Министерство образования и науки Российской Федерации

Южно-Российский государственный
политехнический университет (НПИ) имени М.И.Платова

С.П. Воробьёв

КОМПЬЮТЕРНЫЕ СЕТИ

Учебно-методическое пособие
к выполнению лабораторных работ

Новочеркасск
ЮРГПУ(НПИ)
2017

УДК 004.023 (О76.5)

Рецензент – **Д.В. Гринченков,** канд. техн. наук, декан факультета информационных технологий и управления, заведующий кафедрой «Программное обеспечение вычислительной техники» ЮРГПУ(НПИ)

Воробьев С.П

Компьютерные сети: учебно-методическое пособие к выполнению лабораторных работ / Южно-Российский государственный политехнический университет (НПИ) имени М.И. Платова.– Новочеркасск: ЮРГПУ(НПИ), 2017.– 87 с.

Изложены особенности построения современной архитектуры компьютерных сетей в рамках распределенных корпоративных систем. Представлены тематика, содержание и объем лабораторных работ по разработке и исследованию архитектуры сетей. Приведено описание архитектурных решений, варианты заданий для выполнения лабораторных работ.

Предназначено для студентов вузов, обучающихся по направлениям подготовки «Информационные системы и технологии» (бакалавриат), «Прикладная информатика» (бакалавриат), «Приборостроение» (бакалавриат).

УДК 004.023 (О76.5)

Цель лабораторных работ – закрепление теоретических знаний и получение практических навыков моделирования и программной реализации прикладных предметно-ориентированных модулей информационного обмена в распределенных корпоративных системах.

Отчет по лабораторной работе должен содержать: цель работы, описание задания, краткие теоретические сведения по реализации алгоритма, описание и листинг программы, результаты выполнения программы, выводы о реализации программного модуля.

Лабораторная работа № 1

КОДИРОВАНИЕ ИНФОРМАЦИИ НА ФИЗИЧЕСКОМ УРОВНЕ

Цель работы: изучить основные методы преобразования и кодирования информации, используемые для передачи данных по физическим каналам вычислительных сетей и телекоммуникационных систем.

Задание к лабораторной работе

Разработать и реализовать диалоговую программу, формирующую графическое изображение передаваемого сигнала в линию связи по введенной двоичной последовательности.

Варианты заданий:

1. Код NRZ.
2. Код NRZI.
3. Манчестерский код.
4. Код MLT-3.
5. Код 4B/5B.
6. Код 8B/6T.
7. Код 8B/10B.
8. Дифференциальное манчестерское кодирование.
9. Биполярный код AMI.
10. Код HDB3.
11. Потенциальный код 2B1Q.

Пояснения к работе

Потенциальное кодирование также называется кодированием без возвращения к нулю (NRZ). При передаче нуля он передает потенциал, который был установлен на предыдущем такте (то есть не меняет его), а при передаче единицы потенциал инвертируется на противоположный. Этот код называется потенциальным кодом с инверсией при единице (NRZI).

NRZ

Для передачи единиц и нулей используются два устойчиво различаемых потенциала:
NRZ (прямой):

- биты 0 представляются нулевым напряжением 0 (В);
- биты 1 представляются значением U (В).

NRZ (перевёрнутый) (рис. 1.1):

- биты 0 представляются значением U (В);
- биты 1 представляются нулевым напряжением 0 (В).

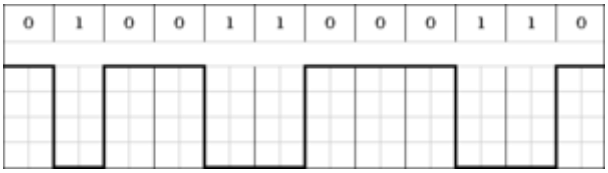


Рис. 1.1. Код NRZ (перевернутый)

NRZI

При передаче последовательности нулей сигнал не возвращается к нулю в течение такта (рис. 1.2). То есть смена сигнала происходит при передаче единицы, а передача нуля не приводит к изменению напряжения.

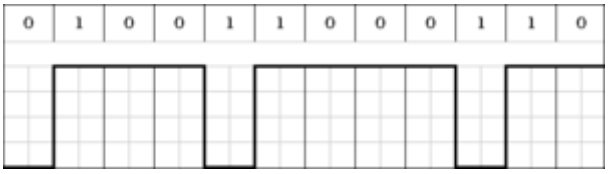


Рис. 1.2. Код NRZI

Достоинства метода NRZ:

— простота реализации;
— метод обладает хорошей распознаваемостью ошибок (благодаря наличию двух резко отличающихся потенциалов);

— основная гармоника f_0 имеет достаточно низкую частоту (равную $N/2$ Гц, где N — битовая скорость передачи дискретных данных [бит/с]), что приводит к узкому спектру.

Недостатки метода NRZ:

— метод не обладает свойством самосинхронизации. Даже при наличии высокоточного тактового генератора приёмник может ошибиться с выбором момента съёма данных, так как частоты двух генераторов никогда не бывают полностью идентичными. Поэтому при высоких скоростях обмена данными и длинных последовательностях единиц или нулей небольшое рассогласование тактовых частот может привести к ошибке в целый такт и, соответственно, считыванию некорректного значения бита;
— вторым серьёзным недостатком метода, является наличие низкочастотной составляющей, которая приближается к постоянному сигналу при передаче длинных последовательностей единиц и нулей. Из-за этого многие линии связи, не обеспечивающие прямого гальванического соединения между приёмником и источником, этот вид кодирования не поддерживают. Поэтому в сетях код NRZ в основном используется в виде различных его модификаций, в которых устранены как плохая самосинхронизация кода, так и проблемы постоянной составляющей.

Манчестерское кодирование

При манчестерском кодировании каждый такт делится на две части. Информация кодируется перепадами потенциала в середине каждого такта. Единица кодируется перепадом от низкого уровня сигнала к высокому, а ноль — обратным перепадом (рис. 1.3) (по стандарту IEEE 802.3, хотя по Д.Е. Томасу кодирование происходит наоборот). В начале каждого такта может происходить служебный перепад сигнала, если нужно представить несколько единиц или нулей подряд. Так как сигнал изменяется по крайней мере один раз за такт передачи одного бита данных, то манчестерский код обладает хорошими самосинхронизирующими свойствами. У манчестерского кода нет постоянной составляющей (меняется каждый такт), а основная гармоника в худшем случае (при передаче последовательности единиц или нулей) имеет частоту N Гц, а в лучшем случае (при передаче чередующихся единиц и нулей) —

N/2 Гц, как и у NRZ. В среднем ширина спектра при манчестерском кодировании в два раза шире чем при NRZ кодировании.

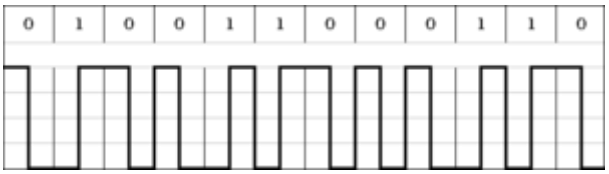


Рис. 1.3. Манчестерский код

Дифференциальное манчестерское кодирование

При дифференциальном манчестерском кодировании в течение битового интервала (времени передачи одного бита) уровень сигнала может меняться дважды (рис. 1.4). Обязательно происходит изменение уровня в середине интервала, этот перепад используется для синхронизации. Получается, что при передаче нуля в начале битового интервала происходит перепад уровней, а при передаче единицы такой перепад отсутствует.

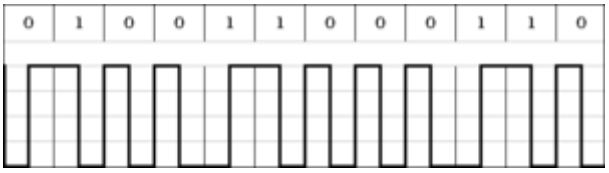


Рис. 1.4. Дифференциальный манчестерский код

Тринарное кодирование

RZ (с возвратом к нулю)

То есть каждый бит передается тремя уровнями напряжения. Поэтому требует в 2 раза больше скорости по сравнению с обычной скоростью. Это квазитроичный код, то есть изменение сигнала происходит между тремя уровнями.

Биполярный код AMI

AMI-код использует следующие представления битов (рис. 1.5):

- биты 0 представляются нулевым напряжением (0 В);
- биты 1 представляются поочерёдно значениями -U или +U (В).

AMI-код обладает хорошими синхронизирующими свойствами при передаче серий единиц и сравнительно прост в реализации. Недостатком кода является ограничение на плотность нулей в потоке данных, поскольку длинные последовательности нулей ведут к потере

синхронизации. Используется в телефонии уровня передачи данных, когда используются потоки мультиплексирования.

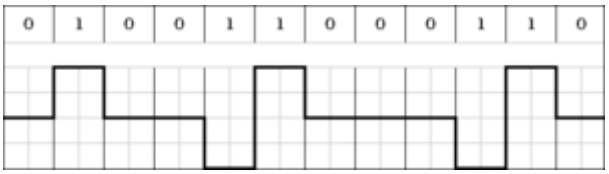


Рис. 1.5. Код AMI

HDB3

Код HDB3 исправляет любые 4 подряд идущие нули в исходные последовательности. Правило формирования кода следующее: каждые 4 нуля заменяются 4 символами, в которых имеется хотя бы один сигнал V. Для подавления постоянной составляющей полярность сигнала V чередуется при последовательных заменах. Для замены используются два способа:

1) если перед заменой исходный код содержал нечётное число единиц, то используется последовательность 000V;
2) если чётное, то 100V,

где V-сигнал единицы запрещённого для данного сигнала полярности.

То же, что и AMI, только кодирование последовательностей из четырех нулей заменяется на код -V, 0, 0, -V или +V, 0, 0, +V — в зависимости от предыдущей фазы сигнала.

MLT-3

MLT-3 Multi Level Transmission — 3 (многоуровневая передача) — метод кодирования, использующий три уровня сигнала. Метод основывается на циклическом переключении уровней -U, 0, +U. Единице соответствует переход с одного уровня сигнала на следующий. Так же, как и в методе **NRZI** при передаче «нуля» сигнал не меняется. В случае наиболее частого переключения уровней (длинная последовательность единиц) для завершения цикла необходимо четыре перехода. Это позволяет вчетверо снизить частоту несущей относительно тактовой частоты, что делает **MLT-3** удобным методом при использовании медных проводов в качестве среды передачи. Метод разработан Cisco Systems для использования в сетях FDDI на основе медных проводов, известных как CDDI. Также используется в Fast Ethernet 100BASE-TX.

Тетрарное кодирование

Потенциальный код 2B1Q

Код 2B1Q передает пару бит за один битовый интервал (рис. 1.6). Каждой возможной паре в соответствие ставится свой уровень из четырех возможных уровней потенциала. Паре 00 соответствует потенциал −2.5 В, 01 соответствует −0.833 В, 11 — +0.833 В, 10 — +2.5 В.

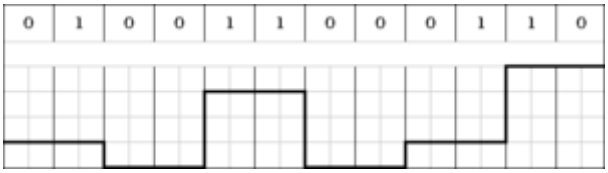


Рис. 1.6. Код 2B1Q

Достоинство метода 2B1Q: Сигнальная скорость у этого метода в два раза ниже, чем у кодов NRZ и AMI, а спектр сигнала в два раза уже. Следовательно, с помощью 2B1Q-кода можно по одной и той же линии передавать данные в два раза быстрее.

Недостаток метода 2B1Q: Реализация этого метода требует более мощного передатчика и более сложного приемника, который должен различать четыре уровня.

Контрольные вопросы

1. Что называется потенциальным кодированием?
2. Чем отличаются коды NRZI и NRZ?
3. Каковы достоинства и недостатки метода NRZ?
4. В чем достоинство манчестерского кодирования?
5. Какие достоинства и недостатки имеет метод 2B1Q?

Лабораторная работа № 2

ИЗУЧЕНИЕ ОСОБЕННОСТЕЙ ОРГАНИЗАЦИИ ПОМЕХОУСТОЙЧИВОГО КОДИРОВАНИЯ

Цель работы: приобретение практических навыков построения циклического кода по заданным характеристикам и проверка его свойства по обнаружению и исправлению ошибок.

Задание к лабораторной работе

1. Получить у преподавателя задание и ознакомиться с ним.
2. Вычислить параметры кода d, m, k, p, l, S. Найти образующий многочлен, воспользовавшись таблицей неприводимых многочленов.
3. Проверить, имеются ли ошибки в исследуемой комбинации, при наличии ошибок – исправить их.
4. Выполнить программную реализацию соответствующих алгоритмов кодирования.

Теоретические сведения

Циклические коды широко применяются при передаче данных в сетях и системах телеобработки данных. По способу и системе коррекции ошибок они относятся к блочным неразделимым кодам.

Основной принцип использования основывается на формировании комбинации кода путем циклического сдвига разрядов влево образующего многочлена. Эта операция аналогична процедуре умножения на X:

$$(X_4 + X_3 + X_2 + 1) \cdot X = X_5 + X_4 + X_3 + X$$
$$0011101 \qquad\qquad 0111010$$

Таким образом, при соответствующем выборе образующего многочлена любая разрешенная комбинация может быть получена в результате умножения образующего многочлена на некоторый другой многочлен.

Основная идея обнаружения и исправления ошибок заключается в делении комбинации кода на образующий многочлен, т. е.:

$$\frac{G(X)\cdot X}{K(X)} = Q(X) + \frac{R(X)}{K(X)},$$

где $G(X)$ – комбинация кода;

$K(X)$ – образующий многочлен;

$Q(X)$ – результат деления;

$R(X)$ – остаток.

Если остаток равен нулю, то исследуемая комбинация – разрешенная, и код не содержит ошибки. В противном случае имеется ошибка.

Пример. Найти образующий многочлен для следующих параметров кода: $d_0 = 3$, $n = 7$.

Решение. Вычислим число проверочных m и информационных k символов.

$$m = \log (n + 1) = 3 \; k = n – m = 7 – 3 = 4.$$

По таблице неприводимых многочленов найдем для $m = 3$ и $d = 3$ образующий многочлен вида 1101 или: $K(X) = X_3 + X_2 + 1$.

Вычислим проверочные разряды и получим образующую матрицу путем умножения всех комбинаций кода на образующий многочлен.

0000		0000000
0001		0001101
0010		0011010
0011		0010111
0100		0110100
0101		0111001
0110		0101110
0111		0100011
1000	· (1101)=	1101000
1001		1100101
1010		1110010
1011		1111111
1100		1001110
1101		1010001
1110		1000110
		1001011

Проверим возможность кода на обнаружение и исправление ошибок. Возьмем комбинацию 0111001. Разделим ее на образующий многочлен:

```
0111001 | 1101
 1101     101
-------
0001101
   1101
-------
   0000
```

Остаток равен 0, следовательно, это разрешенная комбинация. Исказим третий разряд:

```
0111101 | 1101
 1101   | 101
-------
0001001
   1101
-------
   0000
```

Остаток свидетельствует об обнаружении ошибки.

Правила построения циклических кодов, исправляющих одну ошибку

1. Расчет соотношения между разрядами:

$$n = m + k,$$

где m – число проверочных разрядов;

k – число информационных разрядов;

$$m = [\log (n + 1)]$$

или

$$m = [\log \{(k + 1) + [\log (k + 1)]\}].$$

2. Выбор образующего многочлена производится по таблицам неприводимых двоичных многочленов, где m – степень многочлена, d – число единиц в комбинации.

3. Выбор параметров единичной матрицы производится исходя из условия, что число столбцов матрицы определяется числом информационных разрядов.

4. Определение элементов дополнительной матрицы производится по остаткам от деления последней строки транспонированной матрицы на образующий многочлен (это еще один способ формирования образующей матрицы).

5. Образующая матрица составляется путем дописывания элементов дополнительной матрицы справа от единичной матрицы или путем умножения элементов единичной матрицы на образующий многочлен.

6. Комбинациями исходного кода являются строки образующей матрицы и всевозможные суммы по модулю 2 различных сочетаний строк образующей матрицы.

7. Обнаружение и исправление ошибок происходит по остаткам от деления принятой комбинации $G(X)$ на образующий многочлен $K(X)$. Если деление без остатка, то ошибки нет. Для исправления ошибки:

а) принятая комбинация делится на образующий многочлен;

б) подсчитывается вес остатка.

Если $W \in S$, где S – допустимое число исправляемых ошибок, то принятая комбинация складывается по модулю 2 с полученным остатком. Сумма даст исправленную комбинацию.

Если $W > S$, то делим полученную в результате циклического сдвига комбинацию на образующий многочлен. Если в остатке $W \in S$, то складываем делимое с остатком. Затем производим циклический сдвиг вправо комбинации, полученной в результате суммирования последнего делимого с остатком. Если после первого циклического сдвига и последующего деления остаток получается таким, что его вес $W > S$, то процедура повторяется до тех пор, пока $W \in S$. Затем производится циклический сдвиг вправо на столько разрядов, на сколько была сдвинута принятая комбинация. В результате получаем исправленную комбинацию.

Контрольные вопросы

1. В чем заключаются основные идеи обнаружения и исправления ошибок циклическим кодом?
2. Что такое кодовое расстояние?
4. Чем отличается представление циклическим кодом для $d = 3$ и $d = 5$, где d – кодовое расстояние?
5. Какие существуют способы формирования комбинаций циклического кода?
5. В чем достоинство циклических кодов?
6. Что такое транспонированная матрица для циклического кода и ее размерность?

Лабораторная работа № 3

ИЗУЧЕНИЕ ОСОБЕННОСТЕЙ СИНТЕЗА СВЕРТОЧНЫХ КОДОВ

Цель работы: изучить работу кодера и декодера сверточных кодов по алгоритму Витерби. Проанализировать исправляющую

способность декодера.

Варианты заданий:

1. Разработать программу, генерирующую линейный код с параметрами (*m, k, d*), и определить линейные формы ($\beta_J = \sum C_{iJ} X_i$), соответствующие наименьшему значению *k* при заданном *d*.
2. Написать программу для исследования зависимости вероятности ошибки в символе при параметрах *d, n*.
3. Спроектировать код Рида-Маллера для заданных *d, m*.
4. Написать программу для генерации примитивных циклических кодов. Осуществить декодирование слов с внесенными (преподавателем) ошибками.
5. Составить программы для описания примитивных порождающих многочленов (разложением y_n - 1 на простые многочлены) и построение нерасширенных полей.
6. Разработать программу для генерации кодов БЧХ.
7. Разработать программу для декодирования БЧХ кодов.

Теоретические сведения

Сверточные коды можно рассматривать как частный случай блоковых кодов, но наличие сверточной структуры наделяет его дополнительными свойствами, улучшающими его характеристики. Как любой корректирующий код, сверточные коды защищают информацию, добавляя избыточные символы. Кодирующее устройство сверточного кода со скоростью $R = k/n$ (рис. 3.1) обрабатывает входную последовательность, состоящую их k информационных символов, и вычисляет n кодовых (канальных) символов ($n > k$). Если один (например первый) из n символов текущего блока повторяет текущий информационный бит, код называется систематическим.

Сверточный кодер с кодовым ограничением v представляет собой регистр памяти для хранения k информационных символов и преобразователь информационной последовательности в кодовую последовательность. Процесс кодирования производится непрерывно. Информационные двоичные символы $\{a_i\}$ поступают на вход регистра сдвига с ячейками, в котором символы кодовой последовательности формируются суммированием по модулю 2 символов

с выходов некоторых ячеек. Подключение сумматоров к ячейкам регистра задается генераторными полиномами *g1(x)* и *g2(x)*.

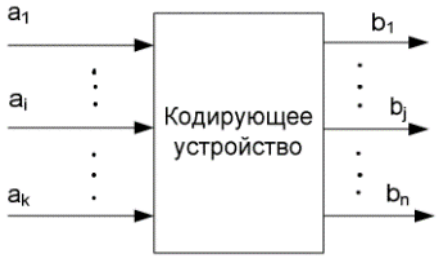


Рис. 3.1. Кодирующее устройство сверточного кода со скоростью *R = k/n*

Будем полагать, что кодирование производится с использованием сверточного (7,5) - кода. Схема сверточного кодера в этом случае будет иметь вид, представленный на рис.3.2.

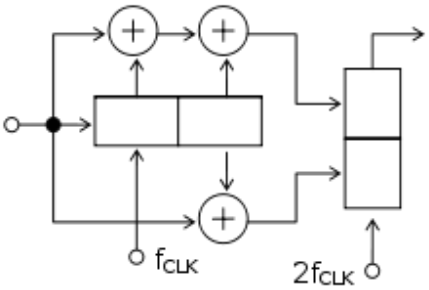


Рис. 3.2. Схема сверточного кодера для кода (7,5)

Алгоритм функционирования такого кодера поясняется следующей диаграммой (рис. 3.3). Решетчатая диаграмма является разверткой диаграммы состояний во времени (рис.3.4).

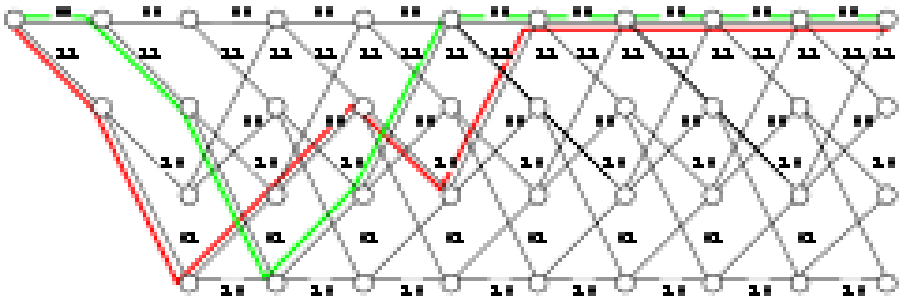


Рис. 3.3. Алгоритм функционирования кодера (7,5)

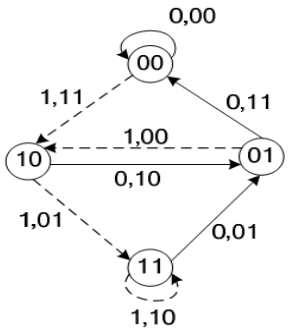


Рис. 3.4. Диаграмма состояний сверточного кодера указанной структуры.

Сверточный кодер можно рассматривать как постоянный во времени конечный автомат, структура которого является периодической и может быть описана с помощью различных диаграмм. Например, сверточный кодер может быть описан диаграммой состояний. Диаграмма представляет собой направленный граф и описывает все возможные переходы кодера из одного состояния в другое, а также содержит выходные символы кодера, которые сопровождают эти переходы. Пример диаграммы состояний показан на рис. 3.4. В кружках указаны четыре возможных состояния кодера 00, 10, 01, 11, линиями со стрелками - возможные переходы. Сплошная линия отмечает переходы, совершаемые при поступлении на вход кодирующего устройства информационного символа 0, пунктирная - при поступлении символа 1. Символы около линий обозначают символы на входе и выходе кодера, соответствующие данному переходу.

Принцип функционирования декодера Витерби состоит в следующем: на вход декодера поступает сегмент последовательности r длиной b, превышающей кодовую длину блока n. Назовем b окном декодирования. Сравним все кодовые слова данного кода (в пределах сегмента длиной b) с принятым словом и выберем кодовое слово, ближайшее к принятому. Первый информационный кадр выбранного кодового слова принимается в качестве оценки информационного кадра декодированного слова. После этого в декодер вводится n_0 новых символов, а введенные ранее самые старые n_0 символов сбрасываются, и процесс повторяется для определения следующего информационного кадра. Таким образом, декодер Витерби последовательно обрабатывает кадр за кадром, двигаясь по решетке, аналогичной используемой кодером. В каждый момент времени декодер не знает, в каком узле находится кодер, и не пытается его

декодировать. Вместо этого декодер по принятой последовательности определяет наиболее правдоподобный путь к каждому узлу и определяет расстояние между каждым таким путем и принятой последовательностью. Это расстояние называется мерой расходимости пути. В качестве оценки принятой последовательности выбирается сегмент, имеющий наименьшую меру расходимости. Путь с наименьшей мерой расходимости называется выжившим путем.

Свободное расстояние кода (7,5) равно минимальному весу пути по диаграмме из состояния 00 в то же состояние (исключая петлю у состояния 00). Диаграмма свободного расстояния для рис. 3.4 изображена на рис. 3.5 и для него d_f=5.

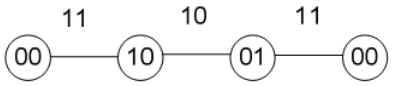


Рис. 3.5. Диаграмма свободного расстояния кодера (7,5)

Рассмотрим работу декодера Витерби на примере сверточного кода (7,5). Все решетчатые диаграммы процесса декодирования представлены на рис. 3.6.

Пользуясь решетчатой диаграммой кодера, попытаемся, приняв некоторый сегмент r, проследить наиболее вероятный путь кодера. При этом для каждого сечения решетчатой диаграммы будем отмечать меру расходимости пути каждому ее узлу. Предположим, что передана кодовая последовательность U = (00000000…), а принятая последовательность имеет вид r = (10001000…), то есть в первом и в третьем кадрах кодового слова возникли ошибки. Как мы уже убедились, процедура и результат декодирования не зависят от передаваемого кодового слова и определяются только ошибкой, содержащейся в принятой последовательности. Поэтому проще всего считать, что передана нулевая последовательность, то есть U = (00000000…). Приняв первую пару символов (10), определяется мера расходимости для первого сечения решетки, приняв следующую пару символов (00), — для второго сечения и т.д. При этом из входящих в каждый узел путей оставляем путь с меньшей расходимостью, поскольку путь с большей на данный момент расходимостью уже не сможет стать в дальнейшем короче. Заметим, что для рассматриваемого примера, начиная с четвертого уровня, метрика (или мера расходимости) нулевого пути меньше любой другой метрики. Поскольку ошибок в канале больше не было,

ясно, что в конце концов в качестве ответа будет выбран именно этот путь.

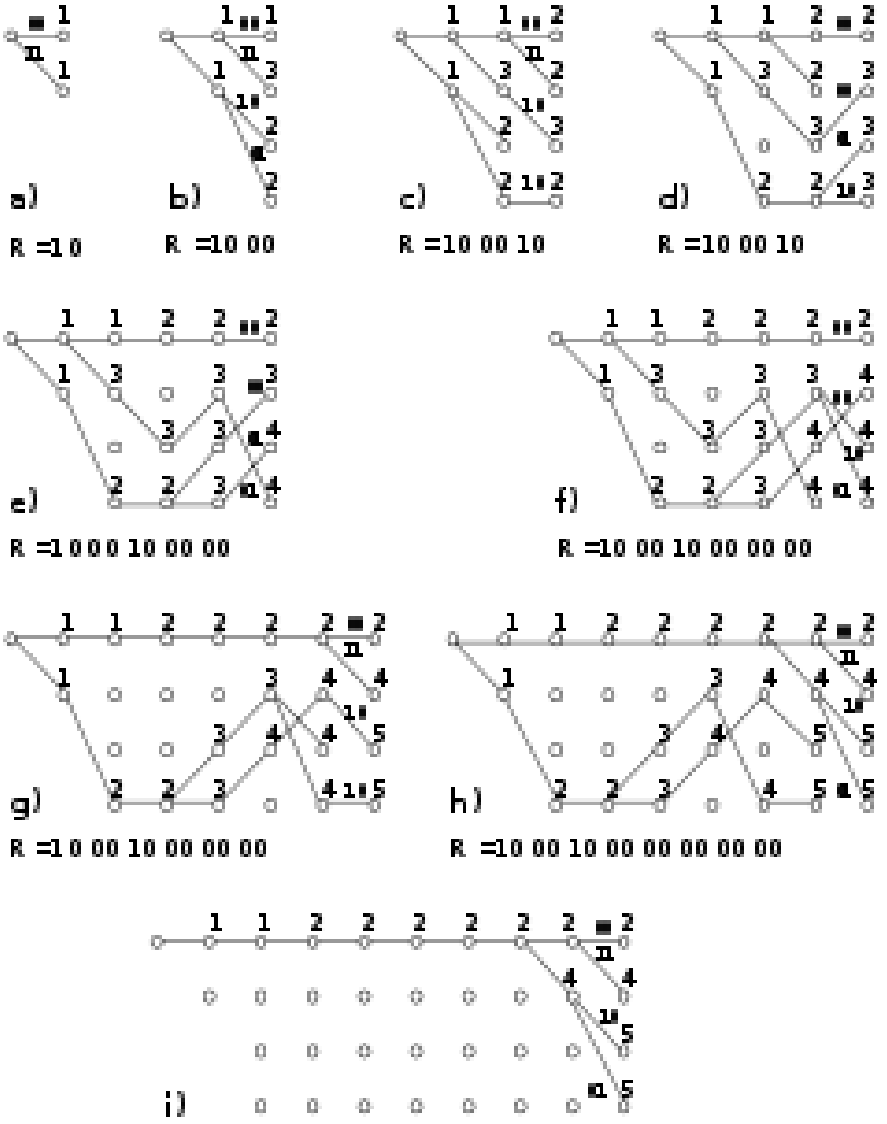


Рис. 3.6. Решетчатые диаграммы сверточного декодера Витерби (7,5)

Контрольные вопросы

1. Назовите основные элементы сверточных кодов.
2. Как производиться оценка исправляющей способности сверточного кода?
3. Поясните алгоритм декодирования Витерби
4. Как различают жесткую и мягкую схему принятия решения?

Лабораторная работа № 4

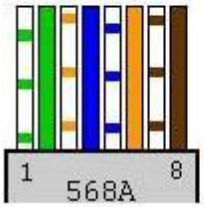
ИЗУЧЕНИЕ КОМПОНЕНТ СТРУКТУРИРОВАННЫХ КАБЕЛЬНЫХ СИСТЕМ

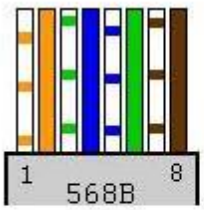
Цель работы: изучить технологию обжима кабеля витая пара.

Варианты заданий:
1. Компьютер – хаб EIA/TIA 568A.
2. Компьютер – компьютер или хаб – хаб EIA/TIA 568A.
3. Компьютер – хаб EIA/TIA 568B.
2. Компьютер – компьютер или хаб – хаб EIA/TIA 568B.

Теоретические сведения

Цветовые схемы обжима в вилке RJ-45

EIA/TIA-568A			
одна сторона	**цвет провода**	**другая сторона**	
1	белый зеленого	1	
2	зеленый	2	
3	белый оранжевого	3	
4	синий	4	
5	белый синего	5	
6	оранжевый	6	
7	белый коричневого	7	
8	коричневый	8	

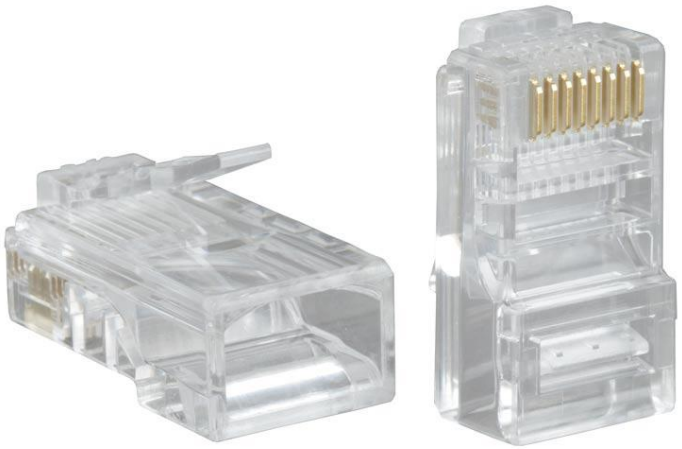
EIA/TIA-568B, AT&T 258A			
одна сторона	**цвет провода**	**другая сторона**	
1	белый оранжевого	1	
2	оранжевый	2	
3	белый зеленого	3	
4	синий	4	
5	белый синего	5	
6	зеленый	6	
7	белый коричневого	7	
8	коричневый	8	

Соединение компьютер – хаб

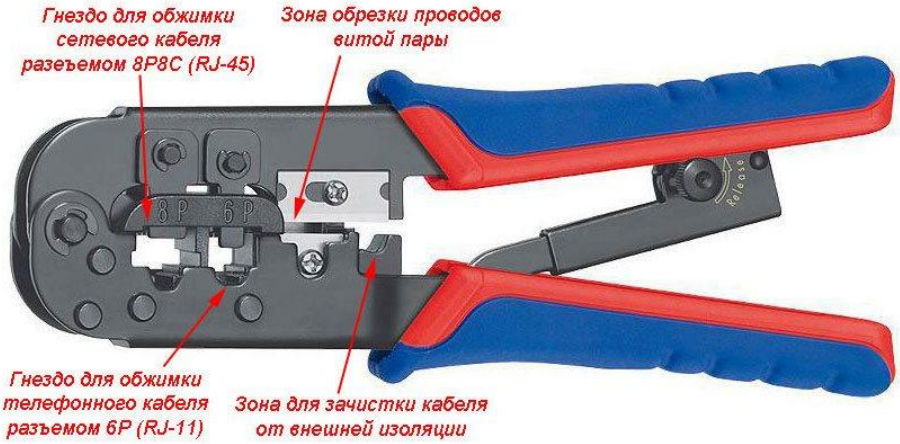
Компьютер – компьютер или хаб – хаб

При подключении по схеме компьютер-компьютер, кабеля обжимаются по разным моделям. Самые современные сетевые карты и хабы, при поддержке технологии Auto-MDIX, способны самостоятельно определять вариант обжатия кабеля и совершают внутреннюю подстройку.

Коннектор RJ-45. Второе название 8P8C (8 Position 8 Contact).



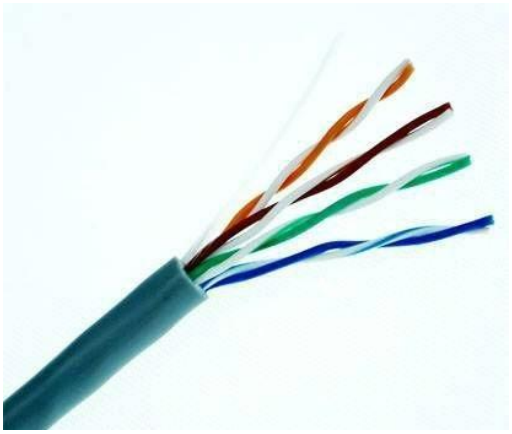
Обжимные клещи или кримпер. С их помощью можно сэкономить время и упростить себе работу. Они бывают разных форм, с лезвием для зачистки изоляции или без них. Некоторые имеют возможность обжимать и провод телефонной линии. Самое важное свойство этого инструмента – равномерное сжатие всех проводов.



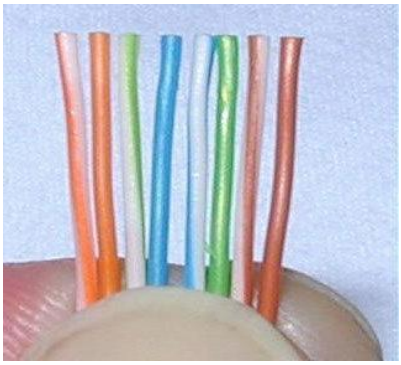
Обжатие провода RJ-45

1. Перед тем, как обжимать разъемы, отрежьте (клещами) кусок кабеля нужной длины под прямым углом так, чтобы все проводники были ровные.

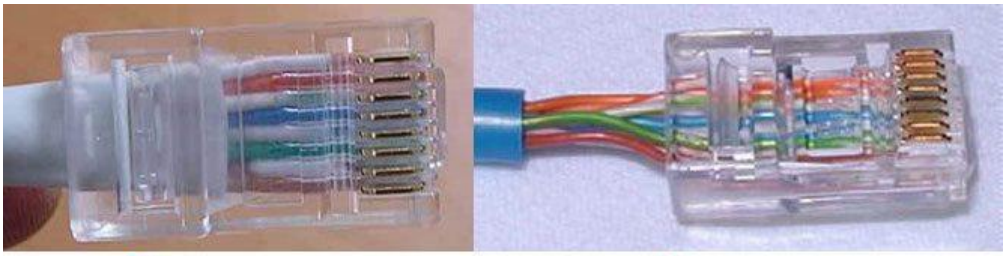
2. С каждой стороны снимите внешнюю оболочку изоляции на 2,5 – 3,5 см. Только не повредите изоляцию самих проводников!



3. Подберите жилы по цветам в правильной последовательности, согласно схеме обжима, которую вы выбрали. Для этого провода сначала расплетают и выравнивают. Затем они раскладываются в ряд в правильной последовательности и прижимаются плотно друг к другу. Концы провода откусывают кримпером на расстоянии 10 – 15 мм от края изоляции.



4. Очень аккуратно одеваем коннектор в кабель. Нужно постоянно наблюдать за проводниками, чтобы они не перепутались и прошли в правильной последовательности. Проталкивайте жилы, пока они не войдут до конца и не упрутся в стенку разъема. Если вы все сделали правильно, проводники были отрезаны ровно, они все зайдут в разъем равномерно и до упора. Изоляция, при этом, окажется внутри корпуса. Если проводники вне коннектора не изолированы, обязательно вытяните их и обрежьте до нужной длины.



5. Когда кабель готов, осталось его только закрепить. Вставьте коннектор в соответствующее гнездо кримпера и до упора плавно зажмите рукоятку.
6. После обжатия, обязательно пошатайте провод рукой. Он должен крепко удерживаться в коннекторе.
7. Обжать другой конец кабеля аналогично схеме.

Контрольные вопросы

1. Назовите основные цветовые схемы обжима в вилке RJ-45.
2. Как можно обеспечить соединение компьютер-хаб?
3. Каким образом можно соединить хаб с хабом?
4. Что такое кримпер?

Лабораторная работа № 5
ИЗУЧЕНИЕ СЕТЕВЫХ УТИЛИТ ОПЕРАЦИОННОЙ СИСТЕМЫ СЕМЕЙСТВА WINDOWS

Цель работы: изучить назначение и основные функции утилит командной строки операционной системы семейства Windows.

Теоретические сведения

Большинство рассматриваемых сетевых утилит для полноценной работы требуют наличия административных привилегий. Для операционных систем семейства Windows 2000/XP достаточно того, чтобы пользователь работал под учетной записью члена группы администраторов. Интерпретатор командной строки **cmd.exe** можно запустить с использованием меню **Пуск - Выполнить - cmd.exe**. В среде операционных систем Widows Vista/Windows 7 интерпретатор **cmd.exe** должен быть запущен для выполнения с использованием пункта контекстного меню "Запустить от имени администратора". Командные файлы, в которых используются сетевые утилиты, также должны выполняться в контексте учетной записи с привилегиями администратора.

В описании команд используются:
< текст > - текст в угловых скобках. Обязательный параметр;
[текст] - текст в квадратных скобках. Необязательный параметр;
(текст) - текст в круглых скобках. Необходимо выбрать один из параметров;
Вертикальная черта | - разделитель для взаимоисключающих параметров. Нужно выбрать один из них;
Многоточие ... - возможно повторение параметров.

Утилита ARP.EXE

Утилита командной строки ARP.EXE присутствует во всех версиях Windows и имеет один и тот же синтаксис. Команда **ARP** позволяет просматривать и изменять записи в кэш ARP (Address Resolution Protocol - протокол разрешения адресов), который представляет собой таблицу соответствия IP-адресов аппаратным адресам сетевых устройств. Аппаратный адрес - это уникальный,

присвоенный при изготовлении 6-байтный адрес сетевого устройства, например сетевой карты. Этот адрес также часто называют MAC-адресом (Media Access Control - управление доступом к среде) или Ethernet-адресом. В сетях Ethernet передаваемые и принимаемые данные всегда содержат MAC-адрес источника (Source MAC) и MAC-адрес приемника (Destination MAC). Два старших бита MAC-адреса используются для идентификации типа адреса:
- первый бит - одиночный (0) или групповой (1) адрес;
- второй бит - признак универсального (0) или локально администрируемого (1) адреса.

Следующие 22 бита адреса содержат специальный код производителя **MFG** или **OUI** - универсальный код организации. Другими словами, любое сетевое устройство имеет аппаратный адрес, состоящий из 2- х частей. Старшую часть MAC - адреса, централизованно выделяемую по лицензии каждому производителю сетевого оборудования. Например, 00:E0:4C - для сетевых устройств REALTEK SEMICONDUCTOR CORP. Крупным производителям сетевого оборудования обычно принадлежит несколько диапазонов OUI . И младшую часть MAC-адреса, которая формируется при производстве оборудования, и уникальна для каждого экземпляра устройства.

Отображение IP-адресов (формируемых программным путем), в аппаратные адреса, выполняется с помощью следующих действий:
- в сеть отправляется широковещательный запрос (ARP-request), принимаемый всеми сетевыми устройствами. Он содержит IP и Ethernet адреса отправителя, а также целевой IP-адрес, для которого выполняется определение MAC-адреса;
- каждое устройство, принявшее запрос, проверяет соответствие целевого IP-адреса, указанного в запросе, своему собственному IP-адресу. При совпадении отправителю передается ARP-ответ (ARP-Reply), в котором содержатся IP- и MAC-адреса ответившего узла. Кадр с ARP-ответом содержит IP- и MAC-адреса как отправителя, так и получателя-составителя запроса;
- информация, полученная в ARP-ответе, заносится в ARP-кэш и может использоваться для обмена данными по IP-протоколу для данного узла. ARP-кэш представляет собой таблицу в оперативной памяти, каждая запись в которой содержит IP, MAC и возраст их

разрешения. Возраст записи учитывается для того, чтобы обеспечить возможность повторного выполнения процедуры ARP при каком-либо изменении соответствия адресов.

Синтаксис ARP.EXE:

arp[-a [InetAddr] [-NIfaceAddr]] [-g [InetAddr] [-NIfaceAddr]] [-dInetAddr [IfaceAddr]] [-sInetAddr EtherAddr [IfaceAddr]]

-a[InetAddr] [-NIfaceAddr] - ключ **-a** - отображает текущую таблицу ARP для всех интерфейсов. Для отображения записи конкретного IP-адреса используется ключ **-a** с параметром InetAdd , в качестве которого указывается IP-адрес. Если узел, отправляющий ARP-запрос, имеет несколько сетевых интерфейсов, то для отображения таблицы ARP нужного интерфейса можно использовать ключ -N с параметром IfaceAddr, в качестве которого используется IP-адрес интерфейса.
-g[InetAddr] [-NIfaceAddr] ключ -g идентичен ключу -a.
-d InetAddr[IfaceAddr] - используется для удаления записей из ARP-кэш. Возможно удаление по выбранному IP или полная очистка ARP кэш. Для удаления всех записей, вместо адреса используется символ ***.** Если имеется несколько сетевых интерфейсов, то очистку можно выполнить для одного из них, указав в поле IfaceAddr его IP .
-s InetAddr EtherAddr [IfaceAddr] - используется для добавления статических записей в таблицу ARP. Статические записи хранятся в ARP-кэш постоянно. Обычно, добавление статических записей используется для сетевых устройств, не поддерживающих протокол ARP или не имеющих возможности ответить на ARP- запрос.
/? - получение справки по использованию arp.exe. Аналогично - запуск arp.exe без параметров.

Утилита IPCONFIG

Утилита командной строки IPCONFIG присутствует во всех версиях Windows. Некоторые параметры командной строки не поддерживаются в версиях предшествующих Windows Vista/Windows 7. Команда **IPCONFIG** используется для отображения текущих настроек протокола TCP/IP и для обновления некоторых параметров, задаваемых при автоматическом конфигурировании сетевых интерфейсов при использовании протокола Dynamic Host Configuration Protocol (DHCP).

Синтаксис:

ipconfig [/allcompartments] [/all] [/renew[Adapter]] [/release[Adapter]] [/renew6[Adapter]] [/release6[Adapter]] [/flushdns] [/displaydns] [/registerdns]

[/showclassidAdapter] [/setclassidAdapter [ClassID]]

/? - отобразить справку по использованию IPCONFIG
/all - отобразить полную конфигурацию настроек TCP/IP для всех сетевых адаптеров. Отображение выполняется как для физических интерфейсов, так и для логических, как например, dialup или VPN подключения.
/allcompartments - вывести полную информацию о конфигурации TCP/IP для всех секций. Применимо для Windows Vista/Windows 7 .
/displaydns - отобразить содержимое кэш службы DNS - клиент.
/flushdns - сбросить содержимое кэш службы DNS - клиент.
/registerdns - инициировать регистрацию записей ресурсов DNS для всех адаптеров данного компьютера. Этот параметр используется для изменения настроек DNS сетевых подключений без перезагрузки компьютера.
/release[Adapter] - используется для отмены автоматических настроек сетевого адаптера, полученных от сервера DHCP. Если имя адаптера не указано, то отмена настроек выполняется для всех адаптеров.
/release6[Adapter] - отмена автоматических настроек для протокола IPv6.
/renew[Adapter] - обновить конфигурацию для сетевого адаптера, настроенного на получение настроек от сервера DHCP. Если имя адаптера не указано, то обновление выполняется для всех адаптеров.
/renew6[Adapter] - как и в предыдущем случае, но для протокола IPv6.
/showclassid Adapter и **/setclassid Adapter[ClassID]** - эти параметры применимы для Windows Vista / Windows 7 и используются для просмотра или изменения идентификатора Class ID, если он получен от DHCP - сервера при конфигурировании сетевых настроек.

Изменение сетевых настроек с помощью команды IPCONFIG, в основном, применимо к тем сетевым адаптерам, которые настроены на автоматическое конфигурирование с использованием службы динамической настройки основных параметров на сетевом уровне DHCP (Dynamic Host Configuration Protocol) или службы автоматической настройки приватных IP - адресов APIPA (Automatic Private IP Addressing). Если в параметрах командной строки IPCONFIG используется имя адаптера, содержащее пробелы, то оно должно заключаться в двойные кавычки. Если имя содержит символы русского алфавита, то оно должно быть представлено в DOS-кодировке. Для имен адаптеров применимо использование символа ***** в качестве шаблона:

***** - любое имя
Локальн* - имя адаптера начинается с " Локальн ";
*** сети *** - имя адаптера содержит строку " сети ".

Утилита GETMAC

Утилита командной строки GETMAC присутствует в версиях Windows XP и старше. Используется для получения аппаратных адресов сетевых адаптеров (MAC-адресов) как на локальном, так и на удаленном компьютере.

Синтаксис:

GETMAC [/S <система> [/U <пользователь> [/P <пароль>]]] [/FO <формат>] [/NH] [/V]

Параметры:

/S <система> - имя или IP-адрес удаленного компьютера.
/U [<домен>\]<пользователь> Имя пользователя. Если не задано, то используется текущая учетная запись.
/P [<пароль>] - Пароль. Если задан параметр /U и не задан пароль, то он будет запрошен.
/FO <формат> - Формат, в котором следует отображать результаты запроса. Допустимые форматы: "TABLE" (таблица), "LIST" (список), "CSV" (разделяемые запятыми поля). Если параметр не задан, то используется вывод в виде таблицы (TABLE) .
/NH - Указывает, что строка заголовков столбцов не должна отображаться в результирующем файле. форматов TABLE и CSV.
/V - Отображение подробной информации. В отображаемой информации присутствует имя сетевого подключения и название сетевого адаптера.
/? - Вывод справки по использованию команды.

Утилита NBTSTAT

Команда NBTSTAT позволяет получить статистику протокола NetBIOS over TCP/IP (NetBT), таблицу имен локальных и удаленных компьютеров и содержимое кэш NetBIOS имен. Применение NBTSTAT позволяет принудительно обновить кэш NetBIOS-имен компьютеров и имена, зарегистрированные с помощью серверов Windows Internet Name Service (WINS).

Синтаксис:

nbtstat[-a RemoteName] [-A IPAddress] [-c] [-n] [-r] [-R] [-RR] [-s] [-S] [Interval]

Параметры командной строки:

-a RemoteName - отображает таблицу имен удаленного компьютера. NetBIOS-имена соответствуют перечню NetBIOS-приложений, выполняющихся на удаленном компьютере.
-A IPAddress - то же самое, что и в предыдущем случае, но вместо имени удаленного компьютера используется его IP-адрес.
-c - отображает кэш имен NetBIOS и соответствующих им IP-адресов.
-n - отображает таблицу NetBIOS-имен на локальном компьютере. Состояние "Зарегистрирован" означает, что имя зарегистрировано с использованием широковещательного запроса или с помощью сервера WINS.
-r - отображает статистику разрешения NetBIOS-имен. На компьютерах под управлением Windows XP и старше выдается раздельная статистика о разрешении имен с помощью широковещательной рассылки и с помощью сервера имен WINS.
-R - очистка кэш NetBIOS-имен и загрузка данных из секции #PRE файла LMHOSTS.
-RR - очистка кэш NetBIOS - имен на локальном компьютере и их повторная регистрация с использованием сервера WINS.
-s - отображает статистику NetBIOS - сессий между клиентом и сервером и NetBIOS-имена удаленных узлов.
-S - отображает статистику сессий между клиентом и сервером и IP-адреса удаленных узлов.
Interval - интервал обновления отображаемых данных в секундах. Для прекращения автоматического обновления используется комбинация клавиш CTRL+C.
/? - отобразить справку по использованию NBTSTAT.

Утилита NETSH.EXE

Утилита сетевой оболочки NETSH (NETwork SHell) - наиболее полное и функциональное стандартное средство управления сетью с использованием командной строки в среде Windows XP и старше. Набор внутренних команд сетевой оболочки пополняется с появлением новых версий операционной системы, что необходимо учитывать при работе в локальной сети с различными ОС. Так, например, команда уровня wlan (netsh wlan - управление беспроводной сетью) может использоваться на компьютерах под управлением Windows Vista и старше и отсутствует в Widows XP. Синтаксис используемых команд и параметров также может различаться в разных операционных системах семейства Windows.

При запуске NETSH.EXE без параметров на экран выводится приглашение к вводу внутренних команд оболочки. Набор команд представляет собой многоуровневую структуру, позволяющую выполнять необходимые действия в выбранном контексте. При вводе знака вопроса **?** можно получить краткую справку по доступному перечню команд на данном уровне. Ввод команды данного уровня со

знаком вопроса вызовет отображение справки по ее использованию. Аналогичную справку можно получить, введя определенную команду и, после перехода на уровень ее выполнения, ввести знак вопроса. При необходимости, можно выполнить нужное действие без использования интерактивного режима, указав в качестве параметров командной строки последовательный набор внутренних команд NETSH и необходимых параметров.

Например:

netsh advfirewall show global последовательно выполняется команда первого уровня **advfirewall**, в ее контексте, команда следующего уровня **show** с параметром **global**

Команды NETSH можно выполнить и на удаленном компьютере с использованием подключения по локальной сети. Netsh также предоставляет возможность выполнения сценариев, представляющих собой группу команд в текстовом файле, выполняемых в режиме очередности на определенном компьютере. В целом, возможности NETSH настолько обширны, что трудно найти сетевую задачу, которую невозможно было бы решить с использованием данной утилиты.

Синтаксис:

NETSH.EXE [-a AliasFile] [-c Context] [-r RemoteMachine] [-u [DomainName\]UserName] [-p Password | *] [Command | -f ScriptFile]

-a AliasFile - не завершать работу, а перейти к приглашению ввода команд после выполнения AliasFile. AliasFile - имя текстового файла, в котором содержатся одна или несколько команд netsh .
-c Context - изменить контекст (уровень) команд netsh.
-r RemoteMachine - выполнять команды netsh на удаленном компьютере. В качестве RemoteMachine может использоваться имя или IP-адрес.
[-u DomainName\]UserName - имя пользователя для подключения к удаленному компьютеру. Если не задано, то используется текущее имя пользователя.
-p Password пароль для подключения к удаленному компьютеру.
Command - команда оболочки netsh , которую необходимо выполнить.
-f ScriptFile - аналогично ключу -a, но после выполнения команд файла сценария Scriptfile, работа netsh завершается.

Утилита NETSTAT.EXE

Утилита netstat.exe присутствует во всех версиях Windows, однако существуют некоторые отличия используемых параметров командной строки и результатов ее выполнения, в зависимости от операционной системы. Используется для отображения TCP и UDP -соединений, слушаемых портов, таблицы маршрутизации, статистических данных для различных протоколов.

Синтаксис:

netstat[-a] [-e] [-n] [-o] [-pProtocol] [-r] [-s] [Interval]

-a - отображение всех активных соединений по протоколам TCP и UDP, а также списка портов, которые ожидают входящие соединения (слушаемых портов).
-b - отображение всех активных соединений по протоколам TCP и UDP, а также списка портов, которые ожидают входящие соединения (слушаемых портов) с информацией об именах исполняемых файлов. Данный параметр применим для операционных систем Widows XP и старше.
-e - отображение статистики Ethernet в виде счетчиков принятых и отправленных байт и пакетов.
-n - отображение номеров портов в виде десятичных чисел.
-o - отображение соединений, включая идентификатор процесса (PID) для каждого соединения.
-p Protocol - отображение соединений для заданного протокола. Протокол может принимать значения **tcp, udp, tcpv6, udpv6** . При использовании совместно с параметром **-s** в качестве протокола можно задавать **tcp, udp, icmp, ip, tcpv6, udpv6, icmpv6, ipv6.**
-s - отображение статистических данных по протоколам TCP, UDP, ICMP, IP , TCP over IPv6, UDP over IPv6, ICMPv6, и IPv6 . Если задан параметр **-p** , то статистика будет отбражаться только для выбранных протоколов.
-r - отображение таблицы маршрутов. Эквивалент команды **route print**
Interval - интервал обновления отображаемой информации в секундах.
-v - отображать подробную информацию.
/? - отобразить справку по использованию netstat

При использовании утилиты **netstat.exe** удобно пользоваться командами постраничного вывода (more), перенаправления стандартного вывода в файл (>) и поиска текста в результатах (find).
netstat -a | more - отобразить все соединения в постраничном режиме вывода на экран.
netstat -a > C:\netstatall.txt - отобразить все соединения с записью результатов в файл C:\netstatall.txt.
netstat -a | find /I "LISTENING" - отобразить все соединения со статусом LISTENING. Ключ **/I** в команде **find** указывает, что при поиске текста не нужно учитывать регистр символов.
netstat -a | find /I "listening" > C:\listening.txt - отобразить все соединения со статусом

LISTENING с записью результатов в файл C:\listening.txt.

Утилита NET.EXE

Утилита NET.EXE существует во всех версиях Windows и является одной из самых используемых в практической работе с сетевыми ресурсами. Позволяет подключать и отключать сетевые диски, запускать и останавливать системные службы, добавлять и удалять пользователей, управлять совместно с используемыми ресурсами, устанавливать системное время, отображать статистические и справочные данные об использовании ресурсов и многое другое.

Выполнение команды **net** без параметров вызывает краткую справку со списком возможных уровней использования, запуск с параметром **help** позволяет получить более подробную информацию об использовании net.exe:

Синтаксис данной команды:

NET HELP
имя_команды
-или-
NET имя_команды /HELP
Можно использовать следующие имена команд:

NET ACCOUNTS NET HELP NET SHARE
NET COMPUTER NET HELPMSG NET START
NET CONFIG NET LOCALGROUP NET STATISTICS
NET CONFIG SERVER NET NAME NET STOP
NET CONFIG WORKSTATION NET PAUSE NET TIME
NET CONTINUE NET PRINT NET USE
NET FILE NET SEND NET USER
NET GROUP NET SESSION NET VIEW

NET HELP SERVICES - эта команда выводит список служб, которые можно запустить.
NET HELP SYNTAX - эта команда выводит объяснения синтаксических правил, используемых при описании команд в Справке.
NET HELP имя_команды | MORE - просмотр справки по одному экрану за раз.

При описании команды **NET** используются следующие синтаксические соглашения:

- заглавными буквами набраны слова, которые должны быть введены без изменений, строчными буквами набраны имена и параметры, которые могут изменяться, например, имена файлов;

- необязательные параметры заключены в квадратные скобки [ ];

- списки допустимых параметров заключены в фигурные скобки { }. Необходимо использовать один из элементов такого списка;

- символ | (вертикальная черта) используется в качестве разделителя элементов списка. Возможно использование только одного из элементов списка. Например, в соответствии с изложенными соглашениями, необходимо ввести NET COMMAND и один из переключателей - SWITCH1 или SWITCH2. Указанное в квадратных скобках имя [name] является необязательным параметром:
NET COMMAND [name] {SWITCH1 | SWITCH2};

- запись [...] означает, что указанный элемент может повторяться. Повторяющиеся элементы должны быть разделены пробелом;

- запись [,...] означает, что указанный элемент может повторяться, но повторяющиеся элементы должны быть разделены запятой или точкой с запятой, но не пробелом;

- при вводе в командной строке можно использовать русские названия служб, при этом они должны быть заключены в кавычки и не допускается изменение прописных букв на строчные и наоборот. Например, команда
NET START "Обозреватель сети"
запускает службу обозревателя сети.

Справочная система NET.EXE, пожалуй, является одной из лучших в семействе операционных систем Windows. Подробную справку по использованию нужной команды, например **use** , можно получить несколькими способами:

net use ? - справка о синтаксисе команды;
net use /help - подробная справка по использованию команды с описанием используемых ключей;
net help use - аналогично предыдущей форме вызова справки;
net help use | more - отобразить справку в постраничном режиме выдачи на экран. Удобно пользоваться в тех случаях, когда тест не помещается на экране. Нажатие **Enter** перемещает текст на одну строку, нажатие пробела - на один экран;
net help use > C:\helpuse.txt - создать текстовый файл справки C:\helpuse.txt

Работа с системными службами

Данный режим использования **NET.EXE** , в некоторой степени, является не характерным для основного предназначения утилиты, и начиная с Windows XP, для управления системными службами

используется специальная утилита командной строки **SC.EXE**. Тем не менее, NET.EXE в среде любой версии операционных систем Windows может быть использована для запуска и остановки системных служб (сервисов). Согласно справочной информации, список служб, которыми можно управлять с помощью **net.exe,** можно получить, используя следующую команду:

net help services

С помощью **net.exe** можно запустить или остановить практически любую системную службу, и в том числе, не представленную в списке, отображаемом при выполнении данной команды. Для остановки используется параметр **stop**, а для запуска - параметр **start**:

net stop dnscache - остановить службу **dnscache**
net start dnscache - запустить службу **dnscache**

Возможно использование как короткого, так и полного имени ("Dnscache" - короткое, "DNS-клиент" - полное имя службы). Имя службы, содержащее символы русского алфавита и пробелы, заключается в двойные кавычки.

net stop "DNS-клиент" - остановить службу **DNS-клиент** .

Полное имя службы можно скопировать из "Панель управления" - "Администрирование" - "Службы" - Имя службы - "Свойства" - "Выводимое имя".

Для приостановки некоторых системных служб или продолжения работы ранее приостановленной службы используются команды **NET PAUSE** и **NET CONTINUE** :
net pause "Планировщик заданий" - приостановить службу "Планировщик заданий"
net continue schedule - продолжить работу службы "Планировщик заданий" . Имя службы задано в коротком формате.

Работа с сетевыми дисками

net use - отобразить список сетевых дисков, подключенных на данном компьютере.

```
Состояние    Локальный    Удаленный           Сеть
-------------------------------------------------------------------------------
Отсоединен   X:           \\SERVER\movies     Microsoft Windows Network
OK           Y:           \\SERVER\shares     Microsoft Windows Network
```

В колонке "Локальный" отображается буква сетевого диска, а в колонке "Удаленный" - имя удаленного сетевого ресурса в формате **UNC**
UNC - это Общее соглашение об именах (Uniform Naming Convention) или универсальное соглашение об именовании (universal naming convention), соглашение об именовании файлов и других ресурсов, дающее определение местоположения ресурса .
Имя, соответствующее UNC - полное имя ресурса в сети, включающее имя сервера и имя совместно используемого (разделяемого, сетевого) ресурса (принтера, каталога или файла).
Синтаксис UNC-пути к каталогу или файлу следующий:
\\Сервер\СетевойКаталог[\ОтносительныйПуть]
Сервер - сетевое имя компьютера, **СетевойКаталог** - это сетевое имя общего каталога на этом компьютере, а необязательный **ОтносительныйПуть** - путь к каталогу или файлу из общего каталога.
СетевойКаталог не обязательно называется так же, как ассоциированный с ним каталог на сервере, имя даётся в ходе открытия общего доступа к каталогу в файловой системе компьютера

В операционных системах семейства Windows, если в конце имени разделяемого ресурса используется знак **$**, то такой ресурс является скрытым и не отображается в проводнике при просмотре сетевого окружения. Это правило относится не только к автоматически создаваемым ресурсам для системного администрирования (C$, D$, ADMIN$ и т.п.), но и для любого пользовательского разделяемого ресурса. Если, например, для сетевого доступа выделена папка под именем "movies", то она будет видна в сетевом окружении, а если - под именем "movies$", то нет.

Для того чтобы скрыть в сетевом окружении отдельный компьютер используется команда:
NET config server /hidden:yes
Чтобы вернуть отображение компьютера в сетевом окружении
NET config server /hidden:no

UNC-пути можно использовать и для локальной машины, только в этом случае вместо имени "Сервер" нужно подставлять знак "?" или ".", а путь к файлу указывать вместе с буквой диска. Например так: "\\?\C:\Windows\System32\file.exe" .

Для отключения сетевого диска или устройства используется

команда **net use** с ключом **/DELETE**

net use X: /delete - отключить сетевой диск X:

Регистр букв в данном ключе не имеет значения и можно использовать сокращения:

net use Y: /del

Работа с файлами и каталогами

NET SHARE - эта команда позволяет выделить ресурсы системы для сетевого доступа. При запуске без других параметров, выводит информацию обо всех ресурсах данного компьютера, которые могут быть совместно использованы. Для каждого ресурса выводится имя устройства или путь и соответствующий комментарий.

net share - получить список разделяемых в локальной сети ресурсов данного компьютера. Пример списка:

```
Общее имя              Ресурс                  Заметки

-------------------------------------------------------------------------------
G$                     G:\                     Стандартный общий ресурс
E$                     E:\                     Стандартный общий ресурс
IPC$                                           Удаленный IPC
 ADMIN$                C:\WINDOWS              Удаленный Admin
INSTALL                C:\INSTALL              Дистрибутивы и обновления
```

net share INSTALL - получить информацию о разделяемом ресурсе с именем INSTALL .

```
Имя общего ресурса     INSTALL
Путь                   C:\INSTALL
Заметки                Дистрибутивы и обновления
Макс. число пользователей  Не ограничен
Пользователи           Administrator
Кэширование            Вручную
```

Для добавления нового разделяемого по сети ресурса используется

параметр **/ADD**

net share TEMP="C:\Documents And Settings\LocalSettings\games" - добавить новый разделяемый каталог под именем **TEMP**

net share TEMP="C:\Documents And Settings\LocalSettings\games" /users:5 - добавить новый разделяемый каталог под именем **TEMP** с максимальным числом одновременно подключающихся пользователей равным 5 .

Кроме этого, при создании разделяемого ресурса можно указать краткое его описание (заметку) с помощью параметра /REMARK и режим кэширования файлов с помощью параметра /CACHE .

NET SHARE имя_ресурса=диск:путь [/USERS:число | /UNLIMITED] [/REMARK:"текст"] [/CACHE:Manual | Automatic | No] [/CACHE:Manual | Documents| Programs | None]

Для удаления существующего разделяемого ресурса используется параметр /DELETE:

net share TEMP /DELETE - удалить разделяемый ресурс под именем TEMP.

Удаление выполняется только для имени разделяемого ресурса и не затрагивает каталог локального диска, связанный с данным именем.

Для работы с файлами, открытыми по сети на данном компьютере, используется команда **NET FILE** . По каждому открытому ресурсу выводится идентификационный номер, путь файла, имя пользователя, которым используется файл, и количество блокировок при совместном использовании. Кроме того, команда **NET FILE** позволяет закрыть совместно используемый файл и снять блокировки .

net file - получить список открытых по сети файлов .

net file 4050 /close - принудительно закрыть файл, идентификатор которого равен 4050.

Для получения списка компьютеров рабочей группы или домена с разделяемыми ресурсами используются команды:

net view - отобразить список компьютеров в сетевом окружении;

net view | more - отобразить список компьютеров в постраничном режиме вывода на экран;

net view > C:\computers.txt - отобразить список компьютеров с записью результатов в текстовый файл.

Синтаксис данной команды:

NET VIEW [\\имя_компьютера [/CACHE] | /DOMAIN[:имя_домена]]
NET VIEW /NETWORK:NW [\\имя_компьютера]

net view \\server - отобразить список сетевых ресурсов компьютера server;

net view /DOMAIN:mydomain - отобразить список компьютеров с разделяемыми ресурсами в домене **mydomain** Если имя домена не указано, то выводится список всех доступных компьютеров локальной сети;

net view /NETWORK:NW - отобразить список серверов Novell Netware, доступных в данной локальной сети;

net view /NETWORK:NW \\NWServer - отобразить список сетевых ресурсов сервера Netware с именем NWServer .

Работа с пользователями и компьютерами

Утилита NET.EXE позволяет отобразить данные об учетных записях пользователей и групп, добавлять новые записи, удалять

существующие, отображать параметры безопасности, связанные с авторизацией пользователей и некоторые другие операции по администрированию на локальном компьютере или контроллере домена.

NET ACCOUNTS - эта команда используется для обновления базы данных регистрационных записей и изменения параметров входа в сеть (LOGON). При использовании этой команды без указания параметров выводятся текущие значения параметров, определяющих требования к паролям и входу в сеть, - время принудительного завершения сессии, минимальную длину пароля, максимальное и минимальное время действия пароля и его уникальность.

Синтаксис данной команды:

NET ACCOUNTS [/FORCELOGOFF:{минуты | NO}] [/MINPWLEN:длина] [/MAXPWAGE:{дни | UNLIMITED}] [/MINPWAGE:дни] [/UNIQUEPW:число] [/DOMAIN]

Пример отображаемой информации по команде NET ACCOUNTS :

```
Принудительный выход по истечении времени через: Никогда
Минимальный срок действия пароля (дней):        0
Максимальный срок действия пароля (дней):       42
Минимальная длина пароля:               0
Хранение неповторяющихся паролей:       Нет
Блокировка после ошибок ввода пароля:     Никогда
Длительность блокировки (минут):            30
Сброс счетчика блокировок через (минут):      30
Роль компьютера:                РАБОЧАЯ СТАНЦИЯ
```

При использовании в локальной сети каждый компьютер может выполнять как роль сервера (server), предоставляющего свои ресурсы для совместного использования, так и рабочей станции (workstation), использующей разделяемые сетевые ресурсы. Основные настройки сетевых служб сервера и рабочих станций можно отобразить с помощью команд:

net config server - настройки сетевых служб для роли сервера;
net config workstation - настройки сетевых служб для роли рабочей

станции.

Настройки служб сервера можно изменить с использованием параметров:

/AUTODISCONNECT:минуты - максимальное время, в течение которого сеанс пользователя может быть не активен, прежде чем соединение будет отключено. Можно использовать значение -1, которое означает, что отключение вообще не производится. Допустимый диапазон значений: от -1 до 65535; по умолчанию используется 15.
/SRVCOMMENT:"текст"
Добавляет текст комментария для сервера, который отображается на экране Windows и при выполнении команды NET VIEW. Максимальная длина этого текста составляет 48 знаков. Текст должен быть заключен в кавычки.
/HIDDEN:{YES | NO} Указывает, должно ли выводиться имя данного сервера в списке серверов. Учтите, что "скрытие" сервера не изменяет параметров доступа к этому серверу. По умолчанию используется значение NO.
net config server /SRVCOMMENT:"Игровой сервер"
/AUTODISCONNECT:5 - автоотключение при неактивности пользователя - 5 минут.
net config server /HIDDEN:YES>/AUTODISCONNECT:-1 - автоотключение при неактивности пользователя не выполняется, сервер не отображается в сетевом окружении.

При выполнении на контроллере домена утилита net.exe позволяет добавлять новые компьютеры в базу данных Active Directory (AD) или удалять существующие компьютеры из нее.

net computer \\notebook /add - добавить в домен компьютер notebook.
net computer \\notebook /del - удалить из домена компьютер notebook
.

Для просмотра списка групп пользователей и изменения их состава, а также добавления новых или удаления существующих групп используются команды **NET GROUP** и **NET LOCALGROUP**.

Первая из них используется только на контроллерах домена и предназначена для работы с группами пользователей в домене.

net group - отобразить список групп пользователей в текущем домене.
net localgroup - отобразить список групп пользователей данного компьютера.
Синтаксис и назначение параметров этих команд практически не отличаются.

NET LOCALGROUP [имя_группы [/COMMENT:"текст"]] [/DOMAIN]
имя_группы {/ADD /COMMENT:"текст"] | /DELETE} [/DOMAIN]
имя_группы имя [...] {/ADD | /DELETE} [/DOMAIN]

имя_группы - имя локальной группы, которую необходимо добавить, изменить или удалить. Если указать только имя группы, то будет выведен список пользователей или глобальных групп, являющихся членами этой локальной группы.
/COMMENT:"текст" - комментарий для новой или существующей группы. Текст должен быть заключен в кавычки.
/DOMAIN - Команда выполняется на основном контроллере домена в текущем домене. В противном случае операция выполняется на локальном компьютере.
имя [...] - Список из одного или нескольких имен пользователей, которые необходимо добавить или удалить из локальной группы. Имена разделяются пробелом. Эти имена могут быть именами пользователей или глобальных групп, но не именами других локальных групп. Если пользователь зарегистрирован в другом домене, его имени должно предшествовать имя домена (например, SALES\RALPHR).
/ADD - Добавляет имя группы или имя пользователя в локальную группу. Регистрационная запись для добавляемых пользователей или глобальных групп должна быть создана заранее.
/DELETE - Удаляет имя группы или пользователя из локальной группы.

Для работы с учетными записями пользователей используется команда **net user**

NET USER [имя_пользователя [пароль | *] [параметры]] [/DOMAIN]
имя_пользователя {пароль | *} /ADD [параметры] [/DOMAIN]
имя_пользователя [/DELETE] [/DOMAIN]

имя_пользователя - имя пользователя, которое необходимо добавить, удалить, изменить или вывести на экран. Длина имени пользователя не должна превосходить 20 знаков.
пароль - пароль для учетной записи пользователя. Пароль должен отвечать установленным требованиям на длину - быть не короче, чем значение, установленное параметром /MINPWLEN в команде NET ACCOUNTS, и в то же время не длиннее 14 знаков.
***** - Вызывает открытие специальной строки ввода пароля. Пароль не выводится на экран во время его ввода в этой строке.

/DOMAIN команда будет выполняться на контроллере домена в текущем домене.
/ADD - добавление нового пользователя.
/DELETE - удаление пользователя.
Параметры - Допустимые параметры :
/ACTIVE:{YES | NO} - Активизирует учетную запись или делает ее не активной. Если учетная запись не активна, пользователь не может получить доступ к серверу. По умолчанию используется значение YES (т.е. учетная запись активна).
/COMMENT:"текст" - Добавляет описательный комментарий об учетной записи (длиной не более 48 знаков). Текст должен быть заключен в кавычки.
/COUNTRYCODE:nnn - Использует кодовую страницу нужного языка для вывода справки и сообщений об ошибках. Значение 0 означает выбор кодовой страницы по умолчанию.
/EXPIRES:{дата | NEVER} - Устанавливает дату истечения срока действия ученой записи. Если используется значение NEVER, то время действия учетной записи не ограничено. Дата истечения срока действия задается в формате дд/мм/гг или мм/дд/гг, в зависимости от того, какая кодовая страница используется. Месяц может быть указан цифрами, названием месяца или трехбуквенным его сокращением. В качестве разделителя полей должен использоваться знак косой черты (/).
/FULLNAME:"имя" - Указывает настоящее имя пользователя (а не кодовое имя, заданное параметром имя_пользователя). Настоящее имя следует заключить в кавычки.
/HOMEDIR:путь Указывает путь к домашнему каталогу пользователя. Этот каталог должен существовать.
/PASSWORDCHG:{YES | NO} Определяет, может ли пользователь изменять свой пароль. По умолчанию используется значение YES (т.е. изменение пароля разрешено).
/PASSWORDREQ:{YES | NO} Определяет, является ли указание пароля обязательным. По умолчанию используется значение YES (т.е. пароль обязателен).
/PROFILEPATH[:путь] Устанавливает путь к профилю пользователя.
/SCRIPTPATH:путь Устанавливает расположение пользовательского сценария для входа в систему.
/TIMES:{промежуток | ALL} - Устанавливает промежуток времени, во время которого пользователю разрешен вход в систему. Этот параметр задается в следующем формате:
день[-день][,день[-день]],время[-время][,время[-время]]
Время указывается с точностью до одного часа. Дни являются днями недели и могут указываться как в полном, так и в сокращенном виде. Время можно указывать в 12- и 24-часовом формате. Если используется 12-часовой формат, то можно использовать am, pm, a.m. или p.m. Значение ALL указывает, что пользователь может войти в систему в любое время, а пустое значение указывает, что пользователь не может войти в систему никогда. Разделителем полей указания дней недели и времени является запятая, разделителем при использовании нескольких частей является точка с запятой.
/USERCOMMENT:"текст" - Позволяет администратору добавлять или изменять текст комментария к учетной записи. **/WORKSTATIONS:{имя_компьютера[,...] | *}** - Перечисляет до восьми различных компьютеров, с которых пользователь может войти в сеть. Если данный параметр имеет пустой список или указано значение *, пользователь может войти в сеть с любого компьютера.

Отправка сообщений по локальной сети

Для отправки сообщений в Windows XP используется команда
NET SEND

NET SEND {имя | * | /DOMAIN[:имя] | /USERS} сообщение

имя - имя пользователя, компьютера или имя для получения сообщений, на которое отправляется данное сообщение. Если это имя содержит пробелы, то оно должно быть заключено в кавычки (" ").
***** - отправка сообщения по всем именам, которые доступны в данный момент.
/DOMAIN[:имя домена] - сообщение будет отправлено по всем именам домена данной рабочей станции. Если указано имя домена, то сообщение отправляется по всем именам указанного домена или рабочей группы.
/USERS - сообщение будет отправлено всем пользователям, подключенным в настоящий момент к серверу.
сообщение - текст отправляемого сообщения.

Для того чтобы получить сообщение, должна быть запущена "Служба сообщений" (MESSENGER). Имена пользователей, компьютеров и текст сообщений на русском языке должны быть в DOS-кодировке.
Перечень доступных активных имен на данном компьютере и состояние службы сообщений можно получить с использованием команды **net name** без параметров. По всему списку имен, отображаемому в результате выполнения данной команды, возможна отправка сообщений.

В операционных системах Windows 7/8 команда **net send** не реализована и для обмена сообщениями в локальной сети используется команда **msg**. Такая же команда существует и в операционных системах WindowsXP/Server 2003, но используется в них только для обмена сообщениями с пользователями терминальных сессий. Тем не менее, при определенных настройках службы сервера **Terminal Server** команда **msg** может использоваться для обмена сообщениями между пользователями Windows XP и более поздних версий Windows. Для этого необходимо на каждом компьютере, которому будут отправляться сообщения, разрешить удаленный вызов процедур для службы сервера терминалов, добавив в раздел реестра **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\ Terminal Server** параметр **AllowRemoteRPC** типа REG_DWORD и равный **1**. Для вступления данного значения в силу требуется перезагрузка. После чего команду **msg** можно будет использовать как альтернативу **net send** на компьютерах с любой версией Windows.

Необходимо также учитывать тот факт, что потребуются настройки брандмауэров, поскольку по умолчанию, передача и прием сообщений по сети, как правило, блокируются.

Справку по работе с командой **msg**. можно получить, введя **/?** в качестве параметра:

MSG {<пользователь> | <имя сеанса> | | @<имя файла> | *}
[/SERVER:<сервер>] [/TIME:<секунд>] [/V] [/W] [<сообщение>]

<пользователь> Имя пользователя.
<имя сеанса> Имя сеанса.
Идентификатор сеанса.
@<имя файла> Файл, содержащий список имен пользователей, сеансов или идентификаторов сеансов, которым отправляется сообщение.
***** Отправить сообщение всем сеансам на указанном сервере.
/SERVER:<сервер> Сервер (по умолчанию - текущий).
/TIME:<секунд> Интервал ожидания подтверждения от получателя.
/V Отображение информации о выполненных действиях.
/W Ожидание ответа от пользователя, полезно вместе с /V.
<сообщение> Отправляемое сообщение. Если не указано, выдается запрос или принимается ввод из STDIN.

Статистика и синхронизация часов

Утилита NET.EXE позволяет получить статистические данные по использованию служб сервера и рабочей станции. Статистика содержит информацию о сеансах, доступе к сетевым устройствам, объемах принятых и переданных данных, отказах в доступе и ошибках, обнаруженных в процессе сетевого обмена.

net statistics server - отобразить статистические данные для службы сервера
net statistics workstation - отобразить статистические данные для службы рабочей станции

Для изменения системного времени компьютера используется команда **NET TIME** :

NET TIME [\\компьютер | /DOMAIN[:домен]| /RTSDOMAIN[:домен]] [/SET] [\\компьютер] /QUERYSNTP [\\компьютер] /SETSNTP[:список серверов NTP]

NET TIME синхронизирует показания часов компьютера с другим компьютером или доменом. Если используется без параметров в домене Windows Server, выводит текущую дату и время дня, установленные на компьютере, который назначен сервером времени для данного домена. Эта команда позволяет задать сервер времени NTP для компьютера.

\\компьютер - имя компьютера, который нужно проверить или с которым нужно синхронизировать показания часов.
/DOMAIN[:домен] Задает домен, с которым нужно синхронизировать показания часов.
/RTSDOMAIN[:домен] - выполняет синхронизацию времени с сервером времени (Reliable Time Server) из указанного домена.
/SET - Синхронизирует показания часов компьютера со временем указанного компьютера или домена.
/QUERYSNTP - Отображает назначенный этому компьютеру сервер NTP (только Windows XP)
/SETSNTP[:ntp server list] - задать список серверов времени NTP для этого компьютера (только Windows XP).
Это может быть список IP-адресов или DNS-имен, разделенных пробелами. Если задано несколько серверов, список должен быть заключен в кавычки.

Параметры /QUERYSNTP и /SETSNTP не поддерживаются в операционных системах Windows 7 и более поздних. Для настройки службы времени в этих ОС используется утилита **w32tm.exe**

Утилита NSLOOKUP.EXE

Утилита **NSLOOKUP** присутствует во всех версиях операционных систем Windows и является классическим средством диагностики сетевых проблем, связанных с разрешением доменных имен в IP-адреса. NSLOOKUP предоставляет пользователю возможность просмотра базы данных DNS-сервера и построения определенных запросов, для поиска нужных ресурсов DNS. Практически утилита выполняет функции службы DNS-клиент в командной строке Windows.

После запуска, утилита переходит в режим ожидания ввода. Ввод символа **?** или команды **help** позволяет получить подсказку по использованию утилиты.

Примеры использования:

nslookup - запуск утилиты
yandex.ru. - отобразить IP-адрес (а) узла с именем yandex.ru . Точка в конце имени желательна для минимизации числа запросов на разрешение имени к серверу DNS. Если завершающей точки нет, то NSLOOKUP сначала попытается разрешить указанное имя как часть доменного имени компьютера, на котором она запущена.
server 8.8.4.4 - установить в качестве сервера имен DNS-сервер Google с IP-адресом 8.8.4.4
yandex.ru. - повторить запрос с использованием разрешения имени DNS-сервером Google
set type=MX - установить тип записи MX
yandex.ru. - отобразить MX-запись для домена yandex.ru - В примере узел обмена почтой для домена - mx.yandex.ru
mx.yandex.ru. - отобразить информацию по mx.yandex.ru
set type=A - установить тип записи в **A**
mx.yandex.ru - получить IP-адреса для mx.yandex.ru
exit - завершить работу с nslookup

Возможно использование утилиты NSLOOKUP не в интерактивном режиме:

nslookup odnoklassniki.ru - определить IP-адрес узла odnokassniki.ru с использованием сервера DNS, заданного настройками сетевого подключения
nslookup odnoklassniki.ru 8.8.8.8 - определить IP-адрес узла odnokassniki.ru с использованием DNS-сервера 8.8.8.8 (публичный DNS-сервер Google)
nslookup 8.8.8.8 - определить имя узла, IP-адрес которого равен 8.8.8.8 с использованием DNS-сервера, заданного настройками сетевого подключения

Утилита PATHPING.EXE

Команда **PATHPING** выполняет трассировку маршрута к конечному узлу аналогично команде **TRACERT** , но дополнительно выполняет отправку ICMP эхо-запросов на промежуточные узлы маршрута для сбора информации о задержках и потерях пакетов на каждом из них.

При запуске **PATHPING** без параметров отображается краткая справка:

pathping [-g Список] [-h Число_прыжков] [-i Адрес] [-n] [-p Пауза] [-q Число_запросов] [-w Таймаут] [-P] [-R] [-T] [-4] [-6] узел

Параметры:
-g Список При прохождении по элементам списка узлов игнорировать предыдущий маршрут. Максимальное число адресов в списке равно 9 . Элементы списка помещаются в

специальное поле заголовка отправляемых ICMP-пакетов.
-h Число_прыжков - Максимальное число прыжков при поиске узла. Значение по умолчанию – 30.
-i Адрес - Использовать указанный адрес источника в отправляемых ICMP-пакетах.
-n - Не разрешать адреса в имена узлов.
-p Пауза - Пауза между отправками (мсек) пакетов. Значение по умолчанию - 250.
-q Число_запросов Число запросов для каждого узла. По умолчанию – 100.
-w Таймаут - Время ожидания каждого ответа (мсек). Значение по умолчанию – 3000.
-R - Тестировать возможность использования RSVP (Reservation Protocol, протокола настройки резервирования ресурсов), который позволяет динамически выделять ресурсы для различных видов трафика.
-T - Тестировать на возможность использования QoS (Quality of Service - качество обслуживания) - системы обслуживания пакетов разного содержания с учетом их приоритетов доставки получателю.
-4 - Принудительно использовать IPv4.
-6 - Принудительно использовать IPv6.

Практически **PATHPING**, запущенная на выполнение с параметрами по умолчанию, выполняет те же действия, что и команда **TRACERT** плюс команды **PING** для каждого промежуточного узла с указанием числа эхо-запросов, равным 100 (ping -n 100 . . .)

Утилита PING.EXE

PING.EXE - это, наверно, наиболее часто используемая сетевая утилита командной строки. Существует во всех версиях всех операционных систем с поддержкой сети и является простым и удобным средством опроса узла по имени или его IP-адресу. Для обмена служебной и диагностической информацией в сети используется специальный протокол управляющих сообщений **ICMP** (Internet Control Message Protocol). Команда **ping** позволяет выполнить отправку управляющего сообщения типа **Echo Request** (тип равен 8 и указывается в заголовке сообщения) адресуемому узлу и интерпретировать полученный от него ответ в удобном для анализа виде. В поле данных отправляемого icmp-пакета обычно содержатся символы английского алфавита. В ответ на такой запрос, опрашиваемый узел должен отправить icmp-пакет с теми же данными, которые были приняты, и типом сообщения **Echo Reply** (код типа в заголовке равен 0). Если при обмене icmp-сообщениями возникает какая-либо проблема, то утилита ping выведет информацию для ее диагностики.

Формат командной строки:

ping [-t] [-a] [-n число] [-l размер] [-f] [-i TTL] [-v TOS] [-r число] [-s число] [[-j списокУзлов] | [-k списокУзлов]] [-w таймаут] конечноеИмя

Параметры:

-t - Непрерывная отправка пакетов. Для завершения и вывода статистики используются комбинации клавиш **Ctrl + Break** (вывод статистики) и **Ctrl + C** (вывод статистики и завершение).
-a - Определение адресов по именам узлов. **-n число** - Число отправляемых эхо-запросов.
-l размер - Размер поля данных в байтах отправляемого запроса.
-f - Установка флага, запрещающего фрагментацию пакета.
-i TTL - Задание срока жизни пакета (поле "Time To Live").
-v TOS - Задание типа службы (поле "Type Of Service").
-r число - Запись маршрута для указанного числа переходов.
-s число - Штамп времени для указанного числа переходов.
-j списокУзлов - Свободный выбор маршрута по списку узлов.
-k списокУзлов - Жесткий выбор маршрута по списку узлов.
-w таймаут - Максимальное время ожидания каждого ответа в миллисекундах.

Утилита ROUTE.EXE

Утилита **ROUTE.EXE** используется для просмотра и модификации таблицы маршрутов на локальном компьютере. При запуске без параметров на экран выводится подсказка по использованию **route**:

route [-f] [-p] [команда [конечная_точка] [mask маска_сети] [шлюз] [metric метрика]] [if интерфейс]]

-f - используется для сброса таблицы маршрутизации. При выполнении команды **route -f** из таблицы удаляются все маршруты, которые не относятся к петлевому интерфейсу (IP 127.0.0.1 маска -255.0.0.0), не являются маршрутами для многоадресной (multicast) рассылки (IP 224.0.0.1 маска 255.0.0.0) и не являются узловыми маршрутами (маска равна 255.255.255.255) .
-p - используется для добавления в таблицу постоянного маршрута. Если маршрут добавлен без использования параметра **–p,** то он сохраняется только до перезагрузки системы (до перезапуска сетевого системного программного обеспечения). Если же, при добавлении маршрута искользовался данный параметр, то информация о маршруте записывается в реестр Windows (раздел HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\PersistentRoutes) и будет использоваться постоянно при активации сетевых интерфейсов.
команда - козможно использование команд **add** - добавление маршрута, **change** - изменение существующего маршрута, **delete** - удаление маршрута или маршрутов, **print** - отображение

текущей таблицы маршрутов.
конечная_точка - IP-адрес, адрес сети или адрес 0.0.0.0 для шлюза по умолчанию.
mask маска_сети - маска сети.
шлюз - IP-адрес шлюза, через который будет выполняться отправка пакета для достижения конечной точки.
metric число - значение метрики (1-9999). Метрика представляет собой числовое значение, позволяющее оптимизировать доставку пакета получателю, если конечная точка маршрута может быть достижима по нескольким разным маршрутам. Чем меньше значение метрики, тем выше приоритет маршрута.
if интерфейс - идентификатор сетевого интерфейса. Может задаваться в виде десятичного или шестнадцатеричного числа. Посмотреть идентификаторы можно с помощью команды **route print**

Утилита TRACERT.EXE

Несмотря на появление утилиты **PATHPIG**, классическая утилита трассировки маршрута до заданного узла **TRACERT** , по-прежнему остается наиболее часто используемым инструментом сетевой диагностики. Утилита позволяет получить цепочку узлов, через которые проходит IP-пакет, адресованный конечному узлу. В основе трассировки заложен метод анализа ответов при последовательной отправке ICMP-пакетов на указанный адрес с увеличивающимся на 1 полем TTL. ("Время жизни" - Time To Live). На самом деле это поле не имеет отношения к времени, а является счетчиком числа возможных переходов при передаче маршрутизируемого пакета. Каждый маршрутизатор, получив пакет, вычитает из этого поля 1 и проверяет значение счетчика TTL. Если значение стало равным нулю, такой пакет отбрасывается и отправителю посылается ICMP-сообщение о превышении времени жизни ("Time Exceeded" - значение 11 в заголовке ICMP). Если бы не было предусмотрено включение поля TTL в IP-пакетах, то при ошибках в маршрутах могла бы возникнуть ситуация, когда пакет будет вечно циркулировать в сети, пересылаемый маршрутизаторами по кругу. При выполнении команды tracert.exe сначала выполняется отправка ICMP-пакета с полем TTL, равным **1,** и первый в цепочке маршрутизатор (обычно это основной шлюз из настроек сетевого подключения), вычтя единицу из TTL, получает его нулевое значение и сообщает о превышении времени жизни. Эта последовательность повторяется трижды, поэтому в строке результата, формируемой tracert.exe, после номера перехода отображаются три значения времени отклика:

1 1 ms <1 <1 192.168.1.1
1 - номер перехода (1 - первый маршрутизатор)
1 ms <1 <1 - время его ответа для 3-х попыток (1ms и 2 ответа менее чем 1 ms)
192.168.1.1 - его адрес (или имя)

Затем процедура повторяется, но TTL устанавливается равным **2** - первый маршрутизатор его уменьшит до 1 и отправит следующему в цепочке, который после вычитания 1 обнулит TTL и сообщит о превышении времени жизни. И так далее, пока не будет достигнут заданный узел, имя или адрес которого заданы в качестве параметра командной строки, например, **tracert yandex.ru** , или до обнаружения неисправности, не позволяющей доставить пакет узлу yandex.ru.

Контрольные вопросы

1. Для чего используется утилита PING?
2. Как с помощью утилиты PING оценить пропускную способность сети?
3. Что такое петля маршрутизации?
4. Как выглядят правила маршрутизации, образующие петлю?
5. Зачем нужна таблица ARP?
6. Объясните разницу во времени между обращениями к одному и тому же хосту по имени и IP адресу.

Лабораторная работа № 6
ПРОГРАММИРОВАНИЕ ЛВС С ИСПОЛЬЗОВАНИЕМ ПРОТОКОЛА TCP/IP

Цель работы: рассмотреть основные методы программной реализации обмена с использованием протокола TCP/IP.

Пояснения к работе

Для создания распределенных приложений наибольшее распространение получили следующие три средства:

1) socket-интерфейс прикладной программы с модулем из состава ОС, реализующим сетевое взаимодействие;
2) интерфейс транспортного уровня (TLI - Transport Level Interface);
3) средства удаленного вызова процедур (RPC - Remote Procedure Call).

Причем socket-интерфейс и TLI обеспечивают возможность прикладным программам взаимодействовать, используя стандартные (или очень похожие на них) средства ввода-вывода ОС UNIX. RPC позволяет одной прикладной программе обращаться к другой (но функционирующей на другом узле сети) фактически так же, как одна функция традиционной программы на языке СИ обращается к другой.

Рассматриваемые средства предназначены для создания распределенных приложений, функционирующих, в первую очередь, согласно модели взаимодействия "клиент-сервер". Эта модель подразумевает, что из двух (в простейшем случае) параллельно выполняющихся на разных (а, возможно, на одном и том же) узлах сети программ одна является сервером (поставщиком услуг), способным решать некоторую задачу вычислительного или информационного характера, а другая - клиентом (потребителем услуг), который в ходе решения собственной проблемы сталкивается с необходимостью получения ответа на задачи, решаемые сервером. Программа-сервер пассивна - непосредственно к решению своей задачи она приступает только при поступлении к ней запроса от клиента, остальное же время она находится в состоянии ожидания такого запроса. После решения поставленной задачи сервер возвращает клиенту найденный ответ. В типичной ситуации программа-клиент приостанавливает свою работу после выдачи запроса серверу в ожидании получения ответа.

Одна программа-сервер может обслуживать (последовательно) несколько клиентов, организуя очередь запросов от них. Клиент и сервер могут меняться местами ролями (если того, конечно, требует логика решаемой проблемы) в разные моменты времени. Сервер, обрабатывая запрос какого-либо клиента, в свою очередь, может стать клиентом, обращаясь за некоторой услугой к другому серверу.

Двумя наиболее общими режимами взаимодействия прикладных программ в вычислительной сети являются:

1) режим с установлением соединения;

2) режим без установления соединения.

Первый режим подразумевает, что взаимодействие осуществляется в три этапа:

- установление логической связи между прикладными программами (по инициативе клиента);
- двусторонний, последовательный (потоковый, без учета каких-либо границ записей), надежный (без потери и дублирования) обмен данными;
- закрытие связи (экстренное или с доставкой буферизованных на передающей стороне данных).

В режиме без установления соединения обмен информацией ведется отдельными блоками данных, часто называемых дейтаграммами и содержащими в себе помимо собственно данных еще и адрес их получателя (соответственно, клиента или сервера). В этом режиме, как правило, не гарантируется надежность доставки данных (они могут быть потеряны или продублированы), может быть нарушена правильная последовательность дейтаграмм на принимающей стороне, но, очевидно, явно присутствуют границы в передаваемых данных.

Программист имеет возможность выбрать режим взаимодействия, отвечающий специфике создаваемого приложения.

Одним из аспектов сетевого программирования является манипулирование адресами, идентифицирующими узлы в вычислительной сети и прикладные программы на этих узлах. К сожалению, способы адресации в сетях, построенных на базе различных протоколов, также различны.

Задание к лабораторной работе

1. Написать клиент-серверную программу на основе транспортного протокола UDP. Реализовать: подтверждение приема для каждой датаграммы, сохранение целостности всей информации. Клиент передает файл или сообщение (несколькими датаграммами), сервер принимает. Продемонстрировать реализованные возможности программ согласно заданию, при одновременной передачи файлов от нескольких клиентов к серверу. Например, при запуске сервера указать, какие пакеты и сколько раз будут потеряны. Результат правильности приема выводить на экран.

2. Серверная программа должна находить номер свободного порта и выводить его на экран. При запуске клиентской программы задавать со строки IP-адрес сервера и порт.
3. Написать программу, обеспечивающую параллельную работу сервера, принимающего файлы от клиента по сети. Условие: мультипроцессная организация на основе функции fork, транспортный протокол – TCP.
4. Обеспечить в сервере завершение «зомби-процессов».
5. Написать клиентскую программу, передающую файл по сети серверу. Продемонстрировать реализованные возможности программ согласно заданию.
6. Серверная программа должна находить номер свободного порта и выводить его на экран. При запуске клиентской программы задавать со строки IP адрес сервера и порт.
7. Разработать программу однопотокового сервера, использующую асинхронный ввод/вывод (организованный с помощью системного вызова select), обеспечивающую псевдопараллельную работу клиентов.
8. Написать клиентскую программу, передающую сообщения на сервер. Продемонстрировать асинхронную работу сервера. Например, при запуске клиента пользователь задает число *i* от 1 до 10. Клиент передает серверу в цикле это число с задержкой в *i* секунд между передачей. Сервер отображает на экран полученную от клиентов информацию.
9. Написать программу, реализующую работу сервера по двум протоколам (TCP и UDP) параллельно с помощью системного вызова select.

10. Написать две клиентские программы, передающие на сервер файлы по протоколам TCP и UDP соответственно.
11. Требуется разработать клиент-серверные программы передачи данных на базе транспортного протокола SCTP [1, 2]. Обеспечить передачу информации по нескольким потокам в одном соединении или ассоциации (особенность протокола SCTP). Клиентская программа посылает в одном соединении или ассоциации: текстовое сообщение в первом потоке; файл с картинкой во втором потоке.
12. Серверная программа принимает в одном соединении или ассоциации от каждого из клиентов: текстовое сообщение с первого потока и

выводит на экран; со второго потока принимает данные и сохраняет в файл.

13. В серверной программе реализовать (на выбор) параллельную или псевдопараллельную обработку клиентов.

Контрольные вопросы

1. В чем отличие между сетевым протоколом и сетевой моделью? Зачем сетевые модели имеют многоуровневую структуру?
2. Какое назначение портов в модели TCP/IP? Могут ли две программы, запущенных на одном компьютере, прослушивать в ожидании данных один порт?
3. Что такое сервер?
4. Можно ли передавать информацию между устройствами по сети, если им не назначены IP-адреса?
5. После установки сокета в режим прослушивания, как программе среагировать на подключение клиента?
6. Почему не следует использовать декриптор сокета полученный сервером при подключении клиента в процессе обмена данными?

Лабораторная работа № 7

ИЗУЧЕНИЕ ПРОГРАММНОГО ПАКЕТА PACKET TRACER

Цель работы: познакомится с основными принципами работы в программе Cisco Packet Tracer на примере создания простой локальной вычислительной сети путем описания пошаговых инструкции по настройке.

Пояснения к работе

Cisco Packet Tracer разработан компанией Cisco и рекомендован использоваться при изучении телекоммуникационных сетей и сетевого оборудования, а также для проведения занятий по лабораторным работам в высших заведениях.

Основные возможности Packet Tracer:

- дружественный графический интерфейс (GUI), что способствует к лучшему пониманию организации сети, принципов работы устройства;

- возможность смоделировать логическую топологию: рабочее пространство для того, чтобы создать сети любого размера на CCNA-уровне сложности;
- моделирование в режиме real-time (реального времени);
- режим симуляции;
- многоязычность интерфейса программы: что позволяет изучать программу на своем родном языке;
- усовершенствованное изображение сетевого оборудования со способностью добавлять / удалять различные компоненты;
- наличие Activity Wizard позволяет сетевым инженерам, студентам и преподавателям создавать шаблоны сетей и использовать их в дальнейшем;
- проектирование физической топологии: доступное взаимодействие с физическими устройствами, используя такие понятия как город, здание, стойка и т.д.

Широкий круг возможностей данного продукта позволяет сетевым инженерам: конфигурировать, отлаживать и строить вычислительную сеть. Также данный продукт незаменим в учебном процессе, поскольку дает наглядное отображение работы сети, что повышает освоение материала учащимися.

Эмулятор сети позволяет сетевым инженерам проектировать сети любой сложности, создавая и отправляя различные пакеты данных, сохранять и комментировать свою работу. Специалисты могут изучать и использовать такие сетевые устройства, как коммутаторы второго и третьего уровней, рабочие станции, определять типы связей между ними и соединять их.

На заключительном этапе, после того как сеть спроектирована, специалист может приступать к конфигурированию выбранных устройств посредством терминального доступа или командной строки (рис. 7.1). Одной из самых важных особенностей данного симулятора является наличие в нем «Режима симуляции» (рис. 7.2). В данном режиме все пакеты, пересылаемые внутри сети, отображаются в графическом виде. Эта возможность позволяет сетевым специалистам наглядно продемонстрировать, по какому интерфейсу в данный момент перемещается пакет, какой протокол используется и т.д.

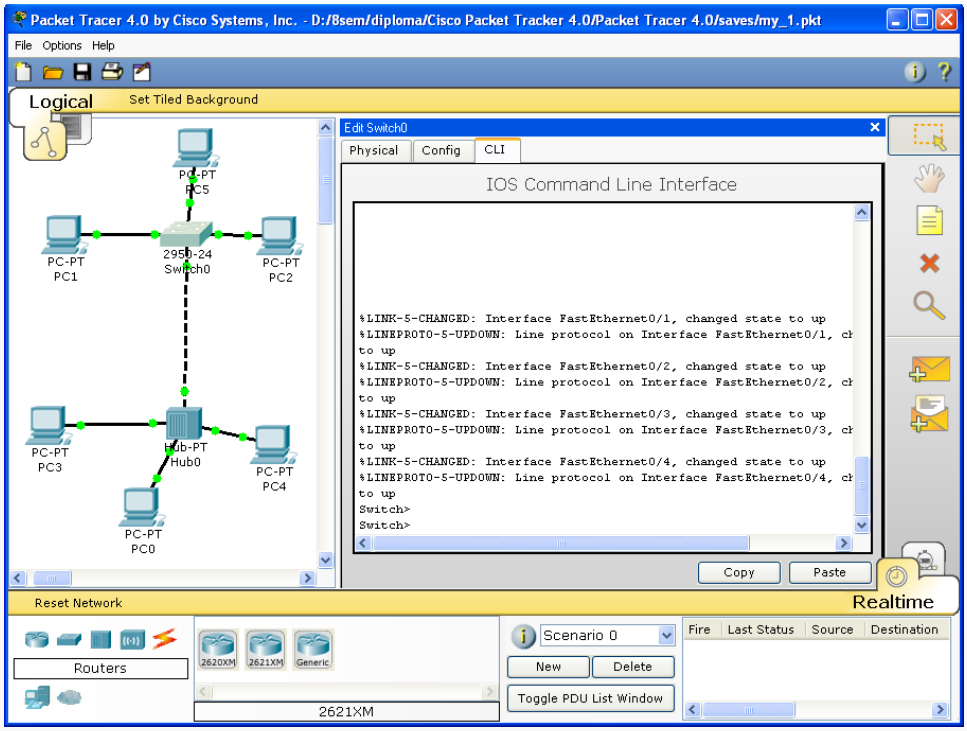


Рис. 7.1. Cisco Packet Tracer

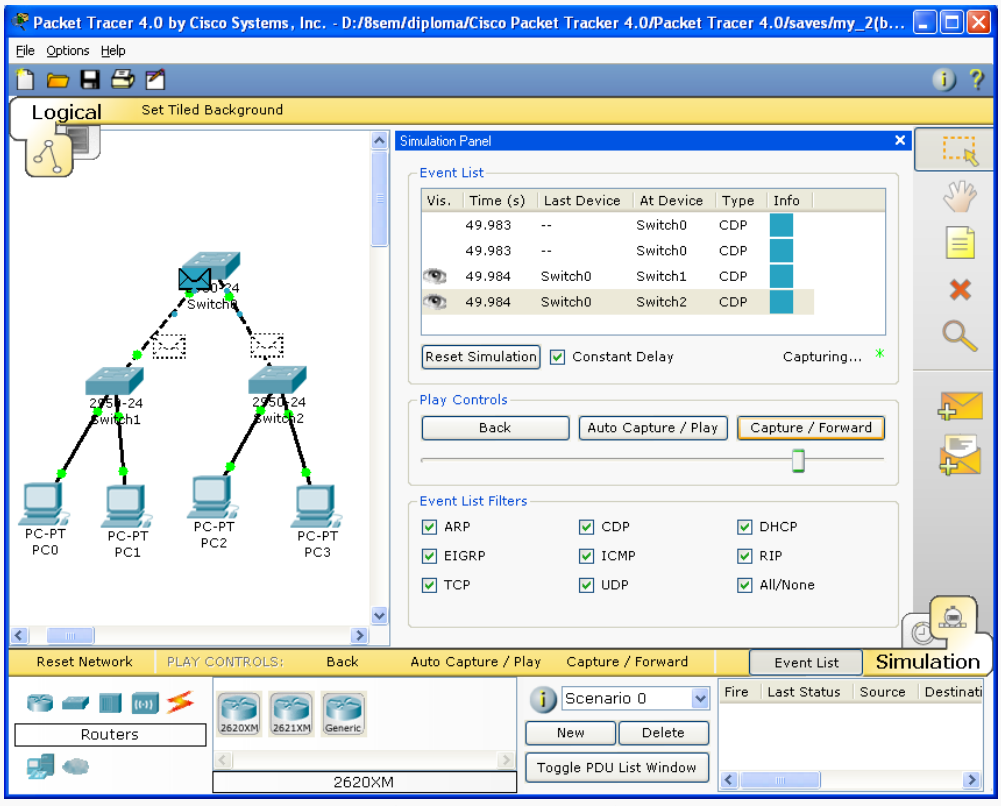


Рис. 7.2. Режим «Симуляции» в Cisco Packet Tracer

Однако это не все преимущества Packet Tracer: в «Режиме симуляции» сетевые инженеры могут не только отслеживать используемые протоколы, но и видеть, на каком из семи уровней модели OSI данный протокол задействован (рис. 7.3).

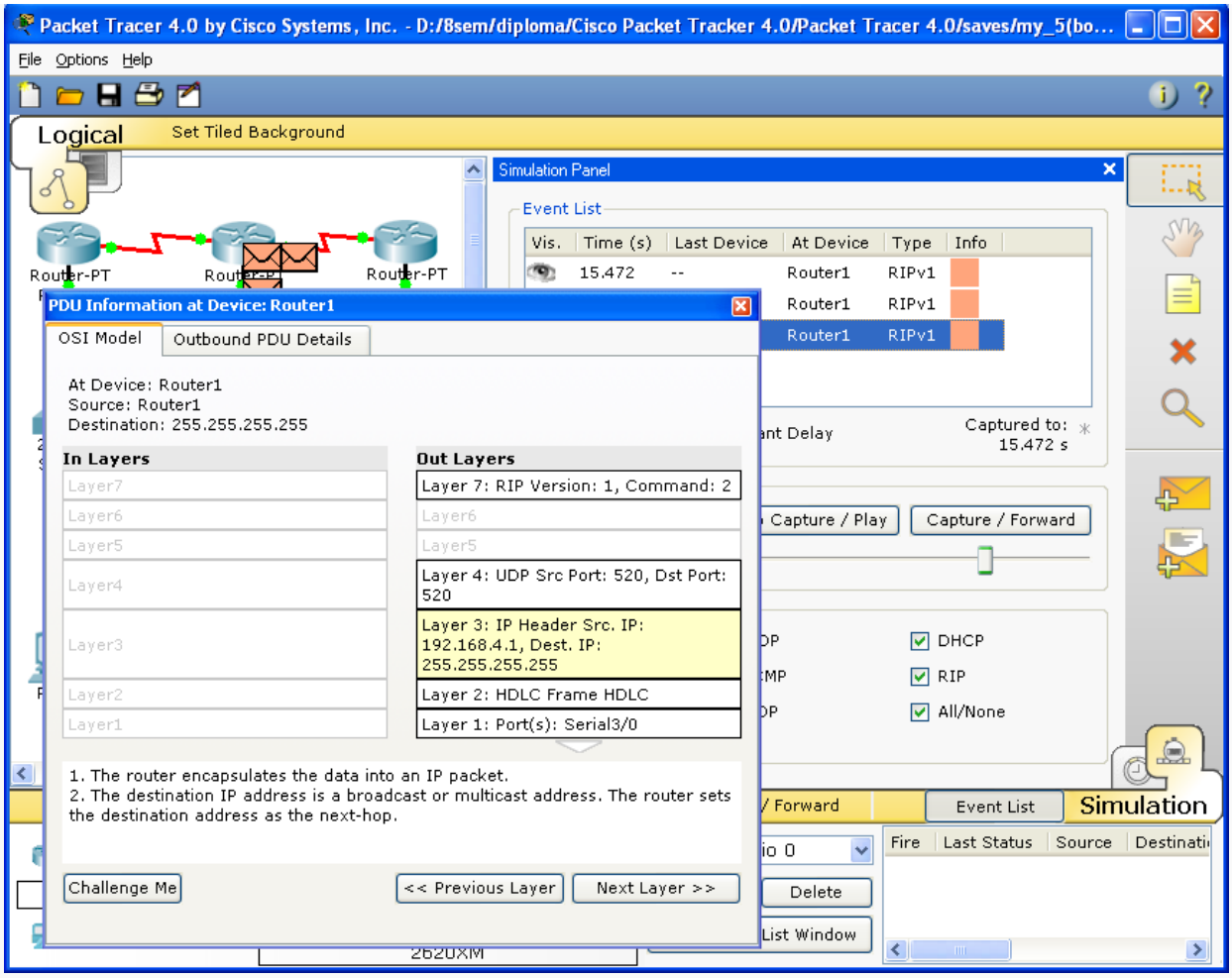


Рис. 7.3. Анализ семиуровневой модели OSI в Cisco Packet Tracer

Такая кажущаяся на первый взгляд простота и наглядность делает практические занятия чрезвычайно полезными, совмещая в них как получение, так и закрепление полученного материала.

Packet Tracer способен моделировать большое количество устройств различного назначения, а также немало различных типов связей, что позволяет проектировать сети любого размера на высоком уровне сложности.

Моделируемые устройства:

- коммутаторы третьего уровня:
  - Router 2620 XM;
  - Router 2621 XM;
  - Router-PT;
- Коммутаторы второго уровня:
  - Switch 2950-24;
  - Switch 2950T;

- Switch-PT;
- соединение типа «мост» Bridge-PT.
- Сетевые концентраторы:
- Hub-PT;
- повторитель Repeater-PT.
- Оконечные устройства:
- рабочая станция PC-PT;
- сервер Server-PT;
- принтер Printer-PT.
- Беспроводные устройства:
- точка доступа AccessPoint-PT.
- Глобальная сеть WAN.

Типы связей:

- консоль;
- медный кабель без перекрещивания (прямой кабель);
- медный кабель с перекрещиванием (кросс-кабель);
- волоконно-оптический кабель;
- телефонная линия;
- Serial DCE;
- Serial DTE.

Также целесообразно привести те протоколы, которые студент может отслеживать:

- ARP;
- CDP;
- DHCP;
- EIGRP;
- ICMP;
- RIP;
- TCP;
- UDP.

Интерфейс Cisco Packet Tracer

Интерфейс программы Cisco Packet Tracer представлен на рис. 7.4.

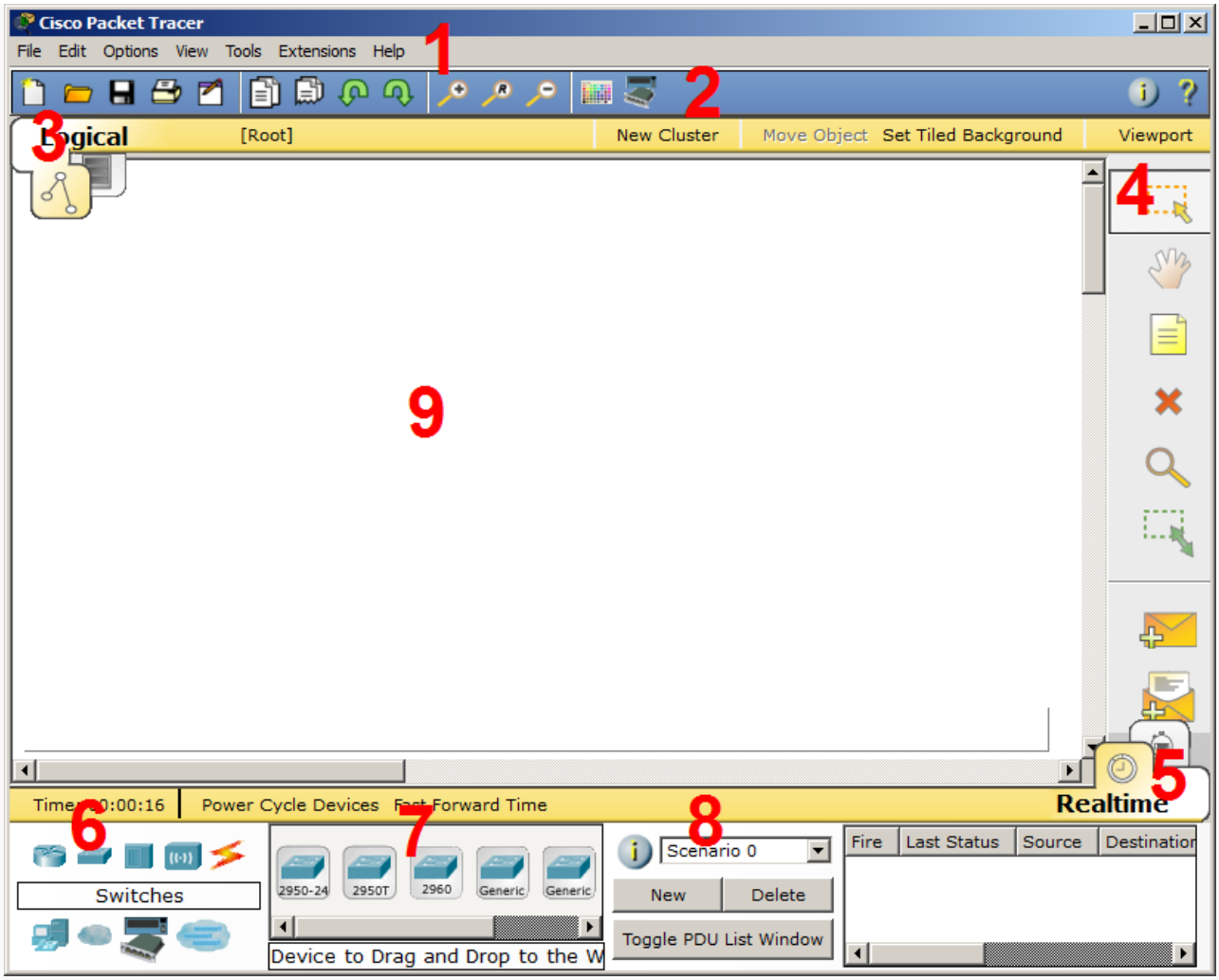


Рис. 7.4. Интерфейс программы Cisco Packet Tracer

1. Главное меню программы.
2. Панель инструментов дублирует некоторые пункты меню.
3. Переключатель между логической и физической организацией;
4. Ещё одна панель инструментов содержит инструменты выделения, удаления, перемещения, масштабирования объектов, а так же формирование произвольных пакетов.
5. Переключатель между реальным режимом (Real-Time) и режимом симуляции.
6. Панель с группами конечных устройств и линий связи.
7. Сами конечные устройства, здесь содержатся всевозможные коммутаторы, узлы, точки доступа, проводники.
8. Панель создания пользовательских сценариев.
9. Рабочее пространство;

Большую часть данного окна занимает рабочая область, в которой можно размещать различные сетевые устройства, соединять их

различными способами и как следствие получать самые разные сетевые топологии.

Сверху, над рабочей областью, расположена главная панель программы и ее меню (рис. 7.5). Меню позволяет выполнять сохранение, загрузку сетевых топологий, настройку симуляции, а также много других интересных функций. Главная панель содержит на себе наиболее часто используемые функции меню.

Рис. 7.5. Главное меню Packet Tracer

Справа от рабочей области расположена боковая панель, содержащая ряд кнопок, отвечающих за перемещение полотна рабочей области, удаление объектов и т.д.

Снизу, под рабочей областью, расположена панель оборудования (рис. 7.6).

Рис. 7.6. Панель оборудования Packet Tracer

Данная панель содержит в своей левой части типы доступных устройств, а в правой части - доступные модели. При выполнении различных лабораторных работ эту панель придется использовать намного чаще, чем все остальные. Поэтому рассмотрим ее более подробно.

При наведении на каждое из устройств в прямоугольнике, находящемся в центре между ними, будет отображаться его тип. Типы устройств, наиболее часто используемые в лабораторных работах Packet Tracer, представлены на рис. 7.7.

Рис. 7.7. Основные типы устройств

Рассматривать конкретные модели устройств каждого типа, не имеет большого смысла. Отдельного рассмотрения заслуживают типы соединений. Перечислим наиболее часто используемые из них (рассмотрение типов подключений идет слева направо, в соответствии с приведенным на рис. 7.8).

Рис. 7.8. Типы соединений устройств в Packet Tracer

- Автоматический тип – при данном типе соединения Packet Tracer автоматически выбирает наиболее предпочтительные типы соединения для выбранных устройств.
- Консоль – консольные соединение.
- Медь Прямое – соединение медным кабелем типа витая пара, оба конца кабеля обжаты в одинаковой раскладке. Подойдет для следующих соединений: коммутатор – коммутатор, коммутатор – маршрутизатор, коммутатор – компьютер и др.
- Медь кроссовер – соединение медным кабелем типа витая пара, концы кабеля обжаты как кроссовер. Подойдет для соединения двух компьютеров.
- Оптика – соединение при помощи оптического кабеля, необходимо для соединения устройств, имеющих оптические интерфейсы.
- Телефонный кабель – обыкновенный телефонный кабель, может понадобится для подключения телефонных аппаратов.
- Коаксиальный кабель – соединение устройств с помощью коаксиального кабеля.

Пример локальной вычислительной сети

Рассмотрим на примере создания локальной вычислительной сети в cisco packet tracer, сеть представлена на рис. 7.9. Далее описывается пошаговая инструкция.

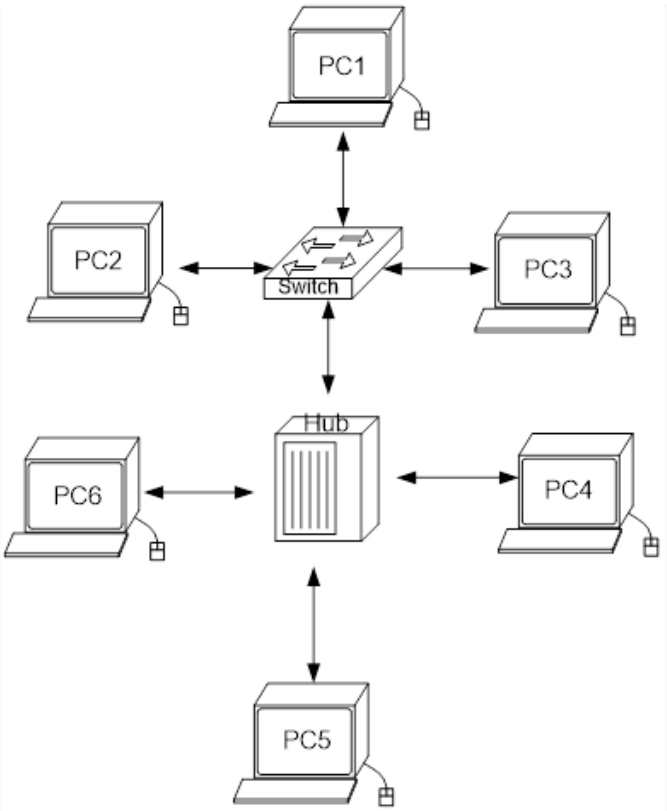


Рис. 7.9. Пример сети в cisco packet tracer.

Последовательность выполняемых действий:

1. В нижнем левом углу Packet Tracer выбираем устройства «Сетевые коммутаторы», и, в списке справа, выбираем коммутатор 2950-24, нажимая на него левой кнопкой мыши, вставляем его в рабочую область. Так же поступает с «Сетевым концентратором (Hub-PT)» и «Рабочими станциями (PC-PT)», в соответствии с рис. 7.11-7.13.

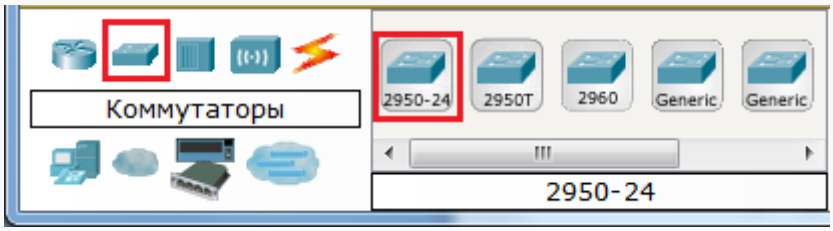


Рис. 7.10 – Выбирается коммутатор 2950-24

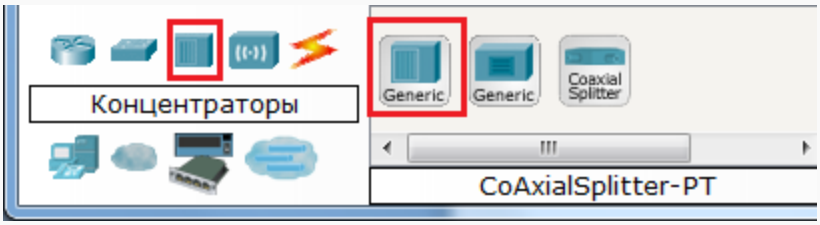


Рис. 7.11. Выбирается концентратор Hub-PT

Рис. 7.12. Выбирается персональный компьютер PC-PT

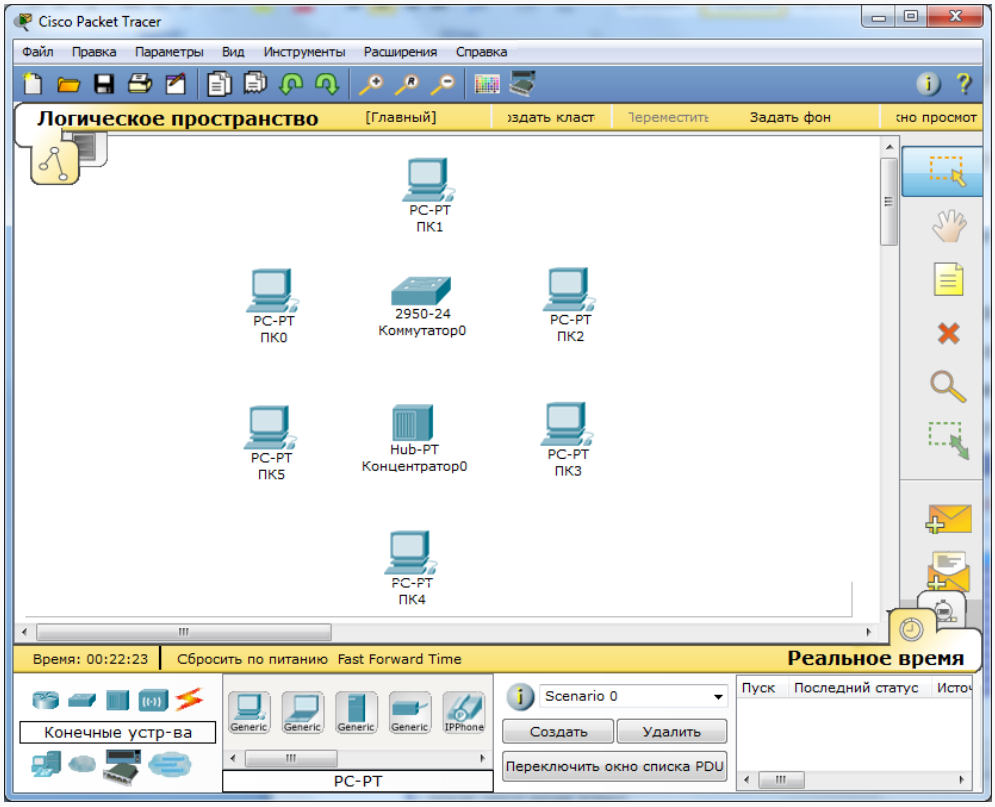


Рис. 7.13. Размещение компьютеров, коммутатора и концентратора на рабочей области

2. Далее необходимо соединить устройства, как показано на рис. 7.8, используя соответствующий интерфейс. Для соединения компьютеров к коммутатору и концентратору используется кабель типа «медный прямой» в соответствии с рис. 7.14.

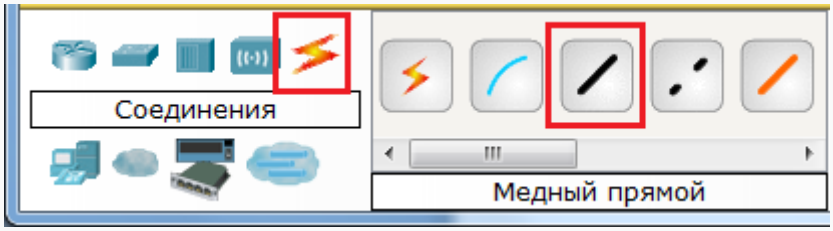


Рис. 7.14. Выбор типа кабеля «медный прямой»

А для соединения между собой коммутатора и концентратора используется медный кроссовер кабель в соответствии с рис. 7.15.

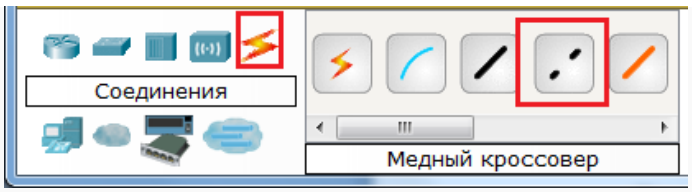


Рис. 7.15. Выбор типа кабеля «медный кроссовер»

Далее, для соединения двух устройств необходимо выбрать соответствующий вид кабеля и нажать на одно устройство (выбрав произвольный свободный порт FastEthernet) и на другое устройство (также выбрав произвольный свободный порт FastEthernet), в соответствии с рис. 7.16 - 7.18.

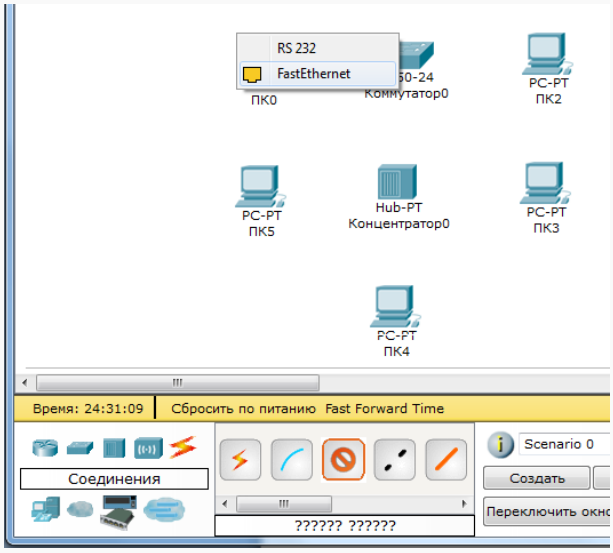


Рис. 7.16. Выбирается свободный порт на компьютере

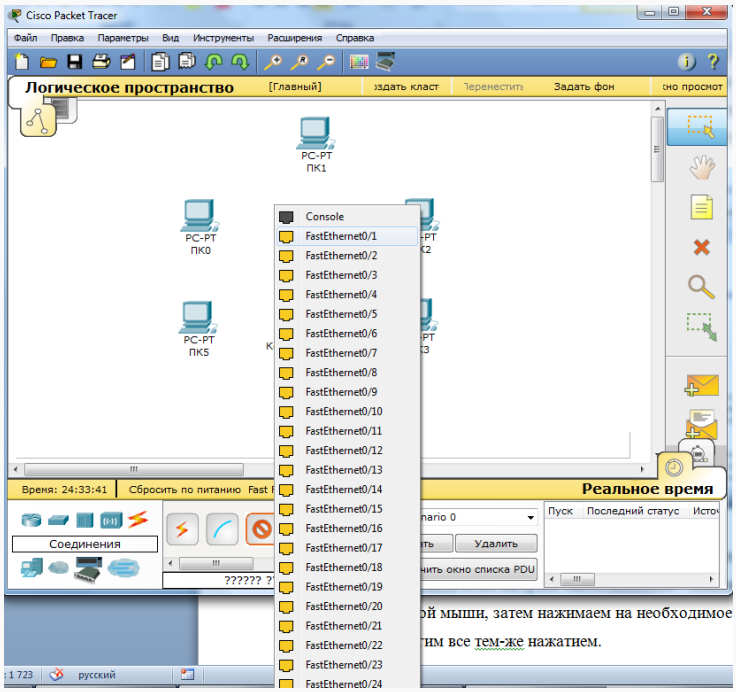


Рис. 7.17. Выбирается свободный порт на коммутаторе

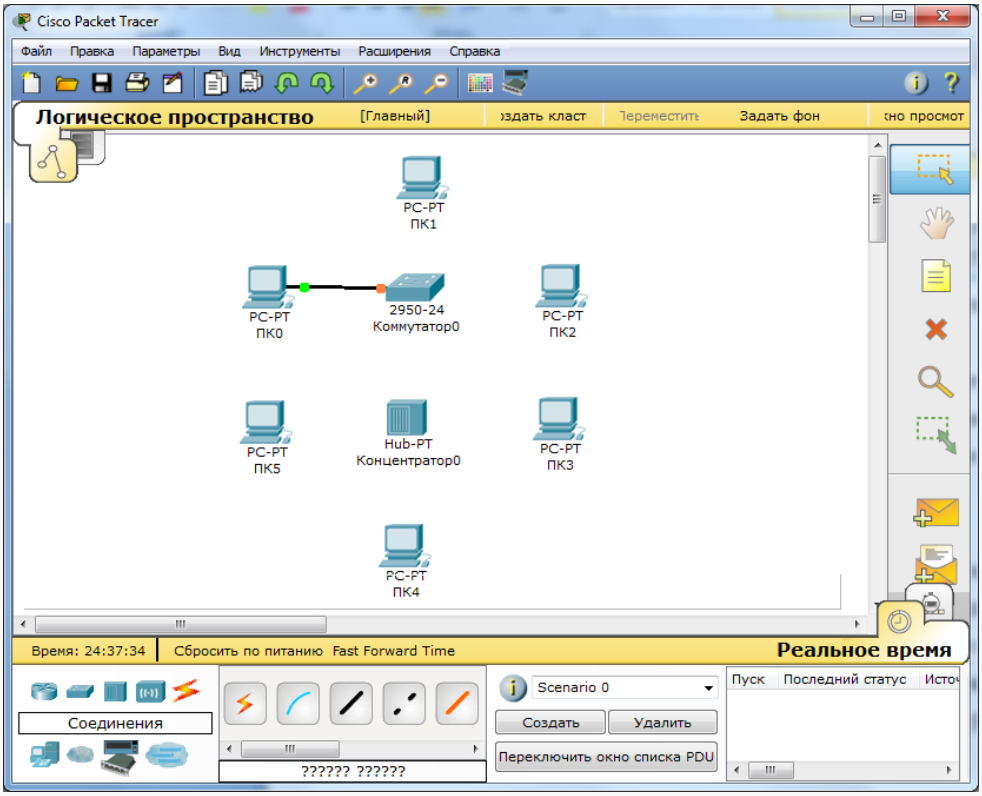


Рис. 7.18. Соединение медным прямым кабелем ПК 0 и коммутатор 0

Аналогично выполняется соединение для всех остальных устройств.

Результат подключения устройств представлен на рис. 7.19.

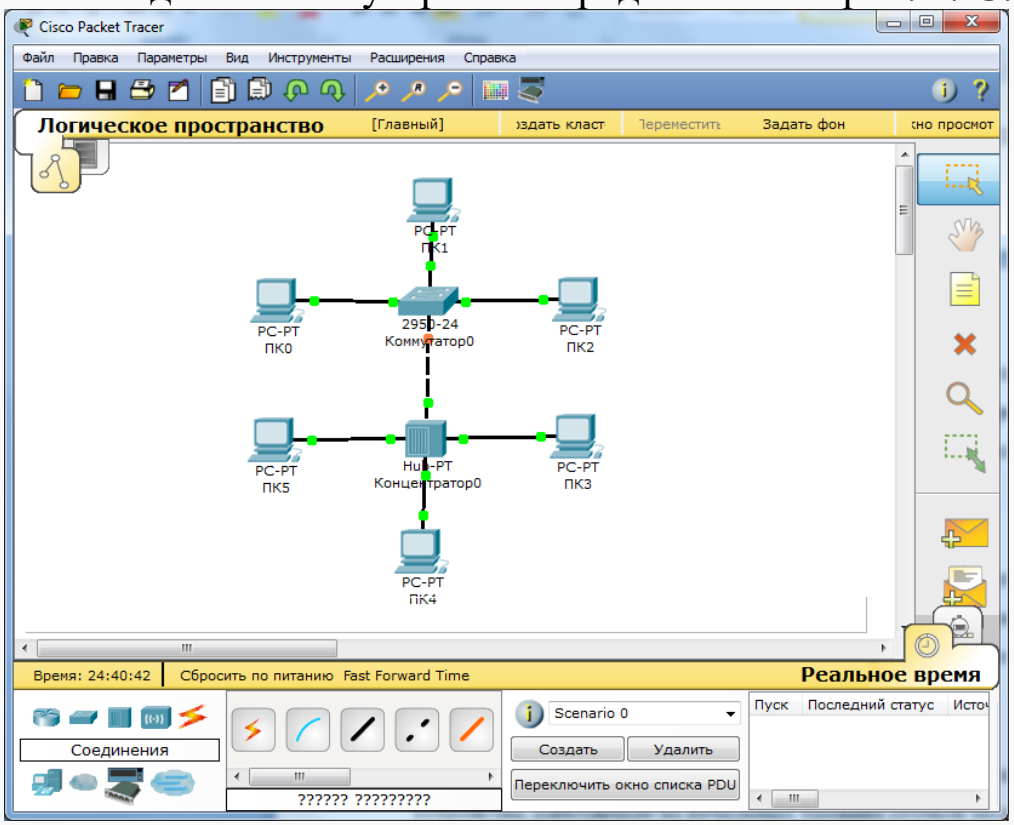


Рис. 7.19. Подключение устройств между собой

3. Этап настройки. Так как используются устройства, работающие на начальных уровнях сетевой модели OSI (коммутатор на 2-м, концентратор – на 1-м), то их настраивать не надо. Необходима лишь настройка рабочих станций, а именно: IP-адреса, маски подсети.

Ниже приведена настройка лишь одной станции (PC1) – остальные настраиваются аналогично.

Производим двойной щелчок по нужной рабочей станции, в соответствии с рис. 7.20.

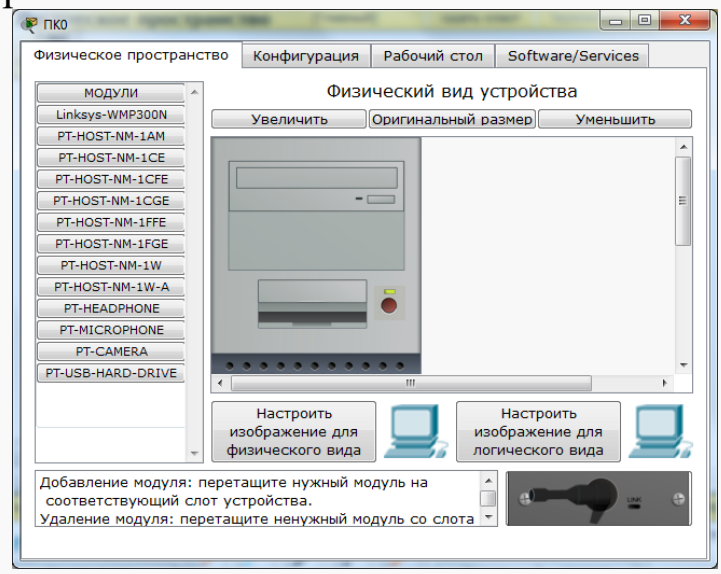


Рис. 7.20. Окно настройки компьютера PC0

В открывшемся окне выбирается вкладку Рабочий стол, далее – «Настройка IP», в соответствии с рис. 7.21.

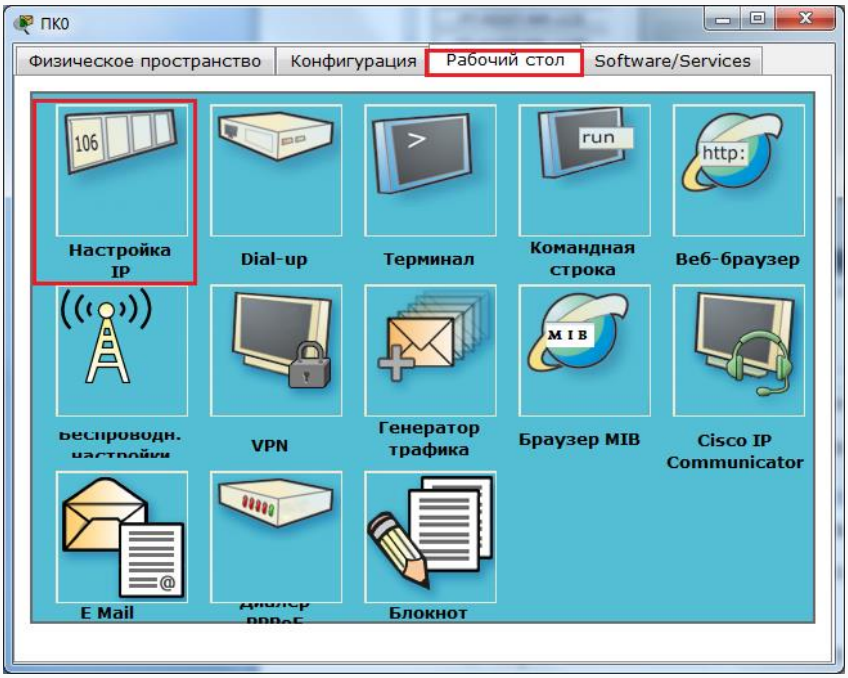


Рис. 7.21. Окно настройки компьютера PC0, вкладка «Рабочий стол»

Открывается окно, в соответствии с рис. 7.22, где нужно ввести IP-адрес и маску.

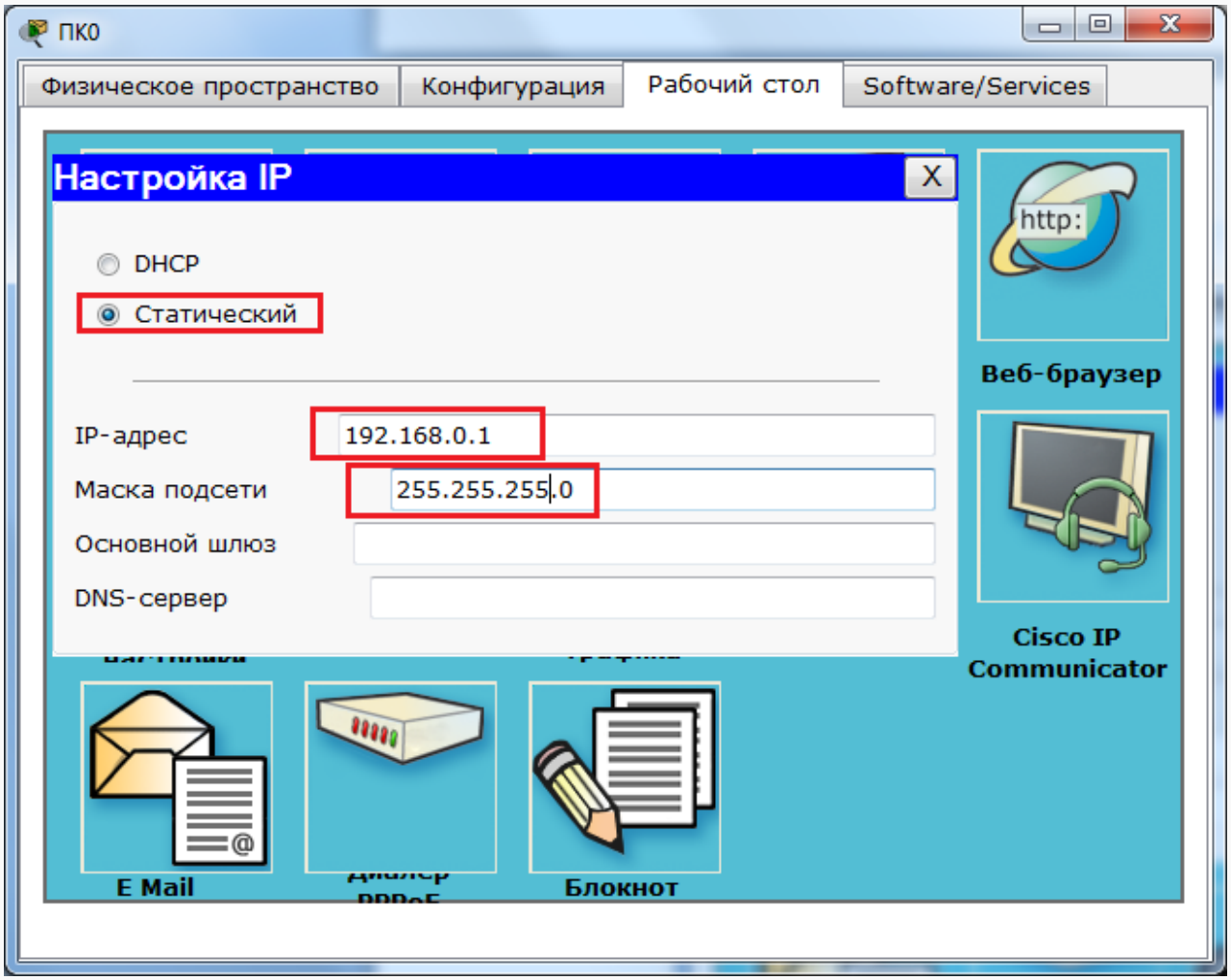


Рис. 7.22. Ввод статического IP-адреса и маски

Аналогично присваиваются IP-адреса всем остальным компьютерам.

Важно! IP-адреса всех рабочих станций должны находиться в одной и тойже подсети (то есть из одного диапазона), иначе процесс ping не выполнится.

Шлюз. Поле можно не заполнять.

DNS-сервер. Поле можно не заполнять.

4. Когда настройка завершена, выполняется ping-процесс. Например, запускается с PC5 и проверять наличие связи с PC1.

Важно! Можно произвольно выбирать, откуда запускать ping-процесс, главное, чтобы выполнялось условие: пакеты должны обязательно пересылаться через коммутатор и концентратор.

Для этого производим двойной щелчок по нужной рабочей станции, в открывшемся окне выбираем вкладку «Рабочий стол», далее – «Командная строка», в соответствии с рис. 7.23.

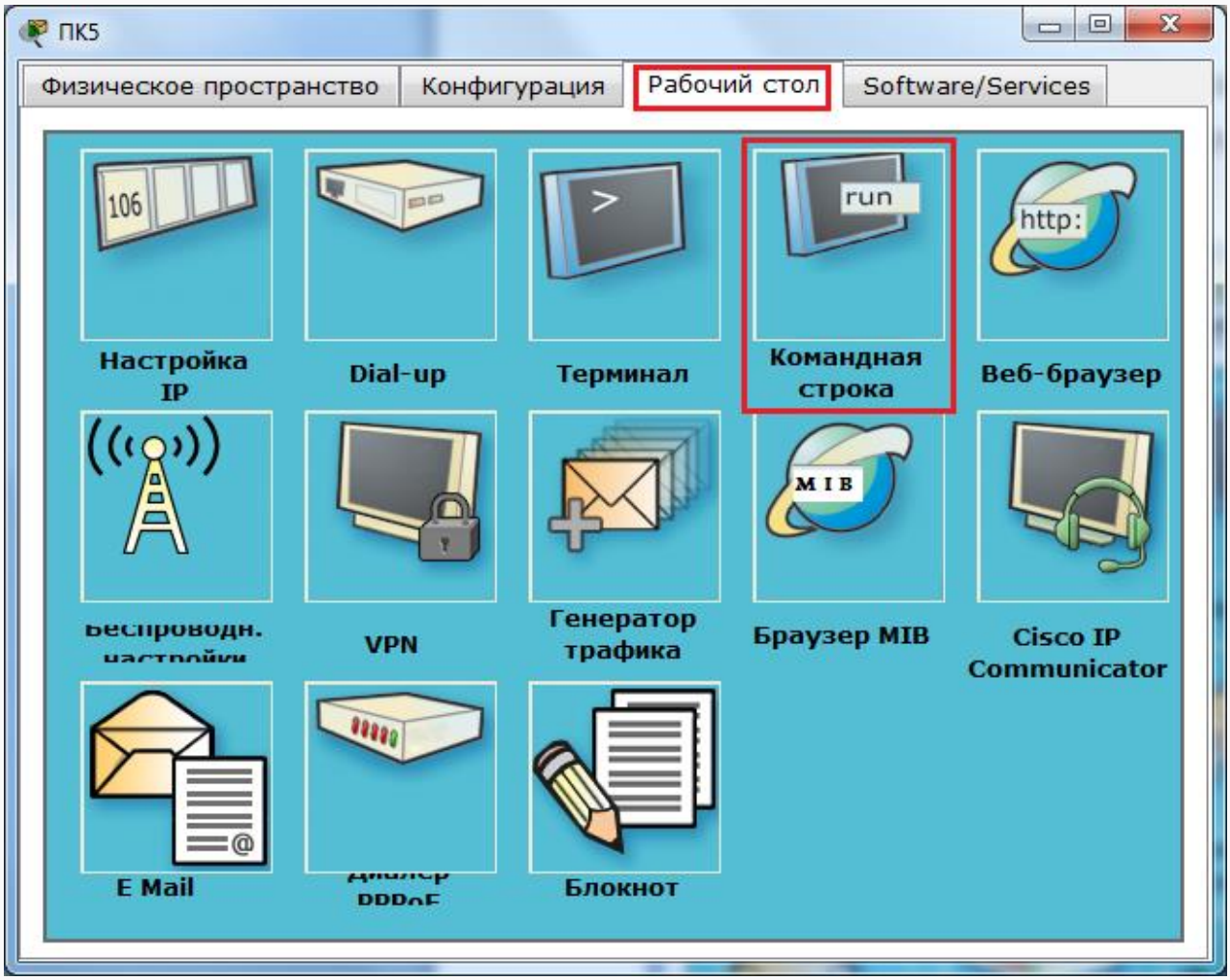


Рис. 7.23. Выбор режима «Командная строка»

Откроется окно командной строки в соответствии с рис. 7.24.

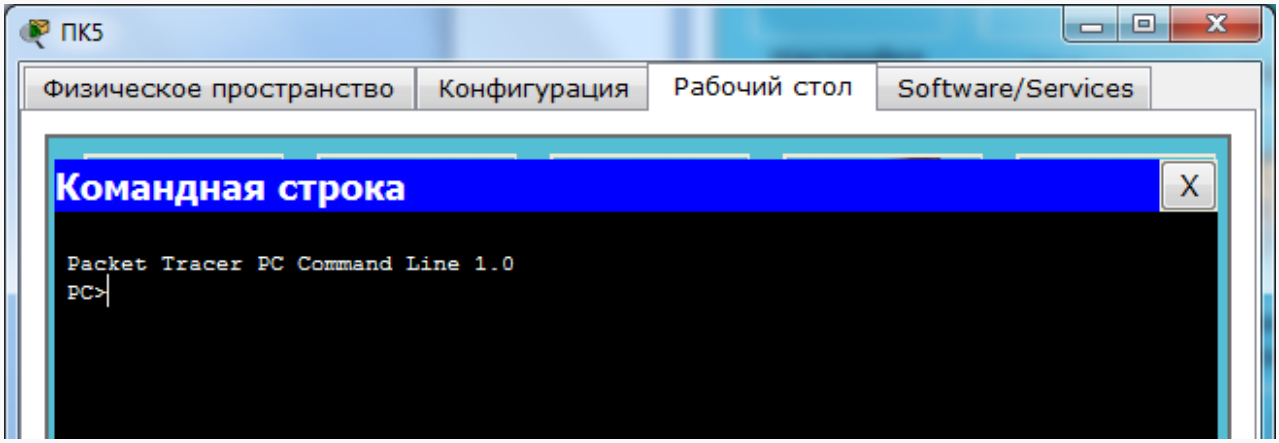


Рис. 7.24. Режим «Командная строка»

Предлагается ввести команду:

PC> ping 192.168.0.1

Если все настроено верно, то появляется информация, представленная на рис. 7.25.

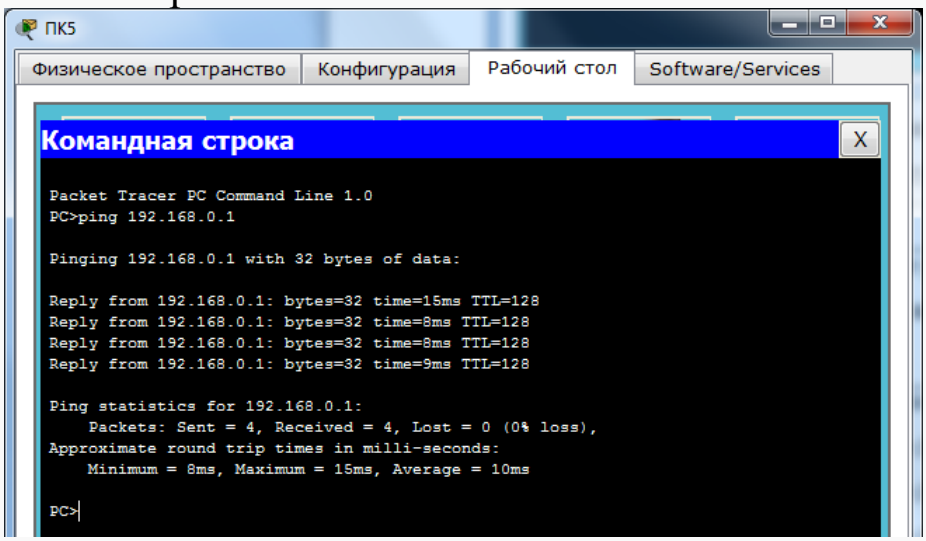


Рис. 7.25. Результат выполнение команды «ping»

Это означает, что связь установлена, и данный участок сети работает исправно.

Также Packet Tracer позволяет выполнять команду «ping» значительно быстрее и удобнее. Для этого, выбирается на боковой панели сообщение в соответствии с рис. 7.26.

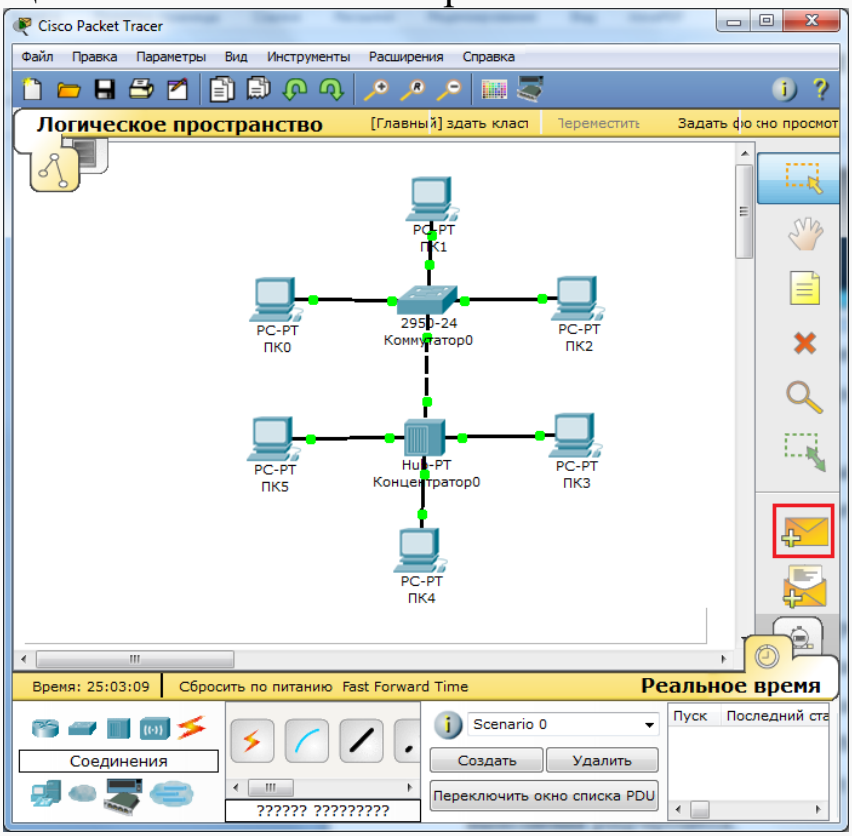


Рис. 7.26. Выбирается сообщение для выполнение команды «ping»

Далее нужно кликнуть мышкой по компьютеру, от кого будет передавать команда «ping» и еще раз щелкнуть по компьютеру, до которого будет выполнять команда «ping». В результате будет выполнена команда «ping», результат отобразиться в нижнем правом угле в соответствии с рис. 7.27.

Для более детального отображения результата выполнения команды необходимо выбрать «Переключить окно списка PDU» в соответствии с рис. 7.28.

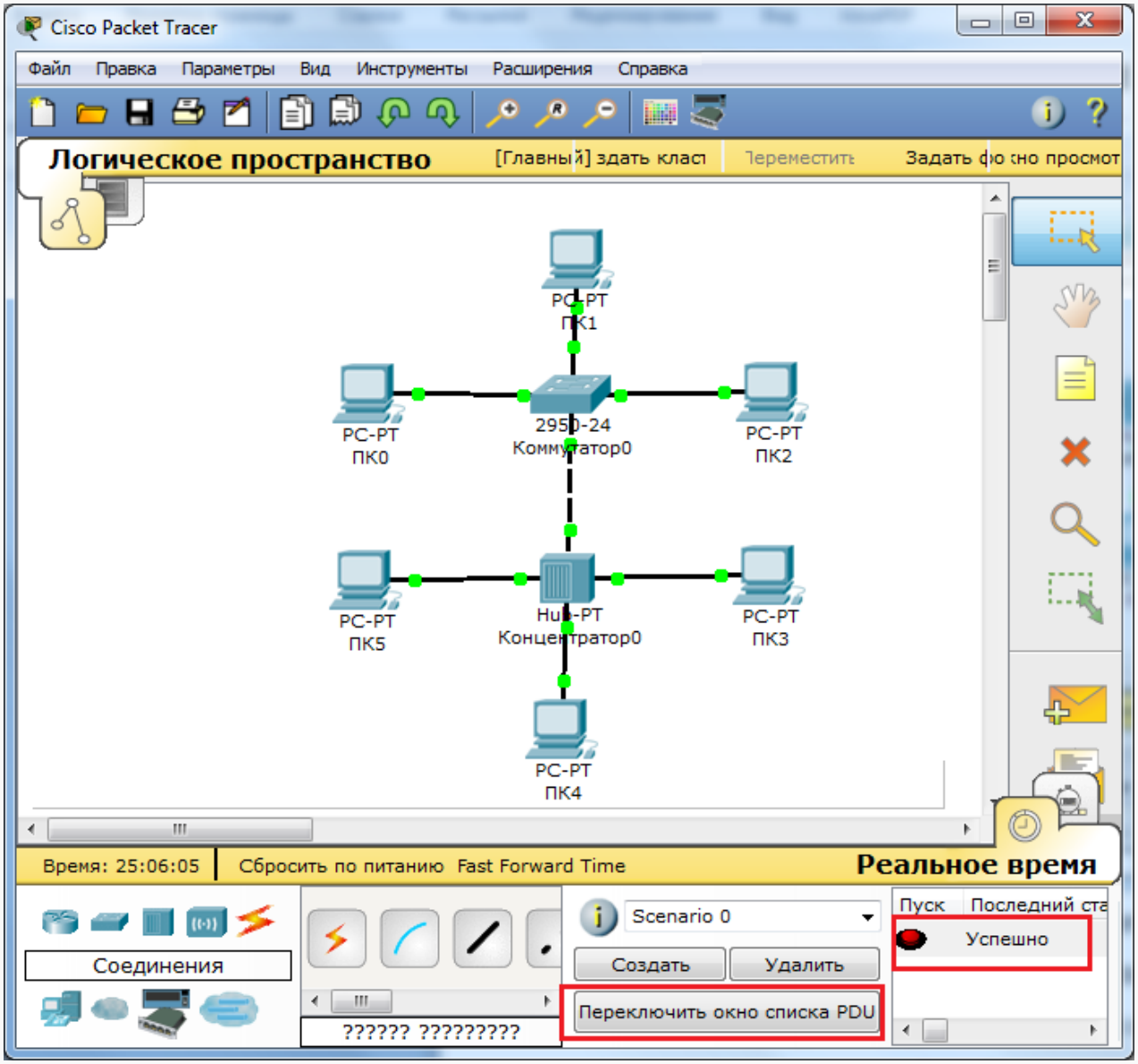


Рис. 7.27. Результат выполнения команды «ping»

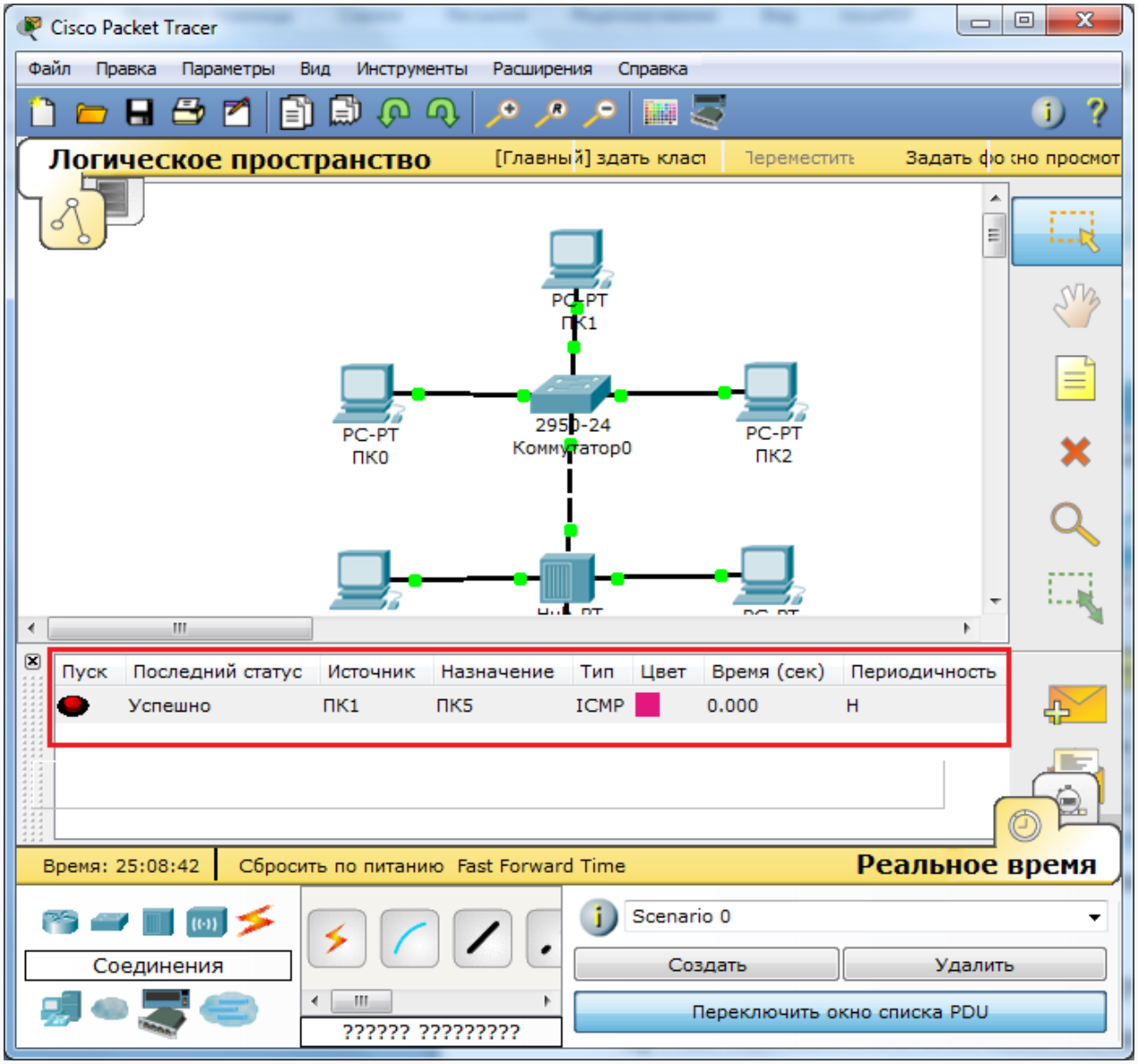


Рис. 7.28. Результат выполнения команды «ping»

5. В Packet Tracer предусмотрен режим моделирования, в котором подробно описывается и показывается, как работает утилита Ping. Поэтому необходимо перейти в «режим симуляции», нажав на одноименный значок в нижнем левом углу рабочей области, или по комбинации клавиш Shift+S. Откроется «Панель моделирования», в которой будут отображаться все события, связанные с выполненем ping-процесса в соответствии с рис. 7.29.

Перед выполнением симуляции необходимо задать фильтрацию пакетов. Для этого нужно нажать на кнопку «Изменить фильтры», откроется окно, в соответствии с рис. 7.30, в котором нужно оставить только «ICMP» и «ARP».

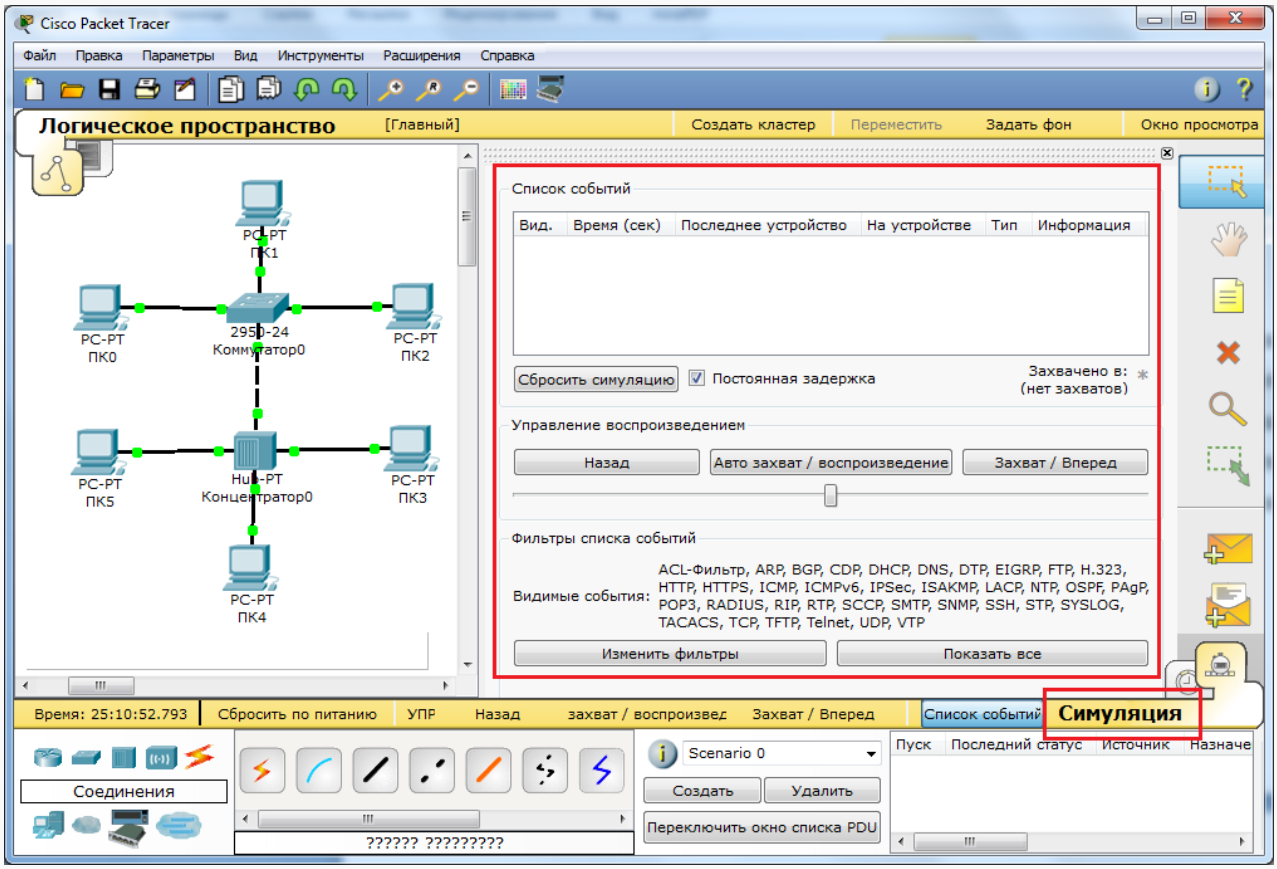


Рис. 7.29. Переход в «режим симуляции»

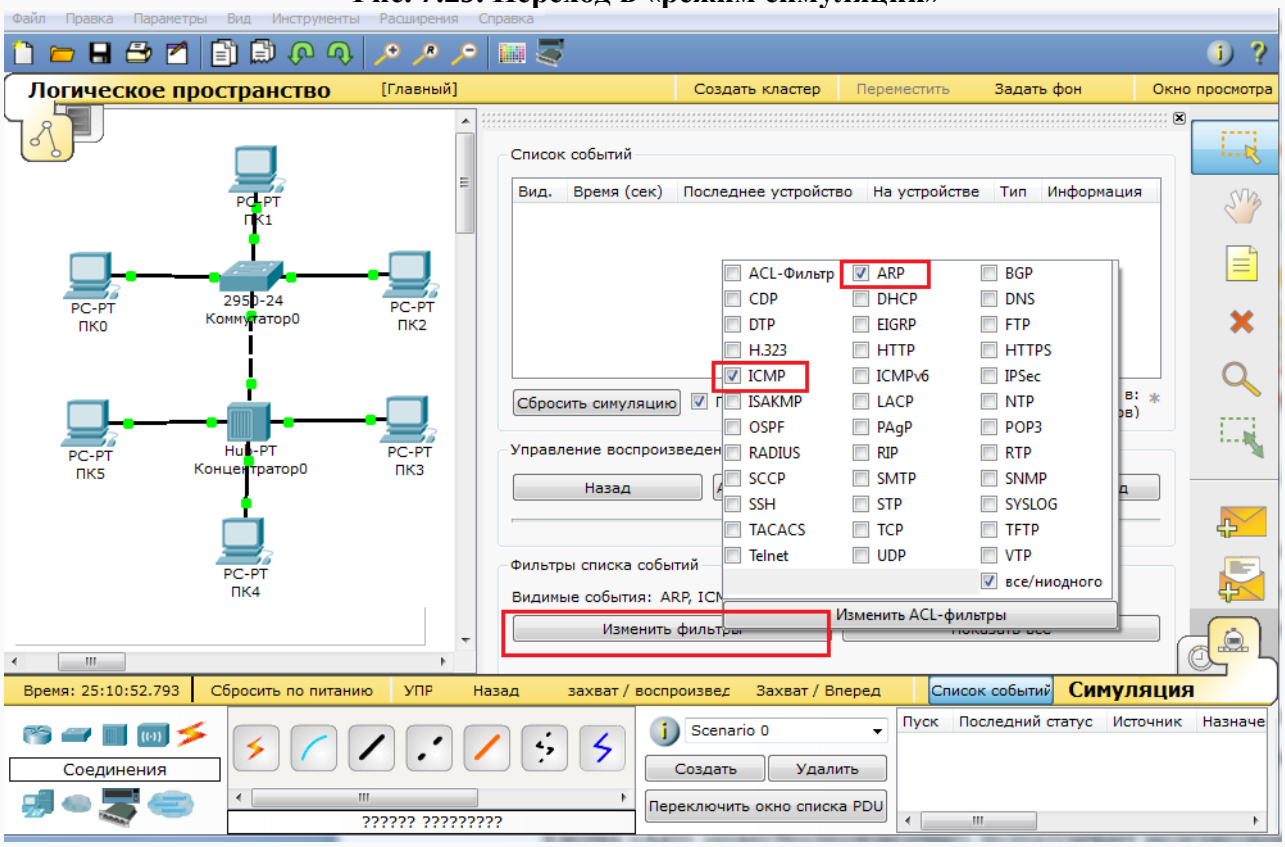


Рис. 7.30. Настройка фильтра

Теперь необходимо повторить запуск ping-процесса. После его запуска можно сдвинуть «Панель моделирования», чтобы на схеме спроектированной сети наблюдать за отправкой/приемкой пакетов.

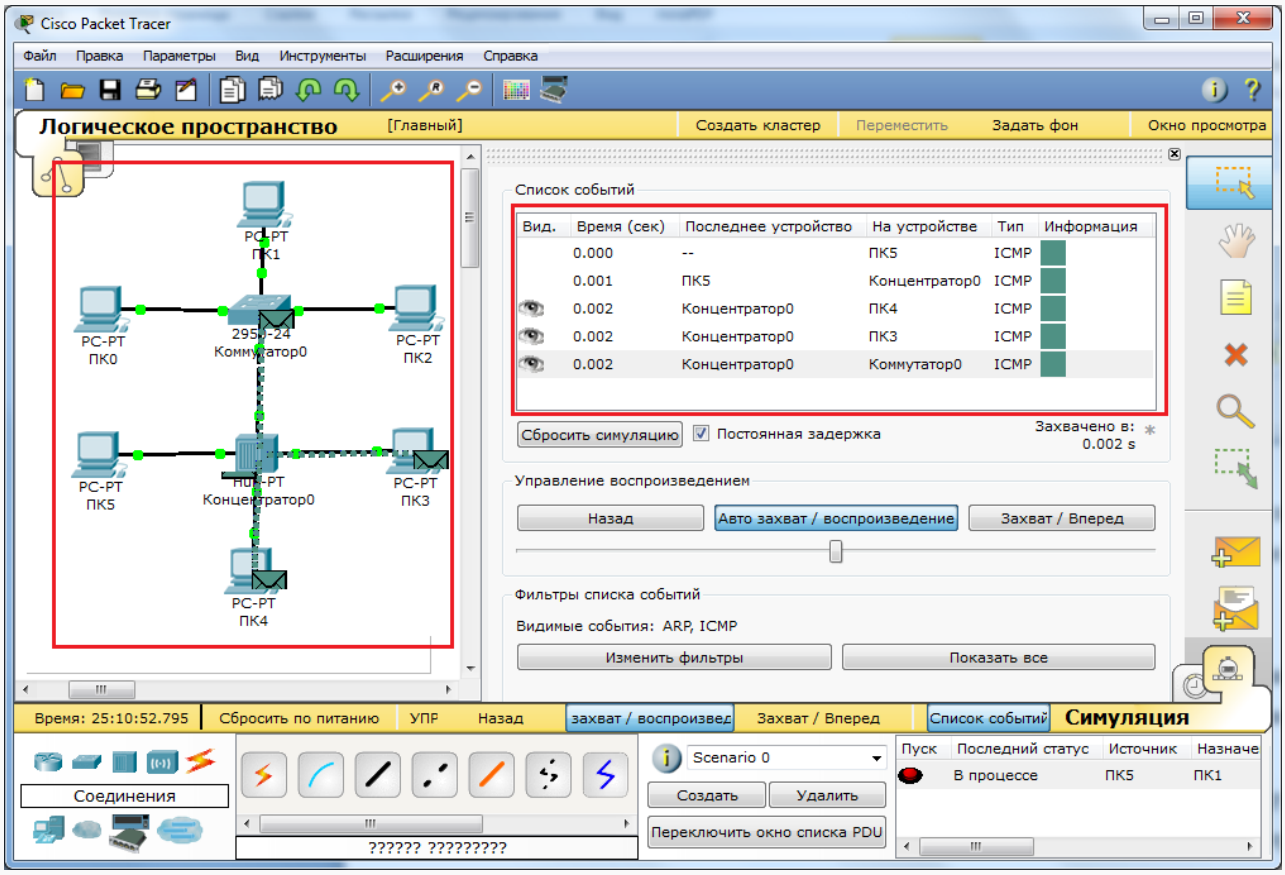


Рис. 7.31. Выполнение процесса симуляции

Кнопка «Авто захват/Воспроизведение» подразумевает моделирование всего ping-процесса, тогда как «Захват/Вперед» позволяет отображать его пошагово. Чтобы узнать информацию, которую несет в себе пакет, его структуру, достаточно нажать правой кнопкой мыши на цветной квадрат в графе «Информация». Моделирование прекращается либо при завершении ping-процесса, либо при закрытии окна «Редактирования» соответствующей рабочей станции.

Для удаления задания нажимается кнопка «Удалить» в нижней части экрана.

Контрольные вопросы

1. Зачем используются среды имитационного моделирования компьютерных сетей?
2. Чем отличается режим рабочей области «Логический» от «Физический»?

3. Какие элементы имеются в основном окне среды CISCO Packet Tracer?
5. Чем отличается маршрутизатор от коммутатора и концентратора?
6. Каким образом можно производить конфигурирования сетевых устройств?

Лабораторная работа № 8

НАСТРОЙКА КОММУТАТОРА

Цель работы: познакомится с основными принципами работы по настройке конфигурации коммутатора Cisco.

Пояснения к работе

Компания Cisco — одна из ведущих транснациональных корпораций на рынке телекоммуникационного оборудования. Продукция компании Cisco получила признание во всем мире из-за своей надежности и неприхотливости.

Настройка оборудования Cisco весьма специфична и несколько отличается от оборудования других производителей. Например, для выполнения первичных настроек коммутаторов компании Cisco потребуется фирменный плоский кабель RJ-45 – RS-232 голубого цвета (идет в комплекте с оборудованием) и наличие COM-порта на компьютере, с которого будет производиться настройка.

Большинство материнских плат современных настольных и портативных компьютеров не имеют соответствующего разъема. В таком случае необходим переходник. В последнее время Cisco стала комплектовать оборудование разъемом Mini—USB для консоли. Для настройки устройства через порт Mini—USB необходимо скачать cisco usb console driver.

В связи с отсутствием HyperTerminal в Windows 7/8 необходимо выполнить копирование папки HyperTerminal с Windows XP (месторасположение каталога – Program Files) в любой удобный каталог Windows 7/8. Для запуска программы используется файл hypertrm.exe, который можно найти в той же папке. Либо использовать программу

Putty, которую помимо подключения к Cisco-оборудованию можно использовать для подключения к серверам, маршрутизаторам, пр. с помощью SSH-подключения.

Переходим к подключению. На передней панели коммутатора ищем разъем RJ-45 с подписью «Console» и подключаем кабель. Включаем питание коммутатора. Заходим на компьютере в HyperTerminal, выбираем интерфейс разъема (COM1), скорость порта – 9600 Б/с, на все дальнейшие вопросы даем отрицательный ответ («No»).

В целях безопасности на коммутаторах Cisco доступно два режима ввода команд: пользовательский режим для проверки состояния коммутатора и привилегированный режим (аналог пользователя root в UNIX или администратора в Windows) для изменения конфигурации коммутатора. Для пользователей, привыкших работать в UNIX-системах, понять, в каком режиме они работают, не составит труда.

Для пользователей, работающих в Windows, дадим пояснение: если строка перед командной начинается с символа «#», вы в привилегированном режиме. То же самое касается ввода пароля, как и в UNIX-системах, пароль, который вводит пользователь не отображается на экране. Для перехода в привилегированный режим служит команда «enable», без кавычек, а для выхода «disable».

Приступим к первичной настройке коммутатора. При первой загрузке устройства мастер установки предложит выполнить пошаговую настройку, отказываемся от этого шага:

```
Continue with configuration dialog? [yes/no]: no
```

После чего оказываемся в пользовательском режиме:

```
Switch>
```

Переходим в привилегированный режим, пароль по умолчанию, как правило, отсутствует, поэтому ничего не вводим, а нажимаем «Enter».

```
Switch> enable
Password:
Switch#
```

Для задания настроек, касающихся всего коммутатора (задание имени коммутатора, IP-адреса, указание сервера синхронизации времени, пр), используется режим глобальной конфигурации, для настройки отдельных интерфейсов существует режим конфигурации интерфейса.

1. Изменим имя нашего коммутатора (по умолчанию имя Switch):

```
Switch# configure terminal
Switch(config)# hostname Switch01 (Задаем имя коммутатора – Switch01)
Switch01(config)#
```

При наличии множества коммутаторов рекомендуем в обязательном порядке присваивать уникальные имена для каждого из них. В последующем это помогает быть уверенным, что конфигурация выполняется именно на нужном устройстве.

Также обращаем ваше внимание на то, что вместо длинных команд как, например, «configure terminal» существуют их короткие аналоги «conf t».

2. Зададим IP-адрес для интерфейса управления коммутатором.

```
Switch01(config)# interface fa0/0 (указываем интерфейс для настройки)
Switch01(config-if)# no shutdown (включаем интерфейс)
Switch01(config-if)# ip address 192.168.0.1 255.255.255.0 (задаем IP-адрес и маску) Switch01(config-if)# exit (выходим из режима конфигурации интерфейса)
Switch01(config)#
```

3. Установим пароль для привилегированного режима:

```
Switch01(config)# enable secret pass1234 (пароль pass1234)
Switch01(config)# exit
Switch01#
```

Важно! Установка пароля может быть выполнена двумя командами password и secret. В первом случае пароль хранится в конфигурационном файле в открытом виде, а во втором - в зашифрованном. Если использовалась команда password, необходимо зашифровать пароли, хранящиеся в устройстве в открытом виде с помощью команды «service password-encryption» в режиме глобальной конфигурации.

4. **Поскольку данные при telnet соединении передаются в открытом виде, для удаленного подключения к коммутатору будем использовать SSH-соединение, позволяющее шифровать весь трафик.**

Switch01# clock set 12:00:00 15 April 2015 (Устанавливаем точное текущее время дату) Switch01# conf t
Switch01(config)# ip domain name geek—nose.com (Указываем домен, если домена нет, то пишем любой)
Switch01(config)# crypto key generate rsa (Выполняем генерацию RSA-ключа для ssh) Switch01(config)# ip ssh version 2 (Указываем версию ssh-протокола)
Switch01(config)# ip ssh autentification-retries 3 (Задаем кол—во попыток подключения по ssh)
Switch01(config)# service password-encryption (Сохраняем пароли в зашифрованном виде)
Switch01(config)# line vty 0 2 (Переходим в режим конф-и терминальных линий) Switch01(config-line)# transport input ssh (Разрешаем подключение только по ssh) Switch01(config-line)# exec timeout 20 0 (Активируем автоматическое разъединение ssh-сессии через 20 минут)
Switch01(config—line)# end (Выходим из режима конфигурирования)
Switch01# copy running-config startup-config (Сохраняем настройки)

Важно! Для выхода из подменю конфигурирования на 1 уровень выше, например, из «config—line» в «config» используется команда «exit». Для полного выхода из режима конфигурирования используйте команду «end».

Выше была описана базовая настройка ssh, с более продвинутой настройкой можно ознакомиться ниже:

Switch01# conf t
Switch01(config)# aaa new-model (Включаем AAA—протокол)
Switch01(config)# username root privilege 15 secret pass1234 (Создаем пользователя root, с максимальным уровнем привилегий – 15, пароль pass1234)
Switch01(config)# access—list 01 permit 192.168.0 0.0.0.255 (Создаем правило доступа с названием 01, регламентирующие право заходить по ssh

всем хостам сети 192.168.0.0/24; вместо адреса сети можно указать конкретный IP-адрес. Внимательно подумайте, есть ли необходимость в такой настройке для ваших задач.)
Switch01(config)# line vty 0 2 (Переходим в режим конф-и терминальных линий) Switch01(config—line)# privilege level 15 (Разрешаем вход сразу в привилегированный режим)
Switch01(config—line)# access—class 23 in (Привязываем созданное правило доступа по ssh к терминальной линии)
Switch01(config—line)# logging synchronous (Очень неудобно, когда лог-сообщения коммутатора прерывают ввод команд. Отключая журнальные сообщения данной командой, коммутатор ждет завершение вводимой команды, а также вывода отчета о ее исполнении, после чего в случае необходимости выводит лог)
Switch01(config—line)# end (Выходим из режима конфигурирования)
Switch01# copy running-config startup-config (Сохраняем настройки)
На этом базовая настройка коммутатора Cisco 2960 завершена.

Контрольные вопросы

1. Назначение коммутатора Cisco Catalyst 2960?
2. Для чего необходим CONSOLE порт?
3. С помощью какого порта можно войти в Web-управление?
4. Какие виды управления вы знаете? Их основные отличия
5. Какие начальные конфигурации имеет коммутатор?

Лабораторная работа № 9

НАСТРОЙКА МАРШРУТИЗАТОРА

Цель работы: познакомиться с основными принципами работы по настройке конфигурации маршрутизатора Cisco.

Пояснения к работе

Назначение имени устройству осуществляется командой **hostname**

```
router (config)# hostname "ИМЯ"
```

HTTP сервер как правило отключается, т.к. подвержен DoS. Для этого используется команда:

```
router# no ip http server.
```

Несмотря на то, что логи обычно записываются на сервер, обычно выделяют буфер в оперативной памяти размером 256 Кб в качестве резервного варианта.

```
router# logging buffered 262144 debugging
```

Для того чтобы не вызывать перегрузку, необходимо поставить ограничение: не более 10 сообщений в секунду.

```
router# logging rate-limit 10 except warnings
```

Из соображений безопасности отключаем опцию IP-пакета **source routing**

```
router# no ip source-route
```

Если по маршруту есть участники с размером MTU меньше чем 1500 байт, то необходимо включить path mtu discovery

```
router# ip tcp path-mtu-discovery
```

Для того чтобы заходить на устройство по команде **telnet,** необходимо поставить защиту. В противном случае доступ по telnet не будет получен.

```
S1(config)# enable secret "пароль"
S1(config)# no ip domain-lookup
S1(config)# line console 0
S1(config-line)# password "пароль"
S1(config-line)# line vty 0 15
S1(config-line)# password "пароль"
S1(config-line)# login
```

Для установки пароля на привилегированный режим необходимо сделать следующее

```
R1(config)# enable secret "пароль"
R1(config-line)# line console 0
R1(config-line)# password "пароль"
R1(config-line)# login
```

Внутреннее время задается с помощью команды **clock set**. Команда вводится в привилегированном режиме.

```
Router# clock set 15:00:00 10 aug 2015
```

Часовой пояс настраиваем с помощью команды **clock timezone**

```
Router(config)# clock timezone UTC 3
```

Настройка SSH подключения

Задаем имя роутеру:
Router1(config)# hostname R1
Назначаем доменное имяЖ
R1(config)# ip domain-name name.com
Генерируем RSA ключ:
R1(config)# crypto key generate rsa
Создаем пользователя root и назначает ему пароль admin
R1(config)# username root secret admin
создаем access-list, в котором можем задать, например с каких IP адресов мы можем подключиться:
R1(config)# access-list 1 permit 10.1.1.12
Настраиваем виртуальный терминал с 0 по 4 и задаем проверку пароля локально
R1(config)# line vty 0 4
R1(config)# login local
С помощью списка доступа разрешаем доступ с IP-адреса
R1(config)# access-class 1 in
И разрешаем работать по протоколы SSH
R1(config)# transport input ssh
R1(config)# transport output ssh
R1(config)# end

Настройка NAT

Настраиваем интерфейс, который смотрит в интернет
interface FastEthernet0/0
ip nat outside
На локальном интерфейсе
interface vlan1
ip nat inside
Создаем список IP-адресов, имеющих доступ к NAT
ip access-list entended NAT
permit ip host 192.168.xxx.xxx any
Включаем NAT (PAT) на внешнем интерфейсе
ip nat inside source list NAT interface FastEthernet0/0 overload
И добавляем инспекцию протоколов
ip inspect name INSPECT_OUT http
ip inspect name INSPECT_OUT https
ip inspect name INSPECT_OUT ftp

Раздаем адреса по DHCP

```
R1(config)# service dhcp
R1(config)# ip dhcp pool
R1(config-pool)# network 192.168.x.x 255.255.255.0
R1(config-pool)# default-router 192.168.20.1
R1(config-pool)# domain-name "имя"
R1(config-pool)# dns-server 192.168.20.101
R1(config-pool)# lease 7
R1(config)# ip dhcp excluded-address 192.168.20.1
```

Сохранение настроек роутера
Для того чтобы не повредить конфигурационные файлы после выключения устройства, можно сделать сохранение настроек командой **copy running-config startup-config**

Просмотр интерфейса show interfaces

Позволяет просмотреть статус всех интерфейсов на устройстве. Если указать конкретный порт, то отобразится информация только по указанному интерфейсу. Например:

```
Router# show interfaces FastEthernet 0/0
FastEthernet0/0 is up, line protocol is up (connected)
Hardware is Lance, address is 0030.a399.3901 (bia 0030.a3
99.3901)
MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
reliability 255/255, txload 1/255, rxload 1/255
Encapsulation ARPA, loopback not set
ARP type: ARPA, ARP Timeout 04:00:00,
Last input 00:00:08, output 00:00:05, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0 (size/max/drops); Total output drops:
0
Queueing strategy: fifo
Output queue :0/40 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
0 packets input, 0 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 a
bort
0 input packets with dribble condition detected
0 packets output, 0 bytes, 0 underruns
0 output errors, 0 collisions, 1 interface resets
0 babbles, 0 late collision, 0 deferred
0 lost carrier, 0 no carrier
0 output buffer failures, 0 output buffers swapped out
```

Основная полученная информация

- FastEthernet0/0 is up, line protocol is up (connected) — порт включен и получает сигнал от подключенного к нему кабеля.
- Hardware is Lance, address is 0030.a399.3901 (bia 0030.a399.3901) — информация о модели интерфейса и его физическом адресе.
- Encapsulation ARPA, loopback not set — указывает тип инкапсулирования для данного интерфейса.

> *Если инкапсулирование данных для интерфейсов локальной сети не нуждается в настройке, то для интерфейсов глобальных сетей это необходимо.*

Интерфейс может быть в различном состоянии:
— «UP» — включен, поднят;
— «DOWN» — выключен, нет линка;
— «Administratively down» — административная блокировка.

Различие заключается в том, что отключенный интерфейс находится в рабочем состоянии, но не обменивается данными с подключенной к нему средой, а в состоянии административной блокировки интерфейс отключен на уровне конфигурации.

Пример административной блокировки порта

```
Router# show interfaces FastEthernet 0/0
FastEthernet0/1 is administratively down, line protocol i
s down (disabled)
  Hardware is Lance, address is 0030.a399.3902 (bia 0030.
a399.3902)
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
  reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
...
```

Настройка интерфейса Ethernet

Заходим на интерфейс
router (config)# interface fastethernet 0/1
Устанавливаем IP на интерфейс
router (config)# ip add 10.0.0.1 255.255.255.0
Поднимаем интерфейс
router (config)# no shutdown
Посмотреть какой IP-адрес назначен интерфейсу можно командой **show ip interface**

```
Router#show ip interface fastEthernet 0/0
FastEthernet0/0 is up, line protocol is down (disabled)
  Internet address is 10.0.0.1/24
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
...
```

Настройка интерфейса Serial

Заходим на интерфейс
router (config)# interface serial 0/0/0
Устанавливаем IP на интерфейс
router (config)# ip add 192.168.1.1 255.255.255.0
Устанавливаем скорость порта
router (config)# clock rate 56000
Поднимаем интерфейс
router (config)# no shutdown
Команда encapsulation определяет тип протокола канального уровня и формат передаваемых данных для конкретного интерфейса. Например:

```
Router# configure
Router(config)# interface serial0/1/1
Router(config-if)# encapsulation ?
  frame-relay Frame Relay networks
  hdlc Serial HDLC synchronous
  ppp Point-to-Point protocol
Router (config-if)# encapsulation ppp
```

Отключение порта shutdown

Команды выключения/включения интерфейса применяется, если необходимо изменить административное состояние интерфейса. Оборудование Cisco не передает данные на интерфейс, если он заблокирован на административном уровне. Выключение порта осуществляется путем входа на нужный интерфейс и ввода команды shutdown. Например:

```
Router# configure
Router(config)# interface FastEthernet0/1
Router(config-if)# shutdown
```

Проверяем полученный результат с помощью уже известной команды **show interfaces**:

```
Router# show interfaces FastEthernet 0/0
FastEthernet0/1 is administratively down, line protocol i
s down (disabled)
```

```
  Hardware is Lance, address is 0030.a399.3902 (bia 0030.
a399.3902)
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
  reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
...
```

Включение интерфейса осуществляется командой **no shutdown**

```
Router# configure
Router (config)# interface FastEthernet0/1
Router (config-if)# no shutdown
```

Добавление описания/комментария с помощью команды description

К каждому из доступных интерфейсов доступно поле длиной не более 255 символов.

```
Router (config)# interface fastEthernet 0/0
Router (config-if)# description uplink_to_3750
```

Проверяем результат

```
Router# show interfaces fastEthernet 0/0
FastEthernet0/0 is up, line protocol is up (connected)
  Hardware is Lance, address is 0030.a399.3901 (bia 0030.
a399.3901)
  Description: uplink_to_3750
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
  reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
...
```

Просмотр информации о VLAN

Команда **show vlan** позволяет посмотреть информацию о созданных виртуальных каналах.

```
Router# show vlan
VLAN Name                              Status    Ports
---- --------------------------------- --------- ---------
-----------
1    default                           active
```

```
VLAN Type  SAID    MTU   Parent RingNo  BridgeNo Stp  Brd
gMode Trans1
---- ----  ------  ----  ------ ------  -------- ---  ---
----- ------
1    enet  100001  1500
```

> *В рабочих конфигурациях использование VLAN с номером 1 не рекомендуется.*

Команды конфигурирования IP-маршрутизации

Для просмотра таблицы IP-маршрутизации используется команда **show ip route**

```
Router# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M -
mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSP
F inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA ext
ernal type 2
       E1 - OSPF external type 1, E2 - OSPF external type
2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2,
ia - IS-IS inter area
       * - candidate default, U - per-user static route,
o - ODR
       P - periodic downloaded static route
Gateway of last resort is not set
  10.0.0.0/24 is subnetted, 1 subnets
  C 10.0.0.0 is directly connected, FastEthernet0/0
```

Результаты данной команды дают администратору сети огромное количество данных. Они являются ключевым инструментом для определения пути, по которому пакет проходит по сети. В приведенном выше примере строка Gateway of last resort — это сетевой адрес маршрутизатора, куда должны будут посылаться пакеты, имеющие пункт назначения вне данной сети. Если статические маршруты и протоколы динамической маршрутизации не заданы, то будет указано «is not set». После представлена сама таблица

маршрутизации. Буква перед маршрутом означает код, пояснения к которому выводятся выше.

Команды для конфигурирования служб имен доменов

Большинство пользователей обращаются к серверам, рабочим станциям и другим IP-устройствам, используя их имя, а не IP-адрес. Роль преобразования IP-адреса в имя отводится DNS серверам. Маршрутизаторы также могут пользоваться DNS-системой для преобразования имен в IP-адреса.

Пример конфигурирования службы DNS маршрутизатора:

```
Router# configure
Router (config)# ip domain-lookup
Router (config)# ip domain-name you_domen_name.ru
Router (config)# ip domain-list you_domen_name.ru
Router (config)# ip domain-list you_domen_name.ru
Router (config)# ip name-server 10.0.0.1 10.0.0.2
```

Проверка установок службы DNS на маршрутизаторе выполняется с помощью команды **show host**

```
router# show host
Default domain is not set
Name/address lookup uses static mappings

Codes: UN - unknown, EX - expired, OK - OK, ?? - revalida
te
    temp - temporary, perm - permanent
    NA - Not Applicable None - Not defined

Host            Port  Flags      Age Type   Address(es)
3750            None  (perm, OK) **   IP    10.10.1.1
c3750           None  (perm, OK) **   IP    10.10.1.2
5300-1          None  (perm, OK) **   IP    10.10.1.3
5300-2          None  (perm, OK) **   IP    10.10.1.4
c3750-1         None  (perm, OK) **   IP    10.10.1.5
c3750-2         None  (perm, OK) **   IP    10.10.1.6
```

Сброс настроек к первоначальным

На маршрутизаторах Cisco по команде

```
R1# erase /all nvram
```

на маршрутизаторах Cisco Catalyst:

```
switch# delete flash:vlan.dat
switch# erase startup-config
```

Контрольные вопросы

1. Как инициализируется маршрутизатор?
2. Какая аббревиатура используется для обозначения пользовательского интерфейса операционной системы IOS?
3. Какие два режима доступа к командам существует?
4. Какие существуют режимы конфигурации маршрутизатора?
5. Как выполняется настройка Serial интерфейса?
6. Как выполняется настройка Ethernet интерфейса?

Библиографический список

1. Берлин А. Н. Основные протоколы Интернет/Интернет-Университет Информационных Технологий, 2008. - 504 с. Режим доступа: http://www.knigafund.ru
2. Ермаков А.Е. Основы конфигурирования корпоративных сетей Cisco: учеб. пособие/М.: Изд-во УМЦ ЖДТ (Маршрут), 2013. - 248 с. Режим доступа: http://www.knigafund.ru
3. Пятибратов А. П., Гудыно Л. П., Кириченко А. А. Вычислительные системы, сети и телекоммуникации/М.: Финансы и статистика, 2013.- 736 с. Режим доступа: http://www.knigafund.ru
4. Лапонина О. Р. Протоколы безопасного сетевого взаимодействия/М.: Национальный Открытый Университет «ИНТУИТ», 2016.- 462 с. Режим доступа: http://www.knigafund.ru
5. Семенов Ю. А. Алгоритмы телекоммуникационных сетей/М.: Интернет-Университет Информационных Технологий, 2007. – 512 с. Режим доступа: http://www.knigafund.ru
6. Воробьев С. П. Локальные вычислительные сети : учеб. пособие. - Новочеркасск: Изд-во НВВКУС, 2006. - 165 с.

Учебно-методическое издание

Воробьёв Сергей Петрович

КОМПЬЮТЕРНЫЕ СЕТИ

Учебно-методическое пособие
к выполнению лабораторных работ

Редактор *Я.В. Максименко*

Подписано в печать 20.03.2017.
Формат 60×84 $^{1}/_{16}$. Бумага офсетная. Печать цифровая.
Усл. печ.л. 5,35. Уч.-изд.л.5,5. Тираж 100 экз. Заказ 46-0610.

Южно-Российский государственный политехнический университет
(НПИ) имени М.И. Платова
Редакционно-издательский отдел ЮРГПУ(НПИ)
346428, г. Новочеркасск, ул. Просвещения, 132

Отпечатано в ИД «Политехник»
346428, г. Новочеркасск, ул. Первомайская, 166
idp-npi@mail.ru