

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Чернігівський державний технологічний університет

# **АНАЛІЗ ФУНКЦІОНУВАННЯ ЛОМ**

МЕТОДИЧНІ ВКАЗІВКИ

до лабораторних робіт з дисципліни  
“Комп’ютерні мережі”  
для студентів напрямку 0915 - “Комп’ютерна інженерія”

Чернігів ЧДТУ 2008

Мережі на базі протоколу ІР. Методичні вказівки до лабораторних робіт з дисципліни “Комп’ютерні мережі” для студентів напрямку “Комп’ютерна інженерія”/ Укл. О. В. Лукин, А. Л. Зінченко - Чернігів: ЧДТУ, 2006 - 36с. Рос. мовою.

Составитель:

А. В. Лукин, старший преподаватель

А. Л. Зинченко, ассистент

Ответственный за выпуск:

В.И. Павловский, зав. кафедрой

информационных и компьютерных систем,  
канд. техн. наук, доцент

Рецензент:

В.И. Павловский, канд. техн. наук, доцент

<u>Введение.....</u>	<u>4</u>
<u>Лабораторная работа 1. Конфигурирование и тестирование сетевых интерфейсов.....</u>	<u>4</u>
<u>Теоретические сведения.....</u>	<u>4</u>
<u>Ход работы.....</u>	<u>10</u>
<u>Содержание отчета.....</u>	<u>10</u>
<u>Контрольные вопросы.....</u>	<u>10</u>
<u>Теоретические сведения.....</u>	<u>11</u>
<u>Факторы, влияющие на пропускную способность сети.....</u>	<u>11</u>
<u>Физическая предельная скорость передачи.....</u>	<u>11</u>
<u>Уровень ошибок в канале передачи данных.....</u>	<u>12</u>
<u>Загрузка сети служебными данными.....</u>	<u>12</u>
<u>Общая загрузка сети.....</u>	<u>13</u>
<u>Задержки в сети.....</u>	<u>13</u>
<u>Измерение пропускной способности сети.....</u>	<u>14</u>
<u>Способы гарантирования заданной пропускной способности сети.....</u>	<u>14</u>
<u>Ход работы.....</u>	<u>15</u>
<u>Содержание отчета.....</u>	<u>16</u>
<u>Контрольные вопросы.....</u>	<u>16</u>
<u>Лабораторная работа</u>	<u>3</u>
<u>Изучение сетевых анализаторов tcpdump и Ethereal.....</u>	<u>17</u>
<u>Цель работы.....</u>	<u>17</u>
<u>Теоретические сведения.....</u>	<u>17</u>
<u>Утилита tcpdump.....</u>	<u>17</u>
<u>Сетевой анализатор ethereal.....</u>	<u>18</u>
<u>Ход работы.....</u>	<u>25</u>
<u>Содержание отчета.....</u>	<u>25</u>
<u>Контрольные вопросы.....</u>	<u>25</u>
<u>Лабораторная работа</u>	<u>4.</u>
<u>Изучение сетевого протокола TCP и протокола уровня приложений telnet.....</u>	<u>25</u>
<u>Теоретические сведения.....</u>	<u>25</u>
<u>Ход работы.....</u>	<u>26</u>
<u>Содержание отчёта.....</u>	<u>27</u>
<u>Лабораторная работа</u>	<u>5.</u>
<u>Изучение сетевого протокола UDP и протокола уроня приложений DNS.....</u>	<u>27</u>
<u>Теоретические сведения.....</u>	<u>27</u>
<u>Протокол UDP.....</u>	<u>27</u>
<u>DNS.....</u>	<u>28</u>
<u>Ход работы.....</u>	<u>28</u>
<u>Содержание отчёта.....</u>	<u>29</u>
<u>Контрольные вопросы.....</u>	<u>29</u>
<u>Рекомендованная литература:.....</u>	<u>30</u>

# Введение

Межсетевой протокол IP на сегодняшний день доминируют как в локальных, так и в глобальных сетях. С развитием сети Интернет стек протоколов TCP/IP «оброс» огромным количеством сетевых сервисов, что и обусловило доминирование данного семейства протоколов во всех разновидностях сетей.

Данное руководство предназначено для начального знакомства со стек протоколов TCP/IP, а так же взаимодействию меж сетевого протокола с несущими сетями, в частности, с сетями, построенными по технологии Ethernet.

Цикл лабораторных работ, предлагаемый в данном пособии, поможет студентам усвоить базовые навыки по конфигурированию сетевых интерфейсов и устройств, диагностике сети, работе с сетевыми анализаторами. Кроме того, студентам предстоит научиться конфигурировать базовые системные сервисы TCP/IP сетей, такие как сервис имен DNS и сервис динамического конфигурирования хостов DHCP.

## Лабораторная работа 1. Конфигурирование и тестирование сетевых интерфейсов.

### Теоретические сведения

Логическое устройство, осуществляющее передачу данных в сети на логическом уровне называется сетевым интерфейсом. Сетевой интерфейс может быть связан с определенным физическим устройством (например, адаптер Ethernet, последовательный порт), а может быть просто логическим интерфейсом (например, интерфейс локальной обратной связи).

Сетевые интерфейсы делятся на 2 основных типа: интерфейс в широковещательную сеть, broadcast (например, Ethernet) и интерфейс “точка-точка”, point-to-pint. Первые ассоциируются с одним сетевым адресом, вторые – с двумя, локальным и удаленным адресом. Сетевой интерфейс может использоваться для передачи данных по различным протоколам: IP v.4, IP v.6, IPX, AppleTalk и т.д. Наибольшее распространение получил на сегодня протокол IP v.4 (далее – IP), поскольку он используется в Интернете. В ОС Windows возможно использование протокола NetBIOS, однако, с точки зрения

“прозрачности” сети рекомендуется использовать как основной протокол IP, с которым ассоциируются службы протокола NetBIOS. В ОС UNIX используются дополнительные программы для организации сервисов протокола NetBIOS (пакет samba). Далее будем рассматривать только протокол IP.

Сетевые интерфейсы могут иметь псевдонимы, т.е. с интерфейсом могут ассоциироваться несколько сетевых адресов. Псевдонимы позволяют использовать различные схемы адресации на одном сегменте сети. Например, возможно одновременное использование частных и публичных IP адресов.

Для конфигурирования сетевых интерфейсов в POSIX-совместимых ОС используется команда `ifconfig`. Команда `ifconfig` без параметров выдает текущую конфигурацию активных сетевых интерфейсов. Ниже приведен пример для машины с двумя адаптерами Ethernet и выходом в Интернет через последовательный порт. Машина работает под управлением ОС Linux.

```
[root@localhost]# ifconfig

ppp0 Link encap:Point-to-Point Protocol

inet addr:193.41.61.30 P-t-P:193.41.61.130 Mask:255.255.255.255

UP POINTOPOINT RUNNING NOARP MULTICAST MTU:1500 Metric:1

RX packets:3 errors:0 dropped:0 overruns:0 frame:0

TX packets:4 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:3

RX bytes:54 (54.0 b) TX bytes:76 (76.0 b)

eth0 Link encap:Ethernet HWaddr 00:04:76:9F:4D:8E

inet addr:192.168.0.16 Bcast:192.168.0.255 Mask:255.255.255.0

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

RX packets:196365 errors:0 dropped:0 overruns:1 frame:0

TX packets:131396 errors:0 dropped:0 overruns:0 carrier:0

collisions:194 txqueuelen:1000

RX bytes:18393948 (17.5 Mb) TX bytes:12045495 (11.4 Mb)

Interrupt:11 Base address:0xc000
```

```
eth0:1 Link encap:Ethernet HWaddr 00:04:76:9F:4D:8E
inet addr:195.69.76.139 Bcast:195.69.76.159 Mask:255.255.255.224
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:196365 errors:0 dropped:0 overruns:1 frame:0
TX packets:131396 errors:0 dropped:0 overruns:0 carrier:0
collisions:194 txqueuelen:1000
RX bytes:18393948 (17.5 Mb) TX bytes:12045495 (11.4 Mb)
Interrupt:11 Base address:0xc000
```

```
lo Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:11044 errors:0 dropped:0 overruns:0 frame:0
TX packets:11044 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:1532839 (1.4 Mb) TX bytes:1532839 (1.4 Mb)
```

Как видно из примера, для каждого устройства в первой строке выдается его имя и, если таковое имеется, устройство, в которое инкапсулируются данные уровня IP. Для `ppp0` – это протокол PPP, для `eth0` это адаптер Ethernet с соответствующим MAC адресом. Устройство `eth0:1` – это псевдоним для устройства `eth0` с дополнительным IP адресом. Обратите внимание, что `eth0` и `eth0:1` имеют одинаковый MAC адрес. Устройство `lo` – это специально устройство локальной обратной связи, предназначенное для обеспечения связи на IP уровне процессов, работающих на одном и том же компьютере.

Следующая строка показывает параметры, связанные с IP адресацией данного сетевого интерфейса. Для устройства `ppp0` показан локальный адрес, удаленный адрес и сетевая маска. Для интерфейсов широковещательного типа, т.е. `eth0` и `eth0:1` показан адрес интерфейса, широковещательный адрес данной подсети и сетевая маска. Для устройства `lo` дан адрес и маска, причем IP параметры этого устройства всегда именно такие.

Третья строчка показывает дополнительные параметры интерфейса. UP обозначает, что интерфейс в данный момент активен. Интерфейс может быть деактивирован командой `ifconfig имя down` и активирован командой `ifconfig имя up`. Следующий параметр характеризует тип интерфейса. Слово `RUNNING` обозначает, что данные

передаются или принимаются через интерфейс в течении последнего времени, обычно 10 секунд. Параметр NOARP обозначает, что данный интерфейс не поддерживает протокол ARP, обеспечивающий привязку адресной схемы несущей сети к IP адресации. Слово MULTICAST обозначает, что данный интерфейс поддерживает групповой режим передачи данных, т.е. данные могут передаваться одним потоком сразу нескольким хостам, участвующим в multicast сессии. Параметр MTU (Maximum Transfer Unit) показывает максимальный размер пакета в байтах, который может быть передан через данный интерфейс. При необходимости IP пакеты могут быть фрагментированы и разбиты на несколько пакетов в соответствии с MTU данного интерфейса. Например, в каком-либо интерфейсе MTU = 500 байт. Тогда IP пакет размером 1500 байт будет разбит на 3 пакета по 500 байт. Параметр Metric показывает условный вес интерфейса, используемый протоколами динамической маршрутизации, которые вычисляют вектор расстояния для определения наилучшего маршрута. Метрика задается при конфигурировании интерфейса и по умолчанию равна 1. Чем больше метрика интерфейса, тем менее приоритетным является маршрут через него. Например, у одного интерфейса метрика равна 1, а у другого равна 2. Наилучший маршрут будет включать первый интерфейс.

Остальные строчки показывают статистику на интерфейсе. RX – стандартное сокращение для приема, TX – для передачи. Статистика показывает общее число пакетов, прошедших через данный интерфейс с момента его инициализации (packets), количество ошибок при приеме и передаче (errors), количество пакетов, «выброшенных» из-за несовпадения контрольной суммы заголовка (dropped), количество пакетов, потерянных из-за переполнения буфера (overruns), количество неправильно принятых фреймов несущей сети (frame), количество случаев потери несущей в физической сети (carrier), количество конфликтов (collisions) и количество пакетов в очереди передачи на момент выполнения команды. Также показано общее количество переданных и принятых через этот интерфейс байт данных. И, наконец, для интерфейсов, имеющих корреспондирующее физическое устройство, показаны некоторые параметры данного устройства.

В ОС FreeBSD команда ifconfig не выдает статистику на интерфейсе и показывает все псевдонимы, связанные с данным интерфейсом. Ниже приведен пример для одного интерфейса типа Ethernet. Следует отметить, что в ОС Linux принято все интерфейсы Ethernet именовать eth0, eth1 и т.д., а в BSD системах имя интерфейса зависит от типа адаптера, т.е. драйвера данного адаптера. Например, NE2000 совместимые карты имеют имя с префиксом ed, карты на чипсетах RealTec 81xx имеют имя с префиксом rl, и т.п.

```
rl0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
inet 195.69.76.130 netmask 0xfffffe0 broadcast 195.69.76.159
inet 192.168.0.10 netmask 0xfffff00 broadcast 192.168.0.255
ether 00:20:ed:5b:6f:e2
media: Ethernet autoselect (100baseTX <full-duplex>)
status: active
```

В ОС Windows редко используется командная строка, однако имеется команда `ipconfig`, выдающая аналогичную информацию.

В обычном режиме работы конфигурирование сетевых интерфейсов происходит в процессе загрузки ОС в соответствии либо с заранее введенными параметрами, либо в соответствии с информацией, полученной в сети от сервера по протоколу DHCP (Dynamic Host Configuration Protocol). В последнем случае посылается широковещательный пакет с запросом к серверу DHCP, который по MAC адресу запрашивающего компьютера выдает пакет с необходимыми для конфигурирования сетевого интерфейса данными.

Рассмотрим процедуру конфигурирования интерфейсов в ОС Linux на примере дистрибутива RedHat и ему подобных, следующих рекомендациям POSIX на процесс загрузки ОС. В виду того, что программы старта все написаны на языке shell и хорошо откомментированы, они очень удобны для ознакомления с процессом конфигурирования сетевых сервисов. И так, после загрузки ядра ОС запускается процесс `init`, который читает файл конфигурации `/etc/inittab` и “раскручивает” ОС до указанного в нем уровня выполнения (run level). Для любого уровня выполнения запускается скрипт на языке shell `/etc/rc.d/rc` с параметром, равным уровню выполнения. Этот скрипт выполняет скрипты в директории `/etc/rc.d/rcX.d/`, где X – номер уровня выполнения. Уровень 3 используется для многопользовательского режима работы без старта графической оболочки, уровень 5 – с графической оболочкой. Файлы в этих директориях представляют символические ссылки на файлы shell программ в `/etc/rc.d/init.d/` для запускаемых в процессе старта ОС сервисов. Имена ссылок начинаются с букв S (Start) или K (Kill), за которыми следует 2 цифры и имя сервиса. Цифры обеспечивают очевидный порядок запуска сервисов. Когда происходит выход с данного уровня выполнения, программа `/etc/rc.d/rc` запускают ссылки, начинающиеся с буквы K с параметром `stop`, когда происходит вход – ссылки, начинающиеся с буквы S с параметром `start`. Следовательно, любой сервис может быть запущен или остановлен вручную при помощи запуска соответствующей программы из `/etc/rc.d/init.d/`. Такой подход обеспечивает простую инсталляцию программ и максимальные удобства по добавлению или удалению сервисов в автоматически стартуемые.

Одним из первых стартует сервис `network`, который и обеспечивает конфигурирование сетевых интерфейсов. Текст в `/etc/rc.d/init.d/network` показывает, что в начале считывается файл `/etc/sysconfig/network`, а затем происходит переход в директорию `/etc/sysconfig/network-scripts`, и выполняются просмотр ее на предмет наличия файлов с шаблоном `ifcfg-*`. Вторая часть имени этих файлов – имя сетевого интерфейса. Эти файлы содержат необходимые параметры для активных сетевых интерфейсов. Для каждого найденного файла выполняется программа `/sbin/ifup` имя\_интерфейса при старте и `/sbin/ifdown` имя\_интерфейса при останове системы. При рассмотрении текста этих программ упрощенно можно сказать, что в конце концов вызывается команда `ifconfig` для каждого интерфейса с соответствующими параметрами. Ниже приведен пример файла конфигурации для сетевого интерфейса типа ethernet.

Конфигурирование интерфейса “вручную” необходимо в редких случаях для отладки специфичных параметров. Для конфигурирования интерфейса в команде `ifconfig` необходимо указать адресные параметры интерфейса и дополнительные опции.

Тестирование работоспособности сети производится при помощи программы `ping`. Данная программа посылает ICMP пакеты `echo request`, дожидается пакетов `echo reply` и показывает задержку между запросом и ответом. В качестве параметра команде `ping` передается адрес близлежащего работоспособного хоста. Ниже приведен пример проверки связи через GPRS.

```
[root@localhost]# ping 193.41.60.55
PING 193.41.60.55 (193.41.60.55) 56(84) bytes of data.
64 bytes from 193.41.60.55: icmp_seq=0 ttl=62 time=689 ms
64 bytes from 193.41.60.55: icmp_seq=1 ttl=62 time=644 ms
64 bytes from 193.41.60.55: icmp_seq=2 ttl=62 time=603 ms
64 bytes from 193.41.60.55: icmp_seq=3 ttl=62 time=544 ms
64 bytes from 193.41.60.55: icmp_seq=4 ttl=62 time=622 ms
64 bytes from 193.41.60.55: icmp_seq=5 ttl=62 time=580 ms
64 bytes from 193.41.60.55: icmp_seq=6 ttl=62 time=675 ms
64 bytes from 193.41.60.55: icmp_seq=7 ttl=62 time=614 ms
64 bytes from 193.41.60.55: icmp_seq=8 ttl=62 time=573 ms
64 bytes from 193.41.60.55: icmp_seq=9 ttl=62 time=670 ms
64 bytes from 193.41.60.55: icmp_seq=10 ttl=62 time=628 ms
64 bytes from 193.41.60.55: icmp_seq=11 ttl=62 time=586 ms
^C
--- 193.41.60.55 ping statistics ---
13 packets transmitted, 12 received, 7% packet loss, time 12021ms
rtt min/avg/max/mdev = 544.915/619.739/689.007/42.723 ms, pipe 2
```

Для каждого принятого echo reply выдается количество принятых байтов, номер последовательности, время жизни вернувшегося пакета и время пробега туда и обратно.

Если номера последовательности идут не по порядку, то пакеты в сети теряются, что свидетельствует о плохом канале связи. Программа ping работает, пока ее не прервут нажатием Ctrl-C. После этого обрабатывается статистика, и выводятся данные о проценте потери пакетов и о минимальном/среднем/максимальном времени распространения пакетов туда и обратно, а так же максимальная девиация времени распространения.

Для оценки качества канала связи необходимо накопить достаточную статистику, т.е. команда ping должна работать некоторое время. Обычно достаточно 5 минут для достаточно достоверной оценки. Если процент потери пакетов не превышает 0.1%, канал считается работоспособным.

При недостижимости указанного в команде адреса наблюдается картина, отличная от приведенной в примере. Это свидетельствует о проблемах с сетью. Поиск проблемы начинается с проверки параметров интерфейсов командой /sbin/ifconfig -a, которая покажет состояние всех сетевых интерфейсов. При этом следует обратить внимание на правильность адресных параметров и на параметры frame, carrier. Ненулевые значения этих параметров свидетельствуют о проблемах с физическим уровнем сети. Проблемы могут появиться и вследствие неправильной маршрутизации, однако этот вопрос будет рассмотрен позже.

## **Ход работы.**

1. Проанализировать состояние сетевых интерфейсов рабочей станции. Для этого необходимо дать команду /sbin/ifconfig -a. Пояснить значение всех параметров сетевых интерфейсов.
2. Проверить время задержки внутри сегмента сети и за пределами сегмента. Для проверки необходимо дать команду ping имя\_хоста и накопить достаточную статистику.
3. Проверить работу команды ping и проанализировать состояние сетевого интерфейса при кратковременном физическом повреждении сети. Для выполнения этого пункта запустите команду ping и во время ее выполнения на короткое время отсоедините сетевой соединитель от сетевой карты компьютера. Проанализируйте и отметьте полученные результаты.
4. Ознакомьтесь со скриптами запуска системы и инициализации сетевых интерфейсов.
5. Ознакомьтесь с документацией по команде ifconfig.

## **Содержание отчета.**

В отчете должны быть приведены результаты выполнения п.1-3 с соответствующими пояснениями.

## **Контрольные вопросы.**

1. Какие параметры сетевого интерфейса свидетельствуют о проблемах в физической среде передачи данных?
2. Какие параметры интерфейса свидетельствуют о чрезмерной загрузке сети?
3. Какое среднее значение задержки в сети 100 Мбит/с?
4. Чем определяется время задержки в сети Ethernet?
5. Почему в стандарте 100baseT длина некоммутируемого сегмента не должна превышать 100 метров? Подкрепите ваше утверждение расчетами времени распространения 1 бита информации и одного кадра информации.
6. Перечислите основные параметры команды `ifconfig` при конфигурировании широкополосного сетевого интерфейса.
7. Перечислите основные параметры команды `ifconfig` при конфигурировании сетевого интерфейса точка-точка.
8. Опишите процесс конфигурирования сетевого интерфейса в ОС RedHat Linux при загрузке.
9. Как добавить сетевой интерфейс? Как добавить псевдоним для сетевого интерфейса?

## Лабораторная работа 2. Определение реальной пропускной способности сети.

### Теоретические сведения

Реальная пропускная способность сети зависит от множества факторов и зачастую является случайным параметром. Только в синхронных сетях передачи данных, используемых в цифровой телефонии, пропускная способность сети является фиксированной величиной, однако реально при наличии ошибок передачи пропускная способность может быть ниже декларированной.

Рассмотрим факторы, влияющие на пропускную способность сети Ethernet и подобных сетей с коммутацией фреймов.

### Факторы, влияющие на пропускную способность сети.

Наибольшее распространение в сетях передачи данных получили стандарты, не поддерживающие гарантии качества передачи. Ярким примером может быть сеть Ethernet, в которой нет никаких механизмов гарантирования скорости передачи, и, кроме того, нет механизмов гарантирования времени ожидания перед передачей очередного пакета и механизмов избежания конфликтов при передаче. Все участники обмена в сети работают в случайном режиме, следовательно, и параметры, описывающие пропускную способность, будут случайными величинами. Таким образом, можно привести лишь статистически усредненные параметры сети. Рассмотрим основные факторы, влияющие на производительности сети.

### Физическая предельная скорость передачи.

Наиболее распространенной на сегодня является сеть Ethernet 100 Мбит/с на базе медной витой пары. Попробуем приблизительно определить предельную скорость в этой сети по данным с использованием протокола TCP.  
 $100000000 \text{ бит/с} / 8 = 12500000 \text{ байт/с}$ . - это предельная максимальная скорость в

дуплексном канале, не учитывающая временного зазора между пакетами и служебный байтов заголовков. Потери на зазоры составляют примерно 5%, т.е. реальная скорость составляет

$12500000 \text{ байт/с} * 0.95 = 11875000 \text{ байт/с}$ . Кроме того, на каждые 1500 байт информации (стандартный MTU), инкапсулированной во фрейм Ethernet, приходится 16 байтов заголовка Ethernet. Далее, в 1500 байтах содержится как минимум один IP заголовок и, скорее всего, один TCP заголовок, т.е.  $20+20=40$  байт, т.е.  $56/1516*100\%=3.7\%$ . Следовательно, в идеальном случае получаем  $11875000*0.963=11435625 \text{ байт/с}$  по данным, т. 11.4 Мбайт/с. Это, естественно, для случая, когда в сети "путешествуют" только пакеты максимального размера. Однако, если размер пакета меньше, то на него приходится больше в процентном отношении служебных данных и реальная скорость по данным будет ниже вычисленной нами величины.

Интересна еще одна цифра – время, за которое передается минимальный пакет.

### **Уровень ошибок в канале передачи данных.**

При определенном уровне ошибок работоспособность сети не утрачивается, а происходит снижение реальной скорости передачи данных, поскольку зачастую при передаче данных используется потоковый протокол TCP, обеспечивающий повторную передачу потерянных данных. Однако, чем выше уровень ошибок в канале, тем больше времени теряется на повторную передачу и, в конце концов, пропускная способность падает до нуля. Для пакетных протоколов (UDP, ICMP, и подобные) потери в канале являются катастрофическими, поскольку не используются никакие механизмы управления потоком данных. Для синхронных сетей уровень ошибок в канале  $10^{-7}$  является нормальным,  $10^{-6}$  – аварийным. Для TCP/IP сетей требования несколько ниже. Нормальным считается уровень ошибок  $10^{-4}$ , критическим  $10^{-3}$ , аварийным –  $10^{-2}$ .

### **Загрузка сети служебными данными.**

Английский термин для данного фактора звучит как network overhead. Данный фактор можно разделить на две составляющие: Первая – длина служебных заголовков пакета данных. Это влияет на пропускную способность напрямую, как видно из приведенных выше расчетов предельной скорости передачи данных. Например, при передаче интерактивных данных по протоколу telnet каждая нажатая на терминале клавиша передается отдельным пакетом, вследствие чего реальная пропускная способность по данным составляет менее 1%. Следовательно, Вторая составляющая – данные различных служебных сетевых протоколов. Например, сети на основе протокола NetBIOS очень активно используют широковещательные пакты (broadcast) для анонсирования и запроса доступности сетевых сервисов. Если не принимать дополнительных мер по ограничению распространения broadcast пакетов, они могут "съесть" до 30% полосы пропускания сети 10 Мбит/с. Данный фактор определяется как типом сервисов, используемым в сети, так и количеством рабочих станций в сегменте сети. Именно поэтому не рекомендуется использовать более 200 рабочих станций в коммутируемом сегменте 100 Мбит/с и не рекомендуется использовать протокол NetBIOS в "чистом виде". Рекомендуется использовать стек протоколов TCP/IP и к нему привязывать службы сетей Windows.

В сетях с общей средой передачи, например, в Ethernet на коаксиале или на концентраторах (hub) высокая загрузка сети вызывает конфликты доступа к среде передачи (collisions), т.е. ситуации, когда одновременно начинают передавать несколько

устройств. При этом пакет данных не теряется, а задерживается в буфере передачи, что приводит к “мягкому” снижению скорости передачи до уровня загрузки около 50% от общей полосы пропускания. При большем уровне загрузки может появиться существенная потеря пакетов за счет переполнения буферов передачи. Естественно, это происходит только в случае использования протоколов без механизма управления потоком. Например, механизм потоковых сокетов просто блокирует процесс, данные которого не успели покинуть буфер, предотвращая таким образом переполнение буфера передачи.

### **Общая загрузка сети.**

Анализируя данный фактор, любой канал связи удобно рассматривать как некоторую “трубу” с определенной пропускной способностью (диаметр на скорость распространения), а пакеты данных – как некоторое зернистое сыпучее вещество. Естественно, что через трубу может просыпаться вполне конкретное количество вещества за единицу времени. Такая аналогия вполне справедлива для некоммутируемых сетей Ethernet, для синхронных и асинхронных каналов точка-точка, для портов маршрутизаторов. Для коммутируемых сетей ситуация несколько иная. Если производительность коммутирующей матрицы позволяет работать всем портам коммутатора со “скоростью провода”, то трафик между двумя отдельными портами совершенно не влияет на работу остальных портов, В противном случае при достижении предельной пропускной способности матрицы скорость будет снижаться для всех портов коммутатора.

Для сетей с маршрутизаторами необходимо использовать хотя бы минимальные механизмы управления потоком при передаче по базовому протоколу (например, IP) иначе потеря пакетов неизбежна. Представьте себе ситуацию, когда у маршрутизатора один порт 100 Мбит/с, а другой – 64 Кбит/с. Данные, приходящие со скоростного порта никак не смогут в том же темпе передаваться во второй порт, следовательно маршрутизатор должен передать источнику пакетов сообщение, что буферы заполнены и источник данных должен временно прекратить передачу. Такие минимальные функции управления потоком IP пакетов заложены в протокол ICMP. Следовательно, пропускная способность сетей с маршрутом определяется пропускной способностью соответствующего порта маршрутизатора и его загрузкой.

### **Задержки в сети.**

Задержки в сети, на первых взгляд, не должны влиять на пропускную способность, однако это справедливо только для протоколов без управления потоком. В случае, когда необходимо дожидаться подтверждения получения определенной порции данных, прежде чем передавать следующую порцию, задержки существенно влияют на скорость передачи данных между парой потоковых сокетов. Например, спутниковый канал с пропускной способностью 10 Мбит/с и задержкой 600 мс для пары сокетов даст всего около 1 Мбит/с. Конечно, если канал используется несколькими сокетами, то он может использоваться на все 100%.

Кроме того, задержки выше 250 мс заметны пользователям интерактивных приложений и существенно ухудшают субъективное качество голосовой связи.

Приведенные выше факторы являются основными, но список не является исчерпывающим. Так, зачастую, маршрутизаторы могут не справляться с полным объемом трафика за счет недостаточной производительности шины, памяти и процессора.

## **Измерение пропускной способности сети.**

Поскольку пропускная способность сети зависит от большого количества факторов, многие из которых носят случайный характер, точный расчет пропускной способности весьма затруднителен. Поэтому гораздо проще измерить реальную пропускную способность сети экспериментально. Здесь следует учитывать, что пропускная способность зависит еще и типа используемого протокола. В TCP/IP сетях необходимо отдельно измерять пропускную способность по двум транспортным протоколам – TCP и UDP. Стоит так же отметить, что не всегда доступно управление и даже оценка загрузки всеми узлами сети по пути распространения пакетов.

В первую очередь необходимо оценить задержки в канале связи и уровень потери пакетов при помощи команды ping (см. выше).

Во вторую очередь следует определиться, какой интервал усреднения нас интересует, т.е. для какого типа сервиса будет измеряться скорость передачи. В пределах данной работы будем считать, что нас интересует потоковая передача большого объема данных.

Для оценки скорости соединения точка-точка по протоколу TCP проще всего использовать сервис FTP. Для этого на одном хосте запускается сервер ftpd, а на другом – любой клиент ftp, который оценивает среднюю скорость передачи, например mc (Midnight Commander). при помощи клиента необходимо осуществить загрузку файла такого объема, что бы успеть пронаблюдать процесс загрузки около 1 минуты. Для более точного измерения пропускной способности существуют специальные программы, например tcplblast. Требуется запустить эту программу на обоих концах соединения.

Для оценки скорости пакетной передачи данных можно воспользоваться командой **ping**. У этой команды есть ключ **-i**, задающий интервал между пакетами и ключ **-s**, задающий размер пакета. Манипулируя этими ключами можно загрузить канал связи до “затопления” (flood), т.е. до ситуации, когда пакеты начнут теряться. Это и есть предельная пропускная способность сети на данный момент времени. Например, команда

**ping имя -i 0.01 -s 125000** загрузит 10Мбит/с более чем на 100%.

По параметру TTL возвращаемого эхо можно понять, на каком по счету от источника запросов маршрутизаторе происходит потеря пакетов.

Путь прохождения пакетов можно посмотреть командой traceroute.

## **Способы гарантирования заданной пропускной способности сети.**

В общем случае стек протоколов TCP/IP v.4, используемый на сегодняшний день, практически не содержит средств гарантирования качества сервиса (Quality of Service, QoS). Сети типа Ethernet тоже не предусматривают механизма обеспечения QoS. Однако, в IP сетях возможно ограничить скорость передачи, регулируя скорость ухода данных из буфера передачи маршрутизатора по какому-либо критерию ( IP адреса, поле TOS и т.п.).

Ограничить можно только скорость передачи, поэтому гарантировать полосу можно только между непосредственно соединенными маршрутизаторами. Естественно, общая полоса пропускания должна позволять это сделать, т.е. сумма полос, разделенных по группе критериев не должна превышать предельной скорости в канале.

## Ход работы.

Работа выполняется бригадой из 3-х человек. Двое занимаются посылкой тестовых последовательностей, третий – измеряет задержки, потери пакетов и уровень ошибок в физическом канале связи.

Для выполнения работы необходимы права суперпользователя на всех трех рабочих станциях.

Данный вариант работы должен выполняться в сети с коммутатором, в сети с концентратором или с другой общей средой передачи результаты будут недостоверными за счет взаимного влияния.

Дальше по тексту мы будем ссылаться на адреса трёх хостов. Host-1, Host-2 – машины, с которых посылаются тестовые последовательности, Host-3 – хост-”жертва”, над которым производится эксперимент.

1. Проверьте задержку в канале связи и уровень потерь при не загруженной сети. Для этого на Host-1 и Host-2 запустите команду: **sh>ping Host-3** и наблюдайте за ее работой пару минут, после чего прервите ее выполнение нажатием Ctrl-C. Запустите так же команду **sh>/sbin/ifconfig** и зафиксируйте количество коллизий и ошибок на интерфейсе eth0.
2. Запустите сервис ftpd на Host-3 выполнением следующей команды: **sh> /etc/rc.d/init.d/vsftpd start** Если команда не выполнилась, проверьте, установлен ли пакет vsftpd командой **sh>rpm -qa |grep ftp** и при необходимости установите нужный пакет. В директории /var/ftp/pub поместите файл подходящего объема (100Мб). На Host-1 запустите в терминале оболочку **mc** и введите следующую команду: **mc>cd <ftp://Host-3>** перейдите в директорию pub и запустите перекачку файла. В другом терминале запустите команду **sh>ping Host-3** и прервите ее в момент перед окончанием перекачивания файла. Во время перекачивания зафиксируйте скорость, которую индицирует оболочка **mc**. Для сети 100Мбит/с это чуть больше 10 Мбайт/с для полнодуплексной карты и портов коммутатора. Зафиксируйте результаты выполнения команды **ping**. Вы должны увидеть, что потери данных в канале не наблюдается, но задержка возросла примерно в 10 раз. Выполните команду **sh>/sbin/ifconfig** на Host-1 и Host-3 и зафиксируйте количество коллизий и ошибок на интерфейсе

- eth0. Количество коллизий может незначительно вырасти, но ошибок и потерь несущей наблюдаться не должно.
3. Повторите п. 2 одновременно на Host-1 и Host-2. Зафиксируйте скорость перекачки файла на обоих хостах и суммарную скорость. Количество коллизий на Host-3 может возрасти значительно, но ошибок появляться не должно.
  4. Вычислите необходимый для 100% загрузки порта размер пакета и интервал посылки пакетов. Запустите команду **ping** с этими параметрами:  
**sh>ping -i *интервал* -s *размер* Host-3**  
одновременно с Host-1 и Host-2. В отдельном терминале запустите “пробник”:  
**sh>ping Host-3**  
Очевидно, что вы загрузили порт на Host-3 на 200%. По истечении пары минут прервите “пробник”, затем прервите тестовый **ping**. Зафиксируйте процент потери пакетов “пробников” и среднюю задержку в канале. Зафиксируйте прирост коллизий и ошибок на всех трех хостах. Понятно, что при 200% загрузки катастрофически вырастет количество потерь пакетов, коллизий и ошибок.
  5. Повторите п. 4 для 190%, 180%, ... 10% загрузки, запуская тестовый **ping** с соответствующими параметрами. Зафиксируйте результаты.
  6. Постройте графическую зависимость процента потерь пакетов “пробника” и задержки от загрузки сети. Объясните полученные результаты.
  7. Подберите процент загрузки сети командой **ping**, при котором перекачка файла в соответствии с п.2 существенно замедляется и останавливается.

## Содержание отчета.

Отчет должен содержать распечатку результатов эксперимента по п.1-п.7 и выводы по каждому пункту, а так же графики по п.6 с пояснениями.

## Контрольные вопросы.

1. Какое время занимает один пакет данных в сети Ethernet 100BaseT?
2. Почему в некоммутируемых сетях Ethernet максимальная длина соединительного кабеля не должна превышать 100М?
3. При каком уровне загрузки сети Ethernet процент потерь возрастает до критического?
4. Сколько времени нужно наблюдать сеть, что бы оценить уровень потерь пакетов?
5. В каком случае в канале допустимы значительные (>250ms) задержки?
6. Как снизить загрузку сети служебными данными?
7. Какой средний процент “полезных” данных в сетевом трафике Вы наблюдали при перекачке файла по протоколу FTP?
8. Объясните, зачем в ходе экспериментов использовалось 3 хоста.
9. Имеется коммутатор на 16 портов 100BaseT с общей пропускной способностью коммутирующей матрицы 1.2 Гбит/с. Оцените качество данного коммутатора.
10. Поясните, почему длина кабеля в коммутируемом сегменте Ethernet 100BaseT может превышать установленные стандартом 100 метров при сохранении удовлетворительной работы сети. На что влияет избыточная длина кабеля?

# Лабораторная работа 3

## Изучение сетевых анализаторов tcpdump и Ethereal.

*Замечание: Для выполнения данной лабораторной работы необходимы права суперпользователя на локальной машине. Должны быть установлены пакеты tcpdump и ethereal. Последний требует установки графической оболочки XWindow и библиотек GNOME.*

### Цель работы.

Целью данной работы является освоение анализаторов сетевого трафика, получение навыков написания фильтров для анализаторов и ознакомление со стеком сетевых протоколов.

### Теоретические сведения.

#### Утилита tcpdump.

Утилита tcpdump предназначена для анализа сетевого трафика и входит в поставку всех POSIX систем. Эта утилита выводит заголовки пакетов, которые соответствуют *заданным критериям*, на сетевом интерфейсе, переведенном предварительно в режим приема всех пакетов (promiscuous mode). Критерии задаются в форме логического выражения. Например:

```
root@kid>tcpdump -i ed1 -vvv -X -e host kid.stu and host ics-76-3.stu

tcpdump: listening on ed1

13:55:29.052244 0:50:ba:57:91:80 0:2:44:3b:b4:b7 ip 98: kid.stu >
ics-76-3.stu: icmp: echo request (ttl 64, id 45630, len 84)

0x0000 4500 0054 b23e 0000 4001 390b c0a8 0701 E..T.>..@.9.....
0x0010 c0a8 070e 0800 49d9 8acc 001b 3130 3f40 .....I.....10?@
0x0020 c7cb 0000 0809 0a0b 0c0d 0e0f 1011 1213 .....
0x0030 1415 1617 1819 1a1b 1c1d 1e1f 2021 2223 .....!"#
0x0040 2425 2627 2829 2a2b 2c2d 2e2f 3031 3233 $%&'()*+,-./0123
0x0050 3435 45

13:55:29.052625 0:2:44:3b:b4:b7 0:50:ba:57:91:80 ip 98: ics-76-3.stu >
kid.stu: icmp: echo reply (ttl 128, id 1659, len 84)

0x0000 4500 0054 067b 0000 8001 a4ce c0a8 070e E..T.{.....
0x0010 c0a8 0701 0000 51d9 8acc 001b 3130 3f40 .....Q.....10?@
```

```
0x0020 c7cb 0000 0809 0a0b 0c0d 0e0f 1011 1213 .....
0x0030 1415 1617 1819 1a1b 1c1d 1e1f 2021 2223 .....!"#
0x0040 2425 2627 2829 2a2b 2c2d 2e2f 3031 3233 $%&'()*+,-./0123
0x0050 3435
```

Данная команда выводит на экран дамп пакетов между хостами kid.stu и ics-76-3.stu на интерфейсе ed1 хоста kid.stu в режиме расширенного вывода (-vvv) с печатью содержимого пакета (-X).

Логические выражения для критериев необходимы для того, что бы из общего сетевого трафика выделить только интересующие нас пакеты. Синтаксис логических выражений включает следующие ключевые слова:

**host** - IP адрес или DNS имя хоста

**net** - адрес сети, например

```
net 192.168.7, net 192.168.7.0 mask 255.255.255.224
```

**port** – номер порта (имеет смысл для протоколов TCP и UDP)

**proto** – тип протокола. Возможные типы: **ether, fddi, tr, ip, ip6, arp, rarp, decnet, lat, sca, mopr, mopdl, iso, esis, isis, icmp, icmp6, tcp and udp**. Например, **tcpdump tcp port 80**

**dir** – направление, возможные значения – **src** или **dst** . Например, **tcpdump src host kid.stu**.

Кроме того, можно использовать адреса несущей сети Ethernet:

```
tcpdump ether dst 00:02:44:5b:ee:9b
```

или IP сеть:

```
tcpdump net src 192.168.7.0/27
```

Более полную информацию о возможностях утилиты tcpdump можно получить из стандартной страницы помощи, дав команду **man tcpdump**.

## Сетевой анализатор **ethereal**.

Сетевой анализатор ethereal построен на той же библиотеке (libpcap), что и утилита tcpdump, но имеет удобный графический пользовательский интерфейс.

Общий вид окна сетевого анализатора приведен на рисунке 1. Для начала захвата пакетов необходимо зайти в меню Capture и выбрать Start. При запуске захвата без фильтра будут захвачены все пакеты, доступные на сетевом интерфейсе в режиме promiscuous mode. Обычно в сети присутствует довольно интенсивный трафик и, следовательно, найти интересующие пакеты среди всех захваченных необычайно трудно. Захваченными окажутся все пакеты, присутствующие на сетевом интерфейсе. Что там будет – зависит от типа сети - коммутированная сеть или сеть с общей средой передачи данных. Если ваш

сегмент сети Ethernet построен на концентраторе (hub), вы увидите все пакеты сегмента. Если сеть построена на базе коммутатора (switch), вы увидите только то, что попало в ваш порт, а собственно трафик к/от вашей сетевой карте плюс широковещательный трафик. Если вы наблюдаете трафик на интерфейсе маршрутизатора, то там все равно будет множество данных, проходящих к различным хостам. Поэтому нужно иметь возможность выделить для захвата только интересующий трафик. Этой цели служит пункт меню Capture Filters.

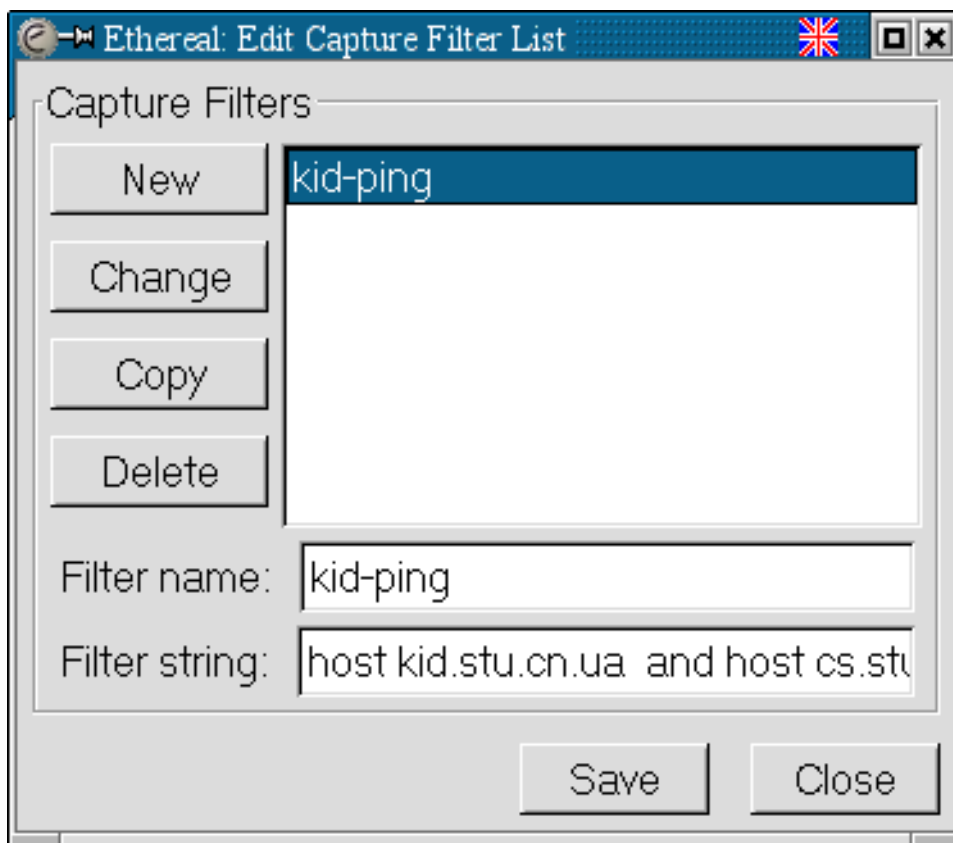


Рисунок 1 – Основное окно ethereal.

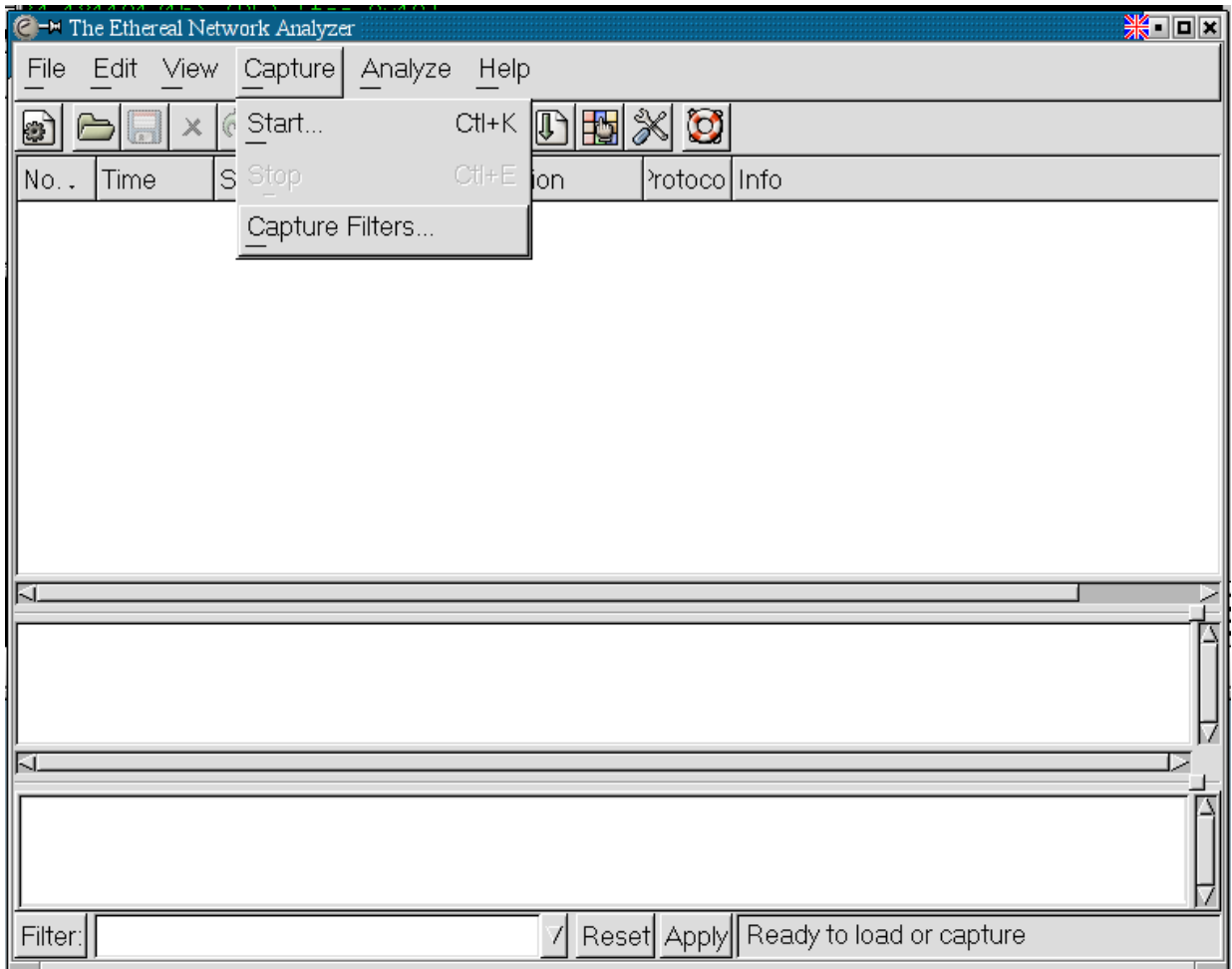


Рисунок 2 – Окно редактирования фильтров

Для того, что бы получить только интересующие пакеты, необходимо написать соответствующий фильтр. Поскольку программа ethereal использует библиотеку libpcap, ту же, на которой построена утилита tcpdump, синтаксис фильтров у ethereal такой же. Подсказку всегда можно получить, дав команду `man tcpdump`. Окно редактирования фильтров захвата приведено на рисунке 2. При выполнении лабораторной работы вам необходимо задать фильтр, который будет выделять только трафик между вашей рабочей станцией и сервером.

На этом рисунке выражение фильтра записано в поле Filter String и представляет собой строку `host kid.stu.cn.ua and host cs.stu.cn.ua`. Данный фильтр позволит захватывать трафик только между двумя указанными хостами, После того, как вы зададите соответствующий фильтр, можно стартовать процесс захвата пакетов.

Окно конфигурации захвата приведено на рисунке 3. Для удобства наблюдения за процессом захвата желательно отметить опции так, как показано на рисунке. Тогда вы сможете наблюдать захват пакетов в реальном масштабе времени.



get

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>501 Method Not Implemented</title>
</head><body>
<h1>Method Not Implemented</h1>
<p>get to /index.html not supported.<br />
</p>
<hr />
<address>Apache/2.0.48 (Unix) PHP/5.0.0a4-alexdupre Server at kid.stu.cn.ua
Port 80</address>
</body></html>
Connection closed by foreign host.
root@stalker>
```

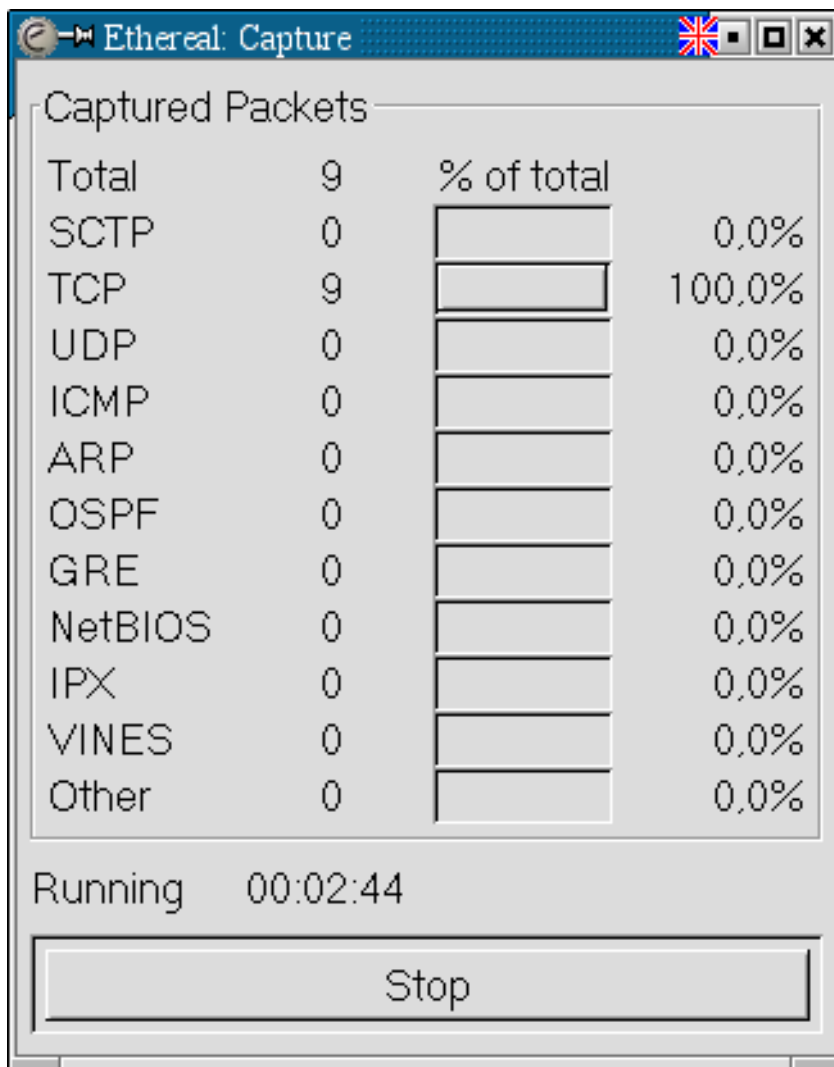


Рисунок 4 – Окно статистики захвата.

Данная команда посылает запрос к веб-серверу и выводит текст ответа на экран. Ниже изображено окно, показывающее процесс захвата пакетов в реальном времени.

Приводится статистика по протоколам для наглядной оценки процента загрузки сети различными протоколами. Данное окно можно использовать для анализа загрузки сети, если перед захватом не устанавливать фильтр.

После того, как вы произвели захват необходимых пакетов, можно остановить захват и разобраться, что мы увидели. На рисунке 4 приведено окно управления захватом. Просто нажмите кнопку “Stop”. В результате вы увидите окно с захваченными пакетами, показанное на следующем рисунке. Окно захваченных пакетов показывает все пакеты, захваченные в соответствии с указанным фильтром.

Верхняя часть окна показывает последовательность всех захваченных пакетов. Средняя часть показывает разобранный по протоколам пакет, на который указывает курсор в верхней части. Нижняя часть – это дамп указанного пакета в шестнадцатеричном (слева) и текстовом виде (справа).

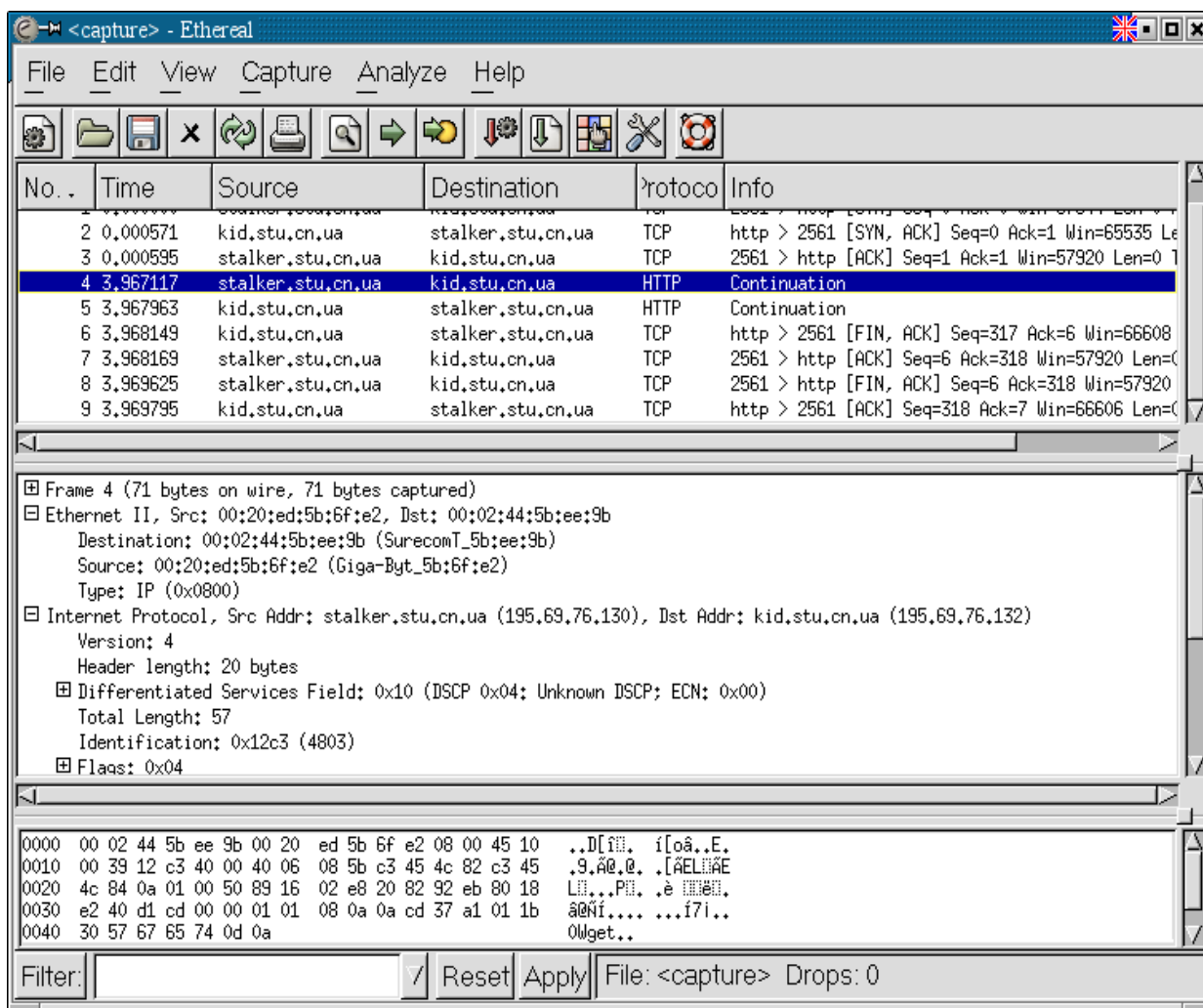


Рисунок 5 – Окно с захваченными пакетами.

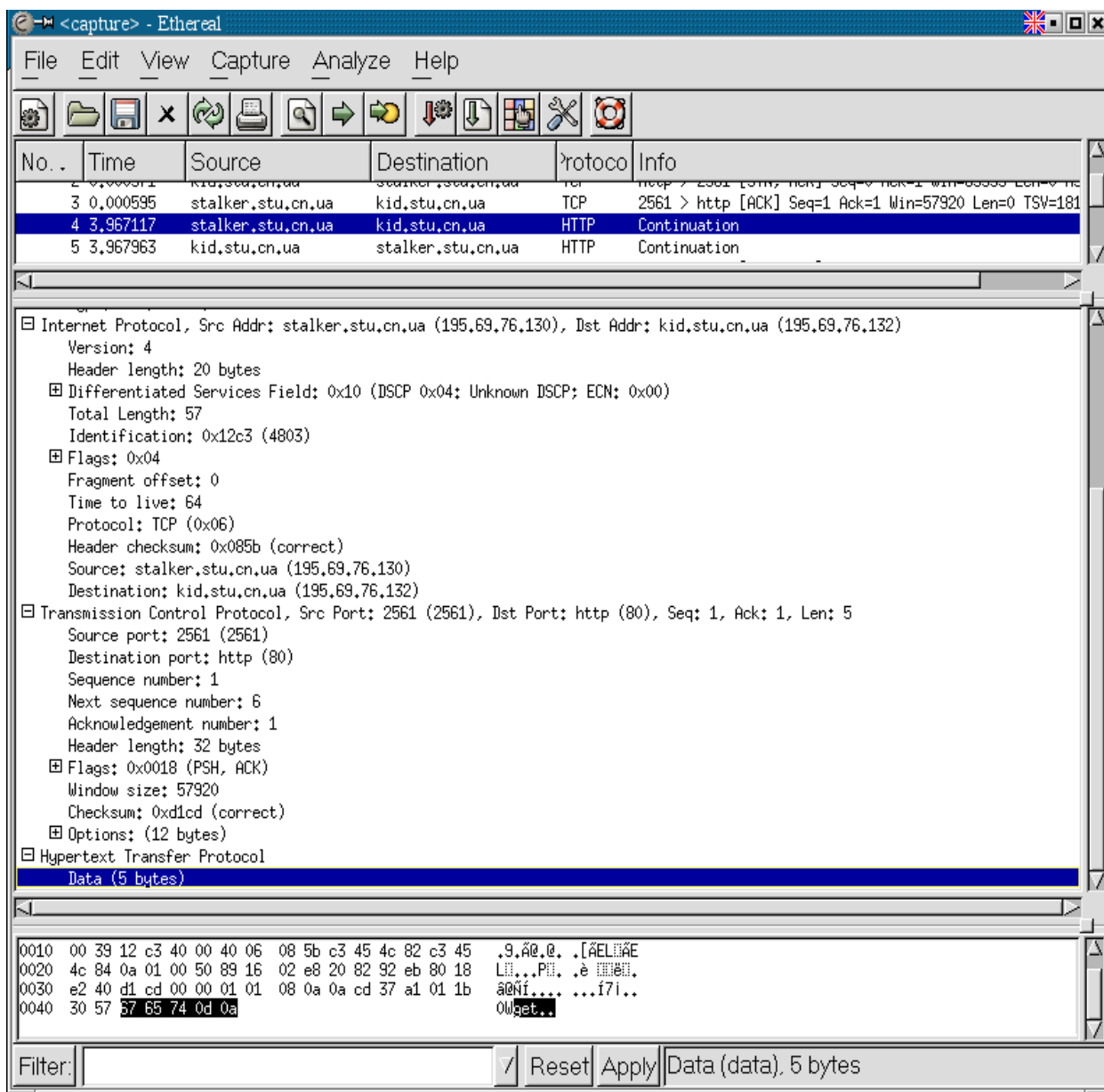


Рисунок 6 – Окно с захваченными пакетами.

На следующем рисунке в развернутом виде показан разбор пакета по уровням протоколов. Как видно из рисунка, курсор можно поставить на соответствующий уровень и в нижнем окне дампа будет отмечен соответствующий фрагмент пакета. Это очень удобно для подробного анализа обмена и отлавливания ошибок в бинарных протоколах обмена. Стоит заметить, что большинство протоколов в сетях TCP/IP используют текстовый формат команд, что значительно облегчает отладку программ в терминальном режиме и в режиме обмена, однако не обеспечивает совершенной никакой защиты информации при передаче. Современные протоколы используют криптографическую защиту данных, обеспечивающую очень высокую степень защиты от перехвата и изменения данных во время передачи по сети.

Вот собственно, и всё, что необходимо для начала. Остальные сведения вы получите в ходе выполнения следующих лабораторных работ.

## **Ход работы.**

1. Проведите захват пакетов утилитой **ethereal** без фильтра в течении нескольких минут и внимательно просмотрите результат. Отметьте, какие протоколы используются в сети.
2. Настройте фильтр на захват только широковещательных пакетов. Произведите захват в течении нескольких минут, рассмотрите результат.
3. Настройте фильтр на захват пакетов ICMP. Проверьте результат, включив захват пакетов и запустив со своего хоста команду ping на соседний хост.
4. Не меняя настроек фильтра, запустите захват пакетов и запустите ping на другом хосте, направлены на третий хост в вашем сегменте. Удалось ли Вам захватить эти пакеты? Объясните результат.
5. Запустите утилиту tcpdump с пустым фильтром с перенаправлением пакетов в файл на несколько минут с ключами, обеспечивающими расшифровку пакетов. Поясните результат захвата пакетов.

## **Содержание отчета.**

Отчет должен содержать последовательность скриншотов по ходу выполнения работы и соответствующие комментарии.

## **Контрольные вопросы.**

1. Почему Вы не видите все пакеты, проходящие в данном сегменте Ethernet?
2. В каком случае хост может видеть все пакеты в данном сегменте Ethernet?
3. Какие пакеты будут видны наблюдающему хосту в коммутируемом сегменте Ethernet?
4. Как обеспечить захват всех пакетов, приходящих в данный сегмент сети и уходящих из него?
5. Что такое “зеркальный” порт коммутатора, и каково его назначение?
6. Какой параметр коммутатора отвечает за общую пропускную способность?

# **Лабораторная работа 4. Изучение сетевого протокола TCP и протокола уровня приложений telnet.**

## **Теоретические сведения.**

Протокол TCP является транспортным протоколом с гарантированной доставкой данных, с установлением соединения и повторной передачей потерянных сегментов.

Установление соединения происходит при помощи механизма т.н. троекратного рукопожатия (three way handshake).

Хост А, инициатор соединения, посылает сегмент без данных с установленным флагом SYN. Иницирует соединения программа - клиент. Порт назначения определяется жёстко либо как хорошо известный сервис (Well Known Service), например web-сервис обычно имеет порт 80, либо задается пользователем. Порт источника обычно выделяется системой из пула свободных не привилегированных портов (>1023).

В ответ хост В посылает сегмент без данных с установленными флагами SYN,ACK. При этом, в зависимости от режима работы серверного сокета, возможна замена фиксированного порта на динамический в качестве исходящего.

Хост А, приняв описанный выше пакет, сигнализирует о готовности к обмену данными, посылая сегмент без данных с установленным флагом ACK. После этого сокет готов к двустороннему обмену данными.

Разрыв соединения происходит аналогично с использованием флага FIN.

Протокол TELNET является протоколом уровня приложений, предназначенным для удалённого доступа по сети к текстовому терминалу. При инициализации соединения посылаются команды и клиентом, и сервером, обеспечивающие согласование возможностей пользовательского терминала и установку необходимых переменных окружения на сервере.

## **Ход работы.**

1. Установите сервис telnet на соседней рабочей станции (назовём ее host2). Для этого в директории /etc/xinetd.d в файле telnet строку "disabled = yes" замените строкой "disabled = no" и перезапустите сервис xinetd следующей командой:

```
/etc/rc.d/init.d/xinetd restart.
```

2. Проверьте работоспособность сервиса:

telnet host2. Вы должны получить терминальный доступ к машине host2.

3. Запустите на своей рабочей станции (далее -host1) программу захвата пакетов ethereal с фильтром, ограничивающим захват трафика между host 1 и host2 по протоколу tcp, что бы в захваченные пакеты не попадал "мусор".

Стартуйте захват пакетов.

4. В терминальном окошке запустите сессию telnet на host2. Войдите в систему, введя логин и пароль. Выйдите из системы командой logout.

Поместите протокол сессии в файл для отчёта.

5. Остановите захват пакетов и сохраните результат в формате tcpdump. Сохранённый результат захвата подайте на вход утилиты tcpdump и результат разбора перенаправьте файл для отчёта.

6. Изучая параллельно протокол сессии telnet, результаты захвата пакетов в окне ethereal и в файле с разобранными пакетами, найдите и прокомментируйте:

установление и разрыв соединения по протоколу TCP с учётом флагов;

пакеты, содержащие логин и пароль пользователя;

7. Покажите механизм инкапсуляции данных на примере пакета с данными от host2.

## **Содержание отчёта.**

Отчет должен содержать протокол telnet сессии и распечатку дампа tcpdump. Пакеты, содержащие существенную информацию для данной лабораторной работы должны быть тщательно прокомментированы. Кроме того, отчёт должен содержать выводы, описывающие процесс изучения протоколов TCP и telnet.

# **Лабораторная работа 5. Изучение сетевого протокола UDP и протокола урона приложений DNS.**

## **Теоретические сведения.**

### **Протокол UDP**

Протокол UDP является простейшим транспортным протоколом без гарантии доставки данных и без установления соединения. Данный протокол обеспечивает

мультиплексирование данных между приложениями при помощи поля port а так же контроль правильности данных при помощи поля checksum. Данный протокол используется для обмена короткими структурированными данными в режиме "запрос-ответ" а так же для посылки широковещательных сообщений. По сравнению с протоколом TCP этот протокол обеспечивает большее быстродействие, поскольку не имеет затрат на установку и разрыв соединения. Удобно так же применение протокола UDP для случаев специального транспорта, когда транспорт TCP по каким-либо соображениям не устраивает разработчика. Однако, следует отметить, что механизмы подтверждения и сборки потока в этом случае должны обеспечиваться приложением.

## DNS

Служба доменных имён DNS является основной системной службой в сетях TCP/IP, поскольку эта служба обеспечивает разрешение символьных имён в IP адреса и наоборот. Каждое приложение, использующее сетевые функции, обращается к базовой системной библиотеке libc, частью которой является так называемый резольвер (resolver). Резольвер имеет свой файл конфигурации /etc/resolv.conf, в котором описаны ближайшие сервера имён и порядок подстановки суффиксов.

```
order bind,hosts
```

```
search stu stu.cn.ua
```

```
nameserver 192.168.0.10
```

```
nameserver 192.168.0.14
```

В сети должно быть как минимум два сервера имён для обеспечения бесперебойного разрешения имён. Обращение резольвера происходит сначала к первому серверу, и если ответ не получен в течение короткого времени, ко второму. Если сервер не может самостоятельно отработать запрос, он обращается к серверам домена корневого домена ". " и производит поиск сервера, способного обработать запрос. Полученный ответ перенаправляется клиенту и кешируется на сервере для ускорения последующих ответов.

Подробнее см. лабораторную работу по настройке сервера имён.

## Ход работы.

1. Запустите анализатор ethereal с фильтром, настроенным на отслеживание трафика от вашего хоста до сервера DNS (далее -dns\_host) и до сервера www (далее - www\_host).

2. Проверьте работоспособность фильтра командами

```
ping www_host
```

```
ping dns_host
```

Анализатор должен показать захваченные пакеты.

3. Перестаруйте захват пакетов.
4. Запустите программу просмотра web на хост `www_host`.
5. Сохраните захваченные пакеты в формате `libpcap`.
6. Запустите анализатор `tcpdump` для разбора сохранённого файла со следующими ключами:

```
tcpdump -vvv -X -r файл_пакетов >файл_разбора
```

Данный файл будет содержать разобранные пакеты обращения браузера к веб-страничке и к серверу DNS.

7. Используя оба анализатора, отследите, как происходили обращения к серверам и прокомментируйте в файле все пакеты.
9. Запустите новую сессию захвата пакетов.

10. Дайте команды

```
host www.yahoo.com
```

```
host 193.193.193.100
```

и повторите описанную выше процедуру для захваченных пакетов.

11. Подробно разберите все запросы к серверу DNS.

## **Содержание отчёта.**

Отчет должен содержать прокомментированные файлы с разобранными результатами захвата пакетов. Пакеты, не представляющие особого интереса в ключе данной работы можно удалить из файла дампа.

## **Контрольные вопросы.**

1. Почему для службы DNS используется протокол UDP?
2. Какое поле заголовка UDP обеспечивает мультиплексирование пакетов между приложениями?
3. Как программы определяют, где находится ближайший сервер имён?
5. Какие дополнительные параметры передаются в ответе сервера имён?

6. Какого типа запросы обрабатывались при просмотре веб-странички?

7. Какой запрос использовался для определения имени хоста по его адресу?

## **Рекомендованная литература:**

1. UNIX. Пособие системного администратора. / Пер. с англ. Под ред. д – К.: ВHV, 2002 г.

2. Э. Таненбаум. Компьютерные сети. / Пер. с англ. Под ред. д – К.: ВHV, 2002 г.

3. В.Г. Олифер, Н.А. Олифер. Компьютерные сети. Принципы, технологии, протоколы. СПб., Питер, 2001-672с.:ил, ШЫИТ 5-8046-0133-4

4. <http://www.rfc-editor.org> RFC center