Архитектура сетей и систем телекоммуникаций.

Практическая работа 4.

Работа с утилитой Netstat.

1. Теоритическая часть.

1.1 TCP и UDP

***Протокол пользовательских датаграмм (UDP)***

UDP протокол без подтверждения связи, то есть он не обеспечивает надежную доставку данных, например он не будет запрашивать повторно какой-то сегмент данных если он прибыл повреждённым. Сегменты UDP называются датаграммами. UDP подробно описан в RFC 768.

Приложения использующие UDP:

- DNS

- потоковое видео

- VoIP

***Протокол управления передачей (TCP)***

TCP - протокол, ориентированный на связь, подробно описывается в RFC 793. TCP гарантирует надежность передачи данных. TCP имеет дополнительные заголовки для реализации этих функций. Дополнительные функции: обеспечение надежной поставки данных, управление потоком передаваемых данных. Каждая сегмент TCP имеет 20 байт в заголовке, в то время как у UDP дейтаграммы заголовок только 8 байт.

Приложения использующие TCP:

- Web-браузеры

- Электронная почта

-FTP/SMB

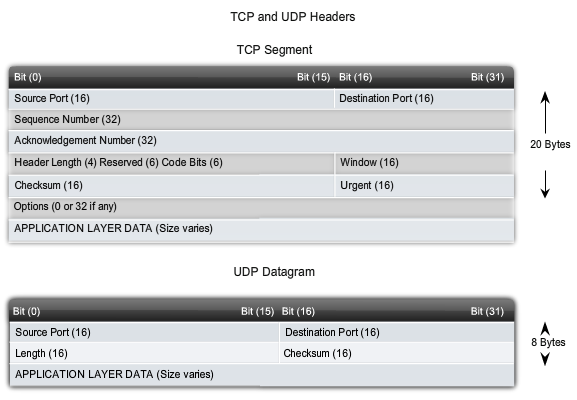


Рисунок 5.Строение сегмента TCP и UDP.

1.2 Адресация на основе портов

Чтобы правильно распределять сегменты для каждого приложения, TCP и UDP имеют заголовки, которые могут однозначно определить эти приложения. В них содержаться номера портов.

В заголовке каждого сегмента или датаграммы, содержится порт отправителя и получателя. Исходный номер порта – номер порта приложения на местном хосте. Номер порта адресата – номер порта приложения на хосте получателя.

Номера портов назначаются по разному, в зависимости от того, является ли сообщение запросом или ответом. Процессам на сервере назначаются постоянные номера портов.

Когда приложение на стороне клиент посылает запрос приложению на стороне сервера, порт адресата, имеющийся в заголовке - номер порта, который назначен для службы выполняющейся на сервере.

ПО клиента должно знать, какой номер порта связан с определенным процессом сервера с которым оно хочет связаться. Порт получателя конфигурируется либо вручную либо автоматически. Большинство приложений имеют заданные по умолчания номера портов, например WEB браузер – 80 порт.

Исходный порт в заголовке сегмента или датаграмме отправляемой клиентом на сервер, обычно назначен рандомно от значения 1023 до 65536 . Если этот порт не назначен на другое приложение в локальной системе, приложение может его выбрать.

Этот номер порта будет адресом портом получателя в пакете являющимся ответом сервера на запрос. Транспортный уровень сервера следит за портом и приложением, которое инициализировало запрос, чтобы ответ от сервера был возвращен правильному приложению.

Комбинация номера порта TCP/UDP с IP адресом называют гнездом.

Например: http://192.168.5.18:80

1.3 Распределение портов

Компания IANA распределяет IP адреса и номера портов для приложений.

**Известные Порты** (от 0 до 1023) - эти номера зарезервированы для служб и приложений. Например HTTP или POP3/SMTP

**Зарегистрированные Порты** (Номера 1024 - 49151) – эти номера назначаются на пользовательские процессы или приложения. Эти процессы – обычно прикладное ПО установленное пользователем. Если эти порты не используются ОС или юзером, то они могут назначаться сетевым приложениям автоматически как и динамические порты.

**Динамические или Частные Порты** (Номера 49152 - 65535) - они назначаются динамически на приложения-клиентов при инициализации подключения. Используются они не часто. Главные их потребители программы торрент клиенты.

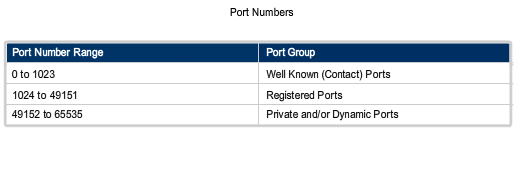


Рисунок 5. Диапазоны TCP/UDP портов.

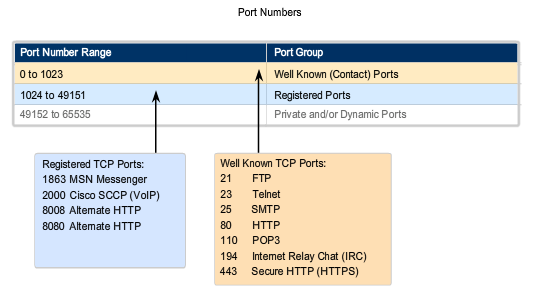


Рисунок 6. Часто используемые TCP порты “известного” диапазона

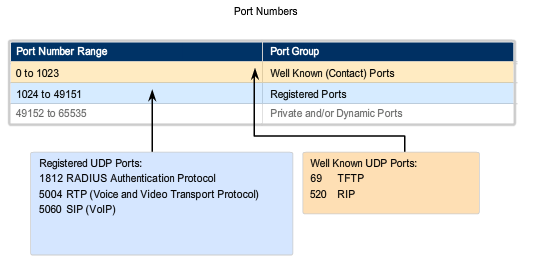


Рисунок 7. Часто используемые UDP порты “известного” диапазона.

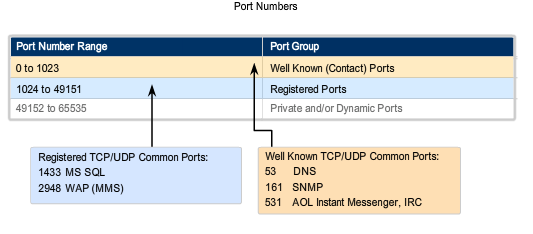


Рисунок 8. Часто используемые TCP/UDP порты “известного” диапазона.

Использование и TCP и UDP

Некоторые приложения могут использовать и TCP и UDP. В зависимости от конкретно выполняемых задач.

1.4 Просмотр открытых портов

Для проверки открыт порт или закрыт или проверки его статуса используется команда netstat. Она позволяет просмотреть информацию о связке локального адреса с портом и отдаленного адреса с портом. Если открыты непонятные соединения возможно это хакерская атака.

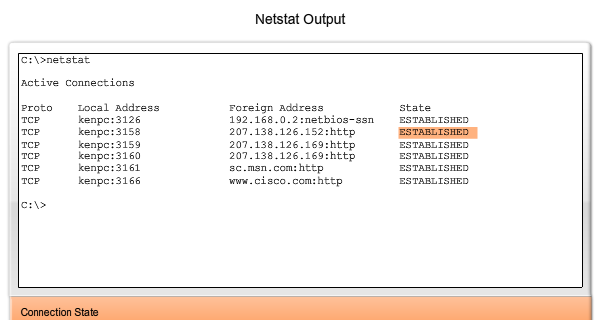


Рисунок 9. Просмотр открытых TCP и UDP портов в ОС Windows.

1.5 Утилита NETSTAT

Netstat это очень полезная утилита, которую некоторые системные администраторы используют каждый день, а некоторым приходится прибегать к ней только для диагностики неисправностей. Но в любом случае понимать эту утилиту и уметь её использовать очень полезно.

Вид команды одинаков для всех операционных систем - netstat -a  
Вывод команды для Windows:

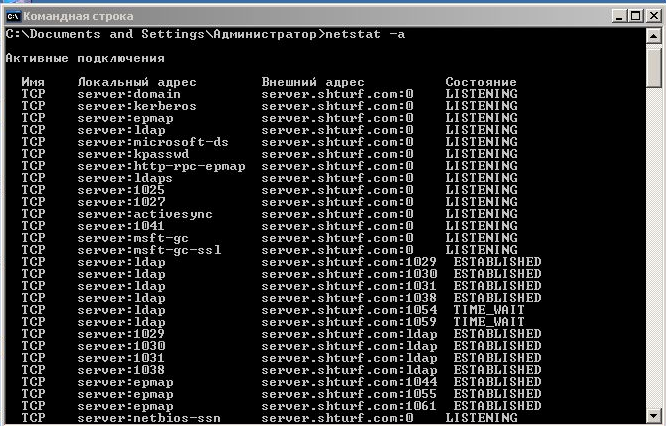


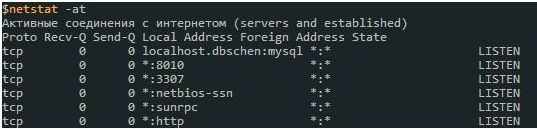
Рисунок 1.

Виды состояний (state):

ESTABLISHED - соединение установлено  
SYN\_SENT - сокет пытается установить соединение  
SYN\_RECV - начальная синхронизация соединения  
FIN\_WAIT1 - сокет закрыт, ожидание отключения соединения  
FIN\_WAIT2 - ожидание отключения удаленной стороны  
TIME\_WAIT - ожидание после закрытия повторной передачи отключения удаленной стороны  
CLOSED - соединение закрыто, сокет не используется  
CLOSE\_WAIT - удаленный узел отключился, сокет скоро будет закрыт  
LAST\_ACK - сначала отключилась удаленная сторона, а затем сокет будет закрыт  
LISTEN - сокет ожидает входящих соединений  
CLOSING - закрытие сокета, точнее сокет закрыт, а затем отключился удаленный узел  
UNKNOWN - состояние сокета неизвестно.

1. Просмотр только tcp соединений –

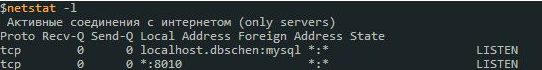
Для Линукс: netstat -at



|  |
| --- |
| Для Windows команда выглядит:  netstat -a -p tcp |

2) Просмотр всех сокетов в состоянии ожидания соединений (Listening)  
  
Для Linux:

netstat –l

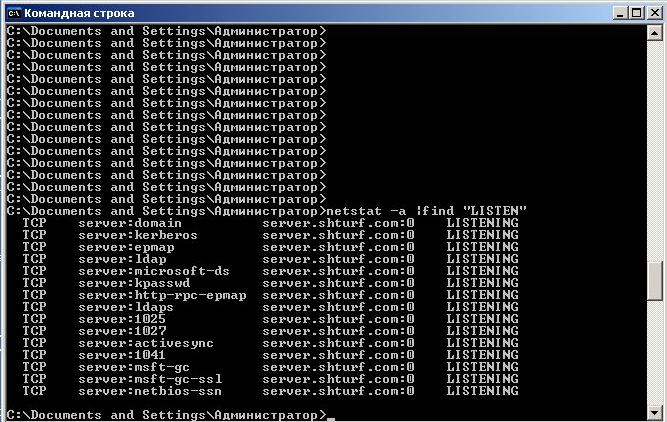


Для freebsd -

netstat -a | egrep 'Proto|LISTEN'

Для Windows –

netstat -a | find "LISTENING"

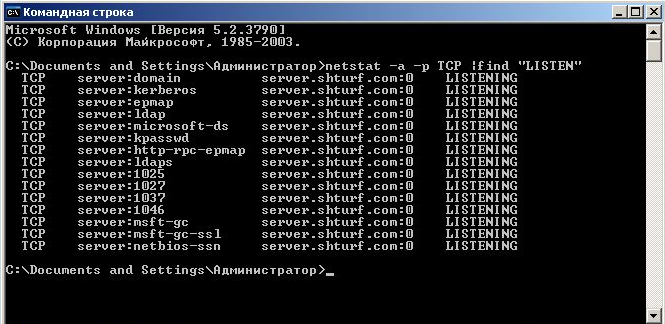


1. Просмотр только ожидающих TCP соединений -

Linux:  
netstat -lt

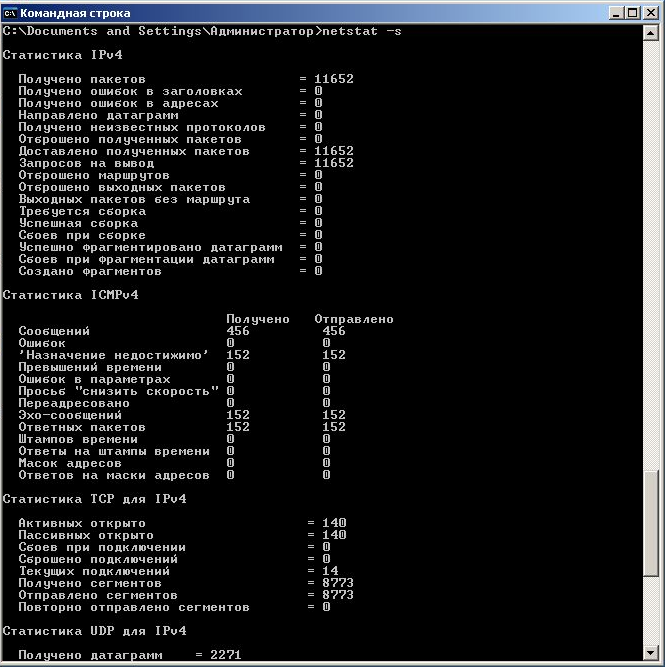
FreeBSD:  
netstat -a -p tcp | egrep 'Proto|LISTEN'

Windows:  
netstat -a -p TCP | find "LISTENING"



Разумеется все ожидающие UDP соединения просматриваются аналогично.

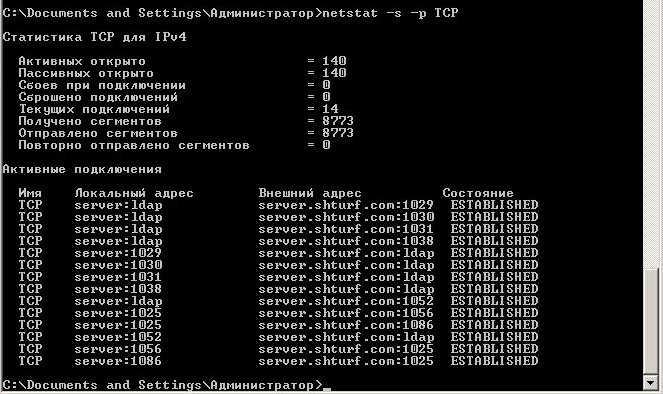
1. Показ **статистики** по каждому протоколу.  
     
   Для всех дистрибутивов -  
   netstat –s

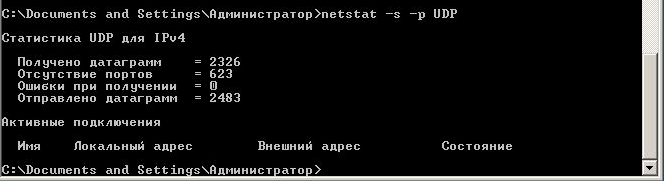


Для конкретных протоколов (например TCP или UDP):

Linux:  
TCP - netstat -st  
UDP - netstat -su

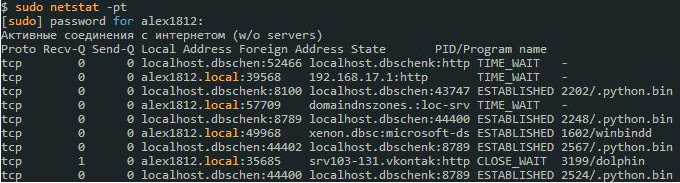
Freebsd и Windows:  
netstat -s -p TCP  
netstat -s -p UDP





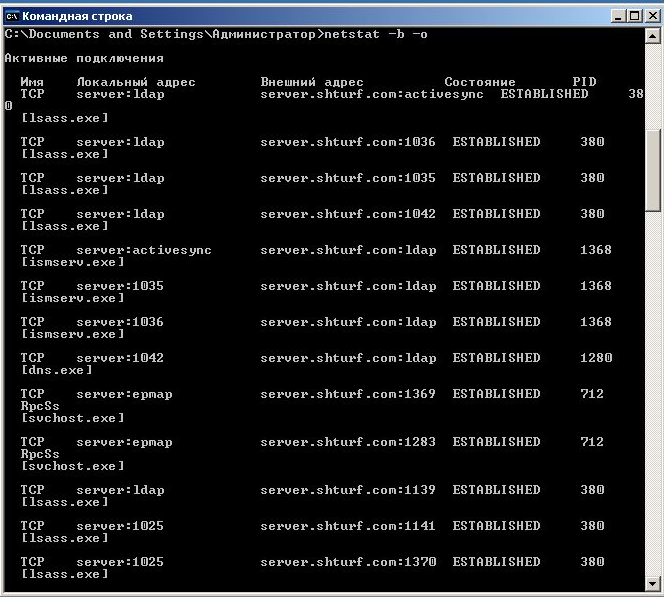
1. **Показывать PID и имя программы**

Для добавления поля PID/Program name в Linux используется ключ -p. Его можно использовать вместе с другими ключами.  
Пример:



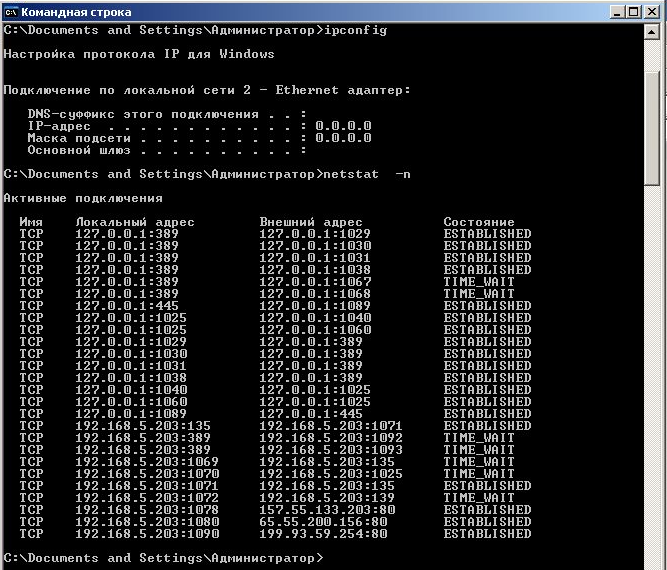
В windows это можно проделать с ключом -b и -o:

netstat -b -o



1. **Не включать хост, порт и имя пользователя в вывод netstat**

Для ускорения вывода, можно не включать вывод имени хоста, порта и имени пользователя, используем ключ -n. При этом будут выводиться числовые значение для имени пользователя, порта и айпи-адрес для доменных имен.

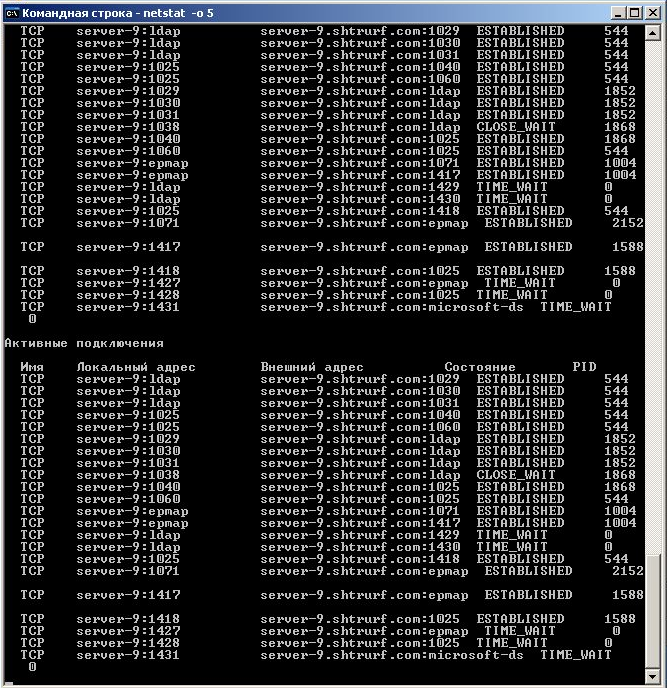


1. Выводить информацию непрерывно

Для Linux используем ключ -с, для вывода информации каждые несколько секунд

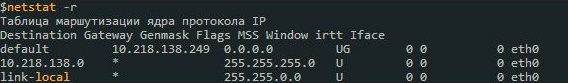
|  |
| --- |
| netstat –c |

Для Windows используется ключ -o  
netstat -o 5 - выводить каждые 5 секунд.

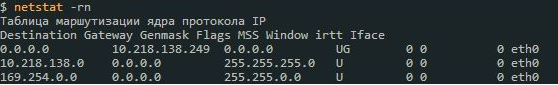


1. **Просмотр таблицы маршрутизации**

Для всех систем используем netstat –r



Для Linux и FreeBSD используем ключ -rn для преобразования имен хостов в ip-адреса



1. Практическая часть.

В лабораторной работе будет использоваться две ВМ, обе на базе ОС Linux. Первая ВМ будет выполнять роль клиента, с нее будут устанавливаться различные соединения к серверу и изучаться вывод программы netstat.

В качестве клиентской ВМ рекомендовано использовать ВМ на базе ОС Linux OpenSuSE. Для серверной ВМ предлагается использовать дистрибутив LiveCD Linux Eagle Server, который содержит в себе уже настроенные службы FTP/TELNET/TFTP/SMTP.

Задание 1.

Настройка ВМ-сервера Linux Eagle-Server.

1. Запустите программу эмуляции виртуальных машин Virtual Box.
2. Создайте в не виртуальную машину для Linux (32 bit).
3. Сетевой интерфейс виртуальной машины должен быть настроен как “Внутренняя сеть”
4. Включите машину
5. Подключите ISO образ с Live CD Linux Linux-eg-server-v2.0.iso
6. Перезагрузите машину
7. Перейдите в режим привилегированного пользователя в окне терминала. Для этого введите команду

**# su -**

пароль cisco

1. введите команду

**# ifconfig**

Определите имя первого в списке сетевого интерфейса. Оно должно быть таким: eth0. Если цифра в имени другая, используйте ее в последующих командах.

1. Адрес на интерфейсе надо настроить следующим образом – 192.168.100+номер по журналу.1, то есть как и в других работах если для примера брать номер по журналу 5, то адрес будет 192.168.105.1. Маска 255.255.255.0.

ifconfig eth0 <IP> netmask 255.255.255.0

В примере это будет:

**ifconfig eth0 192.168.105.1 netmask 255.255.255.0**

Шлюз по умолчанию в данной работе настраивать нет необходимости.

Задание 2.

**Запуск и настройка виртуальной машины клиента – Linux OpenSuSE**

1. Разверните из образа виртуальную машину Linux Suse 13.2 i586 minimal GUI. При развертывании дать название Linux SuSE Client-номер\_зачетки\_netstat
2. Зайти в настройки сетевого интерфейса виртуальной машины (меню VirtualBox) Windows 2008 и выставить тип сети “Внутренняя сеть”.
3. Загрузите виртуальную машину.
4. Проверьте название сетевого интерфейса и наличие на нем IP адреса, для чего использовать команду ifconfig.
5. Откройте графический конфигуратор системы YAST, настройки сети и настройте IP адрес и маску на сетевом интерфейсе enp0s3. По следующему принципу IP адрес – 192.168.100+номер по журналу.2, маска сети 255.255.255.0

Разумеется, настроить IP адрес сети можно и в консоли с помощью утилиты ifconfig, как мы это делаем в задании 1 для ВМ выполняющей роль Server, но данный способ не сохраняет настройки сети после перезагрузки ОС. К тому же стоит уметь настраивать IP адрес используя возможности YAST.

1. Настроить IP адрес 192.168.100+номер по журналу, маска 255.255.255.0

В примере это будет адрес 192.168.105.2 с маской 255.255.255.0

1. Проверить связь до виртуальной машины Cisco Eagle-Server.
2. Открыть файл /etc/hosts и добавить строчку

192.168.100+номер по журналу.1 eagle-server.example.com

В примере это будет:

192.168.105.1 eagle-server.example.com

9. Проверить связь до ВМ Cisco Eagle-Server по имени, связь должна быть успешной.

Задание 3.

1. Запустите командную строку и выполните команду netstat /?, просмотрите описание опций возможно при выполнении этой команды.

Задание 4.

1. Используйте команду netstat –a, чтобы посмотреть все открытые соединения.
2. Используйте команду netstat –n, чтобы посмотреть все открытые соединения, без разрешения имен.
3. С помощью программы telnet установить несколько подключений по разным портам к Eagle Server:

Пример для DNS запроса:

**telnet eagle-server.example.com 53**

Кроме запроса к службе DNS работающей на Eagle Server, надо таким же образом установить соединения еще по 21 (FTP), 25 (SMTP) и 23 (Telnet) порту.

1. С помощью утилиты netstat просмотрите TCP и UDP установленные соединения, а так же таблицу маршрутизации на компьютере и статистику переданного трафика – по всем протоколам, только TCP, только UDP.
2. Вопросы к лабораторной работе
3. Почему для передачи важных данных используется протокол TCP?
4. Почему для передачи видео используется протокол UDP?
5. Какие диапазоны TCP/UDP портов вы знаете?
6. Как называются данные формируемые транспортным уровнем?
7. Какие поля в заголовках пакетов транспортного уровня используются для адресации?
8. С помощью, какой утилиты помимо netstat можно посмотреть настроенные маршруты в ОС Windows?