього на веб-сайт.

Визначте і поясніть результат.