

Лабораторная работа №1

Освоение средств захвата и анализа сетевого трафика.

Рекомендуем выполнять захват пакетов используя самую распространенную программу-анализатор трафика - **Wireshark**.

Wireshark - программа для анализа сетевых протоколов, которая широко используется для захвата сетевых пакетов. Программа распространяется бесплатно.

После запуска вы увидите основное окно программы. Зайдите в меню **Capture > Options** или нажмите **Ctrl+K** для дальнейшей настройки параметров программы

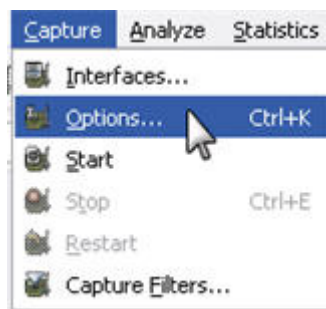


Рис.1.

Вы увидите окно **Capture Options**.

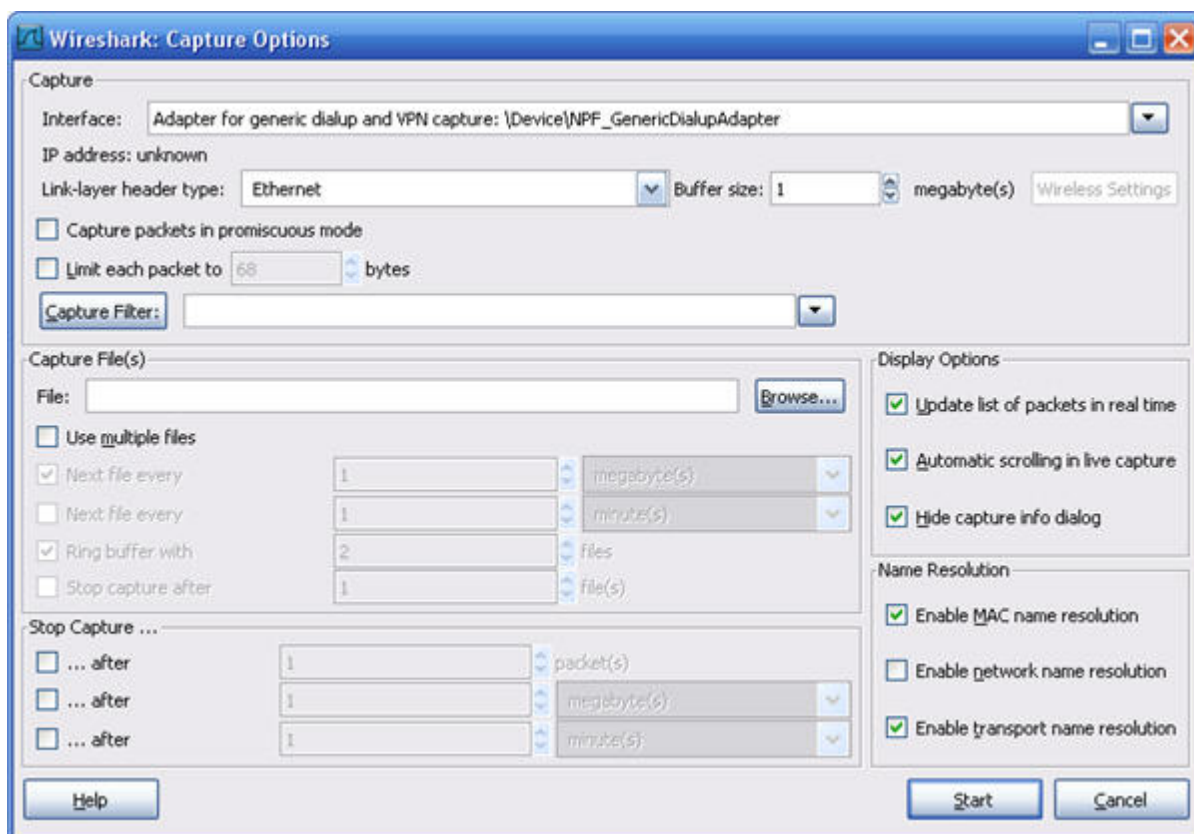


Рис.2. – Окно Capture Options

В разделе **Capture** в поле **Interface** выберите из выпадающего списка Ethernet-адаптер, через который будет происходить захват пакетов.

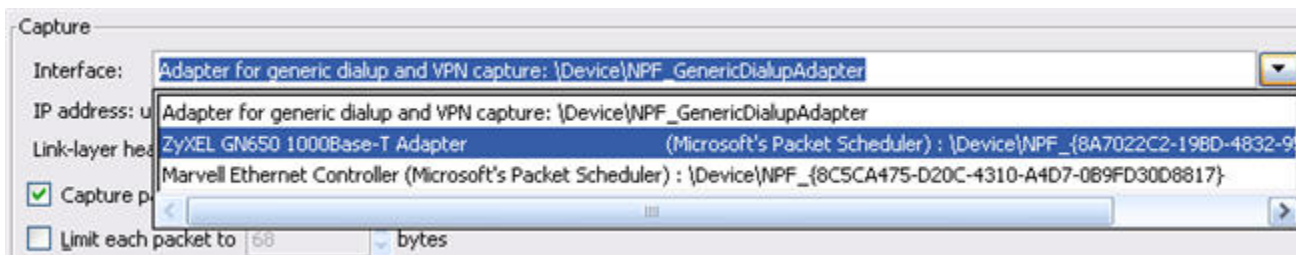
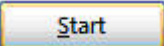


Рис.3.

Затем нажмите кнопку  для начала захвата пакетов.

Подключите ваш компьютер к LAN-порту вашего устройства. Подождите пока ваш компьютер получит IP-адрес от устройства (если включен DHCP-сервер), или установите на компьютере вручную IP-адрес из той же подсети, что и LAN IP-адрес устройства.

В программе Wireshark вы увидите все захваченные пакеты, которые присутствуют на LAN-порту устройства.

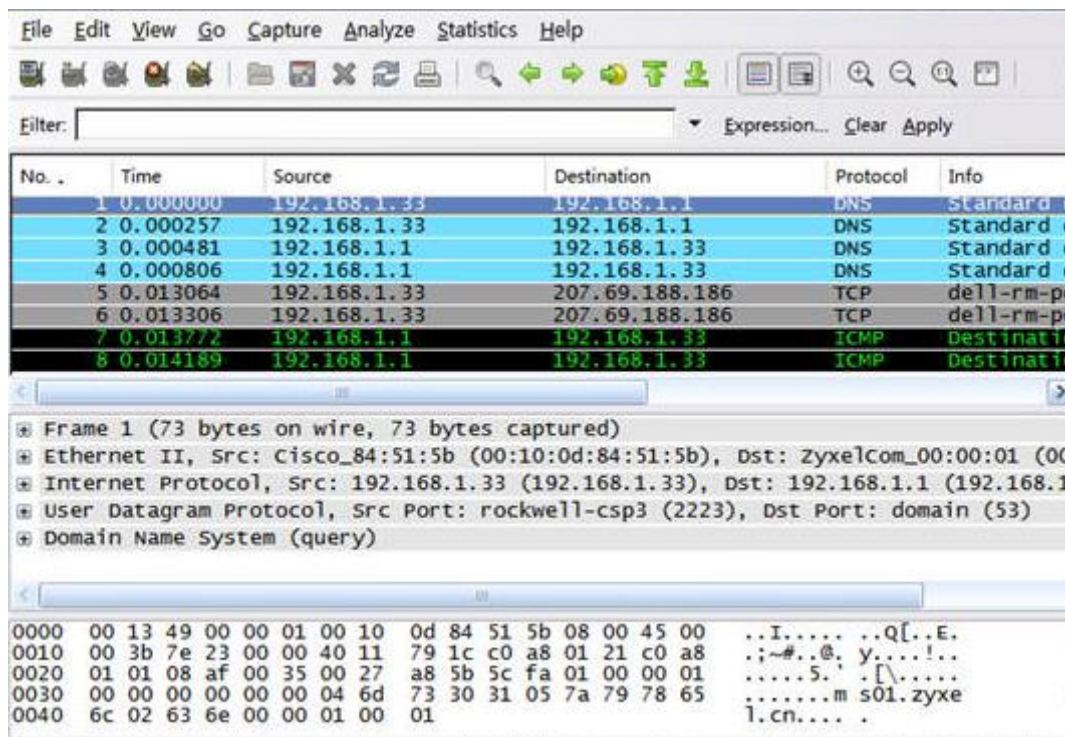



Рис.4.

Для завершения захвата пакетов нажмите кнопку  на панели инструментов программы Wireshark.

Зайдите в меню **File > Save As...** для сохранения захваченных данных в файл (Рис.5).

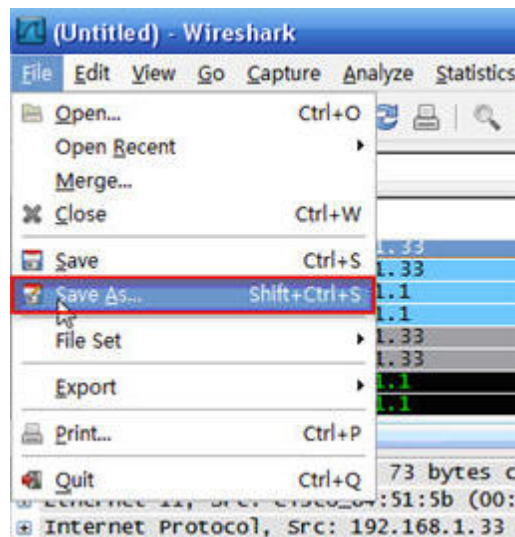



Рис.5.

Выберите местоположение, введите имя файла и нажмите кнопку  для сохранения пакетов.

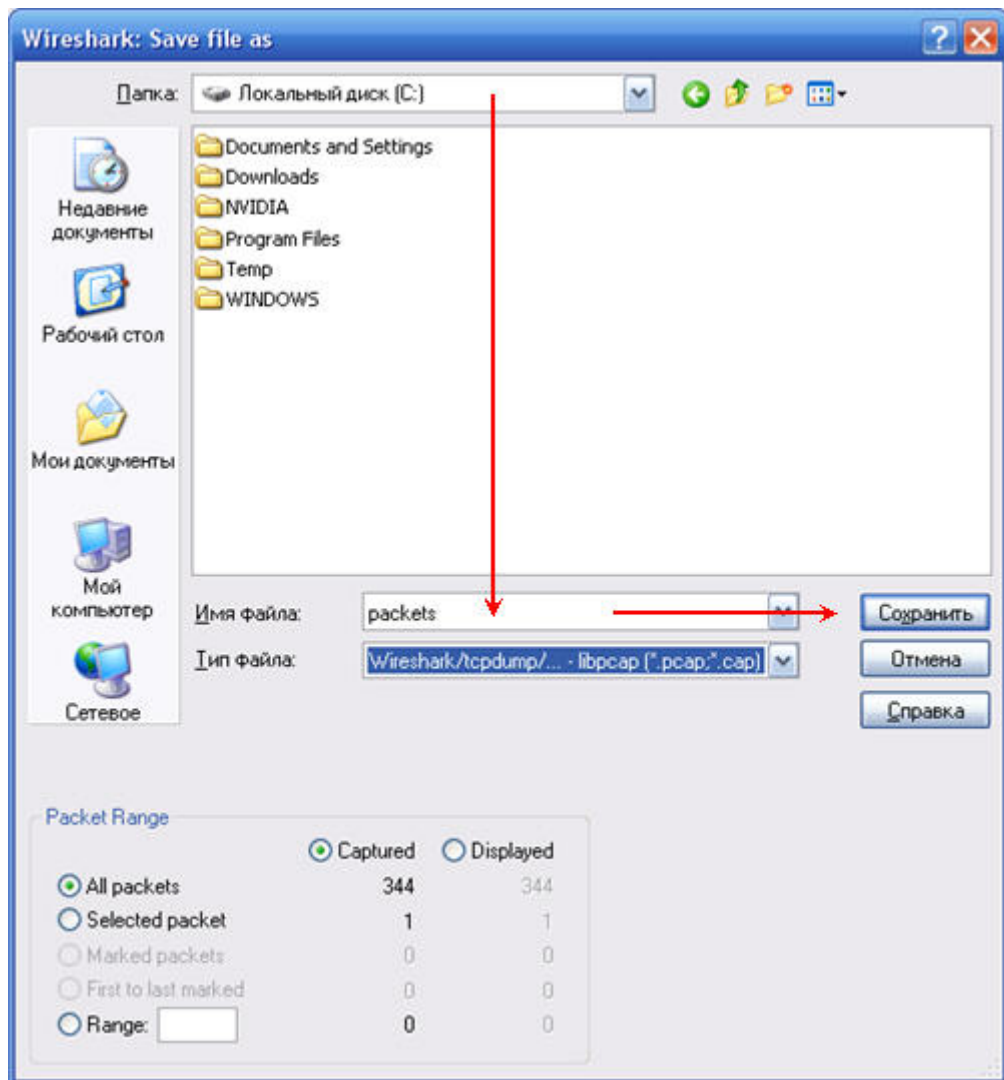
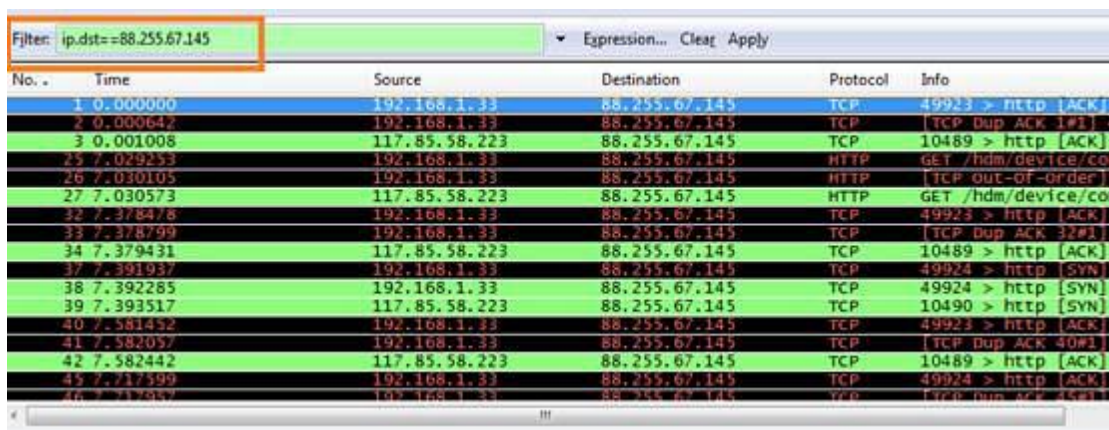


Рис.6.

Для удобства поиска/просмотра информации о нужных пакетах в программе Wireshark можно отфильтровать захваченные пакеты по IP-адресу или номеру порта.

Приведем примеры:

Если вы хотите сделать фильтрацию захваченных пакетов по IP-адресу назначения 88.255.67.145, в поле **Filter** укажите правило фильтра **ip.dst==88.255.67.145**

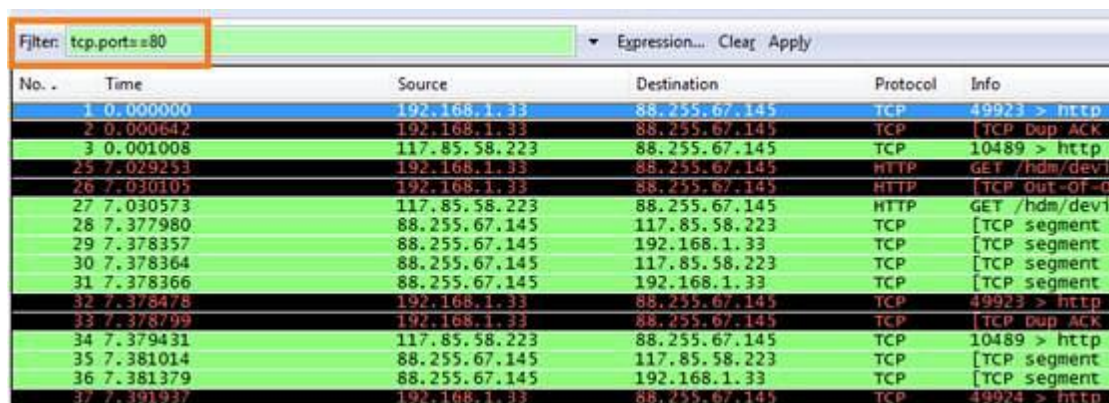


The screenshot shows the Wireshark interface with a filter applied to the packet list pane. The filter text is 'ip.dst==88.255.67.145'. The packet list pane displays a list of captured packets, all of which have the destination IP address 88.255.67.145. The columns shown are No., Time, Source, Destination, Protocol, and Info.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.1.33	88.255.67.145	TCP	49923 > http [ACK]
2	0.000642	192.168.1.33	88.255.67.145	TCP	[TCP Dup ACK 1#1]
3	0.001008	117.85.58.223	88.255.67.145	TCP	10489 > http [ACK]
25	7.029253	192.168.1.33	88.255.67.145	HTTP	GET /hdm/device/co
26	7.030105	192.168.1.33	88.255.67.145	HTTP	[TCP out-of-order]
27	7.030573	117.85.58.223	88.255.67.145	HTTP	GET /hdm/device/co
32	7.378478	192.168.1.33	88.255.67.145	TCP	49923 > http [ACK]
33	7.378799	192.168.1.33	88.255.67.145	TCP	[TCP Dup ACK 32#1]
34	7.379431	117.85.58.223	88.255.67.145	TCP	10489 > http [ACK]
37	7.391937	192.168.1.33	88.255.67.145	TCP	49924 > http [SYN]
38	7.392285	192.168.1.33	88.255.67.145	TCP	49924 > http [SYN]
39	7.393517	117.85.58.223	88.255.67.145	TCP	10490 > http [SYN]
40	7.581452	192.168.1.33	88.255.67.145	TCP	49923 > http [ACK]
41	7.582057	192.168.1.33	88.255.67.145	TCP	[TCP Dup ACK 40#1]
42	7.582442	117.85.58.223	88.255.67.145	TCP	10489 > http [ACK]
45	7.717599	192.168.1.33	88.255.67.145	TCP	49924 > http [ACK]
46	7.717657	192.168.1.33	88.255.67.145	TCP	[TCP Dup ACK 45#1]

Рис.7.

Если вы хотите сделать фильтрацию захваченных пакетов по определенному порту TCP (например, по 80 порту), в поле **Filter** укажите правило фильтра **tcp.port==80**



The screenshot shows the Wireshark interface with a filter applied to the packet list pane. The filter text is 'tcp.port==80'. The packet list pane displays a list of captured packets, all of which have the destination port 80. The columns shown are No., Time, Source, Destination, Protocol, and Info.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.1.33	88.255.67.145	TCP	49923 > http
2	0.000642	192.168.1.33	88.255.67.145	TCP	[TCP Dup ACK]
3	0.001008	117.85.58.223	88.255.67.145	TCP	10489 > http
25	7.029253	192.168.1.33	88.255.67.145	HTTP	GET /hdm/devi
26	7.030105	192.168.1.33	88.255.67.145	HTTP	[TCP out-of-o
27	7.030573	117.85.58.223	88.255.67.145	HTTP	GET /hdm/devi
28	7.377980	88.255.67.145	117.85.58.223	TCP	[TCP segment
29	7.378357	88.255.67.145	192.168.1.33	TCP	[TCP segment
30	7.378364	88.255.67.145	117.85.58.223	TCP	[TCP segment
31	7.378366	88.255.67.145	192.168.1.33	TCP	[TCP segment
32	7.378478	192.168.1.33	88.255.67.145	TCP	49923 > http
33	7.378799	192.168.1.33	88.255.67.145	TCP	[TCP Dup ACK]
34	7.379431	117.85.58.223	88.255.67.145	TCP	10489 > http
35	7.381014	88.255.67.145	117.85.58.223	TCP	[TCP segment
36	7.381379	88.255.67.145	192.168.1.33	TCP	[TCP segment
37	7.391937	192.168.1.33	88.255.67.145	TCP	49924 > http

Рис.8.

Если вы хотите сделать фильтрацию захваченных пакетов по двум определенным IP-адресам (например, по IP-адресам 88.255.67.145 и 117.85.53.223), в поле **Filter** укажите правило фильтра **ip.addr==88.255.67.145 and ip.addr==117.85.53.223**

Filter: ip.addr==88.255.67.145 and ip.addr==117.85.58.223		Expression... Clear Apply			
No. .	Time	Source	Destination	Protocol	Info
3	0.001008	117.85.58.223	88.255.67.145	TCP	10489 > http [ACK]
27	7.030573	117.85.58.223	88.255.67.145	HTTP	GET /hdm/device/cc
28	7.377980	88.255.67.145	117.85.58.223	TCP	[TCP segment of a
30	7.378364	88.255.67.145	117.85.58.223	TCP	[TCP segment of a
34	7.379431	117.85.58.223	88.255.67.145	TCP	10489 > http [ACK]
35	7.381014	88.255.67.145	117.85.58.223	TCP	[TCP segment of a
39	7.393517	117.85.58.223	88.255.67.145	TCP	10490 > http [SYN]
42	7.582442	117.85.58.223	88.255.67.145	TCP	10489 > http [ACK]
43	7.717090	88.255.67.145	117.85.58.223	TCP	http > 10490 [SYN]
47	7.718535	117.85.58.223	88.255.67.145	TCP	10490 > http [ACK]
50	7.719844	117.85.58.223	88.255.67.145	HTTP	GET /hdm/js/global
51	7.906171	88.255.67.145	117.85.58.223	HTTP	HTTP/1.1 200 OK (
53	8.058296	88.255.67.145	117.85.58.223	TCP	[TCP segment of a
57	8.112768	117.85.58.223	88.255.67.145	TCP	10489 > http [ACK]
60	8.268805	117.85.58.223	88.255.67.145	TCP	10490 > http [ACK]
61	8.588570	88.255.67.145	117.85.58.223	HTTP	HTTP/1.1 200 OK (

Рис.9.

Форматы кадров Ethernet

Стандарт 802.3 определяет восемь полей заголовка:

- *Поле преамбулы* состоит из семи байтов синхронизирующих данных. Каждый байт содержит одну и ту же последовательность битов - 10101010. При манчестерском кодировании эта комбинация представляется в физической среде периодическим волновым сигналом. Преамбула используется для того, чтобы дать время и возможность схемам приемопередатчиков (transceiver) прийти в устойчивый синхронизм с принимаемыми тактовыми сигналами.
- *Начальный ограничитель кадра* состоит из одного байта с набором битов 10101011. Появление этой комбинации является указанием на предстоящий прием кадра.
- *Адрес получателя* - может быть длиной 2 или 6 байтов (MAC-адрес получателя). Первый бит адреса получателя - это признак того, является адрес индивидуальным или групповым: если 0, то адрес указывает на определенную станцию, если 1, то это групповой адрес нескольких (возможно всех) станций сети. При широковещательной адресации все биты поля адреса устанавливаются в 1. Общепринятым является использование 6-байтовых адресов.
- *Адрес отправителя* - 2-х или 6-ти байтовое поле, содержащее адрес станции отправителя. Первый бит - всегда имеет значение 0.
- *Двухбайтовое поле длины* определяет длину поля данных в кадре.
- *Поле данных* может содержать от 0 до 1500 байт. Но если длина поля меньше 46 байт, то используется следующее поле - поле заполнения, чтобы дополнить кадр до минимально допустимой длины.
- *Поле заполнения* состоит из такого количества байтов заполнителей, которое обеспечивает определенную минимальную длину поля данных (46 байт). Это обеспечивает корректную работу механизма обнаружения коллизий. Если длина поля данных достаточна, то поле заполнения в кадре не появляется.
- *Поле контрольной суммы* - 4 байта, содержащие значение, которое вычисляется по определенному алгоритму (полиному CRC-32). После получения кадра рабочая станция выполняет собственное вычисление контрольной суммы для этого кадра, сравнивает полученное значение со значением поля контрольной суммы и, таким образом, определяет, не искажен ли полученный кадр.

Кадр 802.3 является кадром MAC-подуровня, в соответствии со стандартом 802.2 в его поле данных вкладывается кадр подуровня LLC с удаленными флагами начала и конца кадра. Формат кадра LLC был описан выше.

Результирующий кадр 802.3/LLC изображен в левой части рисунка 1. Так как кадр LLC имеет заголовок длиной 3 байта, то максимальный размер поля данных уменьшается до 1497 байт.

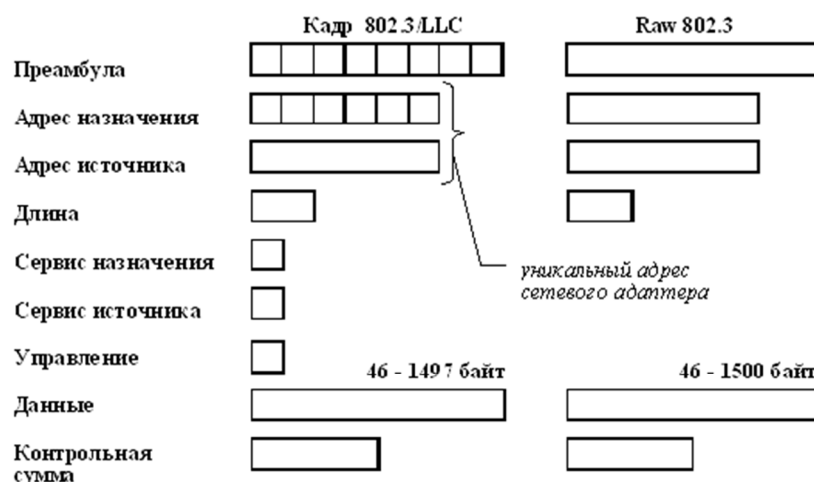


Рис. 10. Форматы кадров Ethernet

Справа на этом рисунке приведен кадр, который называют кадром Raw 802.3 (то есть "грубый" вариант 802.3) или же кадром Novell 802.3. Из рисунка видно, что это кадр MAC-подуровня стандарта 802.3, но без вложенного кадра подуровня LLC. Компания Novell долгое время не использовала служебные поля кадра LLC в своей операционной системе NetWare из-за отсутствия необходимости идентифицировать тип информации, вложенной в поле данных - там всегда находился пакет протокола IPX, долгое время бывшего единственным протоколом сетевого уровня в ОС NetWare.

Теперь, когда необходимость идентификации протокола верхнего уровня появилась, компания Novell стала использовать возможность инкапсуляции в кадр MAC-подуровня кадра LLC, то есть использовать стандартные кадры 802.3/LLC. Такой кадр компания обозначает теперь в своих операционных системах как кадр 802.2, хотя он является комбинацией заголовков 802.3 и 802.2.

Кадр стандарта Ethernet DIX, называемый также кадром Ethernet II, похож на кадр Raw 802.3 тем, что он также не использует заголовки подуровня LLC, но отличается тем, что на месте поля длины в нем определено поле типа протокола (поле Type). Это поле предназначено для тех же целей, что и поля DSAP и SSAP кадра LLC - для указания типа протокола верхнего уровня, вложившего свой пакет в поле данных этого кадра. Для кодирования типа протокола используются значения, превышающие значение максимальной длины поля данных, равное 1500, поэтому кадры Ethernet II и 802.3 легко различимы.

Еще одним популярным форматом кадра является кадр Ethernet SNAP (SNAP - SubNetwork Access Protocol, протокол доступа к подсетям). Кадр Ethernet SNAP определен в стандарте 802.2H и представляет собой расширение кадра 802.3 путем введения дополнительного поля идентификатора организации, которое может использоваться для ограничения доступа к сети компьютеров других организаций.

В таблице 2 приведены данные о том, какие типы кадров Ethernet обычно поддерживают реализации популярных протоколов сетевого уровня.

Команды

netstat

Команда используется для показа состояния сети.

Команда **netstat** имеет следующий синтаксис:

netstat [-AaLnSW] [-f protocol_family] [-p protocol] [-M core] [-N system]

Команда **netstat** показывает содержимое различных структур данных, связанных с сетью, в различных форматах в зависимости от указанных опций. *Первая форма* команды показывает список активных сокетов (sockets) для каждого протокола. *Вторая форма* выбирает одну из нескольких других сетевых структур данных. *Третья форма* показывает динамическую статистику пересылки пакетов по сконфигурированным сетевым интерфейсам; аргумент интервал задает, сколько секунд собирается информация между последовательными показами.

-p имя_протокола Ограничить показ статистики или адресов управляющих блоков только протоколом с указанным именем_протокола, например, tcp.

Опции:

<i>-a</i>	– показывать состояние всех сокетов; обычно сокет, используемый серверными процессами, не показывается.
<i>-A</i>	– показывать адреса любых управляющих блоков протокола, связанных с сокетами; используется для отладки.
<i>-i</i>	– показывать состояние автоматически сконфигурированных (auto-configured) интерфейсов. Интерфейсы, статически сконфигурированные в системе, но не найденные во время загрузки, не показываются.
<i>-n</i>	– показывать сетевые адреса как числа. netstat обычно показывает адреса как символы. Эту опцию можно использовать с любым форматом показа.
<i>-r</i>	– показать таблицы маршрутизации. При использовании с опцией <i>-s</i> , показывает статистику маршрутизации.
<i>-s</i>	– показать статистическую информацию по протоколам. При использовании с опцией <i>-r</i> , показывает статистику маршрутизации.
<i>-f</i> <i>семейство_адресов</i>	– ограничить показ статистики или адресов управляющих блоков только указанным семейством_адресов, в качестве которого можно указывать: <i>inet</i> Для семейства адресов AF_INET <i>unix</i> Для семейства адресов AF_UNIX
<i>-I интерфейс</i>	– выделить информацию об указанном интерфейсе в отдельный столбец; по умолчанию (для третьей формы команды) используется интерфейс с наибольшим объемом переданной информации с момента последней перезагрузки системы. В качестве интерфейса можно указывать любой из интерфейсов, перечисленных в файле конфигурации системы, например, emd1 или lo0.
<i>-p имя_протокола</i>	– Ограничить показ статистики или адресов управляющих блоков только протоколом с указанным именем_протокола, например, tc

ping

Команда используется для отправки пакетов ICMP ECHO_REQUEST сетевым хостам.

Команда **ping** имеет следующий синтаксис:

ping [-AaDdfnoQqRrv] [-c число_пакетов] [-i секунды] [-l preload] [-M mask | time]
[-m ttl] [-P policy] [-p pattern] [-S src_addr] [-s packet_size]

```
[-t timeout] [-z tos] host
ping [-AaDdfLnoQqRrv] [-c число_пакетов] [-I iface] [-i секунд] [-l preload]
[-M mask | time] [-m ttl] [-P policy] [-p pattern] [-S src_addr]
[-s packetsize] [-T ttl] [-t timeout] [-z tos] mcast-group
```

Команда **ping** использует датаграмму ECHO_REQUEST протокола ICMP, чтобы вызвать ответ ICMP ECHO_RESPONSE указанного хоста или сетевого шлюза. Если хост отвечает, **ping** выдает сообщение, что хост включен (хост is alive), в стандартный выходной поток.

Для проверки наличия хоста в сети достаточно ввести команду **ping** с аргументом - именем или адресом хоста:

arp

Команда **arp** отображает ARP-таблицу данного хоста. С помощью параметра *-i* можно специфицировать сетевой интерфейс, информация о котором интересует.

ifconfig

Команда используется для настройки сетевых интерфейсов

Команда **ifconfig** имеет следующий синтаксис:

```
ifconfig [-L] [-m] interface [create] [address_family] [address [dest_address]] [parameters]
ifconfig interface destroy
ifconfig -a [-L] [-d] [-m] [-u] [address_family]
ifconfig -l [-d] [-u] [address_family]
ifconfig [-L] [-d] [-m] [-u] [-C]
```

Команда **ifconfig** используется для настройки сетевых интерфейсов. Команда должна использоваться при загрузке системы для настройки адресов каждого сетевого интерфейса, а также может использоваться после загрузки для изменения параметров сетевых интерфейсов. Если команда введена без аргументов, **ifconfig** выдает информацию о состоянии активных интерфейсов. Если в качестве аргумента указан какой-либо интерфейс, то выдается информация только о состоянии этого интерфейса; если указан один аргумент -а, выдается информация о состоянии всех интерфейсов, даже отключенных.

traceroute

Команда **traceroute** служит для отладки сетевых соединений посредством построения маршрута следования пакетов к хосту назначения. Для этой команды также работает параметр *-n*, при использовании которого IP-адреса не будут заменяться символьными именами хостов.

route

Эта команда используется для просмотра и изменения таблицы маршрутизации хоста. Для этой команды также работает параметр *-n*, при использовании которого IP-адреса не будут заменяться символьными именами хостов.

Пример обычной таблицы маршрутизации для отдельного компьютера в сети:

```
desktop ~ # route -n
Kernel IP routing table
```


Destination	Gateway	Genmask	Flags	Metric	Ref	Uselface
192.168.5.0	0.0.0.0	255.255.255.0	U	0	0	eth1
127.0.0.0	0.0.0.0	255.0.0.0	U	0	0	lo
0.0.0.0	192.168.5.254	0.0.0.0	UG	0	0	eth1

Особый интерес представляет адрес 0.0.0.0, который соответствует хосту назначения по умолчанию.

Для добавления нового маршрута к определённому хосту используются параметры *add* и *host*:

```
desktop ~ # routeadd -host 192.168.0.1 eth0
```

Эта команда создаёт новую строку в таблице маршрутизации, согласно которой все пакеты к узлу 192.168.0.1 должны отправляться в сетевой интерфейс eth0.

Также можно добавлять шлюз для отправки пакетов в определённую сеть или к хосту:

```
desktop ~ # routeadd -host 192.168.0.1 eth0
```

Таким образом, все пакеты для сети 192.168.1.0 будут направляться на узел 192.168.0.5.

Аналогично, маршруты удаляются параметром *del* с указанием всей информации о маршруте:

```
desktop ~ # route del default gw 192.168.0.1
```

Эта команда удаляет маршрут по умолчанию через хост 192.168.0.1.

ЗАДАНИЕ

В качестве задания будет предложено захватить и подробно объяснить сетевой трафик, порождаемый различными сетевыми утилитами и службами. Например, трафик порождаемый **ping**, **tracert**, **tracert**, **DHCP** и др.

КРАТКАЯ СПРАВКА

В качестве средства захвата трафика предлагается Wireshark (возможны другие, по желанию студента). Wireshark установлен под Windows XP на машинах в компьютерном классе 2-504.

Для выполнения лабораторной работы необходимо знание:

- 1) функций и возможностей Wireshark,
- 2) форматов кадров, пакетов и запросов/ответов Ethernet, ARP for IP, IP, UDP, TCP, ICMP, DHCP,
- 3) утилит (команд) **ping**, **tracert**, **tracert**, **arp**, **netstat**, **ipconfig**, **ifconfig**, **route**,
- 4) основ одноадресной IP маршрутизации.

Особенности использования компьютеров в классе 2-504.

1. Вход в Linux: login – **stud**, пароль – **0**.

2. Нумерация однотипных интерфейсов **eth0**, **eth1** и т.д. в Linux выполняется по порядку их размещения в системном блоке сверху вниз. То есть **eth0** – верхний, ... **eth1**-следующий, и т.д.
3. Нумерация однотипных интерфейсов в Windows определяется порядком их инсталляции (установки) под Windows. Если первым был установлен нижний сетевой адаптер, он будет первым в списке (**Панель управления→Сеть→Конфигурация**) и т.д. Поэтому очередность следования сетевых интерфейсов под Windows проверяется опытным путем: либо при помощи **ping**, либо через Wireshark. Однако, в результатах работы утилит **ipconfig**, интерфейсы будут следовать в порядке их размещения в системном блоке сверху вниз.
4. При входе в Linux, как **stud/0**, остановка процессов осуществляется командой/скриптом **Kill**, а не **kill**. Но справку через **man** необходимо получать командой **man kill**.