

Санкт-Петербургский политехнический университет Петра Великого

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №3

Курс: **Администрирование компьютерных сетей**

Тема: **Администрирование сетевых сервисов**

Студент: **Д.В. Круминьш**

Группа: **13541/3**

Преподаватель: **И.А. Малышев**



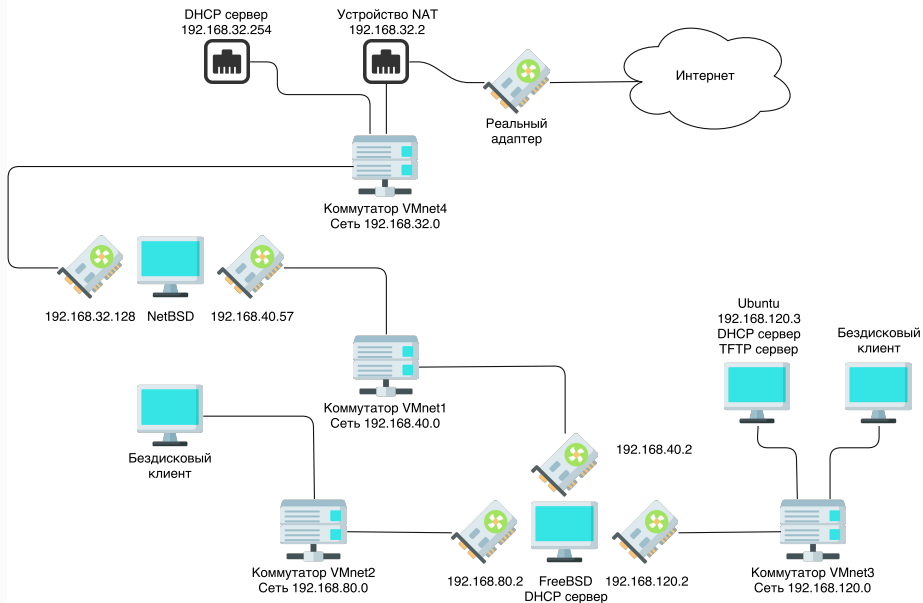
1. Изучение состава и функциональных возможностей сетевых сервисов операционных систем.
2. Разработка и настройка сервисов локальной сети.
3. Разработка и настройка сервисов демилитаризованной зоны.
4. Разработка и настройка сервисов пограничной зоны.

DNCP И УДАЛЕННАЯ ЗАГРУЗКА

Схема подверглась следующим изменениям:

1. К сети VMnet2 и VMnet3 были добавлены бездисковые клиенты;
2. В сеть VMnet3 был добавлен хост, с ОС Ubuntu;
 - С запущенными DHCP и TFTP серверами.
3. Для сети VMnet2 в настройках VMware был отключен DHCP сервер, так как его работой займется DHCP сервер на FreeBSD.

Схема ККС



1. Переходим в каталог `/usr/ports/net/isc-dhcp3-server`
2. Устанавливаем следующей командой:

```
make install clean
```

3. Вносим в файл `/etc/rc.conf` следующие строки:

```
# Включаем DHCP
dhcpd_enable="YES"
# Отключаем вывод избыточной информации
dhcpd_flags="-q"
# Указываем интерфейс для запуска
dhcpd_ifaces="em1"
```

4. Создаем файл `/usr/local/etc/dhcpd.conf` и вносим в него следующие изменения:

```
option domain-name "example.org"; # доменное имя
option domain-name-servers 192.168.32.2; #DNS сервер
default-lease-time 600;
max-lease-time 7200;

subnet 192.168.80.0 netmask 255.255.255.0 {
    range 192.168.80.127 192.168.80.224;
    option routers 192.168.80.2;
    option root-path "192.168.120.3:/usr/tftpboot/";
    next-server 192.168.120.3;
    filename "gpxelinux.0";
}
```

Для начала необходимо корректно сконфигурировать сетевой адаптер для выхода в сеть "Интернет". Для этого были заданы следующие параметры:

- Адрес - 192.168.120.3;
- Маска - 255.255.255.0;
- Шлюз - 192.168.120.2;
- DNS сервер - 192.168.32.2.

1. Выполняем команду:

```
sudo apt-get install isc-dhcp-server
```

2. В файл **/etc/dhcp/dhcpd.conf** вносим следующие изменения:

```
option domain-name "example.org";
option domain-name-servers 192.168.32.2;

default-lease-time 600;
max-lease-time 7200;

subnet 192.168.120.0 netmask 255.255.255.0 {
    range 192.168.120.100 192.168.120.200;
    option routers 192.168.120.2;
    next-server 192.168.120.3; # TFTP server address
    filename "gpxelinux.0"; # PXE boot loader filename
}
```

3. Перезапускаем DHCP-сервер

```
sudo /etc/init.d/isc-dhcp-server restart
```

1. Выполняем установку пакетов командой:

```
sudo apt-get install tftp tftpd-hpa
```

2. Создаем необходимые для работы директории:

```
mkdir -p /usr/tftpboot/images  
mkdir /usr/tftpboot/pxelinux.cfg
```

3. В файл **/etc/rc.conf** вносим изменения:

```
tftpd_enable="YES"  
tftpd_flags="-p -s /usr/tftpboot -B 1024 —ipv4"
```

4. Запускаем TFTP-сервер

```
service tftpd start
```

5. Скачиваем Syslinux версии 6.03 с сайта:

http://www.syslinux.org/wiki/index.php?title=The_Syslinux_Project

6. Извлекаем по пути **/usr/tftpboot** следующие файлы:

- chain.c32
- gpxlinux.0
- ldlinux.c32
- libutil.c32
- memdisk
- menu.c32
- reboot.c32
- vesamenu.c32

7. Скачиваем образ ОС, поддерживающей liveCD, например ubuntu

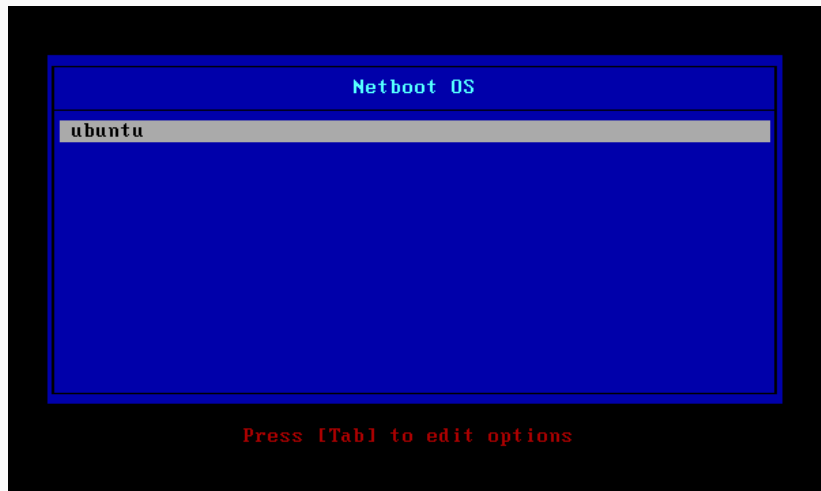
8. Разархивируем образ ОС, по пути `/usr/tftpboot/images/ubuntu/`;
9. По пути `/usr/tftpboot/pxelinux.cfg` создаем файл **default** со следующим содержимым:

```
ui menu.c32
menu title Netboot OS

LABEL ubuntu
    kernel images/ubuntu/casper/vmlinuz.efi
    append root=/dev/nfs boot=casper netboot=nfs nfsroot
    ↪ =192.168.120.3:/usr/tftpboot/images/ubuntu initrd=images/
    ↪ ubuntu/casper/initrd.lz
```

10. Перезгружаем ОС.

Запускаем бездисковые клиенты. При запуске всплывет меню, с выбором дальнейшей загрузки.



```
Terminal
ubuntu@ubuntu: ~

ubuntu@ubuntu:~$ ifconfig
ens33    Link encap:Ethernet  HWaddr 00:0c:29:67:55:a9
          inet addr:192.168.120.100  Bcast:192.168.120.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe67:55a9/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:449030 errors:4 dropped:2 overruns:0 frame:0
          TX packets:208265 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:671178686 (671.1 MB)  TX bytes:14693566 (14.6 MB)
          Interrupt:19 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128  Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:166 errors:0 dropped:0 overruns:0 frame:0
          TX packets:166 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:12158 (12.1 KB)  TX bytes:12158 (12.1 KB)

ubuntu@ubuntu:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=2 ttl=126 time=9.17 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=126 time=10.7 ms
^C
--- 8.8.8.8 ping statistics ---
3 packets transmitted, 2 received, 33% packet loss, time 2019ms
rtt min/avg/max/mdev = 9.173/9.971/10.770/0.804 ms
ubuntu@ubuntu:~$
```

```
Terminal Terminal File Edit View Search Terminal Help 6:54 PM
ubuntu@ubuntu: ~
See "man sudo_root" for details.

ubuntu@ubuntu:~$ ifconfig
ens32      Link encap:Ethernet  HWaddr 00:0c:29:1b:54:15
           inet addr:192.168.80.130  Bcast:192.168.80.255  Mask:255.255.255.0
           inet6 addr: fe80::20c:29ff:fe1b:5415/64 Scope:Link
           UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
           RX packets:360461 errors:26 dropped:18 overruns:0 frame:0
           TX packets:156858 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:1000
           RX bytes:540426723 (540.4 MB)  TX bytes:11133942 (11.1 MB)
           Interrupt:18 Base address:0x2000

lo         Link encap:Local Loopback
           inet addr:127.0.0.1  Mask:255.0.0.0
           inet6 addr: ::1/128 Scope:Host
           UP LOOPBACK RUNNING  MTU:65536  Metric:1
           RX packets:8 errors:0 dropped:0 overruns:0 frame:0
           TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:1000
           RX bytes:396 (396.0 B)  TX bytes:396 (396.0 B)

ubuntu@ubuntu:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=126 time=9.87 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=126 time=11.5 ms
^C
--- 8.8.8.8 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 9.876/10.700/11.524/0.824 ms
ubuntu@ubuntu:~$
```

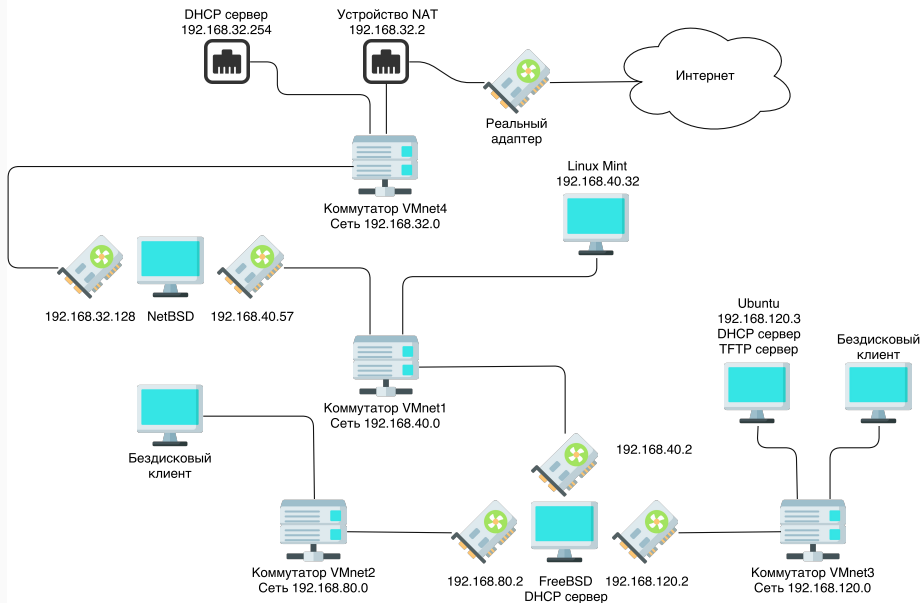
Как видно из результатов, клиенты посредством двух разных DHCP сервером успешно получили адреса, а также подключились к TFTP-серверу, вследствие чего была выполнена удаленная загрузка.

DNS СЕРВИСЫ

В данном разделе будут приведены примеры различных DNS серверов, в частности:

1. **Кэширующий сервер** - ищет все ответы на запросы от пользователей и запоминает их на случай повторного запроса
2. **Первичный мастер** - читает данные зоны из локального файла и является ответственным за эту зону
3. **Вторичный мастер** - получает данные по зоне с другого сервера имен, отвечающего за эту зону

В схеме ККС также будет использоваться Linux Mint(192.168.40.32).



Установим кэширующий DNS на Linux Mint(192.168.40.32)

1. Устанавливаем пакет **bind** командой

```
sudo apt-get install bind9
```

2. Внесем изменения в файл **/etc/bind/named.conf.options**:

```
options {  
    directory "/var/cache/bind";  
    forwarders {  
        8.8.8.8;  
    };  
    dnssec-validation auto;  
    auth-nxdomain no;  
    listen-on-v6 {any;};  
};
```

3. Перезапускаем DNS сервер

```
sudo /etc/init.d/bind9 restart
```

1. Настроим хост **192.168.120.3** на использование данного DNS сервера;
2. В файле **/etc/NetworkManager/NetworkManager.conf** комментируем строку **dns=dnsmasq**, иначе в nslookup будет другой адрес DNS;
3. Перезагружаем систему;
4. Вводим команду **nslookup ya.ru**, после чего в консоль будет выведено:

```
Server:      192.168.40.32
Address:     192.168.40.32#53

Non-authoritative answer:
Name:   ya.ru
Address: 87.250.250.242
```

Как видно, DNS успешно был определен.

Дополнительно проверим кэширующую способность DNS сервера, для этого введем команду **dig google.com** два раза:

```
dig google.com
...
;; Query time: 31 msec
;; SERVER: 192.168.40.32#53(192.168.40.32)
;; WHEN: Fri Mar 23 11:55:03 PDT 2018
;; MSG SIZE rcvd: 346

dig google.com
...
;; Query time: 0 msec
;; SERVER: 192.168.40.32#53(192.168.40.32)
;; WHEN: Fri Mar 23 11:55:03 PDT 2018
;; MSG SIZE rcvd: 346
```

Как видно, время запроса снизилось с 32 до 0 миллисекунд, что означает о наличии кэширующей возможности у DNS сервера.

Первичный сервер - **192.168.40.32**, вторичный - **192.168.120.3**.

1. В файл **/etc/bind/named.conf.local** вносим изменения:

```
zone "example.com" {  
    type master;  
    file "/etc/bind/db.example.com";  
    allow-transfer {192.168.40.0/24;192.168.80.0/24;192.168.120.0/24;};  
};  
zone "40.168.192.in-addr.arpa" {  
    type master;  
    file "/etc/bind/db.192";  
    allow-transfer {192.168.40.0/24;192.168.80.0/24;192.168.120.0/24;};  
};
```

Были созданы зоны **example.com** и обратная ей зона(для получения имени по IP)

Содержимое файла **db.example.com** выглядит следующим образом:

```
;
; BIND data file for example.com
;
$TTL      604800
@ IN      SOA     example.com. root.example.com. (
                    1          ; Serial
                    604800     ; Refresh
                    86400      ; Retry
                    2419200    ; Expire
                    604800 )   ; Negative Cache TTL
;
IN A      192.168.40.32
;
@ IN      NS      ns.example.com
@ IN      A       192.168.40.32
@ IN      AAAA     ::1
ns IN     A       192.168.40.32
```


Содержимое файла **db.192** выглядит следующим образом:

```
;
; BIND reverse data file for local 192.168.40.XXX net
;
$TTL      604800
@         IN      SOA      ns.example.com. root.example.com. (
                        1           ; Serial
                        604800      ; Refresh
                        86400       ; Retry
                        2419200     ; Expire
                        604800 )    ; Negative Cache TTL
;
@         IN      NS       ns.
32        IN      PTR      ns.example.com.
```

В данном случае адресу 192.168.40.32 соответствует имя example.com.

1. Устанавливаем пакет bind9;
2. В файл **/etc/bind/named.conf.local** вносим изменения:

```
zone "example.com" {  
    type slave;  
    file "db.example.com";  
    masters { 192.168.40.32; };  
};  
zone "40.168.192.in-addr.arpa" {  
    type slave;  
    file "db.192";  
    masters { 192.168.40.32; };  
};
```

Теперь данный сервер является вторичным по отношению к 192.168.40.32.

На первичном сервере вводим команду `cat /etc/log/syslog`

```
...  
... zone example.com/IN: loaded serial 1  
... zone 40.168.192.in-addr.arpa/IN: loaded serial 1  
... zone 255.in-addr.arpa/IN: loaded serial 1  
... zone localhost/IN: loaded serial 2  
... all zones loaded  
... running  
... zone 40.168.192.in-addr.arpa/IN: sending notifies (serial 1)  
...
```

Как видно из части лога, зоны были успешно загружены и переданы на вторичный сервер.

В данной работе был получен опыт по настройке DHCP и TFTP серверов, что позволяет загружать по сети ОС или утилиту.

Также были применены навыки работы с различными по функциональности DNS серверами. Так был получен кэширующий DNS сервер, который существенно ускоряет обращение к сетевым ресурсам. Также были сконфигурированы первичный и вторичный DNS сервера, ответственные за домен example.com и обратную зону.