Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Системы обработки информации и управления

Методические указания к лабораторной работе № 8 по дисциплине Сети и телекоммуникации

Для студентов 3-го курса кафедры ИУ5

Разработали:
Старший преподаватель
Антонов А.И.
Старший преподаватель
Лосева С.С.

Москва, 2025 г.

Оглавление

Лабораторная работа 8. Ознакомление с системой и протоколом DNS

3.1 Цель работы

Лабораторная работа ставит цели закрепления теоретического материала по протоколам и программному обеспечению системы доменных имен.

3.2 Задачи

Ознакомиться с работой утилит host, nslookup, dig. Научиться делать обратный DNSзапрос,

3.3 Теоретический материал

DNS (англ. Domain Name System «система доменных имён») — компьютерная распределённая система для получения информации о доменах. Чаще всего используется для получения IP-адреса по имени хоста (компьютера или устройства), получения информации о маршрутизации почты и/или обслуживающих узлах для протоколов в домене (SRV-запись).

Доменная структура DNS представляет собой древовидную иерархию (Рисунок 1), состоящую из узлов, зон, доменов, поддоменов и др. элементов, о которых ниже пойдет речь. «Вершиной» доменной структуры является корневая зона. Настройки корневой зоны расположены на множестве серверов/зеркал, размещенных по всему миру и содержат информацию о всех серверах корневой зоны, а так же отвечающих за **домены первого уровня** (ru, net, org и др). Информация о серверах корневой зоны расположена на данном сайте корневых серверов. Настройки корневой зоны всегда доступны тут. Серверы корневой зоны обрабатывают и отвечают на запросы, выдавая информацию только о доменах первого уровня (то есть отвечают на любые запросы, как на нерекурсивные).

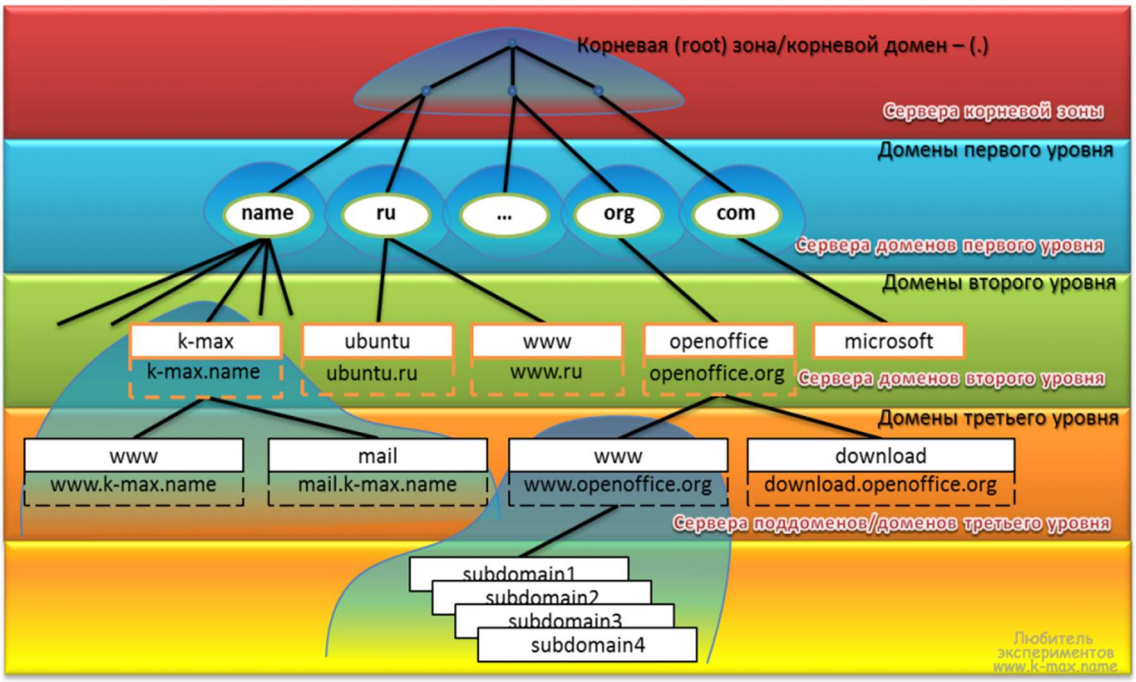

Рисунок 1 – Иерархическая структура DNS

Начиная с 2010 года в систему DNS внедряются средства проверки целостности передаваемых данных, называемые DNS Security Extensions (DNSSEC). Передаваемые данные не шифруются, но их достоверность проверяется криптографическими способами. Внедряемый стандарт DANE обеспечивает передачу средствами DNS достоверной криптографической информации (сертификатов), используемых для установления безопасных и защищённых соединений транспортного и прикладного уровней.

DNS обладает следующими характеристиками:

– Распределённость администрирования. Ответственность за разные части иерархической структуры несут разные люди или организации.

– Распределённость хранения информации. Каждый узел сети в обязательном порядке должен хранить только те данные, которые входят в его зону ответственности, и (возможно) адреса корневых DNS-серверов.

– Кэширование информации. Узел может хранить некоторое количество данных не из своей зоны ответственности для уменьшения нагрузки на сеть.

– Иерархическая структура, в которой все узлы объединены в дерево, и каждый узел может или самостоятельно определять работу нижестоящих узлов, или делегировать (передавать) их другим узлам.

– Резервирование. За хранение и обслуживание своих узлов (зон) отвечают (обычно) несколько серверов, разделённые как физически, так и логически, что обеспечивает сохранность данных и продолжение работы даже в случае сбоя одного из узлов.

Ключевыми понятиями DNS являются:

Зона — это любая часть дерева системы доменных имен, размещаемая как единое целое на некотором **DNS-сервере**. Зону, для бОльшего понимания, можно назвать *«зоной ответственности». Целью выделения части дерева в отдельную зону* является передача ответственности (**Делегирование**) за эту ветвь другому лицу или организации. На иллюстрации, примеры зон выделены синим градиентом (зона *name*., зона *k-max.name*. со всем подчиненными ресурсами, www.openoffice.org со всем подчиненными поддоменами и ресурсами). На иллюстрации выделены не все зоны, а лишь некоторые для общего понимания и представления. В каждой зоне имеется, по крайней мере, один **авторитетный сервер DNS**, который хранит ВСЮ информацию о зоне, за которую он отвечает.

Домен — это именованная ветвь или поддерево в дереве имен DNS, то есть это определенный узел, включающий в себя все подчиненные узлы.

Каждый узел в иерархии DNS отделен от своего родителя точкой. Если провести аналогию с файловой системой Linux, система доменных имен имеет похожую структуру, за тем исключением, что разделитель в файловой системе — *слэш*, а в DNS — *точка*. А так же **DNS адрес** читается справа налево (от корневого домена к имени хоста) в отличии от пути в файловой системе Linux. **Доменное имя начинается с точки** (*корневого домена*) и проходит через домены первого, второго и если нужно третьего и т.д. уровней и завершается именем хоста. Т.о. **доменное имя** полностью отражает *структуру иерархии DNS*. Часто (я бы сказал — всегда в повседневной жизни), последняя точка

(обозначение корневого домена) в доменном имени опускается (то есть в браузере мы вводим не k-max.nam**e.**, а k-max.nam**e**). Итак, разобрав структуру доменного имени, мы незаметно подошли к понятию **FQDN**.

FQDN (англ. *Fully Qualifed Domain Name*, полностью определённое имя домена) — это имя домена, **однозначно** определяющее доменное имя и включающее в себя имена всех родительских доменов иерархии DNS, **в том числе и корневого**. Своеобразный аналог абсолютного пути в файловой системе. Давайте разберем вышесказанное на примере имени домена mail.k-max.name:

```
mail.k-max.name.
|    | | ||
|    | | |+-корневой домен
|    | |  +---домен первого уровня
|    |  +------точка, разделяющая домены/части FQDN
|    +---------домен второго уровня
 +---------------поддомен/домен третьего уровня, возможно - имя хоста
```

Различие между **FQDN** и обычным доменным (**неFQDN**) именем появляется при именовании доменов второго, третьего (и т. д.) уровня. Для **получения FQDN** требуется *обязательно* указать в доменном имени домены более высокого уровня (например, mail является доменным именем, однако FQDN имя выглядит как mail.k-max.name.). Максимальный размер FQDN — 255 байт, с ограничением в 63 байта на каждое имя домена.

Поддомены это — *подчиненные домены*. По большому счету, все домены в интернете являются подчиненными за исключением корневого. Например домен k-max является поддоменом домена name, а name, в свою очередь — поддоменом корневого домена.

Итак, на схеме выше мы рассмотрели **корневой домен**, следующим в иерархии идут **домены первого/верхнего уровня**, они же **TLD**, они же **Top-Level Domain**. К данным доменам относятся ***национальные домены*** (*ru.*, *ua.* и др) и ***общие домены*** (*com.*, *net.*, и др). Существуют так же

специализированные домены, которые не опубликованы в системе DNS, но используются программами (домен .onion используется анонимной сетью Tor для перехвата и последующей маршрутизации обращений к скрытым сервисам этой сети). Еще есть т.н. **зарезервированные доменные имена,** определенные в RFC 2606 (**Reserved Top Level DNS Names** — Зарезервированные имена доменов верхнего уровня) определяет названия доменов, которые следует использовать в качестве примеров (например, в документации), а также для тестирования. К таким именам относятся например example.com, example.org и example.net, а также test, invalid и др. Ниже по иерархии, как видно, идут домены третьего уровня и т.д. Заканчивается доменная иерархия — **именами хостов**, которые задаются соответствующими **ресурсными записями** или **хостовыми записями**.

Ресурсная запись — это то, собственно, ради чего в конечном счете и существует DNS. **Ресурсная запись** — это единица хранения и передачи информации в DNS. Каждая такая запись несет в себе ***информацию соответствия*** какого-то *имени* и *служебной информации* в DNS, например соответствие имени домена — IP адреса.

Запись ресурса состоит из следующих полей:

- **имя (NAME)** — доменное имя, к которому привязана или которому «принадлежит» данная ресурсная запись, либо IP адрес. При отсутствии данного поля, запись ресурса наследуется от предыдущей записи.
- **Time To Live (TTL)** — дословно «время жизни» записи, время хранения записи в кэше DNS (после указанного времени запись удаляется), данное поле может не указываться в индивидуальных записях ресурсов, но тогда оно должно быть указано в начале файла зоны и будет наследоваться всеми записями.
- **класс (CLASS)** — определяет тип сети, (в 99,99% случаях используется IN (что обозначает — Internet). Данное поле было создано из

предположения, что DNS может работать и в других типах сетей, кроме TCP/IP)

– **тип (TYPE)** — тип записи синтаксис и назначение записи

– **данные (DATA)** — различная информация, формат и синтаксис которой определяется типом.

При этом, возможно использовать следующие символы:

– ; - Вводит комментарий

– # - Также вводит комментарии (только в версии BIND 4.9)

– @ — Имя текущего домена

– ( ) — Позволяют данным занимать несколько строк

– * — Метасимвол (только в поле имя)

Со всем набором ресурсных записей можно ознакомиться, например, здесь https://ru.ruwiki.ru/wiki/Типы_ресурсных_записей_DNS. Наиболее часто применяемые **ресурсные записи следующими A** — (*address record/запись адреса*) отображают имя хоста (доменное имя) на адрес IPv4. Для каждого сетевого интерфейса машины должна быть сделана одна **A-запись**. Например, следующая запись отображает *доменное имя k-max.name.* в *IPv4 адрес хоста 81.177.139.65* (поле **NAME** — *k-max.name.*, поле **TTL** — *86400*, поле **CLASS** — *IN*, поле **DATA** — *81.177.139.65*):

k-max.name. 86400 IN A 81.177.139.65

– **AAAA** (*IPv6 address record*) аналогична записи A, но для IPv6.

– **CNAME** (*canonical name record/каноническая запись имени (псевдоним)*) — отображает алиас на реальное имя (для перенаправления на другое имя), например, следующая запись задает алиас ftp для хоста www.k-max.name.:

ftp 86400 IN CNAME www.k-max.name.

– **MX** (*mail exchange*) — указывает хосты для доставки почты, адресованной домену. При этом поле NAME указывает домен назначения, ***поля TTL***, ***CLASS*** — стандартное значение, ***поле TYPE*** принимает значение *MX*, а ***поле DATA*** указывает *приоритет* и через пробел - *доменное имя хоста,*

ответственного за прием почты. Например, следующая запись показывает, что для домена k-max.name направлять почту сначала на mx.k-max.name, затем на mx2.k-max.name, если с mx.k-max.name возникли какие-то проблемы. При этом, для обоих MX хостов должны быть соответствующие A-записи:

k-max.name. 17790 IN MX 10 mx.k-max.name.

k-max.name. 17790 IN MX 20 mx2.k-max.name.

– **NS** (*name server/сервер имён*) указывает на DNS-сервер, обслуживающий данный домен. Вернее будет сказать — указывают сервера, на которые делегирован данный домен. Если записи NS относятся к серверам имен для текущей зоны, доменная система имен их практически не использует. Они просто поясняют, как организована зона и какие машины играют ключевую роль в обеспечении сервиса имен. Например, зону name. обслуживают следующие NS:

name. 5772 IN NS l6.nstld.com.

name. 5772 IN NS m6.nstld.com.

name. 5772 IN NS c6.nstld.com.

name. 5772 IN NS j6.nstld.com.

.....

зону k-max.name обслуживают:

1. k-max.name. 1577 IN NS ns2.jino.ru.
2. k-max.name. 1577 IN NS ns1.jino.ru.

– **PTR** (*pointer*) — отображает IP-адрес в доменное имя (о данном типе записи поговорим ниже в разделе обратного преобразования имен).

– **SOA** (*Start of Authority/начальная запись зоны*) — описывает основные/начальные настройки зоны, можно сказать, *определяет зону ответственности данного сервера*. Для каждой зоны должна существовать только одна запись SOA и она должна быть первая. ***Поле Name*** содержит *имя домена/зоны*, ***поля TTL, CLASS*** — стандартное значение, ***поле TYPE*** принимает значение *SOA*, а ***поле DATA*** состоит из нескольких значений, разделенных пробелами: *имя главного DNS (Primary Name Server)*, *адрес*

администратора зоны, далее в скобках — *серийный номер файла зоны (Serial number)*. При каждом внесении изменений в файл зоны данное значение необходимо увеличивать, это указывает вторичным серверам, что зона изменена, и что им необходимо обновить у себя зону. Далее — ***значения таймеров*** (*Refresh* — указывает, как часто вторичные серверы должны опрашивать первичный, чтобы узнать, не увеличился ли серийный номер зоны, *Retry* — время ожидания после неудачной попытки опроса, *Expire* — максимальное время, в течение которого вторичный сервер может использовать информацию о полученной зоне, *Minimum TTL* — минимальное время, в течение которого данные остаются в кэше вторичного сервера). Ниже в примере приведено 2 одинаковые записи SOA (хотя вторая и записана в несколько строк), но они одинаковы по значению и формат записи второй более понятен в силу его структурированности:

```
k-max.name.        86400   IN   SOA   ns1.jino.ru.        hostmaster.jino.ru.
2011032003 28800 7200 604800 86400

k-max.name.        86400   IN   SOA   ns1.jino.ru. hostmaster.jino.ru. (
                                      2011032003 ; serial (серийный номер)
                                      28800 ; refresh (обновление)
                                      7200 ; retry (повторная попытка)
                                      604800 ; expire (срок годности)
                                      86400) ; minimum TTL (минимум)
```

– **SRV** (*server selection*) — указывают на сервера, обеспечивающие работу тех или иных служб в данном домене (например Jabber и Active Directory).

Делегирование (корректнее сказать **делегирование ответственности**) — это операция *передачи ответственности за часть дерева доменных имен (зону)* другому лицу или организации. За счет делегирования, в DNS обеспечивается распределенность администрирования и хранения зон. Технически, делегирование заключается в выделении какой-либо части дерева в отдельную зону, и размещении этой зоны на DNS-сервере, принадлежащем

другому лицу или организации. При этом, в родительскую зону включаются «склеивающие» ресурсные записи (NS и A), содержащие указатели на авторитативные DNS-сервера дочерней зоны, а вся остальная информация, относящаяся к дочерней зоне, хранится уже на DNS-серверах дочерней зоны. Например, на иллюстрации **корневой домен** делегирует полномочия серверам отвечающим за TLD, TLD же в свою очередь, делегируют полномочия управления зонами — серверам второго уровня, иногда на этом цепочка заканчивается, но бывает, что делегирование простирается до 4 и даже 5 уровней.

Пример. Делегирование управления поддоменом *k-max.name* другому лицу (приводит к созданию **новой зоны**, которая администрируется независимо от остального пространства имен (независимо от вышестоящего name.). **Зона k-max.name** после делегирования полномочий теперь не зависит от name. и может содержать все (вернее сказать — любые имена, которые я захочу) доменные имена, которые заканчиваются на **.k-max.name.* С другой стороны, **зона name.** содержит только доменные имена, оканчивающиеся на *.name., но не входящие в делегированные этой зоны, такие, например, как k-max.name или a-lab.name или любая другая. **k-max.name может быть поделен на поддомены** с именами вроде *mail.k-max.name*, *ftp.k-max.name* и некоторые из этих поддоменов могут быть выделены в самостоятельные зоны, и ответственность за данные зоны может так же быть делегирована. Если ftp.k-max.name будет являться самостоятельной зоной, то зона k-max.name не будет содержать доменные записи, которые заканчиваются на *.ftp.k-max.name.

Т.о. **после делегирования ответственности,** информация, хранимая делегирующей зоной, уже не включает информацию по делегированному поддомену и его ресурсным записям хостов, а хранит информацию о **серверах имен**, являющихся для делегируемого поддомена авторитативными. Это и есть «склеивающие» записи, о чем я выше уже говорил. В таком случае, если у DNS-сервера родительского домена запрашиваются данные об адресе,

принадлежащем делегированному поддомену, в ответ предоставляется список DNS-серверов, которые обладают соответствующей информацией.

DNS-сервер – специализированное ПО для обслуживания DNS, а также компьютер, на котором это ПО выполняется. DNS-сервер может быть ответственным за некоторые зоны и/или может перенаправлять запросы вышестоящим серверам.

DNS-клиент – специализированная библиотека (или программа) для работы с DNS. В ряде случаев DNS-сервер выступает в роли DNS-клиента.

Авторитетность (англ. authoritative) – признак размещения зоны на DNS-сервере. Ответы DNS-сервера могут быть двух типов: авторитетные (когда сервер заявляет, что сам отвечает за зону) и неавторитетные (англ. Non-authoritative), когда сервер обрабатывает запрос, и возвращает ответ других серверов. В некоторых случаях вместо передачи запроса дальше DNS-сервер может вернуть уже известное ему (по запросам ранее) значение (режим кеширования).

DNS-запрос (англ. DNS query) – запрос от клиента (или сервера) серверу. Запрос может быть рекурсивным или нерекурсивным.

Имя и IP-адрес не тождественны –один IP-адрес может иметь множество имён, что позволяет поддерживать на одном компьютере множество веб-сайтов (это называется виртуальный хостинг). Обратное тоже справедливо – одному имени может быть сопоставлено множество IP-адресов: это позволяет создавать балансировку нагрузки.

Для повышения устойчивости системы используется множество серверов, содержащих идентичную информацию, а в протоколе есть средства, позволяющие поддерживать синхронность информации, расположенной на разных серверах.

Протокол DNS использует для работы TCP или UDP-порт 53 для ответов на запросы. Традиционно запросы и ответы отправляются в виде одной UDP-датаграммы. TCP используется, когда размер данных ответа превышает 512 байт.

Термином рекурсия в DNS обозначают алгоритм поведения DNS-сервера: «выполнить от имени клиента полный поиск нужной информации во всей системе DNS, при необходимости обращаясь к другим DNS-серверам».

DNS-запрос может быть рекурсивным –требующим полного поиска, –и нерекурсивным (или итеративным) –не требующим полного поиска.

Аналогично – DNS-сервер может быть рекурсивным (умеющим выполнять полный поиск) и нерекурсивным (не умеющим выполнять полный поиск).

В случае рекурсивного запроса DNS-сервер опрашивает серверы (в порядке убывания уровня зон в имени), пока не найдёт ответ или не обнаружит, что домен не существует (на практике поиск начинается с наиболее близких к искомому DNS-серверов, если информация о них есть в кэше и не устарела, сервер может не запрашивать другие DNS-серверы).

Рассмотрим на примере работу всей системы.

Предположим, мы набрали в браузере адрес ru.wikipedia.org. Браузер ищет соответствие этого адреса IP-адресу в файле hosts. Если файл не содержит соответствия, то далее браузер спрашивает у сервера DNS: «какой IP-адрес у ru.wikipedia.org»? Однако сервер DNS может ничего не знать не только о запрошенном имени, но и даже обо всём домене wikipedia.org. В этом случае сервер обращается к корневому серверу –например, 198.41.0.4. Этот сервер сообщает – «У меня нет информации о данном адресе, но я знаю, что 204.74.112.1 является ответственным за зону org.» Тогда сервер DNS направляет свой запрос к 204.74.112.1, но тот отвечает «У меня нет информации о данном сервере, но я знаю, что 207.142.131.234 является ответственным за зону wikipedia.org.» Наконец, тот же запрос отправляется к третьему DNS-серверу и получает ответ –IP-адрес, который и передаётся клиенту –браузеру.

DNS используется в первую очередь для преобразования символьных имён в IP-адреса, но он также может выполнять обратный процесс. Для этого используются уже имеющиеся средства DNS. Дело в том, что с записью DNS

могут быть сопоставлены различные данные, в том числе и какое-либо символьное имя. Существует специальный домен in-addr.arpa, записи в котором используются для преобразования IP-адресов в символьные имена. Например, для получения DNS-имени для адреса 11.22.33.44 можно запросить у DNS-сервера запись 44.33.22.11.in-addr.arpa, и тот вернёт соответствующее символьное имя. Обратный порядок записи частей IP-адреса объясняется тем, что в IP-адресах старшие биты расположены в начале, а в символьных DNS-именах старшие (находящиеся ближе к корню) части расположены в конце.

3.4 Порядок выполнения лабораторной работы

3.4.1 Разрешение адресов в системе DNS с использованием различных утилит системы Linux

1.Запустить анализатор протоколов Wireshark и указать фильтр для перехвата трафика DNS.

2.В терминале последовательно выполнить разрешение доменных имен согласно варианту из ЛР1 с использованием трех утилит:

а) host

б) nslookup

в) dig

3.Остановить захват пакетов в анализаторе протоколов Wireshark.

4.Проанализировать вывод команд и перехваченные пакеты.

5.В отчете привести последовательно вывод каждой команды, сопроводив его соответствующими перехваченными пакетами и выводами по работе программ.

3.4.2 Получение ресурсных записей различных типов с использованием утилиты *nslookup*

1.Запустить анализатор протоколов Wireshark и указать фильтр для перехвата трафика DNS.

2.Последовательно получить от сервера DNS следующие ресурсные записи для доменного имени согласно варианту:

а) адреса IPv4

б) адреса IPv6

в) почтовые серверы

г) серверы DNS

д) авторитетный сервер для доменного имени

3.Остановить захват пакетов в анализаторе протоколов Wireshark.

4.Проанализировать вывод команд и перехваченные пакеты.

В отчете привести последовательно вывод каждой команды, сопроводив его соответствующими перехваченными пакетами и выводами по работе программ.

3.4.3 Проведение обратного запроса DNS с использованием утилиты *host*

1.Запустить анализатор протоколов Wireshark и указать фильтр для перехвата трафика DNS.

2.Провести обратный запрос DNS для IPv4-адреса из предыдущего пункта.

3.Провести обратный запрос DNS для IPv6-адреса из предыдущего пункта. При формировании команды запроса руководствоваться примером ресурсной записи:

```
host -t PTR a.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.a.0.0.0.8.b.6.0.2.0.a.2.ip6.arpa
```

4.Остановить захват пакетов в анализаторе протоколов Wireshark.

5.Проанализировать вывод команд и перехваченные пакеты.

6.В отчете привести последовательно вывод каждой команды, сопроводив его соответствующими перехваченными пакетами и выводами по работе программ.

3.4.4 **Получение всех ресурсных записей для определенного доменного имени с использованием утилиты** *host*

1.Запустить анализатор протоколов Wireshark и указать фильтр для перехвата трафика: IP-адреса ПК и сервера DNS, протоколы (TCP или UDP).

2.Выполнить запрос «ресурсной записи» ANY для доменного имени согласно варианту.

3.Остановить захват пакетов в анализаторе протоколов Wireshark.

4.Проанализировать вывод команд и перехваченные пакеты.

5.В отчете привести вывод команды, сопроводив его соответствующими перехваченными пакетами и выводами по процедуре получения записи.

3.5 Содержание отчета

1.Заголовок согласно приложению.

2.Цель работы.

3.Результаты работы по каждому из пунктов работы с соответствующими выводами.

3.6 Контрольные вопросы

1.Что такое система доменных имён?

2.Для чего используется файл hosts?

3.Каковы ключевые характеристики DNS?

4.Что такое домен и поддомен?

5.Что такое корневой домен?

6.Что такое рекурсия в DNS?

7.Как выполняется DNS-запрос?

8.Что такое обратный DNS-запрос?