

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"**

ІКНІ
Кафедра ПЗ



ЗВІТ

До лабораторної роботи №12

на тему: “Дослідження роботи DNS сервера та протоколу DHCP.”

з дисципліни: “Організація комп’ютерних мереж”

Лектор:
доцент кафедри ПЗ
Крук О.Г.

Виконав:
студент групи ПЗ-24
Губик А. С.

Прийняв:
доцент кафедри ПЗ
Задорожний І. М.

Тема роботи: Дослідження роботи DNS сервера та протоколу DHCP.

Мета роботи: Вивчити принципи роботи DNS, на практиці ознайомитися з принципами роботи DNS-клієнта на прикладі утиліти nslookup, детально дослідити формат DNS-запиту (і відповіді) за допомогою Wireshark і nslookup, а також ознайомитися з DHCP-повідомленнями.

Теоретичні відомості

Для обміну даними між вузлами мережі використовуються IP-адреси. Однак, людям простіше запам'ятовувати символьні імена, наприклад, google.com (а не 173.194.39.64) або somename@gmail.com (а не somename@113.108.11.220). Є і інша причина для застосування символьних імен: якщо поштовий сервер змінить IP-адресу, символьне ім'я не міняється і користувачеві не доводиться міняти адресу електронної пошти. Тому, оскільки люди оперують символьними іменами, а машини – чисельними, то в мережах повинен існувати якийсь механізм для того, щоб символьним адресам ставити у відповідність IP-адреси (часто говорять: «відображати символьні імена на IP-адреси»). На початку розвитку мереж (в мережі ARPAnet, з якої походить Інтернет) цей механізм був таким. Існував файл hosts.txt, в якому містилася вся інформація про відповідність всіх символьних імен вузлів і їхніх IP-адрес. Цей файл зберігався на одному вузлі мережі ARPAnet і в нього при потребі вносилися зміни (наприклад, додавалася інформація про нові вузли). Інші хости зберігали в себе копії файлу hosts.txt, періодично завантажуючи поновлену версію hosts.txt зі згаданого «основного» вузла. З ростом мережі ARPAnet описаний механізм відображення символьних імен в IP-адреси став неприйнятним з декількох причин. По-перше, очевидно, що з неминучим ростом мережі файл hosts.txt розрісся б непомірно і синхронізація його «центральної» версії з копіями на всіх хостах мережі була б проблематичною – є різниця в опрацюванні записів про кілька сотень і про кілька тисяч вузлів! А по-друге, рано чи пізно трапився б конфлікт імен. Для розв'язання цих проблем було створено DNS (Domain Name System) – систему доменних імен. Визначення системи DNS дається у RFC 1034 та 1035. DNS-система включає в себе три основні компоненти: У DNS дані про відповідність символьних імен і IP-адрес зберігається у розподілених базах даних (дані фізично «розкидані» по різних серверах у всьому світі). DNS-сервери відповідають на запити DNS-клієнтів, шукаючи у базах даних затребувані клієнтами дані про доменні імена. Ключем пошуку даних є доменне ім'я. Спрощено схема відображення символьних імен на IP-адреси виглядає так. Нехай прикладна програма (наприклад, браузер) «знає» символьну адресу і для встановлення TCP-з'єднання потребує IP-адресу. Ця прикладна програма звертається до бібліотечної процедури, яка називається «перетворювач IP-адрес» (“resolver”), передаючи символьне ім'я як параметр цієї процедури. Resolver звертається до DNS-сервера, отримує у відповідь від DNS-сервера IP-адресу та передає її прикладній програмі.

Хід роботи



```
Administrator: Windows PowerShell
> google.com
Server: UnKnown
Address: 190.168.1.1

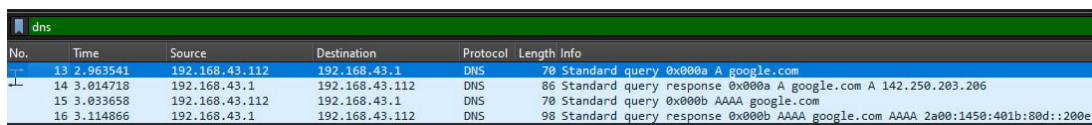
Non-authoritative answer:
Name: google.com
Addresses: 2a00:1450:401b:80d::200e
          142.250.186.206

> wikipedia.org
Server: UnKnown
Address: 190.168.1.1

Non-authoritative answer:
Name: wikipedia.org
Addresses: 2620:0:860:ed1a::1
          208.80.153.224

>
```

Рис. 1:



No.	Time	Source	Destination	Protocol	Length	Info
13	2.963541	192.168.43.112	192.168.43.1	DNS	70	Standard query 0x000a A google.com
14	3.014718	192.168.43.1	192.168.43.112	DNS	86	Standard query response 0x000a A google.com A 142.250.203.206
15	3.033658	192.168.43.112	192.168.43.1	DNS	70	Standard query 0x000b AAAA google.com
16	3.114866	192.168.43.1	192.168.43.112	DNS	98	Standard query response 0x000b AAAA google.com AAAA 2a00:1450:401b:80d::200e

Рис. 2: DNS пакети

```
Administrator: Windows PowerShell

> set d2
> google.com
Server: UnKnown
Address: 190.168.1.1

-----
SendRequest(), len 28
HEADER:
  opcode = QUERY, id = 8, rcode = NOERROR
  header flags: query, want recursion
  questions = 1, answers = 0, authority records = 0, additional = 0

  QUESTIONS:
    google.com, type = A, class = IN
-----
Got answer (44 bytes):
HEADER:
  opcode = QUERY, id = 8, rcode = NOERROR
  header flags: response, want recursion, recursion avail.
  questions = 1, answers = 1, authority records = 0, additional = 0

  QUESTIONS:
    google.com, type = A, class = IN
  ANSWERS:
    -> google.com
        type = A, class = IN, dlen = 4
        internet address = 142.250.186.206
        ttl = 1477 (24 mins 37 secs)
-----
Non-authoritative answer:
-----
SendRequest(), len 28
HEADER:
  opcode = QUERY, id = 9, rcode = NOERROR
  header flags: query, want recursion
  questions = 1, answers = 0, authority records = 0, additional = 0

  QUESTIONS:
    google.com, type = AAAA, class = IN
-----
Got answer (56 bytes):
HEADER:
  opcode = QUERY, id = 9, rcode = NOERROR
  header flags: response, want recursion, recursion avail.
  questions = 1, answers = 1, authority records = 0, additional = 0

  QUESTIONS:
    google.com, type = AAAA, class = IN
  ANSWERS:
    -> google.com
        type = AAAA, class = IN, dlen = 16
        AAAA IPv6 address = 2a00:1450:401b:80d::200e
        ttl = 1509 (25 mins 9 secs)
-----
Name: google.com
Addresses: 2a00:1450:401b:80d::200e
          142.250.186.206

>
```

Рис. 3:

```
Administrator: Windows PowerShell
> set d2
> sdfsd.org
Server: UnKnown
Address: 127.0.0.1

-----
SendRequest(), len 27
  HEADER:
    opcode = QUERY, id = 3, rcode = NOERROR
    header flags: query, want recursion
    questions = 1, answers = 0, authority records = 0, additional = 0

    QUESTIONS:
      sdfsd.org, type = A, class = IN
-----
recvfrom: No error
SendRequest failed
-----
SendRequest(), len 27
  HEADER:
    opcode = QUERY, id = 4, rcode = NOERROR
    header flags: query, want recursion
    questions = 1, answers = 0, authority records = 0, additional = 0

    QUESTIONS:
      sdfsd.org, type = AAAA, class = IN
-----
recvfrom: No error
SendRequest failed
*** UnKnown can't find sdfsd.org: No response from server
>
```

Рис. 4:

```
Administrator: Windows PowerShell

> set recursive
> mail.google.com
Server: UnKnown
Address: 192.168.43.1

-----
SendRequest(), len 33
HEADER:
  opcode = QUERY, id = 16, rcode = NOERROR
  header flags: query, want recursion
  questions = 1, answers = 0, authority records = 0, additional = 0

  QUESTIONS:
    mail.google.com, type = A, class = IN
-----
Got answer (49 bytes):
HEADER:
  opcode = QUERY, id = 16, rcode = NOERROR
  header flags: response, want recursion, recursion avail.
  questions = 1, answers = 1, authority records = 0, additional = 0

  QUESTIONS:
    mail.google.com, type = A, class = IN
  ANSWERS:
    -> mail.google.com
      type = A, class = IN, dlen = 4
      internet address = 142.250.203.133
      ttl = 266 (4 mins 26 secs)
-----
Non-authoritative answer:
-----
SendRequest(), len 33
HEADER:
  opcode = QUERY, id = 17, rcode = NOERROR
  header flags: query, want recursion
  questions = 1, answers = 0, authority records = 0, additional = 0

  QUESTIONS:
    mail.google.com, type = AAAA, class = IN
-----
Got answer (61 bytes):
HEADER:
  opcode = QUERY, id = 17, rcode = NOERROR
  header flags: response, want recursion, recursion avail.
  questions = 1, answers = 1, authority records = 0, additional = 0

  QUESTIONS:
    mail.google.com, type = AAAA, class = IN
  ANSWERS:
    -> mail.google.com
      type = AAAA, class = IN, dlen = 16
      AAAA IPv6 address = 2a00:1450:401b:80d::2005
      ttl = 262 (4 mins 22 secs)
-----
Name: mail.google.com
Addresses: 2a00:1450:401b:80d::2005
          142.250.203.133
>
```

Рис. 5:

```
Administrator: Windows PowerShell

ttl = 47 (47 secs)

-----
Name:      mail.google.com
Addresses: 2a00:1450:401b:804::2005
          142.250.186.197

> set norecursive
> mail.google.com
Server: UnKnown
Address: 192.168.43.1

-----
SendRequest(), len 33
HEADER:
  opcode = QUERY, id = 14, rcode = NOERROR
  header flags: query
  questions = 1, answers = 0, authority records = 0, additional = 0

QUESTIONS:
  mail.google.com, type = A, class = IN
-----
Got answer (49 bytes):
HEADER:
  opcode = QUERY, id = 14, rcode = NOERROR
  header flags: response, recursion avail.
  questions = 1, answers = 1, authority records = 0, additional = 0

QUESTIONS:
  mail.google.com, type = A, class = IN
ANSWERS:
-> mail.google.com
   type = A, class = IN, dlen = 4
   internet address = 142.250.186.197
   ttl = 16 (16 secs)

-----
Non-authoritative answer:
-----
SendRequest(), len 33
HEADER:
  opcode = QUERY, id = 15, rcode = NOERROR
  header flags: query
  questions = 1, answers = 0, authority records = 0, additional = 0

QUESTIONS:
  mail.google.com, type = AAAA, class = IN
-----
Got answer (61 bytes):
HEADER:
  opcode = QUERY, id = 15, rcode = NOERROR
  header flags: response, recursion avail.
  questions = 1, answers = 1, authority records = 0, additional = 0

QUESTIONS:
  mail.google.com, type = AAAA, class = IN
ANSWERS:
-> mail.google.com
   type = AAAA, class = IN, dlen = 16
   AAAA IPv6 address = 2a00:1450:401b:80d::2005
   ttl = 15 (15 secs)

-----
Name:      mail.google.com
Addresses: 2a00:1450:401b:80d::2005
          142.250.186.197

>
```

Рис. 6:

```
Administrator: Windows PowerShell

> help
Commands:  (identifiers are shown in uppercase, [] means optional)
NAME       - print info about the host/domain NAME using default server
NAME1 NAME2 - as above, but use NAME2 as server
help or ?  - print info on common commands
set OPTION - set an option
all         - print options, current server and host
[no]debug  - print debugging information
[no]d2     - print exhaustive debugging information
[no]defname - append domain name to each query
[no]recurse - ask for recursive answer to query
[no]search - use domain search list
[no]vc     - always use a virtual circuit
domain=NAME - set default domain name to NAME
srchlist=N1[/N2/.../N6] - set domain to N1 and search list to N1,N2, etc.
root=NAME  - set root server to NAME
retry=X    - set number of retries to X
timeout=X  - set initial time-out interval to X seconds
type=X     - set query type (ex. A,AAAA,A+AAAA,ANY,CNAME,MX,NS,PTR,SOA,SRV)
querytype=X - same as type
class=X    - set query class (ex. IN (Internet), ANY)
[no]mxsfr  - use MS fast zone transfer
ixfrver=X  - current version to use in IXFR transfer request
server NAME - set default server to NAME, using current default server
lserver NAME - set default server to NAME, using initial server
root       - set current default server to the root
ls [opt] DOMAIN [> FILE] - list addresses in DOMAIN (optional: output to FILE)
-a         - list canonical names and aliases
-d         - list all records
-t TYPE    - list records of the given RFC record type (ex. A,CNAME,MX,NS,PTR etc.)
view FILE  - sort an 'ls' output file and view it with pg
exit      - exit the program

> _
```

Рис. 7:

Висновок

Я навчився користуватись програмою nslookup і вивчив основні поняття про DNS сервери.