МГТУ им. Н. Э. Баумана

Кафедра «Системы обработки информации и управления»

Методические указания к лабораторным работам 7-9 по дисциплине

Сети и телекоммуникации

Для студентов 3-го курса кафедры ИУ5

Разработали:

к.т.н., доцент Галкин В.А.

ст. преподаватель Аксенов А. Н.

ст. преподаватель Антонов А. И.

ассистент Канев А.И.

Москва 2021 г.

Содержание

Лабораторная работа №7, №8 и №9.

«Исследование протоколов сетевого и транспортного уровней IP-сетей с помощью анализатора протоколов».

1. Цель лабораторных работ.

Развитие практических навыков работы с протоколами стека TCP/IP и исследование возможностей протоколов ICMP, UDP, TCP.

2. Необходимое оборудование.

Аппаратные требования

- IBM - совместимый ПК в составе сети Интернет.
- 128 Мбайт оперативной памяти
- 5 Мбайт свободного места на HDD
- Используемый шлюз в Интернет должен пропускать ICMP, TCP и UDP трафик.

Программные требования

Лабораторная работа выполняется с помощью любых средств анализа сетевого траффика, в том числе с помощью встроенных средств и онлайн-сервисов. Рекомендованными средствами являются:

- Пакетный анализатор Wireshark http://www.wireshark.org/
- Сетевой анализатор NetInfo http://netinfo.tsarfin.com/

3. Теоретическая часть.

Протоколы — это правила работы программного обеспечения.

Стек протоколов - набор взаимодополняющих и тесно связанных друг с другом протоколов.

Термин "стек протоколов" происходит из концепции представления сети в виде вертикально расположенных уровней и сложенных в стек протоколов и относится к любой комбинации сетевых уровней и соответствующих протоколов.

В настоящей лабораторной работе предметом исследований является стек протоколов TCP/IP – наиболее распространенный и являющийся основным в сети Интернет.

IP (Internet Protocol) - протокол межсетевого взаимодействия, является протоколом сетевого уровня модели OSI и отвечает за перемещение данных между сетевыми

компьютерами в Интернет.

TCP (Transmission Control Protocol) - протокол управления передачей, который перемещает данные между прикладными программами.

UDP (User Datagram Protocol) - протокол пользовательских дейтаграмм, который также перемещает данные между приложениями. Он - более простой и менее надежный, чем TCP.

ICMP (Internet Control Message Protocol) - протокол управляющих сообщений Интернет, который управляет сетевыми сообщениями об ошибках и другими ситуациями, требующими вмешательства сетевых программ.

Схема движения данных.

Данные по сети передаются в три этапа:

- Информация должна пройти между приложениями и сетью. Это путь сквозь стек протоколов вниз к транспортному уровню.
- Определение сетью адреса получателя данных.
- Маршрутизация данных и прохождение данных сквозь стек протоколов вверх к сетевому приложению.

Схема движения данных пользователя представлена на рис. 1.

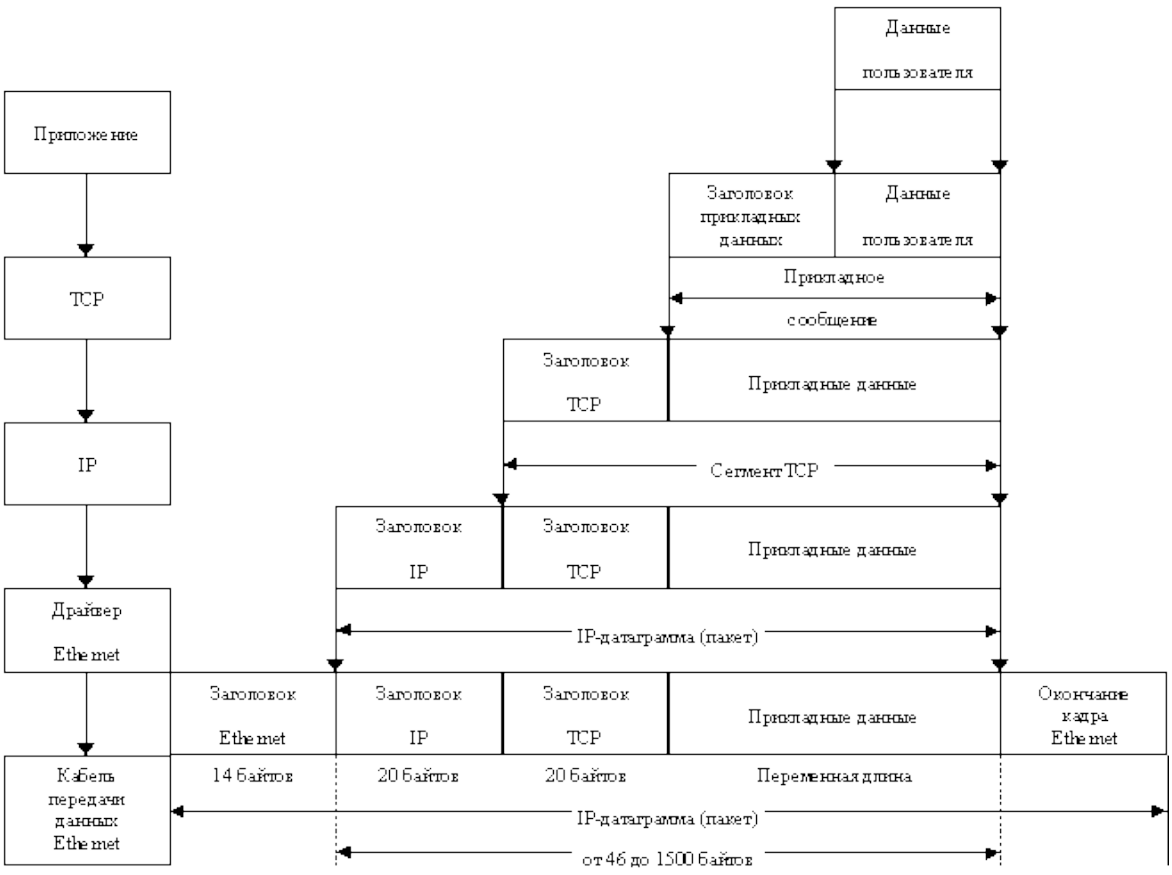

Рис. 1. Схема движения данных пользователя.

Протокол IP

Формат IP-дейтаграммы и поля заголовка представлены на рис.2.

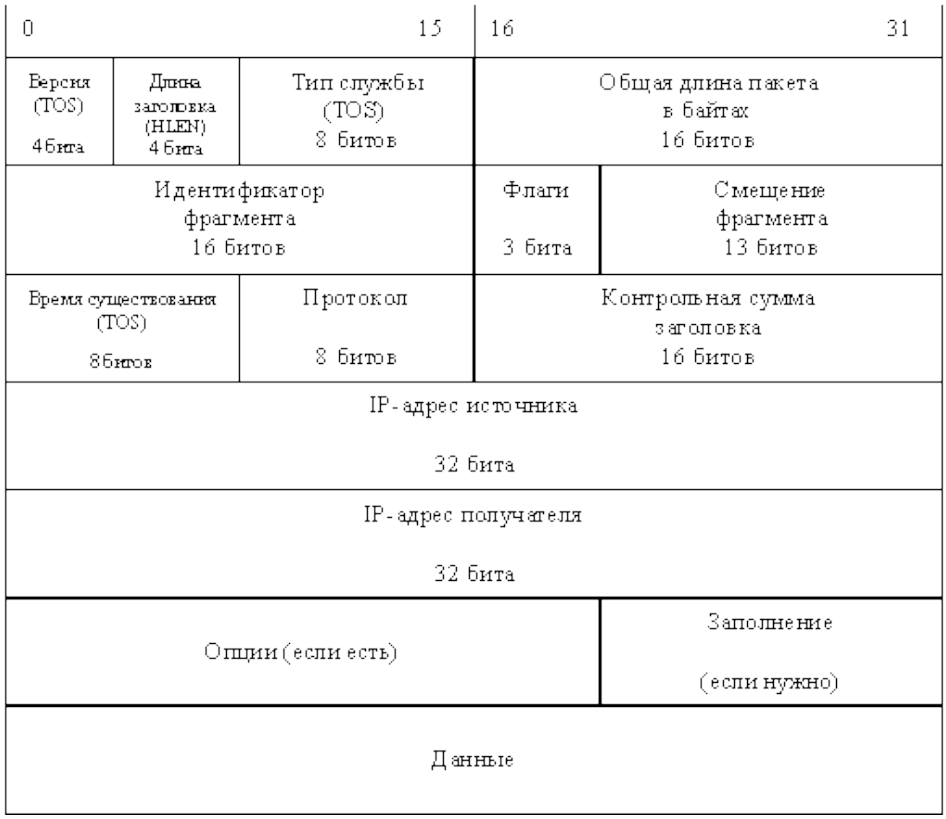

Рис. 2. Формат IP-дейтаграммы и поля заголовка.

Поля IP - протокола.

Номер версии VERS. Протокол IP постоянно развивается, необходимо знать, номер версии, чтобы правильно интерпретировать дейтаграмму.

Длина заголовка (HLEN) в 32 разрядных словах. Чаще всего длина IP-заголовка равна 20 байтам, поэтому данное поле обычно содержит число 5 (0101).

Тип сервиса (TOS). Поле "тип сервиса" разделено на 5 подразделов (рис.3).

Рис. 3. Формат поля TOS.

Первое трехразрядное субполе приоритет (precedence) редко применяется на практике. Последнее безымянное одноразрядное субполе всегда содержит 0. Между ними находятся четыре одноразрядных субполя, которые и называют собственно битами TOS. Каждому из четырех битов TOS сопоставлен определенный критерий доставки дейтаграмм: минимальная задержка, максимум пропускной способности, максимум надежности и минимум стоимости. Только один бит TOS может быть установлен в 1. По умолчанию все четыре бита равны 0, что означает отсутствие особых требований, то есть обычный сервис.

Длина пакета. Поле "длина пакета" задает длину IP-пакета, включая сам заголовок. Если локальная сеть построена по технологии Ethernet, уровень соединения инкапсулирует IP-дейтаграммы в кадры Ethernet перед передачей их в Интернет. Спецификация Ethernet ограничивает длину пакета до 1500 байтов.

Идентификатор. Наличие этого поля обусловлено фрагментацией пакетов в Интернет. Сетевые компьютеры используют поле с целью однозначной идентификации каждого посланного фрагмента для дейтаграммы, к которой он относится.

Флаги и смещение. Информация, содержащаяся в полях идентификации флагов и смещения фрагмента, позволяет правильно собрать фрагментированный пакет.

Время существования (TTL). Время существования определяет «время жизни» пакета в сети и не дает пакету возможность быть вечным скитальцем.

Протокол. Поле «протокол» в IP-заголовке указывает на протокол-источник данных, инкапсулированных в IP-дейтаграмму.

Контрольная сумма заголовка. Поле контрольной суммы в IP-заголовке содержит 16-ти битное число, являющееся контрольной суммой только для IP-заголовка.

IP-адрес источника и получателя. 32-битное поле «адрес источника» содержит IP-адрес компьютера - отправителя данных (вернее адрес его сетевого интерфейса).

Адрес получателя. Адрес получателя является 32-битным адресом пункта назначения пакета. В случае широковещательной передачи он состоит из единиц.

Опции IP. Это поле позволяет тестировать разнообразные сетевые приложения.

Протокол пользовательских дейтаграмм (UDP)

UDP-протокол умеет распознавать то приложение среди многих, работающих внутри компьютера, которому предназначены данные. Как правило, сеть назначает таким приложениям определенный номер порта. UDP пользуется дейтаграммами для доставки данных. Точно так же, как IP прицепляет к данным IP-заголовок, UDP прицепляет к ним UDP-заголовок, структура которого представлена на рис.4.

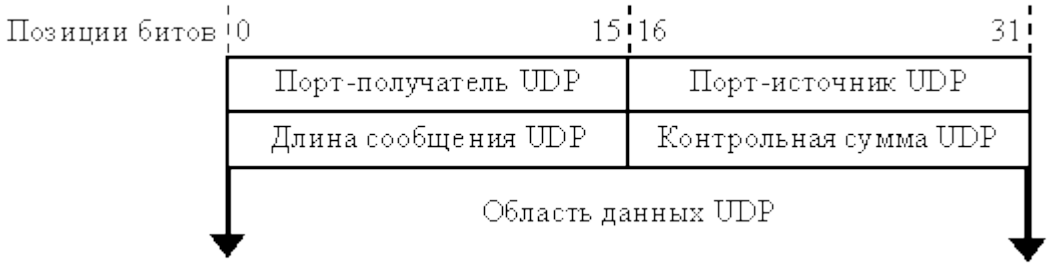

Рис. 4. Структура UDP – заголовка.

Длина UDP-заголовка - восемь байтов. Поля портов состоят из 16-битных целых чисел, представляющих номера портов протоколов. Поле "порт-источник" содержит номер порта, которым пользуется приложение-источник данных. Поле "порт-получатель" соответственно указывает на номер порта приложения-получателя данных. Поле "длина сообщения" определяет длину (в байтах) UDP-дейтаграммы, включая UDP-заголовок. Наконец, поле "контрольная сумма", в отличие от контрольной суммы IP-заголовка, содержит результат суммирования всей UDP-дейтаграммы, включая ее данные, область которых начинается сразу после заголовка.

Модуль UDP отслеживает появление вновь прибывших дейтаграмм, сортирует их и распределяет в соответствии с портами назначения.

Протокол TCP

Протокол Транспортного уровня модели OSI служит для передачи данных между сетевым и прикладным уровнями сетевой модели. TCP призван обеспечивать надежную, потоковую, ориентированную на соединение службу доставки данных.

TCP пытается оптимизировать пропускную способность сети, то есть увеличивает производительность доставки пакетов в Интернет. Для этого он динамически управляет потоком данных в соединении. Если буфер приемника данных переполняется, TCP просит передающую сторону снизить скорость передачи.

Данные TCP всегда переносит IP, то есть данные TCP всегда упаковываются в IP-дейтаграммы.

Для обеспечения надежной доставки и правильной последовательности данных в потоке TCP пользуется принцип скользящего окна и тайм-аута. Принцип скользящего окна позволяет послать несколько сообщений и только потом ожидать подтверждения. TCP накладывает окно на поток данных, ожидающих передачи, и передает все данные, попавшие в окно. Приняв подтверждение о доставке всех данных, TCP перемещает окно дальше по потоку и передает следующие попавшие в него сообщения. Работая сразу с несколькими

сообщениями, TCP может одновременно "выставить" их на сетевой канал и только потом ожидать прихода подтверждения. Метод скользящего окна значительно увеличивает производительность соединения, а также эффективность циклов обмена сообщениями и подтверждениями об их доставке.

<table>
<tr><td colspan="2">0
15</td><td colspan="2">16
31</td></tr>
<tr><td colspan="2">Порт источника 16 бит</td><td colspan="2">Порт назначения 16 бит</td></tr>
<tr><td colspan="4">Позиционный номер 32 бита</td></tr>
<tr><td colspan="4">Квитанция 32 бита</td></tr>
<tr><td>Длина заголовка 4 бита</td><td>Резерв 16 бит</td><td>Флаги</td><td>Размер окна приема 16 бит</td></tr>
<tr><td colspan="2">Контрольная сумма 16 бит</td><td colspan="2">Указатель границы срочных данных 16 бит</td></tr>
<tr><td colspan="4">Опции (если таковые имеются)</td></tr>
<tr><td colspan="4">Данные (если таковые имеются)</td></tr>
</table>

Рис.5. Формат заголовка сегмента TCP.

TCP регулирует полосу пропускания сети, договариваясь с другой стороной о некоторых параметрах данных. Причем процесс регулировки происходит на протяжении всего соединения TCP. В частности, регулировка заключается в изменении размеров скользящего окна. Если сеть загружена не сильно и вероятность столкновения данных минимальна, TCP может увеличить размер скользящего окна. При этом скорость выдачи данных на канал увеличивается и соединение становится более эффективным.

Если, наоборот, вероятность столкновения данных велика, TCP уменьшает размер скользящего окна.

Как правило, модуль TCP передает несколько сегментов, прежде чем скользящее окно заполнится целиком. Большинство систем в Интернет устанавливают окно равным по умолчанию 4096 байтам. Иногда размер окна равен 8192 или 16384 байтам

Заголовок сегмента TCP представлен на рис.5. Обычно (при отсутствии опций) заголовок имеет размер 20 байтов. Напомним, что передаваемый TCP – сегмент с данными инкапсулируется в IP – дейтаграмму.

Номера портов источника и назначения (source port number и destination port number) идентифицируют взаимодействующие приложения.

Позиционный номер (sequence number) сегмента указывает то место в потоке данных от источника до конечного получателя, которое занимает первый байт содержащихся в этом сегменте данных. В начальном сегменте, посылаемом при установлении соединения, присутствует флаг SYN, а в поле позиционный номер содержится так называемый начальный позиционный номер ISN(initial sequence number), выбранный данным хостом для этого нового соединения. Первому байту данных, переданному хостом по новому соединению, будет присвоен позиционный номер, равный ISN+1. Такой сдвиг в нумерации кратко формулируется правилом: флаг SYN поглощает одну позицию.

В поле квитанция (acknowledgement - ACK) передающей стороне сообщается позиционный номер следующего в потоке данных сегмента, ожидаемого принимающей стороной. Это число всегда на единицу больше номера последнего успешно принятого байта.

Поле размер заголовка (header length) необходимо, поскольку в заголовке далее могут следовать поля опций переменной длины. Записанная в этом поле константа означает число отводимых под заголовок 32-разрядных слов, и, следовательно, длина заголовка не превышает 60 байтов. При отсутствии опций размер заголовка всегда равен 20 байтам.

В TCP-заголовке предусмотрены 6 двоичных флагов (flags) , причем некоторые из них могут быть установлены одновременно.

URG – флаг срочных данных. Поле указатель границы срочных данных заголовка имеет смысл только при URG =1

ACK – флаг квитирования. Поле квитанция имеет смысл только при ACK = 1.

PSH – флаг «проталкивания» (push). TCP-модуль хоста назначения должен незамедлительно отдать данные из сегмента своему приложению.

RST – флаг сброса соединения.

SYN – флаг синхронизации позиционных номеров сегментов при установлении соединения.

FIN – флаг окончания передачи. Он означает, что источник сегмента закончил передачу данных и закрывает свой канал вывода в текущем соединении.

Темп передачи потока данных в каждом направлении по TCP-соединению регулируется обеими участвующими в обмене сторонами благодаря тому, что каждая сторона объявляет свой размер окна приема (window size), то есть количество байтов, которое она в данный момент готова принять вслед за байтом, номер которого указан в поле квитанция.

Поле контрольная сумма (TCP – checksum) содержит значение, подсчитанное для всего сегмента, включая его заголовок и данные.

Указатель границы срочных данных (urgent pointer) действует лишь при условии, что в сегменте установлен флаг URG. Это положительная константа, равная смещению номера последнего байта срочных данных относительно позиционного номера в заголовке сегмента. Срочный режим (urgent mode) предусмотрен в TCP, чтобы в поток передаваемых обычных данных приложение могло внедрять цепочки каких-либо особым образом интерпретируемых байтов (например, команд) так, чтобы они были обнаружены и выделены из потока на принимающей стороне.

Открытие TCP соединения

Открытие TCP-соединения состоит из трех фаз.

1. Запрашивающая сторона (обычно это клиент) посылает сегмент с флагом SYN, указывая номер порта получателя (сервера) с которым хочет соединиться, а также начальный позиционный номер ISN(initial sequence number).
2. Сервер отвечает сегментом SYN, где сообщает свой начальный позиционный номер и одновременно подтверждает получение сегмента SYN от клиента – он устанавливает флаг ACK, а в качестве подтверждаемого позиционного номера указывает номер, на единицу больше принятого, то есть ISN клиента плюс один.
3. Клиент подтверждает получение сегмента SYN от сервера, выслав сегмент с флагом ACK и номером квитанции, равным принятому от сервера начальному позиционному номеру ISN плюс один.

Обмен этими тремя сегментами и составляет процедуру установления соединения. Часто такой механизм называют троекратным рукопожатием (three-way handshake).

Закрытие TCP соединения

Если для установления соединения необходим обмен тремя сегментами, то для его закрытия таковых требуется четыре. Поскольку соединение TCP является полнодуплексным (то есть данные могут передаваться в обоих направлениях независимо), каждое направление необходимо закрывать по отдельности. Закрытие одного направления называется полузакрытием (half-close). Согласно протоколу любая из сторон, закончив передачу данных, может послать сегмент FIN. Когда TCP-модуль получает сегмент FIN, он обязан уведомить обслуживаемое приложение, что другая сторона закрыла свое направление передачи данных.

Приход FIN означает лишь то, что поступление данных от партнера по этому соединения прекращается. Но TCP-модуль может посылать данные и после получения им

FIN. Предоставляемая приложению возможность продолжать передачу по полузакрытому соединению на практике используется редко.

Говорят, что сторона, первой закрывающая соединение (то есть посылающая первый FIN), производит активное закрытие соединения (active close). Другая сторона (которая получает этот FIN и отвечает на него своим FIN) выполняет пассивное закрытие соединения (passive close). Итак:

1. TCP-модуль одной из сторон посылает сегмент FIN и тем самым закрывает поток данных со своей стороны.
2. В ответ на пришедший FIN TCP-модуль второй стороны посылает подтверждение полученного позиционного номера плюс один.
3. Приложение на второй стороне закрывает свой поток данных, и его TCP-модуль посылает FIN.
4. Первый хост отвечает сегментом ACK с квитанцией, равной позиционному номеру полученного им сегмента FIN плюс один.

Протокол управляющих сообщений ICMP

Протокол ICMP (Internet Control Message Protocol) служит для обмена сообщениями об ошибках и различных особых случаях, требующих обработки. ICMP-сообщения содержат управляющие данные, используемые либо на IP-уровне, либо на более высоком уровне (TCP или UDP). Некоторые ICMP-сообщения трансформируются в коды ошибок, возвращаемых пользовательским процессам. В иерархии протоколов ICMP часто относят к сетевому уровню, наряду с IP, но ICMP-сообщения инкапсулируются в IP-диаграммы. Структура ICMP-сообщения представлена на рис.6.

0 7	8 15	16 31
Тип (8 бит)	Код (8 бит)	Контрольная сумма(16 бит)
Содержание сообщения (зависит от типа и кода)		

Рис.6. Структура ICMP-сообщения

Первое слово (4 байта) содержит три поля, общие по смыслу и формату для любых разновидностей сообщений. Следующая затем содержательная часть сообщения форматируется по-разному в зависимости от типа сообщения.

Предусмотрено 15 различных значений для поля тип (type), которое идентифицирует разновидность ICMP-сообщения. Кроме того, некоторые типы ICMP-сообщений дополнительно используют значения поля код(code) для конкретизации тех или иных

условий.

Поле контрольная сумма (checksum) относится ко всему ICMP-сообщению и является обязательным

Разновидности ICMP – сообщений

В таблице 1 приведены всевозможные разновидности ICMP-сообщений, определяемые полями тип (type) и код (code). Последние два столбца таблицы позволяют отличить запросы и отклики на них от сообщений об ошибках. Необходимо различать эти две разновидности, потому что обработка ICMP-сообщения об ошибке имеет свою специфику.

Таблица 1 Разновидности ICMP-сообщений

Тип	Код	Описание	Запрос/Ответ	Ошибка
0	0	Эхо-ответ (echo reply)	+	
3		Адресат недоступен (destination unreachable)		
	0	сеть недоступна		+
	1	хост недоступен		+
	2	протокол недоступен		+
	3	порт недоступен		+
	4	необходима фрагментация, но есть флаг DF		+
	5	маршрутизация от источника невыполнима		+
	6	сеть назначения неизвестна		+
	7	хост назначения неизвестен		+
	8	хост источника изолирован(устарело)		+
	9	сеть назначения административно закрыта		+
	10	хост назначения административно закрыт		+
	11	сеть недоступна для данного типа сервиса TOS		+
	12	хост недоступен для данного типа сервиса TOS		+
	13	связь административно закрыта фильтром		+
	14	нарушение старшинства хостов		+
	15	действует отключение по старшинству		+
4	0	Прикрыть источник (source quench)		+
5		Перенаправление(redirect)		
	0	перенаправить путь на сеть		+

	1	перенаправить путь на хост		+
	2	перенаправить путь на сеть для типа сервиса TOS		+
	3	перенаправить путь на хост для типа сервиса TOS		+
8	0	Эхо-запрос (echo request)	+	
9	0	Объявление маршрутизатора (router advertisement)	+	
10	0	Запрос маршрутизатора (router solicitation)	+	
11		Срок истек (time exceeded)		
	0	срок истек на переходе (TTL = 0)		+
	1	срок истек при сборке		+
12		Нарушены параметры дейтаграммы		
	0	испорчен IP-заголовок		+
	1	отсутствует необходимая опция		+
13	0	Запрос отсчета времени (timestamp request)	+	
14	0	Отклик отсчета времени (timestamp reply)	+	
15	0	Запрос информации(устарело)	+	
16	0	Информационный отклик(устарело)	+	
17	0	Запрос адресной маски (address mask request)	+	
18	0	Ответ адресной маски (address mask reply)	+	

В ICMP-сообщении об ошибке всегда возвращается IP-заголовок и первые 8 байтов IP-дейтаграммы, признанной ошибочной. Это позволяет ICMP-модулю сопоставить полученное им сообщение об ошибке с конкретным протоколом TCP или UDP (по значению поля протокол в возвращенном IP-заголовке) и с конкретным пользовательским процессом (по номеру порта, который находится в TCP или UDP-заголовке в возвращенных первых 8 байтах IP-дейтаграммы).

ICMP-сообщение об ошибке никогда не генерируется в ответ на следующие пакеты:

1. На ICMP-сообщение об ошибке.
2. На дейтаграмму, посланную по широковещательному IP-адресу или групповому IP-адресу.
3. На какой-либо фрагмент дейтаграммы, кроме первого ее фрагмента.
4. На дейтаграмму, в которой адрес источника не определяет конкретный хост. Это означает, что адрес источника не может быть нулевым адресом, адресом внутренней петли хоста (loopback) и не может быть широковещательным или групповым адресом.

Эти правила предназначены для предотвращения так называемых широковещательных штормов (broadcast storms).

Как работает Ping

Программа проверяет доступность зондируемого ею объекта в сети подобно локатору: посылает хосту ICMP-сообщение эхо-запрос (echo request) и ждет от него эхо-отклик (echo reply).

Эта утилита внешне действует подобно клиенту, а адресуемый хост, от которого приходит отклик, выступает в роли сервера. Однако на самом деле обработка эхо-запросов и генерация откликов осуществляется не каким-либо пользовательским процессом, а непосредственно ядром.

Формат ICMP сообщений, содержащих эхо-запрос или эхо-отклик, представлен на рис.7

0 7	8 15	16 31
Тип (0 или 8)	Код (0)	Контрольная сумма
Идентификатор	Порядковый номер	
Необязательные данные		

Рис.7. Формат ICMP сообщений

Как и при обработке любых других ICMP-запросов, в ответном сообщении возвращаются поля идентификатор (identifier) и порядковый номер (sequence number) запроса. Также должны быть возвращены и содержащиеся в запросе необязательные данные (optional data), так как они используются источником запроса при интерпретации пришедшего на запрос ответа.

Unix-реализации программы Ping фиксируют в поле идентификатор (identifier) ICMP-сообщения свой системный идентификатор процесса (process ID), пославшего сообщение. Это в последствии позволяет правильно распределять возвращенные ответы, когда на одном хосте параллельно работают несколько копий Ping.

В первом пакете, посылаемом Ping, значение поля порядковый номер (sequence number) устанавливается равным 0. Каждый раз при отправлении нового запроса это значение увеличивается на единицу.

Как работает Traceroute

Эта утилита даёт возможность отследить текущий маршрут движения IP – дейтаграмм от одного хоста к другому. Кроме того, программа позволяет использовать IP – опцию

маршрутизации от источника (source route).

В основе Traceroute лежит идея отправки источником UDP-пакета адресату и постепенного изменения времени жизни (time-to-live, TTL). Первоначально TTL пакета равен 1, и когда пакет достигает первого маршрутизатора, его TTL сбрасывается (текущее значение поля TTL уменьшается на единицу), а маршрутизатор генерирует и отправляет в адрес источника ICMP-пакет со сведениями о превышении лимита времени. Тогда источник увеличивает на 1 начальное значение TTL, так что на сей раз UDP-пакет достигает следующего маршрутизатора, а тот тоже отсылает ICMP-пакет по превышению лимита времени. Совокупность этих ICMP-сообщений дает список IP-адресов, пройденных на пути к конечной точке. Когда TTL увеличится настолько, что UDP-пакет достигнет искомой конечной точки, возвращается ICMP сообщение о недостижимости порта, поскольку на получателе ни один процесс не ждёт вашего сообщения.

4. Изучение программы NetInfo.

Для изучения назначения и возможностей утилит NetInfo загрузите эту программу по адресу.

Утилита Local Info

Утилита для предоставления сетевой информации про локальный хост и текущую версию сокетов Windows.

Вы можете использовать Local Info для:

- идентификации своего компьютера в сети
- определения того, какая локальная информация доступна

Для отображения информации:

1. Выберите вкладку Local Info. Результат будет выведен в область Response.
2. Нажатие кнопки Refresh обновит информацию.

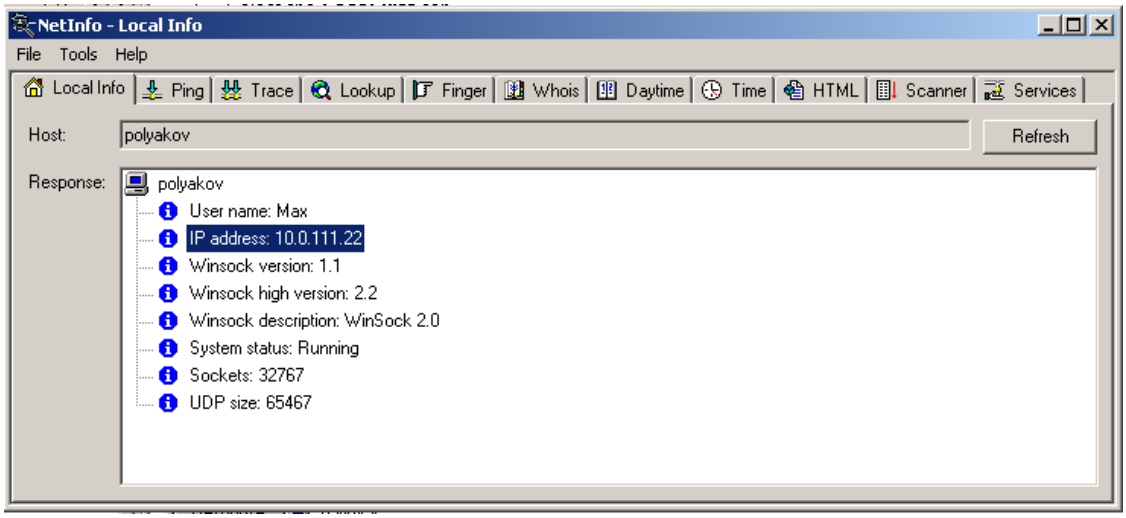

Рис.8. Пример использования Local Info

Утилита Ping (Packet Internet Groper)

Диагностическая утилита для проверки доступности удалённого хоста. Посылается ICMP эхо-запрос и ожидается ответ.

Вы можете использовать Ping для:

- «пинга» удалённого хоста для проверки сетевого соединения
- «пинга» удалённого хоста для определения скорости передачи данных в физической среде
- «пинга» хостов «по спирали» начиная с локального (127.0.0.1), затем всё более дальние компьютеры и шлюзы, для определения различных неисправностей.

Для «пинга» хоста:

1. Выберите вкладку Ping
2. В строке Host введите имя хоста или его IP адрес.
3. В опциях (вызываются нажатием правой клавиши мыши и выбором соответствующего пункта меню Options) устанавливаются следующие настройки:
   - Packets to send - Количество посылаемых пакетов.
   - Timeout - Сколько секунд будет ожидаться ответ от хоста.
   - Packet size - Длина в байтах каждого пакета.

Нажмите кнопку Start. Утилита посылает эхо запрос и ждёт ответа. Если «пинг» успешен, то в области Response выводятся результаты.

Если Ping не получает ответа за время определённое в Timeout, выдаётся сообщение о неудаче. Существует несколько причин неудачи: удалённый хост может не функционировать, может не работать сеть или какой-либо шлюз или маршрутизатор на пути к удалённому хосту или же сервис Ping просто не поддерживается удалённым хостом.

Во время выполнения запроса кнопка Start изменяется на Stop, и Вы можете прекратить «пинг» в любой момент.

Примечание: для использования Ping необходим статический IP адрес для Вашего компьютера. Также подойдёт IP адрес назначаемый DHCP сервером. Ping не будет работать в системах с эмуляцией IP адресов, таких, как, например, UNIX с запущенными TIA или SliRP. Ping не будет работать через брандмауэры, если только брандмауэр не сконфигурирован для пропуска ICMP пакетов.

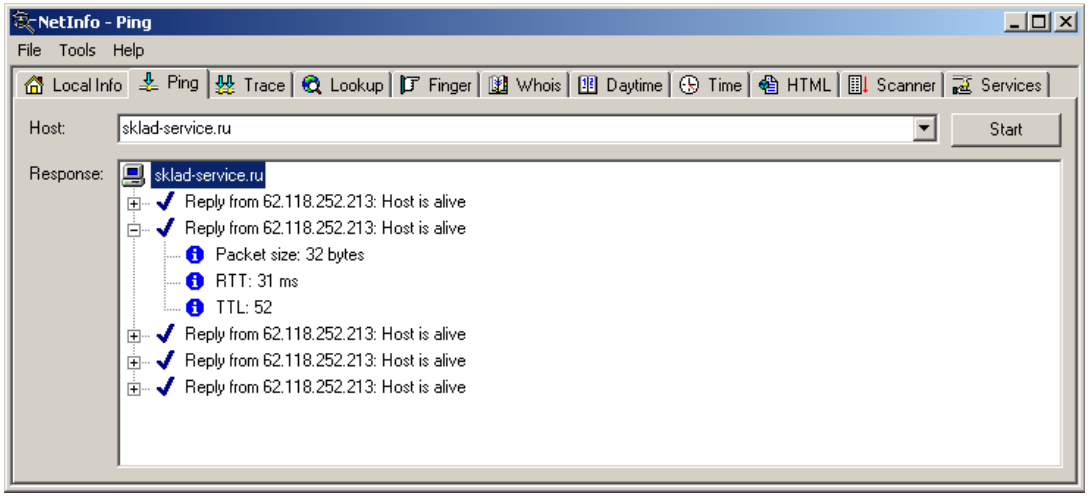

Рис. 9 Пример использования утилиты Ping

←

Утилита Trace (traceroute)

Утилита, которая сообщает обо всех маршрутизаторах между компьютером, пославшим запрос, и удалённым хостом.

Traceroute также показывает время ответа (в миллисекундах), которое показывает сколько нужно пакету времени для прохождения отрезка пути до определённого маршрутизатора и обратно. Это время зависит от загруженности сети.

Для инициации TraceRoute:

1. Выберите вкладку Trace.
2. В строке Host введите имя хоста или его IP адрес.
3. В опциях (вызываются нажатием правой клавиши мыши и выбором соответствующего пункта меню Options) устанавливаются следующие настройки:

- Timeout - Сколько секунд TraceRoute будет пытаться найти путь к удалённому хосту.
- Packet Size - Величина пакета в байтах.
- Number of Hops - Число узлов до удалённого хоста (как правило за 30 «прыжков» можно достигнуть любого хоста)

Нажмите кнопку Start. Traceroute начинает поиск и выводит результаты в область Response.

Во время выполнения запроса кнопка Start изменяется на Stop, и вы можете прекратить поиск в любой момент.

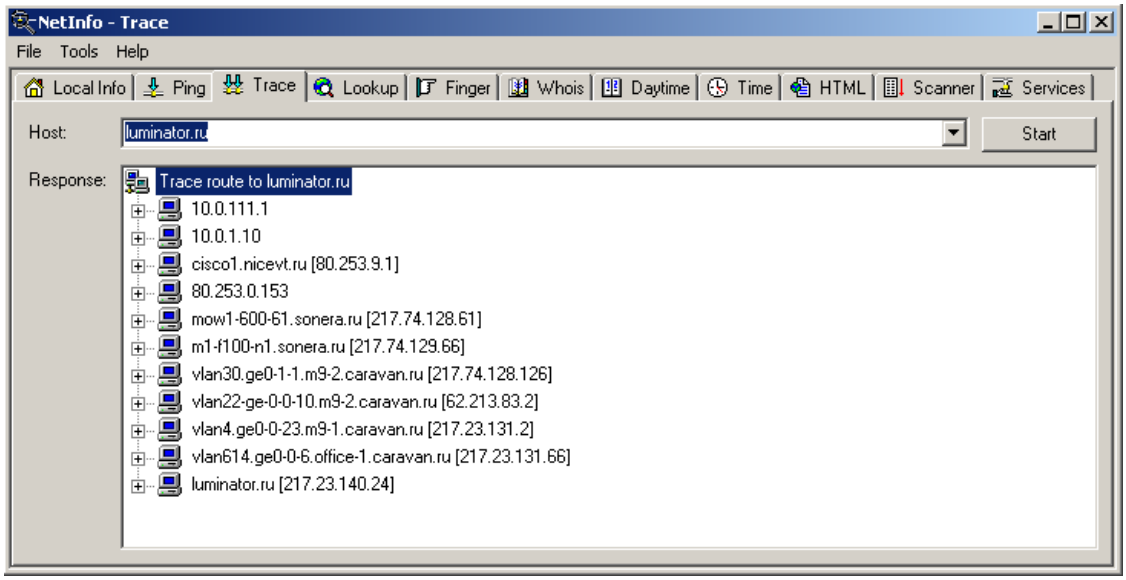

Рис. 10. Пример использования утилиты Trace

Утилита Lookup

Утилита возвращает официальное имя хоста, IP адрес и псевдоним (если он существует) из DNS.

Вы можете использовать утилиту Lookup для:

- Получения имени хоста из его IP адреса
- Получения IP адреса хоста из его имени.

Для инициализации запроса Lookup:

1. Выберите вкладку Lookup.
2. В строке Host введите имя хоста или его IP адрес.

Нажмите кнопку Start.

Lookup начинает поиск и выводит результаты в область Response.

Во время выполнения запроса кнопка Start изменяется на Stop, и Вы можете прекратить поиск в любой момент.

Примечание: Для Lookup необходима связь с сетью имеющей DNS или WINS сервер или какой-нибудь другой сервер имён. Ваш компьютер должен быть сконфигурирован для доступа к DNS.

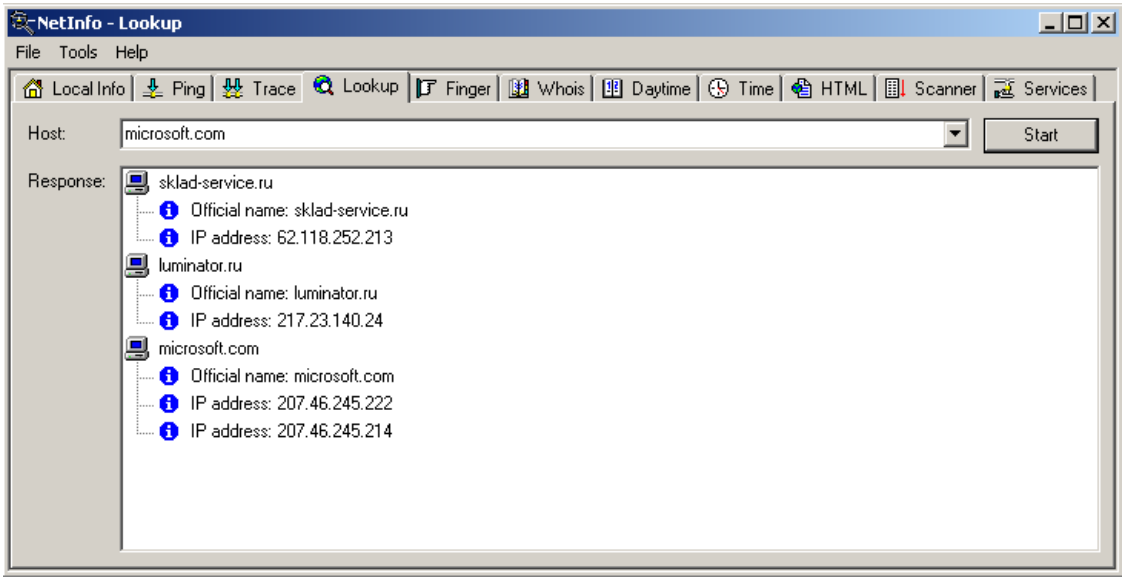

Рис.11. Пример использования утилиты Lookup

Утилита Finger

Утилита, позволяющая найти и отобразить информацию обо всех пользователях сетевого узла. Эта информация включает в себя список пользователей, подключенных к хосту в данный момент (их идентификаторы и имена). Также для каждого пользователя указывается его корневая директория, время подключения, место нахождения офиса, когда они последний раз получали почту и когда они последний раз читали почту.

Запрос Finger также отображает всю информацию, содержащуюся в файлах .plan или .project в корневой директории. Эти файлы часто используются как простой путь для хранения информации. Например, сервер Finger на quake@geophys.washington.edu выдаёт систематизированную по датам информацию о землетрясениях, которые происходили в северо-западном регионе Соединённых Штатов.

Для инициализации запроса Finger:

1. Выберите вкладку Finger.
2. В строке Host введите имя хоста или его IP адрес.
3. Нажмите кнопку Start.

Клиент Finger связывается с сервером Finger. Результаты запроса отображаются в области Response. Если на удалённом хосте нет сервера Finger, то клиент выводит соответствующее сообщение — No server found there.

Примечание: обычно только UNIX или NT хосты поддерживают сервер Finger. Многие системные администраторы выключают серверы Finger, так как они представляют потенциальный риск.

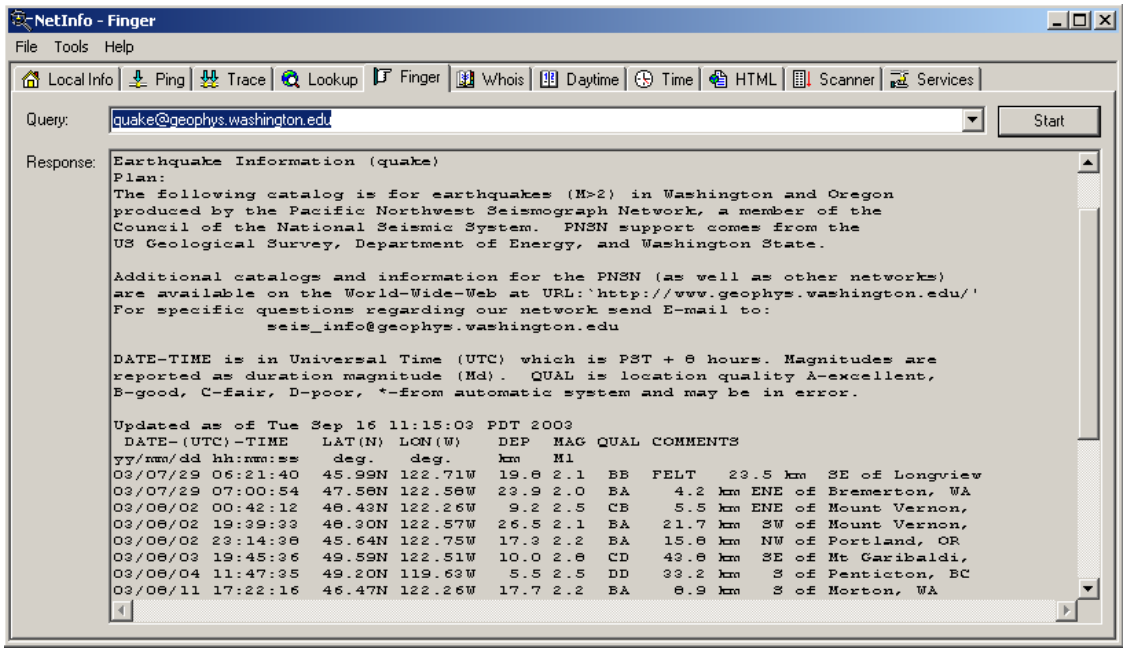

Рис.12. Пример использования утилиты Finger

Утилита Whois

Сетевая информационная утилита, которая предоставляет информацию о том, кто владеет Интернет хостом или доменом, и с кем вы можете проконтактировать относительно этого хоста или домена. Whois запрос показывает фамилию контактного лица, адрес его электронной почты, номер телефона, и сетевой почтовый ящик для всех пользователей и организаций, которые зарегистрированы на одном из официальных Whois серверов, таких как база данных Internet Network Information Center (interNIC).

Для инициализации запроса Whois:

1. Выберите вкладку Whois.
2. В текстовом поле Query, введите текст запроса, например microsoft.com.

Введите имя или “указатель” (уникальный идентификатор, который соответствует Whois записи) человека или организации. Выпадающий список показывает запросы, которые были введены ранее. Запрос ищет все записи в базе данных Whois до точного совпадения имени или “указателя”.

Если вы не знаете имени или указателя, вы можете ввести строку запроса частично, поставив в конце одну или несколько точек. Например, введя “Mack.” вы найдете “Mack,” “Mackall,” “Mackey.”

Если получено более, чем одно местонахождение имени, Whois возвращает краткое описание каждого. Вы можете затем взять указатель имени (показанный в скобках), о котором вы хотите получить больше информации и ввести его как строку запроса, с восклицательным знаком впереди, например “!ABC.”

3. В диалоговом окне Options находится набор опций, которые вы можете использовать.

NetInfo позволяет автоматически определять Whois сервер, основываясь на стране, взятой из строки запроса. Вы можете убрать флажок Autodetect для того, чтобы запретить вашим запросам обращаться к серверам, указанным в установках Whois:

whois.internic.net - Предоставляет информацию о пользователях и организациях, зарегистрированных в Internet Network Information Center (interNIC).

whois.arin.net - Предоставляет информацию о пользователях и организациях, зарегистрированных на сервере American Registry for Internet Numbers.

whois.nic.gov - Предоставляет информацию о пользователях и организациях, зарегистрированных в сети U.S. government.

whois.nic.mil - Предоставляет информацию о пользователях и организациях, зарегистрированных в сети U.S. military.

whois.ripe.net - Предоставляет информацию о пользователях и организациях, зарегистрированных на сервере European IP Address Allocations.

whois.nic.uk - Предоставляет информацию о пользователях и организациях, зарегистрированных на сервере United Kingdom IP Address Allocations.

whois.ripn.net - Предоставляет информацию о пользователях и организациях, зарегистрированных на сервере Russian IP Address Allocations.

whois.apnic.net - Предоставляет информацию о пользователях и организациях, зарегистрированных на сервере Asia Pacific IP Address Allocations. whois.aunic.net - Предоставляет информацию о пользователях и организациях, зарегистрированных на сервере Australia IP Address Allocations.

Примечание: Вы можете выбирать из текущего списка известных Whois серверов и добавлять новые серверы к списку.

4. Нажмите кнопку Start.

Whois клиент соединится с указанным Whois сервером. Результат запроса появится в области Response. Если запрос найдет единственное нахождение строки поиска (человека или организации), он показывает детальную информацию для этого человека или организации. Если он находит больше, чем одно вхождение строки поиска, он показывает краткую информацию о каждой записи, которая совпадает с условием поиска.

Примечание: чтобы показать экран помощи по использованию сервисов, которые предоставляются NIC через Whois, пошлите Whois запрос с текстом “help”.

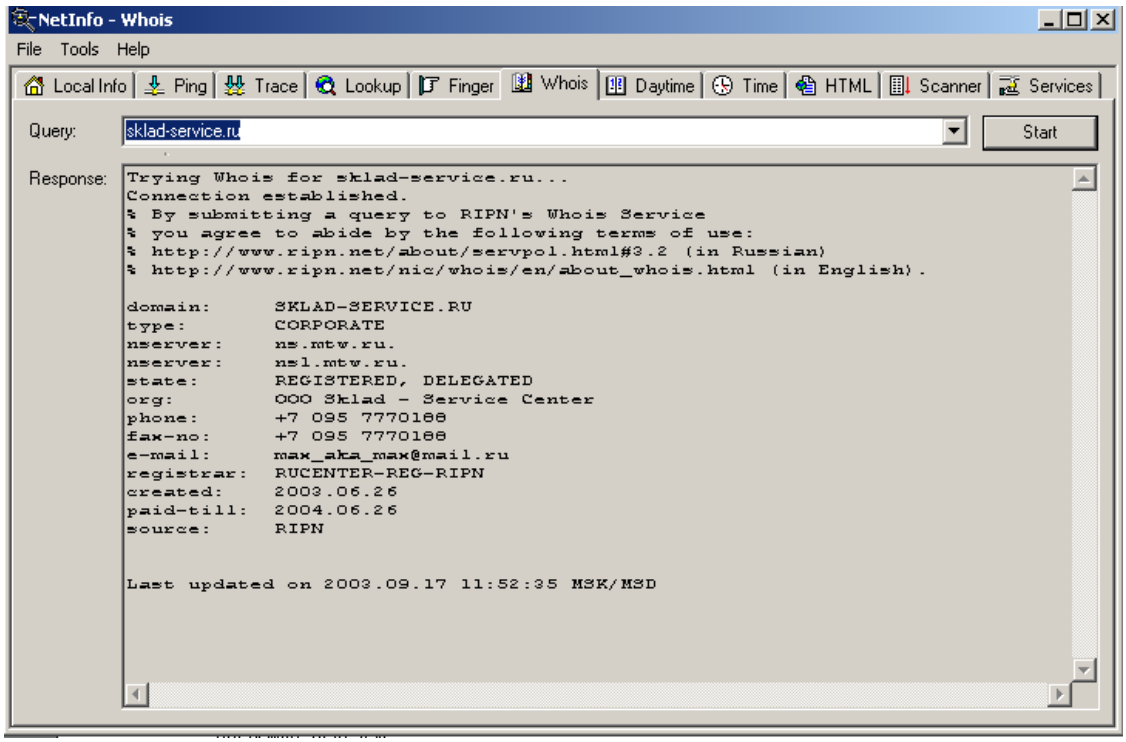

Рис.13. Пример использования утилиты Whois

Утилита Daytime

Сетевая информационная утилита, которая принимает локальное время с другого компьютера. Множество веб-серверов и DNS-серверов отвечают на такой запрос.

Чтобы получить локальное время, сделайте следующее:

1. Выберете вкладку Daytime.
2. В поле Server, введите имя хоста или ip адрес удаленного daytime сервера (например, www.mit.edu).

Выпадающий список показывает ранее введенные имена или ip адреса.

3. Нажмите на кнопку Start.

Daytime клиент соединяется с daytime сервером. Результат запроса показывается в области Response.

При попытке соединиться с сервером, который не поддерживает этот сервис, результатом будет сообщение об отказе соединения.

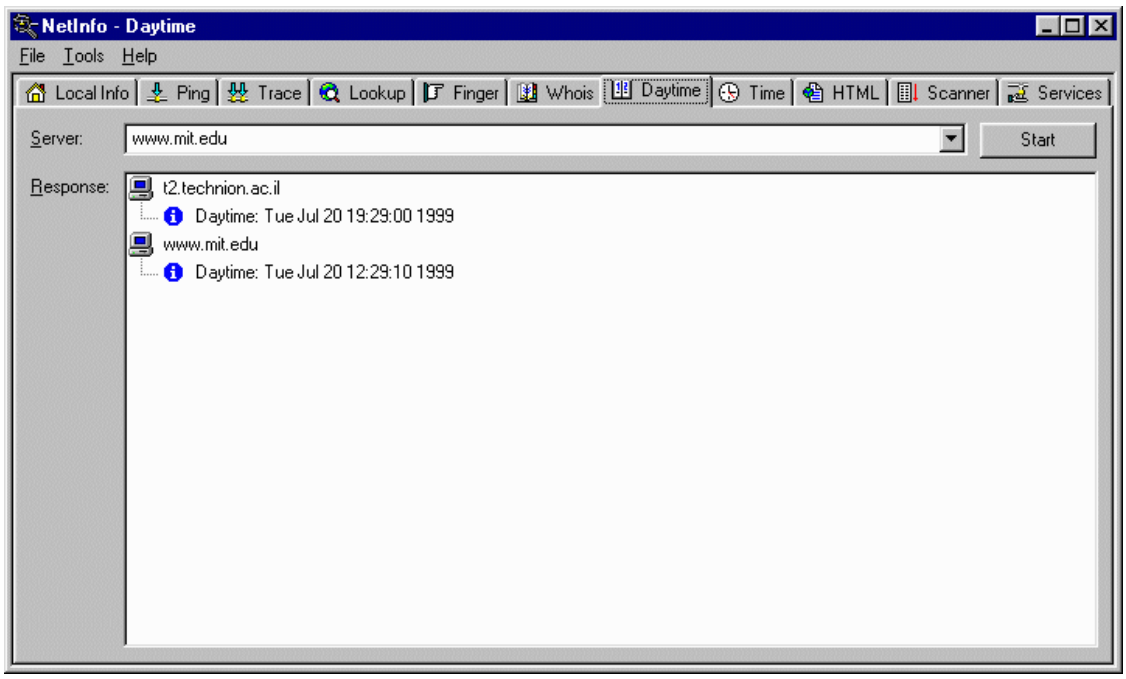

Рис.14. Пример использования утилиты Daytime

Утилита Time

Сетевая информационная утилита, которая получает точное значение времени с сервера времени. Эта утилита позволяет вам синхронизовать ваши локальные системные часы с часами на удаленном сервере.

Для синхронизации сделайте следующее:

1. Выберете вкладку Time.
2. В текстовом поле Server, введите имя или IP адрес сервера времени (например, www.mit.edu).

   Выпадающий список показывает ранее введенные имена или ip адреса.
3. Нажмите кнопку Start.

Утилита Time устанавливает соединение с удаленным сервером и показывает имя сервера, текущее время, полученное с сервера, а также различие во времени между часами сервера и вашими.

4. Нажмите правой кнопкой на сервере времени в области Response, чтобы показать выпадающее меню, а затем выберете Synchronize.

При успешном изменении, утилита Time показывает сообщение, которое говорит о том, что ваши часы были обновлены.

Примечание: Временной протокол имеет разрешение в одну секунду и не дает представления о времени прохождения пакета.

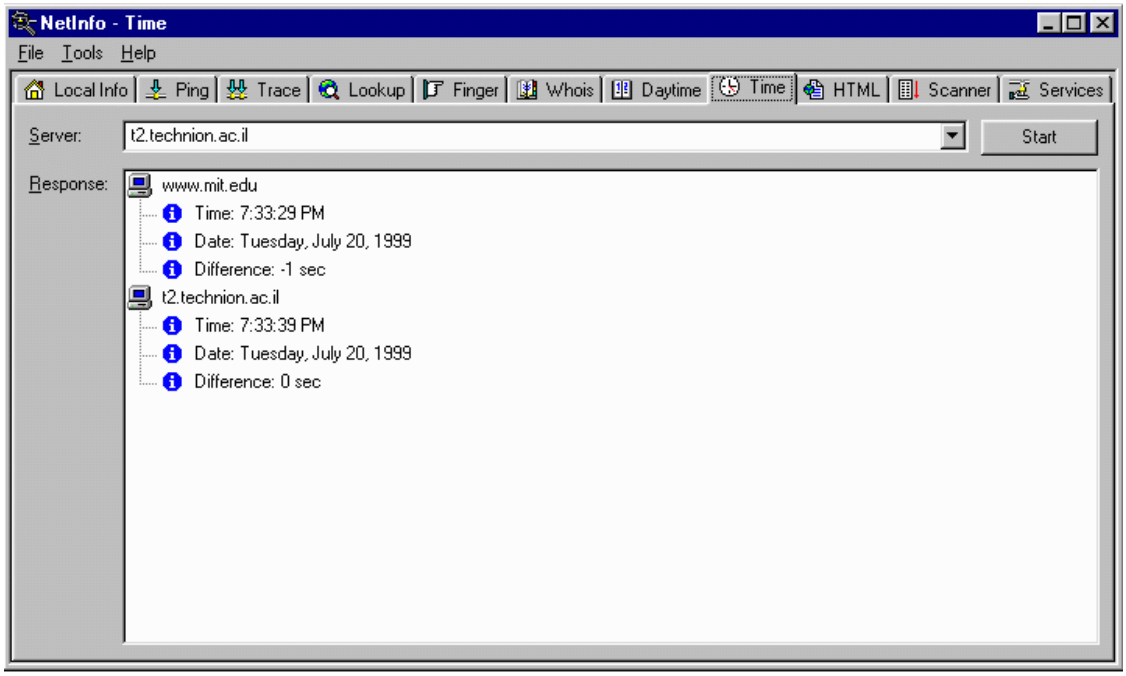

Рис.15. Пример использования утилиты Time:

Утилита HTML (HyperText Markup Language) – Гипертекстовый язык разметки

Сетевая информационная утилита, которая посылает запрос на указанный веб адрес (URL) и возвращает полную заголовочную информацию (включая cookies), а также возвращает информацию со страницы в форматированном HTML коде.

Вы можете использовать утилиту HTML для отладки вашего сайта.

Для запуска утилиты сделайте следующее:

1. Выберете вкладку HTML.
2. В поле URL введите адрес веб страницы, которую вы хотите запросить. Он должен указывать файл с веб сайта (например: http://hostname/page или hostname/page). Выпадающий список показывает адреса, введенные ранее.
3. В диалоговом окне Options находится набор опций:

Get from the wire - Получить данные из провода, даже если они были локально закэшированы.

Do not cache - Не кэшировать данные.

Full header information - Показывать полную заголовочную информацию.

Page data - Показывать данные со страницы.

Нажмите кнопку Start.

Результаты запроса появится в области Response.Если указанный хост не имеет веб сервера, утилита HTML покажет сообщение: No server found there.

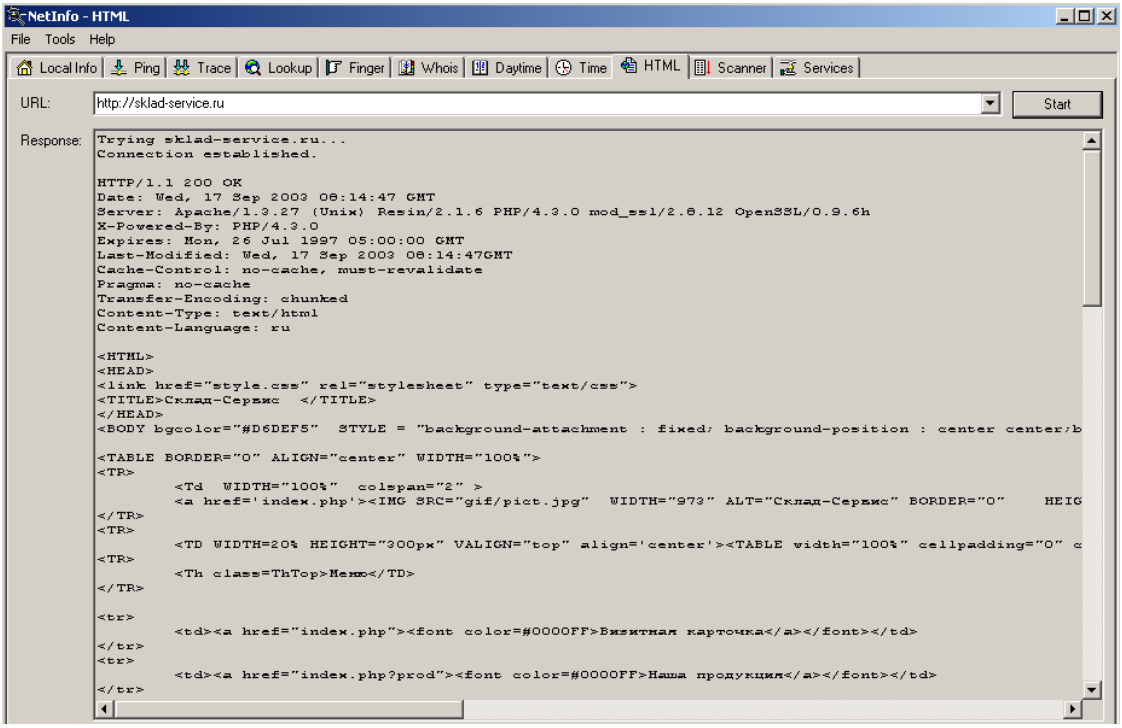

Рис.16. Пример использования утилиты HTML:

Утилита Scanner

Сетевая информационная утилита, которая сканирует все хосты в указанном диапазоне IP адресов и проверяет статус хостов.

Чтобы запустить Scanner, сделайте следующее:

1. Выберете вкладку Scanner.
2. В поле Address, введите IP адрес для сканирования (например, 192.41.61.50).
3. В диалоговом окне Options находится набор опций:

Ascending - Когда эта опция включена, Scanner сканирует все имена хостов в в порядке возрастания IP адреса.

Enabled - Когда эта опция включена, Scanner проверяет статус хоста для каждого IP адреса.

Timeout - Количество секунд, когда Scanner проверяет хост, который не отвечает.

Retries - Количество попыток проверки не отвечающего хоста.

4. Нажмите кнопку Start.

Утилита Scanner сканирует диапазон IP адресов. Результат сканирования появляется в области Hosts.

В течение сканирования кнопка Start превращается в Stop. Вы можете нажать Stop в любое время, чтобы остановить процесс сканирования.

Замечание:

Проверка статуса хостов может значительно увеличить время, требующееся для завершения сканирования.

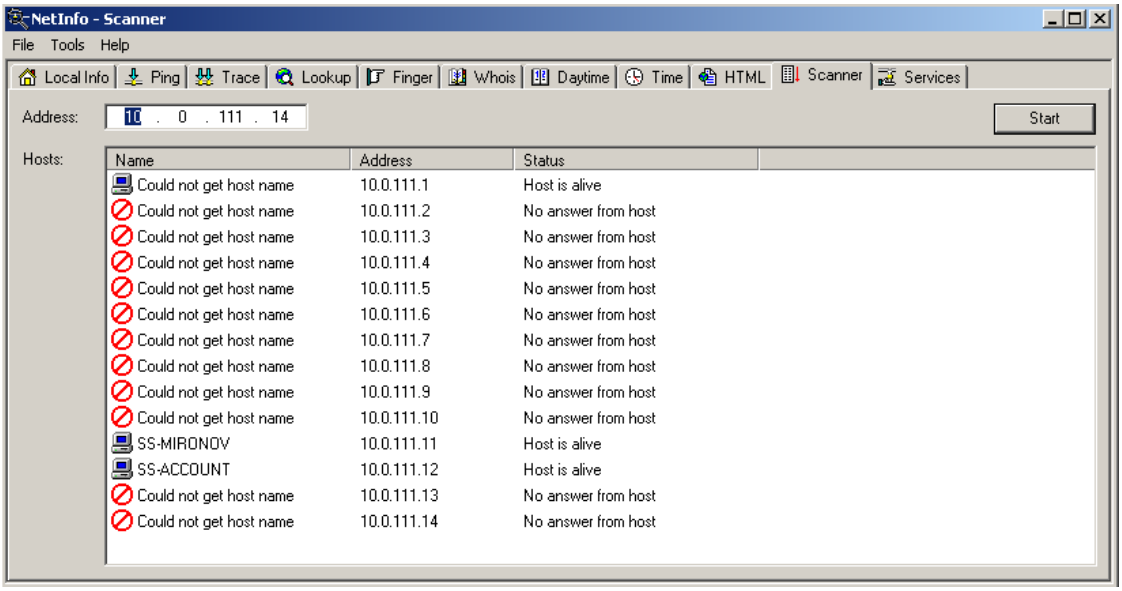

Рис.17. Пример использования утилиты Scanner.

Утилита Services

Сетевая диагностическая утилита, которая проверяет статус сервисов хоста.

Чтобы запустить Services, сделайте следующее:

1. Выберете вкладку Services.
2. В текстовом поле Host введите имя или IP адрес удаленного (например, www.mit.edu).
3. В диалоговом окне Options находится набор сервисов, которые вы хотите проверить.
4. Нажмите кнопку Check.

Утилита Services проверяет статус сервисов хоста. Результат сканирования появляется в области Response.

В течение проверки, кнопка Check превращается в Stop. Вы можете нажать Stop в любое время для завершения проверки.

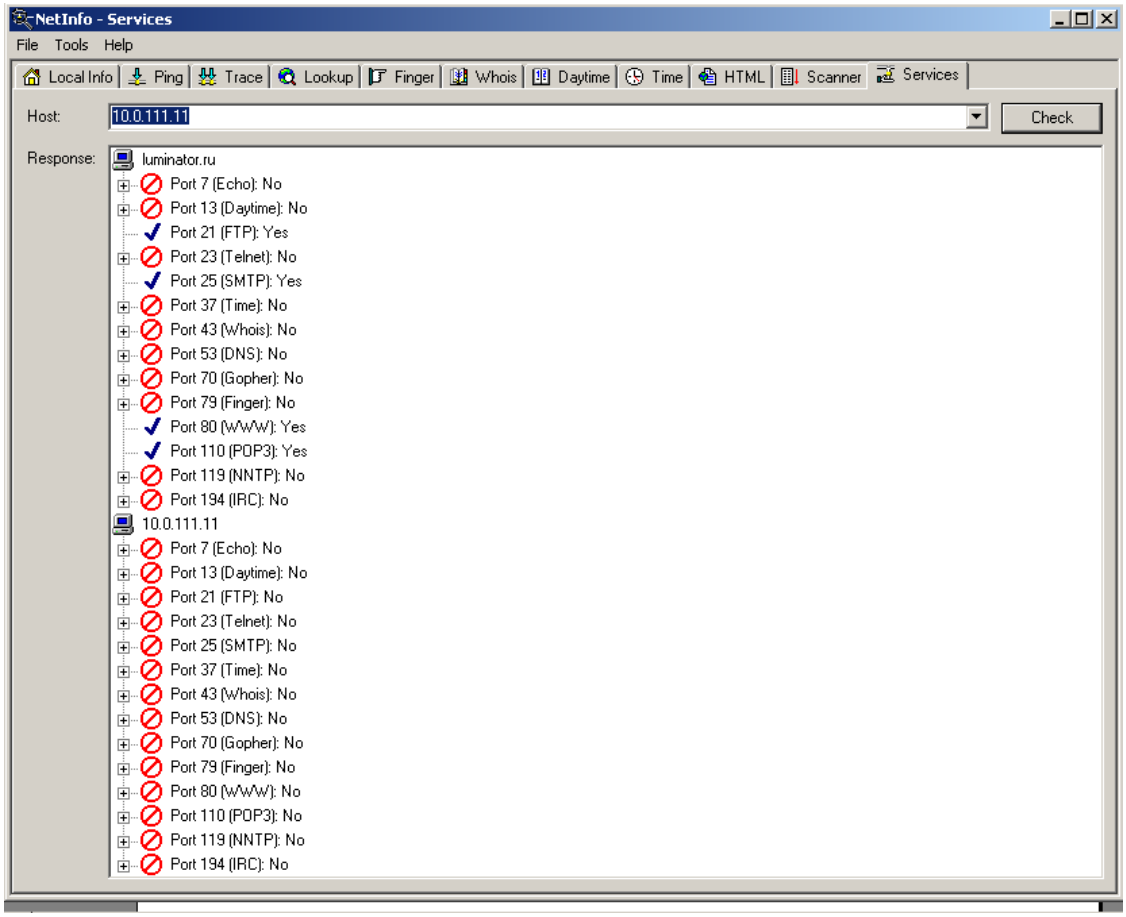

Рис.18. Пример использования утилиты Services:

Утилита IPMonitor

Утилита системного трея включена как часть NetInfo для наблюдения за включением / выключением компьютеров. IPMonitor проверяет сетевую доступность списка хостов, определенных пользователем и предупреждает вас об ошибках, используя аудио сигналы и оповещение с помощью иконок.

Чтобы создать список хостов, сделайте следующее:

1. Запустите IPMonitor.
2. Нажмите правой кнопкой на иконку IPMonitor в системном трее и в выпадающем меню выберете Details.
3. Нажмите кнопку Add. В текстовом поле Host, введите имя хоста или его IP адрес (например, www.netscape.com). В поле Description введите описание для этого хоста (например, Netscape Netcenter). Также вы можете использовать следующие опции:

Interval - Количество минут, которое IPMonitor ждет до следующей проверки.

Timeout - Количество секунд, когда Scanner проверяет хост, который не отвечает.

Retries - Количество попыток проверки не отвечающего хоста.

4. Нажмите кнопку OK.

IPMonitor сразу начинает проверять хосты. Результаты проверки появляются в области Details.

5. Повторите пункты 3 и 4, чтобы еще добавить хосты к списку.
6. Нажмите кнопку Close.

Примечание: Вы можете два раза кликнуть по иконке IPMonitor в трее, чтобы появилось диалоговое окно Details. Вы можете нажать правой кнопкой в поле Details, чтобы появилось выпадающее меню.

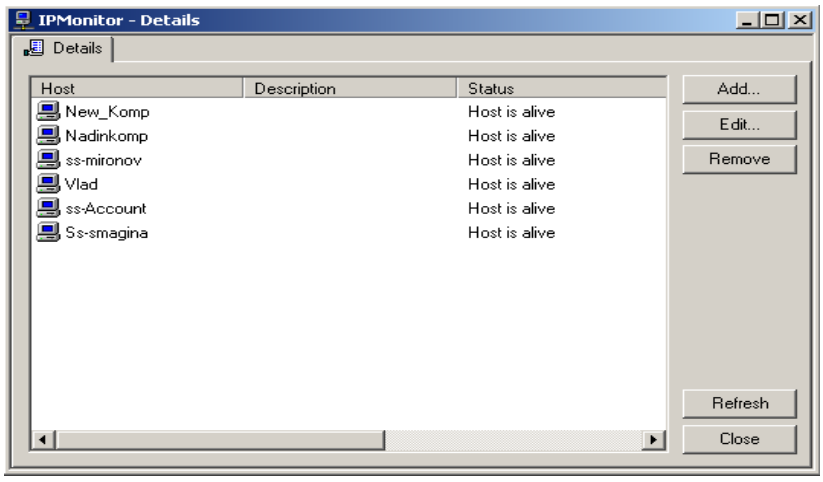

Рис.19. Пример использования утилиты IPMonitor:

5. Изучение пакетного анализатора Wireshark.

Пакетный анализатор Wireshark предназначен для отслеживания трафика, проходящего через сетевой интерфейс узла. Программа предоставляет возможность анализа широкого набора сетевых протоколов и позволяет динамически отображать на экране информацию о процессе поступления и отсылки пакетов. Помимо этих возможностей программа предоставляет различные средства статистического анализа просмотренного трафика.

Основное окно программы

На рис.20 представлено основное окно программы. Оно разделено на несколько функциональных областей.

Область 1 содержит набор кнопок, управляющих работой программы. Наибольшее значение имеет первая группа кнопок.

Interfaces - открывает список доступных для анализа сетевых интерфейсов. В этом окне можно выбрать интерфейс, который будет просматриваться программой.

Options – открывает окно настройки анализа (см. далее).

Start – запуск анализа.

Stop – останов.

Restart – очистка результатов и перезапуск анализа.

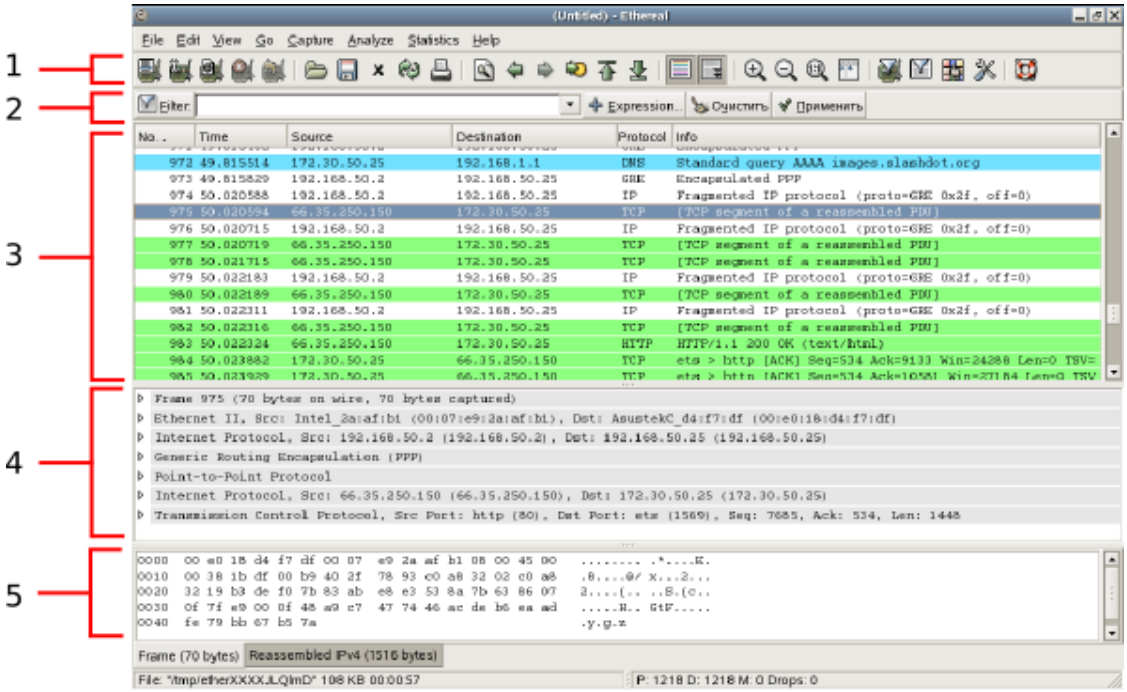

Рис.20. Главное окно Wireshark.

Область 2 – строка настройки фильтра. Wireshark позволяет выделять из потока только пакеты интересующих пользователя протоколов. Для обозначения правил фильтрации используется развитый язык выражений, позволяющий выбирать необходимые протоколы, вводить ограничения на содержимое пакетов, комбинировать условия с помощью логически операций и пр. (см. далее).

В указанной области находится кнопка Filters, открывающая окно выбора одного из предопределённых фильтров и создания собственных фильтров. Далее следует строка ввода, позволяющая быстро применить новое условие фильтрации к текущему анализу. Кнопка Expression открывает окно конструктора выражений фильтрации. Здесь представлен широкий набор возможных выражений и операций над ними. Кнопки Apply и Clear служат для активации и очистки выражения в строке ввода.

В область 3 выводится основная информация. Здесь отображается список отслеженных пакетов, прошедших через сетевой интерфейс. Для каждого пакета указывается информация о времени прохождения, адресах источника и приёмника, протоколе и краткое описание его содержимого.

При выделении в списке одного из пакетов в области 4 отображается информация о его содержимом. Она структурирована по всем протоколам, инкапсулирующим передаваемые данные. В раскрывающихся списках содержатся названия и значения полей заголовков протоколов. Выделение одного из полей в таком списке сопровождается подсветкой соответствующей информации в области 5. Здесь отображается необработанное содержимое пакета в шестнадцатеричном виде.

Настройка параметров анализа

Рассмотрим процесс настройки работы анализатора (рис. 21).

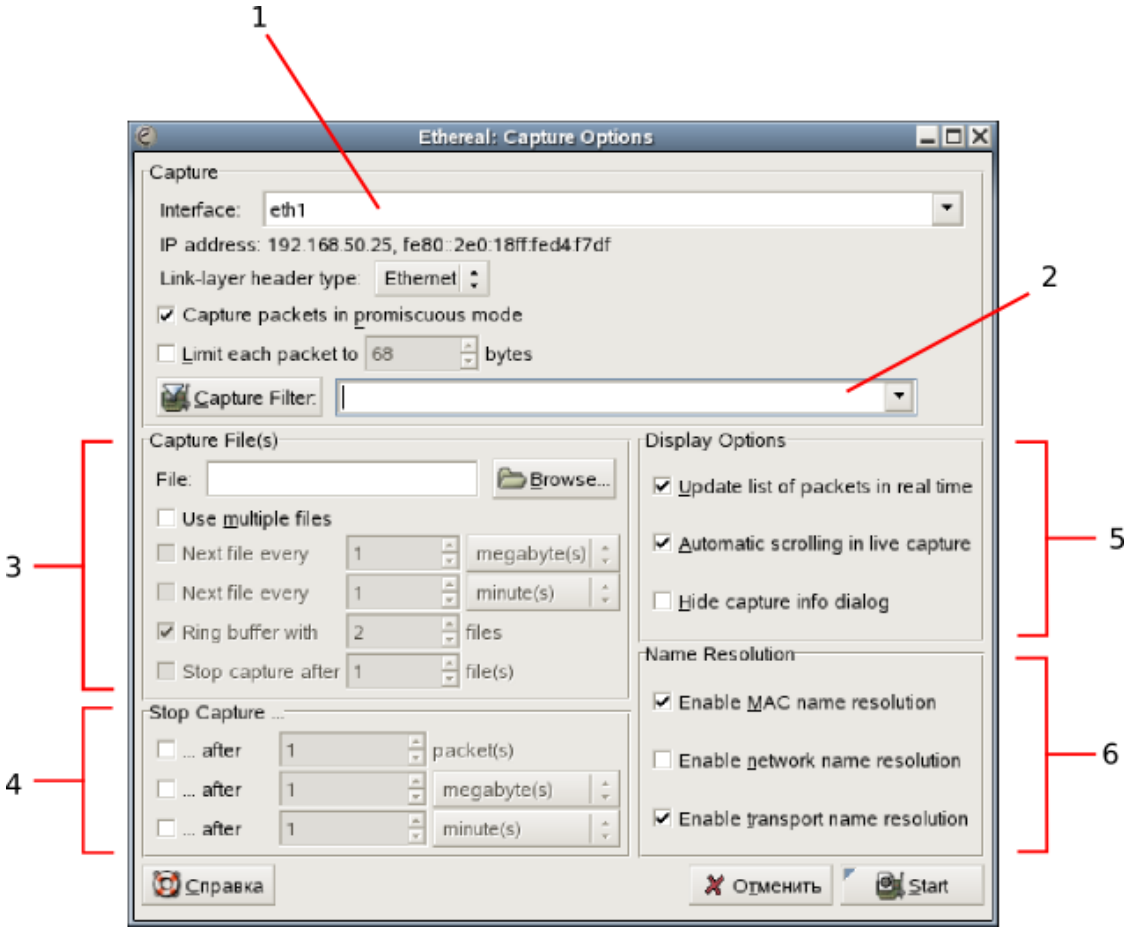

Рис. 21. Окно настройки параметров анализа.

Поле 1 позволяет выбрать просматриваемый интерфейс. В поле 2 можно задать необходимый фильтр (это можно сделать и позже, в процессе анализа). В области 3 можно задать имя файла для сохранения результатов анализа и настроить параметры автоматического сохранения. Область 4 позволяет задать условия прекращения работы программы. Область 5 содержит следующие опции:

Update list of packets in real time – включить динамическое обновление списка просмотренных пакетов;

Automatic scrolling in live capture – автоматическая прокрутка списка пакетов во время работы анализатора;

Hide capture info dialog – скрывать окно, содержащее статистику по количеству просмотренных пакетов.

Область 6 позволяет настроить преобразование имён узлов в символическое представление.

В нижней части окна располагается кнопка Start, запускающая анализатор.

Фильтрация пакетов

Wireshark обладает развитым механизмом задания параметров фильтрации. Требуемые параметры задаются с помощью строки, содержащей выражение особой формы. Выражение состоит из одного или более условий, связанных логическими операциями. В качестве условий могут выступать названия протоколов и ограничения на значения отдельных полей пакетов. Использование названия протокола означает разрешение на просмотр всех пакетов данного протокола, в том числе и инкапсулирующих данные других протоколов. Для задания ограничений на отдельное поле пакета требуется указать его имя в форме <имя_протокола>.<имя_поля> и определить отношение его значения к определённому значению с помощью знаков >, <, >=, <=, ==, !=, contains, matches present.

Условия комбинируются с помощью логических операторов and и or. Употребление ключевого слова not перед условием позволяет инвертировать его значение. Допускается использование скобок.

Пример: tcp and (not http) and (ip.ttl >= 10)

(все пакеты TCP, не содержащие пакетов HTTP и имеющие значения поля TTL заголовка IP не менее 10).

Граф анализа

Программа предоставляет пользователю набор средств автоматизированного анализа полученной информации, среди которых инструмент Flow Graph (меню Statistics). Он позволяет построить диаграмму перемещения пакетов по узлам. На рис. 22 представлен пример такой диаграммы, полученный по результатам анализа работы утилиты traceroute.

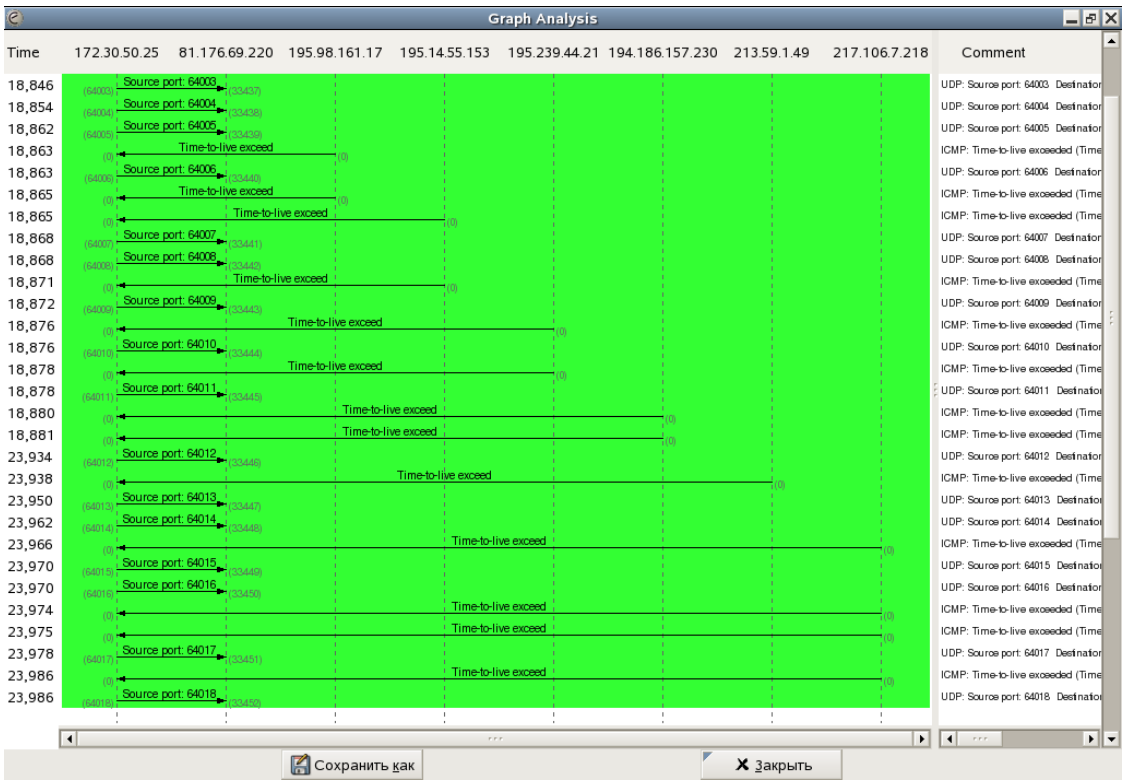

Рис. 22. Граф анализа работы программы traceroute.

Порядок работы с приложением.

Обычно для выполнения анализа трафика следует выполнить следующие действия.

1. Открыть окно настройки параметров анализа;
2. Выбрать требуемый сетевой интерфейс из списка обнаруженных;
3. Ввести необходимые условия фильтрации.
4. Если требуется, выбрать режим автоматического сохранения информации, условия останова работы анализатора;
5. Установить флаги Update list of packets in real time и Automatic scrolling in live capture;
6. Нажать кнопку Start.
7. Отслеживать обрабатываемый трафик в основном окне.
8. При необходимости остановить анализ, сменить параметры фильтрации и продолжить работу.
9. Остановить анализ, когда вся необходимая информация представлена на экране.
10. Анализировать информацию, просматривая содержимое пакетов и используя встроенные средства анализа.

6. Порядок выполнения работы.

Лабораторная работа №7.

Исследование протоколов сетевого уровня IP-сетей.

Просканировать порты удалённого хоста, и зафиксировать и описать службы на открытых портах. Сделать запросы, соответствующие варианту. Объяснить реакцию хоста на каждый запрос. Результаты анализа представить в отчете.

Отчет по лабораторной работе должен содержать сценарии выполнения вышеуказанных процедур, описание служб на открытых портах удалённого хоста, описание реакции на запросы.

Контрольные вопросы.

1. Какое назначение имеет протокол IP?
2. Как работает утилита ping?
3. При помощи какой утилиты можно получить информацию обо всех пользователях сетевого узла?

Лабораторная работа №8.

Исследование протоколов сетевого уровня IP-сетей.

В соответствии с вариантом либо послать ICMP эхо-запрос на удалённый хост либо определить количество «хопов» до удалённого хоста. При помощи пакетного анализатора проанализировать все пакеты, приходящие на сетевой интерфейс. Для нечётных вариантов просматривать ICMP трафик, а для чётных ICMP и UDP трафик. Результаты анализа представить в отчете.

Отчет по лабораторной работе должен содержать сценарии выполнения вышеуказанных процедур, результаты анализа пакетов, описание реакции на запросы.

Контрольные вопросы.

1. Какое назначение имеет протокол IP?
2. Как работает утилита ping?
3. При помощи какой утилиты можно получить информацию обо всех пользователях сетевого узла?
4. Как осуществляется настройка фильтрации пакетов в пакетном анализаторе?
5. Как работает утилита traceroute?

Лабораторная работа №9.

Исследование протоколов транспортного уровня IP-сетей.

Инициировать TCP сеанс с хостом, указанным в варианте, и проанализировать его открытие, поддержку и закрытие при помощи пакетного анализатора. Перенести в отчет описание процедур открытия, поддержания и закрытия TCP соединения посредством обмена заголовками протокола TCP для этого сеанса. Найти и перенести в отчет все пакеты, отвечающие за вышеуказанные события.

По своей сути TCP является протоколом транспортного уровня. Он позволяет осуществить соединение одного сокета (IP-адрес + порт) хоста источника с сокетом хоста назначения. Заголовок IP будет содержать информацию, связанную с IP-адресами, а заголовок TCP — информацию о порте.

Заголовок TCP

Заголовки TCP перемещаются по сети для установления, поддержки и завершения TCP-соединений, а также передачи данных.

TCP Header

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Source port																Destination port															
4	32	Sequence number																															
8	64	Acknowledgment number (if ACK set)																															
12	96	Data offset				Reserved 0 0 0				NS	CWR	ECE	URG	ACK	PSH	RST	SYN	FIN	Window Size														
16	128	Checksum																Urgent pointer (if URG set)															
20 ...	160 ...	Options (if *data offset* > 5. Padded at the end with "0" bytes if necessary.) ...																															

Рисунок 23. Заголовок TCP

В заголовке TCP содержаться следующие поля:

- Source port (16 бит): порт источника. Порт хоста, от которого исходит запрос.

- Destination port (16 бит): порт назначения. Порт хоста, куда направляется запрос.

- Sequence number, SYN (32 бита): порядковый номер

- Acknowledgement number, ACK (32 бита): номер подтверждения. Когда сообщение содержит флаг ACK, то значение в номере подтверждения должно соответствовать следующему порядковому номеру (SYN), которое отправитель сообщения с флагом ACK ожидает получить от передающей системы

Механизм передачи сообщений TCP

Перед тем, как данные могут быть переданы между двумя узлами, в TCP, в отличие от

UDP, предусмотрена стадия установки соединения. Также, после того, как все данные были переданы, наступает стадия завершения соединения. Таким образом, осуществление каждого TCP-соединения можно условно разделить на три фазы:

Инициализация соединения.

Установка соединения осуществляется с помощью, так называемого трехстороннего рукопожатия TCP. Инициатором соединения может выступать любая сторона. Однако, чтобы упростить рассмотрения данного вопроса в рамках данной статьи, мы рассмотрим пример, когда клиент инициализирует соединение с сервером.

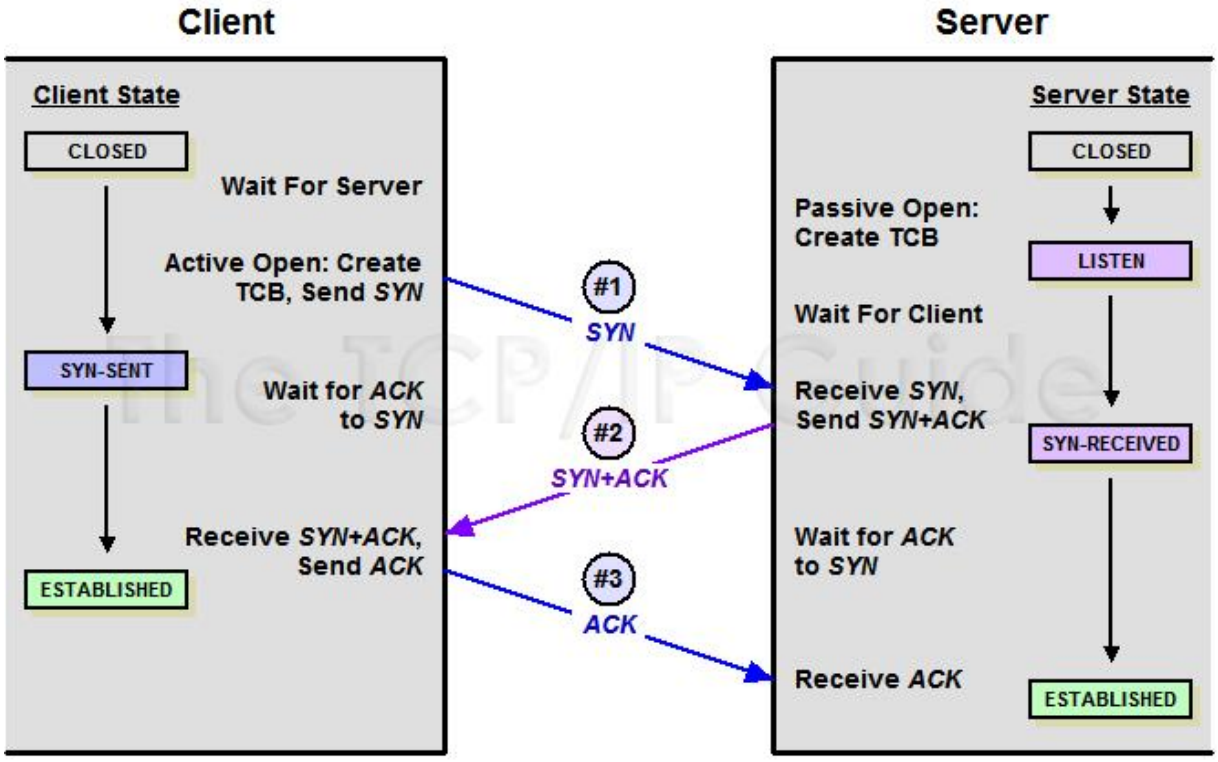

Рисунок 24. Трехстороннее рукопожатие TCP

(Пакет №1). Клиент отправляет пакет с установленным флагом SYN и случайным числом («R1»), включенным в поле порядкового номера (sequence number).

(Пакет №2). При получении пакета №1 сервер в ответ отправляет пакет с установленным флагом SYN, а также с установленным флагом ACK. Поле порядкового номера будет содержать новое случайное число («R2»), а поле номера подтверждения будет содержать значение порядкового номера клиента, увеличенного на единицу (то есть «R1 + 1»). Таким образом, он будет соответствовать следующему порядковому номеру, который сервер ожидает получить от клиента.

(Пакет №3). В ответ на пакет SYN от сервера (пакет №2) клиент отправляет пакет с установленным флагом ACK и полем номера подтверждения с числом «R2 + 1». По

аналогии, это число будет соответствовать следующему порядковому номеру, который клиент ожидает получить от сервера.

Загрузка данных.

После инициализации соединения полезная нагрузка будет перемещаться в обоих направлениях TCP-соединения. Все пакеты в обязательном порядке будут содержать установленный флаг ACK. Другие флаги, такие как, например, PSH или URG, могут быть, а могут и не быть установленными.

Завершение соединения.

При нормальном завершении TCP-соединения в большинстве случаев инициализируется процедура, называемая двухсторонним рукопожатием, в ходе которой каждая сторона закрывает свой конец виртуального канала и освобождает все задействованные ресурсы.

Разница между TCP и UPD

Сравнивая оба протокола, очевидно, что протокол TCP – это, можно сказать, «снайпер». Прицелился, выстрелил, зафиксировал попадание, ищет следующую цель. UDP – это, скорее, «пулеметчик» - выставил ствол в направлении врага и начал долбить очередями, не слишком заботясь о точности. Как в войсках важны обе эти воинские специальности, так и в интернете важны оба этих протокола. TCP применяется там, где требуется точная и подтверждаемая передача данных – например, отправка фотографий, или переписка между пользователями. UDP, в свою очередь, нужен для общения в голосовом формате, или при передаче потокового видео, например, с веб-камер или IP-камер.

Контрольные вопросы.

1. По каким флагам заголовка протокола TCP можно идентифицировать фазу TCP соединения?
2. Как осуществляется настройка фильтрации пакетов в пакетном анализаторе?
3. Как работает утилита traceroute?
4. Сколько фаз проходит TCP соединение?

7. Варианты заданий.

ИУ5-51

Вариант	Удалённый хост	Ping	Traceroute	Проверка работающих сервисов	Запрос
1	bmstu.ru	+		+	who-is, HTML

2	yandex.ru		+	+	who-is, HTML
3	google.com	+		+	lookup, HTML
4	mail.ru		+	+	who-is, lookup
5	hh.ru	+		+	who-is. HTML
6	habr.com		+	+	lookup, HTML
7	rambler.ru	+		+	who-is, lookup
8	vk.com		+	+	who-is. HTML
9	facebook.com	+		+	who-is, lookup
10	twitter.com		+	+	lookup, HTML
11	livejournal.com	+		+	who-is. HTML
12	d3.ru		+	+	who-is, lookup
13	tjournal.ru	+		+	lookup, HTML
14	design.ru		+	+	who-is, lookup
15	gmail.com	+		+	who-is, lookup
16	avito.ru		+	+	who-is. HTML
17	gismeteo.ru	+		+	who-is, lookup
18	rbk.ru		+	+	who-is, HTML
19	wikipedia.org	+		+	lookup, HTML
20	kinopoisk.ru		+	+	who-is, lookup
21	apple.com	+		+	who-is. HTML
22	afisha.ru		+	+	lookup, HTML
23	youtube.com	+		+	who-is, lookup
24	instagram.com		+	+	who-is, HTML
25	amazon.com	+		+	lookup, HTML

ИУ5-52

Вариант	Удалённый хост	Ping	Traceroute	Проверка работающих сервисов	Запрос
1	bmstu.ru	+		+	who-is, HTML
2	yandex.ru		+	+	who-is, HTML
3	google.com	+		+	lookup, HTML
4	mail.ru		+	+	who-is, lookup
5	hh.ru	+		+	who-is. HTML
6	habr.com		+	+	lookup, HTML
7	rambler.ru	+		+	who-is, lookup
8	vk.com		+	+	who-is. HTML
9	facebook.com	+		+	who-is, lookup
10	twitter.com		+	+	lookup, HTML
11	livejournal.com	+		+	who-is. HTML
12	d3.ru		+	+	who-is, lookup
13	tjournal.ru	+		+	lookup, HTML
14	design.ru		+	+	who-is, lookup
15	gmail.com	+		+	who-is, lookup
16	avito.ru		+	+	who-is. HTML

17	gismeteo.ru	+		+	who-is, lookup
18	rbk.ru		+	+	lookup, HTML
19	wikipedia.org	+		+	who-is. HTML
20	kinopoisk.ru		+	+	who-is, lookup
21	apple.com	+		+	lookup, HTML
22	afisha.ru		+	+	lookup, HTML
23	youtube.com	+		+	who-is, lookup
24	instagram.com		+	+	who-is, HTML
25	amazon.com	+		+	lookup, HTML

ИУ5-53

Вариант	Удалённый хост	Ping	Traceroute	Проверка работающих сервисов	Запрос
1	bmstu.ru	+		+	who-is, HTML
2	yandex.ru		+	+	who-is, HTML
3	google.com	+		+	lookup, HTML
4	mail.ru		+	+	who-is, lookup
5	hh.ru	+		+	who-is. HTML
6	habr.com		+	+	lookup, HTML
7	rambler.ru	+		+	who-is, lookup
8	vk.com		+	+	who-is. HTML
9	facebook.com	+		+	who-is, lookup
10	twitter.com		+	+	lookup, HTML
11	livejournal.com	+		+	who-is. HTML
12	d3.ru		+	+	who-is, lookup
13	tjournal.ru	+		+	lookup, HTML
14	design.ru		+	+	who-is, lookup
15	gmail.com	+		+	who-is, lookup
16	avito.ru		+	+	who-is. HTML
17	gismeteo.ru	+		+	who-is, lookup
18	rbk.ru		+	+	lookup, HTML
19	wikipedia.org	+		+	who-is. HTML
20	kinopoisk.ru		+	+	who-is, lookup
21	apple.com	+		+	lookup, HTML
22	afisha.ru		+	+	time, whois
23	youtube.com	+		+	who-is, lookup
24	instagram.com		+	+	who-is. HTML
25	amazon.com	+		+	who-is, lookup

ИУ5-54

Вариант	Удалённый хост	Ping	Traceroute	Проверка работающих сервисов	Запрос
1	bmstu.ru	+		+	who-is, HTML
2	yandex.ru		+	+	who-is, HTML
3	google.com	+		+	lookup, HTML
4	mail.ru		+	+	who-is, lookup
5	hh.ru	+		+	who-is. HTML
6	habr.com		+	+	lookup, HTML
7	rambler.ru	+		+	who-is, lookup
8	vk.com		+	+	who-is. HTML
9	facebook.com	+		+	who-is, lookup
10	twitter.com		+	+	lookup, HTML
11	livejournal.com	+		+	who-is. HTML
12	d3.ru		+	+	who-is, lookup
13	tjournal.ru	+		+	lookup, HTML
14	design.ru		+	+	who-is, lookup
15	gmail.com	+		+	who-is, lookup
16	avito.ru		+	+	who-is. HTML
17	gismeteo.ru	+		+	who-is, lookup
18	rbk.ru		+	+	lookup, HTML
19	wikipedia.org	+		+	who-is. HTML
20	kinopoisk.ru		+	+	who-is, lookup
21	apple.com	+		+	who-is. HTML
22	afisha.ru		+	+	lookup, HTML
23	youtube.com	+		+	who-is, lookup
24	instagram.com		+	+	who-is, HTML
25	amazon.com	+		+	lookup, HTML

ИУ5-55

Вариант	Удалённый хост	Ping	Traceroute	Проверка работающих сервисов	Запрос
1	bmstu.ru	+		+	who-is, HTML
2	yandex.ru	+		+	who-is, HTML
3	google.com		+	+	lookup, HTML
4	mail.ru		+	+	who-is, lookup
5	hh.ru	+		+	who-is. HTML
6	habr.com	+		+	lookup, HTML
7	rambler.ru		+	+	who-is, lookup
8	vk.com		+	+	who-is. HTML
9	facebook.com	+		+	who-is, lookup
10	twitter.com	+		+	lookup, HTML
11	livejournal.com		+	+	who-is. HTML
12	d3.ru		+	+	who-is, lookup

13	tjournal.ru	+		+	lookup, HTML
14	design.ru	+		+	who-is, lookup
15	gmail.com		+	+	who-is, lookup
16	avito.ru		+	+	who-is. HTML
17	gismeteo.ru	+		+	who-is, lookup
18	rbk.ru	+		+	lookup, HTML
19	wikipedia.org		+	+	who-is. HTML
20	kinopoisk.ru		+	+	who-is, lookup
21	apple.com	+		+	who-is. HTML
22	afisha.ru	+		+	lookup, HTML
23	youtube.com		+	+	who-is, lookup
24	instagram.com		+	+	who-is, HTML
25	amazon.com	+		+	lookup, HTML

РТ5-51

Вариант	Удалённый хост	Ping	Traceroute	Проверка работающих сервисов	Запрос
1	bmstu.ru		+	+	who-is, HTML
2	yandex.ru	+		+	who-is, HTML
3	google.com		+	+	lookup, HTML
4	mail.ru	+		+	who-is, lookup
5	hh.ru		+	+	who-is. HTML
6	habr.com	+		+	lookup, HTML
7	rambler.ru		+	+	who-is, lookup
8	vk.com	+		+	who-is. HTML
9	facebook.com		+	+	who-is, lookup
10	twitter.com	+		+	lookup, HTML
11	livejournal.com		+	+	who-is. HTML
12	d3.ru	+		+	who-is, lookup
13	tjournal.ru		+	+	lookup, HTML
14	design.ru	+		+	who-is, lookup
15	gmail.com		+	+	who-is, lookup
16	avito.ru	+		+	who-is. HTML
17	gismeteo.ru		+	+	who-is, lookup
18	rbk.ru	+		+	lookup, HTML
19	wikipedia.org		+	+	who-is. HTML
20	kinopoisk.ru	+		+	who-is, lookup
21	apple.com		+	+	who-is. HTML
22	afisha.ru	+		+	lookup, HTML
23	youtube.com		+	+	who-is, lookup
24	instagram.com	+		+	who-is, HTML
25	amazon.com		+	+	lookup, HTML