

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

**Інститут КНІТ
Кафедра ПЗ**

ЗВІТ

До лабораторної роботи № 4

З дисципліни: “Організація комп’ютерних мереж”

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

Лектор:

викл. каф. ПЗ
Задорожний І. М.

Виконав:

ст. гр. ПЗ-22
Чаус Олег

Прийняв:

викл. каф. ПЗ
Задорожний І. М.

« ____ » _____ 2023 р.

Σ= ____ .

Львів – 2023

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

В якому випадку DNS-сервер буде реалізовувати рекурсію для відповіді на запит?

DNS-сервер реалізує рекурсію у випадку, коли клієнт запросив рекурсивний запит, а також сервер підтримує цей тип запитів. Коли сервер не може негайно знайти відповідь на запит в своєму кеші (локальному сховищі), він буде шукати відповідь на запит, виконуючи запит до іншого DNS-сервера, який містить необхідну інформацію, та продовжує цей процес рекурсивно, поки не знайде необхідну інформацію.

Яким є призначення протоколу DHCP?

Протокол DHCP (Dynamic Host Configuration Protocol – протокол динамічного налаштування вузла) – мережевий протокол для динамічного призначення комп'ютерам IP-адрес і інших параметрів для роботи в мережі TCP/IP. До таких параметрів належать IP-адреса маршрутизатора за замовчанням, маска підмережі, адреси серверів DNS, ім'я домену DNS.

Опишіть субполя поля «Прапорці» DNS-повідомлення.

Поле «прапорці» (біти з 16 по 31) мають субполя:

QR (Query Type) – тип повідомлення: 0 – запит, 1 – відповідь;

opcode (Operation Code) – код операції;

AA (Authoritative Answer) – авторитетна відповідь

TC (Truncated) – «обрізано»; якщо TC=1, то це означає, що повний розмір відповіді перевищив деякий граничний розмір;

RD (Recursion Desired) – клієнт запитує рекурсію;

RA (Recursion Available) – сервер може забезпечити рекурсію;

rcode (4 біти) – містить коди помилок. Типові значення цього субполя: 0 – помилка немає і 3 – помилка імені.

ЗАВДАННЯ

1. Ознайомтеся з теоретичними відомостями.
2. Запустіть аналізатор протоколів і застосуйте фільтр dns. У ньому потрібно буде відстежувати всі зміни, спричинені командами, заданими з командного рядка.
3. Випробуйте інтерактивний режим роботи утиліти nslookup. Для цього виконайте наступні дії:
 1. Дізнайтеся поточні активні параметри (команда set all). Зафіксуйте одержані дані, щоб їх можна було відновити після всіх змін. Окрім «свого» DNS-сервера (налаштованого за замовчанням), знайдіть ще декілька DNS-серверів, один з них – кореневий.
 2. Запитайте у DNS-сервера, заданого за замовчанням, інформацію про доменні імена google.com і wikipedia.org. Запитайте цю ж інформацію в інших DNS-серверів. Порівняйте відповіді, знайдіть та проаналізуйте відмінності у відповідях. Висновки подайте у звіті.
 3. Ввімкніть поглиблений режим відлагодження. Повторіть запит якогось із доменних імен в одного з DNS-серверів і подивіться відповідь тепер. Перейдіть у звичайний режим відлагодження, повторіть той самий запит інформації про доменне ім'я і отримайте відповідь, далі вимкніть режим відлагодження взагалі і знову повторіть запит. Знайдіть і опишіть відмінності у відповідях на запит при різних режимах
 4. Підмініть кореневий DNS-сервер, заданий за замовчанням, якимось іншим корневим сервером імен. Пересвідчіться у тому, що зміни набули чинності (команда set all). Скасуйте зміни.
 5. При ввімкненому режимі відлагодження запитайте інформацію про неіснуюче доменне ім'я. Ним може бути довільне слово, що складається більше ніж з 3 літер, але тільки не родове доменне ім'я. Детально ознайомтеся з відповіддю. Знайдіть код помилки у DNS-відповіді.
 6. Виконайте рекурсивний і нерекурсивний запити про доменне ім'я gmail.com (команди set recurse і set norecurse). Дослідіть відмінності у відповідях і поясніть їх.
 7. Детально вивчивши усі відповіді, отримані раніше, дізнайтеся, які сервери імен підтримують рекурсивні запити, а які – ні.
4. Самостійно дізнайтеся інформацію про решту ключових слів, що застосовуються з командою set і подайте її у звіті.
5. Знайдіть детальну інформацію про відмінності між абсолютними та відносними доменними іменами..
6. Дослідіть автономний (звичайний) режим роботи утиліти nslookup. Випробуйте усі можливі параметри (результати подайте у звіті).

7. Самостійно знайдіть відповідь на одне з наступних запитань (варіант запитання відповідає номеру студента в журналі) та представте цю відповідь у звіті:

Опишіть цілі, формат і приклад застосування команди `lserver` утиліти `nslookup`.

8. Сформууйте звіт зі структурою, аналогічною до звітів з попередніх лабораторних робіт. У теоретичних відомостях слід дати відповіді на 3 вибрані викладачем запитання з числа контрольних запитань. У висновку слід подати результати осмислення одержаних результатів.

ХІД ВИКОНАННЯ

Дізнався поточні активні параметри, використавши команду set all.

```
C:\Users\0leh>nslookup
Default Server:  router.lan
Address:  192.168.88.1

> set all
Default Server:  router.lan
Address:  192.168.88.1

Set options:
nodebug
defname
search
recurse
nod2
novc
noignoretc
port=53
type=A+AAAA
class=IN
timeout=2
retry=1
root=A.ROOT-SERVERS.NET.
domain=
MSxfr
IXFRversion=1
srchlist=
```

У Wireshark можна побачити перехоплений пакет запиту.

```
▼ Domain Name System (query)
  Transaction ID: 0x0001
  > Flags: 0x0100 Standard query
  Questions: 1
  Answer RRs: 0
  Authority RRs: 0
  Additional RRs: 0
  > Queries
    [Response In: 142]
```

Запитав у DNS-сервера, заданого за замовчанням, інформацію про доменні імена google.com і wikipedia.org. Запитав цю ж інформацію в інших DNS-серверів.

```

> google.com
Server: router.lan
Address: 192.168.88.1

Non-authoritative answer:
Name: google.com
Addresses: 2a00:1450:401b:810::200e
          142.250.186.206

> wikipedia.com
Server: router.lan
Address: 192.168.88.1

Non-authoritative answer:
Name: wikipedia.com
Addresses: 2620:0:862:ed1a::3
          91.198.174.194

```

```

> Frame 85276: 70 bytes on wire (560 bits), 70 bytes captured (560 bit
> Ethernet II, Src: Micro-St_19:ab:cb (04:7c:16:19:ab:cb), Dst: Router
> Internet Protocol Version 4, Src: 192.168.88.20, Dst: 192.168.88.1
> User Datagram Protocol, Src Port: 50141, Dst Port: 53
v Domain Name System (query)
  Transaction ID: 0x0002
  > Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  v Queries
    v google.com: type A, class IN
      Name: google.com
      [Name Length: 10]
      [Label Count: 2]
      Type: A (Host Address) (1)
      Class: IN (0x0001)
      \[Response In: 85277\]

```

▼ Domain Name System (response)

- Transaction ID: 0x0002
- > Flags: 0x8180 Standard query response, No error
- Questions: 1
- Answer RRs: 1
- Authority RRs: 0
- Additional RRs: 0
- ▼ Queries
 - ▼ google.com: type A, class IN
 - Name: google.com
 - [Name Length: 10]
 - [Label Count: 2]
 - Type: A (Host Address) (1)
 - Class: IN (0x0001)
- ▼ Answers
 - ▼ google.com: type A, class IN, addr 142.250.186.206
 - Name: google.com
 - Type: A (Host Address) (1)
 - Class: IN (0x0001)
 - Time to live: 282 (4 minutes, 42 seconds)
 - Data length: 4
 - Address: 142.250.186.206

[\[Request In: 85276\]](#)

```

> Ethernet II, Src: Micro-St_19:ab:cb (04:7c:16:19:ab:cb), Dst: Routerbo_4
> Internet Protocol Version 4, Src: 192.168.88.20, Dst: 192.168.88.1
> User Datagram Protocol, Src Port: 50143, Dst Port: 53
▼ Domain Name System (query)
  Transaction ID: 0x0004
  > Flags: 0x0100 Standard query
  Questions: 1
  Answer RRs: 0
  Authority RRs: 0
  Additional RRs: 0
  ▼ Queries
    ▼ wikipedia.com: type A, class IN
      Name: wikipedia.com
      [Name Length: 13]
      [Label Count: 2]
      Type: A (Host Address) (1)
      Class: IN (0x0001)
    [Response In: 85351]

▼ Domain Name System (response)
  Transaction ID: 0x0004
  > Flags: 0x8180 Standard query response, No error
  Questions: 1
  Answer RRs: 1
  Authority RRs: 0
  Additional RRs: 0
  ▼ Queries
    ▼ wikipedia.com: type A, class IN
      Name: wikipedia.com
      [Name Length: 13]
      [Label Count: 2]
      Type: A (Host Address) (1)
      Class: IN (0x0001)
  ▼ Answers
    ▼ wikipedia.com: type A, class IN, addr 91.198.174.194
      Name: wikipedia.com
      Type: A (Host Address) (1)
      Class: IN (0x0001)
      Time to live: 600 (10 minutes)
      Data length: 4
      Address: 91.198.174.194
    [Request In: 85348]
    [Time: 0.140788000 seconds]

```

Також здійснив запити на сервер 8.8.8.8.


```

C:\Windows\system32\cmd.exe - nslookup

> google.com 8.8.8.8
Server: [8.8.8.8]
Address: 8.8.8.8

Non-authoritative answer:
Name: google.com
Addresses: 2a00:1450:401b:814::200e
           192.178.25.174

> wikipedia.com 8.8.8.8
Server: [8.8.8.8]
Address: 8.8.8.8

Non-authoritative answer:
Name: wikipedia.com
Addresses: 2620:0:862:ed1a::3
           91.198.174.194

```

- > Ethernet II, Src: Micro-St_19:ab:cb (04:7c:16:19:ab:cb), Dst: Rout
 - > Internet Protocol Version 4, Src: 192.168.88.20, Dst: 8.8.8.8
 - > User Datagram Protocol, Src Port: 52253, Dst Port: 53
 - ▼ Domain Name System (query)
 - Transaction ID: 0x0006
 - > Flags: 0x0100 Standard query
 - Questions: 1
 - Answer RRs: 0
 - Authority RRs: 0
 - Additional RRs: 0
 - ▼ Queries
 - ▼ google.com: type A, class IN
 - Name: google.com
 - [Name Length: 10]
 - [Label Count: 2]
 - Type: A (Host Address) (1)
 - Class: IN (0x0001)
- [\[Response In: 97187\]](#)

```

> Internet Protocol Version 4, Src: 8.8.8.8, Dst: 192.168.88.20
> User Datagram Protocol, Src Port: 53, Dst Port: 52253
▼ Domain Name System (response)
  Transaction ID: 0x0006
  > Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 0
    Additional RRs: 0
  ▼ Queries
    ▼ google.com: type A, class IN
      Name: google.com
      [Name Length: 10]
      [Label Count: 2]
      Type: A (Host Address) (1)
      Class: IN (0x0001)
  ▼ Answers
    ▼ google.com: type A, class IN, addr 192.178.25.174
      Name: google.com
      Type: A (Host Address) (1)
      Class: IN (0x0001)
      Time to live: 137 (2 minutes, 17 seconds)
      Data length: 4
      Address: 192.178.25.174
      [Request In: 97186]
      [Time: 0.024140000 seconds]

```

Увімкнув поглиблений режим відлагодження та повторив попередній запит. На зображеннях можна побачити детальний опис кожного надісланого та отриманого пакетів.

```

> set d2
> wikipedia.com 8.8.8.8
Server: [8.8.8.8]
Address: 8.8.8.8

-----
SendRequest(), len 31
  HEADER:
    opcode = QUERY, id = 10, rcode = NOERROR
    header flags: query, want recursion
    questions = 1, answers = 0, authority records = 0, additional = 0

  QUESTIONS:
    wikipedia.com, type = A, class = IN

-----
Got answer (47 bytes):
  HEADER:
    opcode = QUERY, id = 10, rcode = NOERROR
    header flags: response, want recursion, recursion avail.
    questions = 1, answers = 1, authority records = 0, additional = 0

  QUESTIONS:
    wikipedia.com, type = A, class = IN
  ANSWERS:
    -> wikipedia.com
        type = A, class = IN, dlen = 4
        internet address = 91.198.174.194
        ttl = 269 (4 mins 29 secs)

-----
Non-authoritative answer:
-----
SendRequest(), len 31
  HEADER:
    opcode = QUERY, id = 11, rcode = NOERROR
    header flags: query, want recursion
    questions = 1, answers = 0, authority records = 0, additional = 0

  QUESTIONS:
    wikipedia.com, type = AAAA, class = IN

-----

```

```
-----  
Got answer (59 bytes):  
  HEADER:  
    opcode = QUERY, id = 11, rcode = NOERROR  
    header flags:  response, want recursion, recursion avail.  
    questions = 1,  answers = 1,  authority records = 0,  additional = 0  
  
  QUESTIONS:  
    wikipedia.com, type = AAAA, class = IN  
  ANSWERS:  
-> wikipedia.com  
   type = AAAA, class = IN, dlen = 16  
   AAAA IPv6 address = 2620:0:862:ed1a::3  
   ttl = 600 (10 mins)  
  
-----  
Name:      wikipedia.com  
Addresses: 2620:0:862:ed1a::3  
           91.198.174.194
```

Перейшов у звичайний режим відлагодження та повторив запит. Тепер можна побачити інформацію лише про отримані пакети.

```

> set debug
> wikipedia.com 8.8.8.8
Server: [8.8.8.8]
Address: 8.8.8.8

-----
Got answer:
  HEADER:
    opcode = QUERY, id = 12, rcode = NOERROR
    header flags: response, want recursion, recursion avail.
    questions = 1, answers = 1, authority records = 0, additional = 0

  QUESTIONS:
    wikipedia.com, type = A, class = IN
  ANSWERS:
    -> wikipedia.com
        internet address = 91.198.174.194
        ttl = 600 (10 mins)

-----
Non-authoritative answer:
-----
Got answer:
  HEADER:
    opcode = QUERY, id = 13, rcode = NOERROR
    header flags: response, want recursion, recursion avail.
    questions = 1, answers = 1, authority records = 0, additional = 0

  QUESTIONS:
    wikipedia.com, type = AAAA, class = IN
  ANSWERS:
    -> wikipedia.com
        AAAA IPv6 address = 2620:0:862:ed1a::3
        ttl = 600 (10 mins)

-----
Name:    wikipedia.com
Addresses: 2620:0:862:ed1a::3
          91.198.174.194

```

Вимкнув режим відлагодження. Тепер інформація про отримані чи надіслані пакети не відображається.

```

> set nodebug
> wikipedia.com 8.8.8.8
Server: [8.8.8.8]
Address: 8.8.8.8

Non-authoritative answer:
Name:    wikipedia.com
Addresses: 2620:0:862:ed1a::3
          91.198.174.194

```

Змінив кореневий сервер, використавши команду set root=.

```
> set root=ABCD
> set all
Default Server:  router.lan
Address:  192.168.88.1

host =  wikipedia.com
Addresses: 2620:0:862:ed1a::3
           91.198.174.194

Set options:
nodebug
defname
search
recurse
nod2
novc
noignoretc
port=53
type=A+AAAA
class=IN
timeout=2
retry=1
root=ABCD
domain=
MSxfr
IXFRversion=1
srchlist=
```

При увімкненому режимі відлагодження надіслав запит для неіснуючого доменного імені.

```

> set d2
> TESTNAME
Server:  router.lan
Address:  192.168.88.1

-----
SendRequest(), len 26
  HEADER:
    opcode = QUERY, id = 18, rcode = NOERROR
    header flags:  query, want recursion
    questions = 1,  answers = 0,  authority records = 0,  additional = 0

  QUESTIONS:
    TESTNAME, type = A, class = IN

-----
Got answer (26 bytes):
  HEADER:
    opcode = QUERY, id = 18, rcode = NXDOMAIN
    header flags:  response, want recursion, recursion avail.
    questions = 1,  answers = 0,  authority records = 0,  additional = 0

  QUESTIONS:
    testname, type = A, class = IN

-----
SendRequest(), len 26
  HEADER:
    opcode = QUERY, id = 19, rcode = NOERROR
    header flags:  query, want recursion
    questions = 1,  answers = 0,  authority records = 0,  additional = 0

  QUESTIONS:
    TESTNAME, type = AAAA, class = IN

-----
Got answer (26 bytes):
  HEADER:
    opcode = QUERY, id = 19, rcode = NXDOMAIN
    header flags:  response, want recursion, recursion avail.
    questions = 1,  answers = 0,  authority records = 0,  additional = 0

  QUESTIONS:
    testname, type = AAAA, class = IN

-----
*** router.lan can't find TESTNAME: Non-existent domain

```

У пакетах-відповідях можна побачити, що поле rcode має значення NXDOMAIN (неіснуючий домен).

Увімкнув режим рекурсії та надіслав запит для домену gmail.com

```

> set recurse
> gmail.com
Server: router.lan
Address: 192.168.88.1

-----
SendRequest(), len 27
  HEADER:
    opcode = QUERY, id = 20, rcode = NOERROR
    header flags: query, want recursion
    questions = 1, answers = 0, authority records = 0, additional = 0

  QUESTIONS:
    gmail.com, type = A, class = IN

-----
Got answer (43 bytes):
  HEADER:
    opcode = QUERY, id = 20, rcode = NOERROR
    header flags: response, want recursion, recursion avail.
    questions = 1, answers = 1, authority records = 0, additional = 0

  QUESTIONS:
    gmail.com, type = A, class = IN
  ANSWERS:
    -> gmail.com
        type = A, class = IN, dlen = 4
        internet address = 216.58.208.197
        ttl = 12 (12 secs)

```

У полі header flags пакету запиту можна побачити прапорець want recursion.

У полі header flags пакету відповіді можна також побачити прапорець recursion avail.

Надіслав аналогічний запит не в рекурсивному режимі.


```

> set norecurse
> gmail.com
Server: router.lan
Address: 192.168.88.1

-----
SendRequest(), len 27
HEADER:
  opcode = QUERY, id = 22, rcode = NOERROR
  header flags: query
  questions = 1, answers = 0, authority records = 0, additional = 0

QUESTIONS:
  gmail.com, type = A, class = IN

-----

Got answer (43 bytes):
HEADER:
  opcode = QUERY, id = 22, rcode = NOERROR
  header flags: response, want recursion, recursion avail.
  questions = 1, answers = 1, authority records = 0, additional = 0

QUESTIONS:
  gmail.com, type = A, class = IN
ANSWERS:
-> gmail.com
   type = A, class = IN, dlen = 4
   internet address = 142.250.186.197
   ttl = 280 (4 mins 40 secs)

```

Серед прапорців запиту немає want recursion.

| | |
|--|---|
| <pre> > Internet Protocol Version 4, Src: 192.168.88.20, Dst: 192.168.88.1 > User Datagram Protocol, Src Port: 61234, Dst Port: 53 Domain Name System (query) Transaction ID: 0x0014 Flags: 0x0100 Standard query 0... .. = Response: Message is a query .000 0... .. = Opcode: Standard query (0) = Truncated: Message is not truncated 1... .. = Recursion desired: Do query recursively = Z: reserved (0) = Non-authenticated data: Unacceptable Questions: 1 Answer RRs: 0 Authority RRs: 0 Additional RRs: 0 Queries gmail.com: type A, class IN Name: gmail.com [Name Length: 9] [Label Count: 2] Type: A (Host Address) (1) Class: IN (0x0001) [Response In: 205441] </pre> | <pre> Domain Name System (query) Transaction ID: 0x0016 Flags: 0x0000 Standard query 0... .. = Response: Message is a query .000 0... .. = Opcode: Standard query (0) = Truncated: Message is not truncated = Recursion desired: Don't do query recur = Z: reserved (0) = Non-authenticated data: Unacceptable Questions: 1 Answer RRs: 0 Authority RRs: 0 Additional RRs: 0 Queries gmail.com: type A, class IN Name: gmail.com [Name Length: 9] [Label Count: 2] Type: A (Host Address) (1) Class: IN (0x0001) [Response In: 205777] </pre> |
|--|---|

Сервери, які підтримують рекурсивні запити, називаються рекурсивними перетворювачами (resolver). Вони є першою зупинкою в ланцюжку DNS-запитів і зв'язуються з іншими DNS-серверами, щоб знайти IP-адресу для заданого доменного імені. Вони також кешують інформацію, яку отримують від інших серверів, щоб пришвидшити майбутні запити.

Сервери, які не підтримують рекурсивні запити, називаються кореневими серверами імен, серверами імен верхнього рівня і авторитетними серверами імен. Вони не виконують рекурсію, а лише видають посилання на інші DNS-сервери. Вони відповідають лише на запити, які запитують їхнє доменне ім'я або розширення, і не кешують жодної інформації з інших серверів.

Дізнався про всі параметри команди set.

```
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]msxfr      - use MS fast zone transfer
ixfrver=X      - current version to use in IXFR transfer request
```

Дізнався про різницю між абсолютними та відносними доменними іменами.

Абсолютні доменні імена:

Технічно, якщо домен верхнього рівня "А" містить субдомен "В", який в свою чергу містить субдомен "С", повне доменне ім'я для "С" буде "С.В.А.". Доменне ім'я "С.В.А." є повністю визначеним, оскільки воно вказує повне місцезнаходження конкретного домену, який носить його ім'я, у всьому просторі імен DNS.

Відносні доменні імена:

Існують також ситуації, в яких ми можемо посилатися на пристрій, використовуючи неповну специфікацію імені. Відносне доменне ім'я можна використовувати лише в контексті певного батьківського домену, абсолютне доменне ім'я якого відоме. Потім ми можемо знайти абсолютне доменне ім'я частково вказаного доменного імені, додавши часткове ім'я до абсолютного імені батьківського домену. Якщо ми маємо PQDN "Z" в контексті FQDN "Y.X.", ми знаємо, що FQDN для "Z" - це "Z.Y.X.".

Адміністратор домену може використовувати відносні імена як скорочення для позначення пристроїв або субдоменів без необхідності повторювати повне ім'я.

Дослідив автономний режим роботи утиліти nslookup.

```

C:\Users\Oleh>nslookup /help
Usage:
C:\Users\Oleh>nslookup [-opt ...]          # interactive mode using default server
nslookup [-opt ...] - server                # interactive mode using 'server'
nslookup [-opt ...] host                    # just look up 'host' using default server
nslookup [-opt ...] host server            # just look up 'host' using 'server'

C:\Users\Oleh>nslookup youtube.com
Server:  router.lan
Address:  192.168.88.1

Non-authoritative answer:
Name:     youtube.com
Addresses: 2a00:1450:401b:810::200e
          192.178.25.174

```

Дав відповідь на запитання.

Опишіть цілі, формат і приклад застосування команди *lserver* утиліти *nslookup*. Команда *lserver* змінює початковий (initial) DNS-сервер, на відміну від default сервера, якого можна змінити командою *server*. Вона може використовуватись у випадках, коли сервер за замовчуванням (який відрізняється від початкового сервера) не розпізнає назви сервера, якого користувач намагається використати. В такому випадку можна використати команду *lserver*, яка надішле запит на сервер, який був на початку сесії. Цю команду можна використати так:

```

> lserver 1.1.1.1
Default Server: [1.1.1.1]
Address: 1.1.1.1

```

ВИСНОВКИ

Вивчив 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.