УНИВЕРСИТЕТ ИТМО

Т.И. Алиев, В.В. Соснин, Д.Н. Шинкарук

КОМПЬЮТЕРНЫЕ СЕТИ И ТЕЛЕКОММУНИКАЦИИ: ЗАДАНИЯ И ТЕСТЫ

Санкт-Петербург
2018

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
УНИВЕРСИТЕТ ИТМО

Т.И. Алиев, В.В. Соснин, Д.Н. Шинкарук

КОМПЬЮТЕРНЫЕ СЕТИ И ТЕЛЕКОММУНИКАЦИИ: ЗАДАНИЯ И ТЕСТЫ

Рекомендовано к использованию в Университете ИТМО по направлениям подготовки бакалавров “Информатика и вычислительная техника” и “Программная инженерия” в качестве учебно-методического пособия для реализации основных профессиональных образовательных программ высшего образования.

Санкт-Петербург
2018

Алиев Т.И., Соснин В.В., Шинкарук Д.Н. Компьютерные сети и телекоммуникации: задания и тесты. – СПб: Университет ИТМО, 2018. – 112 с.

Рецензент: Арустамов С.А., д.т.н., профессор кафедры проектирования и безопасности компьютерных систем Университета ИТМО.

Пособие содержит описание учебно-исследовательских работ по дисциплине “Сети ЭВМ и телекоммуникации”, выполняемых в рамках лабораторных и практических занятий. По каждой работе приводятся краткие теоретические сведения, описываются этапы и порядок выполнения работы, перечислены требования к оформлению отчета и представлен краткий перечень контрольных вопросов.

Учебно-методическое пособие предназначено для студентов, обучающихся по дисциплине “Сети ЭВМ и телекоммуникации” бакалаврских программ по направлениям подготовки “Информатика и вычислительная техника” и “Программная инженерия”, и может быть полезным при изучении других дисциплин, связанных с сетевыми и телекоммуникационными технологиями в рамках бакалаврских и магистерских программ других направлений подготовки.

Рекомендовано к печати учёным советом факультета программной инженерии и компьютерной техники 23 января 2018 года (протокол № 1).

УНИВЕРСИТЕТ ИТМО

Университет ИТМО – ведущий вуз России в области информационных и фотонных технологий, один из немногих российских вузов, получивших в 2009 году статус национального исследовательского университета. С 2013 года Университет ИТМО – участник программы повышения конкурентоспособности российских университетов среди ведущих мировых научно-образовательных центров, известной как проект «5 в 100». Цель Университета ИТМО – становление исследовательского университета мирового уровня, предпринимательского по типу, ориентированного на интернационализацию всех направлений деятельности.

© Университет ИТМО, 2018

СОДЕРЖАНИЕ

Введение

Дисциплина “Сети ЭВМ и телекоммуникации”, изучаемая на кафедре вычислительной техники Университета ИТМО в рамках бакалаврской подготовки по направлениям “Информатика и вычислительная техника” и “Программная инженерия”, является введением в технологии компьютерных и телекоммуникационных сетей и направлена на изучение общих принципов их структурной и функциональной организации на примере локальных сетей Ethernet и глобальных сетей, построенных на основе наиболее распространенного стека протоколов TCP/IP. Теоретический материал, излагаемый на лекциях, поддерживается практическими работами, выполняемыми в виде учебно-исследовательских работ (УИР) в рамках лабораторных занятий и домашних заданий.

В данном учебно-методическом пособии представлены описания шести заданий на выполнение УИР, содержащие краткие теоретические сведения, описание этапов и порядка выполнения работ, а также требования к содержанию отчетов, которые могут быть представлены в электронном или в печатном виде. Теоретические сведения, предваряющие описание каждого задания, содержат минимум информации, необходимой для выполнения каждой конкретной УИР, при этом предполагается, что более полную информацию можно получить в учебных пособиях [1, 2, 5].

Для подготовки к защите отчета по выполненной работе по каждому заданию представлен примерный перечень контрольных вопросов.

Раздел 1. Задания

Задание 1. Кодирование данных в телекоммуникационных сетях

1.1. Цель и краткая характеристика работы

Цель работы: изучение методов физического и логического кодирования, используемых в цифровых сетях передачи данных.

В процессе выполнения учебно-исследовательской работы (УИР) необходимо:

- выполнить физическое и логическое кодирование исходного сообщения в соответствии с заданными методами кодирования;
- провести сравнительный анализ рассмотренных методов кодирования и сформулировать достоинства и недостатки;
- рассчитать частотные характеристики сигналов, используемых для передачи исходного сообщения, и требуемую полосу пропускания канала связи;
- выбрать и обосновать наилучший метод для передачи исходного сообщения.

Ориентировочная трудоемкость выполнения задания для:

- 2-х методов кодирования – 4 часа;
- 3-х методов кодирования – 5 часов;
- 4-х методов кодирования – 6 часов.

1.2. Теоретические сведения

1.2.1. Цифровое кодирование

Цифровое кодирование дискретных данных осуществляется с использованием потенциальных или импульсных кодов. Для представления двоичных нулей и единиц в потенциальных кодах используются разные значения потенциала сигнала, а в импульсных кодах – импульсы разной полярности или перепады потенциала.

Качество передачи данных, а именно: надежность и достоверность доставки, возможность обнаружения и исправления возникающих ошибок, стоимость реализации, – существенно зависит от выбранного метода цифрового кодирования, который, в свою очередь, в значительной мере определяет пропускную способность среды передачи.

В связи с этим, для обеспечения качества передачи данных к методам цифрового кодирования предъявляется ряд требований:

- уменьшение спектра сигнала при одной и той же битовой скорости;

- поддержка синхронизации между передатчиком и приёмником сигналов за счёт наличия в передаваемых сигналах признаков, на основе которых реализуется самосинхронизация;
- отсутствие постоянной составляющей в сигнале, сдвигающей спектр сигнала в область низких частот;
- возможность обнаружения ошибок и их исправления;
- низкая стоимость реализации метода кодирования, зависящая от количества уровней сигнала.

Минимизация спектра результирующего сигнала обеспечивает при заданной полосе пропускания канала связи передавать б*о*льший объём данных за единицу времени. Это может быть реализовано, например, за счёт использования частотного мультиплексирования путем организации нескольких логических каналов в одной и той же линии связи, что и позволяет увеличить скорость передачи данных.

Кроме того, в спектре сигнала должна отсутствовать постоянная составляющая, то есть отсутствовать постоянный ток между передатчиком и приемником. Это обусловлено применением в электрических линиях связи трансформаторных схем для *гальванической развязки,* препятствующей прохождению постоянного тока.

Спектр результирующего сигнала зависит от:

- метода кодирования и модуляции;
- скорости модуляции, влияющей на скорость передачи данных;
- состава передаваемых данных.

Для синхронизации передатчика и приёмника сигналов с целью определения момента считывания в приёмнике значения очередного битового интервала применяются специальные *самосинхронизирующиеся методы кодирования.* В этих методах синхронизация приемника с передатчиком выполняется на основе признака, в качестве которого служит любой резкий перепад сигнала, называемый фронтом сигнала.

Требование отсутствия постоянной составляющей в сигнале обусловлено необходимостью поддержки синхронизации приёмника с передатчиком. Кроме того желательно, чтобы нижняя частота передаваемого сигнала отличалась от нуля. Это позволяет уменьшить спектр сигнала, а также не препятствует прохождению постоянного тока в электрических линиях связи при наличии трансформаторных схем гальванической развязки.

Желательным, но необязательным требованием, предъявляемым к методам цифрового кодирования, является возможность обнаружения ошибок и, в идеале, их исправления. Это позволяет сэкономить время, поскольку ошибка обнаруживается на физическом уровне. При этом ошибочный кадр отбрасывается до завершения полного приёма в буфер.

Стоимость реализации метода цифрового кодирования связана с количеством уровней сигнала, причем чем больше уровней сигнала, тем более мощное требуется приёмно-передающее оборудование и, следовательно, более дорогого.

Предъявляемые к методам цифрового кодирования требования являются противоречивыми. При этом каждый из методов цифрового кодирования по сравнению с другими обладает своими конкретными достоинствами и недостатками, которые рассматриваются ниже.

1.2.2. Методы физического кодирования

На рисунке 1.1 представлены различные методы кодирования 19-разрядного двоичного сообщения 0101010000111101100 и рассмотрены основные достоинства и недостатки каждого из методов.

На примере метода потенциального кодирования NRZ проиллюстрирован подход, позволяющий приближенно оценить основные частотные характеристики сигнала, формируемого при кодировании сообщения, в качестве которых рассматриваются:

- верхняя и нижняя границы частот в передаваемом сообщении (спектр сигнала);
- среднее значение частоты в спектре передаваемого сигнала;
- полоса пропускания, необходимая для качественной передачи данного сообщения.

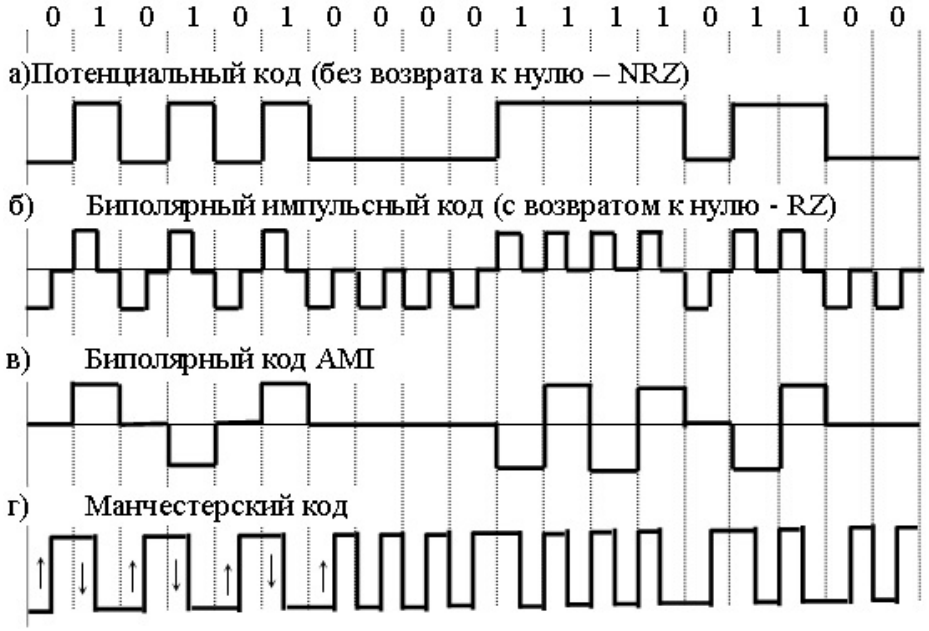


Рисунок 1.1. Методы кодирования дискретных данных

1.2.2.1. Потенциальный код без возврата к нулю (NRZ)

Наиболее простым и очевидным методом кодирования двоичных сообщений является метод потенциального кодирования *без возврата к нулю* – NRZ (Non Return to Zero), в котором значению бита «1» соответствует высокий уровень потенциала, а значению «0» – низкий (рисунок 1.1,а).

Для определения **верхней границы частот** необходимо найти наиболее высокочастотную составляющую спектра в передаваемом сообщении. В коде NRZ высокочастотная составляющая образуется при передаче чередующихся значений 0 и 1, при этом период синусоиды (гармонического сигнала), используемой для передачи прямоугольных сигналов 0 и 1, будет равен удвоенной длительности битового интервала t: $T = 2t$, где t определяется как величина, обратная значению скорости передачи данных (пропускной способности канала): $t = 1/C$. Отсюда верхняя граница частот будет равна $f_{\text{в}} = 1/T = C/2$. При пропускной способности канала связи $C = 1\,\text{Мбит/с}$ частота основной гармоники равна $f_{\text{в}} = 500$ кГц.

В общем случае, при кодировании любого сообщения по методу NRZ наибольшая (верхняя) частота достигается при передаче чередующихся значений 0 и 1, а наименьшая (нижняя) – при передаче длинных (в пределе бесконечных) последовательностей нулей и единиц, что делает нижнюю границу частот близкой и в пределе равной нулю: $f_{\text{н}} = 0$. Следовательно, в предельном случае **ширина спектра** $S = f_{\text{в}} - f_{\text{н}} = f_{\text{в}} = C/2$.

С другой стороны, при передаче конкретного сообщения **нижняя частота** всегда больше нуля и зависит от максимальной длины последовательностей нулей или единиц. В этом случае для расчета нижней границы частот необходимо в коде передаваемого сообщения найти наиболее длинную последовательность единиц или нулей. В представленном на рисунке 1.1,а сообщении, закодированном по методу NRZ, низкочастотная составляющая образуется при передаче четырёх последовательных единиц и четырёх последовательных нулей. Период синусоидального сигнала при передаче таких последовательностей равен 8 битовым интервалам и нижняя граница частот соответственно будет равна: $f_{\text{н}} = 1/8\,t = C/8$. Тогда ширина спектра при передаче данного сообщения кодом NRZ равна $S = f_{\text{в}} - f_{\text{н}} = 0{,}375C = 375$ кГц

Отметим, что полученные значения нижней границы частот и, соответственно, спектра справедливы именно для этого конкретного сообщения. При передаче других сообщений эти значения будут другими. Таким образом, можно утверждать, что при кодировании по методу NRZ ширина спектра сигнала $S < C/2$.

Среднее значение частоты передаваемого сообщения находится в интервале $(f_{\text{н}}; f_{\text{в}})$ и показывает, какие частоты (низкие или высокие) превалируют в спектре передаваемого сигнала.

Для расчёта среднего значения частоты передаваемого сообщения необходимо для каждого битового интервала определить соответствующую частоту сигнала, просуммировать их и разделить на количество битовых интервалов. В нашем случае: частота основной гармоники $f_0 = C/2$ соответствует 7-ми битовым интервалам, 4-м битовым интервалам соответствует частота вдвое меньшая, чем частота основной гармоники, т.е. $f_0/2$, и 8-ми битовым интервалам соответствует частота $f_0/4$.

Тогда средняя частота рассматриваемого сообщения равна:

$$f_{ср} = (7f_0 + 4f_0/2 + 8f_0/2)/19 \approx 0{,}58f_0 = 290\,кГц.$$

Поскольку середине спектра рассматриваемого сообщения соответствует частота $f_{1/2} = (f_н + f_в)/2 = 0б625f_0 = 312{,}5\,кГц$, можно констатировать, что в спектре сигнала незначительно превалируют низкие частоты: $f_{ср} < f_{1/2}$.

Для качественной передачи двоичных сигналов по реальному каналу связи и возможности их распознавания на приёмной стороне с минимальным количеством ошибок, желательно на передающей стороне формировать сигналы, приближающиеся к прямоугольной форме. Однако, спектр таких сигналов оказывается слишком большим. В то же время, для качественного распознавания сигнала на приемной стороне при передаче чередующихся значений 0 и 1 достаточно сформировать сигнал, содержащий первые 4 гармоники (поскольку более высокочастотные гармоники оказывают незначительное влияние на результирующий сигнал) с частотами $f_0 = C/2$, $f_1 = 3f_0$, $f_2 = 5f_0$, $f_3 = 7f_0$. В этом случае верхняя граница частот $f_в = 7f_0$, а ширина спектра сигнала при передаче рассматриваемого сообщения соответственно будет равна $S = f_в - f_н = 7f_0 - f_0/4 = 6{,}75f_0 = 3{,}375\,МГц$.

Полоса пропускания F, необходимая для передачи данного сообщения, должна быть больше спектра S, например, $F = 4$ МГц.

Рассмотрим теперь достоинства и недостатки метода кодирования NRZ.

Достоинствами кода NRZ являются:

- простота и низкая стоимость, обусловленная наличием только двух уровней потенциала;
- малая ширина спектра сигнала, которая меньше, чем у других методов кодирования: $S = f_в = 0{,}5C$ Гц, где C – скорость передачи данных [бит/с].

В компьютерных сетях код NRZ в чистом виде не используется ввиду наличия следующих недостатков:

- отсутствие самосинхронизации, что может привести к рассинхронизации часов приёмника и передатчика при передаче длинной последовательности единиц или нулей;

- невозможность использования в электрических каналах связи при наличии гальванических развязок между приёмником и источником.

Тем не менее, используются модификации код NRZ, в которых устраняют постоянную составляющую за счёт применения методов логического кодирования, в частности, избыточного кодирования.

1.2.2.2. Биполярный импульсный код (RZ)

В импульсных кодах данные представлены полным импульсом или же его частью – фронтом. Одним из наиболее простых среди импульсных кодов является трехуровневый *биполярный импульсный код с возвратом к нулю* (Return to Zero, RZ), в котором единица представлена импульсом одной полярности, а ноль – импульсом другой полярности (рисунок 1.1,б). Каждый импульс длится половину битового интервала. В середине битового интервала происходит возврат к нулевому потенциалу.

К достоинствам кода RZ относятся:

- наличие самосинхронизации: признаком (стробом) для синхронизации часов приёмника служит возврат в середине каждого битового интервала к нулевому потенциалу
- отсутствие постоянной составляющей.

В то же время метод RZ обладает следующими недостатками:

- наличие трёх уровней сигнала требует увеличения мощности передатчика для обеспечения достоверности приёма сигналов, что увеличивает стоимость реализации;
- спектр сигнала шире, чем у потенциальных кодов: при передаче последовательности нулей или единиц верхняя граница частот будет равна $f_{\text{в}} = C$ Гц, а нижняя граница при передаче чередующихся нулей и единиц будет равна $f_{\text{н}} = C/4$, что увеличивает спектр сигнала в полтора раза по сравнению с кодом NRZ: $S = f_{\text{в}} - f_{\text{н}} = 0{,}75C$.

Из-за указанных недостатков биполярный импульсный код “в чистом виде” используется редко.

1.2.2.3. Биполярное кодирование с чередующейся инверсией (AMI)

Биполярное кодирование с альтернативной инверсией (Bipolar Alternate Mark Inversion, AMI) является модификаций метода RZ. В AMI также используются три уровня потенциала: положительный, нулевой и отрицательный (рисунок 1.1,в). Двоичный «0» кодируется нулевым потенциалом, а двоичная «1» – либо положительным, либо отрицательным потенциалом, при этом всегда потенциал следующей единицы противоположен потенциалу предыдущей.

В качестве основных достоинств метода AMI можно отметить:

- отсутствие проблемы постоянной составляющей и возможность синхронизации приёмника с передатчиком при передаче длинных последовательностей единиц, так как в этом случае сигнал представляет собой последовательность разнополярных импульсов;
- в общем случае спектр сигнала при кодировании AMI меньше, чем при RZ, что обеспечивает б*о*льшую пропускную способность канала связи, в частности, при передаче чередующихся единиц и нулей верхняя граница частот, как и при передаче чередующихся нулей и единиц кода NRZ , равна $f_{\text{в}} = C/2$ Гц, а ширина спектра сигнала $S < C/2$;
- возможность распознавать ошибочные (запрещённые) сигналы при нарушении чередования полярности сигналов в процессе передачи единиц, когда после единичного сигнала появляется единичный сигнал той же полярности.

К недостаткам метода AMI относятся:

- наличие трёх уровней сигнала требует увеличения мощности передатчика, что, естественно, увеличивает стоимость;
- в случае длинных последовательностей нулей в сигнале присутствует постоянная составляющая, сдвигающая спектр в низкочастотный диапазон.

1.2.2.4. Потенциальный код с инверсией при единице (NRZI)

Потенциальный код с инверсией при единице (Non Return to Zero with ones Inverted, NRZI) в отличие от AMI имеет только два уровня сигнала: при передаче двоичного нуля сохраняется уровень, который был установлен в предыдущем такте, а при передаче единицы – уровень сигнала меняется на противоположный.

Достоинство: наличие двух уровней сигнала уменьшает стоимость реализации по сравнению с трехуровневым кодом AMI.

1.2.2.5. Манчестерский код

Манчестерский код (рисунок 1.1,г) нашел широкое применение в локальных сетях Ethernet. Для кодирования используются два уровня сигнала, при этом для представления двоичных единиц и нулей используется переход сигнала в середине каждого битового интервала:

- двоичной «1» соответствует переход от высокого уровня сигнала к низкому;
- двоичному «0» – переходом от низкого уровня сигнала к высокому.

В случае последовательности из нескольких единиц или нулей в начале каждого битового интервала происходит дополнительный служебный переход сигнала.

К достоинствам манчестерского кода следует отнести:

- самосинхронизация: сигналом для синхронизации приёмника с передатчиком может служить изменение сигнала в середине каждого битового интервала;
- меньший спектр по сравнению с биполярным импульсным кодом в среднем в 1,5 раза: верхняя граница частот при передаче последовательности единиц или нулей равна $f_{в} = C$ Гц, а нижняя граница при передаче чередующихся единиц и нулей $f_{н} = C/2$ Гц, тогда спектр $S = f_{в} - f_{н} = 0{,}5C$;
- наличие только двух уровней потенциала;
- отсутствие постоянной составляющей.

Недостатком манчестерского кода является более широкий спектр сигнала по сравнению с кодами NRZ и AMI.

1.2.2.6. Дифференциальный манчестерский код

Дифференциальный или *разностный манчестерский код* применяется в сетях Token Ring и является разновидностью манчестерского кода, в котором:

- «0» кодируется изменением потенциала в начале битового интервала (а не в середине);
- «1» – сохранением предыдущего уровня потенциала.

В середине каждого битового интервала обязательно присутствует переход с одного уровня потенциала на другой. Помимо "0" и "1" также могут передаваться так называемые запрещённые символы “J” и “K”, в которых в середине битового интервала отсутствует изменение уровня потенциала. “J” и “K” применяются в качестве начального и конечного разделителя кадров. Дифференциальный манчестерский код, в отличие от простого манчестерского кода (см. п. 1.2.2.5), позволяет обеспечить корректное декодирование сигнала, даже если в результате ошибки весь передаваемый сигнал в канале связи инвертируется (т.е. если низкий потенциал ошибочно заменится на высокий и наоборот).

1.2.2.7. Код трехуровневой передачи MLT-3

В методе кодирования трехуровневой передачи MLT-3 (Multi Level Transmission-3) двоичной «1» соответствует переход на границе битового интервала *последовательно* с одного уровня сигнала на другой, а при передаче нуля сигнал не меняется. При этом максимальная частота сигнала достигается при передаче длинной последовательности единиц, когда

изменение сигнала происходит последовательно с одного уровня на другой с учетом предыдущего перехода.

К недостаткам этого метода кодирования относятся:

- отсутствие самосинхронизации;
- наличие трёх уровней сигнала;
- наличие в сигнале постоянной составляющей при передаче длинной последовательности нулей.

1.2.2.8. Пятиуровневый код PAM-5

В пятиуровневом коде PAM-5 используется 5 уровней сигнала, причем четыре уровня кодируют два бита передаваемых данных: 00, 01, 10, 11. Таким образом, в одном битовом интервале передаются два бита. Пятый (средний) уровень добавлен для создания избыточности кода, используемого для исправления ошибок.

Основное достоинство метода PAM-5 состоит в том, что при одной той же скорости модуляции данные передаются в два раза быстрее по сравнению с AMI или NRZI.

К недостаткам метода относятся:

- наличие постоянной составляющей в сигнале при передаче длинных последовательностей одинаковых пар бит;
- наличие пяти уровней требует большей мощности передатчика, что значительно увеличивает стоимость реализации.

1.2.3. Логическое кодирование

Логическое кодирование используется для улучшения потенциальных кодов AMI, NRZI или MLT-3 за счет ликвидации длинных последовательностей единиц или нулей, приводящих к постоянному потенциалу.

К логическому кодированию относятся *избыточное кодирование* и *скремблирование*.

1.2.3.1. Избыточное кодирование

При избыточном кодировании исходный двоичный код представляется в виде последовательностей нескольких битов, каждая из которых заменяется новой последовательностью, содержащей большее количество бит, чем исходная.

К методам избыточного кодирования относятся: 4B/5B, 5B/6B, 8B/10B, 64B/66B.

Буква «B» в названии кода означает, что элементарный сигнал имеет 2 состояния (binary – двоичный), а цифры указывают, какое количество бит содержится в одной последовательности исходного и результирующего кода соответственно. Например, метод 4B/5B означает, что каждые 4 бита в

исходном коде заменяются 5-ю битами в результирующем коде. Для этого используется таблица перекодировки (таблица 1.1), устанавливающая соответствие между исходными четырёхбитовыми и результирующими пятибитовыми последовательностями.

В результате такой замены количество результирующих кодовых последовательностей больше количества исходных. В коде 4B/5B результирующих последовательностей 2^5=32, в то время как исходных 2^4=16. Следовательно, количество избыточных (запрещённых) кодов: 32–16=16. Появление запрещённых символов означает ошибку в передаваемых данных.

Среди результирующих последовательностей отобраны 16 таких, любое сочетание которых содержит в худшем случае 8 подряд расположенных единиц.

Таблица 1.1.

Исходные символы	*Результирующие символы*	*Исходные символы*	*Результирующие символы*
0000	11110	1000	10010
0001	01001	1001	10011
0010	10100	1010	10110
0011	10101	1011	10111
0100	01010	1100	11010
0101	01011	1101	11011
0110	01110	1110	11100
0111	01111	1111	11101

Можно отметить следующие достоинства избыточного кодирования:

- появляется свойство самосинхронизации, поскольку исчезают длинные последовательности нулей и единиц;
- сужается спектр сигнала в связи с отсутствием постоянной составляющей;
- появляется возможность обнаружения ошибок за счёт наличия запрещённых символов;
- простая реализация в виде таблицы перекодировки.

Недостатки избыточного кодирования:

- уменьшается полезная пропускная способность канала связи, так как часть пропускной способности тратится на передачу избыточных бит;
- возникают дополнительные временные затраты в узлах сети на реализацию логического кодирования.

Основным недостатком избыточного кодирования является появление “лишнего” бита, приходящегося на 4 информационных бита, т.е. избыточность кода 4B/5B составляет 25% (1/4 = 0,25). Это означает, что реальная пропускная способность канала будет меньше номинальной на 20%. Для сохранения заданной пропускной способности необходимо увеличить тактовую частоту передатчика на 25%, что, в свою очередь, приведет к увеличению спектра сигнала.

В методе логического кодирования 8B/6T для кодирования 8 бит (B) исходного сообщения используется код из 6 троичных (T) символов с тремя состояниями сигнала. Количество избыточных (запрещённых) кодов: $3^6 – 2^8 = 729 – 256 = 473$. Таким образом, в 8B/6T доля запрещенных кодов больше, чем в 4B/5B (65% против 50%), что повышает эффективность обнаружения ошибок.

1.2.3.2. Скремблирование

Скремблирование – преобразование исходного двоичного кода по заданному алгоритму, позволяющему исключить или, по крайней мереЮ уменьшить длинные последовательности нулей или единиц.

Например, алгоритм преобразования может иметь вид:

$B_i = A_i \oplus B_{i-3} \oplus B_{i-5}$ (i = 1, 2,…),

где A_i, B_i – значения i-го разряда соответственно исходного и результирующего кода; B_{i-3} и B_{i-5} – значения соответственно (i-3)-го и (i-5)-го разряда результирующего кода; $\oplus$ – операция исключающего ИЛИ (операция сложения по модулю 2).

Для исходной последовательности A=110110000001 использование такого скремблера приведет к следующему результату:

$B_1 = A_1 = 1$;

$B_2 = A_2 = 1$;

$B_3 = A_3 = 0$;

$B_4 = A_4 \oplus B_1 = 1 \oplus 1 = 0$;

$B_5 = A_5 \oplus B_2 = 1 \oplus 1 = 0$;

$B_6 = A_6 \oplus B_3 \oplus B_1 = 0 \oplus 0 \oplus 1 = 1$;

$B_7 = A_7 \oplus B_4 \oplus B_2 = 0 \oplus 0 \oplus 1 = 1$;

$B_8 = A_8 \oplus B_5 \oplus B_3 = 0 \oplus 0 \oplus 0 = 0$;

$B_9 = A_9 \oplus B_6 \oplus B_4 = 0 \oplus 1 \oplus 0 = 1$;

$B_{10} = A_{10} \oplus B_7 \oplus B_5 = 0 \oplus 1 \oplus 0 = 1$;

$B_{11} = A_{11} \oplus B_8 \oplus B_6 = 0 \oplus 0 \oplus 1 = 1$;

$B_{12} = A_{12} \oplus B_9 \oplus B_7 = 1 \oplus 1 \oplus 1 = 1$.

Таким образом, на выходе скремблера появится результирующий код B=110001101111, в котором отсутствует последовательность из шести нулей, присутствовавшая в исходном коде.

Дескремблер восстанавливает исходный код с использованием обратного соотношения:

$C_i = B_i \oplus B_{i-3} \oplus B_{i-5}$ ($i = 1, 2,\ldots$).

Легко убедиться, что восстановленный код совпадает с исходным, т.е. $C_i = A_i$.

Различные алгоритмы скремблирования могут отличаться количеством слагаемых для определения значения разряда результирующего кода и величиной сдвига между слагаемыми, например, величина сдвига может составлять 5 и 23 позиции или иметь любые другие значения.

Основным достоинством скремблирования по сравнению с избыточным кодированием является сохранение полезной пропускной способности канала связи, поскольку отсутствуют избыточные биты.

Недостатками скремблирования следует считать:

- наличие дополнительных затрат (накладных расходов) в узлах сети на реализацию алгоритма скремблирования-дескремблирования;
- отсутствие 100-процентной гарантии исключения длинных последовательности нулей и единиц, а также возможность появления других (новых) последовательности нулей и единиц в результирующем коде.

1.3. Этапы выполнения работы и варианты заданий

Этап 1. Формирование сообщения

В качестве исходного сообщения, подлежащего передаче, используются фамилия и инициалы студента, выполняющего задание. Для цифрового представления сообщения используются шестнадцатеричные коды в соответствии с кодировочной таблицей (см. таблицу 1.2).

Записать исходное сообщение в шестнадцатеричном и двоичном кодах. Определить длину сообщения.

Таблица 1.2.

Сим-вол	**Код**	**Сим-вол**	**Код**	**Сим-вол**	**Код**	**Сим-вол**	**Код**	**Сим-вол**	**Код**
А	C0	Р	D0	а	E0	р	F0	пробел	20
Б	C1	С	D1	б	E1	с	F1	,	2C
В	C2	Т	D2	в	E2	т	F2	.	2E
Г	C3	У	D3	г	E3	у	F3	0	30
Д	C4	Ф	D4	д	E4	ф	F4	1	31
Е	C5	Х	D5	е	E5	х	F5	2	32
Ж	C6	Ц	D6	ж	E6	ц	F6	3	33
З	C7	Ч	D7	з	E7	ч	F7	4	34
И	C8	Ш	D8	и	E8	ш	F8	5	35
Й	C9	Щ	D9	й	E9	щ	F9	6	36
К	CA	Ъ	DA	к	EA	ъ	FA	7	37
Л	CB	Ы	DB	л	EB	ы	FB	8	38
М	CC	Ь	DC	м	EC	ь	FC	9	39
Н	CD	Э	DD	н	ED	э	FD		
О	CE	Ю	DE	о	EE	ю	FE		
П	CF	Я	DF	п	EF	я	FF		

Пример:

исходное сообщение: Ф.И.О.

в шестнадцатеричном коде: D4 2E C8 2E CE 2E

в двоичном коде: 11010100 00101110 11001000 00101110 11001110 00101110

длина сообщения: 6 байт (48 бит)

Этап 2. Физическое кодирование исходного сообщения

Выполнить физическое кодирование исходного сообщения с использованием манчестерского кодирования и ещё двух (на оценку «удовлетворительно»), трёх (на оценку «хорошо») или четырёх (на оценку «отлично») разных способов кодирования, наиболее приемлемых для передачи данного сообщения.

Результаты кодирования для первых четырех байт изобразить в виде временны*х* диаграмм.

Для каждого способа кодирования определить (полагая, что пропускная способность канала связи равна 1 Гбит/с):

- верхнюю и нижнюю границы частот в передаваемом сообщении (спектр сигнала);
- среднее значение частоты в спектре передаваемого сигнала;
- полосу пропускания, необходимую для качественной передачи данного сообщения.

Провести сравнительный анализ рассмотренных способов кодирования (определить достоинства и недостатки).

Выбрать два наилучших способа кодирования для передачи исходного сообщения и обосновать этот выбор.

Этап 3. Логическое (избыточное) кодирование исходного сообщения

Выполнить логическое кодирование исходного сообщения по методу 4B/5B. Записать полученное сообщение в двоичном и шестнадцатеричном кодах.

Определить длину нового сообщения и его избыточность.

Пример:

в двоичном коде: 1101 1010 1010 1001 1100 1101 0100 1010
1001 1100 1101 0111 0010 1001 1100

в шестнадцатеричном коде: DAA9CD4A9CD729C

длина сообщения: 7,5 байт (60 бит)

избыточность: 1,5/6=12/48=0,25 (25%)

Для полученного нового сообщения выполнить физическое кодирование с использованием двух способов кодирования, выбранных в качестве наилучших на втором этапе.

Результаты кодирования для первых четырёх байт изобразить в виде временны*х* диаграмм.

Для каждого способа кодирования определить (полагая, что пропускная способность канала связи равна 1 Гбит/с):

- верхнюю и нижнюю границы частот в передаваемом сообщении (спектр сигнала);
- среднее значение частоты в спектре передаваемого сигнала;
- полосу пропускания, необходимую для качественной передачи данного сообщения.

Выбрать наилучший способ физического кодирования для передачи нового избыточного сообщения и обосновать этот выбор.

Этап 4. Скремблирование исходного сообщения

Выбрать из представленных ниже полиномов или предложить другой полином для скремблирования исходного сообщения и обосновать этот выбор.

$B_i = A_i \oplus B_{i-3} \oplus B_{i-5}$;

$B_i = A_i \oplus B_{i-5} \oplus B_{i-7}$.

Выполнить скремблирование исходного сообщения.

Записать полученные скремблированные сообщения в двоичном и шестнадцатеричном кодах.

Для полученного нового скремблированного сообщения выполнить физическое кодирование с использованием двух способов кодирования, выбранных на втором этапе.

Результаты кодирования для первых четырех байт изобразить в виде временных диаграмм.

Для каждого способа кодирования определить (полагая, что пропускная способность канала связи равна 1 Гбит/с):

- верхнюю и нижнюю границы частот в передаваемом сообщении (спектр сигнала);
- среднее значение частоты в спектре передаваемого сигнала;
- полосу пропускания, необходимую для качественной передачи данного сообщения.

Выбрать наилучший способ физического кодирования для передачи скремблированного сообщения и обосновать этот выбор.

Этап 5. Сравнительный анализ результатов кодирования

Выполнить сравнительный анализ результатов, полученных на этапах 2, 3 и 4. Результаты сравнения представить в виде сводной таблицы.

1.4. Порядок выполнения работы

1. Ознакомиться с постановкой задачи и изучить необходимые теоретические сведения.

2. Сформировать исходное сообщение в соответствии с этапом 1.

3. Выполнить физическое кодирование исходного сообщения не менее, чем тремя способами, включая, в качестве обязательного, манчестерское кодирование. Рассчитать частотные характеристики передаваемого сигнала для рассматриваемых способов кодирования и определить требуемую для эффективной передачи сообщения пропускную способность канала связи (этап 2).

4. Выполнить логическое кодирование исходного сообщения, используя избыточное кодирование 4B/5B и скремблирование. Рассчитать частотные характеристики передаваемого сигнала для рассматриваемых способов кодирования и определить требуемую для эффективной передачи сообщения пропускную способность канала связи (этапы 3 и 4).

5. Выполнить сравнительный анализ рассмотренных способов кодирования и выбрать наилучший способ для передачи исходного сообщения (этап 5).

6. Оформить отчёт и сдать его на проверку.

7. В назначенное преподавателем время защитить задание.

1.5. Требования к содержанию отчёта

Отчёт может быть представлен в электронном или бумажном виде и должен содержать следующие пункты.

1. Краткая постановка задачи.

2. Исходное сообщение и его представление в шестнадцатеричном и двоичном виде, длина исходного сообщения (в байтах и битах).

3. Временн**ы**е диаграммы для рассмотренных способов физического кодирования (включая манчестерское кодирование) первых четырёх байт исходного сообщения.

Рассчитанные для каждого способа кодирования:

- верхняя и нижняя границы частот в передаваемом сообщении (спектр сигнала);
- среднее значение частоты в спектре передаваемого сигнала;
- полоса пропускания, необходимая для качественной передачи данного сообщения.

4. *Результаты сравнительного анализа* рассмотренных способов кодирования (достоинства и недостатки), представленные в виде таблицы, и обоснованный выбор двух лучших способов кодирования для передачи исходного сообщения.

5. *Результат логического кодирования* исходного сообщения *по методу 4B/5B*, записанный в виде избыточного сообщения в *двоичном* и *шестнадцатеричном* кодах.

Значение *длины* нового сообщения и его *избыточность*.

6. *Временны**е** диаграммы* для двух способов физического кодирования (включая манчестерское кодирование) избыточного сообщения, а также рассчитанные для всех способов кодирования:

- верхняя и нижняя границы частот в передаваемом сообщении (спектр сигнала);
- среднее значение частоты в спектре передаваемого сигнала;
- полоса пропускания, необходимая для качественной передачи данного сообщения.

7. *Результаты сравнительного анализа* рассмотренных способов кодирования (достоинства и недостатки), представленные в виде таблицы, и *обоснованный выбор наилучшего способа кодирования* для передачи исходного сообщения.

8. *Вид полинома*, используемого для скремблирования исходного сообщения, и *обоснование* его выбора. *Последовательность получения разрядов* скремблированного сообщения. *Результат скремблирования*, записанный в виде скремблированного сообщения в *двоичном* и *шестнадцатеричном* кодах.

9. *Временны**е** диаграммы* для двух способов физического кодирования (включая манчестерское кодирование) скремблированного сообщения.

Рассчитанные для каждого способа кодирования:

- верхняя и нижняя границы частот в передаваемом сообщении (спектр сигнала);
- среднее значение частоты в спектре передаваемого сигнала;
- полоса пропускания, необходимая для качественной передачи данного сообщения.

10. Результаты сравнительного анализа рассмотренных способов кодирования (*достоинства и недостатки*), представленные в виде таблицы, и *обоснованный выбор наилучшего способа кодирования* для передачи исходного сообщения.

11. Краткие в*ыводы с обоснованием наилучшего способа логического и физического кодирования* для передачи исходного сообщения.

12. Список использованной литературы.

1.6. Контрольные вопросы для самопроверки

При подготовке к защите отчета по выполненной работе следует руководствоваться следующим примерным перечнем вопросов и задач для самостоятельной проработки.

1. Что такое потенциальное кодирование?
2. При каком методе кодирования скорость модуляции (бод) и скорость передачи данных (бит в секунду) совпадают?
3. Как изменяется спектр сигнала при потенциальном кодировании, если в передаваемом сообщении появляется длинная последовательность нулей или единиц?
4. В каком случае при потенциальном кодировании в спектре сигнала отсутствует постоянная составляющая?
5. Почему потенциальные коды на каналах тональной частоты никогда не используются?
6. В чем отличие импульсных кодов от потенциальных?
7. Достоинства и недостатки методов кодирования NRZ, RZ, AMI, MLT-3, Манчестер 2, ….
8. Проиллюстрировать на диаграмме методы кодирования NRZ, RZ, AMI, MLT-3, Манчестер 2, ….
9. У какого из известных вам методов верхняя граница частот имеет наименьшее значение?
10. Нарисовать диаграммы методов кодирования NRZ, RZ, AMI, MLT-3, Манчестер 2… для сообщения, заданного в шестнадцатеричном коде: C5.
11. Определить частоту основной гармоники для сообщения, заданного в шестнадцатеричном коде: C5, при использовании методов кодирования NRZ, RZ, AMI, MLT-3, Манчестер 2….
12. Какой метод кодирования применяется в ЛВС Ethernet и Token Ring?
13. Перечислить методы логического кодирования.
14. Для чего используются методы логического кодирования?
15. Пояснить принципы метода избыточного кодирования и скремблирования.
16. Какой метод логического кодирования используется в ЛВС Fast Ethernet и FDDI?
17. Пояснить суть методов логического кодирования 4B/5B, 5B/6B, 8B/10B, 8B/6T.
18. Что такое «запрещенные коды» в методах избыточного кодирования?

19. Какой метод избыточного кодирования обладает наибольшей (наименьшей) избыточностью и почему?
20. Сколько избыточных кодов содержит метод кодирования 4B/5B (5B/6B, 8B/10B, 8B/6T).
21. Основной недостаток методов избыточного кодирования.
22. Что такое дескремблер?

Задание 2. Передача кодированных данных по каналу связи

2.1. Цель и краткая характеристика работы

Цель работы: исследование влияния свойств канала связи на качество передачи сигналов при различных методах физического и логического кодирования, используемых в цифровых сетях передачи данных.

В процессе выполнения учебно-исследовательской работы необходимо:

- для заданного исходного сообщения и заданных методов кодирования выполнить исследование качества передачи физических сигналов по каналу связи в зависимости от уровня шумов в канале, степени рассинхронизации передатчика и приёмника и уровня граничного напряжения (которое можно трактовать как уровень затухания сигнала);
- сравнить рассматриваемые методы кодирования;
- выбрать и обосновать наилучший метод для передачи исходного сообщения по реальному каналу связи с учетом затухания, шумов в канале и рассинхронизации.

Ориентировочная трудоемкость выполнения задания для:

- 2-х методов кодирования – 3 часа;
- 3-х методов кодирования – 4 часов;
- 4-х методов кодирования – 5 часов.

2.2. Теоретические сведения

В простейшем случае двоичные данные могут быть представлены в виде синусоидального сигнала, в котором положительная часть синусоиды соответствует двоичной «1», а отрицательная – «0». Период синусоиды равен $T = 2t$, где t – длительность битового интервала связана с пропускной способностью канала C зависимостью $t = 1/C$. Отсюда частота синусоидального сигнала $f = C/2$.

Передача сигнала по реальному каналу связи на большие расстояния характеризуется следующими особенностями.

1. Затухание сигнала в процессе распространения по каналу, в результате которого его мощность в точке приёма оказывается значительно меньше мощности исходного информативного сигнала, причем уменьшение мощности прямо пропорционально длине канала.

2. Искажение формы информативного сигнала из-за шумов каналообразующей аппаратуры и влияния различного рода помех. В результате этого реальный сигнал в точке приёма оказывается мало похожим на исходный сигнал, что может привести к тому, что сигнал пропадет (не будет считан), либо будет считано неверное значение.

3. Наличие внутренних шумов в канале связи, обусловленных техническими характеристиками среды передачи (линии связи) и каналообразующей аппаратуры, которые приводят к появлению фонового сигнала, искажающего информативный сигнал. Для того чтобы шум в канале связи не воспринимался на приёмной стороне как информативный сигнал, в приёмнике обычно устанавливается некоторое граничное значение уровня сигнала, соответствующее уровню естественного шума. Если мощность информативного сигнала в точке приёма меньше уровня шума, то он будет неразличим и, следовательно, потерян.

4. Необходимость синхронизации приемника с передатчиком для того, чтобы в приемнике снимать отсчёт в центре битового интервала, что позволит с большой уверенностью распознать значение информативного сигнала, поскольку наибольшую мощность синусоидальный сигнал сохраняет в центре битового интервала. Для качественного распознавания сигналов на приёмной стороне необходимо, чтобы часы передатчика и приёмника работали синхронно. Известно, что все часы имеют погрешность, которая со временем приводит к существенной разнице в показаниях часов, находящихся в узле-передатчике и узле-приёмнике. Это может привести к тому, что на приёмной стороне некоторые биты могут быть пропущены, либо значения некоторых битов будут считаны дважды. Для решения проблемы синхронизации в компьютерных сетях применяются самосинхронизирующиеся коды.

При потенциальном кодировании исходный прямоугольный сигнал, отображающий двоичные «1» и «0», является идеальным теоретическим сигналом, обладающим бесконечным спектром частот, который получается непосредственно из формул Фурье для периодической функции. Если дискретные данные, содержащие последовательность чередующихся «1» и «0», передаются с битовой скоростью C бит/с, то спектр состоит из постоянной составляющей нулевой частоты и бесконечного ряда гармоник с частотами f_0, $f_1 = 3f_0$, $f_2 = 5f_0$, $f_3 = 7f_0$,..., где f_0 – *частота основной гармоники*, при этом амплитуды гармоник убывают по отношению к

амплитуде основной гармоники A_0: $A_1 = A_0/3$, $A_2 = A_0/5$, $A_3 = A_0/7$, … . Таким образом, спектр потенциального кода требует для качественной передачи данных полосу пропускания в пределе равную бесконечности.

Реальные сигналы обладают ограниченным спектром, обусловленным наличием переднего и заднего фронта потенциального сигнала. Поэтому спектр результирующего сигнала потенциального кода при передаче произвольных данных занимает полосу от некоторой величины, близкой к 0 Гц, до некоторого конечного значения. На практике при передаче таких сигналов верхний предел спектра обычно ограничивается значениями $3f_0$, $5f_0$ или $7f_0$. Гармониками с частотами выше $7f_0$ можно пренебречь из-за их малого вклада в результирующий сигнал, поскольку амплитуды этих гармоник составляют 11% и менее от амплитуды основной гармоники.

Отметим, что при цифровой передаче данных для восстановления исходного сигнала требуется меньше гармоник, чем при аналоговой передаче. Технология передачи и приема цифровых сигналов позволяет восстановить исходный сигнал по основной гармонике (несущей), однако для уменьшения числа ошибок необходимо присутствие хотя бы первой гармоники, что, правда, втрое увеличивает спектр передаваемого сигнала и, следовательно, требуемой полосы пропускания канала связи.

2.3. Этапы выполнения работы и варианты заданий

Этап 1. Освоение программы для исследования качества передачи физических сигналов по каналу связи

Для исследования качества передачи исходного сообщения (сигналов) по каналу связи используется программа «Network Fourier 2», разработанная студентом Алексеем Безгодовым.

Назначение программы. Программа «Network Fourier 2» предназначена для имитационного моделирования процесса передачи дискретного сообщения с ограниченным спектром с учетом влияния шумов, рассинхронизации и уровня граничного напряжения. Сообщение может быть закодировано четырьмя способами физического и тремя способами логического кодирования.

Описание интерфейса. На рисунке 2.1 показан пользовательский интерфейс (окно) программы.

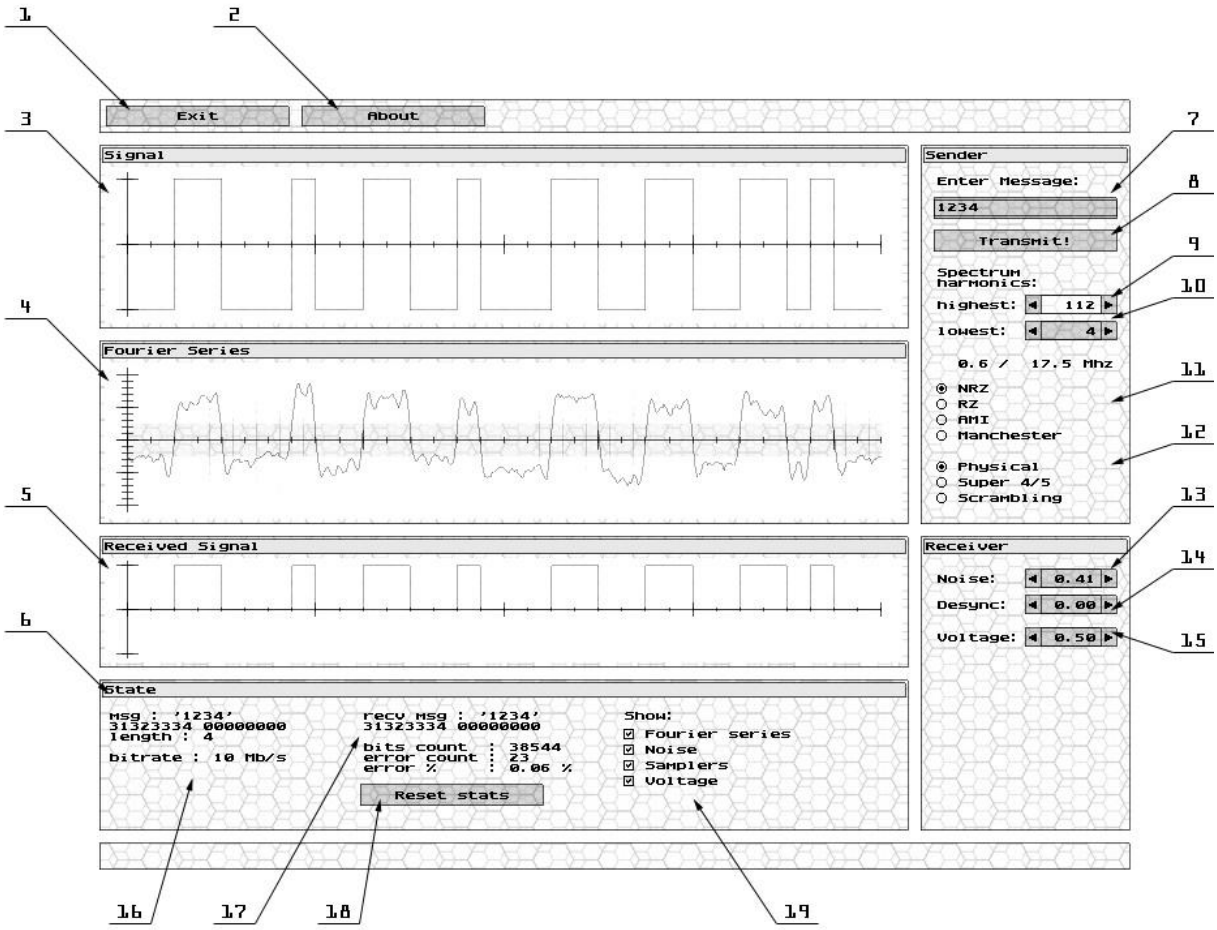


Рисунок 2.1. Интерфейс пользователя

Элементы интерфейса имеют следующие значения.

1. Кнопка выхода.
2. Кнопка вывода окна «о программе».
3. График закодированного сообщения.
4. График физического представления сигнала с учетом ограниченного спектра и шумов.
5. График принятого и дешифрованного сигнала.
6. Панель состояния.
7. Поле редактирования для ввода кодируемого сообщения. Сообщение может быть представлено либо в виде символов ASCII, либо в виде шестнадцатеричных чисел. Для ввода шестнадцатеричных чисел следует перед сообщением поставить символ «\». Например, «\123AB» будет соответствовать шестнадцатеричному числу 123AB. Для ввода текстового

сообщения, начинающегося с символа «\», следует ввести символ «\» два раза. Например, «\\xyz» будет представлено как «\xyz».

8. Кнопка пересылки сообщения.
9. Счетчик высшей гармоники ряда Фурье, диапазон [0..255].
10. Переключатель логического кодирования.
11. Счётчик для установки уровня шума, диапазон [0..2].
12. Счётчик для установки уровня рассинхронизации, диапазон [0..1].
13. Счетчик для установки граничного напряжения, диапазон [0..1].
14. Информация о передаваемом сообщении, ASCII и шестнадцатеричное представление сигнала, длина, скорость передачи (бит/с).
15. Счетчик низшей гармоники ряда Фурье, диапазон [0..255].
16. Переключатель физического кодирования.
17. Информация о принятом сообщении, количестве принятых бит, ошибочных бит и процентное количество ошибок.
18. Кнопка сброса статистики.
19. Флажковый переключатель отображаемой информации на графике физического представления сигнала.

Примечание: *для ускорения выбора требуемого значения в элементе управления «счётчик» можно использовать клавиши «вправо/влево».*

Описание алгоритма. Считается, что сообщение является периодическим, например, начальное сообщение «ABCD» будет представлено во времени как «…ABCDABCDABCDABCD…». Приложение постоянно осуществляет пересылку сообщения длиной в один период примерно 50 раз в секунду и производит сбор статистики об ошибках.

Ряд Фурье для функции периодической на интервале длиной $2m$ имеет вид:

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty} (a_k \cos\frac{k\pi x}{m} + b_k \sin\frac{k\pi x}{m}),$$

где коэффициенты ряда рассчитываются по формулам:

$$a_k = \frac{1}{m}\int_{-m}^{m} f(x)\cos\frac{k\pi x}{m}\,dx; \quad b_k = \frac{1}{m}\int_{-m}^{m} f(x)\sin\frac{k\pi x}{m}\,dx, \quad (k = 0,1,2,\ldots$$

Шум представляет собой функцию вида:

$$N(x,t) = \frac{a}{2}\sum_{i=1}^{\infty} \frac{\sin(ix+i^4 t)}{i},$$

где a– амплитуда, x– значение сигнала (напряжение), t– системное время (это дает практически случайный сдвиг фаз при имитации случайного шума).

Под уровнем рассинхронизации Δx подразумевается ширина интервала, в котором снимается уровень сигнала (напряжения). Расчёт значения сигнала осуществляется следующим образом:

$$x = x_0 + rand(\Delta x) - \frac{\Delta x}{2},$$

где $rand(a)$ – функция, которая возвращает случайное вещественное значение в интервале $[0; a]$.

Алгоритм функционирования модели. При моделировании передачи сообщения по каналу связи в имитационной модели каждые 20 мс выполняются следующие повторяющиеся шаги.

1. Проверка элементов управления и установка начальных параметров.
2. Формирование незакодированного сигнала на основе введённого сообщения.
3. Логическое кодирование сообщения.
4. Физическое кодирование сообщения.
5. Построение ряда Фурье с учетом выбранного спектра.
6. Наложение функции шума.
7. Сэмплирование сигнала с учетом граничного напряжения и рассинхронизации.
8. Физическое декодирование сигнала.
9. Логическое декодирование сигнала.
10. Подсчёт ошибок и сбор статистики.

Порядок работы с программой. Для выполнения экспериментов с помощью имитационной модели необходимо выполнить следующие шаги:

1. Установить требуемые параметры передачи сигнала: спектр, уровень шума (Noise), степень рассинхронизации (Desync) и граничное напряжение (Voltage).

2. Установить нижнюю (lowest) и верхнюю (highest) границу спектра (Spectrum harmonics) передаваемого сигнала.

3. Выбрать метод кодирования (NRZ, RZ, Manchester).

4. Ввести заданное сообщение в поле “Enter Message” и нажать клавишу “Transmit!”.

5. Сбросить статистику (клавиша “Reset stats”).

6. Дождаться выполнения требуемого количества пересылок (порядка 100 000 бит) и зафиксировать процент ошибок (error %).

7. Пункты 1-6 при необходимости выполняются для других параметров и методов кодирования.

Системные требования. Для корректного функционирования имитационной модели необходимо наличие не менее 32 МБ ОЗУ, а также операционная система Win98, WinXP или выше. Видеокарта должна иметь 3D-ускоритель для быстрого отображения элементов пользовательского интерфейса. Графический пользовательский интерфейс создан с использованием библиотеки OpenGL.

Этап 2. Определение минимальной полосы пропускания канала связи

Минимально требуемая полоса пропускания канала связи для качественной передачи сообщения (двоичного сигнала) определяется для *идеального* канала, в котором:

- отсутствуют шумы и помехи, искажающие форму сигнала;
- передающий и принимающий сигналы узлы абсолютно синхронизированы, т.е. нет рассинхронизации между ними;
- сигналы не затухают и нет необходимости устанавливать какой-то уровень граничного напряжения, позволяющего различить единичный и нулевой сигнал.

Для этого необходимо установить нулевые значения уровней: шумов (Noise), рассинхронизации (Desync) и граничного напряжения (Voltage) .

Затем в поле «Enter message» ввести исходное сообщение. В качестве исходного сообщения используется, как и в задании №1, первые четыре байта фамилии студента, выполняющего данное задание.

Указание. *Символы исходного сообщения вводятся в шестнадцатеричном виде в обратном порядке, т.е. вначале вводится шестнадцатеричный код четвертого байта, затем – третьего и т.д. В качестве признака шестнадцатеричного кода перед вводимым сообщением необходимо поставить символ «\».*

Последовательно изменяя значения нижней и верхней гармоник спектра сигнала, определить граничные значения, при которых сообщение передается без ошибок. Соответствующие им значения частот представляют собой нижнюю и верхнюю границы, определяющие минимальную полосу пропускания канала связи.

Этап 3. Определение максимально допустимых уровней шумов, рассинхронизации и затухания

На этом этапе последовательно определяются максимально допустимые уровни шумов, рассинхронизации и затухания, при которых сохраняется качественная передача сообщения, т.е. не наблюдается возникновение ошибок.

Вначале изменяется уровень шумов (Noise) и определяется максимально допустимый уровень шумов, при котором исходное сообщение передается без ошибок. При этом значения уровней рассинхронизации и граничного напряжения должны быть нулевыми.

Затем уровень шумов устанавливается в нулевое значение и изменяется уровень рассинхронизации (Desync) и определяется максимально допустимый уровень рассинхронизации, при котором исходное сообщение будет принято без ошибок.

Затем уровень рассинхронизации устанавливается в нулевое значение и изменяется уровень граничного напряжения (Voltage) и определяется максимально допустимый уровень граничного напряжения, при котором исходное сообщение передается без ошибок.

Этап 4. Оценка достоверности распознавания сигналов на приемном конце

На этом этапе определяется процент ошибок при передаче сообщения при найденных на предыдущем этапе значениях уровней шумов, рассинхронизации и граничного напряжения и минимальной полосы пропускания канала связи.

Установить найденные на предыдущем этапе максимально допустимые значения уровней шумов, рассинхронизации и граничного напряжения и определить процент ошибок на приемном конце канала связи.

Указание. *Этапы 2–4 последовательно выполняются для заданных преподавателем методов физического и логического кодирования. Полученные значения заносятся в таблицу результатов.*

Этап 5. Определение значений уровней шумов, рассинхронизации и граничного напряжения для реального канала связи

Рассчитать значения уровней шумов, рассинхронизации и граничного напряжения для реального канала связи как средние значения по всем рассмотренным методам кодирования.

Этап 6. Определение требуемой полосы пропускания реального канала связи

Требуемая полоса пропускания реального канала связи определяется из условия, что передача сообщения должна происходить без потерь при рассчитанных уровнях шумов, рассинхронизации и граничного напряжения для всех рассмотренных методов кодирования.

Установить рассчитанные значения уровней шумов, рассинхронизации и граничного напряжения для реального канала связи.

Последовательно изменяя значения порядкового номера нижней гармоники от нуля и верхней гармоники от максимального значения (255) спектра сигнала, определить граничные значения, при которых сообщение передается без ошибок по реальному каналу связи. Соответствующие им значения частот определяют требуемую полосу пропускания канала связи при рассматриваемом методе кодирования.

Указание. *Этот пункт выполняется для всех тех же методов физического и логического кодирования. Полученные значения занести в таблицу результатов.*

Этап 7. Анализ полученных результатов и выбор наилучшего способа кодирования исходного сообщения

Проанализировать полученные результаты и выбрать наилучший способ кодирования исходного сообщения из всех рассмотренных способов, аргументировано обосновав это выбор.

2.4. Порядок выполнения работы

1. Ознакомиться с постановкой задачи.
2. Ознакомиться с программой для исследования качества передачи физических сигналов по каналу связи (этап 1).
3. С использованием этой программы выполнить исследования в соответствии с этапами 2–6 и занести результаты в таблицу 2.1.
4. Выполнить сравнительный анализ рассмотренных способов кодирования и выбрать наилучший способ для передачи исходного сообщения (этап 7).
5. Подготовить отчёт по выполненной работе.

Таблица 2.1.

Шестнадцатеричный код сообщения: _____			Метод кодирования				
			NRZ	RZ	M-II	4B/5B	Scramb
Полоса пропускания *идеального* канала связи	Номера гармоник	min					
		max					
	Частоты, МГц	min					
		max					
Минимальная полоса пропускания *идеального* канала связи							
Уровень *шума*		max					
Уровень *рассинхронизации*		max					
Уровень *граничного напряж.*		max					
Процент ошибок при max уровнях и *минимальной* полосе пропускания КС							
Уровень *шума*		ср.					
Уровень *рассинхронизации*		ср.					
Уровень *граничного напряж.*		ср.					
Полоса пропускания *реального* канала связи	Гармоники	min					
		max					
	Частоты, МГц	min					
		max					
Требуемая полоса пропускания *реального* канала связи							

2.5. Требования к содержанию отчёта

1. Краткая постановка задачи.

2. Исходное сообщение и его представление в шестнадцатеричном виде.

3. Скриншот программы “Network Fourier 2”, на котором должно присутствовать передаваемое сообщение и должны быть выставлены характеристики реального канала связи.

4. Результаты исследований рассмотренных способов кодирования, представленные в виде таблицы 2.1, анализ полученных результатов и обоснованный выбор наилучшего способа кодирования для передачи исходного сообщения.

5. Краткие выводы с обоснованием наилучшего способа логического и физического кодирования для передачи исходного сообщения.

6. Список использованной литературы.

2.6. Контрольные вопросы для самопроверки

При подготовке к защите задания 2 следует руководствоваться следующим примерным перечнем вопросов и задач для самостоятельной проработки.

1. В чем состоит удобство вычисления затухания сигнала в дБ?
2. Во сколько раз уменьшится мощность сигнала на расстоянии 100 м, если его ослабление равно: d=10 дБ/км?
3. Нарисовать график гармонического сигнала и показать на графике его параметры. Записать функцию, описывающую гармонический сигнал.
4. Записать и пояснить представление функции, отображающей непрерывные данные, в виде ряда Фурье.
5. Понятие сигнала (функции) с ограниченным спектром.
6. Какой спектр частот характерен для дискретных сигналов?
7. При каких условиях обеспечивается качественная передача сигнала?
8. Проиллюстрировать на графике понятие полосы пропускания линии связи. Какую полосу пропускания имеет телефонный канал (аналоговая проводная линия связи)?
9. По каким каналам можно передавать дискретные сигналы в их естественном виде – без модуляции (в первичной полосе частот)?

10. Как передаются сигналы в высокоскоростных каналах связи с резко ограниченной полосой частот?
11. Что такое модуляция и для чего она нужна?
12. Чем манипуляция отличается от модуляции?
13. Пояснить принцип амплитудной, частотной и фазовой модуляции.
14. Что такое ИКМ?
15. Пояснить различие между АИМ и ИКМ.
16. Показать, за счет чего обеспечивается скорость передачи данных в 64 кбит/с (56 кбит/с) при ИКМ.
17. Пояснить принцип адаптивной разностной (дифференциальной) ИКМ.
18. В чём различие между линейным и первичным сигналом?
19. Перечислить характеристики цифрового канала связи.
20. От чего зависит пропускная способность канала связи?
21. Рассчитать максимально возможную пропускную способность (Мбит/с) канала связи при условии, что ширина полосы пропускания равна 20 МГц, а отношение мощности сигнала к мощности шума равно 3.
22. В чём отличие пропускной способности от скорости передачи данных?
23. Какие скорости передачи данных обеспечивает телефонный канал?
24. Какие методы мультиплексирования используются в вычислительных сетях?
25. Как называется процесс представления непрерывных данных в виде физических сигналов для их передачи по каналам связи?
26. Как называется процесс представления дискретных данных в виде физических сигналов для их передачи по каналам связи?
27. От чего зависит спектр результирующего модулированного сигнала?
28. Как спектр результирующего модулированного сигнала зависит от скорости модуляции (скорости передачи данных)? Ответ пояснить.
29. Перечислить требования к методам цифрового кодирования.
30. Как битовая скорость связана со спектром результирующего сигнала?
31. В чем заключается проблема синхронизации при передаче цифровых сигналов?
32. Что такое самосинхронизирующийся код?
33. Какие методы кодирования относятся к самосинхронизирующимся?

34. От чего зависит стоимость реализации метода кодирования?
35. Что такое постоянная составляющая спектра сигнала и почему она нежелательна?
36. Какие методы кодирования имеют постоянную составляющую в спектре сигнала?
37. Почему в телекоммуникационных сетях для синхронизации не используется схема, основанная на отдельной тактирующей линии связи?
38. Почему проблема синхронизации в телекоммуникационных сетях решается сложнее, чем при обмене данными между компьютером и принтером?

Задание 3. Анализ трафика компьютерных сетей утилитой Wireshark

3.1 Цель и краткая характеристика работы

Цель работы – изучить структуру протокольных блоков данных, анализируя реальный трафик на компьютере студента с помощью бесплатно распространяемой утилиты Wireshark.

В процессе выполнения домашнего задания выполняются наблюдения за передаваемым трафиком с компьютера пользователя в Интернет и в обратном направлении. Применение специализированной утилиты Wireshark позволяет наблюдать структуру передаваемых кадров, пакетов и сегментов данных различных сетевых протоколов. При выполнении УИР требуется анализировать последовательности команд и назначение служебных данных, используемых для организации обмена данными в следующих протоколах: ARP, DNS, FTP, HTTP, DHCP.

При составлении задания 3 использовались материалы книги J.F. Kurose “Computer Networking: A Top-Down Approach” (6th edition, 2012) при непосредственном участии студентки кафедры ВТ Университета ИТМО Полины Нужиной.

Приблизительная трудоёмкость УИР №3 для выполнения:

- пяти пунктов задания – 4 астрономических часа;
- семи пунктов задания – 6 астрономических часов;
- всех пунктов задания – 9 астрономических часов.

3.2 Теоретическая справка

Процесс передачи данных по компьютерным сетям является сложным комплексом процедур, выполняемых с применением большого количества разнообразным программных и аппаратных средств. Для упрощения анализа и проектирования таких сложных систем, общепринятой практикой является декомпозиция сложного процесса на модули и/или иерархические структуры.

Обычно целью декомпозиции является получение таких модулей, которые выполняют отведённые им функции изолированно от других модулей, передавая соседним модулям лишь конечные результаты работы. Это позволяет рассматривать и проектировать модули независимо друг от друга различным, не связанным друг с другом группам инженеров, каждая из которых обладает узкой квалификацией, необходимой для реализации конкретного модуля. Кроме этого, система, обладающая модульной

структурой, позволяет при необходимости модифицировать внутреннюю реализацию отдельных модулей, не изменяя что-либо в соседних модулях.

В компьютерных сетях общепринятой моделью декомпозиции процесса передачи данных является OSI-модель, разработанная международной организацией по стандартизации. OSI означает “Open Systems Interconnection”, т.е. взаимодействие открытых систем. Модель OSI определяет 7 модулей, называемых уровнями (layer), каждый из которых описывает реализацию некоторого множества родственных сетевых операций, которые выполняет ЭВМ, начиная от момента получения данных от пользователя и заканчивая отправлением физического сигнала (например, радиоволны) в сеть. Уровни связываются между собой строго последовательно:

“пользователь” ⇔7 ⇔6 ⇔5 ⇔4 ⇔3 ⇔2 ⇔1 ⇔“сеть”

(здесь цифрами обозначены номера соответствующих уровней). Это значит, что после выполнения сетевых функций некоторого уровня результаты его деятельности могут быть переданы только соседним уровням. Результатами деятельности являются закодированные блоки данных, называемые PDU (протокольные блоки данных). Обычно при движении PDU по OSI-модели от “пользователя” в “сеть” каждый уровень дополняет полученный PDU своими служебными данными. В результате PDU 2-го уровня может иметь следующую структуру (прямоугольники обозначают последовательность бит: количество бит пропорционально длине прямоугольника):

СД2	СД3	СД4	СД5	СД6	СД7	ДП

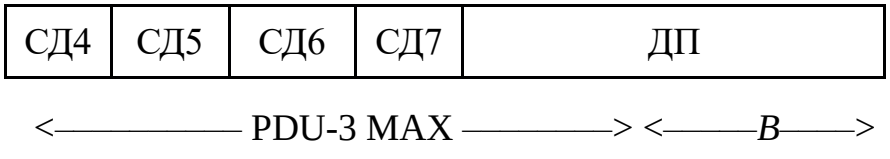
Здесь СД*i* – это служебные данные, добавленные *i*-м уровнем, а ДП – это данные пользователя, который он хотел передать по сети. Служебные данные некоторых уровней могут быть организованы в виде двух частей: заголовка и концевика (в этом случае структура PDU выглядит иначе, чем показано выше).

При движении PDU по OSI-модели в обратном направлении (т.е. из “сети” к “пользователю”) каждый уровень сначала использует одноимённые служебные данные для выполнения заданной сетевой функции, а затем “отстёгивает” их при передаче следующему в цепочке уровню. Описанный подход гарантирует невмешательство уровней в работу друг друга, обеспечивая хорошую “модульность“ процесса, однако предполагает достаточно большое количество служебных данных, которые

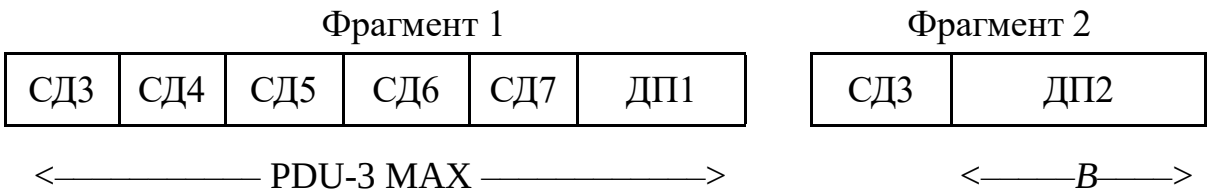
могут дублироваться на разных уровнях, что увеличивает накладные расходы на передачу полезных ДП.

В некоторых случаях правила уровня могут накладывать ограничения на размер PDU, который может быть корректно обработан уровнем. В этом случае при попытке передать PDU большего размера, PDU будет либо отвергнут с сообщением об ошибке, либо будет фрагментирован на несколько частей, каждая из которых будет передана независимо. При использовании фрагментирования требуется, чтобы принимающая фрагменты сторона могла соединить фрагменты воедино.

Пусть следующий PDU 4-го уровня имеет размер, который превышает предельно допустимый PDU 3-го уровня (PDU-3 MAX) на *B* байт:



В этом случае на вход 2-го уровня вместо одного будет передано сразу несколько PDU 3-го уровня, которые будут представлять собой фрагменты, имеющие допустимую для 3-го уровня длину:



В рассмотренном случае удалось разбить исходный PDU 4-го уровня всего на два PDU 3-го уровня. Данные пользователя пришлось “разрезать” на две части ДП1 и ДП2. Суммарный размер ДП1 и ДП2, очевидно, равен ДП. Однако при фрагментировании пришлось добавить СД3 в оба фрагмента, что привело к увеличению доли накладных расходов в передаваемом сообщении. Это необходимо для того, чтобы была возможность корректно собрать PDU-4 из фрагментов на приёмной стороне.

Дадим краткую характеристику каждому из 7 уровней OSI-модели, двигаясь по модели в направлении от пользователя в сеть (более подробное описание уровней см. в [1]).

Прикладной уровень (Application Layer, L7) описывает, как выглядит процесс передачи с точки зрения конечного пользователя или приложения. L7 предоставляет понятные пользователю высокоуровневые “рычаги” для получения сервисов уровней L1-L6. Конечный пользователь или приложение при работе с сетью взаимодействует только с L7, а все нижележащие уровни от него скрыты, т.е. инкапсулированы в L7. Этот уровень имеет нечёткие границы, так как может описывать не только функции сетевого приложения, но и возможные действия пользователя. На уровне L7 *может* описываться:

- авторизация и аутентификация пользователя;
- контроль целостности конечных пользовательских данных, которые были получены из сети;
- синхронизация действий или файлов пользователей, взаимодействующих по сети (например, при совместном редактировании файла несколькими пользователями);
- … а также любая из функций уровней L2-L6 (см. ниже), если она не была реализована на L2-L6, либо была реализована не в полном соответствии с потребностями приложения или пользователя.

Уровень представления (Presentation Layer, L6) описывает, как взаимодействующие стороны “договариваются” о формате, в котором будут представлены данные пользователя при передаче по сети. На этом уровне могут описываться:

- процедура согласования формата представления данных на этапе установки соединения (например, выбор UTF-8 для кодирования текста пользователя или выбор алгоритма сжатия и его параметров и т.п.);
- правила изменения текущего формата представления данных в некотором уже установленном соединении (например, изменение кодека, сжимающего аудиопоток во время VoIP-разговора, при обнаружении перегрузки канала связи);
- описание синтаксиса выбранного формата представления данных, если он не является общеизвестным стандартом и не может быть описан простой ссылкой на стандарт (например, новый патентованный

алгоритм сжатия или особый порядок следования байтов, отличный от Big Endian и Little Endian).

Сеансовый уровень (Session Layer, L5) описывает процесс установки, разрыва и поддержания соединений. На этом уровне может описываться:

- процедура установки соединения и согласования параметров соединения (например, требований QoS – см. ниже), при этом фактическая реализация запрошенных требований осуществляется на уровне L4;
- процедура разрыва соединения как при явном запросе пользователя, так при получении от L4 сообщения о невозможности выполнить запрошенные требования QoS (может быть описан штатный разрыв без потери данных пользователя и/или быстрый “жёсткий” сброс соединения с риском потери данных);
- процедура (ре)синхронизации состояния соединения (соединение может рассинхронизироваться при возникновении ошибок в сети, при переводе часов, при выходе за границу окна передачи и т.п.).

Транспортный уровень (Transport Layer, L4) описывает процесс межоконечной (end-to-end, “из конца в конец”) передачи данных по сети, т.е. передачу с точки зрения наблюдателя, для которого все промежуточные сетевые устройства между абонентами рассматриваются как единый “черный ящик”, структура которого неизвестна. На этом уровне может описываться:

- процедура установки/поддержания/разрыва соединения и передачи данных с учётом соблюдения требований QoS, полученных от L5 (например, может понадобиться установить сразу несколько L4-соединений или выбрать такие L4-параметры, которые гарантируют соблюдение QoS-требований с большим запасом, ввиду отсутствия возможности точной настройки);
- процедура реагирования на обнаружение искажённых или потерянных пакетов (следует ли повторить передачу или же допустимо игнорировать потерю/искажение?);

- процедура сохранения корректного порядка поступления PDU конечному абоненту (PDU снабжаются порядковыми номерами, при этом может потребоваться буфер для временного хранения PDU, поступивших с нарушением порядка);
- процедуры манипуляции с размером PDU: мультиплексирование потоков, разбиение больших PDU на более мелкие, объединение маленьких PDU в большие и т.п.

Сетевой уровень (Network Layer, L3) описывает процесс передачи PDU через промежуточные узлы сети, включая выбор маршрута следования при наличии нескольких маршрутов. При этом маршрут передачи может пересекать несколько объединённых разнотипных сетей. Сетевой уровень полностью скрывает от вышестоящих уровней (L4-L7) особенности маршрутизации и передачи PDU через разнородные сети, т.к. реализует их самостоятельно или средствами уровня L2. На уровне L3 может описываться:

- установка/поддержание/разрыв соединения в условиях, когда необходимо пересекать границы нескольких сетей (при пересечении границы между сетями разных провайдеров может потребоваться инициировать установку соединения независимо от установки межоконечного соединения на L5);
- процедура сохранения корректного порядка поступления PDU на границе сетей (ср. с аналогичным пунктом в L3);
- правила маршрутизации и построения маршрутных таблиц при пересечении границ сетей;
- процедуры манипуляции с размером PDU (см. L4);
- правила подтверждения получения PDU приёмником.

Канальный уровень (Data Link Layer, L2) описывает логические правила передачи PDU в пределах простой сети, построенной в рамках единой технологии с одинаковыми однотипными линиями связи. L2 скрывает от вышестоящих уровней физические особенности сети. На этом уровне может описываться:

- установка, поддержание и разрыв соединения в рамках одной локальной сети с возможностью согласования узлами параметров

передачи (при подключении к Wi-Fi требуется установить L2-соединение с базовой станцией Wi-Fi независимо от установки соединения на L3 и L5);

- процедура реагирования на обнаружение искажённых или потерянных пакетов (сравни с L4);
- синхронизация приемо-передатчиков сетевых устройств для корректного распознавания границ PDU (например, отправка блока известной абонентам длины с чередованием 0 и 1: “01010101...”; или использование запрещённых сигналов J, K);
- процедура сохранения корректного порядка поступления PDU внутри сети (сравни с аналогичным пунктом в L3 и L4);
- правила разделения потока PDU на несколько подпотоков для возможности их одновременной передачи по нескольким физическим линиям связи (например, с использованием нескольких радиоканалов с разным диапазоном частот);
- правила маршрутизации и построения маршрутных таблиц внутри сети (сравни с аналогичным пунктом в L3; обычно в L2-сетях маршрутизация не требуется, т.к. в них всегда существует ровно один маршрут).

Физический уровень (Physical Layer, L1) описывает с физической точки зрения процессы передачи PDU по некоторой конкретной линии связи. Сюда может входить спецификация физических свойств:

- *среды передачи*: ширина полосы пропускания радиоканала в МГц, максимальная длина провода при передаче по витой паре или оптоволокну, количество и назначение проводов или волокон в кабеле и т.п.;
- *передаваемого сигнала*: используемые длины волн или напряжение тока, способ кодирования битов в виде конкретного уровня напряжения, длительность времени передачи бита или группы битов, мультиплексирование нескольких физических сигналов в одной линии связи, показатели QoS линии связи (задержка распространения сигнала, доля битовых ошибок BER, скорость передачи в бодах и др.) и т.п.;

- *сетевого оборудования*: количество и назначение контактов в штекере сетевой карты или маршрутизатора, количество и физическое устройство антенн в радиопередатчике, способ передачи (полнодуплексный, полудуплексный, симплексный), способ активации линии связи при включении или начале передачи и т.п.

Существуют также некоторые универсальные функции, которые реализуются почти на всех уровнях OSI-модели. К таким функциям могут относиться следующие:

- **Адресация**. На каждом уровне (кроме L6) для идентификации взаимодействующих сторон может использоваться адрес определённого формата. На L2 это может быть адрес физического устройства сети, специфичный для некоторой конкретной технологии построения сети; на L3 это будет универсальный межсетевой адрес, пригодный для передачи между разнородными сетями; на L5 адресом будет номер соединения; на L7 адресом может быть имя пользователя или понятный человеку текстовый адрес. В итоге к моменту формирования PDU L1 в передаваемом блоке данных может содержаться 6 различных адресов-идентификаторов.
- **Обнаружение ошибок**. На каждом уровне есть своя специфика процесса обнаружения ошибок. На L1 об ошибке может сигнализировать запрещённое значение напряжения тока или неверная последовательность радиоимпульсов. На более высоких уровнях для обнаружения ошибок могут использоваться различные виды контрольных сумм, которые в том числе позволяют исправлять найденные ошибки (например, код Хэмминга). Если на L3 ошибки могут быть обнаружены в промежуточных узлах сети (при пересечении границ сетей), то на L4 проверка на ошибки возможна только в конечном узле-получателе. Ещё один класс обнаруживаемых ошибок связан с нарушением логических правил протокола обмена сообщениями, например, если абонент высылает первый блок данных до окончания процедуры установки соединения на L5.
- **Показатели QoS** (Quality of Service), т.е. метрики качества передачи данных по сети, например: задержка передачи, время установления соединения, доля потерянных пакетов и др. На разных уровнях OSI-модели рассматриваются различные аспекты QoS. На L1 метрики

характеризуют качество передачи по одной конкретной линии связи, например в виде значения мощности сигнала базовой станции WiFi; на L2 контролируется QoS в рамках локальной сети; на L3 описывается QoS при пересечении границ сетей; на L4 контролируется качество передачи из конца в конец; на L5 описан порядок согласования абонентами желаемых показателей QoS во время установки соединения; на L7 пользователь может запросить желаемый битрейт потока для обеспечения нужного качества сетевой видеотрансляции и т.д.

- **Установка соединения**. При описании функций L2, L3 и L5 (см. выше) показано, что в процессе передачи данных может потребоваться выполнить несколько независимых процедур установки соединения. Например: L2-соедниение к Wi-Fi-точке, L3-соединение к провайдеру Интернет-услуг, L5-соединение или L7-содинение с публичным FTP-сервером, находящимся в Интернете (т.е. с конечным адресатом).

Существующие реализации OSI-модели. Стандарт с описанием OSI-модели был опубликован в 1984 году, однако с тех пор так и не появилось ни одной популярной сетевой технологии, которая бы строго реализовала все уровни этой модели. Наиболее популярный стек сетевых протоколов TCP/IP, разработанный до публикации OSI-модели, лишь с большой натяжкой можно соотнести с уровнями OSI:

- **Канальный уровень TCP/IP** приблизительно реализует функции L1 и L2. В качестве адреса на этом уровне используется MAC-адрес сетевого устройства, а PDU этого уровня называется кадром (фреймом, frame). Этот уровень описывает процесс передачи данных в рамках локальной сети.

- **Сетевой уровень TCP/IP** приблизительно соответствует L3. В качестве адреса на этом уровне используется IP-адрес, а PDU этого уровня называется пакетом (packet). Этот уровень описывает процесс передачи данных через несколько объединенных и, возможно, разнородных локальных сетей.

- **Транспортный уровень TCP/IP** приблизительно реализует функции L4 и L5. В качестве адреса на этом уровне используется пара чисел, однозначно идентифицирующих соединение: порт отправителя и порт

получателя (например, UDP-порты), а PDU этого уровня называется сегментом или датаграммой (segment, datagram).

- **Прикладной уровень TCP/IP** приблизительно реализует функции L5, L6 и L7 (обратите внимание, что L5 фигурирует дважды: на прикладном и на транспортном уровне TCP/IP). В качестве адреса на этом уровне используется URL сайта, DNS-имя хоста, имя пользователя, email-адрес и т.д.

3.3. Этапы выполнения работы и варианты заданий

Для выполнения УИР необходимо установить на компьютер бесплатно распространяемую программу Wireshark, представляющую из себя анализатор сетевых пакетов, проходящих через интерфейсы компьютера. Скачать Wireshark можно с официального сайта: https://www.wireshark.org/#download. На рисунке 3.1 представлено главное окно *Wireshark*.

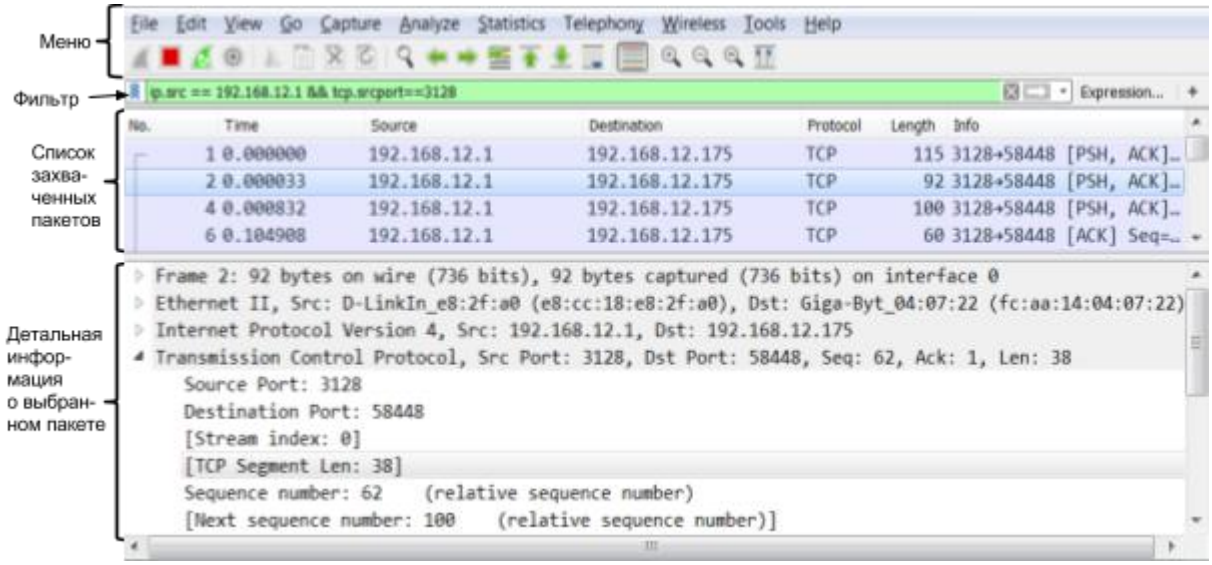


Рисунок 3.1. Графический интерфейс пользователя Wireshark

Используя “Меню”, можно выбрать сетевой интерфейс, “прослушивание” которого будет осуществлять Wireshark (кнопка “Capture options”). При выполнении работы следует удостовериться, что интерфейс выбран правильно, так как при наличии нескольких каналов доступа в Интернет (Wi-Fi, 4G, FastEthernet), как правило, по умолчанию используется только один из них и именно с него Wireshark должен “захватывать” проходящие пакеты.

В поле “Фильтр” пользователь может указать булево выражение (в стиле языка С), которое используется для выборочного отображения захваченных пакетов в “Списке захваченных пакетов”. Например, если в “Фильтре” указать строку “(*ip.src==192.168.12.1) && (tcp.srcport==3128)*” (без кавычек), то в “Списке захваченных пакетов” будут отображаться только те пакеты, которые были отправлены с IP-адреса *192.168.12.1* и при этом в поле “порт источника” протокола TCP содержат число 3128. Если фильтр принимает значение “http”, то будут отображаться только пакеты, переданные с использованием протокола http.

Дальнейшее выполнение лабораторной работы состоит из следующих шагов:

1. Запустить Wireshark (иногда для этого требуются права Администратора). В появившемся окне выбрать интерфейс для которого необходимо осуществлять анализ проходящих через него пакетов. В качестве интерфейса, используемого для захвата трафика, выбрать физический адаптер, через который компьютер подключён к Интернету (обычно этот адаптер называется Local или “Подключение по локальной сети”). Если меню для выбора адаптера не появляется при запуске Wireshark, нужно запустить из “Меню” команду “Capture->Options”. После выбора адаптеру, нужно запустить процесс захвата трафика (кнопка Start).
2. Инициировать процесс передачи трафика по сети (например, в браузере открыть сайт, заданный по варианту, или запустить соответствующую сетевую утилиту – см. ниже);
3. Установить значение “Фильтра”, чтобы из всего множества перехватываемых пакетов Wireshark отобразил только те, которые имеют отношение к выполняемому заданию. Для корректного создания фильтра следует пользоваться всплывающими подсказками Wireshark, которые активизируются при наборе фильтра. В качестве альтернативного способа можно использовать интерактивный конструктор фильтра, нажав на кнопку “Expression” в правой части элемента “Фильтр”.
4. Дождаться появления данных в списке захваченных пакетов и убедиться, что количество пакетов достаточно для выполнения задания.

5. Сохранить захваченный трафик в файл-трассу (pcap). Указанный файл нужно предъявить по первому требованию преподавателя во время защиты, если в этом возникнет необходимость.
6. Описать в отчёте структуру наблюдаемых PDU (т.е. протокольных блоков данных: кадров, пакетов, сегментов) как для запросов, так и ответов. Указать название и назначение всех заголовков всех уровней OSI-модели в пакетах с учётом порядка инкапсуляции (для этого нужно раскрывать соответствующие значки «+» в поле с детальной информацией о выбранном пакете).
7. Написать в отчёте ответы на вопросы задания (для этого может потребоваться самостоятельно изучить назначение соответствующей заданию сетевой утилиты, использованной для создания трафика).
8. Поместить в отчёт скриншоты окна Wireshark, иллюстрирующие ответы из вышеуказанных п.6 и п.7.

В качестве адреса сайта в заданиях следует использовать один из следующих URL (следует выбрать один из пунктов в порядке перечисления):

- Адрес, выбранный по явному указанию преподавателя. Если преподаватель не давал соответствующих указаний, нужно использовать следующие пункты.
- Адрес сайта с домашней страницей студента. Автор страницы должен легко идентифицироваться с этой страницей по содержимому сайта.
- Адрес сайта, в название которого лексически входит фамилия студента (например: www.sidorovivan.ru).
- Адрес сайта, в котором по очереди встречаются инициалы (ФИО) студента в латинской транскрипции (например, для имени Иванов Фёдор Михайлович подойдёт адрес сайта http://ifmo.ru).

Примечание 1. При выполнении анализа HTTP-трафика не принимать во внимание HTTP-запрос и HTTP-ответ для файла *favicon.ico*. Появление ссылки на данный файл означает, что браузер автоматически запрашивает сервер о наличии значка веб-сайта, который отображается браузером в адресной строке перед адресом страницы (и в некоторых других местах).

Примечание 2. Все используемые в УИР утилиты доступны как в ОС MS Windows, так и Linux, однако в примерах к заданию указывается

синтаксис и ключи командной строки для MS Windows. В Linux команды будут иметь несколько иной синтаксис.

3.4 Порядок выполнения работы

3.4.1 Анализ трафика утилиты ping

Необходимо отследить и проанализировать трафик, создаваемый утилитой ping, запустив её следующим образом из командной строки:

“ping -l размер_пакета адрес_сайта_по_варианту”.

Например, “ping -l 2000 wireshark.org” (без кавычек).

В качестве “размера_пакета” необходимо поочерёдно использовать различные значения от 100 до 10000, самостоятельно выбрав шаг изменения. По результатам анализа собранной трассы, необходимо ответить на следующие вопросы и выполнить указанные задания.

1. Имеет ли место фрагментация исходного пакета, какое поле на это указывает?
2. Какая информация указывает, является ли фрагмент пакета последним или промежуточным?
3. Чему равно количество фрагментов при передаче ping-пакетов?
4. Построить график, в котором на оси абсцисс находится размер_пакета, а по оси ординат – количество фрагментов, на которое был разделён каждый ping-пакет.
5. Как изменить поле TTL с помощью утилиты ping?
6. Что содержится в поле данных ping-пакета?

3.4.2 Анализ трафика утилиты tracert (traceroute)

Необходимо отследить и проанализировать трафик, создаваемый утилитой tracert (или traceroute в Linux), запустив её следующим образом из командной строки:

“tracert -d адрес_сайта_по_варианту”

Например, tracert wireshark.org.

По результатам анализа собранной трассы, ответьте на следующие вопросы.

1. Сколько байт содержится в заголовке IP? Сколько байт содержится в поле данных?
2. Как и почему изменяется поле TTL в следующих друг за другом ICMP-пакетах tracert? Для ответа на этот вопрос нужно проследить

изменение TTL при передаче по маршруту, состоящему из более чем двух хопов.

3. Чем отличаются ICMP-пакеты, генерируемые утилитой tracert, от ICMP-пакетов, генерируемых утилитой ping (см. предыдущее задание).
4. Чем отличаются полученные пакеты «ICMP reply» от «ICMP error» и зачем нужны оба этих типа ответов?
5. Что изменится в работе tracert, если убрать ключ “-d”? Какой дополнительный трафик при этом будет генерироваться?

3.4.3 Анализ HTTP-трафика

Необходимо отследить и проанализировать HTTP-трафик, создаваемый браузером при посещении Интернет-сайта, заданного по варианту. В списке захваченных пакетов необходимо проанализировать следующую пару HTTP-сообщений (запрос-ответ):

- GET-сообщение от клиента (браузера);
- ответ сервера.

Для этого в поле с детальной информацией о пакете нужно развернуть строку “HTTP”. Затем необходимо обновить страницу в браузере так, чтобы вместо «HTTP GET» был сгенерирован «HTTP CONDITIONAL GET» (так называемый «условный GET»). Условные запросы GET содержат поля If-Modified-Since, If-Match, If-Range и подобные, которые позволяют при повторном запросе не передавать редко изменяемые данные. В ответ на условный GET тело запрашиваемого ресурса передается только в том случае, если этот ресурс изменялся после даты «If-Modified-Since». Если ресурс не изменялся, сервер вернет код статуса «304 Not Modified».

По результатам анализа собранной трассы покажите, каким образом протокол HTTP передавал содержимое страницы при первичном посещении страницы и при вторичном запросе-обновлении от браузера (т.е. при различных видах GET-запросов).

3.4.4 Анализ DNS-трафика

Необходимо отследить и проанализировать трафик протокола DNS, сгенерированный в результате выполнения следующих действий:

- настроить Wireshark-фильтр: “ip.addr == ваш_IP_адрес”;
- очистить кэш DNS с помощью команды ipconfig в командной строке: ipconfig /flushdns

- очистить кэш браузера;
- зайти на Интернет-сайт, заданный по варианту.

По результатам анализа собранной трассы, ответьте на следующие вопросы.

1. Почему адрес, на который отправлен DNS-запрос, не совпадает с адресом посещаемого сайта?
2. Какие бывают типы DNS-запросов?
3. В какой ситуации нужно выполнять независимые DNS-запросы для получения содержащихся на сайте изображений?

3.4.5 Анализ ARP-трафика

Необходимо отследить и проанализировать трафик протокола ARP, сгенерированный в результате выполнения следующих действий:

- очистить ARP-таблицу командой “netsh interface ip delete arpcache” (проверить очистилась ли таблица можно с помощью команды команды “arp -a”, выводящей таблицу на экран);
- очистить кэш браузера;
- зайти на Интернет-сайт, заданный по варианту.

По результатам анализа собранной трассы, ответьте на следующие вопросы.

1. Какие MAC-адреса присутствуют в захваченных пакетах ARP-протокола? Что означают эти адреса? Какие устройства они идентифицируют?
2. Какие MAC-адреса присутствуют в захваченных HTTP-пакетах и что означают эти адреса? Что означают эти адреса? Какие устройства они идентифицируют?
3. Для чего ARP-запрос содержит IP-адрес источника?

3.4.6 Анализ трафика утилиты nslookup

Это задание является необязательным, его необходимо выполнить только для желающих получить оценку «**хорошо**» или «**отлично**». Необходимо отследить и проанализировать трафик протокола DNS, сгенерированный в результате выполнения следующих действий:

1. Настроить Wireshark-фильтр: “ip.addr == ваш_IP_адрес”.
2. Запустить в командной строке команду “nslookup адрес_сайта_по_варианту”.

3. Дождаться отправки трёх DNS-запросов и трёх DNS-ответов (в работе нужно использовать только последние из них, т.к. первые два набора запросов/ответов специфичны для nslookup и не генерируются другими сетевыми приложениями).
4. Повторить предыдущие два шага, используя команду: “nslookup -type=NS имя_сайта_по_варианту”.

По результатам анализа собранной трассы, ответьте на следующие вопросы.

1. Чем различается трасса трафика в п.2 и п.4, указанных выше?
2. Что содержится в поле «Answers» DNS-ответа?
3. Каковы имена серверов, возвращающих авторитативный (authoritative) отклик?

3.4.7 Анализ FTP-трафика

Это задание является необязательным, его необходимо выполнить только для желающих получить оценку «**хорошо**» или «**отлично**». Необходимо отследить и проанализировать трафик протокола FTP, сгенерированный в результате выполнения следующих действий:

- настроить Wireshark-фильтр «ftp || ftp-data»;
- скачать в браузере небольшой файл с соответствующего варианту FTP-сервера в Интернете.

В адресной строке путь к скачиваемому файлу должен начинаться с «ftp://». Адрес сайта нужно выбрать, руководствуясь правилами, указанными в п. 3.3 задания №3.

По результатам анализа собранной трассы, ответьте на следующие вопросы.

1. Сколько байт данных содержится в пакете FTP-DATA?
2. Как выбирается порт транспортного уровня, который используется для передачи FTP-пакетов?
3. Чем отличаются пакеты FTP от FTP-DATA?

3.4.8 Анализ DHCP-трафика

Это задание является необязательным, его необходимо выполнить для желающих получить оценку «отлично». Необходимо отследить и проанализировать трафик протокола DHCP, сгенерированный в результате выполнения следующих действий:

1. Убедиться, что для назначения IP-адреса на компьютере был использован DHCP и что компьютеру был назначен IP-адрес.

2. Настроить Wireshark-фильтр «bootp» (во время защиты УИР следует объяснить, почему именно такой фильтр используется для анализа DHCP-трафика).
3. Сбросить текущий IP-адрес, выданный накануне перед этим DHCP-сервером, с помощью команды:
   “ipconfig /release“.
4. Запросить новый IP-адрес с помощью команды:
   “ipconfig /renew“.
5. Повторить п.3 и п.4.

Нарисуйте временную диаграмму, иллюстрирующую последовательность обмена первыми четырьмя DHCP-пакетами Discover/Offer/Request/ACK. Укажите для каждого пакета номера портов источника и назначения. По результатам анализа собранной трассы, ответьте на следующие вопросы.

1. Чем различаются пакеты «DHCP Discover» и «DHCP Request»?
2. Как и почему менялись MAC- и IP-адреса источника и назначения в переданных DHCP-пакетах.
3. Каков IP-адрес DHCP-сервера?
4. Что произойдёт, если очистить использованный фильтр “bootp”?

3.4.9 Анализ Skype-трафика

Это задание является необязательным, его необходимо выполнить для желающих получить оценку «отлично». Необходимо отследить и проанализировать трафик Skype (или любой другой аналогичной по функциональности программы), сгенерированный в результате выполнения следующих действий:

- отправить текстовое сообщение и получить ответ;
- осуществить короткий сеанс аудио-общения;
- осуществить короткий сеанс видео-общения.

Для упрощения анализа передачи различных видов трафика Скайпом (тест, аудио, видео) можно независимо собрать трассы трафика для каждого из трёх перечисленных пунктов, останавливая и возобновляя захват трафика так, чтобы получить три отдельных файла. По результатам анализа трёх собранных видов трасс трафика ответьте на следующие вопросы.

1. Чем различаются пакета разных видов Skype-трафика (текст, аудио, видео)?

2. Какой Wireshark-фильтр следует использовать для независимой идентификации Skype-трафика разных видов (текст, аудио, видео)?

Примечание. При выполнении п. 2.4.9 вместо Skype можно использовать любое другое аналогичное по функциональности программное обеспечение (Yahoo Messenger, MSN, Tox, «Mail.ru Агент» и любые другие)

3.5 Требования к содержанию отчёта

В работе требуется проанализировать сетевой трафик, захваченный с помощью программы Wireshark. В отчёте, предоставляемом в электронном или бумажном виде, следует привести скриншоты, иллюстрирующие ответы на поставленные в задании вопросы. Каждый скриншот должен иметь поясняющий текст, подробно раскрывающий содержание ответа на соответствующие вопросы.

Также в отчёте необходимо привести структуру наблюдаемых пакетов (как запросов, так и ответов), коротко описав назначение всех заголовков всех уровней с учётом порядка инкапсуляции. Стоит привести описание только тех пакетов, которые существенно различаются структурно (однотипные похожие пакеты приводить не надо), либо имеют непосредственное отношение к ответам на вопросы задания.

При защите отчёта необходимо иметь при себе сохраненную версию захваченного трафика на flash-носителе в формате *pcap* (так называемую трассу, или дамп, трафика).

3.6 Контрольные вопросы для самопроверки

При подготовке к защите задания следует руководствоваться следующим примерным перечнем вопросов и задач для самостоятельной проработки.

1. Что такое OSI-модель и для чего она нужна?
2. Перечислите уровни OSI-модели и дайте им краткую характеристику.
3. Какие преимущества даёт многоуровневая архитектура OSI-модели? Какие бы возникли сложности, если процесс передачи данных по сети был одноуровневым?
4. Если перед отправкой данных выполняется их сжатие, то на каком уровне OSI-модели следует выполнять эту операцию?
5. Как соотносится модель OSI с реальной структурой существующих стеков протоколов?

6. Функции каких уровней OSI-модели выполняет протокол TCP?
7. Что схожи и чем различаются протоколы UDP, TCP, SCTP и DCCP?
8. Приведите примеры протоколов (или целых стеков), которые нарушают канонические требования OSI-модели.
9. Покажите на примере Wireshark-скриншотов, какие поля в заголовках разных уровней и как именно соотносятся с функциями соответствующих уровней OSI-модели.
10. Каковы основные функциональные возможности программы Wireshark?
11. Какие существуют аналоги программы Wireshark?
12. Каким образом можно перенести (экспортировать) данные о пакетах в Wireshark-трассе в таблицу MS Excel?
13. Какие уровни OSI-модели “умеет“ анализировать программа Wireshark? Приведите конкретные примеры протоколов.
14. Используя шестнадцатеричное представление пакетов в окне Wireshark укажите в какой последовательности передаются байты в сеть на примере IP-адреса в заголовке пакета. Для ответа на этот вопрос см. самостоятельно темы https://ru.wikipedia.org/wiki/Порядок_байтов или https://en.wikipedia.org/wiki/Endianness.
15. Какими средствами можно отправить в сеть пакеты, записанные в файле-трассе pcap?
16. Сколько различных MAC- и IP-адресов можно закодировать, используя отведённое для них количество байт в заголовках?
17. Какой процент избыточности вносят заголовки разных уровней в передаваемые сообщения?
18. Какие поля в заголовках разных уровней вне поля данных можно использовать в целях стеганографии, т.е. для передачи скрытых (дополнительных) данных так, чтобы добавленные данные не мешали передаче пакетов. Какие условия должны при этом соблюдаться?
19. Что означает цветовая дифференциация пакетов в Wireshark?

Задание 4. Основы администрирования маршрутизируемых компьютерных сетей

4.1. Цель и краткая характеристика работы

Цель работы – изучение основных методов настройки маршрутизируемых компьютерных сетей на примере сети, состоящей из компьютеров под управлением ОС Linux.

В процессе выполнения работы изучается сетевой уровень модели OSI. Производится базовая настройка связности в сети, управление таблицами маршрутизации и правилами трансляции сетевых адресов. При помощи утилиты tcpdump выполняются наблюдения за передачей трафика по каналам связи в маршрутизируемой компьютерной сети. Применение утилиты tcpdump позволяет непосредственно в терминале (это основной метод управления сетевым оборудованием) наблюдать проходящие через интерфейсы компьютера пакеты и изучить их внутреннюю структуру.

В данной работе изучаются методы маршрутизации в сетях IPv4 и IPv6, а также широко распространенная в компьютерных сетях технология NAT.

Приблизительная трудоёмкость УИР №4 для выполнения:

- общей части задания – 6 астрономических часов;
- общей части задания и одного вариативного пункта – 9;
- общей части задания и двух вариативных пунктов – 11.

4.2. Теоретическая справка

Маршрутизация пакетов в каждом узле компьютерной сети представляет собой многоэтапный процесс. На каждом этапе происходят проверки различных условий, позволяющие определить дальнейшие действия, выполняемые с маршрутизируемым пакетом.

В Linux процесс маршрутизации разбивается на следующие этапы:

1. Фильтрация и начальная обработка поступающих в маршрутизатор пакетов.
2. Определение таблицы маршрутизации, в которой будет производиться поиск подходящего маршрута для пакета.
3. Поиск внутри таблицы маршрутизации и принятие решения о продвижении пакета.
4. Фильтрация и изменение маршрутизируемых пакетов с учетом информации о продвижении пакетов.

Рассмотрим подробнее каждый из этих этапов.

По умолчанию в Linux существует 3 таблицы маршрутизации: local, main и default. Для каждого пакета осуществляется поиск маршрута в каждой из таблиц маршрутизации по очереди до тех пор, пока не будет найден подходящий маршрут либо не будут перебраны все маршруты.

Таблица **local** содержит маршруты, принадлежащие сетевым интерфейсам данного маршрутизатора (например, для широковещательных адресов подключённых подсетей или для адресов, присвоенных интерфейсам маршрутизатора). Если пакет подходит под какой-либо маршрут из данной таблицы, значит, он предназначен для этого маршрутизатора. С пакета будет снят заголовок сетевого уровня, и он будет направлен на обработку дальше по стеку протоколов на транспортный уровень.

Таблица **main** содержит все основные маршруты, по которым выполняется маршрутизация на данном компьютере. Пакет, для которого найдено правило в этой таблице, будет перенаправлен в исходящий интерфейс маршрутизатора.

Таблица **default** используется для указания маршрутов для пакетов, которые не были обработаны ни в одной из предыдущих таблиц. Обычно данная таблица является пустой.

Если для пакета не будет найден маршрут ни в одной таблице маршрутизации, он будет отброшен. При этом, обычно на IP-адрес, указанный в поле source пакета, будет отправлено ICMP-сообщение об ошибке «Destination host unreachable».

Порядок проверки таблиц маршрутизации задается списком правил. По умолчанию данный список выглядит следующим образом (команда “ip rule list”):

```
0:      from all lookup local
32766:  from all lookup main
32767:  from all lookup default
```

До знака «:» указан индекс (порядковый номер правила). Все правила обрабатываются в порядке увеличения их номеров. После знака «:» указывается правило, по которому осуществляются действия для текущего пакета, над которым производятся маршрутизация. После слова «lookup» указывается таблица, в которой будет производиться поиск маршрута для текущего пакета.

В указанном примере правило «from all» означает, что под него подпадают пакеты с любым адресом источника. Для дополнительных

пользовательских правил рекомендуется использовать порядковые номера правил от 1 до 32765, чтобы не нарушать базовую логику обработки пакетов.

При прохождении пакета через маршрутизатор выполняются проверки пакета на различных этапах (маршрутизация, перемаркировка полей пакета, этап NAT) (рисунок 4.1). На каждом из этих этапов происходит проверка полей пакета на соответствие определенным правилам. В терминах Linux, этим этапам соответствуют следующие таблицы:

- **mangle:** изменение полей ToS, TTL в пакете, а также установка на пакет специальной числовой метки, которая может быть использована в других таблицах, но не передаётся за пределы текущего Linux;
- **nat**: Network Address Translation, т.е. изменение IP-адресов и UDP/TCP-портов в проходящем пакете;
- **filter:** фильтрация пакетов (основная функция фаервола/брандмауэра, позволяющая отбросить нежелательные пакеты для предотвращения атаки);
- **raw** и **security** (в целях упрощения изложения не рассматриваются).

В каждой из этих таблиц находятся цепочки правил, срабатывающие на различных стадиях прохождения пакета через маршрутизатор: на рисунке 4.1 имена цепочек обозначены заглавными буквами (PREROUTING, INPUT, FORWARD, OUTPUT, POSTROUTING). По умолчанию все таблицы в цепочках пустые, поэтому для выполнения нужных действий требуется явным образом их заполнять правилами.

В сценарии 1 приходящий из сети пакет адресован запущенной на текущем компьютере локальной программе, т.е. адрес назначения совпадает с одним из адресов интерфейсов компьютера. В сценарии 2 приходящий из сети пакет нужно переслать другому компьютеру. В сценарии 3 локальная программа на текущем компьютере передаёт пакет в сеть.

Совпадающие имена таблиц в цепочках не означают их эквивалентность или взаимозависимость. Например, если добавить новое правило в таблицу mangle в цепочке OUTPUT, то это правило не появится в таблицах mangle остальных цепочек. И наоборот: совпадающие имена цепочек означают их эквивалентность (чтобы подчеркнуть этот факт,

одноимённые цепочки графически объединены прямоугольниками серого цвета).

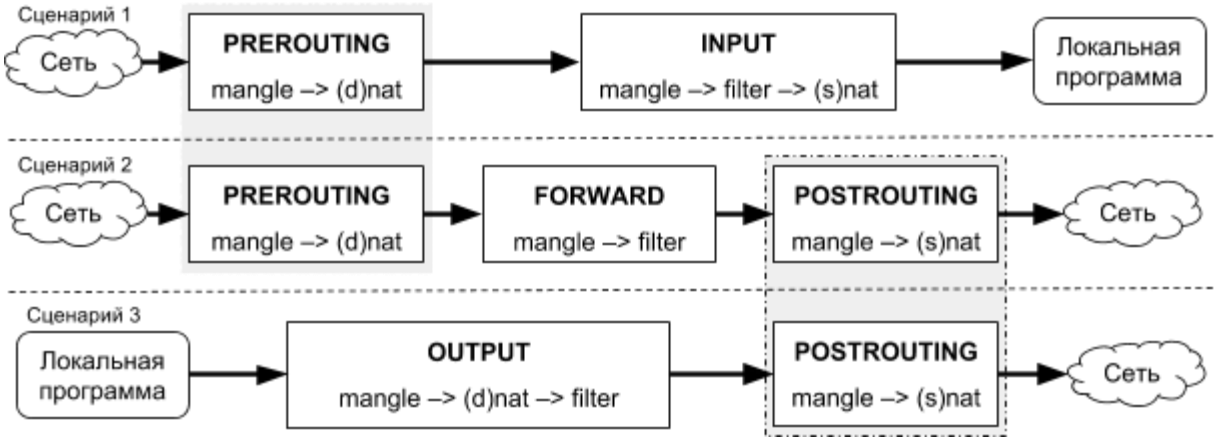


Рисунок 4.1. Упрощённая схема прохождения пакета по цепочкам таблиц межсетевого экрана Linux

Может показаться избыточным присутствие некоторых таблиц (например, mangle) сразу в нескольких цепочках. Эта избыточность позволяет гибко задать правила обработки пакета. Например: если добавить правило в таблицу mangle цепочки FORWARD, то оно будет применено только по отношению к ретранслируемым пакетам из сценария 2, но не к пакетам из сценариев 1 и 3. При этом добавление правила в одноимённую таблицу mangle в цепочке POSTROUTING приведёт к тому, что правило будет применено ко всем пакетам из сценариев 2 и 3, но не 1 (этот факт проиллюстрирован с помощью штрихпунктирной линии на рисунке 4.1). Заметим, что в каждом сценарии присутствует уникальная цепочка, которой нет в остальных. Это позволяет реализовать независимую логику обработки для любого типа пакетов каждого из сценариев.

Буква “**d**” в скобках перед таблицей nat означает, что в соответствующей цепочке в рамках технологии NAT имеет смысл подменять лишь адрес назначения (от англ. **d**estination). Это объясняется тем, что подменить адрес назначения необходимо до поиска маршрута передачи (название цепочки PREROUTING дословно переводится как “перед маршрутизацией”). Аналогично, буква “**s**” в скобках означает, что в соответствующей таблице nat следует подменять лишь адреса источника (от англ. **s**ource).

4.3. Этапы выполнения работы

Этап 1. Настройка виртуальных машин

Для выполнения УИР необходимо установить на компьютер бесплатно распространяемую программу Oracle VirtualBox, представляющую из себя среду виртуализации, позволяющую запускать виртуальные машины и соединять их в изолированные компьютерные сети. Скачать VirtualBox можно с официального сайта: https://www.virtualbox.org/wiki/Downloads .

В качестве ОС Linux в данной работе рекомендуется использовать “Ubuntu 14.04.5 Server”, которую можно скачать по ссылке: http://mirror.yandex.ru/ubuntu-releases/14.04.5/ubuntu-14.04.5-server-amd64.iso (или по следующей сокращённой ссылке: https://goo.gl/y3ueas).

При настройке каждой виртуальной машины достаточно выделить 400 МБ оперативной памяти и 8 ГБ дискового пространства.

Необходимо создать достаточное для выполнения варианта задания число виртуальных машин. Соединение между виртуальными машинами в соответствии с указанной в варианте топологией выполняется при помощи изолированных внутренних сетей (рисунок 4.2).

Если в соответствии с топологией виртуальные машины должны быть соединены, нужно указывать одинаковое имя сети для соответствующих сетевых адаптеров (на рисунке 4.2 указано имя внутренней сети “intnet”). Кроме того, у всех сетевых адаптеров в рамках одной локальной сети должны быть разные MAC-адреса.

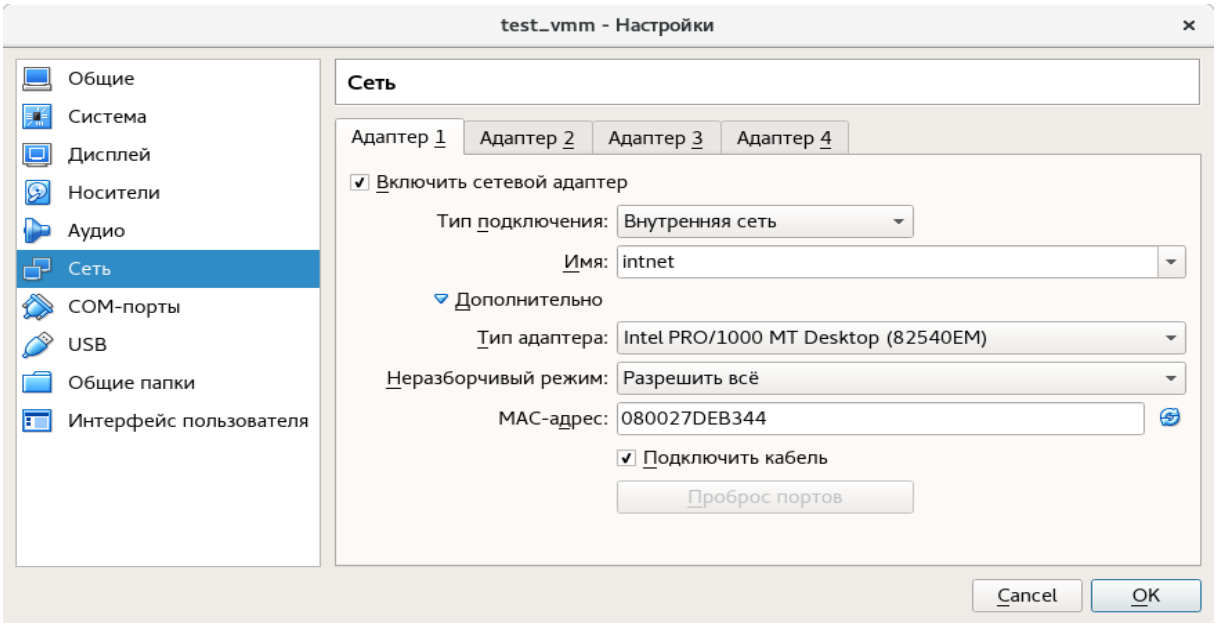


Рисунок 4.2. Настройка внутренней сети для сетевого адаптера

Этап 2. Ознакомление с используемыми командами Linux

Ниже приводятся и объясняются команды Linux, которые могут пригодиться для выполнения УИР. Здесь и далее строки, начинающиеся с «>», означают выполнение Linux-команды в командной строке, а «#» означает начало комментария, объясняющего смысл команды. Некоторые команды для выполнения требуют прав суперпользователя *root*.

> ip a *# вывод списка интерфейсов и адресов*
> ip link set <имя_интерфейса> up *# включить интерфейс*
> ip link set <имя_интерфейса> down *# выключить интерфейс*

Здесь и далее <имя_интерфейса> – это строка вида “eth*N*”, где *N* – натуральное число (включая 0). На рисунке 4.3 представлены интерфейсы с именами *eth0*, *eth1*, *eth2*.

Интерфейс с именем *lo* является локальной петлей, в данной работе он служит для назначения уникального IP-адреса для идентификации конкретного компьютера. Если у компьютера имеется несколько интерфейсов, то IP-адрес, заданный на локальной петле может служить универсальным адресом для обращения к данному компьютеру из различных подсетей.

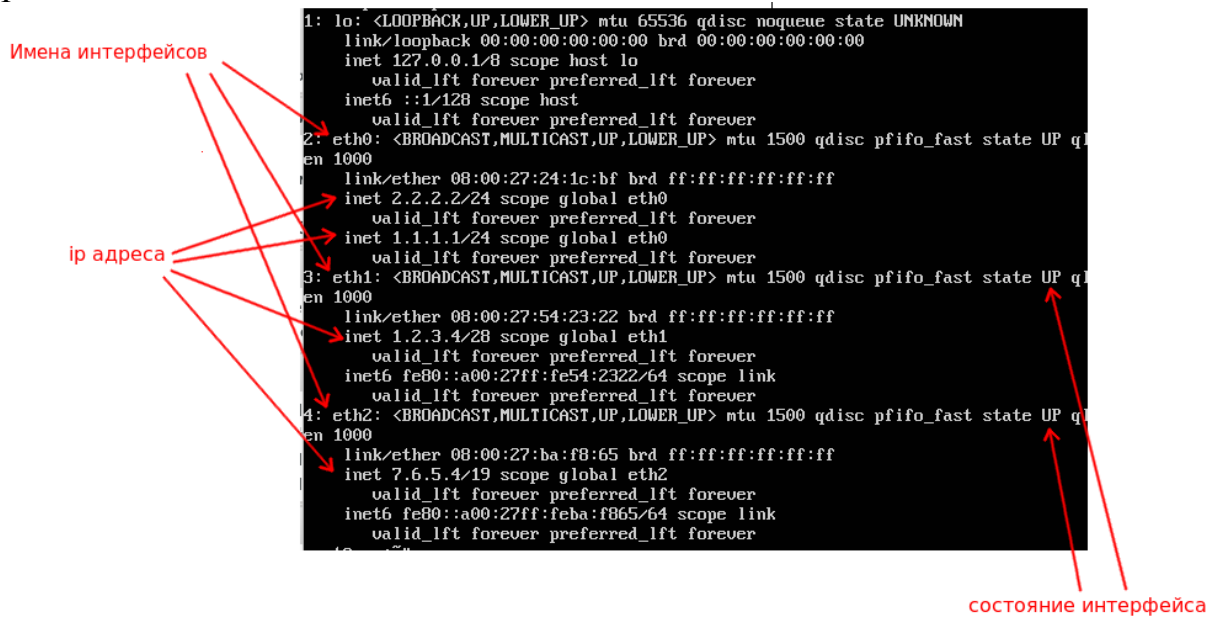


Рисунок 4.3. Пример вывода команды **ip a**

Следует обратить внимание на то, что для адресов, задаваемых на *lo*-интерфейс, необходимо указывать маску **255.255.255.255**.

> ip a add <ip-address/mask> dev <имя_интерфейса> *# назначить интерфейсу ip-адрес с указанной маской*

> ip a del <ip-address/mask> dev <имя_интерфейса> *# удалить из настроек интерфейса указанный ip-адрес*

> ip ro # вывод системной таблицы маршрутизации.

После присвоения ip-адреса интерфейсу в системной таблице маршрутов автоматически появляется маршрут к подключенной сети через этот интерфейс (рисунок 4.4).

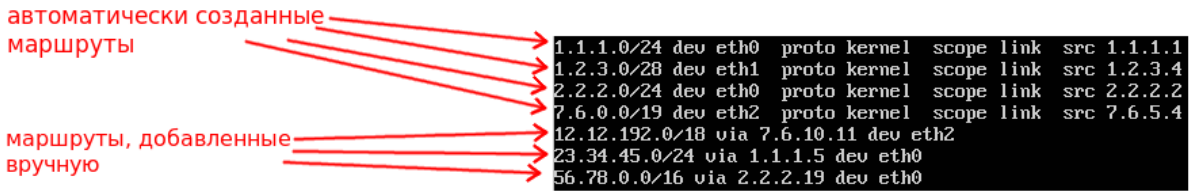


Рисунок 4.4. Пример вывода команды **ip ro**

> ip ro add <dest_ip/mask> via <gateway_ip> *# добавить маршрут к сети/хосту через шлюз*

> ip ro del <dest_ip/mask> via <gateway_ip> *# удалить маршрут*

Здесь *dest_ip/mask* означает адрес сети или хоста назначения с соответствующей маской подсети; *gateway_ip* – это шлюз, т.е. адрес, на который будут посланы пакеты, адрес назначения которых соответствует комбинации *dest_ip/mask.*

> ip ro add <dest_ip/mask> via <gateway_ip> table <table_number> # добавить маршрут в таблицу с номером *table_number*.

> ip ro list table <table_number> # вывод на экран таблицы маршрутизации с номером *table_number*.

> ip ro add <dest_ip/mask> nexthop via <gateway_ip_1> weight <weight_1> nexthop via <gateway_ip_2> weight <weight_2> nexthop via <gateway_ip_3> weight <weight_3> # задание маршрута с балансировкой трафика через различные шлюзы с указанием весов для маршрутов

Здесь *gateway_ip_1, gateway_ip_2, gateway_ip_3 – это* IP-адреса шлюзов; *weight_1, weight_2, weight_3* – веса для каждого из шлюзов: чем больше вес, тем больше вероятность выбора конкретного шлюза для пакета. В данном примере вероятность выбора *gateway_ip_1* будет равна

weight_1/(weight_1 + weight_2 + weight_3).

> ip rule # вывод правил выбора таблиц маршрутизации

> ip rule add prio <rule_number> from <src_ip/mask> lookup <table_number> # добавить правило, в соответствии с которым выбор таблицы маршрутизации для пакетов определяется адресом отправителя (такой подход называется PBR, policy-based routing – он отличается от обычной маршрутизации, которая использует только адрес назначения для принятия решения).

Здесь *rule_number* – номер правила в списке; *src_ip/mask* – адрес подсети источника пакета, *table_number* – номер таблицы маршрутизации, по которой будет осуществляться поиск маршрута пакетов из подсети *src_ip/mask*.

Для проверки соединений при помощи ICMP Echo Request для протокола IPv6 необходимо использовать команду **ping6.** Для указания адреса источника в ICMP Echo Request в команде **ping** можно указать параметр **-I**. Обязательным условием является наличие указанного в параметре **-I** адреса на компьютере. Например: **> ping -I 192.168.1.101 192.168.1.1**

Для всех команд **ip**, при настройке адресов и маршрутов IPv6, необходимо использовать параметр **-6**. Например:
> ip -6 a add 2000::1/64 dev eth0
> ip -6 ro add 2001::/64 via 2000::2

Далее следует серия команд для управления межсетевым экраном netfilter в ОС Linux с помощью утилиты iptables.

> iptables -nvL # вывести на экран все правила в цепочках таблицы фильтрации (filter)

> iptables -nvL -t nat # вывод правил в цепочках таблицы NAT

> iptables -nvL --line-numbers # вывод правил фильтрации с порядковыми номерами

> iptables -A <имя_цепочки> <спецификаторы_фильтра> -j <действие> # добавить правило в цепочку таблицы фильтрации

> iptables -t nat -A <имя_цепочки> <спецификаторы_фильтра> -j <действие> # добавить правило в таблицу NAT заданной цепочки

Здесь **<имя_цепочки>** для таблицы ***filter*** может принимать значения *INPUT, OUTPUT, FORWARD*, а для таблицы ***nat*** – *PREROUTING, INPUT, OUTPUT, POSTROUTING. Значение*

<спецификаторы_фильтра> задаёт набор набор правил, по которым проверяется пакет. А значение **<действие>** определяет, что нужно сделать с пакетом, который подошёл под правила фильтра.

>iptables -t nat -A PREROUTING -i eth1 -p tcp -d 10.1.1.1 -j DNAT --to-destination 192.168.0.2 # добавить правило, в соответствии с которым в TCP-пакетах, приходящих из сети на интерфейс eth1 и отправленных на IP-адрес 10.1.1.1, нужно заменить этот адрес на 192.168.0.2.

> iptables -D <имя_цепочки> <спецификаторы_фильтра> -j <действие> # удаление правила из цепочки таблицы фильтрации

> iptables -t nat -D <имя_цепочки> <спецификаторы_фильтра> -j <действие> # удаление правила из цепочки таблицы NAT

> iptables -D <имя_цепочки> <номер_правила> # удаление правила из цепочки по номеру

> tcpdump -ni <имя_интерфейса> # вывод на экран всех проходящих через интерфейс пакетов

> tcpdump -ni <имя_интерфейса> -w file.pcap # сбор дампа всех проходящих через интерфейс пакетов в файл **file.pcap**

Замечание 1. По умолчанию в Linux отключена маршрутизация пакетов, поэтому, чтобы включить возможность прохождения транзитом пакетов через компьютеры в сети, необходимо включить режим маршрутизации. Для IPv4 это можно сделать, выполнив следующие команды:

> sysctl -w net.ipv4.ip_forward=1

или

> echo 1 > /proc/sys/net/ipv4/ip_forward

Замечание 2. Некоторые виды сетевых атак основаны на подделке адреса отправителя в пакетах, поэтому по умолчанию в Linux отбрасываются те пакеты, в которых адрес отправителя недостижим через тот интерфейс, на который пришёл пакет. Для проверки достижимости адреса используется таблица маршрутизации. В некоторых разновидностях Linux это правило может смягчаться: пакет не отбрасывается, если адрес источника достижим через любой интерфейс компьютера (а не только через тот, на который пришёл пакет). Для выполнения некоторых вариантов задания требуется отключить эту защитную функцию Linux. Для IPv4 это можно сделать, выполнив следующие команды:

sysctl -w "net.ipv4.conf.all.rp_filter=0" или
echo 0 > /proc/sys/net/ipv4/conf/all/rp_filter

Этап 3. Выполнение общей части задания

Включить маршрутизацию на каждом компьютере в сети. Выполнить настройку сетевых интерфейсов всех компьютеров сети. Настроить таблицы маршрутизации таким образом, чтобы каждый компьютер мог осуществлять взаимодействие с любым другим компьютером. Топология сети должна при этом быть выбрана в соответствии с вариантом V1 (формула для расчёта V1 приведена в разделе 4.4), т.е. взаимодействие компьютеров должно осуществляться по соответствующим маршрутам этого варианта. Настроить простейшие правила фильтрации запрещённых пакетов. Проверку сетевой доступности следует осуществить при помощи утилиты **ping**, а корректность маршрутов – с помощью **tracert**.

Этап 4. Выполнение задания по варианту

Выполнение общей части задания позволяет получить оценку 3E. Для получения более высокой оценки (от 3D до 5A) необходимо выполнить дополнительное индивидуальное задание в соответствии с таблицей 4.1: формулы для расчёта V1 и V2 приведены в разделе 4.4. Необходимо обеспечить функционирование сети по схеме, описанной в варианте задания (см. ниже).

Таблица 4.1

Пункт задания	**Пункты задания, необходимые для получения оценки**				
	3E	**3D**	**4C**	**4B**	**5A**
Общая часть задания (для топологии из V1)	+	+	+	+	+
Вариант V1 с IPv4		+	+	+	+
Вариант V1 с IPv6			+	+	+
Вариант V2 с IPv4				+	+
Вариант V2 с IPv6					+

4.4. Порядок выполнения работы

1. Выбрать вариант для выполнения работы по формулам:
V1 = 1 + (N mod 5), V2 = 6 + (N mod 5),
где V1 и V2 – номера вариантов; N – сумма количества букв в фамилии и имени студента; mod – операция взятия остатка от деления.

2. На всех адаптерах всех компьютеров в топологии, представленной в варианте, настроить IPv4-адреса (и IPv6, если необходимо). IPv4-адрес выбирается следующим образом:

A.B.X.Y/M,

где A – количество букв в имени студента; B – количество букв в фамилии студента; X, Y – числа, выбираемые студентом самостоятельно; M – маска подсети (выбирается максимально длинная маска для обеспечения связности в сети). IPv6-адрес формируется из IPv4-адреса в соответствии с нотацией перевода адресов из IPv4 в IPv6. Например:

IPv4: 10.10.12.11

IPv6: 0:0:0:0:0:ffff:a0a:c0b (или иначе: ”::ffff:10.10.12.11”).

3. На всех компьютерах настроить таблицы маршрутизации таким образом, чтобы обеспечивалась полная сетевая доступность (каждый компьютер должен “пинговался” с каждого другого компьютера).

4. Изучить Linux-утилиту **nc** (или её аналоги: netcat, ncat, pnetcat). Запустить её в режиме клиента на машине А и в режиме сервера – на машине Б, используя для передачи произвольный порт (машины А и Б должны быть максимально удалены друг от друга). Передать в виде текстового сообщение свое имя от Б к А.

5. Изучить назначение Linux-утилиты iptables (например, тут: www.k-max.name/linux/iptables-v-primerax) и создать на компьютерах А и/или Б простейший Firewall (межсетевой экран) с помощью этой утилиты следующим образом:

- Запретить передачу только тех пакетов, которые отправлены на **TCP**-порт, заданный в настройках утилиты **nc**.
- Запретить приём только тех пакетов, которые отправлены с **UDP**-порта утилиты **nc**.
- Запретить передачу только тех пакетов, которые отправлены с IP-адреса компьютера А.

- Запретить приём только тех пакетов, которые отправлены на IP-адрес компьютера Б.
- Запретить приём и передачу **ICMP**-пакетов, размер которых превышает 1000 байт, а поле TTL при этом меньше 10.

6. Убедиться с помощью команды **ping, traceroute** или **nc**, что настроенные правила фильтрации iptables работоспособны, для чего нужно сначала попытаться передать запрещенный пакет, а затем разрешенный.

7. Выполнение пунктов с 1 по 6 позволяет получить оценку “3E”. Для получения более высокой оценки необходимо выполнить дополнительные задания в соответствии с вариантом V1 и V2 (см. таблицу 4.1). Дополнительные задания могут потребовать изменения настроек, сделанных на шагах со 2 по 6-й.

4.4.1 Вариант 1

На рисунке 4.5 изображена топология сети и требуемый путь прохождения сетевых пакетов. Необходимо так настроить хосты сети, чтобы ping-запрос (request) от компьютера 4 к компьютеру 2 шёл по пути, указанному сплошной стрелкой. При этом ping-ответ (reply) должен возвращаться другим путём: по пунктирной стрелке.

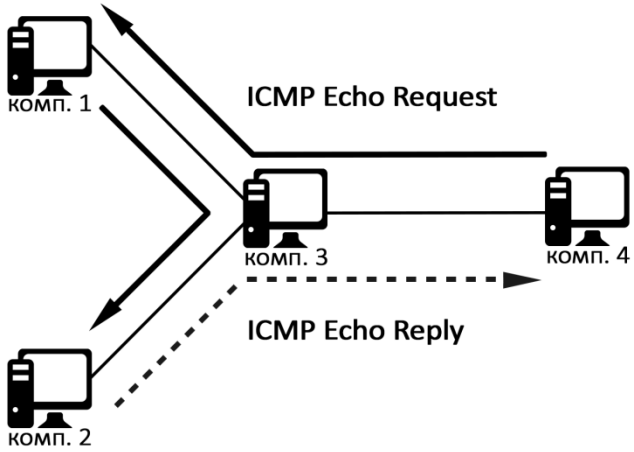


Рисунок 4.5. Топология сети и схема прохождения трафика для варианта 1

4.4.2 Вариант 2

На рисунке 4.6 изображена топология сети и один путь прохождения сетевых пакетов от **nc client 1** до **nc server**. На компьютере 4 должно быть запущено два nc-клиента, которые подключаются к различным nc-

серверам. Nc-серверы запущены на компьютерах 1 и 2, они должны принимать подключения на адреса интерфейсов, обозначенных на рисунке **nc server** и **nc server 2**.

На компьютерах должна быть настроена маршрутизация таким образом, чтобы запросы от **nc client 1** к **nc server** проходили через компьютер 2, запросы от **nc client 2** к **nc server 2** проходили через компьютер 1. Сплошными линиями на рисунке обозначен путь запроса от **nc client 1** к **nc server**, штриховыми — путь ответных пакетов от **nc server**. Nc-серверы и клиенты должны работать в режиме TCP.

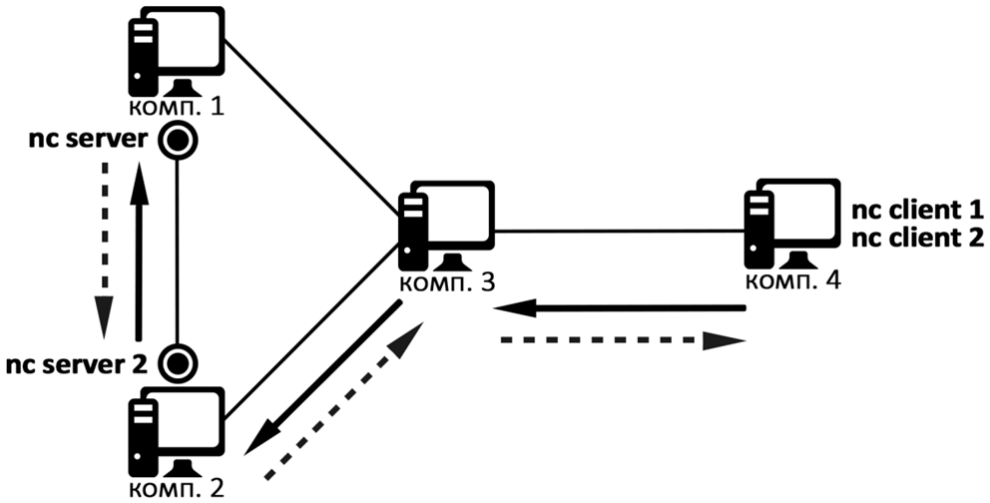


Рисунок 4.6. Топология сети и схема прохождения трафика для варианта 2

В отчёте в дополнение к общим пунктам (см. подраздел 4.5) нужно предоставить логи выполнения **nc**-серверов и клиентов, а также передаваемые сообщения.

4.4.3 Вариант 3

На рисунке 4.7 изображена топология сети и требуемый путь прохождения сетевых пакетов. Сплошными линиями показаны сообщения от клиента к серверам, штриховыми – сообщения от серверов к клиенту.

На компьютерах 1 и 2 запущены два nc-сервера. На компьютере 4 запущен один nc-клиент, который осуществляет подключение к IP-адресу, который не представлен ни на одном из интерфейсов в сети. На компьютере 3 должен быть настроен DNAT и дублирование трафика таким образом, чтобы оба nc-сервера получали запросы от nc-клиента. Ответные

сообщения от обоих nc-серверов должны приходить к nc-клиенту. Клиент и серверы должны работать по протоколу UDP.

В отчёте в дополнение к общим пунктам (см. подраздел 4.5) нужно предоставить текстовые сообщения с nc-серверов и клиента, которые были переданы и получены в ходе обмена данными, с комментариями выполняемых действий. Также следует предоставить объяснения того, как организовать приведенную схему взаимодействия с использованием протокола TCP.

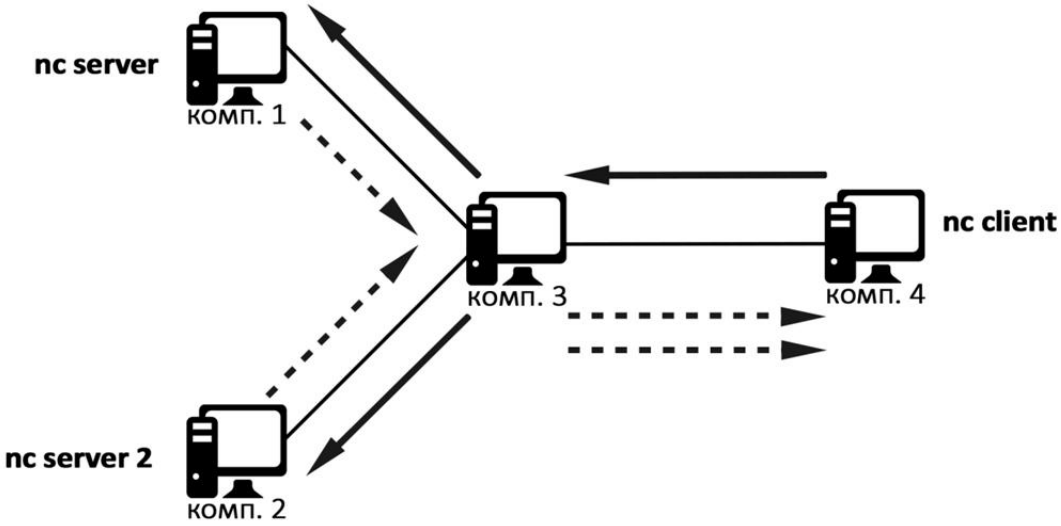


Рисунок 4.7. Топология сети и схема прохождения трафика для варианта 3

4.4.4 Вариант 4

На рисунке 4.8 изображена топология сети и требуемый путь прохождения сетевых пакетов. В компьютере 2 имеется два сетевых адаптера, на которых должны быть настроены различные IP адреса: IP1 и IP2. IP1 — адрес интерфейса, подключенного к компьютеру 1, IP2 — адрес интерфейса, подключенного к компьютеру 3.

Необходимо настроить сеть таким образом, чтобы при отправлении ICMP Echo Request с компьютера 4 на IP1 пакет проходил через компьютер 3 и компьютер 1, а ICMP Echo Reply шел с компьютера 2 сразу через компьютер 3 (штрих-пунктирная линия на рисунке 4.8). При отправлении ICMP Echo Request с компьютера 4 на IP2 пакет должен пройти через компьютер 3 сразу на компьютер 2, а ICMP Echo Reply должен пройти с компьютера 2 на компьютер 1 и далее на компьютер 3 (штриховая линия на рисунке 4.8). Требования к составу отчёта см. в подразделе 4.5.

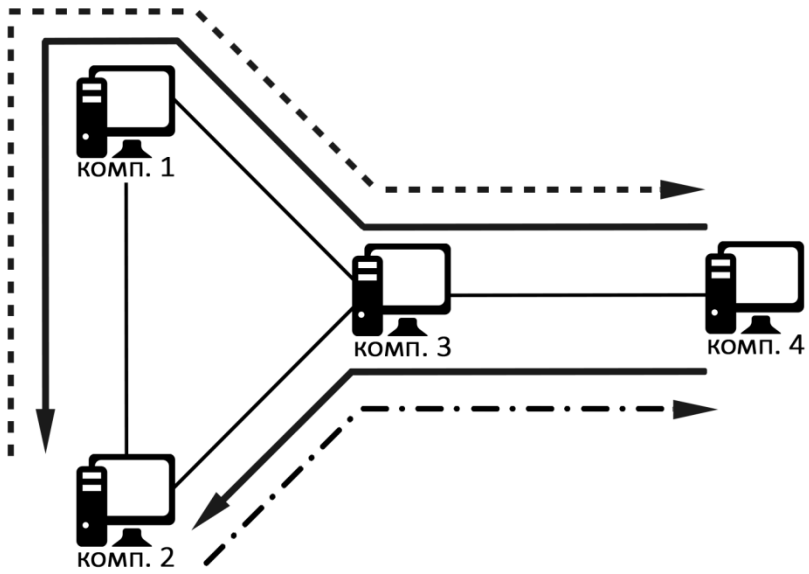


Рисунок 4.8. Топология сети и схема прохождения трафика для варианта 4

4.4.5 Вариант 5

На рисунке 4.9 изображена топология сети и требуемый путь прохождения сетевых пакетов. С компьютера 4 посылается ICMP Echo Request на адрес, который не существует в данной сети. На компьютерах 1, 2 и 3 должны быть настроены таблицы маршрутизации и правила NAT таким образом, чтобы пакет поочередно прошел через компьютеры 3, 2, 1 и снова пройдя через компьютер 3 пришел на компьютер 4 (сплошные линии на рисунке 4.4.5) с IP заголовком, в котором IP адрес источника и IP адрес назначения будут поменяны местами. Таким образом, компьютер 4 получит ICMP Echo Request на свой локальный адрес и ответит на него. ICMP Echo Reply должен пройти обратный путь (4->3->1->2->3->4) и прийти на компьютер 4 (штриховые линии на рисунке 4.9) с поменяными местами адресами источника и назначения. В результате выполнения команды **ping** должна быть выведена информация об успешном выполнении. Т.о. компьютер 4 сам отвечает на собственные ICMP запросы, однако пакет проходит через внешнюю сеть маршрутизаторов. Требования к составу отчёта см. в подразделе 4.5.

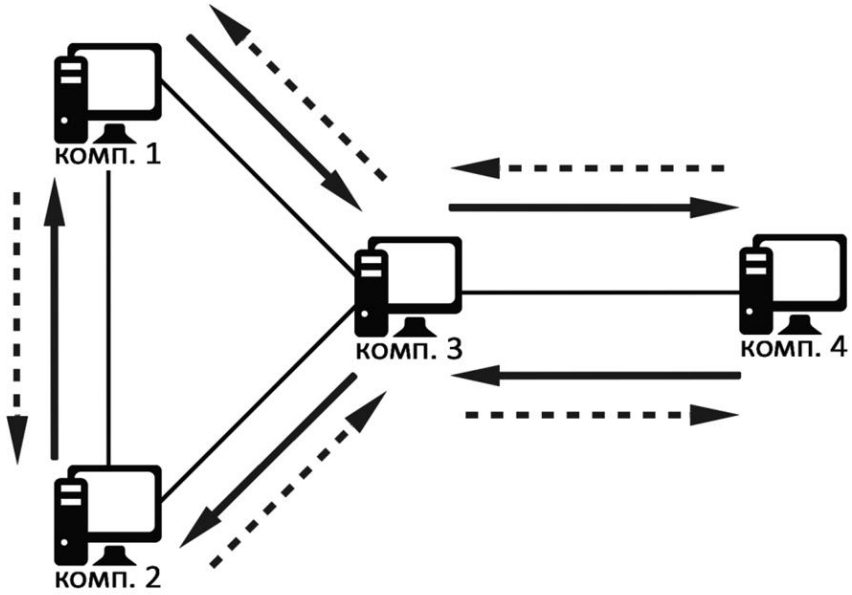


Рисунок 4.9. Топология сети и схема прохождения трафика для варианта 5

4.4.6 Вариант 6

На рисунке 4.10 изображена топология сети. В каждом компьютере для интерфейса *lo* должен быть задан уникальный IP-адрес. Нужно настроить таблицы маршрутизации на компьютерах сети таким образом, чтобы связь каждого компьютера с каждым другим осуществлялась через 3 канала компьютерной сети.

Связность должна быть обеспечена между IP адресами, указанными на *lo* интерфейсах, т.е. команда **ping** должна выполняться с указанием адреса источника.

Пример:

Компьютер 1: *lo* – 1.1.1.1/32
Компьютер 2: *lo* – 2.2.2.2/32
Компьютер 3: *lo* – 3.3.3.3/32
Компьютер 4: *lo* – 4.4.4.4/32

При выполнении команды **ping 1.1.1.1 -I 3.3.3.3** на компьютере 3 пакет пройдет по каналам между компьютерами 3 и 4, 4 и 2, 2 и 1 (штрих-пунктирные линии на рисунке 4.10). При выполнении команды **ping 2.2.2.2 -I 3.3.3.3** на компьютере 3 пакет пройдет по каналам между компьютерами 3 и 4, 4 и 1, 1 и 2 (штриховые линии на рисунке 4.10). При выполнении команды **ping 4.4.4.4 -I 3.3.3.3** на компьютере 3 пакет пройдет по каналам

между компьютерами 3 и 2, 2 и 1, 1 и 4 (сплошные линии на рисунке 4.10). Требования к составу отчёта см. в подразделе 4.5.

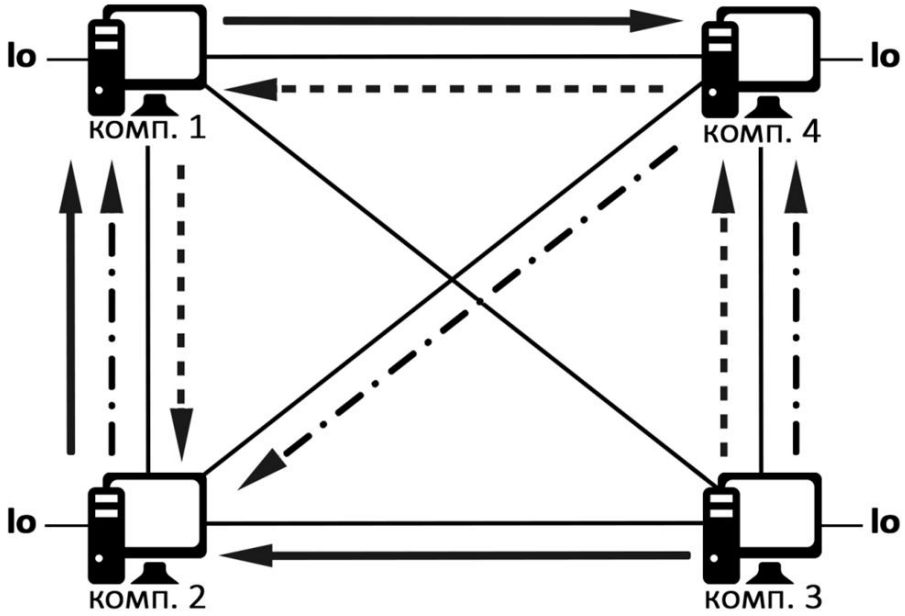


Рисунок 4.10. Топология сети и схема прохождения трафика для варианта 6

4.4.7 Вариант 7

На рисунке 4.11 изображена топология сети и один из путей прохождения сетевых пакетов. Между компьютерами 1 и 2 проведено 2 изолированных друг от друга канала. IP1 – адрес интерфейса в компьютере 1, подключенный к каналу 1. IP2 – адрес интерфейса в компьютере 2, подключенный к каналу 1.

Необходимо настроить таблицы маршрутизации и правила выбора таблиц маршрутизации таким образом, чтобы ICMP Echo Request, отправленный с компьютера 3 на IP1, прошел через компьютер 2 и канал 1 (сплошные стрелки на рисунке 4.11), а ICMP Echo Reply прошел через канал 2 в компьютер 2, после чего в отправился в компьютер 3 (штриховые стрелки на рисунке 4.11). ICMP Echo Request, отправленный с компьютера 3 на IP2 прошел через компьютер 1 и канал 1, а ICMP Echo Reply прошел через канал 2 в компьютер 1, после чего в отправился в компьютер 3.

В отчёте в дополнение к общим пунктам (см. подраздел 4.5) нужно предоставить доказательства того, что каналы 1 и 2 изолированы друг от друга (например, в виде скриншотов дампов трафика и распечатки логов тестирования).

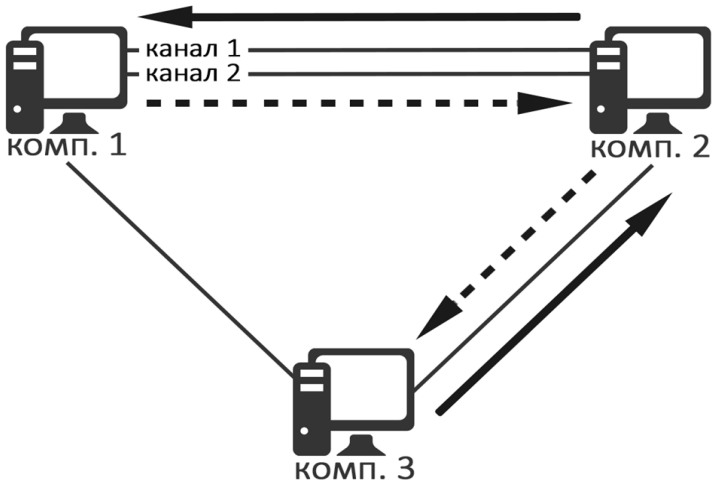


Рисунок 4.11. Топология сети и схема прохождения трафика для варианта 7

4.4.8 Вариант 8

На рисунке 4.12 изображена топология сети и требуемый путь прохождения сетевых пакетов. В каждом компьютере для интерфейса *lo* должен быть задан уникальный IP-адрес. На всех компьютерах необходимо настроить балансировку нагрузки при помощи задания маршрутов с несколькими шлюзами и весами для всех шлюзов. В каждом компьютере до любого другого компьютера должно быть задано 3 маршрута через разные шлюзы, имеющие различные веса. Веса для маршрутов во всех компьютерах необходимо выбрать таким образом, чтобы сумма весов маршрутов, проходящих через каждый канал, была одинакова.

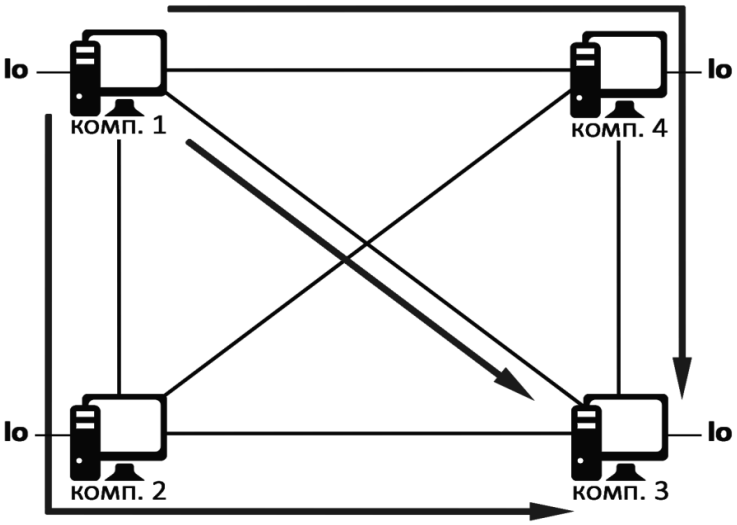


Рисунок 4.12. Топология сети и схема прохождения трафика для варианта 8

Связность должна быть обеспечена между IP адресами, указанными на *lo* интерфейсах, т.е. команда **ping** должна выполняться с указанием адреса источника. Требования к составу отчёта см. в подразделе 4.5.

4.4.9 Вариант 9

На сетевом интерфейсе компьютера 4 должен быть задан MTU = 1000. С компьютера 4 отправить ICMP Echo request в компьютер 1, размер которого больше заданного MTU, но меньше 2*MTU, т.о. пакет будет разделен на 2 фрагмента. На всех компьютерах настроить таблицы маршрутизации таким образом, чтобы второй фрагмент дошел до компьютера 1 раньше, чем первый фрагмент.

На рисунке 4.13 изображена топология сети и пример прохождения трафика по каналам сети. Первый фрагмент проходит через компьютеры 3 и 2 (штриховая стрелка на рисунке 4.13), а второй фрагмент только через компьютер 3 (сплошная стрелка).

В отчёте в дополнение к общим пунктам (см. подраздел 4.5) нужно предоставить дамп трафика, доказывающий факт получения компьютером 1 второго фрагмента раньше первого.

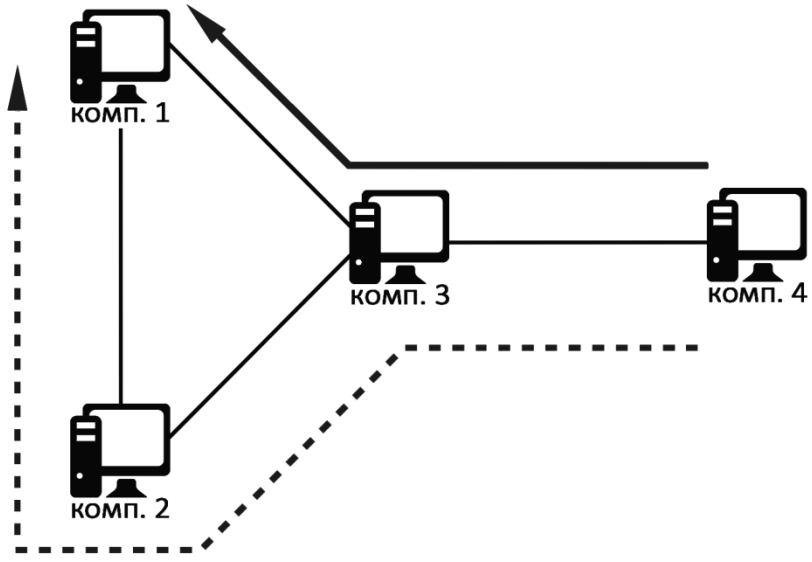


Рисунок 4.13. Топология сети и схема прохождения трафика для варианта 9

4.4.10 Вариант 10

На рисунке 4.14 изображена топология сети. С компьютера 5 посылается ICMP Echo Request на компьютер 3. Размер пакета и MTU сетевого интерфейса должны быть выбраны таким образом, чтобы пакет

был фрагментирован на 3 фрагмента. При этом компьютер 1 получит 3 пакета, которые должны быть направлены по 3 различным каналам.

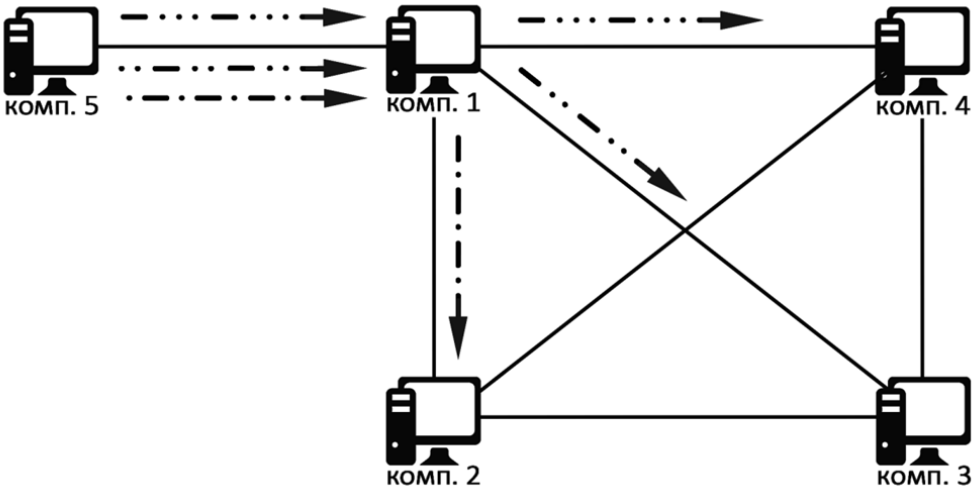


Рисунок 4.14. Топология сети и схема прохождения трафика для варианта 10

После получения всех фрагментов, в компьютере 3 формируется ICMP Echo Reply, который должен быть разделен на 2 фрагмента. Два фрагмента должны пройти различный по длине обратный путь к компьютеру 5. Требования к составу отчёта см. в подразделе 4.5.

4.5. Требования к содержанию отчёта

Отчёт предоставляется в электронном или бумажном виде. Для общего задания следует привести схему сети с указанием всех IP-адресов всех сетевых интерфейсов сети. Должны быть представлены таблицы маршрутизации списки выбора таблиц маршрутизации для всех компьютеров сети, а также перечень команд терминала, использованных для их настройки. Также должны быть представлены настройки iptables на каждом узле и соответствующие использованные команды.

Для заданий, указанных для каждого из вариантов, необходимо привести скриншоты, иллюстрирующие выполнение задания. Каждый скриншот должен иметь поясняющий текст, подробно раскрывающий ход выполнения задания и полученные результаты.

При защите отчёта необходимо иметь при себе сохраненную версию захваченного трафика на flash-носителе в формате pcap (так называемую трассу, или дамп, трафика), доказывающего факт прохождения или непрохождения пакетов в зависимости от задания.

4.6. Контрольные вопросы для самопроверки

При подготовке к защите задания следует руководствоваться следующим примерным перечнем вопросов и задач для самостоятельной проработки.

1. Протоколы каких уровней модели OSI используются при продвижении пакета через маршрутизируемую сеть?
2. Как соотносятся уровни модели OSI и уровни стека сетевых протоколов TCP/IP?
3. Какое максимальное количество узлов (хопов) может пройти пакет по маршрутизируемой сети?
4. Какое минимальное значение может иметь поле TTL при передаче пакета между компьютерами?
5. Приведите примеры инкапсуляции протоколов при передаче данных на сетевом уровне модели OSI.
6. Каким образом осуществляется идентификация удаленного хоста в TCP-соединении?
7. Какие проблемы могут возникнуть при наличии в маршрутизируемой сети двух устройств с одинаковыми MAC- или IP-адресами?
8. В чем заключаются основные причины создания протокола IPv6?
9. В чем заключаются основные преимущества протокола IPv4 перед IPv6?
10. Как осуществляется связь компьютеров, работающих по протоколу IPv6 через сеть, в которой используется только IPv4?
11. Кто отвечает за дефрагментацию пакетов в сети?
12. На какое максимальное количество фрагментов может быть фрагментирован пакет IPv4 или IPv6? Во сколько раз при этом суммарный размер всех фрагментов будет больше исходного пакета?
13. Для чего используется технологии NAT и NAPT на практике?
14. В чём разница между целями SNAT, DNAT и MASQUERADE, которые применяются после ключа “-j” в iptables?

Задание 5. Технологии QoS в компьютерных сетях

5.1. Цель и краткая характеристика работы

Цель работы – изучение эффективности приоритезации трафика для управления качеством обслуживания (Quality of Service, QoS) в компьютерных сетях.

В работе изучаются современные дисциплины обслуживания, которые применяются администраторами компьютерных сетей для управления качеством обслуживания (Quality of Service, QoS), предоставляемым различным видам трафика пользователей. В качестве метрик качества обслуживания принято использовать, например, среднюю задержку пакетов при прохождении сети, вариацию (джиттер) этой задержки, а также процент потерянных пакетов. Пропускную способность маршрута также относят к показателям QoS, но вследствие того, что она косвенно влияет на величину трёх уже перечисленных метрик, то её не всегда рассматривают в качестве самостоятельного показателя QoS.

Приблизительная трудоёмкость работы составляет семь астрономических часов для выполнения всех пунктов задания.

5.2. Теоретическая справка

Достаточно типичной является ситуация, когда на пути следования трафика по маршруту существует медленный сегмент сети, являющийся «узким местом» (bottleneck). В узком месте пакеты трафика различных пользователей скапливаются в очередях в буферах сетевых устройств, что влияет на все три вышеперечисленные метрики QoS. Решением проблемы может быть модернизация «узкого места» (установка более быстрого канала связи), однако это может оказаться дорогостоящей и не всегда возможной процедурой. Альтернативным решением является приоритезация трафика, когда системный администратор, управляющий «узким местом» сети, устанавливает наиболее важному трафику более высокий приоритет, чтобы улучшить значения метрик QoS этого трафика за счёт ухудшения метрик QoS низкоприоритетного трафика.

Второй подход является оправданным, например, когда конкурирующие виды трафика имеют различные требования к сети. В соответствии с международным стандартом «ITU-T Y.1541» для качественной передачи видеозвонка Skype по сети требуется обеспечить Skype-пакетам среднюю задержку передачи IP-пакетов менее 100 мс с вариацией не более 50 мс и с потерей не более 0.1% пакетов, тогда как для просмотра прямой телевизионной трансляции (так называемое “видео по запросу” – ВПЗ) соответствующие QoS-требования существенно более мягкие: 1000 мс на среднюю задержку при тех же 0.1% для потерь.

Примерами популярных дисциплин обслуживания, применяемых в компьютерных сетях, являются FIFO (дисциплина обслуживание в порядке поступления без приоритетов), PQ (Priority Queueing – дисциплина обслуживания с относительными приоритетами) и WFQ (Weighted Fair Queueing – взвешенное честное обслуживание).

При использовании FIFO можно ожидать, что все конкурирующие классы трафика получают одинаковое качество обслуживания. Это ожидание действительно может оправдаться, если показателем качества обслуживания считать среднюю задержку ожидания в буфере. Однако если качество обслуживания связывается с общей задержкой пакета, включающей время ожидания в буфере и время выдачи пакета в канал связи, то правило FIFO не обеспечит справедливое равное распределение этого сетевого ресурса.

При конфигурации PQ, администратор может установить различный приоритет нескольким видам трафика, при этом низкоприоритетные пакеты передаются, только если в очереди отсутствуют высокоприоритетные пакеты. Это позволяет обеспечить наилучшее качество обслуживания для высокоприоритетного потока, однако в условиях перегрузки практически полностью блокирует очередь низкоприоритетных пакетов: отсутствуют какие-либо гарантии, что диспетчер очереди переключится на обслуживание низкоприоритетных пакетов в течение конечного интервала времени.

При настройке WFQ у администратора появляется возможность установить в байтах условный вес *Wi* для каждого *i*-го из *n* классов конкурирующего трафика. За каждый цикл работы WFQ из очереди *i*-го класса передаются пакеты суммарным размером *Wi* байт. Значит, чем выше вес *Wi*, тем лучшее качество обслуживания может ожидать *i*-й класс. Это позволяет более гибко управлять QoS-характеристиками, чем при использовании PQ. Кроме того, заданный вес является гарантией того, что даже в условиях высокой загрузки соответствующая очередь будет получать доступ к каналу связи за конечное время.

Указанный эффект иллюстрируется на рисунке 5.1 в условиях высокой загрузки канала связи тремя классами трафика (здесь и далее используются некоторые рисунки из магистерской диссертации Поповой Д.А. “Исследование свойств дисциплины обслуживания WFQ в

компьютерных сетях", Университет ИТМО, 2016). На рисунке 5.1 видно, что при использовании FIFO распределение перегруженного канала между трафиком разных видов имеет хаотичную природу. При этом использование WFQ позволило нужным образом "поделить" канал связи между тремя видами трафика с помощью установки весов Wi нужным системному администратору образом.

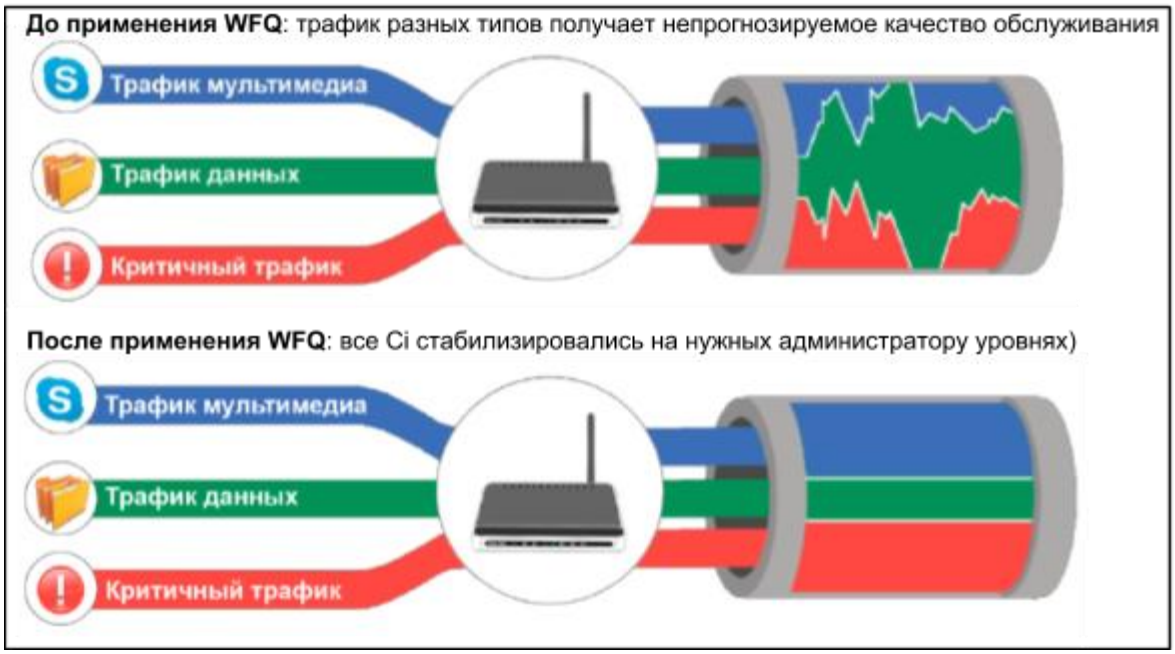


Рисунок 5.1. Иллюстрация эффекта от применения WFQ

5.3. Этапы выполнения работы и варианты заданий

Студент кафедры ВТ Университета ИТМО на каникулах собирается поехать на море и при этом планирует использовать планшет как для онлайн-трансляции видео о посещаемых достопримечательностях (т.е. ВПЗ), так и для одновременного с этими трансляциями разговора с родными и близкими с использованием Skype или любого другого подобного программного обеспе́чения, генерирующего потоковое видео реального времени. Тарифы на интернет-связь в роуминге на море достаточно высоки, поэтому студент хочет подобрать самый низкоскоростной тариф, который бы обеспечил качественный Skype-разговор при одновременном комфортном качестве онлайн-трансляции.

Полагая, что исходящий интернет-канал на планшете является узким местом при передаче трафика, студент решил настроить в сетевом драйвере планшета различные дисциплины обслуживания (ДО), которые обеспечат

различные характеристики качества передачи для Skype- и ВПЗ-трафика при соблюдении требований «ITU-T Y.1541». Требуется найти такую ДО, при которой требуемая скорость исходящего канала связи будет минимальной. По результатам экспериментов необходимо сравнить особенности исследованных ДО и выбрать оптимальную из них для поездки на море.

Для решения этой задачи предлагается использовать имитационную модель, предоставляемую преподавателем в виде alp-файла (это специализированный формат программы Anylogic). Существует альтернативный способ запустить имитационную модель в онлайн-режиме из браузера, имеющего поддержку технологии “Java applets”, по следующей ссылке www.runthemodel.com/models/2089/). Общий вид панели управления модели представлен на рисунке 5.2.

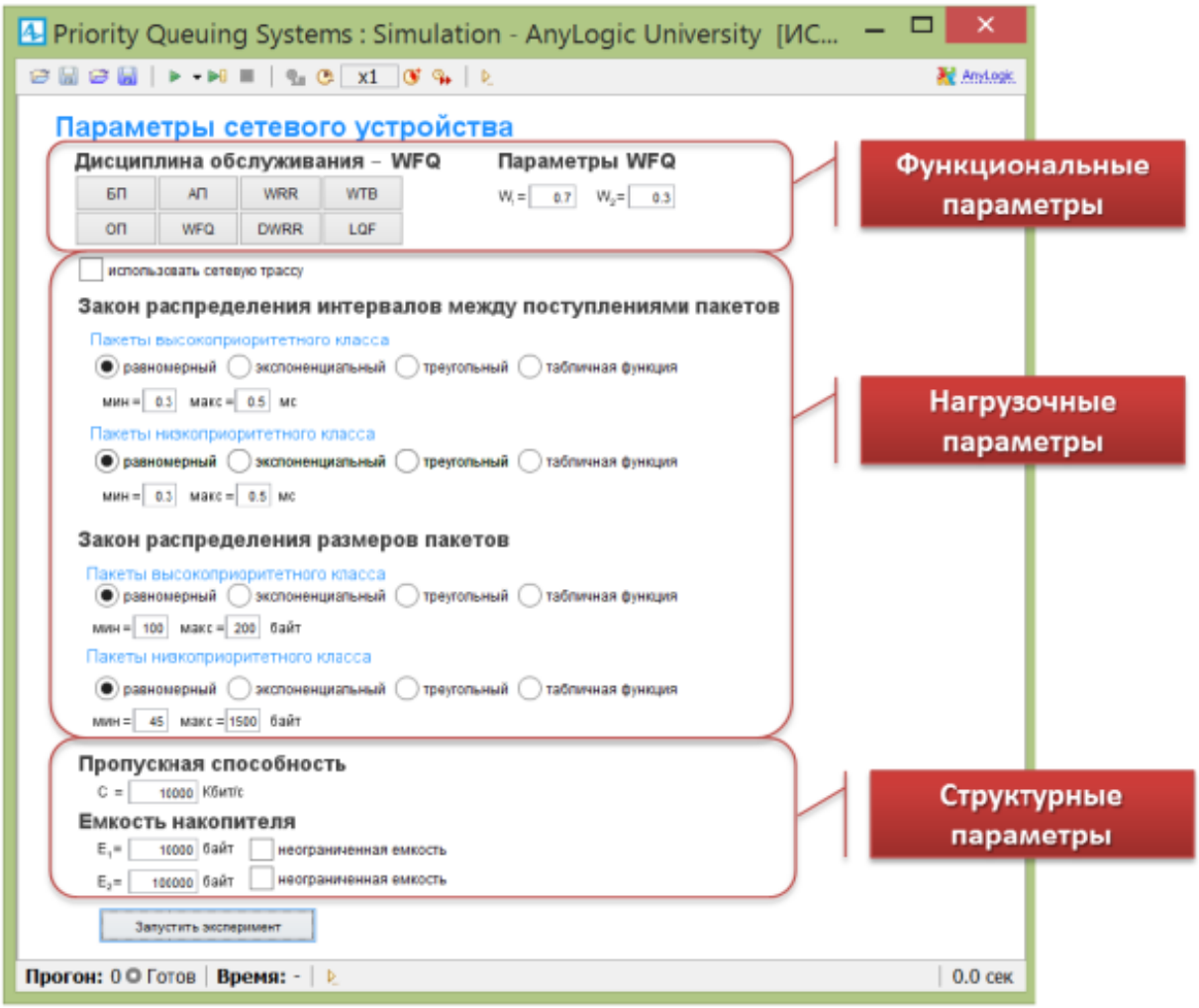


Рисунок 5.2. Панели управления с параметрами модели

Модель позволяет задать функциональные параметры в виде дисциплины обслуживания, а также нагрузочные параметры, описывающие особенности трафика, проходящего через канал связи. Важными структурными параметрами являются размеры буферов сетевого устройства. От вместительности этих буферов напрямую зависят метрики QoS.

В общем случае выбор дисциплины обслуживания зависит не только от скорости трафика и пропускной способности канала связи, но и от внутренней структуры трафика. Известно, что при равном битрейте нескольких потоков наименьшие задержки в буферах будут у того из них, в котором пакеты одинакового размера образуют регулярный поток, т.е. такой поток, в котором межпакетный интервал является константой. Трафик реальных приложений, к сожалению, отличается от этого оптимального варианта, поэтому в имитационной модели предусмотрена возможность указать произвольные законы распределений как для межпакетных интервалов, так и для размера пакетов.

Пример заданной по 16 точкам функции распределения межпакетного интервала приведён на рисунке 5.3. В xslx-файле на листах **Лист1** и **Лист2** в столбце «A» записаны значения функции распределения; в столбце «B» – соответствующие значения случайной величины (в данном случае это межпакетный интервал, выраженный в мс). В ячейке E1 должно быть записано количество точек столбца A, которое следует использовать для построения функции распределения.

Составленный таким образом xslx-файл необходимо загрузить в имитационную модель, предоставленную преподавателем. Аналогичным образом можно загрузить информацию о функции распределения размера пакетов, выраженного в байтах (два листа в xslx-документе соответствуют двум входящим потокам трафика в модель). В результате при запуске модели информация о соответствующих функциях распределения будет отображаться, как показано на рисунке 5.4, на котором интервалы времени между пакетами указаны в мс, размер пакетов указан в байтах.

E1 =COUNTA(A$1:A$1048576)

	A	B	C	D	E
1	0,00000	0,10500			16
2	0,00494	3,31930			
3	0,00511	6,53360			
4	0,01297	9,74790			
5	0,02078	12,96220			
6	0,02720	16,17650			
7	0,05461	19,39080			
8	0,99265	22,60510			
9	0,99610	25,81940			
10	0,99984	29,03370			
11	0,99988	32,24800			
12	0,99989	45,10520			
13	0,99990	54,74810			
14	0,99992	57,96240			
15	0,99998	61,17670			
16	1,00000	64,39100			
17					

Рисунок 5.3. Файл с поточечно заданной функцией распределения межпакетных интервалов

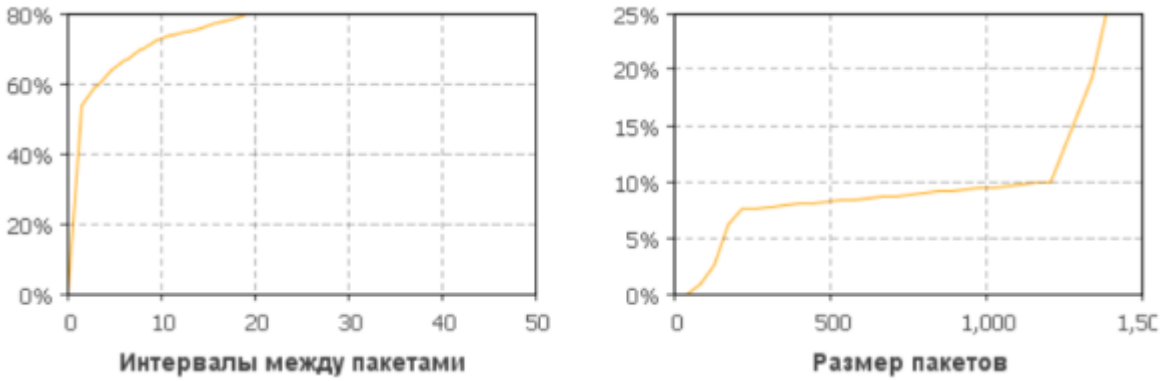


Рисунок 5.4. Графики функций распределения в модели Anylogic

Более подробное описание имитационной модели, а также подробные инструкции по её настройке и применению можно найти в магистерской диссертации Поповой Д.А. “Исследование свойств дисциплины обслуживания WFQ в компьютерных сетях” (Университет ИТМО, 2016), доступной для скачивания по следующей ссылке: https://isu.ifmo.ru/pls/apex/f?p=2143:0:108185710433474:DWNLD_F:NO::FILE:258C6D264DB9832BF14EEC8B82D95477
или используя её укороченный вариант: https://goo.gl/EC3JiW

5.4. Порядок выполнения работы

1. Установить бесплатную учебную версию системы имитационного моделирования AnyLogic Free PLE (не ниже версии 7.1.2), доступную для скачивания на сайте российской фирмы XjTek: http://www.anylogic.ru/downloads/.

2. Запустить в AnyLogic предоставленную преподавателем имитационную модель в alp-файле и разобраться в её работе (авторы модели – Попова Д.А. и Гомзина Т.К., выпускницы кафедры ВТ Университета ИТМО).
3. Установить размеры буферов, равными S килобайт, где S – количество букв в фамилии студента.
4. Установить скорость канала связи, равной N Мбит/с, где N – количество букв в имени студента.
5. Установить в модели законы распределения размера пакетов и межпакетного интервала для трафика каждого из двух типов следующим образом.
   5.1. Для получения оценки «удовлетворительно» можно установить любые законы распределения кроме детерминированных так, чтобы битовые скорости (bps) поступления каждого из двух типов трафика различались ровно в два раза.
   5.2. Для получения оценки «хорошо» нужно найти в Интернете сведения о приблизительных значениях битовой скорости каждого из двух типов трафика и установить любые законы распределения кроме детерминированных так, чтобы средний размер пакетов и межпакетный интервал соответствовали найденным значениям. В отчёте следует привести ссылку на использованный интернет-источник.
   5.3. Для получения оценки «отлично» нужно с помощью программы Wireshark или tcpdump записать трафик каждого из двух типов (минимум 10 000 пакетов). Затем экспортировать трафик каждого из двух исследуемых типов в CSV-файлы, используя Wireshark-фильтры. Найти в полученных файлах временны**е** метки поступления пакетов и размеры пакетов. Построить функции распределения межпакетных интервалов и размера пакетов и загрузить их в модель, как было показано в примере выше. Примеры сервисов для трансляции ВПЗ можно

найти по следующей ссылке: http://useiteasy.ru/internet/157/servisy-dlya-online-translyaciy.html.

6. Для дисциплины обслуживания FIFO (аналог бесприоритетной дисциплины обслуживания) провести эксперименты, в которых нужно постепенно уменьшать или увеличивать скорость канала связи до тех пора, пока не будет найдена минимальная скорость, при которой характеристики QoS каждого вида трафика всё ещё соответствуют нормам «ITU-T Y.1541». Для каждого использованного в экспериментах значения скорости канала связи нужно записывать полученные значения задержек и процента потерь, чтобы затем построить график по этим значениям.
7. Провести аналогичные предыдущему пункту эксперименты с дисциплиной обслуживания PQ, установив высокий приоритет более требовательному классу.
8. Провести аналогичные предыдущему пункту эксперименты с дисциплиной обслуживания WFQ, установив в K раза больший вес более требовательному классу, где $K = 2 + ((S + N) \bmod 7)$, где операция “$X \bmod Y$” означает “взять остаток от деления X на Y”. После того как будет найдена минимальная скорость, провести эксперименты с другими соотношениями весов и найти такое оптимальное соотношение весов, при котором скорость канала связи будет минимальной.
9. Написать отчёт о проделанной работе.

5.5. Требования к содержанию отчёта

Отчёт о выполнении УИР предоставляется студентом в бумажном или электронном виде (PDF) и должен иметь следующую структуру:

1. Титульный лист с названием вуза, ФИО студента и полного названия УИР.
2. Оглавление с указанием номеров страниц (номера страниц должны быть проставлены в колонтитулах каждой страницы).
3. Краткая постановка задач и цели исследования.
4. Обзор материалов, использованных для получения статистических данных о внутренней конфигурации исследуемых типов трафика (размеры пакетов и межпакетных интервалов). Если сбор статистики осуществлялся самостоятельно, нужно привести скриншоты Wireshark (включающие использованные фильтры) и построенные графики плотности и функции распределения. Если использовались внешние

источники, нужно привести соответствующий URL, а также пересказ основных тезисов.

5. Скриншоты имитационной модели, на которых должно быть видно использованные все параметры модели и полученные результаты экспериментов для лучшего и худшего результатов по каждой дисциплине обслуживания.
6. Скриншоты Wireshark (tcpdump), на которых должны быть видны IP-адреса, порты, протоколы и использованные фильтры (для оценки “отлично”).
7. Расчёт битовой скорости поступления (bps) каждого из типов трафика, полученных с использованием указанных на скриншотах параметров.
8. Графики изменения средней задержки и процента потерь каждого из двух классов трафика, которое происходило при варьировании пропускной способности канала связи в ходе экспериментов с различными дисциплинами обслуживания (значения пропускной способности следует указать на оси абсцисс).
9. Подробные выводы с анализом каждого из приведённых графиков с результатами.
10. Общие выводы по УИР.

5.6. Контрольные вопросы для самопроверки

При подготовке к защите УИР рекомендуется использовать следующий перечень вопросов и заданий для самостоятельной проработки и подготовки к защите.

1. Перечислите метрики, используемые для оценки качества передачи IP-пакетов по компьютерной сети?
2. Как измеряется джиттер (вариация задержки) передаваемых пакетов? Как учитываются потерянные пакеты при измерении джиттера и средней задержки?
3. В чём отличие джиттера от RTT?
4. Приведите примеры программ, аналогичных по функционалу программе Wireshark.
5. Какие существуют классы качества обслуживания в соответствии с документом ITU-T Y.1540? Чем различаются эти классы?
6. Приведите несколько примеров приложений, предъявляющих различные требования к качеству передачи данных по сети?

7. Приведите примеры специализированных сред имитационного моделирования компьютерных сетей.
8. Для чего в использованной в исследовании имитационной модели рассчитывается доверительный интервал?
9. Можно ли уменьшить джиттер задержки передачи с помощью искусственного увеличения средней задержки передачи?
10. Является ли стационарным процесс, протекающий в моделируемой системе? Изменяются ли законы распределения размера пакетов и межпакетный интервал со временем?
11. Как зависит размер IP-пакетов, генерируемых приложениями, от типа приложения и предоставляемых им сетевых сервисов?
12. Приведите примеры методов QoS, применяемых в IP-сетях, с их краткой характеристикой.
13. Проанализируйте достоинства и недостатки двух предложенных методов устранения узкого места в сети.
14. Какие дисциплины обслуживания сетевого трафика доступны в виде штатных средств в операционных системах Linux, Windows?
15. Какова ёмкость буферной памяти в современных маршрутизаторах и коммутаторах?
16. Как ёмкость буферной памяти, отводимой на хранение передаваемых пакетов в промежуточных узлах компьютерной сети, влияет на метрики QoS?
17. Как работает дисциплина обслуживания FQ_CODEL в Linux? Каковы её основные характеристики?
18. Как работает дисциплина обслуживания HTB в Linux? Каковы её основные характеристики?
19. Как работает дисциплина обслуживания TBF в Linux? Каковы её основные характеристики?
20. Как работает алгоритм WRED? Для чего он применяется?
21. Чем отличается дисциплина обслуживания pfifo от pfifio_fast в операционной системе Linux?

Раздел 2. Тесты

В этом разделе приведены вопросы для подготовки к тестированию по нескольким темам. В конце раздела в пункте 6 можно найти ответы на вопросы и пояснения к некоторым из них.

1. Общие вопросы и OSI-модель

В данном параграфе приведены вопросы по следующим темам: OSI-модель, виды сетевого оборудования, терминология компьютерных сетей.

1.1. Какие элементы включает в себя канал связи?

a. Линия связи.
b. Сетевой шлюз (Gateway).
c. Сетевой мост (Bridge).
d. Каналообразующее оборудование.
e. Маршрутизатор.
f. Протокольный стек.

1.2. Как называется процесс объединения нескольких входящих в узел потоков данных в один выходящий из узла поток?

a. Демультиплексирование.
b. Демультипликатирование.
c. Коммутирование.
d. Коммутация.
e. Мультиплексирование.
f. Перколяция.

1.3. Какие виды сетей описываются аббревиатурой WAN?

a. Сенсорная вычислительная сеть.
b. Домашняя вычислительная сеть.
c. Персональная вычислительная сеть.
d. Виртуальная вычислительная сеть.
e. Локальная вычислительная сеть.
f. Глобальная вычислительная сеть.

1.4. Как называется компьютерная сеть, которая используется для объединения телефонов, карманных ПК, смартфонов?

a. MAN.
b. PAN.
c. LAN.
d. GAN.
e. WAN.

f. SAN.

1.5. Какая из перечисленных технологий используется наиболее часто для организации сетей MAN?

a. Zigbee.
b. Ethernet.
c. ATM.
d. WiMAX.
e. Bluetooth.
f. MPLS.

1.6. В какой полосе частот передаются данные в каналах тональной частоты?

a. от 300 Гц до 3400 Гц.
b. от 10 кГц до 20 кГц.
c. от 0 кГц до 100 кГц.
d. от 0 кГц до 20000 кГц.
e. от 300 кГц до 20000 кГц.
f. от 10 кГц до 2000 кГц.

1.7. Сохранение работоспособности при изменении структуры вычислительной сети в результате выхода из строя отдельных компонентов или при замене оборудования называется ... ?

a. ...гибкостью.
b. ...открытостью.
c. ...эффективностью.
d. ...адекватностью.
e. ...прозрачностью.
f. ...масштабируемостью.

1.8. Укажите корректное сопоставление номера уровня OSI-модели его названию.

a. Прикладной – L6.
b. Канальный – L2.
c. Транспортный – L3.
d. Уровень представления – L1.
e. Сетевой – L4
f. Физический – L7

1.9. На какие подуровни разбивается в IEEE-модели канальный уровень?

a. LLC.
b. ATM.
c. BER.

d. UDP.
e. UTP.
f. MAC.
g. STP.

1.10. Как называется совокупность правил, регламентирующих формат и процедуры взаимодействия процессов одноимённых уровней OSI-модели?

a. Стек.
b. Физическое кодирование.
c. Интерфейс.
d. Логическое кодирование.
e. Протокол.
f. Скремблироание.
g. Бит-стафинг.

1.11. Как уровни OSI-модели называются низшими?

a. Физический.
b. Прикладной.
c. Сетевой.
d. Уровень представления.
e. Транспортный.
f. Сессионный.
g. Канальный.

1.12. Как называется протокольный блок данных (PDU), передаваемый на канальном уровне TCP/IP-модели?

a. Пакет.
b. Кадр.
c. Сегмент.
d. Датаграмма.
e. Сокет.
f. Блок.

1.13. Что из представленного является корректным MAC-адресом?

a. C0-4A-00-58-C1-32
b. 01-AB-CD-EF-GH-10
c. C4-AA-BB-CC-DG-EF
d. 00-01-05-95-91-90-00
e. 01-00-BB-CC-DD-EF
f. 01-AA-BB-CC-DD

1.14. Какие уровни описывает модель TCP/IP?

a. Физический.
b. Канальный.

c. Сетевой.
d. Транспортный.
e. Сеансовый.
f. Прикладной

2. Сетевые топологии и методы коммутации

В данном параграфе приведены вопросы по следующим темам: характеристики сигналов и линий связи (спектр сигнала, полоса пропускания), виды модуляции, виды топологий и др.

2.1. Сколько каналов связи требуется для построения компьютерной сети, состоящей из *n* узлов (при использовании указанных топологий)?

a. Полносвязная топология: $n(n-1)/2$
b. Общая шина: $n(n-1)$
c. Звезда: $n(n+1)/2$
d. Кольцо: n
e. Дерево: $n-1$
f. Кольцо: $n(n-1)/2$
g. Звезда: $n(n-1)$

2.2. Сеть с топологией "Кольцо" состоит из *n* компьютеров. Из какого числа хопов в среднем состоит маршрут доставки сообщений в такой сети, если пакеты могут двигаться только в одном направлении, а все компьютеры одинаково часто взаимодействуют с другими абонентами сети?

a. n*2
b. n+1
c. n-1
d. n/2
e. n(n-1)/2
f. n*n

2.3. Какой способ коммутации использовался в традиционных (аналоговых) телефонных сетях?

a. Коммутация пакетов.
b. Коммутация каналов.
c. Коммутация сообщений.
d. Коммутация ячеек.
e. Коммутация линий.
f. Коммутация маршрутов.

2.4. Укажите верные утверждения, касающиеся сопоставления физической и логической топологии сети.

a. Логическая топология сети определяется только структурой связи узлов.
b. Физическая топология сети определяется только последовательностью передачи данных между узлами.
c. Физическая топология сети "Кольцо" может совпадать с физической топологией "Полносвязная".
d. Физическая топология полностью определяется структурой связи узлов.
e. На основе полносвязной физической топологии можно реализовать любую логическую топологию.
f. Физическая топология всегда отличается от логической.

2.5. Укажите верные утверждения, касающиеся сопоставления сетей с различными видами коммутации при условии, что пропускная способность каналов связи в этих сетях идентична.

a. При коммутации каналов затраты буферной памяти в промежуточных узлах сети меньшие, чем при любых других способах коммутации.
b. Время доставки сообщений максимально при коммутации каналов (по сравнению с другими методами коммутации).
c. При коммутации пакетов показатели надёжности доставки сообщения выше, чем при коммутации сообщений.
d. При коммутации ячеек накладные расходы в виде передаваемых служебных данных меньше, чем при коммутации сообщений.
e. При коммутации сообщений не требуется наличие буферной памяти в транзитных узлах для хранения передаваемых данных.
f. При коммутации пакетов все каналы связи должны иметь одинаковую пропускную способность на всём маршруте передачи.

2.6. За счёт чего время доставки сообщений при коммутации пакетов меньше, чем при коммутации сообщений?

a. Сокращение затрат буферной памяти при передаче пакетов позволяет увеличить процент потерь.
b. Разные сообщения передаются параллельно по разным каналам.

c. Разные пакеты одного и того же сообщения передаются последовательно по одному и тому же каналу.
d. Пропускная способность при передаче пакетов выше, чем при передаче сообщений.
e. При коммутации сообщений меньше задержки в узлах связи.
f. Разные пакеты одного и того же сообщения передаются параллельно по разным каналам связи.

2.7. На чём основан метод маршрутизации по предыдущему опыту?

a. Маршрутизатор ассоциирует адрес отправителя в транзитном пакете с номером интерфейса (порта), через который пакет поступил в маршрутизатор.
b. Изменение маршрутной таблицы зависит от состояний выходных буферов данного маршрутизатора и не зависит от состояния соседних узлов.
c. Изменение маршрутной таблицы зависит от состояний соседних узлов (маршрутизаторов).
d. Магистральный маршрутизатор на границе сети централизованно устанавливает таблицы маршрутизации на основе предыдущего успешного сеанса работы.
e. Маршрутизатор, выполняющий роль шлюза, координирует заполнение таблиц маршрутизации в подчинённых (slave) маршрутизаторах.
f. Отправитель сообщения размещает в пакете всю цепочку адресов промежуточных маршрутизаторов.

2.8. С какой целью применяется процедура бит-стаффинга в протоколе HDLC?

a. Чтобы при физическом кодировании бит были исключены длинные последовательности нулей и единиц.
b. Для увеличения пропускной способности канала связи за счёт замены повторяющихся последовательностей бит.
c. Для повышения надёжности передачи за счёт дублирования передаваемых бит.
d. Чтобы обеспечить корректный расчёт контрольной суммы кадра (CRC-16) за счёт дополнения длины кадра до числа, кратного 16.
e. Чтобы исключить появление в пользовательских данных контрольной последовательности, используемой для разделения кадров.
f. Для обеспечения корректности расчёта бита чётности, добавляемого в сообщение для обнаружения ошибок передачи.

2.9. В каких единицах измерения принято указывать пропускную способность канала связи?

a. Бод/с
b. Кибибит/с
c. Бит/с
d. Герц/с
e. Байт/с
f. 1/с
g. Децибел/с

2.10. Какие типы сигналов используются в компьютерных сетях для передачи данных?

a. Электрические.
b. Акустические.
c. Гравитационные
d. Электромагнитные.
e. Оптические.
f. Инерционные.
g. Магнитные.

2.11. Укажите верные утверждения, касающиеся сравнения различных режимов двунаправленной передачи данных.

a. В дуплексном канале связи возможна передача от приёмника к передатчику и обратно в один и тот же момент времени.
b. В полудуплексном канале связи возможна как передача от приёмника к передатчику, так и обратно, но лишь в режиме разделения времени.
c. В симплексном канале связи возможна передача данных только в одном направлении.
d. Пример симплексного канала связи – спутниковое цифровое телевидение.
e. Полудуплексные каналы связи никогда не применялись в компьютерных сетях.

2.12. Укажите верные утверждения, касающиеся измерения изменения мощности сигнала при передаче данных.

a. При усилении сигнала в 10 раз изменение сигнала составляет +10 дБ.
b. При уменьшении сигнала в 2 раза изменение сигнала составляет -2 дБ.
c. При уменьшении сигнала в 100 раз изменение сигнала составляет -20 дБ.
d. При усилении сигнала в 100 раз изменение сигнала составляет +2 дБ.

e. При усилении сигнала в 1000 раз изменение сигнала составляет -30 дБ.
f. При усилении сигнала в 2 раза изменение сигнала составляет +1 дБ.

2.13. Во сколько раз уменьшится мощность сигнала на расстоянии 2000 м, если его ослабление равно 10 дБ/км?

2.14. Укажите верные утверждения, касающиеся частотных характеристик сигнала и каналов связи.

a. Невозможно передать по каналу связи сигнал, спектр которого уже полосы пропускания канала связи.
b. Для корректной передачи сигнала ширина полосы пропускания канала связи должна быть не меньше ширины спектра сигнала.
c. Полоса пропускания канала связи измеряется в Герцах.
d. Полоса пропускания зависит от физических свойств проводника, по которому происходит передача.
e. Спектр сигнала представляет из себя амплитудно-частотную характеристику канала связи, по которому передаётся сигнал.
f. Спектр сигнала измеряется в Герцах.

2.15. Рассчитать по формуле Шеннона-Хартли максимально возможную пропускную способность канала связи при условии, что полоса пропускания равна 100 МГц, а мощность сигнала равна мощности шума?

2.16. Каким образом модуляция применяется для передачи данных по каналу связи?

a. Модем преобразует цифровой сигнал в последовательность модуляций прямоугольных импульсов с максимальной амплитудой.
b. Модулятор использует N различных гармоник несущего сигнала, имеющих близкую частоту, для кодирования передачи N различных уровне цифрового сигнала.
c. При кодировании сигнала модулируется спектр сигнала с учётом состава гармоник для отображения передаваемого сообщения на частоты гармоник.
d. Передатчик представляет символы передаваемого сообщения в виде сигналов разной амплитуды, частоты или фазы несущей.
e. Для передачи двоичного кода полоса пропускания модулирует значения 0 и 1 в виде идентичных гармоник.

2.17. Укажите верные утверждения, касающиеся процессов квантования и дискретизации сигналов.

a. При квантовании по уровню каждое измеренное значение сигнала заменяется на ближайшее к нему значение уровня (число таких уровней фиксировано и заранее известно).
b. При дискретизации сигнал измеряется не непрерывно, а через фиксированные промежутки времени.
c. Можно выполнить либо квантование по уровню, либо дискретизацию сигнала, но не то и другое одновременно.
d. Частота квантования по уровню должна минимум в два раза превосходить частоту любой из гармоник сигнала.
e. Период дискретизации должен быть хотя бы в два раз меньше любого из периодов гармоник сигнала.

2.18. Какая минимальная пропускная способность необходима для передачи речевого сигнала, закодированного с помощью импульсно-кодовой модуляции, если число уровней квантования равно *N*, а частота дискретизации равна *H* кГц?

2.19. Укажите верные окончания следующей фразы: “При использовании метода логического кодирования 8B/10B по сравнению с 4B/5B …”.

a. ...передаётся больше избыточных данных (в процентах).
b. …существует больше запрещённых комбинаций.
c. …размер таблицы кодирования больше в 32 раза.
d. …размер таблицы кодирования меньше в 2 раза.
e. …невозможно применять метод кодирования NRZ .

2.20. Какие методы мультиплексирования используются в телекоммуникационных сетях?

a. Волновое.
b. Временно́е.
c. Амплитудное.
d. Фазовое.
e. Частотное.
f. Триплетное.

3. Технологии физического уровня

В данном параграфе приведены вопросы по следующим темам: методы физического и логического кодирования, методы доступа к общей среде передачи (CSMA/CD, FDMA, маркерный доступ.

3.1. Какие достоинства присущи волоконно-оптическим кабелям по сравнению с витой парой?

a. Меньшая стоимость сетевых устройств.
b. Более высокая пропускная способность
c. Отсутствие электромагнитного излучения.
d. Простота монтажа при обрыве кабеля.
e. Меньший вес кабеля.
f. Высокое электрическое сопротивление, обеспечивающее гальваническую развязку.

3.2. Какой порядок величины имеет диаметр световодной жилы многомодового оптического волокна?

a. Единицы зептометров (10^{-21} м).
b. Единицы аттометров (10^{-18} м).
c. Единицы фемтометров (10^{-15} м).
d. Единицы пикометров (10^{-12} м).
e. Единицы нанометров (10^{-9} м).
f. Единицы микрометров (10^{-6} м).
g. Единицы миллиметров (10^{-3} м).

3.3. Укажите верные утверждения, касающиеся беспроводных технологий передачи данных.

a. С повышением частоты электромагнитного поля излучения (ЭПИ) понижается проницаемость ионизированного слоя атмосферы.
b. Огибание электромагнитной волной зданий, деревьев и других объектов называется дифракцией.
c. С уменьшением частоты ЭПИ явление дифракции проявляется в меньшей мере.
d. Радиус действия компьютерных сетей, использующих инфракрасное излучение для передачи данных, составляет несколько километров.
e. Круговая экваториальная орбита движения спутника с радиусом обращения 12 часов называется геостационарной.
f. Вследствие рефракции радиоволн в атмосфере они распространяются не прямолинейно, а по дуге.
g. Радиорелейные линии связи используют принцип ретрансляции для передачи данных.

3.4. Какие протоколы канального уровня используются для выделенных линий связи (точка-точка)?

a. CSMA/CD
b. HDLC
c. TCP

d. CSMA/CA
e. ICMP
f. PPP

3.5. Укажите верные утверждения, касающиеся спутниковых систем связи.

a. Экваториальная синхронная орбита с периодом обращения 24 часа называется геостационарной.
b. VSAT – это спутниковый терминал с диаметром антенны более трёх метров.
c. Перигеем называется форма орбиты в виде эллипса.
d. Геостационарный спутник расположен на высоте менее 40 километров.
e. Высокоэллиптическая орбита спутника не позволяет обеспечить радиосвязь в высоких широтах.
f. Апогей – наиболее удалённая от Земли точка орбиты.

3.6. Укажите сетевые технологии, в которых для передачи данных в Интернет используются традиционные проводные телефонные сети общего назначения.

a. FDDI.
b. SONET.
c. Token Ring.
d. NRZ.
e. ISDN.
f. ADSL.

3.7. Укажите верные утверждения, касающиеся методов физического и логического кодирования.

a. манчестерское кодирование применяется в 10Base-T.
b. методы 4B/5B, NRZI и MLT-3 применяются в Fast Ethernet.
c. методы 8B/10B, PAM-5, NRZ применяются в 1 GigE.
d. методы 64B/66B, PAM-16 применяется в 10 GbE.
e. метод 64B/66B применяется в 100 Gigabit Ethernet.

3.8. Укажите методы логического кодирования (в отличие от методов физического кодирования).

a. 4B/5B.
b. Скремблирование.
c. RZ.
d. MLT-3.
e. NRZ.
f. 64B/66B.

3.9. Что означает термин “основополосная передача” (аналогичный термин в английском языке – baseband)?

a. Передача данных нескольких радиоканалов в единой полосе частот с помощью технологии разделения времени.
b. Передача цифрового сигнала непосредственно в линию связи без модуляции несущей.
c. Передача сигнала, при которой низкие гармоники сигнала передаются в основной полосе частот канала связи, а высокие гармоники сдвигаются в нижнюю часть спектра.
d. Передача с применением физического кодирования, исключающая применение логического кодирования.
e. Совместная передача данных нескольких радиоканалов с помощью технологии разделения частот.

3.10. Какие из представленных технологий используют физическую топологию “Кольцо”?

a. WiMAX.
b. LTE.
c. TokenRing.
d. FDDI.
e. WiFi.
f. Ethernent.

3.11. Укажите верные утверждения, касающиеся семейства технологий Ethernet (стандарт 802.3).

a. Скорость передачи данных в технологии 10Base5 составляет 5 Мбит/с.
b. В Ethernet для передачи применяется витая пара, оптоволокно, коаксиальный кабель и радиоканал.
c. В технологии 1000BASE-FX используется витая пара.
d. В технологиях 10Base-T, 100Base-T, 1000Base-T максимальная длина кабеля до коммутатора составляет 100 м.
e. Длина оптоволоконного кабеля в технологии Ethernet может составлять несколько километров.
f. Скорость передачи данных в технологии 10GBASE-CX4 составляет 10 Гбит/с.
g. Межкадровый интервал во всём семействе технологий Ethernet составляет 96 нс.

3.12. Укажите корректное значение англоязычных терминов, применяемых в сфере сетевых технологий.

a. Hub – коммутатор в глобальной сети.
b. Router – маршрутизатор.
c. Frame – кадр, являющийся PDU канального уровня.

d. Packet – преамбула в начале блока данных.
e. Switch – концентратор в локальной сети.
f. Token – маркер в сети FDDI.
g. Datagram – коллизия в Ethernet-сегменте.
h. Hop – метод скремблирования без потерь.

3.13. Укажите, какие из перечисленных технологий являются беспроводными.

a. Ethernet.
b. LTE.
c. Bluetooth.
d. WiFi.
e. WiMax.
f. HSPA.
g. IrDA.
h. FDDI.

4. Беспроводные сети

В данном параграфе приведены вопросы по следующим темам: беспроводные компьютерные сети, включая методы кодирования и технологии, которые применяются в беспроводных сетях.

4.1. Укажите верные утверждения, касающиеся мобильной телефонии.

a. Все поколения мобильной телефонии являются цифровыми за исключение аналогового поколения 1G.
b. Скорость передачи в сетях 4G может составлять от 0.5 до 10 Гбит/с в зависимости от мощности радиосигнала.
c. LTE и WiMAX обычно относят к поколению 4G.
d. Скорость передачи в сетях 3G составляет от 1 до 100 Мбит/с в зависимости от мощности радиосигнала.
e. При кодовом разделении канала связи (CDMA) одновременно осуществляется разделение по времени (TDMA) и частоте (FDMA).
f. Скорость передачи в сетях 2G не превышает 20 кбит/с.

4.2. В чем суть технологии OFDM?

a. Несколько битовых потоков объединяются в один поток, который передается на заданной частоте.
b. Частота несущей меняется случайным образом на основе псевдослучайной последовательности.
c. Каждый "единичный" бит заменяется двоичной последовательностью из N бит, а каждый "нулевой" бит

кодируется инверсным значением расширяющей последовательности.

d. Несколько потоков объединяются на основе одной несущей.
e. Каждый узел использует некоторую расширяющую последовательность, которая позволяет выделить данные из суммарного сигнала.
f. Битовый поток разделяется на подпотоки, каждый из которых модулируется своей несущей частотой.

4.3. При передаче данных через один канал связи, каждый узел сети использует собственную расширяющую последовательность, которая выбирается так, чтобы принимающий узел мог выделить данные из суммарного сигнала. В какой технологии используется этот принцип?

a. CDMA.
b. OFDM.
c. FHSS.
d. DSSS.
e. CSMA.
f. UGRS.
g. OFOM.

4.4. При передаче данных частота несущей меняется случайным образом на основе псевдослучайной последовательности. В какой технологии используется этот принцип?

a. OFDM.
b. FHSS.
c. CDMA.
d. DSSS.
e. CSMA.
f. UGRS.
g. OFOM.

4.5. Каждый "единичный" бит заменяется двоичной последовательностью из N бит, а каждый "нулевой" бит кодируется инверсным значением расширяющей последовательности. В какой технологии используется этот принцип?

a. OFDM.
b. FHSS.
c. CDMA.
d. DSSS.
e. CSMA.
f. UGRS.
g. OFOM.

4.6. Перечислите особенности технологии Bluetooth (IEEE 802.15.1).

a. Применяется метод расширения спектра FHSS.
b. В одной пикосети одновременно взаимодействовать могут не более 8 устройств.
c. Спектр передаваемых сигналов лежит в районе 2.4 МГц.
d. Возможна скорость передачи более 20 Мбит/с.
e. Область покрытия от 0 м до 1000 м.
f. Для передачи применяется экранированная витая пара.
g. Используется метод доступа CSMA.

5. Модель и стек протоколов TCP/IP

В данном параграфе приведены вопросы по следующим темам: стандартные протоколы, входящие в модель TCP/IP, их особенности и назначение

5.1. Что из перечисленного не является корректным IPv4-адресом?

a. 192.168.1.256
b. 145.0.0.1
c. 125.14.14.14
d. 199.255.255.2
e. 5.6.7.8
f. 13.0.0.13

5.2. Укажите верные утверждения, касающиеся протокола IP.

a. Длина IP-адреса может составлять 4 или 16 байт.
b. Минимальный размер IPv4-заголовка равен 20 байт.
c. Максимальный размер IPv4-заголовка равен 127 байт.
d. Максимальный размер IPv4-пакета равен 65535 байт.
e. Максимальное число маршрутизаторов на пути IP-пакета равно $(2^{32}–1)$
f. В заголовке IPv4 используется контрольная сумма, а в IPv6 – нет.

5.3. Чему равно максимальное число хостов (компьютеров) в сети с CIDR-маской 255.255.255.0?

5.4. Какие из перечисленных адресов являются “серыми”? Пояснение: “серые” адреса используются только в локальных сетях и не обрабатываются маршрутизаторами для отправки пакетов в Интернет при использовании технологии NAT.

a. от 10.0.0.0 до 10.255.255.255.
b. от 172.16.0.0 до 172.31.255.255 .
c. от 100.0.0.0 до 100.255.255.255.

d. от 172.0.0.0 до 172.255.255.255.
e. от 192.168.1.0 до 162.168.1.255
f. от 172.16.1.0 до 182.16.1.255
g. от 192.168.0.0 до 192.168.255.255.

5.5. Что такое “ширина окна” в протоколе TCP?

a. Максимальный размер положительной квитанции.
b. Максимальное количество байт, которое может быть передано без получения подтверждения.
c. Минимальное количество пакетов, которое может быть получено без отправки подтверждения
d. Минимальное количество байт, которое может быть передано без получения подтверждения.
e. Максимальное количество пакетов, которое может быть получено без отправки подтверждения.
f. Минимальный размер положительной квитанции.

5.6. Укажите корректные адреса подсетей при использовании бесклассовой адресации (CIDR) с соответствующими масками.

a. 172.17.0.0/9.
b. 172.19.3.0/22.
c. 172.31.237.0/19.
d. 172.22.0.0/18.
e. 172.25.8.8/30.
f. 172.17.0.192/28.

5.7. Укажите верные утверждения, касающиеся протокола из стека TCP/IP.

a. Протокол DHCP используется для автоматизации назначения IP-адресов для компьютеров сети.
b. Протокол ARP позволяет установить соответствие между IP- и MAC-адресом компьютера.
c. OSPF используется для автоматического построения таблиц маршрутизации.
d. RTP используется для передачи трафика реального времени.
e. В отличие от протокола TCP, протокол UDP не может контролировать скорость передачи данных и отправлять подтверждения о получении пакетов.
f. DNS используется для определения IP-адреса устройства по его известному символьному адресу (имени).

5.8. Какие адреса из представленных ниже являются корректной однозначной записью IPv6-адреса в соответствии с правилами RFC-5952?

a. 16:AX::BG:23
b. 16:17:18:19:20:215:FF
c. 00-03-24-56-16-44--01
d. 16:A104::BB:23
e. 44:ED:39:64:0:55:1:1
f. ::1
g. 00-A3-24-BB-16-AA
h. IP:V6:12:26:44:36
i. 78:B1:17FE:AB18:19:20:215:FF:44EB
j. 143A:7654:AC4F:1AF2:66AE:D6CC:44E9:980B
k. ABAB::673A:78::FF10:E1CB
l. 44:ED:39:64::55:1:1

5.9. Укажите метрики качества обслуживания (Quality of Service, QoS), используемые на уровне протокола IP (сетевой уровень L3).

a. Мощность радиосигнала.
b. Задержка передачи пакета между двумя точками маршрута.
c. Доля потерянных пакетов.
d. Отношение сигнал/шум в канале связи (SNR).
e. Скорость передачи данных (goodput).
f. Вариация задержки передачи (джиттер).

6. Ответы

N	**Ответ на вопрос номер *N* из указанного блока вопросов**				
	Блок 1	**Блок 2**	**Блок 3**	**Блок 4**	**Блок 5**
1	a, d	a, d, e	b, c, e, f	a, c, f	a*)
2	e	d*)	f	f	a, b, d, f*)
3	f	b	b, d, f, g	a	254*)
4	b	c, d, e*)	b, f	b	a, b, g
5	d	a, c	a, f	d	b
6	a	f	e, f	a, b, d*)	d, e, f*)
7	a	a	все		все
8	b	e	a, b, f		d, e, f, j*)
9	a, f	c*)	b		b, c, e, f
10	e	a, d, g	c, d		
11	a, c, g	a, b, c, d	d, e, f		
12	b	a, c	b, c, f		
13	a, e*)	в 100 раз	b, c, d, e, f, g		
14	b, c, d, f	b, c, d, f			
15		100 Мбит/с			
16		d			
17		a, b, e			
18		$1000*H*\lceil \log_2 N \rceil$ бит/с ***)**			
19		b, c			
20		a, b, e			

***) Пояснения к ответам:**

1.13. Корректный MAC-адрес должен состоять из шести байт, каждый из которых записывается шестнадцатеричным числом от 00 до FF.

2.2. В силу симметрии условий задачи достаточно рассмотреть среднюю длину маршрута для одного любого компьютера. Кратчайший маршрут состоит из 1 хопа, самый длинный маршрут – из (n-1) хопа. Тогда суммарную длину *S* всех возможных маршрутов можно рассчитать как сумму арифметической прогрессии от 1 до (n-1). Зная, что всего существует (n-1) различных маршрутов, среднюю длину маршрута можно рассчитать как S/(n-1).

2.4. Топология “Кольцо” совпадает с “Полносвязной” при наличии трёх узлов; в “Полносвязной” топологии можно реализовать любую последовательность передачи от узла к узлу, что позволяет реализовать любую логическую топологию.

2.9. Пропускную способность канала связи не стоит путать с его полосой пропускания, измеряемой в Герцах. Варианты b и e содержат корректные единицы измерения пропускной способности, однако они не используются на практике (в отличие от варианта c).

2.18. Операция, обозначенная квадратными полускобками, означает округление до ближайшего большего целого.

4.6. Полоса пропускания Bluetooth лежит в районе 2.4 ГГц, область покрытия – до 100 м, скорость передачи, начиная с Bluetooth 3.0, может быть равна или превышать 24 Мбит/с.

5.1. Каждый из четырёх элементов в записи IPv4-адреса должен быть целым числом от 0 до 255 (включительно). Для первого байта IPv4-адреса не допускается использовать значения 0 и 255.

5.2. Максимальный размер пакета IPv4-заголовка равен 60 байт, максимальное число маршрутизаторов на пути IP-пакета равно 255.

5.3. Из 32-х бит приведённой маски восемь являются нулевыми, т.е. отведёнными для нумерации хостов внутри сети. Значит, всего существует $2^8 = 256$ адресов внутри сети, но два из этих адресов (0 и 255) имеют специальное служебное назначение и не могут использоваться для адресации хостов, поэтому ответ на задачу есть $256 – 2 = 254$.

5.6. При маске, заданной в виде "/*n*", адрес подсети считается корректным, если в нём последние (32-*n*) бита являются нулевыми при $0 \le n \le 30$. При n=31 нулевое значение последнего бита адреса может означать как адрес подсети, так и адрес одного из двух хостов.

5.8. Корректная запись IPv6-адреса подчиняется следующим правилам: 1) восемь двухбайтовых групп (ДГ) при записи адреса разделяются двоеточиями, 2) при записи шестнадцатеричных цифр *не* используются прописные буквы, 3) незначащие старшие нули внутри ДГ не пишутся, 4) наиболее длинная последовательность из подряд идущих нулевых ДГ заменяется на "::" (если таких длинных последовательностей окажется несколько, то заменяется только первая слева), однако одиночная нулевая ДГ в окружении ненулевых ДГ на "::" не заменяется.

Список рекомендуемой литературы

1. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы. Учебник. – 5-е изд. – Питер, 2016. – 992 с.
2. Таненбаум Э., Уэзеролл Д. Компьютерные сети. – 5-е изд. – Питер, 2016. – 960 с.
3. Куроуз Д., Росс К. Компьютерные сети. Нисходящий подход. – 6-е изд. – Эксмо, 2016. – 912 с.
4. Стивенс У. Протоколы TCP/IP. Практическое руководство. – БХВ-Петербург, 2003. – 672 с.
5. Алиев Т.И. Сети ЭВМ и телекоммуникации. - СПб: СПбГУ ИТМО, 2011. - 400 с.

КАФЕДРА ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

Кафедра вычислительной техники Университета ИТМО создана в 1937 году и является одной из старейших и авторитетнейших научно-педагогических школ России. Первоначально кафедра называлась кафедрой математических и счетно-решающих приборов и устройств и занималась разработкой электромеханических вычислительных устройств и приборов управления. Своё нынешнее название кафедра получила в 1963 году.

Кафедра вычислительной техники является одной из крупнейших в университете, на которой работают высококвалифицированные специалисты, в том числе 7 профессоров и 14 доцентов, обучающие более 500 студентов и аспирантов. На кафедре ведется подготовка бакалавров и магистров по направлениям подготовки «Информатика и вычислительная техника» и «Программная инженерия».

Кафедра имеет 5 компьютерных классов, объединяющих более 80 компьютеров в локальную вычислительную сеть кафедры и обеспечивающих доступ студентов ко всем информационным ресурсам кафедры и выход в Интернет. Кроме того, на кафедре имеются учебные и научно-исследовательские лаборатории по вычислительной технике, в которых работают студенты, магистранты и аспиранты.

Тауфик Измайлович Алиев,
Владимир Валерьевич Соснин,
Дмитрий Николаевич Шинкарук

КОМПЬЮТЕРНЫЕ СЕТИ И ТЕЛЕКОММУНИКАЦИИ: ЗАДАНИЯ И ТЕСТЫ

Учебно-методическое пособие

В авторской редакции
Редакционно-издательский отдел Университета ИТМО
Зав. РИО _____ Н.Ф. Гусарова
Подписано к печати _____
Заказ № _____
Тираж _____
Отпечатано на ризографе