Цели занятия:

Учебные:

1. Знать основные принципы построения современных ЭВМ и вычислительных систем.

Воспитательные:

1. Стимулировать активную познавательную деятельность обучаемых, способствовать формированию творческого мышления, добросовестного отношения к освоению учебного материала, настойчивости и целеустремленности в овладении знаниями.

В ходе занятия формируются следующие компетенции (части компетенций):

Код компетенции	Формируемые компетенции (части компетенций)
ДПК-3	Способность владеть основными методами, способами и средствами получения, хранения, переработки информации и работы в глобальных компьютерных сетях

Учебные вопросы:

1. Классическая многоуровневая модель OSI. Стек протокола TCP/IP.
2. Программа-анализатор трафика WireShark.

Время: 4 академических часа.

Материально-техническое обеспечение:

1. Используемая УМБ: Мультимедийный проектор типа BenQ-232.
2. СИО: УММ к ЛР№2.

Литература:

1. А.С. Шаламов Интегрированная логистическая поддержка наукоемкой продукции. Университетская книга, 2008. – 463 с.

2. Е.В. Судов Интегрированная информационная поддержка жизненного цикла машиностроительной продукции. Принципы. Технологии. Методы. Модели. ООО Издательский дом «МВМ», 2003. – 263 с.

3. Е.В. Судов, А.И. Левин, В.В. Барабанов, А.Н. Давыдов Концепция развития CALS-технологий в промышленности России // М.: ВИМИ, 2002.

Фонд оценочных средств для текущего контроля успеваемости:

Перед занятием:

1. Дайте определение понятию «рендеринг».
2. Перечислите виды рендеринга.
3. Перечислите основные методы рендеринга.
4. Дайте определение понятию «экструдирование объектов».
5. Виды анимации в Blender.

В ходе и по окончании занятия ответить на контрольные вопросы.

Оценивание осуществляется по следующей шкале:

100-балльная шкала	Результат освоения
Менее 40	Критерий не сформирован
41-70	Критерий четко не выражен
71-100	Критерий выражен четко

Для оценивания ситуационных заданий используется следующая шкала:

100-балльная шкала	Результат освоения
Менее 30	Обучающийся не может сформулировать проблему,

	представленную в задании
31-50	Обучающийся формулирует поставленную задачу, у него сформированы изолированные знания и умения, однако отсутствуют интегрированные понятия и навыки, в результате чего допущены ошибки в решении и задание не выполнено
51-80	Задание выполнено, обучающийся применяет знания для решения поставленной проблемы, однако не сформированы компетенции, вследствие чего обучающийся испытывает затруднения в демонстрации способов решения задачи
81-100	Задание выполнено как в теоретическом, так и в практическом плане, обучающийся легко демонстрирует свою компетентность по данному вопросу

ВВОДНАЯ ЧАСТЬ

Модель взаимодействия открытых систем (Open System Interconnection, OSI) определяет различные уровни взаимодействия систем в сетях с коммутацией пакетов, дает им стандартные имена и указывает, какие функции должен выполнять каждый уровень.

В модели OSI средства взаимодействия делятся на семь уровней:

1. прикладной,
2. представительный,
3. сеансовый,
4. транспортный,
5. сетевой,
6. канальный,
7. физический.

Вопрос №1. Классическая многоуровневая модель OSI. Стек протокола TCP/IP.

1.1. Изучение всех уровней модели OSI.

Физический уровень (Physical layer).

Самый нижний уровень модели, предназначен непосредственно для передачи потока данных. Осуществляет передачу электрических или оптических сигналов в кабель и соответственно их приём и преобразование в биты данных в соответствии с методами кодирования цифровых сигналов. Другими словами, осуществляет интерфейс между сетевым носителем и сетевым устройством. На этом уровне работают концентраторы и повторители (ретрансляторы) сигнала.

Канальный уровень (Data Link layer).

Этот уровень предназначен для обеспечения взаимодействия сетей на физическом уровне и контроле за ошибками, которые могут возникнуть. Полученные данные от физического уровня он упаковывает в кадры данных, проверяет на целостность, если нужно исправляет ошибки и отправляет на сетевой уровень. Канальный уровень может взаимодействовать с одним или несколькими физическими уровнями, контролируя и управляя этим взаимодействием. Спецификация IEEE 802 разделяет этот уровень на 2 подуровня – MAC (Media Access Control) регулирует доступ к разделяемой физической среде, и LLC (Logical Link Control) обеспечивает обслуживание сетевого уровня. На этом уровне работают коммутаторы, мосты и сетевые адаптеры.

В программировании этот уровень представляет драйвер сетевой платы, в операционных системах имеется программный интерфейс взаимодействия канального и сетевого уровня между собой, это не новый уровень, а просто реализация модели для конкретной ОС. Примеры таких интерфейсов: ODI, NDIS.

Сетевой уровень (Network layer).

3-й уровень сетевой модели OSI, предназначен для определения пути передачи данных. Отвечает за трансляцию логических адресов и имён в физические, определение кратчайших маршрутов, коммутация и маршрутизация пакетов, отслеживание неполадок и заторов в сети. На этом уровне работает такое сетевое устройство, как маршрутизатор.

Транспортный уровень (Transport layer).

4-й уровень модели, предназначен для доставки данных без ошибок, потерь и дублирования в той последовательности, как они были переданы. При этом неважно какие данные передаются, откуда и куда, то есть он предоставляет сам механизм передачи. Блоки данных он разделяет на фрагменты, размер которых зависит от протокола, короткие объединяет в один, длинные разбивает. Протоколы этого уровня предназначены для взаимодействия типа точка-точка.

Сеансовый уровень (Session layer).

Отвечает за поддержание сеанса связи, позволяя приложениям взаимодействовать между собой длительное время. Уровень управляет созданием/завершением сеанса, обменом информацией, синхронизации задач, определением права на передачу данных и поддержание сеанса в периоды неактивности приложений. Синхронизация передачи обеспечивается помещением в поток данных контрольных точек, начиная с которых возобновляется процесс при нарушении взаимодействия.

Уровень представления (Presentation layer).

Этот уровень отвечает за преобразование протоколов и кодирование/декодирование данных. Запросы приложений, полученные с уровня приложений, он преобразует в формат для передачи по сети, а полученные из сети данные преобразует в формат, понятный приложениям. На этом уровне может осуществляться сжатие/распаковка или кодирование/раскодирование данных, а также перенаправление запросов другому сетевому ресурсу, если они не могут быть обработаны локально.

Прикладной уровень (Application layer).

Верхний (7-й) уровень модели, обеспечивает взаимодействие сети и пользователя. Уровень разрешает доступ к сетевым службам приложениям пользователя, таким как обработчик запросов к базам данных, доступ к файлам, пересылке электронной почты. Также отвечает за передачу служебной информации, предоставляет приложениям информацию об ошибках и формирует запросы к уровню представления.

Задание 1. Необходимо расставить по уровням модели OSI следующее (отчет представить в виде таблиц с распределением предложенных составных частей модели OSI по уровням):

- повторитель (repeater);
- концентратор (hub);
- мост (bridge);
- коммутатор (switch);
- маршрутизатор (router);
- шлюз (gateway);
- разъем RJ-45;
- MAC-адрес;
- IP-адрес;
- документ RFC792;
- стандарт IEEE 802.3;
- единицу данных «кадр» (frame);
- единицу данных «пакет» (packet);
- единицу данных «сообщение» (message);
- протокол SSL;
- протокол SPX;
- протокол HTTP;
- протокол ARP;
- протокол OSPF;
- протокол PPP;
- стек протоколов NetBIOS/SMB.

1.2. Стек протоколов TCP/IP

TCP/IP – собирательное название для набора (стека) сетевых протоколов разных уровней, используемых в Интернет. Особенности TCP/IP:

- открытые стандарты протоколов, разрабатываемые независимо от программного и аппаратного обеспечения;
- независимость от физической среды передачи;
- система уникальной адресации;

- стандартизованные протоколы высокого уровня для распространенных пользовательских сервисов.

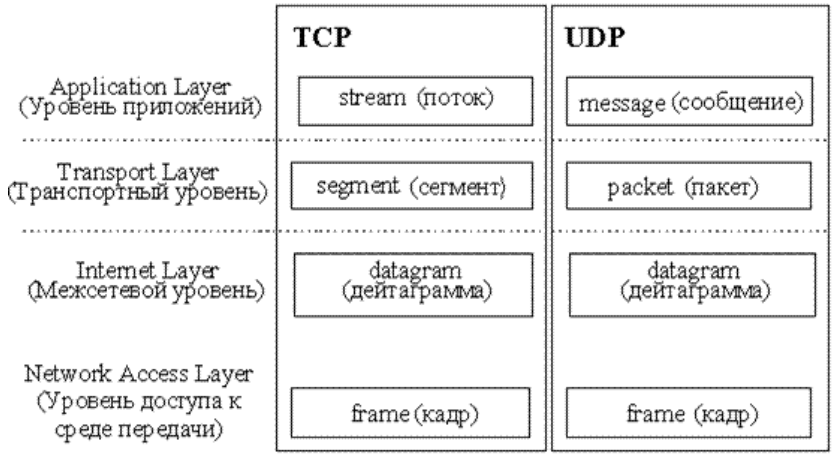

Рис. 1. Уровни стека протоколов TCP/IP

Стек протоколов TCP/IP делится на 4 уровня: прикладной (application), транспортный (transport), межсетевой (internet) и уровень доступа к среде передачи (network access). Термины, применяемые для обозначения блока передаваемых данных, различны при использовании разных протоколов транспортного уровня – TCP и UDP, поэтому на рисунке 1 изображено два стека. Как и в модели OSI, данные более верхних уровней инкапсулируются[1] в пакеты нижних уровней (см. рис. 2).

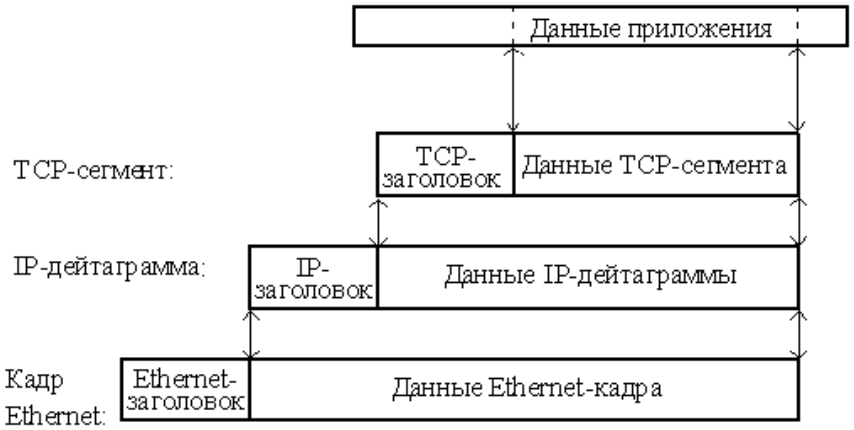

Рис. 2. Пример инкапсуляции пакетов в стеке TCP/IP

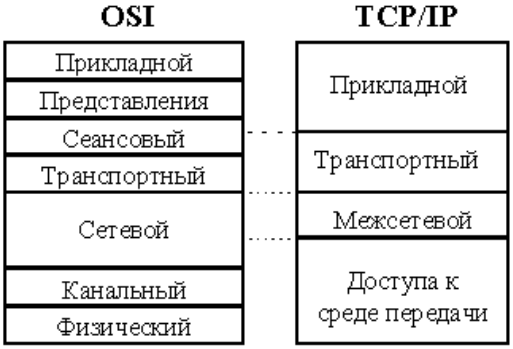

Рис. 3. Соотношение уровней стеков OSI и TCP/IP.

[1] Инкапсуляция – это упаковка данных и функций в один компонент (например, класс) и последующий контроль доступа к этому компоненту, создавая тем самым «чёрный ящик» из объекта. По этой причине, пользователю необходимо знать только интерфейс этого класса (то есть данные и функции, предоставляемые для взаимодействия с классом извне), а не то, как он реализован внутри.

Примечание. Принцип функционирования протоколов в стеке TCP/IP (собственно говоря, это справедливо и для остальных протоколов) никак не зависит от операционной системы!

Ниже кратко рассматриваются функции каждого уровня и примеры протоколов. Программа, реализующая функции того или иного протокола, часто называется модулем, например, «IP-модуль», «модуль TCP».

Уровень приложений. Приложения, работающие со стеком TCP/IP, могут также выполнять функции уровней представления и частично сеансового модели OSI; например, преобразование данных к внешнему представлению, группировка данных для передачи и т. п.

Распространенными примерами приложений являются программы telnet, ftp, HTTP-серверы и клиенты, программы работы с электронной почтой и др.

Для пересылки данных другому приложению, приложение обращается к тому или иному модулю транспортного уровня.

Транспортный уровень. Протоколы транспортного уровня обеспечивают прозрачную (сквозную) доставку данных (end-to-end delivery service) между двумя прикладными процессами. Процесс, получающий или отправляющий данные с помощью транспортного уровня, идентифицируется на этом уровне номером, который называется номером порта. Таким образом, роль адреса отправителя и получателя на транспортном уровне выполняет номер *порта* (см. далее).

Анализируя заголовок своего пакета, полученного от межсетевого уровня, транспортный модуль определяет по номеру порта получателя, какому из прикладных процессов направлены данные, и передает эти данные соответствующему прикладному процессу (возможно, после проверки их на наличие ошибок и т.п.). Номера портов получателя и отправителя записываются в заголовок транспортным модулем, отправляющим данные; заголовок транспортного уровня содержит также и другую служебную информацию; формат заголовка зависит от используемого транспортного протокола.

На транспортном уровне работают два основных протокола: UDP и TCP.

TCP (Transmission Control Protocol – протокол контроля передачи, *RFC 793*) – это транспортный механизм, предоставляющий поток данных, с предварительной установкой соединения, за счёт этого дающий уверенность в безошибочности получаемых данных, осуществляет повторный запрос данных в случае потери пакетов и устраняет дублирование при получении двух копий одного пакета. Естественно, что в общем случае данные не могут быть гарантировано доставлены до адресата; в таком случае клиентский процесс получает об этом уведомление.

Данными для TCP является не интерпретируемая протоколом последовательность пользовательских октетов, разбиваемая для передачи по частям. Каждая часть передается в отдельном TCP-сегменте. Для продвижения сегмента по сети между компьютером-отправителем и компьютером-получателем модуль TCP пользуется сервисом межсетевого уровня (вызывает модуль IP). Протокол TCP гарантирует, что приложение получит данные точно в такой же последовательности, в какой они были отправлены, и без потерь.

UDP (User Datagram Protocol, протокол пользовательских дейтаграмм, *RFC* 768) фактически не выполняет каких-либо особых функций дополнительно к функциям межсетевого уровня (протокола IP см. далее). Протокол UDP используется либо при пересылке коротких сообщений, когда накладные расходы на установление сеанса и проверку успешной доставки данных оказываются выше расходов на повторную (в случае неудачи) пересылку сообщения, либо в том случае, когда сама организация процесса-приложения обеспечивает установление соединения и проверку доставки пакетов.

Пользовательские данные, поступившие от прикладного уровня, предваряются UDP-заголовком, и сформированный таким образом UDP-пакет отправляется на межсетевой уровень.

Межсетевой уровень и протокол IP. Основным протоколом этого уровня является протокол IP (Internet Protocol, *RFC 791*).

Протокол IP доставляет блоки данных, называемых дейтаграммами, от одного сетевого узла к другому.

В современной сети Интернет используется IP четвёртой версии, также известный как IPv4. В протоколе IP этой версии каждому узлу сети ставится в соответствие IP-адрес длиной 4 октета (иногда говорят «байта», подразумевая распространённый восьмибитовый минимальный адресуемый фрагмент памяти ЭВМ).

В настоящее время вводится в эксплуатацию шестая версия протокола — IPv6, которая позволяет адресовать значительно большее количество узлов, чем IPv4. Эта версия отличается повышенной разрядностью адреса, встроенной возможностью шифрования и некоторыми другими особенностями. Переход с IPv4 на IPv6 связан с трудоёмкой работой операторов связи и производителей программного обеспечения и не может быть выполнен одномоментно.

Данные для IP дейтаграммы передаются IP-модулю транспортным уровнем. IP-модуль предваряет эти данные заголовком, содержащим IP-адреса отправителя и получателя и другую служебную информацию, и сформированная таким образом дейтаграмма передается на уровень доступа к среде передачи (например, одному из физических интерфейсов) для отправки по каналу передачи данных.

Не все сетевые узлы могут непосредственно связаться друг с другом; часто для того, чтобы передать дейтаграмму по назначению, требуется направить ее через один или несколько промежуточных узлов по тому или иному маршруту. Задача определения маршрута для каждой дейтаграммы решается протоколом IP.

Когда модуль IP получает дейтаграмму с нижнего уровня, он проверяет IP-адрес назначения. Если дейтаграмма адресована данному компьютеру, то данные из нее передаются на обработку модулю вышестоящего уровня (какому конкретно – указано в заголовке дейтаграммы). Если же адрес назначения дейтаграммы – чужой, то модуль IP может принять два решения: первое – уничтожить дейтаграмму, второе – отправить ее дальше к месту назначения, определив маршрут следования – так поступают промежуточные станции – маршрутизаторы.

Также может потребоваться, на границе сетей с различными характеристиками, разбить дейтаграмму на фрагменты, а потом собрать в единое целое на компьютере-получателе. Это тоже задача протокола IP.

Если модуль IP по какой-либо причине не может доставить дейтаграмму, она уничтожается. При этом модуль IP может отправить компьютеру-источнику этой дейтаграммы уведомление об ошибке; такие уведомления отправляются с помощью протокола ICMP, являющегося неотъемлемой частью модуля IP. Более никаких средств контроля корректности данных, подтверждения их доставки, обеспечения правильного порядка следования дейтаграмм, предварительного установления соединения между компьютерами протокол IP не имеет. Эта задача возложена на транспортный уровень.

Уровень доступа к среде передачи. Функции этого уровня:

- отображение IP-адресов в физические адреса сети (MAC-адреса, например, Ethernet-адрес в случае сети Ethernet). Эту функцию выполняет протокол ARP;

- инкапсуляция IP-дейтаграмм в кадры для передачи по физическому каналу и извлечение дейтаграмм из кадров. При этом не требуется какого-либо контроля безошибочности передачи (хотя он может и присутствовать), поскольку в стеке TCP/IP такой контроль возложен на транспортный уровень или на само приложение. В заголовке кадров указывается точка доступа к сервису (SAP, Service Access Point) - поле, содержащее код протокола межсетевого уровня, которому следует передать содержимое кадра (в нашем случае это протокол IP);

- определение метода доступа к среде передачи - то есть способа, с помощью которого компьютер устанавливает свое право на произведение передачи данных (передача токена[2], опрос компьютеров, множественный доступ с детектированием коллизий и т. п.).

[2] Токен — это цифровой актив (сертификат), который представляет определенную стоимость, функционирует на основе блокчейна или другой децентрализованной сети и гарантирует обязательства компании перед его владельцем. Токены могут быть использованы для предоставления доли в проекте, доступа к определенным услугам или продуктам, наград и так далее.

- определение представления данных в физической среде;
- пересылка и прием кадра.

Стек TCP/IP не подразумевает использования каких-либо определенных протоколов уровня доступа к среде передачи и физических сред передачи данных. От уровня доступа к среде передачи требуется наличие интерфейса с модулем IP, обеспечивающего передачу дейтаграммы между уровнями. Также требуется обеспечить преобразование IP-адреса узла сети, на который передается дейтаграмма, в MAC-адрес. Часто в качестве уровня доступа к среде передачи могут выступать целые протокольные стеки, тогда говорят об IP поверх ATM, IP поверх IPX, IP поверх X.25 и т. п.

Обобщенная модель взаимодействия узлов на базе протоколов TCP/IP представлена на рис 4.

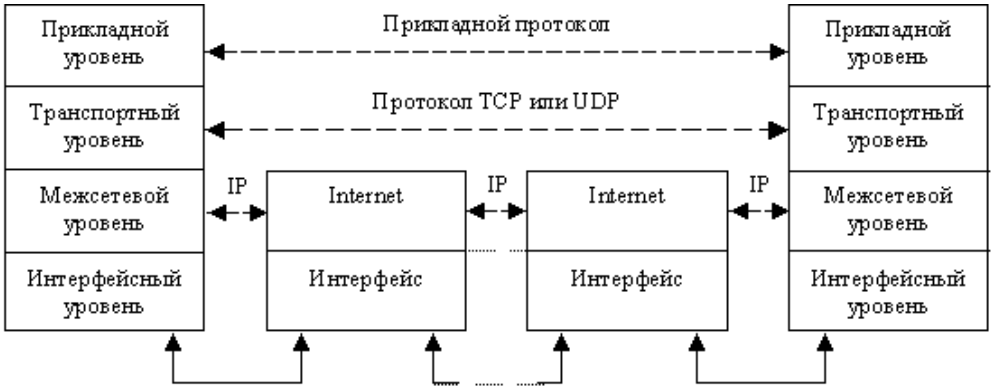

Рис. 4. Модель взаимодействия стеков TCP/IP

Понятие сетевых портов и сокетов

Основные прикладные сетевые сервисы используют средства транспортного уровня для взаимодействия.

Любые 2 сетевых процесса могут идентифицировать друг друга при помощи 3-х компонент: ip-адрес, протокол (TCP/UDP), порт. Часто данные компоненты носят название *сокетами*.

Сокеты – это название программного интерфейса для обеспечения информационного обмена между процессами. Т.е. для прикладных сетевых процессов взаимодействие осуществляется через сокеты. Рассмотрим понятие портов более подробно.

Порт – параметр протоколов TCP и UDP, определяющий пункт назначения для данных, принимаемых по сети. Порту сопоставляется номер от 1 до 65535, позволяющие различным программам, выполняемым на одном хосте, получать данные независимо друг от друга. В этом случае каждая из них обрабатывает данные, поступающие на определённый порт (иногда говорят, что программа «слушает» на том или ином порту).

Согласно IP, в каждом пакете присутствуют IP адрес узла-источника и IP адрес узла-назначения. В TCP/UDP пакетах дополнительно указываются порт источника и порт назначения. Узел назначения, получив пакет, смотрит на порт назначения и передает пакет соответствующему у себя приложению. Использование портов позволяет независимо использовать TCP/UDP протокол сразу многим приложениям на одном и том же компьютере.

Для сетевых приложений нотация указания порта следующая: «ip**:**port». Например, http://web-service.org:8888

Пояснение понятия портов представлено на рис. 5. На самом деле сетевой порт – это всего лишь числовой параметр в сетевом пакете протоколов TCP и UDP. Такие понятия как «открыть порт» означают что пакеты, адресованные на данный порт, будут приниматься на обработку.

Порты из диапазона 1-1024 являются привилегированными. Называются они так, потому что для их открытия (и, соответственно, запуска соответствующих сетевых сервисов) на большинстве ОС требуются права системного администратора. Большая часть привилегированных портов распределена для общеупотребительных сетевых протоколов. В табл. 1 перечислены некоторые протоколы и порты, за которыми они закреплены. Данные

порты являются портами по умолчанию для соответствующих служб и чаще всего не перенастраиваются.

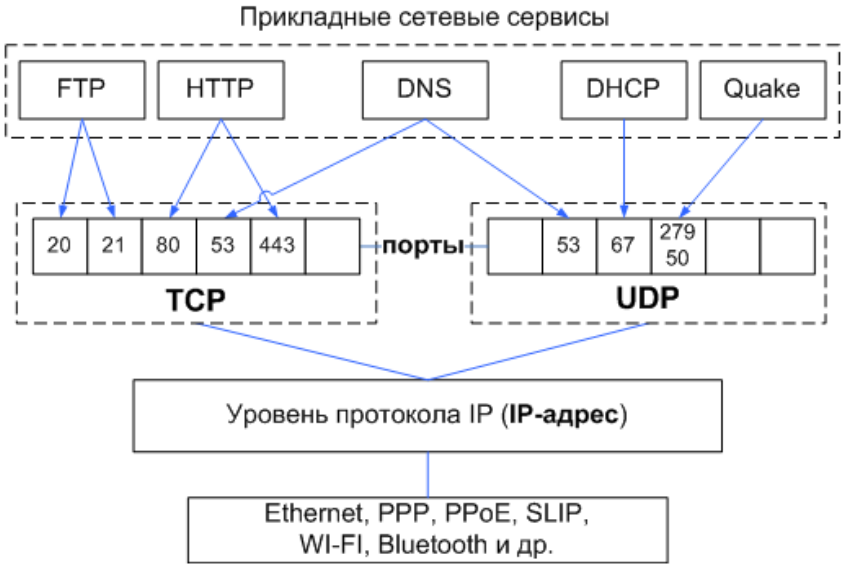

Рис. 5. Компоненты сокетов

Табл. 1 Примеры некоторых стандартных сетевых портов

Порт / Протокол	Сервис	Описание
20/TCP	ftp-data	Порт данных FTP
21/TCP	ftp	Порт протокола передачи файлов (File Transfer Protocol, FTP); иногда используется протоколом файловой службы (File Service Protocol, FSP)
22/TCP	ssh	Служба Безопасной Оболочки (Secure SHell, SSH)
23/TCP	telnet	Служба Telnet
25/TCP	smtp	Протокол простой передачи почты (Simple Mail Transfer Protocol, SMTP)
53/(UDP, TCP)	domain	Службы доменных имён (такие как BIND)
80/TCP	http	Протокол передачи гипертекста (HyperText Transfer Protocol, HTTP) для служб всемирной паутины (World Wide Web, WWW)
110/TCP	pop3	Протокол почтового отделения (Post Office Protocol) версии 3
443/TCP	https	Протокол HTTP поверх SSL
992/TCP	telnets	Telnet поверх SSL (TelnetS)
993/TCP	imaps	IMAP поверх SSL (IMAPS)
994/TCP	ircs	IRC поверх SSL (IRCS)
995/TCP	pop3s	POP 3 поверх SSL (POP3S)

Мониторинг сетевой активности и анализ работы сетевых приложений

Для закрепления знаний на практике рассмотрим с вами ряд полезных сетевых утилит, которые позволяют вести мониторинг состояния стека TCP/IP.

Первой утилитой, которую рассмотрим, будет утилита **netstat** (см. ПЗ№1). Данная утилита позволяет получать информацию об активных сетевых соединениях на уровне TCP и UDP протоколов, получать базовую статистику по количеству переданных и полученных пакетов уровня IP и т. д. Синтаксис команды можно почерпнуть из встроенной справки в саму утилиту («netstat /?»)

Рассмотрим несколько примеров.

«netstat - a» – получение информации обо всех установленных соединениях и открытых на прослушивание портах (см. рис. 6)

«netstat - n - b» – получение информации обо всех активных соединениях и процессах, инициировавших их (см. рис. 7)

«netstat - e - s» – получение основной статистики по всем протоколам (ethernet, IPv4, IPv6, TCP, UDP) (см. рис. 8)

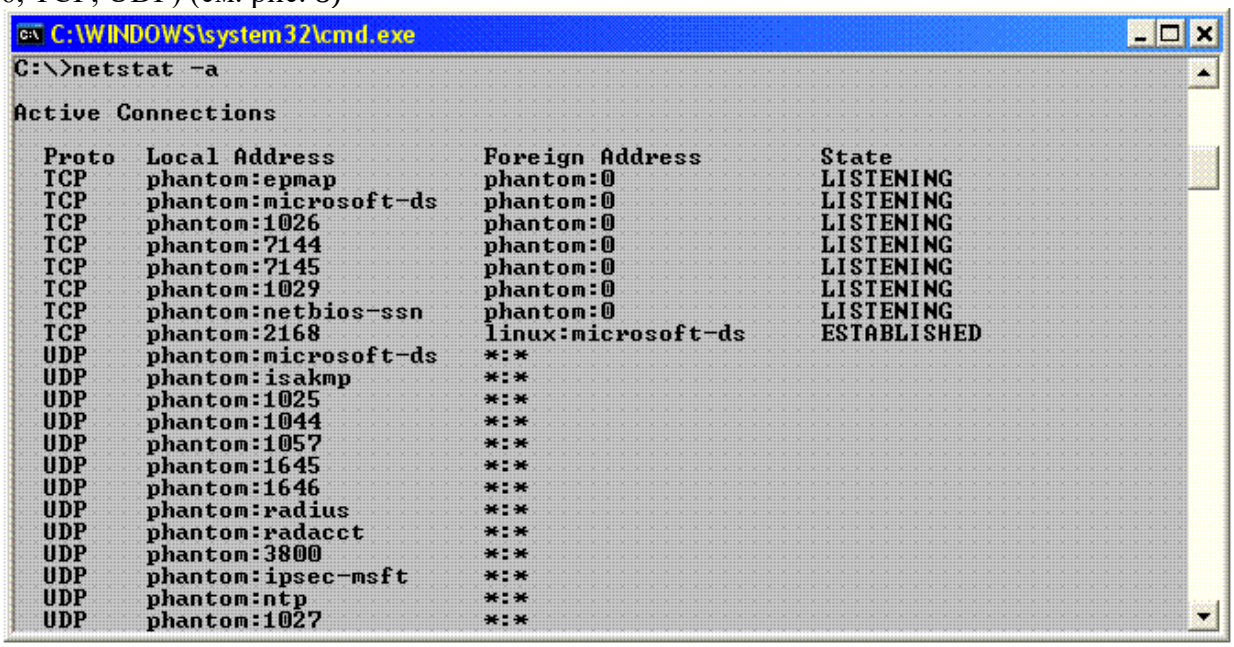

Рис. 6. Список всех установленных TCP/UDP соединений

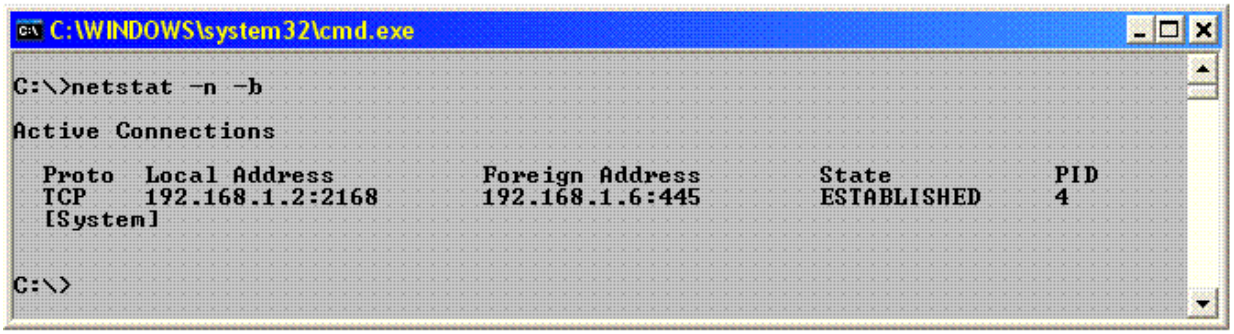

Рис. 7. Список активных соединений с указанием процесса

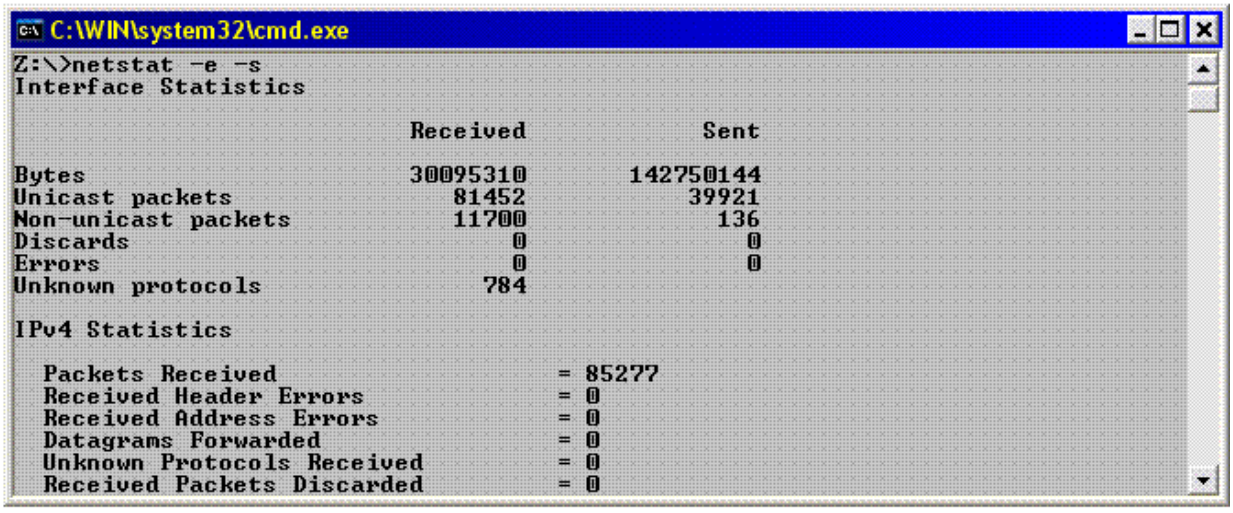

Рис. 8. Статистика по протоколам.

Очень удобным аналогом утилиты netstat для отслеживания активных соединений является утилита TCPView (см. рис. 9). Она позволяет в интерактивном режиме отслеживать сетевые соединения, а также позволяет получать информацию о процессах, установивших соединение и завершать их в случае необходимости. Утилита является бесплатной и ее можно

скачать с сайта фирмы Майкрософт (www.microsoft.com/technet/sysinternals/utilities/tcpview.mspx)

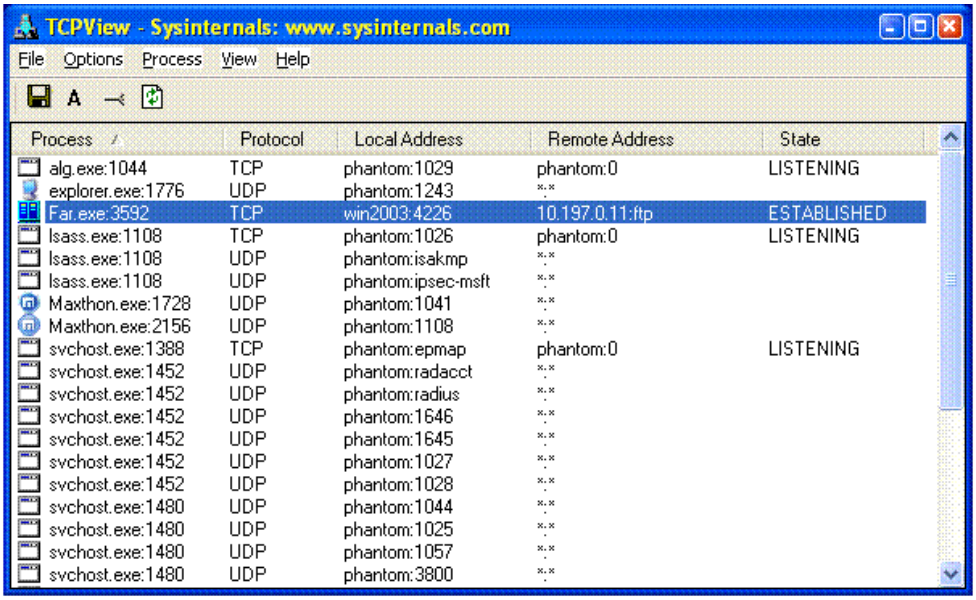

Рис. 9. Утилита TCPView.

Сканеры портов

Часто возникает необходимость узнать, какие сетевые сервисы запущены на удаленной машине. Для решения данной задачи служат т. н. сканеры портов. Данная группа утилит позволяет с некоторой точностью узнать, какие порты открыты на удаленной машине и некоторые другие параметры. Зная номера портов зачастую, можно с достаточно большой уверенностью предположить, какие сервисы запущены на удаленной машине.

Наиболее достоверная информация может быть получена для протокола TCP. Задача детектирования UDP-сервисов не всегда может быть решена. Поэтому высокая точность сканирования UDP-сервисов не гарантируется.

Одним из самых распространенных профессиональных сканеров портов является сканер «NMAP» (см. рис. 10)

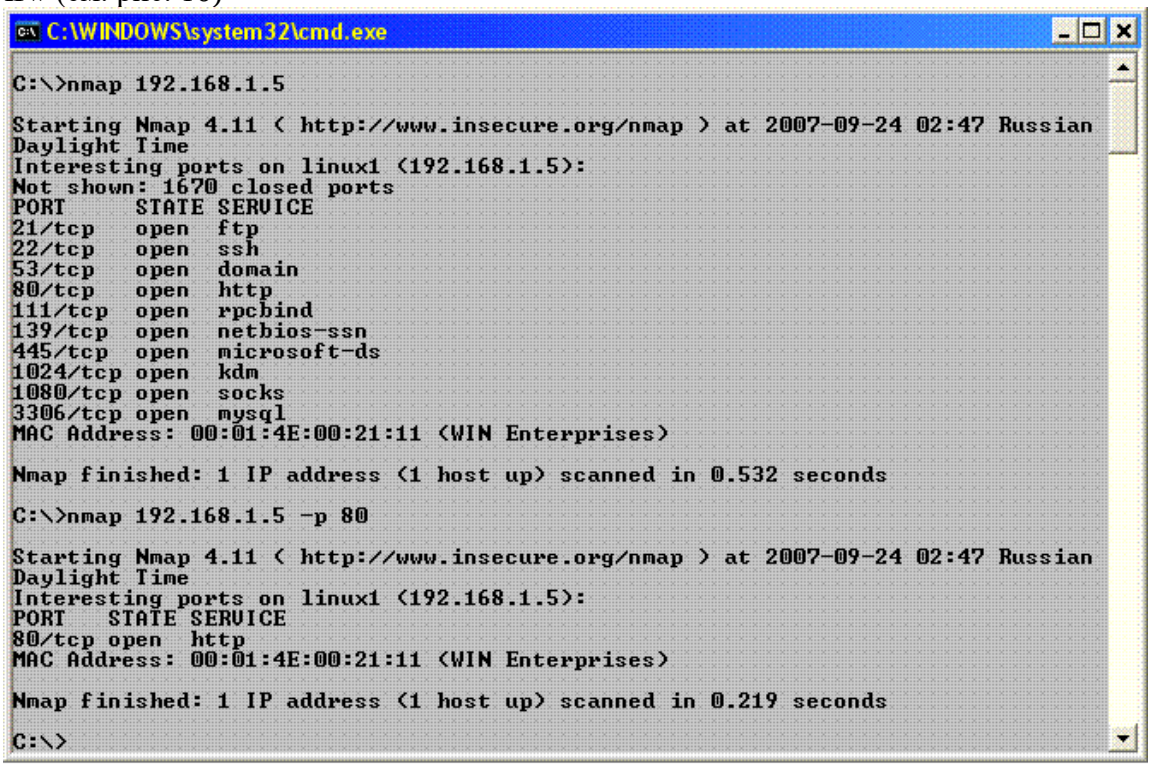

Рис. 10. Сканер портов *nmap*

Сканер *nmap* портирован на большинство распространенных платформ: windows, linux, FreeBSD, OpenBSD и т. д.

Контрольные вопросы

1. Что такое сетевой протокол?
2. Зачем необходима стандартизация протоколов?
3. Понятие стека протоколов
4. Зачем введена модель OSI/ISO
5. Перечислите уровни стека протоколов TCP/IP и кратко охарактеризуйте их назначение.
6. Что такое IP-адрес?
7. В чем принципиальное отличие протоколов TCP и UDP.
8. Что такое сокет?
9. Зачем введен механизм сетевых портов?
10. Есть ли различие в протоколах реализованных, например, для ОС Windows и Linux?

Задание на лабораторную работу

1. Изучить стек протоколов TCP/IP.
2. Найти описание протоколов IP, TCP и UDP в соответствующих RFC.
3. Рассчитать примерную эффективность использования пропускной способности сетевого канала при использовании протоколов TCP и UDP для пакетов различной длины. Построить график эффективности (эффективность передачи от размера полезных данных). При этом считать максимальную полезную длину Ethernet-сегмента (MSS – maximum segment size) в 1500 байт. Параметры заголовков взять из описания протоколов в RFC.
4. Изучить утилиты netstat и tcpview: проанализировать текущие сетевые соединения на сетевой машине, получить статистику по протоколам (только netstat).

Вопрос №2. Программа-анализатор трафика WireShark.

Постановка задачи:

В соответствии с заданием требуется проанализировать трафик, захваченный программой *Wireshark,* а именно:

1) рассмотреть структуру пакета, указав назначение каждого заголовка;

2) пояснить механизм инкапсуляции протоколов. В отчете привести скриншоты, иллюстрирующие ответы на поставленные в задании вопросы (также пакет можно распечатать прямо из программы *Wireshark*). Необходимо иметь с собой на flash-носителе сохраненную версию захваченного трафика (так называемый дамп трафика) в формате pcap (это стандартный формат сохранения трафика в Wireshark). Во всех вариантах задания необходимо выполнить следующие этапы исследования.

Протокол IP

1) запустить *Wireshark;*

2) запустить процесс захвата трафика;

3) в командной строке:

tracert конечный_узел

например, tracert wireshark.org

в качестве конечного узла использовать URL, в котором по очереди встречаются инициалы фамилии и имени студента в латинской транскрипции (например, для имени Пётр Иванов подойдут адреса сайтов http://pictures.com или http://nopix.ru/;

4) остановить захват трафика. В информационном поле разверните строку, содержащую «Internet Protocol».

Ответьте на следующие вопросы.

1. Проанализируйте первый пакет ICMP Echo Request, отправленный вашим компьютером: укажите ваш IP-адрес.
2. Приведите значение поля, определяющее протокол верхнего уровня.

3. Сколько байт содержится в заголовке IP? Сколько байт в поле данных?
4. Укажите значение TTL. Как изменяется это поле в разных ICMP Echo Request?
5. Какое значение содержится в поле «Identification»? Для чего используется это поле?

Фрагментация пакетов

1) запустить *Wireshark;*
2) запустить процесс захвата трафика;
3) в командной строке:
ping –l 2000 конечный_узел (ключ -l позволяет указать загрузку поля «Data» пакета в байтах)
например, ping –l 2000 wireshark.org
в качестве конечного узла использовать URL, в котором присутствуют любые три буквы из фамилии студента в латинской транскрипции;
4) остановить захват трафика.

Ответьте на следующие вопросы.

1. Проанализируйте пакет ICMP Echo Request: имеет ли место фрагментация исходного пакета, какое поле на это указывает?
2. Проанализируйте фрагменты IP-дейтаграммы: какая информация указывает, является ли фрагмент пакета последним или промежуточным?
3. Укажите количество фрагментов исходного пакета.

Описание программы Wireshark

Для выполнения лабораторной работы необходимо установить на компьютер программу-анализатор сетевых пакетов Wireshark. На рис. 1 представлено главное окно Wireshark.

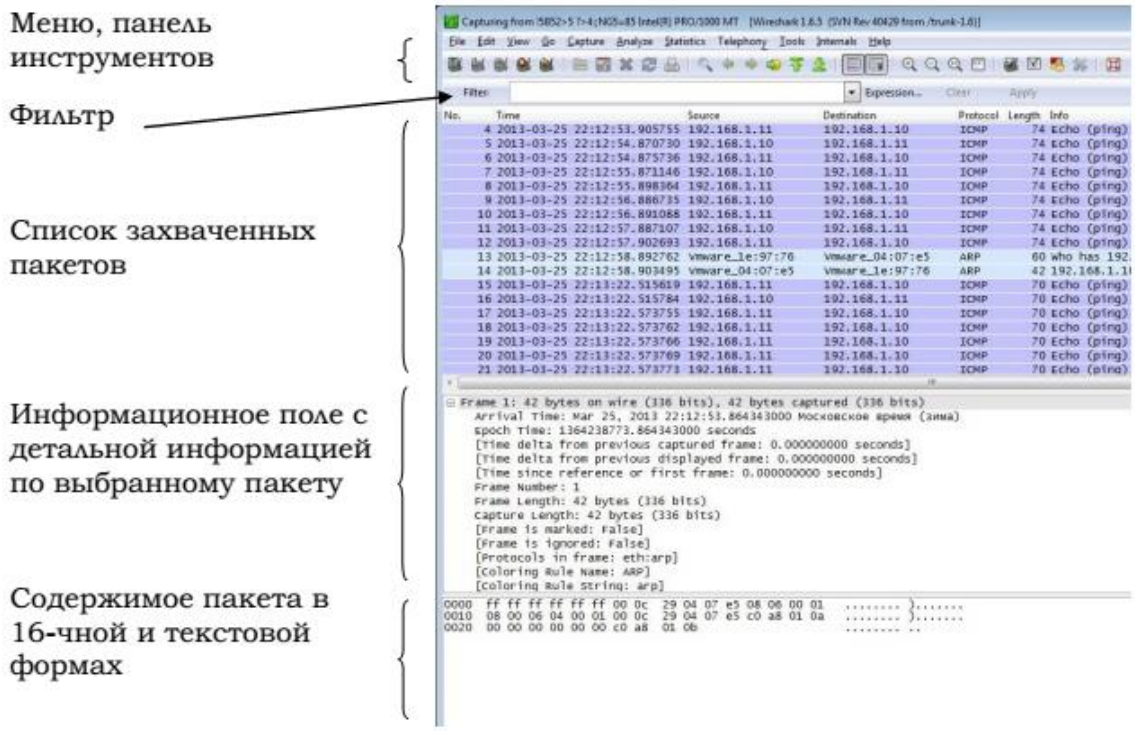

Рис. 1. Wireshark GUI

Начать работу с Wireshark следует следующим образом:
1) открыть браузер;
2) запустить *Wireshark*:
a) установить параметры для захвата трафика; в качестве интерфейса, используемого для захвата трафика, выбрать физический адаптер, тип адаптера — Local (рис. 2);
b) запустить процесс захвата трафика (кнопка *Start*);
3) в браузере открыть любой сайт (например, http://www.wireshark.org/docs/wsug_html_chunked/);

4) установить значение фильтра, равным http;

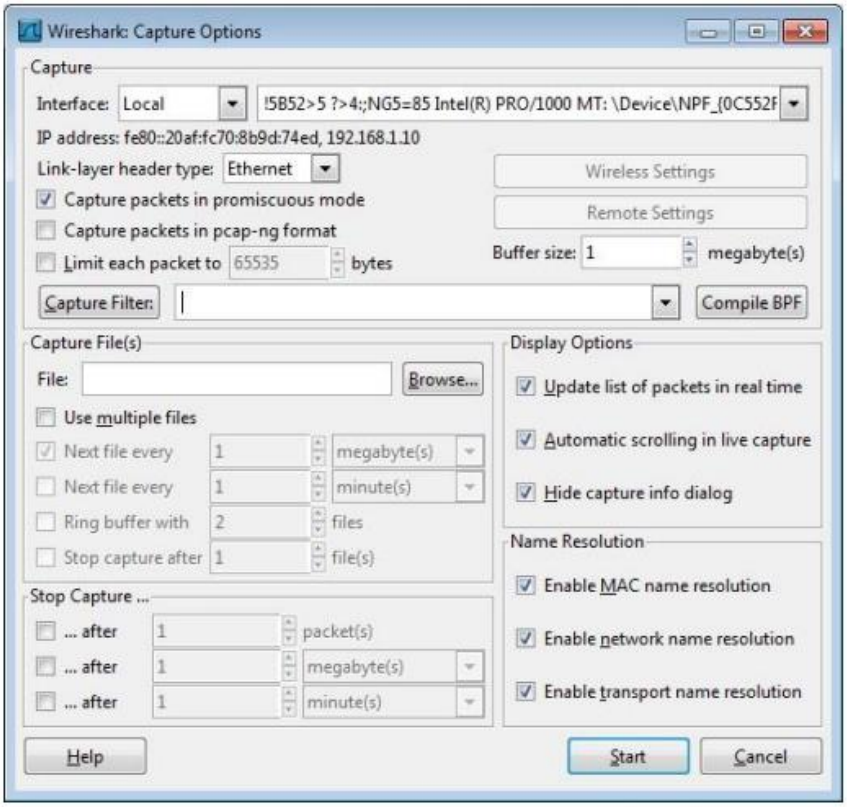

Рис. 2. Параметры захвата трафика

5) выбрать первое http сообщение в списке пакетов — это будет сообщение HTTP GET, отправленное на указанный хост (например, www.wireshark.org); в информационном поле отображена детальная информация по заголовкам пакета.

Варианты задания

Вариант 1. HTTP: Basic HTTP GET/response

1) запустить *Wireshark;*

2) настроить фильтр (http);

3) запустить процесс захвата трафика;

4) URL: например, http://wiki.wireshark.org/;

в URL должны присутствовать любые три буквы из фамилии студента в латинской транскрипции;

5) остановить захват трафика.

Прим.: Не принимать во внимание HTTP запрос и ответ для favicon.ico. Появление ссылки на данный файл означает, что браузер автоматически запрашивает сервер о наличии маленького значка веб-сайта, т.н. «Favicon» (отображается браузером в адресной строке перед URL страницы, а также в качестве картинки рядом с закладкой, во вкладках и в других элементах интерфейса).

В списке захваченных пакетов найдите пару HTTP сообщений (запрос-ответ): GET сообщение и ответ сервера. В информационном поле разверните строку, содержащую HTTP, и отметьте указанную ниже информацию.

1. Версия HTTP.
2. Принимаемые браузером языки.
3. IP-адреса вашего компьютера и сервера.
4. Код состояния HTTP. Что он означает?
5. Длина тела сообщения. (Содержимое поля заголовка объекта ContentLength указывает длину тела сообщения в октетах (десятичное число), или в случае метода HEAD, размер тела объекта, который мог бы быть послан при запросе GET.)
6. Протокол транспортного уровня, который использует HTTP.

Вариант 2. HTTP: HTTP CONDITIONAL GET/response (Условный GET[3])

1) очистить кэш браузера;

2) открыть браузер;

3) запустить *Wireshark;*

4) запустить процесс захвата трафика;

5) URL: например, http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html

в URL должны присутствовать любые три буквы из фамилии студента в латинской транскрипции;

6) быстро обновить страницу в браузере;

7) остановить захват трафика;

8) настроить фильтр (http).

Прим.: Не принимать во внимание HTTP запрос и ответ для favicon.ico. Появление ссылки на данный файл означает, что браузер автоматически запрашивает сервер о наличии маленького значка веб-сайта, т.н. «Favicon» (отображается браузером в адресной строке перед URL страницы, а также в качестве картинки рядом с закладкой, во вкладках и в других элементах интерфейса).

Ответьте на следующие вопросы.

1. Укажите версию HTTP и принимаемые браузером языки.

2. Укажите IP-адреса вашего компьютера и сервера.

3. Есть ли в первом запросе HTTP GET строка «IF-MODIFIED-SINCE»?

4. Проанализируйте ответ сервера на первый запрос: передал ли сервер явным образом содержимое запрашиваемого ресурса?

5. Теперь просмотрите содержимое второго запроса HTTP GET: есть ли там строка «IF-MODIFIED-SINCE» (какую информацию содержит)?

6. Что означает код состояния в ответе сервера на второй запрос HTTP GET? Передал ли сервер явным образом содержимое запрашиваемого ресурса?

Вариант 3. DNS

1) очистить кэш DNS с помощью *ipconfig* (в командной строке)*:* ipconfig /flushdns

2) очистить кэш браузера;

3) запустить *Wireshark;*

4) настроить фильтр: ip.addr == ваш_IP_адрес;

5) запустить процесс захвата трафика;

6) URL: например, http://www.ietf.org/;

в URL должны присутствовать любые три буквы из фамилии студента в латинской транскрипции;

7) остановить захват трафика.

Ответьте на следующие вопросы.

1. Найдите DNS-запрос и ответ. Поверх какого протокола транспортного уровня работает DNS?

2. Укажите порты источника/назначения для DNS-запроса и DNS-ответа.

3. На какой IP-адрес отправлен DNS-запрос? Совпадает ли этот адрес с адресом вашего DNS-сервера?

4. Укажите тип DNS-запроса.

5. Что содержится в поле «Answers» DNS-ответа?

---

[3] Кроме обычного метода GET, различают ещё условный GET и частичный GET. Условные запросы GET содержат заголовки If-Modified-Since, If-Match, If-Range и подобные. Метод GET изменяется на «условный GET», если сообщение запроса включает в себя поле заголовка «If-Modified-Since». В ответ на условный GET, тело запрашиваемого ресурса передается только, если он изменялся после даты, указанной в заголовке «If-ModifiedSince». Алгоритм определения этого включает в себя следующие случаи:

• Если код статуса ответа на запрос будет отличаться от «200 OK», или дата, указанная в поле заголовка «If-Modified-Since» некорректна, ответ будет идентичен ответу на обычный запрос GET.

• Если после указанной даты ресурс изменялся, ответ будет также идентичен ответу на обычный запрос GET.

• Если ресурс не изменялся после указанной даты, сервер вернет код статуса «304 Not Modified». Использование метода условный GET направлено на разгрузку сети, так как он позволяет не передавать по сети избыточную информацию.

6. Проверьте, какой IP-адрес содержится в последующем пакете TCP SYN, который был отправлен вашим компьютером.

7. Формирует ли ваш компьютер новые DNS-запросы для получения содержащихся на сайте изображений?

Вариант 4. DNS

nslookup — утилита, предоставляющая пользователю интерфейс командной строки для обращения к системе DNS (проще говоря, DNS-клиент). Позволяет задавать различные типы запросов и запрашивать произвольно указываемые сервера.

Использование *nslookup:*

nslookup [-opt ...] # интерактивный режим с использованием сервера по умолчанию
nslookup [-opt ...] – server # интерактивный режим с использованием сервера "server"
nslookup [-opt ...] host # поиск узла "host" с использованием сервера по умолчанию
nslookup [-opt ...] host server # поиск узла "host" с использованием сервера "server"

1) запустить *Wireshark;*
2) настроить фильтр: ip.addr == ваш_IP_адрес;
3) запустить процесс захвата трафика;
4) в командной строке:

nslookup host например, nslookup ifmo.ru

в качестве узла (host) использовать URL, в котором присутствуют любые три буквы из фамилии студента в латинской транскрипции;

5) остановить захват трафика.

Прим.: Всего *nslookup* отправил три DNS-запроса и получил три DNS-ответа. Для дальнейшего анализа использовать последние два пакета. (Первые два набора запросов/ответов не генерируются стандартными интернет-приложениями и специфичны для *nslookup*).

Ответьте на следующие вопросы.

1. Найдите DNS-запрос и ответ. Поверх какого протокола транспортного уровня работает DNS?

2. Укажите порты источника/назначения для DNS-запроса и DNS-ответа.

3. На какой IP-адрес отправлен DNS-запрос? Совпадает ли этот адрес с адресом вашего DNS-сервера?

4. Укажите тип DNS-запроса.

5. Что содержится в поле «Answers» DNS-ответа?

Повторите предыдущий эксперимент, но в командной строке введите команду: nslookup -type=NS host

например, nslookup -type=NS ifmo.ru

в качестве узла (host) использовать URL, в котором присутствуют любые три буквы из фамилии студента в латинской транскрипции;

Ответьте на следующие вопросы.

1. Укажите порты источника/назначения для DNS-запроса и DNS-ответа.

2. На какой IP-адрес отправлен DNS-запрос? Совпадает ли этот адрес с адресом вашего DNS-сервера?

3. Укажите тип DNS-запроса.

4. Проанализируйте DNS-ответ: укажите имена серверов, возвращающих авторитативный[4] отклик.

Вариант 5. ICMP

1) запустить *Wireshark;*
2) настроить фильтр (icmp);
3) запустить процесс захвата трафика;
4) в командной строке:

[4] Под авторитативным (autorative) сервером зоны понимается такой DNS сервер, который официально поддерживает описание зоны. При обращении к такому серверу с запросом по поводу информации о поддерживаемой им (сервером) официально зоне клиент (resolver) получает авторитативный отклик.

ping –n 10 конечный_узел

например, ping –n 10 wireshark.org

в качестве конечного узла использовать URL, в котором присутствуют любые три буквы из фамилии студента в латинской транскрипции;

5) остановить захват трафика.

Ответьте на следующие вопросы.

1. Сколько всего пакетов захватила программа? Почему?
2. Какой IP-адрес вашего компьютера, адрес назначения?
3. Проанализируйте ping request, отправленный с вашего компьютера:

укажите тип и код ICMP. Какие еще поля содержит ICMP пакет? Сколько байт занимают поля «Checksum», «Sequence number», «Identifier»?

4. Проанализируйте ping reply: укажите тип и код ICMP. Какие еще поля содержит ICMP пакет? Сколько байт занимают поля «Checksum», «Sequence number», «Identifier»?

6) запустить *Wireshark;*

7) настроить фильтр (icmp);

8) запустить процесс захвата трафика;

9) в командной строке:

tracert конечный_узел

например, tracert wireshark.org

в качестве конечного узла использовать URL, в котором присутствуют любые три буквы из фамилии студента в латинской транскрипции;

10) остановить захват трафика.

Ответьте на следующие вопросы.

1. Какой IP-адрес вашего компьютера, адрес назначения?
2. Проанализируйте пакет ICMP echo: отличаются ли эти пакеты от пакетов в первой части эксперимента? Чем?
3. Проанализируйте пакет ICMP error: какие поля в нем содержатся?
4. Чем отличаются пакеты ICMP reply (полученные) и ICMP error?

Вариант 6. DHCP

1) в командной строке:

ipconfig /release (IP-адрес станет 0.0.0.0)

2) запустить *Wireshark;*

3) настроить фильтр (bootp);

4) запустить процесс захвата трафика;

5) в командной строке:

ipconfig /renew (получение нового IP-адреса) и еще раз:

ipconfig /release

ipconfig /renew

6) остановить захват трафика.

Ответьте на следующие вопросы.

1. Поверх какого протокола транспортного уровня работает DHCP?
2. Нарисуйте временную диаграмму, иллюстрирующую последовательность обмена первыми четырьмя пакетами Discover/Offer/Request/ACK. Укажите для каждого пакета номера портов источника, назначения.
3. Какими значениями отличаются пакеты DHCP Discover и DHCP Request?
4. Укажите значения поля «Transaction-ID» для всех пакетов (Discover/Offer/Request/ACK), что это поле означает?
5. Укажите IP-адреса источника, назначения для всех пакетов.
6. Укажите IP-адрес DHCP сервера.
7. Поясните назначение сообщения DHCP release.
8. Очистите фильтр. Появились ли пакеты ARP, отправленные или полученные в течение обмена DHCP пакетами? Почему?

Вариант 7. Ethernet и ARP

Ethernet

1) очистить кэш браузера;

2) запустить *Wireshark;*

3) запустить процесс захвата трафика;

4) URL: например, http://ru.wikipedia.org/wiki/Ethernet

в URL должны присутствовать любые три буквы из фамилии студента в латинской транскрипции;

5) остановить захват трафика;

6) в меню «Analyze» → «Enabled Protocols» можно снять галочку IP: тогда в списке пакетов не будет отображаться информация по протоколам верхнего уровня (после IP) — необязательный пункт.

Выберите кадр Ethernet, содержащий сообщение HTTP GET.

Ответьте на следующие вопросы.

1. Укажите 48-битный Ethernet адрес вашего компьютера.

2. Укажите 48-битный Ethernet адрес назначения. Что это за адрес? (Адрес сервера?)

3. Укажите 16-чное значение двухбайтового поля «Type»: какому протоколу верхнего уровня оно соответствует?

Выберите кадр Ethernet, содержащий ответ HTTP.

Ответьте на следующие вопросы.

1. Укажите значение Ethernet адреса источника. Какое устройство имеет такой адрес?

2. Укажите Ethernet адрес назначения: это адрес вашего компьютера?

3. Укажите 16-чное значение двухбайтового поля «Type»: какому протоколу верхнего уровня оно соответствует?

ARP

1) очистить ARP кэш:

«Пуск» → «Выполнить»: netsh interface ip delete arpcache

Вывести на экран ARP-таблицу можно с помощью команды: arp –a

2) очистить кэш браузера;

3) запустить *Wireshark;*

4) запустить процесс захвата трафика;

5) URL: например, http://ru.wikipedia.org/wiki/Ethernet в URL должны присутствовать любые три буквы из фамилии студента в латинской транскрипции;

6) остановить захват трафика;

7) в меню «Analyze» → «Enabled Protocols» снять галочку IP: в списке пакетов теперь не будет отображаться информация по протоколам верхнего уровня (после IP) — необязательный пункт.

Ответьте на следующие вопросы.

1. Укажите 16-чные значения адресов источника и назначения в пакете, содержащем ARP запрос (ARP ответ).

2. Укажите 16-чное значение двухбайтового поля «Type»: какому протоколу верхнего уровня оно соответствует (для ARP запроса/ARP ответа)?

3. Укажите значение поля «opcode» (для ARP запроса/ARP ответа).

4. Содержит ли ARP запрос IP-адрес источника (для ARP запроса/ARP ответа)?

Вариант 8. FTP

1) запустить *Wireshark;*

2) запустить процесс захвата трафика;

3) скачать файл с FTP-сервера (например, ftp://ftp.canon.ru/);

в URL должны присутствовать любые три буквы из фамилии студента в латинской транскрипции;

4) остановить захват трафика;

5) настроить фильтр (ftp || ftp-data).

Ответьте на следующие вопросы.

1. Сколько байт данных содержится в пакете FTP-DATA?

2. Укажите IP-адреса FTP-сервера и вашего компьютера.
3. Укажите MAC-адрес FTP-сервера.
4. Укажите протокол транспортного уровня, который использует протокол FTP.
5. Укажите порт, который используется при передаче данных по протоколу FTP.
6. Поясните, чем отличаются пакеты FTP и FTP-DATA.

Вариант 9. UDP

1) запустить *Wireshark;*
2) настроить фильтр (udp);
3) запустить процесс захвата трафика;
4) создать сеанс связи с помощью программы *TeamViewer*;
5) остановить захват трафика.

Прим.: можно ничего не делать, просто запустить захват трафика — UDP пакеты все равно найдутся.

Ответьте на следующие вопросы.

1. Выберите один UDP пакет из списка пакетов. Сколько полей в UDP заголовке? Что это за поля?
2. Какова длина (в байтах) каждого поля заголовка?
3. Длина чего указана в поле «Length»?
4. Какова максимальная длина поля данных UDP?
5. Какой максимально возможный номер порта источника?

ЗАКЛЮЧИТЕЛЬНАЯ ЧАСТЬ

Предъявить результаты работы в виде файла на экране монитора преподавателю. Быть готовым показать текст этого файла. Оформить отчет по работе и быть готовым ответить на контрольные вопросы.

Подводятся итоги занятия по степени достижения поставленных целей занятия, активности обучаемых. При возникновении вопросов дается краткий ответ или же назначается время консультации. Объявляются оценки за занятие. Объявляется тема следующего занятия. Выдается задание на самостоятельную работу. Оформляется журнал учебной группы.

Задание на самостоятельную работу:

1. Изучить и доработать вопросы занятия по рекомендованной литературе.
2. Темой следующей лабораторной работы №3 будет «Исследование протоколов TELNET и HTTP».

Преподаватель кафедры С-20
кандидат технических наук

Р. Татарников