Частное учреждение образования

«Колледж бизнеса и права»

|  |
| --- |
| УТВЕРЖДАЮ  Ведущий методист колледжа  \_\_\_\_\_\_\_\_\_\_Е.В. Паскал  «\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_2021 |
|  |

|  |  |
| --- | --- |
| Специальность:  2-40 01 01 Программное обеспечение информационных технологий | Учебная дисциплина: “Компьютерные сети” |

ЛАБОРАТОРНАЯ РАБОТА №19

Инструкционно-технологическая карта

ТЕМА: Взаимодействие прикладных программ с помощью протокола FTP и протоколов прикладного уровня.

ЦЕЛЬ РАБОТЫ: Отработать навык разработки прикладных программ, осуществляющих взаимодействие и передачу данных друг другу на основе прикладных протоколов.

ВРЕМЯ РАБОТЫ: 2 часа

СОДЕРЖАНИЕ РАБОТЫ:

1. Теоретические сведения
2. Практическое задание
3. Справочные данные
4. Контрольные вопросы
5. Литература
6. **ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ**

FTP (File Transfer Protocol – протокол передачи данных) – один из старейших протоколов в Internet и входит в его стандарты. Первые спецификации FTP относятся к 1971 году. С тех пор FTP претерпел множество модификаций и значительно расширил свои возможности. FTP может использоваться как в программах пользователей, так и в виде специальной утилиты операционной системы.

FTP предназначен для решения задач разделения доступа к файлам на удаленных узлах, прямого или косвенного использования ресурсов удаленных компьютеров, обеспечения независимости клиента от файловых систем удаленных узлов, эффективной и надежной передачи данных.

Обмен данными в FTP происходит по TCP-каналу. Обмен построен на технологии «клиент-сервер». FTP не может использоваться для передачи конфиденциальных данных, поскольку не обеспечивает защиты передаваемой информации и передает между сервером и клиентом открытый текст. FTP-сервер может потребовать от FTP-клиента аутентификации. Однако пароль и идентификатор пользователя будут переданы от клиента на сервер открытым текстом.

Простейшая модель работы протокола FTP представлена на рис. П.2.

|  |
| --- |
|  |
| *Рис.1. Простейшая модель работы протокола FTP* |

FTP соединение инициируется интерпретатором протокола пользователя.

Управление обменом осуществляется по каналу управления в стандарте протокола TELNET. Команды FTP генерируются интерпретатором протокола пользователя и передаются на сервер. Ответы сервера отправляются пользователю также по каналу управления.

Команды FTP определяют параметры канала передачи данных и самого процесса передачи. Они также определяют и характер работы с удаленной и локальной файловыми системами.

Сессия управления инициализирует канал передачи данных. При организации канала передачи данных последовательность действий другая, отличная от организации канала управления. В этом случае сервер инициирует обмен данными в соответствии с согласованными в сессии управления параметрами.

Канал данных устанавливается для того же узла, что и канал управления, через который ведется настройка канала данных. Канал данных может быть использован как для приема, так и для передачи данных.

**Алгоритм работы** протокола FTP состоит в следующем:

1. Сервер FTP использует в качестве управляющего соединение на TCP порт 21, который всегда находится в состоянии ожидания соединения со стороны пользователя FTP.

2. После того как устанавливается управляющее соединение модуля «Интерпретатор протокола пользователя» с модулем сервера – «Интерпретатор протокола сервера», пользователь (клиент) может отправлять на сервер команды. FTP-команды определяют параметры соединения передачи данных: роль участников соединения (активный или пассивный), порт соединения (как для модуля «Программа передачи данных пользователя», так и для модуля «Программа передачи данных сервера»), тип передачи, тип передаваемых данных, структуру данных и управляющие директивы, обозначающие действия, которые пользователь хочет совершить (например, сохранить, считать, добавить или удалить данные или файл и другие).

3. После того как согласованы все параметры канала передачи данных, один из участников соединения (пассивный модуль) становится в режим ожидания открытия соединения на определенный порт. После этого другая сторона (активный модуль) открывает соединение на указанный порт и начинается обмен данными.

4. После окончания передачи данных соединение между «Программой передачи данных сервера» и «Программой передачи данных пользователя» закрывается, но управляющее соединение «Интерпретатора протокола сервера» и «Интерпретатора протокола пользователя» остается открытым. Пользователь, не закрывая сессии FTP, может еще раз открыть канал передачи данных.

Возможна ситуация, когда данные должны передаваться на третий узел. В этом случае имеет место другая модель работы протокола FTP, при которой пользователь организует канал управления с двумя серверами и прямой канал данных между ними. Команды управления идут через пользователя, а данные – напрямую между серверами. Подробное рассмотрение этой модели работы протокола FTP выходит за рамки данной лабораторной работы.

**Особенности управления процессом обмена данными**

Основу передачи данных FTP составляет механизм установления соединения между соответствующими портами и выбора параметров передачи. Каждый участник FTP-соединения должен поддерживать порт передачи данных по умолчанию. По умолчанию «Программа передачи данных пользователя» использует тот же порт, что и для передачи команд, а «Программа передачи данных сервера» использует порт L-1, где L – управляющий порт. Однако для ускорения процессов обмена данными участники соединения часто используют другие порты передачи данных.

Передача данных может вестись в активном или пассивном режиме. Если доступ к ftp-серверу осуществляется через proxy-сервер, то возможна работа только в пассивном режиме.

***Установление******соединения******передачи******данных******при******активном******режиме******работы*** происходит следующим образом:

1. Клиент создает сокет на выбранном им порте Р и активизирует его (переводит в режим ожидания соединения).

2. Клиент направляет серверу по управляющему соединению команду PORT (описание приведено ниже), в которой указывает свой IP-адрес и выбранный для передачи данных порт Р.

3. В случае успешного получения и обработки команды PORT сервер по управляющему соединению отправляет клиенту положительный отклик. Затем сервер пытается соединиться с портом Р клиента со своего локального порта данных А (может быть выбран сервером произвольно).

4. Клиент обнаруживает событие соединения на порт Р и начинает процесс передачи данных с использованием управляющего соединения для отправки команд FTP-сервиса и соединения передачи данных для получения либо отправки данных.

***Установление******соединения******передачи******данных******при******пассивном******режиме работы*** происходит следующим образом:

1. Клиент отправляет по управляющему соединению команду PASV, указывающую серверу о намерении клиента работать в пассивном режиме.

2. В случае успешного получения и обработки команды PASV сервер создает сокет на выбранном им порте Р, активизирует его (переводит его в режим ожидания соединения) и отправляет клиенту по управляющему соединению отклик, в котором указываются параметры соединения (IP-адрес сервера и порт Р).

3. Клиент, получив положительный отклик с параметрами соединения, создает на произвольном порте А сокет и активизирует его, то есть пытается соединиться с портом Р сервера.

4. В случае успешного установления соединения клиент начинает процесс передачи данных с использованием управляющего соединения для отправки команд FTP-сервиса и соединения передачи данных для получения либо отправки данных.

Одновременно с передачей данных по установленному соединению в обоих режимах по каналу «Интерпретатор протокола сервера»– «Интерпретатор протокола пользователя» могут передаваться уведомления о получении данных. Протокол FTP требует, чтобы управляющее соединение было открыто, пока по каналу обмена данными идет передача.

Как правило, сервер FTP ответственен за открытие и закрытие канала передачи данных. Сервер FTP должен самостоятельно закрыть канал передачи данных в следующих случаях:

1. Сервер закончил передачу данных в формате, который требует закрытия соединения.

2. Сервер получил от пользователя команду «прервать соединение».

3. Пользователь изменил параметры порта передачи данных.

4. Было закрыто управляющее соединение.

5. Возникли ошибки, при которых невозможно возобновить передачу данных.

**Команды и ответы протокола ftp**

Все команды протокола FTP отправляются «Интерпретатором протокола пользователя» в текстовом виде – по одной команде в строке. Каждая строка команды – идентификатор и аргументы – заканчиваются символами CRLF. Имя команды отделяется от аргумента символом пробела.

Обработчик команд возвращает код обработки каждой команды, состоящий из трех цифр. Коды обработки составляют определенную иерархическую структуру и, как правило, определенная команда может возвратить только определенный набор кодов. За кодом обработки команды следует символ пробела и текст пояснения. Описание команд и основных кодов ответов приведено в п. 5.5 «Справочные данные».

Команды протокола FTP, которыми обмениваются «Интерпретатор протокола сервера» и «Интерпретатор протокола пользователя», можно разделить на три группы.

1. *Команды управления доступом к системе* обеспечивают авторизацию пользователя в системе, выход из нее и настройку некоторых текущих параметров соединения.

2. *Команды управления потоком данных* устанавливают параметры передачи данных. Все параметры, описываемые этими командами, имеют значение по умолчанию, поэтому команды управления потоком используются только тогда, когда необходимо изменить значение параметров передачи, используемых по умолчанию. Команды управления потоком могут подаваться в любом порядке, но все они должны предшествовать командам FTP-сервиса.

3. *Команды FTP-сервиса* определяют действия, которые необходимо произвести с указанными файлами. Как правило, аргументом команд этой группы является имя файла.

1. **ПРАКТИЧЕСКОЕ ЗАДАНИЕ**

1. Добавим на рабочее поле программы 5 коммутаторов Switch 2960-24TT. По умолчанию они имеют имена Switch0 – Switch4.

2. Добавим на рабочее поле восемь компьютеров с именами по умолчанию PC0 – PC7.

3. Соединим устройства в сеть Ethernet , как показано на рис. 1.1. Компьютер с коммутатором соединяются витой парой, а коммутаторы между собой – кросс-кабелем. На всех устройствах для подключения используются порты FastEthernet.



Рис. 1.1. Модель сети Ethernet

4. Сохраним созданную топологию, используя команду *Save* в меню *File*.

5. Откроем окно свойств устройства PC0, щелкнув левой кнопкой мыши на его изображении. Перейдем на вкладку *Desktop* и откроем командную строку, нажав на *Command Prompt*.

6. Для вывода списка доступных команд необходимо в командной строке ввести *?* и нажать *Enter*.

7. Конфигурирование компьютера осуществляется с помощью команды *ipconfig*. Например, чтобы устройству PC0 установить сетевой адрес 192.168.1.2 и задать маску 255.255.255.0 в командной строке необходимо ввести:

*ipconfig 192.168.1.2 255.255.255.0*

8. Для проверки назначенных значений сетевого адреса и маски в командной строке нужно еще раз ввести *ipconfig*. Появится сообщение о заданных сетевых параметрах устройства:

*FastEthernet0 Connection:(default port)*

*Link-local IPv6 Address.........: FE80:230:A3FF:FEAA:BD12*

*IP Address.............................: 192.168.1.2*

*Subnet Mask..........................: 255.255.255.0*

*Default Gateway....................: 0.0.0.0*

9. IP адрес и маску сети можно также задать, используя графический интерфейс устройства. Для этого в окне свойств на вкладке *Desktop* необходимо выбрать *IP Configuration*. Откроется окно, показанное на рис. 1.2. Сетевой адрес вводится в поле *IP Adress*, а маска – в поле



Рис. 1.2. Окно настроек IP Configuration

*Subnet Mask*. Переключатель способа назначения IP адреса должен находиться в положении *Static*.

10.Аналогично для всех остальных компьютеров назначим адреса, используя один из выше приведенных способов. Сетевые адреса устройств и маска подсети приведены в табл. 1.1.

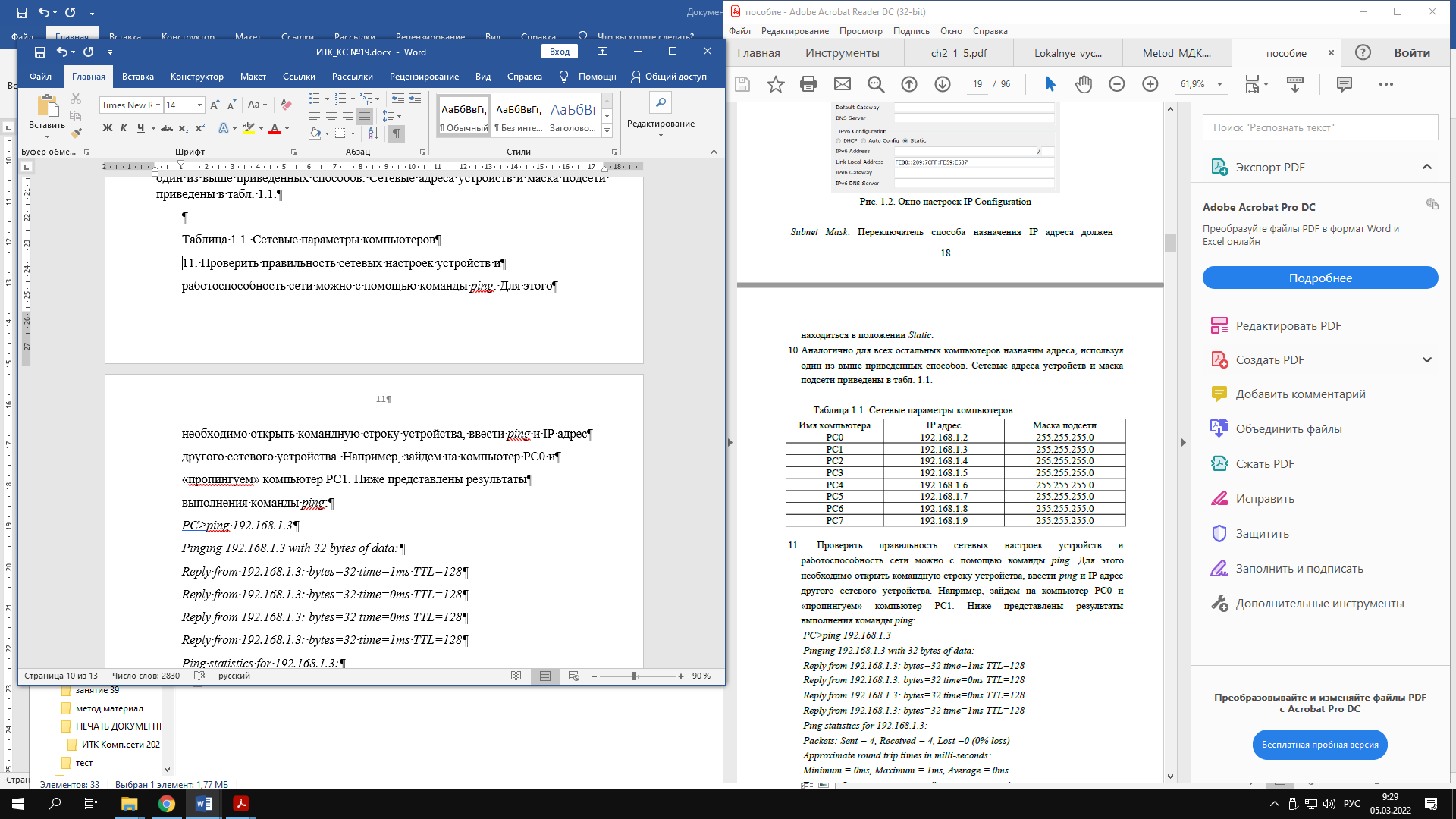


Таблица 1.1. Сетевые параметры компьютеров

11. Проверить правильность сетевых настроек устройств и работоспособность сети можно с помощью команды *ping*. Для этого необходимо открыть командную строку устройства, ввести *ping* и IP адрес другого сетевого устройства. Например, зайдем на компьютер PC0 и «пропингуем» компьютер PC1. Ниже представлены результаты выполнения команды *ping*:

*PC>ping 192.168.1.3*

*Pinging 192.168.1.3 with 32 bytes of data:*

*Reply from 192.168.1.3: bytes=32 time=1ms TTL=128*

*Reply from 192.168.1.3: bytes=32 time=0ms TTL=128*

*Reply from 192.168.1.3: bytes=32 time=0ms TTL=128*

*Reply from 192.168.1.3: bytes=32 time=1ms TTL=128*

*Ping statistics for 192.168.1.3:*

*Packets: Sent = 4, Received = 4, Lost =0 (0% loss)*

*Approximate round trip times in milli-seconds:*

*Minimum = 0ms, Maximum = 1ms, Average = 0ms*

Таким образом, если устройства сети сконфигурированы правильно, можно пропинговать с каждого компьютера любой другой.

12. В среде Packet Tracer существует возможность проследить движение пакетов различных сетевых протоколов с помощью режима симуляции. Для перехода в режим симуляции нужно нажать на кнопку *Simulation Mode* в правом нижнем углу рабочего пространства либо комбинацию клавиш Shift+S.

Справа от рабочей области откроется окно *Simulation Panel* (см. рис. 12), в верхней части которого находится область событий *Event List* и кнопка очистки списка событий *Reset Simulation*. Управление воспроизведением осуществляется с помощью кнопок *Play Controls*. Так, для перехода к следующему событию нужно нажать кнопку *Capture / Forward*. В нижней части окна находится фильтр протоколов.

13. С узла PC1 пропингуем узел PC3. Из предложенного для исследования списка протоколов выберем только протокол ICMP, чтобы исключить случайный трафик между узлами. Откроем командную строку устройства PC1, введем *ping* и IP адрес сетевого устройства PC3. После чего на узле PC1 образуется пакет («конвертик») (рис. 1.3).



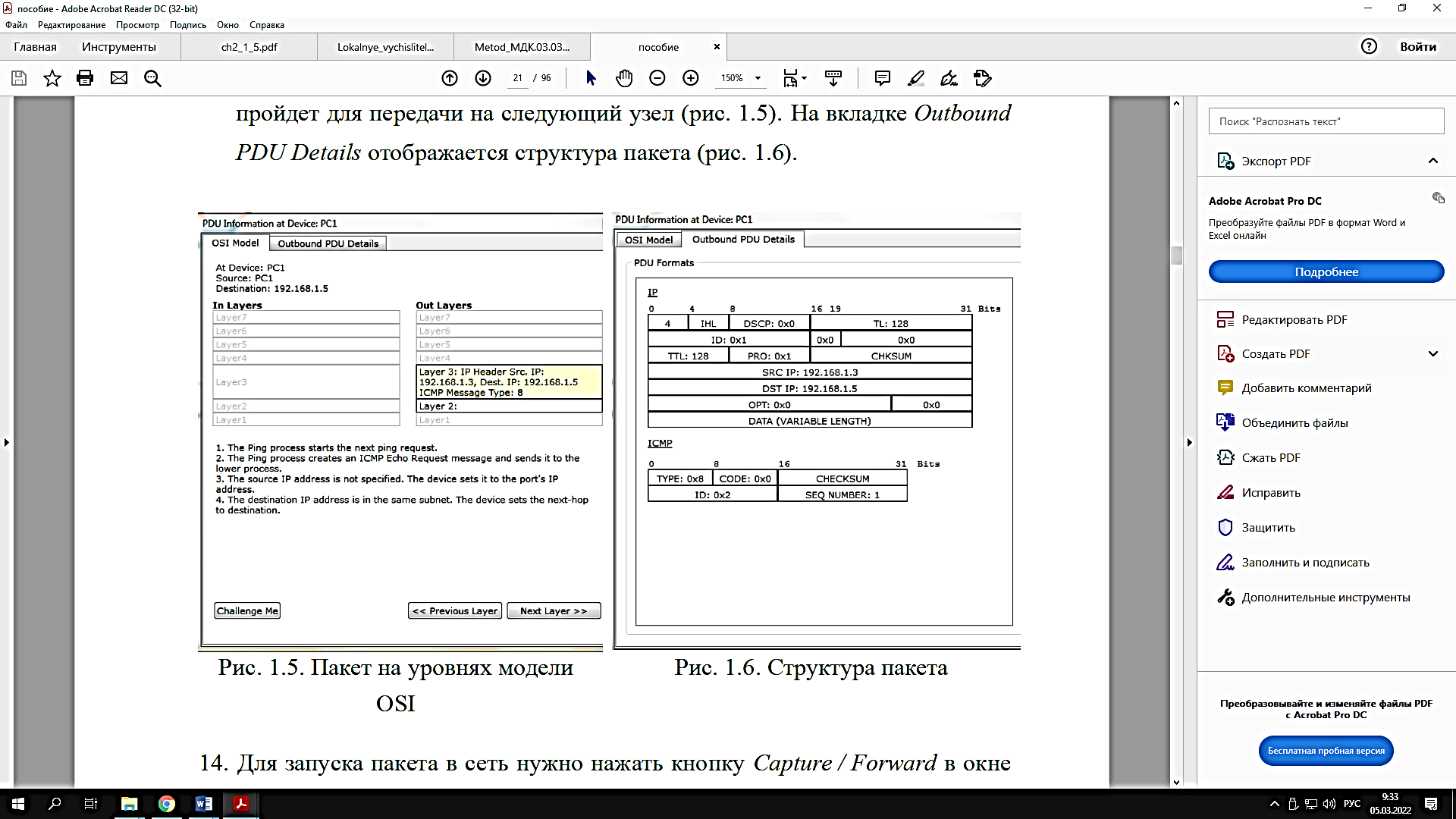
Рис. 1.3. Создание запроса в режиме симуляции

При этом в поле *Event List* появится данный пакет, с указанием его типа (ICMP) и источника формирования (рис. 1.4).



Рис. 1.4. Мониторинг работы протоколов

Для получения подробной информации о пакете нужно щелкнуть на нем левой кнопкой мыши. На вкладке *OSI Model* можно увидеть, на каком уровне сетевой модели OSI был сформирован пакет и какие уровни он пройдет для передачи на следующий узел (рис. 1.5). На вкладке *Outbound PDU Details* отображается структура пакета (рис. 1.6).



14. Для запуска пакета в сеть нужно нажать кнопку *Capture / Forward* в окне *Simulation Panel*. Пакет перейдет на коммутатор Switch0, поскольку это единственное сетевое подключение узла PC1. Коммутатор Switch0 пересылает пакет на коммутатор Switch4. В свою очередь, коммутатор Switch4 пересылает пакет на коммутатор Switch1, после чего Switch1 посылает передаваемый пакет на узел PC3. Получив пакет, PC3 определяет, что он предназначен ему, и, сформировав ответ, посылает пакет на PC1.

После того как PC1 получил ответный пакет от PC3, в окне командной строки появляется следующая запись, сообщающая об успешном прохождении эхо-запроса:

*PC>ping 192.168.1.5*

*Pinging 192.168.1.5 with 32 bytes of data:*

*Reply from 192.168.1.5: bytes=32 time=7ms TTL=128*\_\_

1. **СПРАВОЧНЫЕ ДАННЫЕ**

***Команды протокола FTP.***

1. *Команды управления доступом к системе.*

**USER.** Как правило, эта команда открывает сессию FTP между клиентом и сервером. Аргументом команды является имя (идентификатор) пользователя для работы с файловой системой. Эта команда может подаваться не только в начале, но и в середине сессии, если, например, пользователь желает изменить идентификатор, от имени которого будут проводиться действия. При этом все переменные, относящиеся к старому идентификатору, освобождаются. Если во время изменения идентификатора происходит обмен данными, обмен завершается со старым идентификатором пользователя.

**PASS.** Данная команда подается после ввода идентификатора пользователя и содержит в качестве аргумента пароль пользователя.

**CWD.** Команда обеспечивает возможность работы с различными каталогами удаленной файловой системы. Аргументом команды является строка, указывающая путь каталога удаленной файловой системы, в котором желает работать пользователь.

**REIN.** Команда реинициализации. Эта команда очищает все переменные текущего пользователя, сбрасывает параметры соединения. Если в момент подачи команды происходит передача данных, передача продолжается и завершается с прежними параметрами.

**QUIT.** Команда закрывает управляющий канал. Если в момент подачи команды происходит передача данных, канал закрывается после окончания передачи данных.

2. *Команды управления потоком данных*.

**PORT.** Команда указывает серверу адрес и порт, которые будут использоваться клиентом для прослушивания соединения.

Синтаксис команды:

«PORT<SP>h1,h2,h3,h4,p1,p2». Аргументами команды являются 32-битный IP адрес и 16-битный номер порта соединения. Эти значения разбиты на шесть 8-битных полей и представлены в десятичном виде, где hN – байты адреса (от старшего к младшему), а pN – байты порта (от старшего к младшему).

**PASV.** Эта команда отправляется серверу для указания, что он должен «слушать» соединение. Ответом на данную команду является строка, содержащая адрес и порт узла, находящегося в режиме ожидания соединения. Формат представления данных соответствует формату команды PORT.

Команды **TYPE**, **STRU**, **MODE** определяют, соответственно, тип передаваемых данных (ASCII, Image и другие), структуру или формат передачи данных (*File*, *Record*, *Page*), способ передачи (*Stream*, *Block* и другие). Использование этих команд очень важно при построении взаимодействия в гетерогенных средах и весьма отличающихся операционных и файловых систем взаимодействующих узлов.

3. *Команды ftp-сервиса*.

**RETR.** Эта команда указывает модулю «Программа передачи данных сервера» передать копию файла, заданного параметром этой команды, модулю передачи данных на другом конце соединения.

**STOR.** Команда указывает модулю «Программа передачи данных сервера» принять данные по каналу передачи данных и сохранить их как файл, имя которого задано параметром этой команды. Если такой файл уже существует, он будет перезаписан, если нет, будет создан новый.

**RNFR** и **RNTO.** Команды должны следовать одна за другой. Первая команда содержит в качестве аргумента старое имя файла, вторая – новое. Последовательное применение этих команд переименовывает файл.

*Основные коды ответов на команды протокола FTP*

|  |  |
| --- | --- |
| Код | Описание |
| 110 | Комментарий |
| 125 | Канал открыт, обмен данными начат |
| 150 | Статус файла правилен,подготавливается открытие канала |
| 200 | Команда корректна |
| 220 | Слишком много подключений к FTP-серверу (можете попробовать позднее). В некоторых версиях указывает на успешное завершение промежуточной процедуры |
| 221 | Успешное завершение по команде quit |
| 225 | Канал сформирован,но информационный обмен отсутствует |
| 226 | Закрытие канала, обмен завершен успешно |
| 230 | Пользователь идентифицирован,продолжайте |
| 250 | Запрос прошел успешно |
| 331 | Имя пользователя корректно,нужен пароль |
| 332 | Для входа в систему необходима аутентификация |
| 421 | Процедура невозможна, канал закрывается |
| 425 | Открытие информационного канала невозможно |
| 426 | Канал закрыт, обмен прерван |
| 450 | Запрошенная функция не реализована, файл недоступен, например, занят |
| 451 | Локальная ошибка, операция прервана |
| 452 | Ошибка при записи файла (недостаточно места) |
| 500 | Синтаксическая ошибка, команда не может быть интерпретирована (возможно, она слишком длинна) |
| 501 | Синтаксическая ошибка (неверный параметр или аргумент) |
| 503 | Неудачная последовательность команд |
| 504 | Команда не применима для такого параметра |

**ABOR.** Команда предписывает серверу прервать выполнение предшествующей сервисной команды (например, передачу файла) и закрыть канал передачи данных.

**DELE.** Удаление указанного файла.

**MKD.** Создание указанного в аргументе каталога.

**RMD.**Удаление указанного в аргументе каталога.

**LIST.** Получение списка файлов в указанном каталоге. Передача списка осуществляется по соединению «Программа передачи данных сервера»– «Программа передачи данных клиента».

1. **КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. Сколько логических соединений необходимо для работы протокола FTP? Каким образом они используются?

2. Какие стадии можно выделить в FTP-сессии?

3. В чем заключается особенность работы протокола FTP в активном режиме?

4. В чем заключается особенность работы протокола FTP в пассивном режиме?

5. Какая сторона соединения отвечает за закрытие канала передачи данных? В каких случаях это происходит?

6. Опишите структуру команды и ответа протокола FTP.

7. На какие группы можно разбить команды протокола FTP? Каково назначение команд каждой из групп?

**5. ЛИТЕРАТУРА**

Олифер В.Г. Олифер Н.А., Компюьютерные сети. Учебник //СПБ.:Питер 2016

Преподаватель М.О. Кудрявцева

Рассмотрено на заседании цикловой

комиссии программного обеспечения

информационных технологий №5

Протокол № \_\_\_ от \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Председатель ЦК\_\_\_\_\_\_\_\_ К.О.Якимович