

Министерство образования и науки Российской Федерации
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра сверхвысоких частот и квантовой радиотехники
(СВЧ и КР)

**ИССЛЕДОВАНИЕ СЕТЕВЫХ ПРОТОКОЛОВ ПЕРЕДАЧИ
ДАНЫХ**

Томск-2011

Министерство образования и науки Российской Федерации

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра сверхвысоких частот и квантовой радиотехники
(СВЧ и КР)

УТВЕРЖДАЮ
Зав. Кафедрой СВЧ и КР
_____ С.Н. Шарангович
“ ____ ” _____ 2011 г.

**ИССЛЕДОВАНИЕ СЕТЕВЫХ ПРОТОКОЛОВ ПЕРЕДАЧИ
ДАНЫХ**

РУКОВОДСТВО
к компьютерной лабораторной работе по дисциплине
“Волоконно-оптические локальные сети и системы
кабельного телевидения ”
для студентов специальности
210401 “Физика и техника оптической связи”

Разработчики
Доцент кафедры СВЧ и КР
_____ Н.Д. Хатьков

ассистент кафедры СВЧ и КР
_____ Д.Г. Осетров

1 ВВЕДЕНИЕ

Цель работы - на примере готовых программных модулей, изучить работу различных сервисов и протоколов на реально - действующей сети. Работа сетей будет дополнена полученными реакциями воздействующих команд модулей на сеть в процессе обмена данными с серверами.

2 ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ

Из истории вопроса о передаче данных с помощью новой технологии «коммутации пакетов» (packet switching). Эта технология, в которой данные, предназначенные для отправки в другое место, разбивались на пакеты, каждый из которых имел свой «адрес назначения» («forwarding address»). Это давало возможность нескольким пользователям одновременно работать по одной и той же линии связи. Не менее важным с точки зрения оборонного ведомства было то, что эта технология обладала возможностью автоматической маршрутизации данных по различным линиям связи. Целью АКРА было развитие надежной сети передачи данных, которая могла бы выдержать даже ядерное нападение, когда любой компьютер или линия связи между компьютерами может перестать функционировать в любой момент времени. Поэтому от создаваемой сети требовалась устойчивость работы: часть сети может быть разрушена без ущерба для функционирования сети в целом. ARPANet (так была названа эта новая сеть) эту задачу решила. Пакетная система передачи данных использовала «скоростные магистрали», по которым огромное число пакетов движутся в одном и том же направлении. Каждому пакету выдается компьютерный эквивалент адреса получателя, благодаря которому все пакеты, возможно, разными путями дойдут до получателя, где их снова соберут в сообщение, понятное человеку или компьютеру. Первым революционным применением созданной сети, сразу обеспечившей ей успех, стала электронная почта, поскольку оказалось, что подобные электронные письма можно пересылать со скоростью телефонного звонка.

В 1969 году в качестве узлов новой сети были выбраны компьютеры четырех университетов: Калифорнийский университет в Лос-Анджелесе, Станфордский научно-исследовательский институт, Калифорнийский университет в Санта-Барбаре и Университет штат Юта. Получившаяся в результате сеть получила имя ARPANet. Несмотря на то, что эта разработка проходила по линии Министерства обороны, отнесена она была к общественной сфере. Это означало, что любая организация, компания или частное лицо могли получить техническую документацию, описывающую протоколы работы этой сети, и разрабатывать программы для этой технологии. Открытость системы неминусом должна была вызвать создание аналогичных, основанных на близких протоколах сетей.

Что со временем и произошло и «завершилось» их объединением в глобальную информационную систему.

По мере того, как ARPANet росла, группа энтузиастов разработала способ ее использования для проведения электронных конференций. Вначале это были научные дискуссии, но скоро конференции так разрослись, что охватывали уже практически все аспекты человеческой жизни. По сути телеконференции были развитием идеи электронной почты, только в этом случае письмо отправлялось не одному человеку, а сотням и даже тысячам «подписчиков» конференций на различные темы. В семидесятых годах наступил период развития локальных вычислительных сетей. Появились первые суперкомпьютеры и операционная система UNIX. В версию UNIX Калифорнийского университета в Беркли (BSD LUNIX) были включены, как часть операционной системы протоколы Интернет. Поскольку в университетах и научных кругах эта версия UNIX широко использовалась, это дало новый толчок к применению протоколов TCP/IP в локальных вычислительных сетях. Стали являться родственные ARPANet сети. Например, NASA создала свою сеть.

В семидесятых годах при поддержке ARPA были разработаны протоколы передачи данных между различными компьютерными сетями. К концу семидесятых годов были разработаны связи между ARPANet и ее атэтнорами (ATNET) в других странах. Эти протоколы сделали возможным объединение сетей во всемирную Сеть, которую мы сейчас имеем и которая соединяет компьютеры всех существующих видов теперь уже по всему миру.

В восьмидесятых годах эта Сеть сетей, которая стала известна под именем Интернет, развилась до невероятной степени. В 1982 году был создан единый протокол TCP/IP, объединяющий различные, уже действующие протоколы. Сотни, а потом и тысячи научных организаций, учебных заведений и правительственных ведомств стали подключать свои компьютеры к всемирной Сети. Со временем появились предприимчивые компании, которые стали предлагать доступ в Интернет для частных лиц. Теперь любой владелец компьютера мог купить модем и получить доступ к электронной почте и конференциям.

В девяностых годах продолжается экспоненциальный рост Сети. В ответ на это правительства и другие заинтересованные организации и лица непрерывно работают над расширением возможностей самой Сети. Когда-то основной канал передачи данных Сети в Соединенных Штатах, как его называют «хребет» (backbone), передавал данные со скоростью 56000 бит в секунду. По причине все возрастающего объема пересылаемых данных скорость его работы подняли сначала до 1,5 миллиона, затем до 45 миллионов бит в секунду и т.д. Другое серьезное новшество, способствовавшее развитию Интернет — это появление коммерческих служб, предоставляющих услуги по скоростному

межсетевому обмену. В середине 1994 года правительство Соединенных Штатов фактически отошло от ежедневного управления работой Сети, переложив эту работу на местных и национальных поставщиков услуг Интернет.

Сеть работает по различным протоколам в том числе и по ТСР/IP. Протокол это структура, описывающая формат пакета данных, передаваемого по сети. Пакет представляет собой поток битов, и именно протокол определяет, где располагается адрес и другая служебная информация, а не сами передаваемые данные. Протокол ТСР предназначен для контроля целостности передаваемой информации. Компьютеры обмениваются пакетами протокола IP и контролируют их передачу по протоколу ТСР. Пакеты IP, пришедшие с ошибками, или по каким-то причинам потерянные, посылаются повторно. Такова, в самых общих чертах, техническая сторона вопроса. Протокол ТСР/IP позволяет передавать информацию, а его, в свою очередь, используют разнообразные сервисы или службы Интернет по доставке и обработке информации. От эффективности этих сервисов, простоты и удобства зависит целиком вся эффективность работы систем связи. Интернет замечателен тем, что сеть и ее сервисы стали широко распространены в жизни общества, что они оказались достаточно хороши, чтобы значительная часть информации передавалась и размещалась в Интернет. Интернет не решил проблему хранения и упорядочения информации, но решил проблему ее передачи и отчасти поиска. В Интернет можно получить любую информацию где угодно, когда угодно и, вообще говоря, кому угодно. Трудность состоит только в том, чтобы найти в Сети то, что вам нужно. Эта замечательная способность передачи информации привела к грандиозным темпам роста Интернет на протяжении всей его истории. Причем темпы роста не только не замедлились, но продолжают увеличиваться, особенно это касается нашей страны. Вследствие децентрализованности Интернет трудно точно сказать, каковы сейчас размеры Сети. Сегодня Интернет объединяет множество разных сетей, миллионы компьютеров, более 100 миллионов пользователей всех континентов и, по разным оценкам, число таких пользователей увеличивается на 50-80% ежегодно. Около 10 миллионов пользователей проживают в России. Российский Интернет растет еще быстрее. В российском Интернет зарегистрировано свыше 10 миллионов документов. За 1999 год объем российского Интернета вырос в 16 раз. Примерно в 4 раза увеличилось количество серверов и в среднем в 4 раза увеличился средний объем сервера за этот же период. Помимо увеличения технических показателей, Интернет развивается качественно. Появляются новые услуги, такие как, например, электронная коммерция. Высшая власть в Интернет принадлежит ISOC (Internet Society). ISOC — общество с добровольным членством, цель которого — способствовать глобальному обмену информацией через Интернет. Оно назначает «совет

старейшин», который отвечает за техническую политику, поддержку и управление Сетью. Этот Совет представляет собой группу приглашенных лиц, которые добровольно изъявили желание принять участие в его работе. Называется он IAB (Internet Architecture Board — Совет по архитектуре Интернет.). IAB регулярно собирается, чтобы рассмотреть и утвердить стандарты и распределить ресурсы, такие, например, как адреса. IAB ответственен за стандарты. Он решает, насколько необходим тот или иной стандарт и каким ему следует быть. Пользователи Интернет высказывают свои жалобы и предложения на встречах IETF (Internet Engineering Task Force). IETF - это другая добровольная организация, которая регулярно собирается, чтобы обсудить текущие эксплуатационные и назревающие технические проблемы. При обсуждении достаточно важной проблемы IETF создает рабочую группу добровольцев для ее дальнейшего исследования. Рабочие группы выполняют различные функции: выпуск документации, стратегические исследования, разработку новых стандартов и протоколов, изменение существующих стандартов.

Стандарты Интернет описываются в документах под названием Request for Comments (RFC), имеющих свою нумерацию. Например, RFC 1800 и т.д. Поскольку Сеть является открытым и постоянно меняющимся образованием, то и не все из этих документов являются стандартами. Некоторые из них носят чисто информационное назначение, другие являются, по сути форумми, предоставляющими место для обмена информацией, со временем эти рекомендательные документы могут стать новыми стандартами Сети.

На примере готовых программных модулей, изучить работу различных сервисов и протоколов на реально действующей сети. Работа сетей будет дополнена полученными реакциями воздействующих команд модулей на сеть в процессе обмена данными с серверами. Набор заданий к лабораторной работе.

Практическое задание каждого раздела в целом предполагает наличие работающей сети, подключенной к интернету. Поэтому, прежде чем приступить к выполнению заданий, необходимо убедиться в его наличии. Кроме того, настройки сети ее администратором, предполагают запрет на использование некоторых протоколов передачи данных, разнообразных запросов сетевых ресурсов и др. Это часто связано с возможными внешними внутренними информационными угрозами, дестабилизирующими устойчивую работу сети. В связи с этим, при обнаружении неработоспособности исследуемого протокола, необходимо указать в отчете к лабораторной работе сообщения, которые выдавала компьютерная программа и кратко описать ее поведение, после этого переходить к другому разделу. Как видно из предоставленного перечня элементов лабораторной работы помимо протоколов, исследуется, как основные сервисы, так и кодирование данных, что является

существенным дополнением к пониманию в целом работы компьютерной сети.

2.1 Получение адреса в сети

Одной из самых важных тем при рассмотрении TCP/IP является адресация IP. Адрес IP - числовой идентификатор, приписанный каждому компьютеру или устройству в сети IP и обозначающий местонахождение этого устройства в сети. Адрес IP состоит из четырех байтов, каждый байт представляет свой раздел адреса. Обычно IP-адреса записываются десятичными числами, каждый байт отдельно от других и каждая часть отделяется точкой. Например — 192.168.1.12. Каждый байт адреса представляет свой раздел, — тем самым получается, что IP-адрес имеет иерархическую структуру. В качестве примера, можно вспомнить нумерацию томских телефонов, где первые две цифры обозначают «номер» АТС или, можно сказать, обозначают район города. На пример, 26 — Советский р-н. и т.д. Оставшиеся цифры телефонного номера идентифицируют абонента в рамках городского района. IP адреса также имеют двухуровневую иерархическую систему нумерации. В зависимости от значения первого байта первый байт, два или три первых байта обозначают номер сети, а оставшиеся байты — номер устройства внутри этой сети (см. таблицу).

Структура IP адреса

| Класс | Формат | Шаблон первых бит первого октета | Десятичные значения первого байта адреса сети | Максимальное число сетей | Максимальное число узлов в одной сети |
|-------|------------------------------|----------------------------------|---|--------------------------|---------------------------------------|
| A | сеть. узел. узел. узел | 0 | 1 — 127 | 127 | 16777216 |
| B | сеть. сеть. узел. узел | 10 | 128— 191 | 16384 | 65534 |
| C | сеть. сеть. сеть, узел | 110 | 192 — 223 | 2097152 | 254 |
| D | | 1110 | Групповой адрес | Групповой адрес | Групповой адрес |

Из таблицы видно, что существуют только четыре варианта адресов. Последний вариант предназначен для передачи сообщений группе сетевых устройств. Что же касается сетей, то сети делятся на три

класса. Видно, что сетей класса А может быть только 127, зато такие сети могут иметь довольно большое число узлов — до 16777216. Сетей класса С может быть значительно больше, зато машин в такой сети может быть не более 254. Адреса узлов, состоящие только из нулей или единиц, имеют специфическое назначение и устройствам сети не присваиваются. Если вы подключены к Интернет через локальную сеть небольшой компании, то по протоколу IP ваша сеть, с большой вероятностью, относится к классу С, т.е. из нее к Интернет могут подключиться не более 254 машин, если ваш провайдер услуг Интернет не наложит на это количество свое ограничение. Использовать IP-адрес напрямую при обращении к нужной вам машине неудобно — нужно запоминать числа, которые запомнить трудно. Если бы у машины было имя, то с именем было бы проще. К счастью, эта проблема давно решена, и, когда вы обращаетесь к WWW-серверам, вы используете обращение к компьютеру по имени.

Раньше соответствие между адресом и именем определялось из специального текстового файла -- hosts. Но со временем, когда количество машин в Internet значительно выросло, была создана специальная служба DNS (Domain Name System — служба именования доменов), позволяющая получать эту информацию по Сети от серверов DNS, и был создан соответствующий протокол ARP (Address Resolution Protocol).

Сегодня в Интернет применяется протокол IP, использующий для адреса компьютера 32 бита. Однако, учитывая все ускоряющийся рост сети, адресное пространство может просто закончиться, причем это событие прогнозируется в пределах ближайших десяти лет. Для решения этой проблемы разрабатывается протокол IP нового поколения — IPng, в котором для адреса будет использоваться 128 бит, что позволяет адресовать астрономическое количество объектов. Переход на новый протокол предполагается планомерно осуществить в оставшийся промежуток времени, но кто знает, сколь болезненным он окажется — ведь такие изменения требуют практически полной смены программного обеспечения и сетевого оборудования.

2.2 Протокол времени суток - Daytime Protocol

При работе с различными сервисами Интернет в некоторых сетевых приложениях (чаще всего это серверы) необходимо точное знание времени для синхронизации в получении и приема различного рода данных, команд. Для этого был разработан специальный протокол, предназначенный для получения даты и времени от специальных серверов даты и времени в Интернет и интранет в соответствии со стандартом RFC 867, называемым «Протоколом времени суток» (Daytime Protocol, 1983). Следует отметить, что в настоящее время, подобный протокол может использоваться практически только в сетях интранет из-за возможного использования данного протокола деструктивным программным

обеспечением. Поэтому в ряде случаев, выходя в Интернет, можно и не получить запрашиваемые данные от какого - либо указанного сервера. Кроме того некоторые администраторы сетей могут использовать разный набор адресов для портов. Вот некоторые адреса, с которых можно начать работу - www.netmastersllc.com (порт - 13), time.nist.gov (порт - 37). Приложения для Интернет часто используют понятие порта. Любому пользователю знакомы аппаратные порты компьютера, например, последовательные порты, к которым подключаются модемы, мыши, параллельные порты для подключения принтера или сканера и т.д. В данном случае имеются в виду программные порты, понятие, применяемое в Windows, на транспортном сетевом уровне. В переносном смысле можно сказать, что программы подключаются к таким портам для передачи друг другу необходимой информации. Портам приписываются номера. Некоторые из них исторически закреплены за определенными службами или протоколами. Например, порт 21 используется протоколом передачи файлов FTP, порт 110 - это собственность протокола POP3, протокола приема электронной почты и т.д. В некоторых случаях, когда номер порта неизвестен, используют автоматические сканеры портов. Ну, а Вам для наглядности в ряде случаев это делать можно будет вручную.

2.3 Семейство протоколов TCP/IP

Комплект протоколов TCP/IP (протокол управления передачей/межсетевой протокол — Transmission Control Protocol/Internet Protocol) — это объединение различных функций, управляемых точными, обязательными и основанными на соглашении правилами их выполнения, реализации и т.д. Международной организацией по стандартизации (International Standards Organization -создана эталонная модель OSI (Open Systems Interconnection) в качестве руководства и образца для открытого множества протоколов. Эта модель имеет семь уровней. Сокращенный вариант модели OSI носит название модель DOD и имеет четыре уровня. На рис.1 представлен набор протоколов TCP/IP и их отношение к уровням модели DOD. Модель DOD и семейство протоколов TCP/IP представлена в таблицах:

| | | | | | | | | | |
|---|--------|-----|------|------|-----|-----|------|----------|--------|
| Process/ Application (приложение) | Telnet | FTP | TFTP | SMTP | LPD | NFS | SNMP | X Window | Другие |
| Host to Host (Транспорт) | TCP | | | | | | UDP | | |

| | | | | |
|----------|------|-------|-----|------|
| Тип | IP | IP | IP | IP |
| Internet | ICMP | BootP | ARP | RARP |

| | | | | | |
|---------------------------------------|----------|------------|------|----------------|--------|
| Network Access (сетевой интерфейс) | Ethernet | Token Ring | FDDI | Frame relay | Другие |
|---------------------------------------|----------|------------|------|----------------|--------|

Протоколом уровня Интернет является Internet Protocol (IP). Именно этот протокол обслуживает такую важнейшую задачу, как маршрутизация. Остальные протоколы этого уровня выступают в качестве поддержки протокола IP. Кроме задачи маршрутизации IP-протокол обеспечивает единый сетевой интерфейс для протоколов верхних уровней, в результате чего программы приложений избавлены от необходимости использовать в программах конкретные методы сетевого доступа — Ethernet, Token Ring и т.д. Протоколов уровней приложений существует довольно много. Эти протоколы используются различными сервисами Интернет, которые и будут рассматриваться далее, и служат для выполнения своих конкретных задач — передачи eхe-файлов, отправки и получения электронных сообщений и т.д.

В качестве простого исследования возьмем протокол TCP/IP для обмена простыми текстовыми сообщениями в кодах ASCII по Интернет или интранет. Под термином интранет (intranet) имеются в виду локальные вычислительные сети, ресурсы которых подобны ресурсам Интернет, но доступны они только сотрудникам этих организаций и ограниченному кругу лиц, т.к. доступ к ресурсам в организациях обычно регламентируется. При этом на хост-компьютере, на который посылается сообщение должен быть запущен сервер, использующий этот протокол. У Вас будет два программных модуля - первый модуль, это сервер, а второй это клиент, работающие по одинаковому протоколу. Для успешной отправки сообщения Вам нужно задать имя удаленного компьютера или его IP-адрес, и указать порт, который слушает сервер. Также, нужно задать имя того, кто посылает сообщение. В некоторых случаях, если компьютер не подключен к сети, то вместо IP адреса (который Вы уже определяли ранее для этого случая в п.1) иногда пишут localhost. Проверьте эту возможность задания адреса.

2.4 Сервис Finger

Finger (протокол Finger, описанный в стандарте RFC 1288) — один из старейших инструментов Интернет, позволяет получить информацию о пользователе сети. Правда происходит это только в том случае, если почтовый ящик и домашний каталог этого пользователя находятся на компьютере, который умеет отвечать на Finger-запросы. В системе UNIX

это является стандартной возможностью, а в Windows этого нет. Для Finger существует шлюз из WWW. По адресу <http://www.mit.edu:8001/finger> вы найдете страницу с формой, в которой нужно заполнить единственное поле, указав в нем либо доменный IP адрес компьютера, либо электронный адрес человека, и нажать Enter. Например, послав через этот шлюз запрос по адресу amenti.Rutgers.EDU, вы получите список пользователей, работающих в данный момент на этой многопользовательской машине под управлением UNIX.

Если Вы перед именем машины укажете какое-нибудь имя пользователя и за ним символ @, то получите сведения только об этом пользователе. В последнее время данный протокол используют чаще всего в деструктивных целях и поэтому по запросу чаще всего можно и не получить ответа, если данный порт закрыт администратором.

2.5 Сервис Echo

Сервис Echo используется для отсылки текстовых сообщений на эхо-сервер и получения этого же сообщения обратно в соответствии со стандартом RFC 862. Он обычно применяется для тестирования и настройки сети, поскольку позволяет оценить время ответа сервера.

2.6 Протокол POP3

Протокол POP3 (Post Office Protocol Version 3 — протокол обработки почты) применяется для получения электронных писем (e-mail) от сервера. Он называется POP3 сервер или почтовый сервер. Этот протокол необходим для использования в некоторых корпоративных приложениях, использующих в том числе свои, внутренние почтовые серверы.

Перед тем, как получать электронную почту при помощи программного модуля необходимо иметь следующие компоненты успешной работы - а) быть зарегистрированным на почтовом сервере (иметь свой логин и пароль, либо получить эти данные от преподавателя для почтового адреса, используемого в учебных целях) б) иметь доступ к серверу через сетевое соединение или модем по соответствующему адресу в Интернет.

2.7 Протокол SMTP

Стандартный протокол SMTP (Simple Mail Transfer Protocol — простой протокол отправки почты) позволяет отправить электронную почту через почтовый сервер Интернет или выполнить некоторые другие команды, описанные в стандарте RFC 821. Порт по умолчанию, используемый этим протоколом, — 25. Регистрация на многих SMTP-серверах при подключении не требуется, соответственно вводить имя пользователя при работе с такими серверами не требуется. Для работы необходимо установить соединение с сервером.

Перед тем, как отсылать электронную почту при помощи программного модуля необходимо иметь следующие компоненты успешной работы - а) быть зарегистрированным на почтовом сервере (иметь свой логин и пароль, либо получить эти данные от преподавателя для почтового адреса, используемого в учебных целях) б) иметь доступ к серверу через сетевое соединение или модем по соответствующему адресу в Интернет.

2.8 Протокол UDP

Отправка пакетов датаграмм по сети Интернет или интранет осуществляется на основе использования протокола UDP (User Datagram Protocol). Стандарт протокола описан в RFC 768.

Перед тем, как отправлять пакеты датаграмм, нужно задать имя удаленного компьютера и порт, на которые будут посылаться данные. Эти значения нужно записать в соответствующие секции программных модулей. Для отправки и приема данных используются два подхода: передаются потоки или массивы символов (или буфер из оперативной памяти). Когда поступили данные по протоколу UDP и их можно использовать, происходит событие соответствующее событию, по которому сервер в зависимости от настроек выполняет какое-либо действие - формирует файл, пересылает данные в базу и т.д.

2.9 Кодирование и декодирование файлов в сервисах

Задача кодирования файлов возникает обычно из-за ограничений накладываемых символьной передачей данных, поскольку в программном обеспечении могут встречаться команды, которые отсутствуют в таблице символов, передаваемых по сети. В настоящее время кодирование и декодирование файлов осуществляется по алгоритму MIME или UUEncodes. Процесс осуществляется посредством передачи файлов в закодированном виде по Сети. После их получения выполняется обратная задача — декодирование.

Чтобы программный модуль мог выполнить свою задачу, необходимо указать имя обрабатываемого файла, расположенного на диске. Затем выбрать метод кодирования или декодирования, и указать имя файла, в который будет записываться результат преобразования и, соответственно, определить метод Encode или Decode. По умолчанию используется кодирование по алгоритму MIME.

2.10 Протокол HTTP

Протокол HTTP (HyperText Transfer Protocol) , язык разметки гипертекста HTML и унифицированный локатор ресурса URL составляют основу Web. HTML — это простой стандарт, который описывает формат документа, и достаточно быстрый для распределенных и гипермедийных информационных систем, в которых он и применяется. WWW-серверы и

программы просмотра (браузеры) сообщаются друг с другом при помощи протоколов верхнего уровня; в основном это протокол передачи гипертекста — (HTTP) и протокол передачи файлов — File Transfer Protocol (FTP) — которые позволяют клиенту и серверу обмениваться сообщениями и данными. Заметим, что разрабатываемые сегодня прикладные программы могут использовать любой протокол, который основывается на TCP/IP и используется для связи в Интернет. И наоборот, вы не должны ограничивать себя использованием протокола HTTP только в программах просмотра Всемирной паутины, а можете применять его для связи друг с другом любых прикладных программ. Протокол основывается на идее «ссылки», реализованной в универсальном идентификаторе ресурса (Uniform Resource Identifier, URI, RFC1630), универсальном локаторе ресурса (Uniform Resource Locator, RL, RFC1738) и универсальном имени ресурса (Uniform Resource Name, URN, RFC1737). Формат сообщений HTTP похож на формат, используемый в электронной почте и формат MIME. HTTP часто используется, как протокол для связи пользовательских программ-агентов и шлюзов в другие протоколы Интернет, такие как SMTP, NNTP, FTP и WAIS. Вторая составная часть протокола HTTP — это идея запроса-ответа. Клиент должен Установить соединение с сервером и отправить ему запрос, в котором указывается метод запроса, URI, версия протокола и сообщение в стиле MIME, в котором находятся модификаторы запроса, информация о клиенте и необязательное тело сообщения. В свой ответ сервер включает строку состояния, версию протокола, код ошибочного или нормального завершения запроса и затем сообщение, содержащее дополнительную информацию, в том числе тело сообщения. Первоначальный стандарт имел номер версии 0.9. Потом был стандарт версии 1.0 и сейчас действует стандарт версии 1.1. Последний из них описан в документе RFC 2068, а предыдущий в RFC 1945. Протокол HTTP применяется в WEB с 1990 г.

На сегодняшний день протокол HTTP, это один из наиболее распространенных протоколов обмена данными в сетях клиент/сервер. Одна из замечательных особенностей этого протокола заключается еще и в том, что он позволяет клиенту и серверу обмениваться данными без необходимости поддерживать постоянное сетевое соединение. Это протокол, ориентированный в основном на транзакции: клиент запрашивает порцию данных и сервер, удовлетворив его запрос, завершает соединение.

Если вы используете Web-браузер для просмотра, например, URL <http://www.tusur.ru>, то Web-страница будет перемещаться в браузер, после чего клиент сможет ее прочитать. После того, как загрузка страницы закончится, сетевое соединение будет разорвано. По этой причине можно рассматривать сервер как средство доставки пакетов данных. Для приложения, которое большую часть времени использует сеть неактивно

(например, когда клиент читает Web-страницу), использование пакетного метода передачи данных является удачным выходом из трудного положения, потому что сервер сможет тогда обрабатывать запросы других клиентов на получение данных, вместо того, чтобы тратить свои ресурсы на поддержание холостого соединения. Таким образом, программные средства WWW являются универсальными для различных сервисов Интернет, а сама информационная система WWW играет интегрирующую роль. Все эти богатые возможности WWW в представлении различного рода информации обеспечиваются благодаря трем «кистам»: языку HTML (HyperText Markup Language — язык разметки гипертекста), на котором составляются документы WWW, протоколу передачи гипертекстовых документов HTTP (HyperText Transfer Protocol — протокол передачи гипертекста) и универсальному локатору ресурса URL (Universal Resource Locator). Внешний вид гипертекстового документа на экране монитора пользователя определяется оборудованием. Если вы работаете за графическим или текстовым терминалом, то в каждом случае документ будет выглядеть по-своему, но структура его останется неизменной, поскольку она задана форматом HTML. WWW - сервис прямого доступа, требующий полноценного подключения к Интернет, и желательно по быстрой линии связи в случае, если документы, которые вы просматриваете, содержат много графики и другой мультимедийной информации. Однако с помощью специальных программ вы можете закачать весь сайт на свою машину с сохранением связей между документами, а потом просмотреть его в офф-лайновом режиме. Возможность такая есть, но при этом теряется прелесть непосредственного общения. Например, в языке HTML существует понятие формы. Пользователь вводит в них некоторую информацию, которая затем может передаваться на сервер. Таким способом заполняются анкеты, регистрационные карты, проводятся социологические опросы. С помощью формы также вы можете определять выводимую для просмотра информацию и сделать для себя процесс работы максимально удобным и быстрым.

Язык разметки гипертекста HTML. Идея гипертекстовой информационной системы состоит в том, что пользователь имеет возможность просматривать страницы документа в том порядке, в котором ему это нужно, а не последовательно, как при чтении обычного, бумажного документа. Осуществляется эта возможность через систему гипертекстовых ссылок, связывающих между собой различные страницы документа. Со временем идея гипертекста разрослась, и современный стандарт можно назвать уже не гипертекстом, а гипермедиа, поскольку теперь документы HTML включают в себя графическую, аудио и видеoinформацию.

Что касается самой идеи гипертекста, то она появилась на свет, возможно, в 1945 г., когда В. Буш, советник по науке президента

Рузвельта, предложил проект электромеханической информационной системы Metex. Идея не была реализована. Затем в 1965 г. Т. Нельсон ввел в обращение сам термин «гипертекст», развил и даже реализовал некоторые «гипертекстовые» идеи. Его проект назывался Xanadu, развивался он достаточно долго, с 1988 г. проект финансировался Autodesk, которая, однако заморозила его в 1992 г. и вскоре закрыла. Проект погубила преждевременность его рождения и закрытость. В 1975 году идеи гипертекста были воплощены в информационной системе атомного авианосца «Карл Винстон». Работы над этой идеей продолжались потом различными фирмами и людьми. В 1987 г. была проведена первая специализированная конференция Hypertext-87, пока все это не воплотилось в Web.

Созданная в 1989 г. в CERN (Conseil European pour la Recherche Nucleaire — Европейский центр исследований по физике высоких энергий) Web, опираясь на протокол HTTP, обеспечивает простой доступ к документам Web, архивам файлов стандарта FTP, группам новостей UseNet и др. Но, возможно, основной идеей WEB является идея гипертекста, которую развивали уже давно разные люди. Но только в конце 80-х годов Тим Бернерс Ли предложил удачную рабочую модель распределенной гипертекстовой системы. Помимо того, что он предложил стандарт языка HTML, он разработал первоначальный стандарт протокола HTTP, по которому можно было обмениваться гипертекстовыми документами, и разработал идею URL, как общую систему адресации. Теперь, когда прошли годы, становится очевидным, что была предложена удачная система, которую уже давно ждали, и бурное развитие которой мы наблюдаем с тех пор и по сей день. Информация на Web-сервере хранится в виде так называемых Web-страниц или Web-документов. Web-страница — это файл в формате HTML. Данный формат определяет внешний вид документа, взаимное расположение текстовой, графической и мультимедийной информации. Поскольку Web включает в себя мультимедийную информацию, она стала идеальной средой для хранения и распространения информации в Интернет.

Стандарт HTML интуитивно понятен: «управляется» документ HTML при помощи тегов. О значении определенных тегов вполне можно догадаться и самим. В качестве примера, рассмотрим документ следующего содержания: <HTML>

```
В документах Web вы можете выделить текст<BK>
<B> жирным шрифтом </B> <BR>
<I>или курсивом </I>
</HTML>
```

Этот документ содержит три строки текста: первая из этих строк «В документах Web вы можете выделить текст» будет выводиться стандартным Шрифтом, вторая строка «жирным шрифтом» — жирным шрифтом, а третья курсивом. Тег включает режим жирного текста, а

тег `` отменяет его; тег `
` используется для перевода строки. Составить и отредактировать этот и любой другой Web-документ можно простым текстовым редактором, а результат всегда можно проверить в браузере в режиме off-line. В случае же, если вы используете в документах графику, таблицы и ссылки на другие документы и ресурсы Web, то такие Документы удобнее создавать с помощью специализированных HTML редакторов. Последний стандарт HTML 4.0 основан на стандарте SGML (Standard Generalized Markup Language) ISO 8879. Описание последнего стандарта HTML вы можете найти на сервере www.w3.org.

Итак, стандарт языка HTML 4.0 позволяет авторам документов делать следующее: Публиковать документы, содержащие текст, заголовки, таблицы, списки, графику и т.д. Переходить на другие документы с помощью механизма гиперссылок простым щелчком мыши, разрабатывать формы для ввода данных от пользователя и передавать эти данные в удаленные службы, например, для поиска информации, покупки различных товаров, резервирования мест в гостинице и т.д., включать в документы электронные таблицы, видеоклипы, звуковые файлы и другую мультимедийную информацию и приложения. Один и тот же документ может быть просмотрен на мониторе современного PC, на алфавитно-цифровом мониторе, подключенном к мейнфрейму, на дисплее сотового телефона и т.д. При этом, конечно, на алфавитно-цифровом мониторе мы не увидим графики, но все что можно воспроизвести, а это будет текст, — монитор воспроизведет (по возможности, с соблюдением размеров и начертания символов). При этом не произойдет никаких ошибок. Такой подход позволяет создавать одну версию документа, а не поддерживать одновременно несколько, для различных аппаратных платформ пользователя.

Минимальный и корректный документ HTML должен выглядеть следующим образом:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0/EN"
http://www.w3.org/TR/REC-html40/strict.dtd> <HTML>
<HEAD>
<TITLE>Мой первый HTML документ! </TITLE>
</HEAD>
<BODY>
Всем привет!
</BODY> </HTML>
```

В самом начале документа находится секция, в которой вы сообщаете тип (DTD — Document Type Definition) вашего документа: HTML 4.0. За ней идут две следующие обязательные секции: декларативный заголовок документа и тело документа. Здесь используются три обязательных тега `<HTML>`, `<HEAD>` и `<BODY>`. Первый тег указывает браузеру на начало HTML-документа.

Заканчиваться документ должен тегом `</HTML>` В HTML, как вы видите, теги используются парами, один включает некий режим, а другой, имеющий то же имя, перед которым поставлен символ «слэш», отменяет его. Между тегами `<HTML>` и `</HTML>` должны помещаться секции заголовка (теги `<HEAD>` и `</HEAD>`) и тела документа (между тегами `<BODY>` и `</BODY>`). Если в документе применяются фреймы, а документ разбивается на несколько независимых колонок, или горизонтальных секций, в каждую из которых выводится свой HTML-документ, то в этом случае секция `<BODY> ... </BODY>` заменяется на конструкцию `<FRAMESET> ... </FRAMESET>`.

Все элементы языка HTML обычно «обкладываются» парами тегов: `<имя тега>` и `</имя_тега>`, между которыми помещается содержимое по-английски — content; это слово стало жаргонным, и применительно к веб о «контенте» вы будете слышать постоянно. Существуют и непарные P теги, например, `<P>` и `
`. Первый тег объявляет новый параграф — и текст, который идет за ним, будет выведен с новой строки и с отступом. И так до тех пор, пока не встретится новый тег `<P>` или `
` или другой форматированный тег. Вообще говоря у тега `<P>` есть завершающий тег `</P>`. Но его использовать не обязательно, следующий тег `<P>` будет завершать действие предыдущего тега `<P>`. В заголовочной части документа указывается различная служебная информация. Здесь всегда должна быть строка заголовка, который вы видите в заголовочной строке окна браузера. Заголовок помечается тегами `<TITLE>` и `</TITLE>`. Стоит упомянуть «богатый» своими возможностями элемент `<META>`, который может размещаться там же. Здесь автор документа может указать описание документа, указать свое имя, ключевые слова, по которым поисковые сервера могут включать этот документ в свои выборки и другую информацию. В тексте документа можно управлять цветом фона и шрифтов, применять различные способы выравнивания текста, менять шрифты, размеры символов. В тексте документа можно размещать графическую информацию (тег ``) и т.д. Так называемые гиперссылки вводятся тегом `<A...>`.

Например:
`` Ссылка на сервер Rambler ``
При просмотре в браузере, от этой строки останется последняя часть «Ссылка на сервер Rambler», а «техническая» информация будет от пользователя скрыта. В текст документа можно помещать комментарии. Выглядят они следующим образом:
`<!-- Комментарий -->` может занимать несколько строк. Браузер игнорирует комментарии, если только там не задаются скрипты. Комментарии часто используются для этой цели. Браузеры, которые не работают со скриптами, будут их игнорировать, как и любой текст между символами `<!-- ... -->`. Те кто умеют — те поймут. Скриптами называются Функции на Java, Visual Basic и др. языках, вызывать которые вы можете

прямо из HTML-документа. Таковы в общем случае особенности применения прокола HTTP.

3 ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ

3.1 Получение адреса в сети

- Загрузить программный модуль для получения IP адреса - [lab0_IPadres.zip](#)
- После загрузки его необходимо разархивировать в произвольную папку и запустить. Его внешний вид представлен на рисунке.

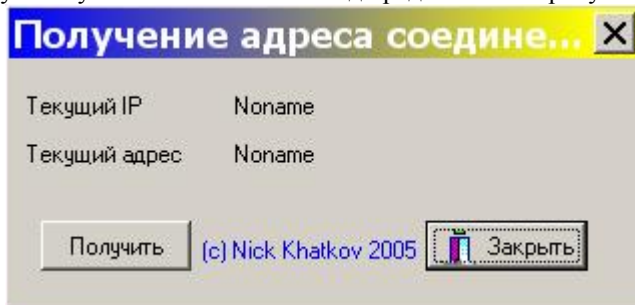


Рис.1 Внешний вид программного модуля для получения сетевого адреса.

- Получить IP и текущий адреса - определите между ними разницу. Получить адреса, как для локальной машине - не подключенной к сети, так и к сетевой машине - подключенной в сеть. Определить по полученным адресам к какому классу относиться сеть в которой Вы работаете.

3.2 Протокол времени суток - Daytime Protocol

- Используя сеть Интернет и поисковые системы самостоятельно определить адреса и порты серверов, обеспечивающих протоколы времени по ключевым словам - Daytime Protocol.
- Загрузить программный модуль для получения даты и времени - [lab1_datetime.zip](#)
- После загрузки его необходимо разархивировать в произвольную папку и запустить. Его внешний вид представлен на рисунке.

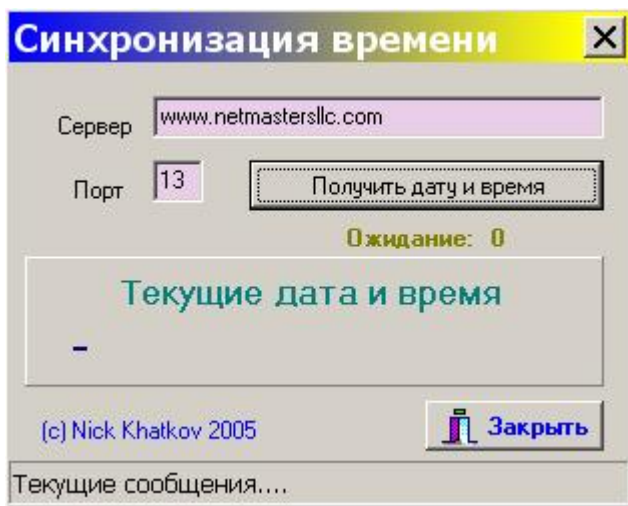


Рис.2 Внешний вид программного модуля для получения времени и даты.

- Получить текущие дату и время с помощью указанных ранее серверов, если не получилось, то поварьируйте номера портов этих серверов, либо проверьте наличие доступа к этим серверам в Интернете. Получить текущие дату и время с серверов найденных через поисковые системы, аналогично, как и в предыдущем случае. Определить ориентировочно время получения отклика на запрос к серверу с помощью программы (надпись - время ожидания). Описать трудности, которыми сопровождалась данные операции

3.3 Семейство протоколов TCP/IP

- Используя сеть Интернет и поисковые системы самостоятельно определить дополнительные номера портов, обеспечивающих передачу строковых данных по протоколу TCP.
- Загрузить пакет из двух программных модулей для организации передачи строковых сообщений между ними - sentstring.zip
- После загрузки его необходимо разархивировать в произвольную папку и запустить два модуля последовательно на одном и том же компьютере (последовательность запуска не играет никакой роли). Их внешний вид представлен на рисунках.

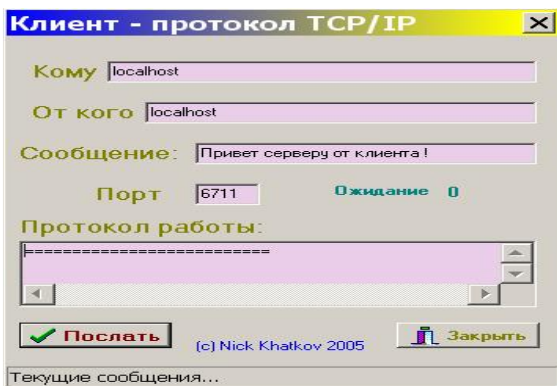


Рис.3 Внешний вид программного модуля клиента для передачи сообщений.

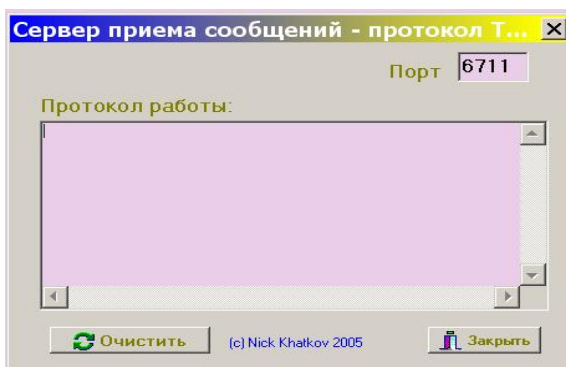


Рис.4 Внешний вид программного модуля сервера для приема сообщений.

- В секциях кому и от кого прописать текущий IP адрес компьютера. Передать сообщение серверу и проверьте получено ли им сообщение. Выделить длинную строку или некоторый объем текста в произвольном текстовом редакторе и методом копирования через буфер обмена вставить его в передаваемую секцию сообщения или попробовать загрузить в эту секцию послание в виде очень длинной строки. Проверить, что получит сервер.
- При наличии локальной сети попробовать на соседнем компьютере запустить свой сервер сообщений, предварительно узнав его IP адрес и переслать ему по сети свое сообщение. Либо аналогичную работу проделать и с удаленным компьютером, указанным преподавателем - осуществить односторонний обмен сообщениями.
- Для подобранных дополнительных номеров портов в интернете или по справке от администратора сети осуществить прием-передачу строчковых сообщений.

- Для всех режимов работы клиента и сервера сохранять проколы работы, формируемые в больших секциях через буфер обмена в отчете.

3.4 Сервис Finger

- Используя сеть Интернет и поисковые системы самостоятельно определить дополнительные адреса серверов и номера портов, обеспечивающих работу сервиса Finger.
- Загрузить пакет из двух программных модулей для организации передачи строковых сообщений между ними - finger.zip
- После загрузки его необходимо разархивировать в произвольную папку и запустить программный модуль.

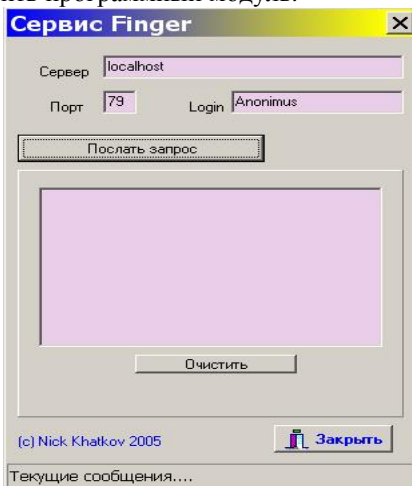


Рис.5 Внешний вид программного модуля для запроса о клиенте.

- Указать адрес сервера для запроса. В секции запроса сформировать запрос согласно формату сообщения и сделать запрос. Записать протокол работы сервиса через буфер обмена в отчет.
- При наличии локальной сети попробовать сделать запрос на свой счет, предварительно узнав IP адрес сервера с помощью программного модуля п.1.
- При получении отрицательного ответа по запросу уточнить у администратора сети номер порта и возможность работы по данному протоколу в данной сети. Записать протокол работы программного модуля в отчет.

3.5 Сервис Echo

- Используя сеть Интернет и поисковые системы самостоятельно определить дополнительные адреса серверов и номера портов, обеспечивающих работу сервиса Echo.

- Загрузить программный модуль для организации времени получения отклика от тестируемого сервера - echo.zip
- После загрузки его необходимо разархивировать в произвольную папку и запустить программный модуль.

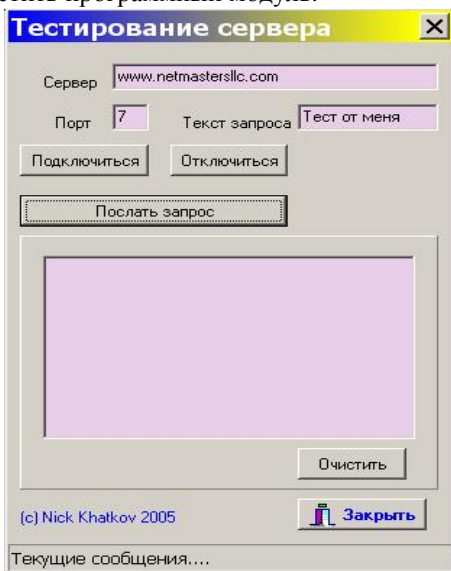


Рис.6 Внешний вид программного модуля Echo.

- Указать адрес сервера для запроса. В секции запроса сформировать запрос согласно формату сообщения и сделать запрос. Записать протокол работы сервиса через буфер обмена в отчет.
- При наличии локальной сети попробовать сделать запрос на Ваш сервер, предварительно узнав его IP адрес с помощью программного модуля п.3.1.
- При получении отрицательного ответа по запросу уточнить у администратора сети номер порта и возможность работы по данному протоколу в данной сети. Записать протокол работы программного модуля.

3.6 Протокол POP3.

- Загрузить программный модуль для организации получения почтовых сообщений - mail.zip
- После загрузки его необходимо разархивировать в произвольную папку и запустить программный модуль.

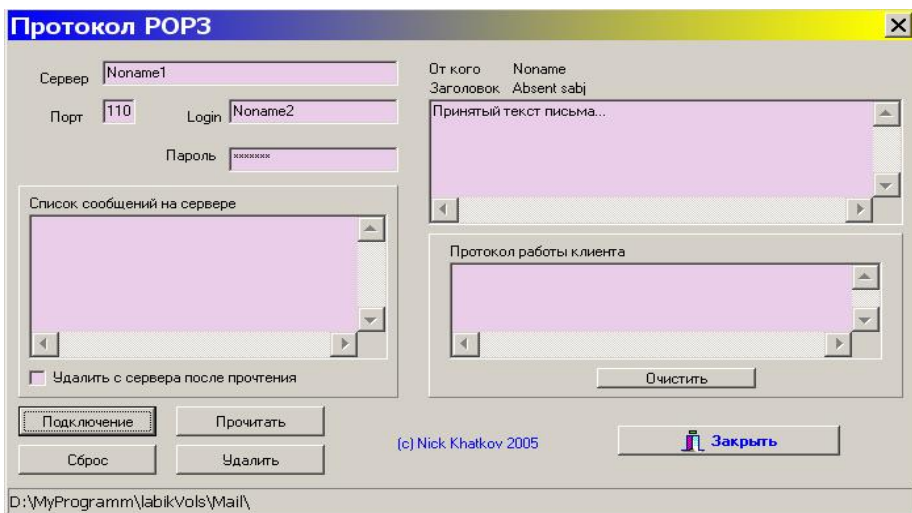


Рис.7 Внешний вид программного модуля для протокола POP3.

- Указать адрес почтового сервера для запроса. Ввести логин и пароль в секциях. Осуществить подключение к почтовому серверу. Установить курсор на нужное сообщение и прочитать его. Попробовать удалять тестовые письма с определением момента удаления - в сессию подключения или нет. Разобрать структуру полученного почтового сообщения и представить его в отчете. Записать протокол работы почтового сервиса через буфер обмена в отчет на всех стадиях работы с почтовыми сообщениями, начиная с момента подключения и кончая моментом отключения от почтового сервера.

3.7 Протокол SMTP

- Загрузить программный модуль для организации получения почтовых сообщений - mailout.zip
- После загрузки его необходимо разархивировать в произвольную папку и запустить программный модуль.

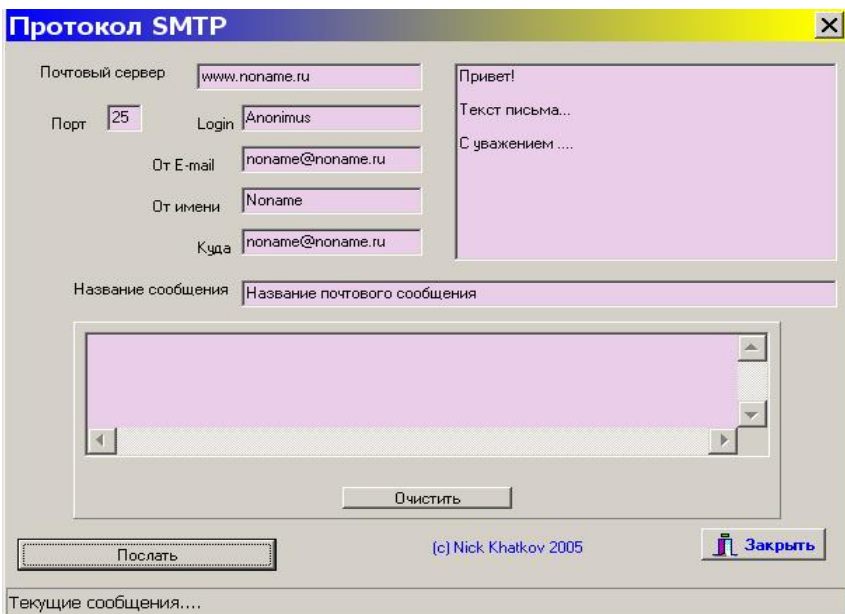


Рис.8 Внешний вид программного модуля для протокола SMTP.

- Указать адрес почтового сервера для запроса. Ввести логин, если это необходимо и заполнить остальные секции. Осуществить подключение к почтовому серверу и отправить произвольное сообщение. Используя программный модуль из п.3.6 получить почту и сделать анализ полученного письма - основные отличия, служебные сообщения. Представить все это в отчете. Записать протокол работы почтового сервиса через буфер обмена в отчет на всех стадиях работы с почтовыми сообщениями, начиная с момента подключения и кончая моментом отключения от почтового сервера.

3.8 Протокол UDP

- Загрузить программные модули для организации получения и передачи датаграмм - datagramm.zip
- После загрузки его необходимо разархивировать в произвольную папку и запустить на локальной машине сначала программный модуль сервера, а затем клиента.

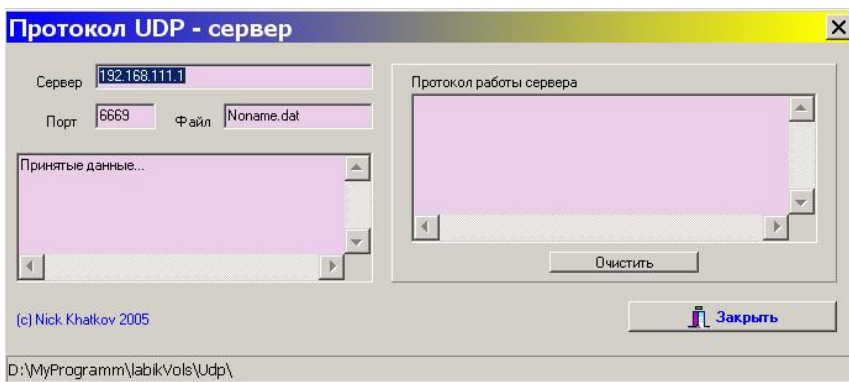


Рис.9 Внешний вид программного модуля сервера протокола UDP.

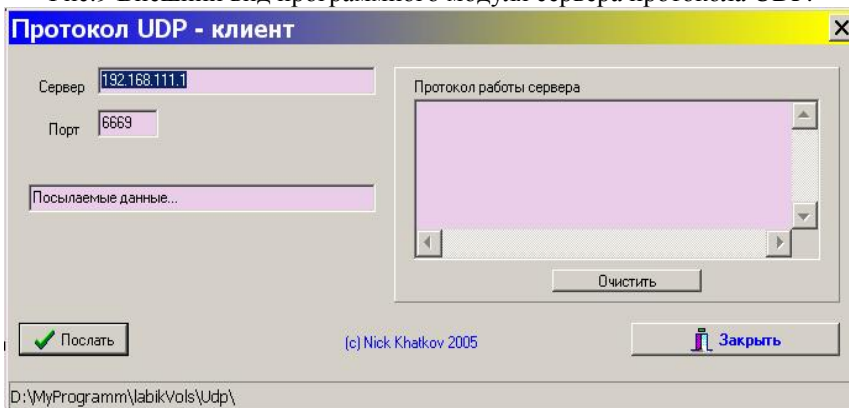


Рис.10 Внешний вид программного модуля клиента протокола UDP.

- Если IP адрес не определился автоматически правильно, то определить его самостоятельно с помощью программного модуля п.3.1 лабораторной работы. аналогично поступить и с клиентом. Для начала использовать текущее значение порта, а потом попробовать поварьировать его значение. Сделать вывод о свойствах порта и описать это в отчете. Заполнить произвольными данными строку передачи, которая имитирует содержание датаграммы. Осуществить передачу датаграммы сначала на локальном компьютере (сервер и клиент находятся на одном и том же компьютере) и на удаленном (только сервер находится на другом компьютере). Оценить приблизительно(лучше или хуже) скорость передачи по сравнению с аналогичным процессом. Записать протоколы работы клиента и сервера в соответствующих режимах передачи данных на всех стадиях их работы с датаграммами, начиная с момента подключения и кончая моментом отключения.

3.9 Кодирование и декодирование файлов в сервисах

- Загрузить программный модуль для организации кодирования и декодирования файлов - code_decode.zip
- После загрузки его необходимо разархивировать в произвольную папку и запустить на локальной машине.

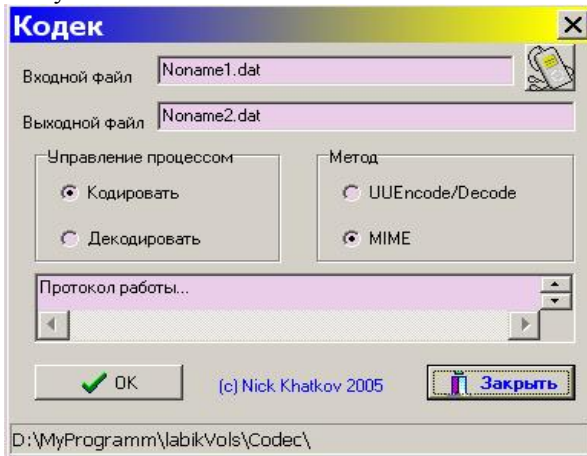


Рис.11 Внешний вид программного модуля кодирования декодирования файлов.

- Создайте текстовый файл небольшого объема с какими-либо данными и загрузить его в программный модуль. Осуществить кодирование файла разными методами. Для каждого из методов привести пример в виде небольшой выдержки в отчете. Осуществить последующее декодирование и проверить правильность кодирования. Оценить помехоустойчивость кодирования - декодирования посредством частичного удаления строк, их перестановки, частичной замены и удаления символов и др. Привести получившиеся примеры в отчете. Осуществить прием электронной почты с помощью программного модуля представленного в п.3.6 лабораторной работы. Получаемое письмо должно иметь помимо содержания еще и передаваемый файл. Осуществить декодирование переданного файла, определив метод кодирования. Записать протоколы работы в соответствующих режимах программного модуля в отчет.

3.10 Протокол HTTP.

- Загрузить программный модуль для организации кодирования и декодирования файлов - brayzer.zip
- После загрузки его необходимо разархивировать в произвольную папку и запустить на локальной машине.

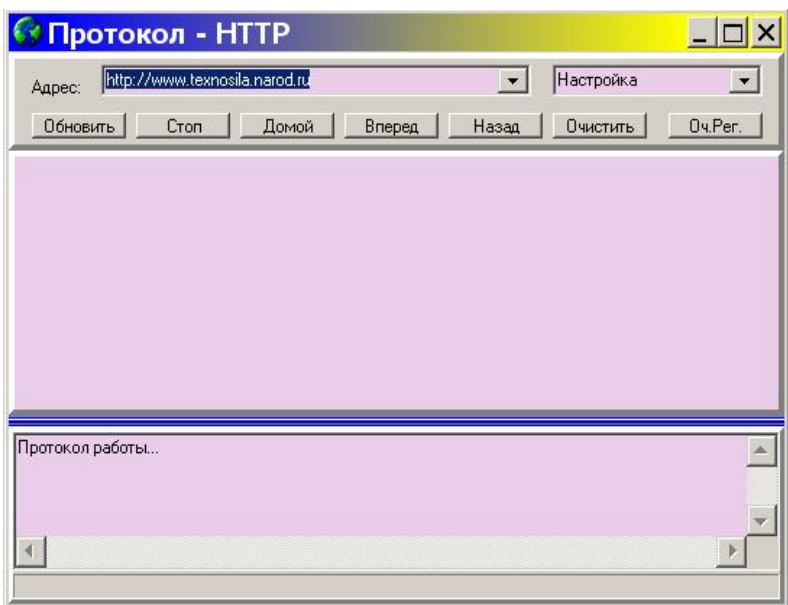


Рис.12 Внешний вид программного модуля для работы с протоколом HTTP.

- Получить открытый доступ в Интернет. Набрать адрес www.tusur.ru и загрузить гипертекстовую страницу с этого сервера. Осуществить просмотр прокола работы и записать его в отчет с Вашими комментариями. Набрать адрес какого-либо поискового сервера и получить страницу с этого сервера, сравнить затем проколы работы серверов и приведите результаты сравнения в отчете. Также рассмотреть разницу в работе программного модуля с помощью секции настройка. Для этого на компьютере необходимо определить временную папку для интернет файлов в операционной системе. Найти сервер с паролным доступом и аналогично провести анализ его работы при неудачной попытке доступа. Если Вам попался сомнительный сервер создающий неконтролируемое количество дополнительных окон и самостоятельно портящий реестр, так что при загрузке произвольного адреса, подключается один и тот же сервер, то привести пример прокола работы такого сервера. Очистить негативное влияние подобных серверов можно нажатием на кнопку Оч. Рег. Записать протоколы работы в соответствующих режимах программного модуля в отчет.

3.11 Записать выводы по каждому разделу в отчет.

4 КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое протокол передачи данных в компьютерной сети?
2. Сколько классов сетей существует и как они называются?
3. Сколько уровней имеют IP адреса в иерархической системе нумерации?
4. Сколько вариантов адресов существует в сети?
5. Для чего была создана специальная служба DNS?
6. Для чего необходим Daytime Protocol?
7. Все ли сервера поддерживают Daytime Protocol?
8. Сколько уровней имеет модель OSI?
9. Является ли протокол IP протоколом уровня?
10. Какие сети подразумеваются под термином интранет?
11. Для чего необходим протокол Finger?
12. Для чего необходим сервис Echo?
13. Для чего необходим протокол POP3?
14. Для чего необходим протокол SMTP?
15. Для чего необходим протокол UDP?
16. Существенно ли отличаются методы кодирования и декодирования по алгоритму MIME или UUEncodes друг от друга?
17. В каких протоколах используются методы кодирования и декодирования по алгоритму MIME или UUEncodes?
18. Для чего необходим протокол HTTP?
19. Какой порт необходим для работы с протоколом HTTP?

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

Основная

1. **Ефанов В.И.** Электрические и волоконно-оптические линии связи. 2-е изд., доп. – Томск : Томск. гос. ун-т систем упр. и радиоэлектроники, 2007. – 256 с. [25]
2. **Ефанов В.И.** Основы проектирования сетей кабельного телевидения: Учеб. Пособие. Томск, Том. гос. ун-т систем управления и радиоэлектроники, 2007. – 104 с. [25]

Дополнительная

3. **Козлов А.В.** Программирование для интернет в Delphi 5., изд. ЗАО "Издательство БИНОМ", 2001г.,368 с.
4. **Шапошников И.В.** Интернет программирование., СПб.: БХВ - Санкт-Петербург, 2000. - 224 с.
5. **Олифер В., Олифер Н.** Новые технологии и оборудование IP - сетей., СПб.: БХВ - Санкт-Петербург, 2000. - 324 с.
6. Создание Intranet. Официальное руководство Microsoft. СПб.: БХВ - Санкт-Петербург, 2000. - 214 с.