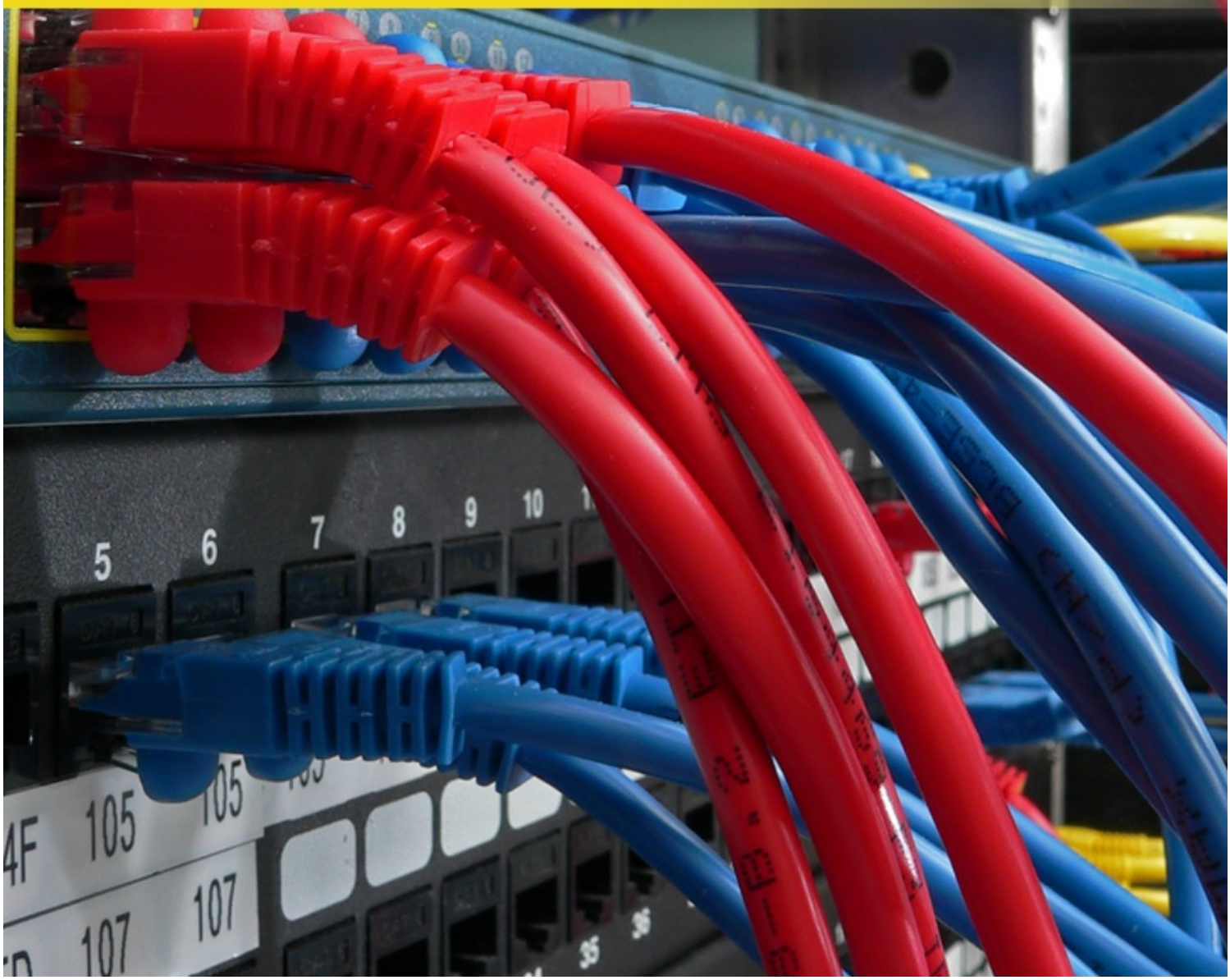
Схемы и диаграммы. Практические работы.
Проблемно-ориентированный подход

Д.Н. Буторин

КОМПЬЮТЕРНЫЕ СЕТИ
практический курс

Учебное пособие



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГБОУ ВПО «КРАСНОЯРСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.П. Астафьева»

Буторин Денис Николаевич

Кандидат педагогических наук

Компьютерные сети. Практический курс

Учебное пособие

2012

ББК 32.97
УДК 007.72

Рецензенты:

д.п.н., профессор, Пак Николай Инсебович, зав.кафедрой ИВТ
КГПУ им. В.П. Астафьева

Умяров Рауль Фаридович, КГПУ

Буторин Д.Н. Компьютерные сети. Практический курс: учебное пособие; Красноярский гос.пед.ун-т. им. В.П. Астафьева, 2012, 241с.

Настоящее учебное пособие разработано на средства программы стратегического развития Красноярского государственного педагогического университета им. В.П.Астафьева, проект № 03-1/12. Пособие подготовлено с учетом программы дисциплины «Компьютерные сети» для специальностей 230201 – «Информационные системы и технологии, 050202 – «Информатика» и др. Оно обладает яркой практической и проблемно-ориентированной направленностью. В нем собрано более 30 практических работ по всем основным темам, охватывающих установку, первичную настройку, обслуживание локальных сетей и прикладных служб. Материалы практических работ имеют кросс-платформенную составляющую, так как все они подготовлены для платформ — Windows и Linux. Вместе с практическими работами в пособии присутствует важная теоретическая информация, которая представлена в максимально эффективной и наглядной форме, понятной специалистам различных профилей. Учебное пособие предназначено для студентов, аспирантов и преподавателей, занимающихся изучением и преподаванием дисциплин, связанных с компьютерными сетями и сетевыми технологиями.

Содержание

Введение

Настоящее учебное пособие разработано с учетом программы дисциплины «Компьютерные сети» для специальностей 230201 – «Информационные системы и технологии, 050202 – «Информатика», и будет полезно для других специальностей, изучающих компьютерные сети, информационные и сетевые технологии. Учебное пособие имеет два раздела – теоретический материал и практические работы.

Раздел с теоретическим материалом представлен в сжатой форме, с помощью схем и диаграмм, максимально наглядно демонстрирующих принципы работы протоколов и технологий. Цель подобного представления материала состоит в том, чтобы более интенсивно погрузить обучаемых в сущность сетевых протоколов и принципов их работы. Существует достаточное число учебников описывающих аналогичный материал более подробно, однако, представляющих в лаконичной и высоко визуализированной форме практически нет. Материал каждого раздела организован отдельной страницей, что позволяет одним взглядом охватывать смысл и сущность размещенной информации. Благодаря этому теоретический раздел можно использовать как в качестве учебного пособия, так и в виде конспекта лекции и справочника.

При изложении теоретического материала используется диалектический и проблемно-ориентированный подход. Это означает, что поводом для изучения каждой следующей темы (протокола, технологии) служит некоторое противоречие в текущей ситуации, которое разрешается в новом разделе.

Учебное пособие содержит более 30 практических работ. Все они являются пошаговым руководством по выполнению практикума, для закрепления теоретического материала, развития навыков и компетенций будущих специалистов. Практическая часть имеет кросс-платформенную ориентацию, что означает опору на несколько различных программных платформ, в данном случае Windows и Linux. Это удобно как с точки зрения преподавателя при организации практикумов с выбором той или иной платформы, так и особенно со стороны обучающегося. Благодаря достижению в ходе работ похожих целей в различных операционных системах и программных платформах лучше закрепляются знания, развиваются профессиональные навыки и компетенции.

Особенности практических работ

Практические работы имеют яркую прикладную направленность. Каждая работа сопровождается описанием цели, указанием общего времени выполнения, списка необходимых инструментов, программного и аппаратного обеспечения, схемы сети и порядка действий. Указанный временной объем следует оценивать как средний расчетный, определенный автором опытным путем в ходе преподавания дисциплины «Компьютерные сети» в течение 6-ти лет. При этом один час представляет собой академический час.

В инструкциях к работам подробно представлены все манипуляции в сети, за исключением простейших, входящих в набор базовых информационно-коммуникационных компетенций. При этом во многих работах действия выстроены так, чтобы образовывались проблемные ситуации, которые заставляли бы активизировать познавательную и поисковую деятельность обучаемых. Инструкции, в данном случае, это не руководство по эксплуатации, они организованы так, чтобы обучаемый, выполняя работу, осознавал и понимал смысл выполняемых действий и лучше чувствовал их значимость. Однако все промежуточные рассуждения опущены, и оставлены на рассмотрение в обстановке лабораторной работы.

Практические работы основаны на использовании виртуальных машин, для этого потребуется установка приложения для виртуализации десктопа VirtualBox. Можно использовать и другие программные средства для создания виртуальных машин, такие как VirtualPC и VMWare. Благодаря такому подходу каждый обучаемый воссоздает учебную сеть с набором прикладных служб для тренировки и обучения сетевым технологиям на персональном рабочем месте.

Набор практических работ покрывает большинство тем теоретического раздела. Работы выстроены в последовательности, соответствующей традиционному подходу к настройке сетевых устройств и служб, а также с учетом порядка следования теоретического материала. Они начинаются с темы подключения сетевых интерфейсов, создания локальной сети, затем продолжаются настройкой маршрутизации, добавления различных прикладных служб, таких как HTTP, FTP и т.д.

Предполагается, что все практические работы выполняются каждым обучаемым индивидуально на собственном наборе виртуальных машин.

Поэтому следует учитывать доступный объем дискового пространства и оперативной памяти. Для каждой работы указан набор виртуальных машин и требуемых ресурсы. Многие работы связаны друг с другом и рассчитывается, что они выполняются последовательно. Вместе с тем, выполнять работы можно и в произвольном порядке, так как в инструкциях присутствуют необходимые ссылки на другие работы или имеются иные компенсирующие действия.

Все практические работы реализованы под платформы Windows и Linux. Такая особенность является сильным преимуществом данного учебного пособия, поскольку позволяет более глубоко изучить технологии, получить навыки работы в различных операционных системах и реализациях прикладных службы, а также прочувствовать общую концепцию, обобщить знания и провести параллели между выполняемыми операциями.

В пособии в качестве пользовательских платформ используются Windows XP и SLAX Linux (Live CD), а в качестве серверных — Windows Server 2003 и openSuSE. Данный выбор обосновывается широкой популярностью систем и относительно небольшими требованиями к системным ресурсам, по сравнению с более новыми версиями систем. В практических работах собраны наиболее часто используемые в реальной сети задачи, расположенные в удобной последовательности для изучения.

Темы в учебном пособии никогда не смогут считаться полностью и окончательно рассмотренными, поскольку информационные технологии быстро меняются, изменяется взгляды на те или иные подходы. Для получения большей информации и специальных сведений следует обращаться к дополнительной литературе. Однако, осознано выполнив весь объем практических работ обучаемый закрепит теоретические знания, приобретёт начальный опыт, выработает необходимые базовые навыки и компетенции будущего специалиста по сетевым технологиям.

Предлагаемые работы будут востребованы будущими специалистами по информационно-технологическому профилю для расширения опыта в сетевых технологиях, преподавателями соответствующих дисциплин в качестве методического пособия по организации лабораторных работ, а также практикующими администраторами в качестве базового справочника.

Предисловие

Развитие сетевых технологий происходит весьма активно и это процесс начался фактически с момента создания первого соединения между вычислительными машинами в Массачусетсе и Калифорнии через телефонную линию в 1965 году. Сейчас очевидно, что создание компьютерных сетей происходило под влиянием различных факторов — от научных до политических, поскольку сфера применения их чрезвычайно широка. Способствуют этому не только сами сетевые технологии и способы коммуникации, но и прочное взаимодействие их с вычислительными технологиями. Без развития вычислительных машин, аппаратного и программного обеспечения, компьютерные сети не достигли бы современного уровня и положения в промышленности и быту. Вместе с тем, совокупность несвязанных друг с другом компьютеров представляет существенно меньшую ценность, чем они же, подключенные в единую коммуникационную сеть.

Компьютерные сети стали платформой для реализации технологий в совершенно разных, несмежных с ними, областях. Например, в ядерных исследованиях, невозможно было бы проводить вычислительную обработку больших массивов данных на компьютерах расположенных на больших расстояниях друг от друга. В промышленности стало бы невозможным наладить взаимодействие и обмен данными между высокотехнологичным оборудованием, осуществить удаленное управление сложными процессами реального времени. В том числе без сетей и сетевых технологий немыслима индустрия связи и развлечений. Телевидение, мобильная телефонная связь, Интернет так быстро и прочно вошли в наш быт именно благодаря широкому развитию сетевых технологий.

Нет ничего удивительного, в том, что компьютерные сети широко применяются в педагогической практике. Благодаря этому организуются взаимодействия между субъектами образовательного процесса или воссоздаётся распределенная образовательная среда. Сети способствуют реализации универсального хранилища разнообразных ресурсов, в том числе и образовательного назначения. Глобальная сеть Интернет стала неисчерпаемым и легкодоступным источником информации и средством коммуникации.

Внутреннее устройство сетевых технологий изначально было достаточно сложным, их развитие и расширение разнообразия усугубило ситуацию.

Необходимость в специалистах, способных не только развивать их, но и сопровождать, и эксплуатировать, подобные системы существует практически на любом предприятии. Это означает, что их подготовка будет актуальной и востребованной, пока существуют сети в любом их виде. Следует заметить, что само понятие «сетевые технологии» является весьма широким и многогранным, поэтому и профиль специалистов может быть различным.

В узком смысле под термином «сетевая технология» понимается совокупность определенных инструментов и средств аппаратных и/или программных, нацеленных на реализацию одной или нескольких функций в коммуникационной сети. В более широком и традиционном техническом смысле сетевая технология — это согласованный набор стандартных протоколов и реализующих их программно-аппаратных средств, достаточных для построения вычислительной сети.

Несмотря на то, что изучение сетевых технологий и получение навыков по созданию и обслуживанию компьютерных сетей является нетривиальным, базовые знания этой области перестали быть атрибутом узких специалистов. Умение использовать сетевые возможности компьютерной техники давно соизмеримо с обычной грамотностью. Проникновение сетевых технологий в бытовые устройства предоставляют уникальные возможности по созданию «цифрового дома» и «умного дома». Все это приводит к тому, что основами компьютерных сетей и сетевых технологий должны владеть не только специалисты. Ведь совершенно нерационально было бы не использовать сетевые функции лишь потому, что отсутствуют знания по их настройке и эксплуатации. Естественно, что произвести настройку бытового и офисного оборудования, а также программных средств по инструкции не представляется сложным. Однако реализовать нужные функции в реальных условиях при существующих материальных и физических ограничениях является более важной задачей, научиться решать которую помогут базовые знания в области сетевых технологий.

Условные обозначения

Теоретический материал данного учебного пособия представлен в виде схем, диаграмм и таблиц. Для правильной интерпретации и лучшего понимания значения различных блоков определим основные условные обозначения.

В блоках прямоугольной формы отражаются различные тезисы, содержания понятий, а также различные другие объекты.

В прямоугольных блоках с закругленными углами отражаются тезисы, названия понятий.

В стилизованном бумажном листе с загнутым уголком отражается справочная и служебная информация.

В таком блоке, похожем на оторванный лист бумаге, располагаются некоторые вспомогательные сведения и комментарии.

68/UDP

В таких блоках отображается важная справочная, общепринятая и известная информация.

Разработчик: **IBM**

Год создания: **1991**

В таких блоках отображается справочная информация, относящаяся к данному разделу, вместо формулировок в длинных предложениях. Кроме того, при беглом пролистывании страниц, они должна вызывать ассоциации с разделом.

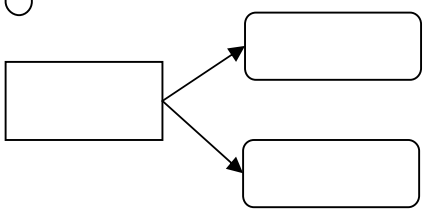
Такие блоки содержат формулировки противоречий и возникающих проблем, складывающихся или решаемых в текущем разделе.

Если имеется набор таких блоков.

То в них последовательность логически рассуждений.

С некоторым выводом.

Понятие – его определение…. присутствует в подобных блоках. Здесь приводятся важные определения понятий, которые нужно знать и помнить.



Различного рода стрелки показывают логические следования, принадлежность категорий понятий, а также связывают элементы блок-схем алгоритмов.

0 Введение в компьютерные сети

0.1 История телекоммуникации и компьютерных сетей в событиях и лицах

1800 – Алессандро Вольта изобретает источник постоянного тока. С этого события достаточно активно происходят открытия в области электричества.

1809 – Самуил Томас Земеринг создал электрохимический телеграфный аппарат.

1832 – Павел Львович Шиллинг создает первый электромагнитный телеграф.

1837 – Куком и Уинстоном создан электромагнитный телеграф в Великобритании. В это же время в США создан и запатентован телеграфный аппарат Морзе электромеханического типа. Опыты Чарльза Пейджа по передаче звуковых сигналов с помощью электрического тока.

1851 – проложен первый телеграфный кабель по дну пролива Ла-Манш.

1854 – Шарль Бурсель, инженер-механик вице-инспектор парижского телеграфа, разработал идею телефонирования и изложил принцип действия телефона в своей диссертации. До практического осуществления телефонной связи он не дошёл, но Ш. Бурсель первый, кто употребил слово «телефон».

1857-1866 – телеграфный кабель, проложенный под Атлантикой, связал Европу с Америкой, затем с Африкой, Индией, в 1870-х – с Китаем, Японией и Южной Америкой.

1861 – немецкий физик и изобретатель Иоганн Филипп Рейс продемонстрировал устройство, которое могло передавать музыкальные тона и человеческую речь по проводам. Аппарат имел микрофон оригинальной конструкции, источник питания (гальваническую батарею) и динамик.

1876 – в США запатентован аппарат под названием телефон Александром Беллом. Любопытно, что А. Белл подал заявку в Вашингтонское патентное бюро на свое изобретение 14 февраля 1876 года. Двумя часами позже заявку на «Устройство для передачи и приема вокальных звуков телеграфным способом» подал Э. Грей из Чикаго.

1891 – Никола Тесла в Сент-Луис (штат Миссури, США) в ходе лекций публично описал принципы передачи радиосигнала на большие расстояния.

1893 — Никола Тесла патентует радиопередатчик и изобретает мачтовую антенну, с помощью которой в 1895 г. передаёт радиосигналы на расстояние более 48 км.

1895 – 7 мая российский учёный Александр Степанович Попов на заседании Русского Физико-Химического Общества продемонстрировал прибор, названный им «грозоотметчик», который был предназначен для регистрации электромагнитных волн. Этот прибор считается первым в мире аппаратом беспроводной телеграфии, радиоприемником.

1896 – в Великобритании итальянец Гулиельмо Маркони подал патент «об улучшениях, произведенных в аппарате беспроводной телеграфии». Аппарат, представленный Маркони, в общих чертах повторял конструкцию Попова, многократно к тому времени описанную в европейских научно-популярных журналах.

1897 – при помощи аппаратов беспроводной телеграфии А.С. Попов осуществил прием и передачу сообщений между берегом и военным судном.

1945 – Ванневар Буш (Vannevar Bush) американский ученый, научный консультант президента США высказывает идею гипертекста, как особого вида текста связывающего документы по контексту, взамен традиционным библиотечным каталогам.

Рис. 1. Ванневар Буш

1956 – Закончилась прокладка первой трансатлантической кабельной телефонной линии связи между Великобританией и США (через Канаду).

Состоялся разговор между председателем компании «AT&T» в Нью-Йорке и министром Почтовой связи Великобритании. Стоимость линии обошлась в 42 миллиона долларов. Было проложено два кабеля, то есть по одному в каждое направление, расположенные на расстоянии 32 км друг от друга. По длине каждого кабеля был установлен 51 ретранслятор, содержащий около 5000 элементов, суммарной стоимости около $100000. Начальная пропускная способность 12 одновременных разговоров.

Для телефонной связи необходимые усилители погружали вместе с кабелем на дно океана, где они должны работать десятки лет. Первые океанские, встроенные в кабель усилители, были ламповыми и питались с берега, по тем же проводникам коаксиальной пары, по которым осуществлялась связь, на расстоянии 2-3 тысячи километров. Лампы были рассчитаны на срок службы линии связи, т.е. на десятки лет, и работали весь этот срок исправно. До появления спутниковых линий связи подобные кабельные линии являлись самым надежным и качественным средством связи.

В тот же год английская компания «Multitone» впервые в мире представила систему персонального радиовызова (пейджинговую систему), которая была развернута в одной из лондонских больниц.

1957 – 4 октября в СССР состоялся запуск первого искусственного спутника Земли. Только через полгода 1 февраля американцы запустили собственный спутник.

1958 – лауреаты Нобелевской премии Артур Шавлов и Чарльз Г. Таунс разработали лазер.

В тот же год в США произошло другое важное событие. Было создано агентство ARPA – Advanced Research Projects Agency, Агентство Передовых Исследовательских Проектов. Одна из задач агентства, по заказу правительства США, заключалось в создании сети передачи данных как средства коммуникации, способной выдержать ядерную атаку.

1961 – студент Леонард Клейнрок (Leonard Kleinrock) из Массачусетского технологического института (Massachusetts Institute of Technology) опубликовал свою первую статью по пакетной коммутации.

1962 – в ARPA был приглашен Джозеф Карл Робнетт Ликлайдер (Joseph Carl Robnett Licklider), который впервые изложил идеи интерфейсов, работающих по принципу указания и выбора (point-and-click), описал многие сервисы

информационного общества и сети Интернет, электронные библиотеки, электронную коммерцию, удаленное банковское обслуживание.

Рис. 2. Леонард Клейнрок

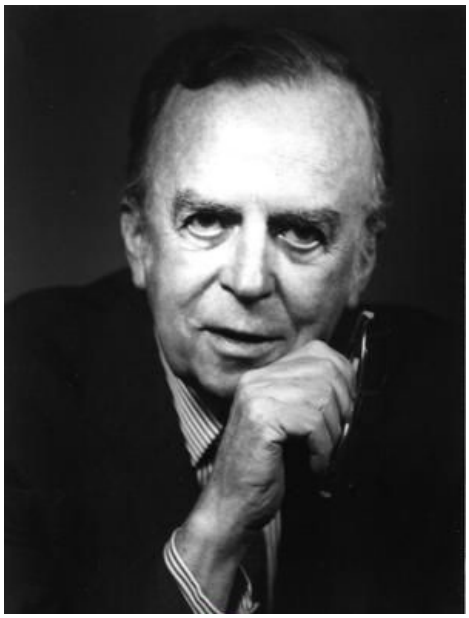


Рис. 3. Джозеф Карл Робнетт Ликлайдер

1963 – создан институт IEEE (Институт инженеров по электронике и электротехнике). Данная организация играет важную роль в стандартизации многих сетевых технологий.

1964 – 18 июня открылось телефонное обслуживание по трансатлантическому кабелю между Японией и США. Кабель емкостью 138 телефонных каналов и длиной 9822 км был проложен от Oahu (Гавайи) до Японии и соединился с существующими кабелями (от Гавайев до материковых штатов, Канады и Австралии).

1965 – под руководством Джозефа Ликлайдера исследователи Лоренс Робертс и Томас Меррилл соединили компьютер TX-2 в Массачусетссе с компьютером Q-32 в Калифорнии, используя низкоскоростную телефонную линию.

1966 – в Англии Чарльз Г. Као и Джордж А. Хокем предложили использовать стекловолокно для передачи света. Для построения эффективных систем оптической связи необходим коэффициент затухания не более 20 дБ/км. В то время был достигнут коэффициент около 1000дБ/км.

1968 – появились узлы сети ARPANet, первый из которых был построен в Калифорнийском университете в Лос-Анджелесе (University of California in Los-Angeles, UCLA), второй – в Стенфордском исследовательском институте (Stanford Research Institute, SRI).

1969 – в сентябре состоялась передача первого компьютерного сообщения между этими центрами, что фактически ознаменовало рождение сети ARPANet. К декабрю 1969 г. ARPANet насчитывала 4 узла, в июле 1970 г. – восемь, а в сентябре 1971 г. уже 15 узлов.

1969 – в IBM разработан язык GML (General Markup Language – общий язык разметки) для решения задач переноса документов между различными платформами и ОС. В 80-е годы – GML расширен и принят ISO в качестве международного стандарта SGML

1970 – фирма "Корнинг Инкорпорэйтид" произвела оптические волокна со ступенчатым профилем показателя преломления и достигла коэффициента затухания менее 20 дБ/км на длине волны 633 нм.

1970 – Деннис Ритчи (Dennis Ritchie) и Кеннет Томсон (Ken Thompson) выпускают первую версию Unix.

Рис. 4. Деннис Ритчи

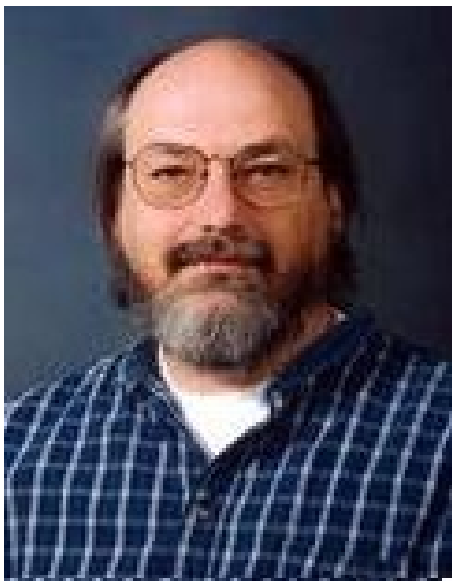


Рис. 5. Кеннет Томсон

1971 – Программист Рэй Томлисон разработал систему электронной почты(e-mail). Впервые был применен символ @ в адресации абонентов (@ – коммерческая эт, на жаргоне «собака», «обезьяна», «лягушка», «розочка» и т.д.).

1972 – Боб Меткалф (выпускник Массачусетсского технологического института; аспирант Гарвардского университета) сотрудник фирмы Xerox разработал экспериментальный протокол Ethernet для связи компьютеров в локальную сеть (3 Мбит/с), чтобы обеспечить доступ клиентов к сетевому лазерному принтеру – в то время одной и перспективной разработке Xerox.

Рис. 6. Рэй Томлинсон

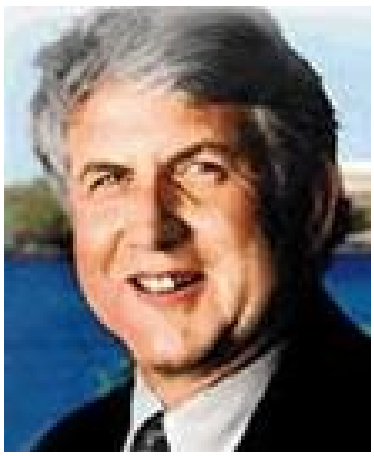


Рис. 7. Боб Меткалф

1974 – Винт Серф (Vint Cerf) и Роберт Кан (Robert Kahn) разработали набор сетевых протоколов TCP/IP

1975 – в феврале студенты Гарвардского университета Билл Гейтс и Пол Аллен основали компанию по разработке программного обеспечения для микрокомпьютеров, позже названной Microsoft.

1977 –десятки научных и военных организаций через телефонные, радио и спутниковые каналы связи уже соединены в единую сеть.

1979 – протокол Ethernet стал общепризнанным коммерческим стандартом передачи данных со скоростью 10 Мбит/с. Боб Меткалф основывает корпорацию 3Com

1980 – первая коммерческая ВОЛС (волоконно-оптическая линия связи) между Бостоном и Ричмондом в США. Скорость 45Мбит/с.

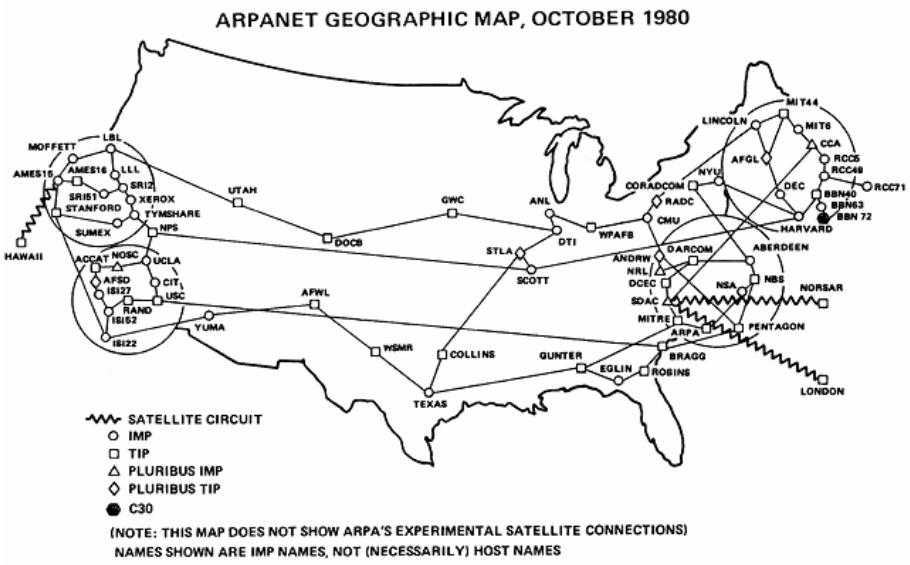


Рис. 8. Карта сети ARPA, октябрь 1980 года

1980 – Джон Постел разработал протокол UDP (User Datagram Protocol).

1982 – Джон Постел разработал протокол SMTP (Simple Mail Transfer Protocol).

1983 – научный сотрудник института информатики Пол Мокапетрис разработал систему доменных имен DNS.

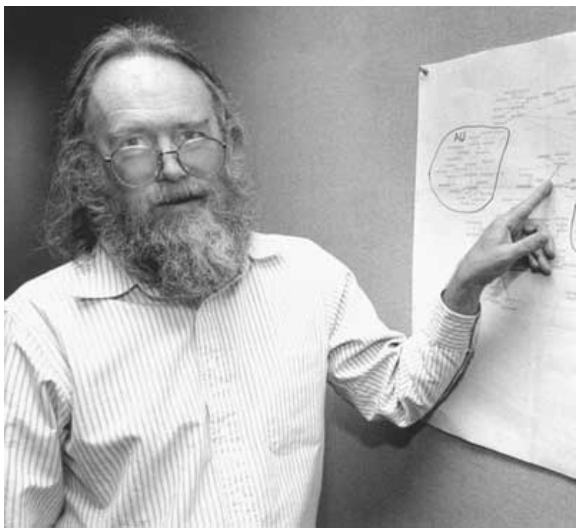


Рис. 9. Джон Постел

Рис. 10. Пол Мокапетрис

1985 – Джон Постел разработал протокол FTP (File Transfer Protocol).

1987 – Система DNS стал признанным стандартом и произошел массовый переход на DNS с устаревшего централизованного каталога имен и адресов хостов.

1987 – Через Атлантический океан проложен первый волоконно-оптический кабель.

1983 – принятие протоколов TCP/IP.

1986 – Национальным Фондом Науки США (The National Science Foundation – NSF) запущена в эксплуатацию NSFNet. В 1996 году NSFNet была приватизирована. В академических кругах это решение признано ошибочным, и с того же года ведутся эксперименты по воссозданию некоммерческой сети научных и образовательных учреждений.

1991 – 17 сентября финский программист Линус Торвальдс в новостной группе ОС Minix выложил исходный код собственной операционной системы Linux для общедоступной загрузки.

1991 – Тим Бернерс-Ли специалист из CERN разработал Протокол Передачи Гипертекста (HyperText Transmission Protocol – HTTP) и первую версия

HTML (HyperText Makeup Language) для обмена научно-исследовательской информацией и данными между учеными. 6 августа 1991 – Бернерс-Ли создал первый в мире веб-сайт. В том же году начата разработка службы – Всемирная паутина (World Wide Web - WWW), в Европейском центре ядерных исследований (European Center for Nuclear Research, CERN).

Рис. 11. Тим Бернерс-Ли

1993 – в Национальном центре суперкомпьютерных приложений (National Center for Supercomputing Applications, NCSA) создан первый – браузер Mosaic.

1993 – Протянута ВОЛС Карлслунде (Дания) – Кингисепп (Россия).

1995 – Реализован Восточный проект (Россия – Япония – Корея) морская волоконно-оптическая линия соединила города: Находка – Наоэцу – Пусан. Линия длиной 1762 км со скоростью передачи 560 Мбит/с по технологии PDH. Позднее морской участок соединен с наземным участком Находка – Хабаровск.

В последующие годы реализован Южный проект: морская оптическая линия ИТУР (Италия - Турция - Украина - Россия) Прошла через города Палермо - Стамбул - Одесса - Новороссийск длиной 3540 км, скорость передачи 565 Мбит/с, PDH. Плюс наземная ВОЛС Новороссийск - Ростов-на-Дону - Москва, длиной 1683 км, скорость передачи 2448 Мбит/с, технология SDH (оборудование поставляла компания Siemens).

1994 – Американский предприниматель Джефф Безос создал портал для онлайн-торговли Amazon.com, а в 1995 году магазин начал онлайновую торговлю книгами. В июне 1998 года магазин начинает продавать музыкальные записи, а в ноябре того же года — видеопродукцию.

1995 – 4 сентября в Сан-Хосе (штат Калифорния) программист Пьер Омидьяр создал онлайновый аукцион под названием AuctionWeb как часть своего личного веб-сайта. В 1997 году проект переименован в eBay.

1996 – в качестве учебного проекта студентов Стэнфордского университета Ларри Пейдж и Сергей Брин работали над поисковой системой BackRub, а в 1998 году на её основе создали поисковую систему Google.

1997 – 23 сентября была официально анонсирована поисковая система yandex.ru. Как отдельная компания «Яндекс» образовался в 2000 году.

2000 – 9 марта финансист Джимми Уэйлсом и философ Ларри Сэнгером из компании Bomis основали проект Нупедии (Nupedia), бесплатного англоязычного энциклопедического онлайн-проекта, чьи статьи были написаны специалистами и рецензированы в рамках формального процесса.

2001 – 15 января как дополнительный проект к Нупедии появился проект Википедия.

2002г. – насчитывалось 170 млн. хостов (компьютеры в сети с уникальным IP-адресом), а также 689 млн. пользователей Интернет.

2003 – 28 октября студент-второкурсник Гарвардского университета Марк Цукерберг написал интернет-сайт Facemash, в которой использовались фотографии, размещенные по парам, с целью выбрать, кто из двух людей более привлекателен. Уже 4 февраля 2004 года начала работу социальная сеть Facebook.

2005г. – преодолен рубеж в 1 миллиард пользователей.

2005 – в феврале тремя бывшими работниками PayPal в Сан-Бруно (Калифорния) был создан сервис, предоставляющий услуги видеохостинга YouTube. В ноябре 2006 года была завершена покупка YouTube компанией Google.

2009 – 5 ноября начат прием заявок на регистрацию доменов в зоне .рф, а 12 мая 2010 года около 17:20 по московскому времени домен .рф делегирован в корневой зоне DNS.

2010 – с 22 января прямой доступ в Интернет получил экипаж Международной космической станции.

2011 – в январе организацией IANA были выделены последние доступные блоки IP-адресов (IPv4) класса A.

2011 – 27 января число пользователей сети Интернет достигло 2 миллиардов.

0.2 Развитие телекоммуникаций и сетей

- **Системы пакетной обработки** (50-е гг.). Создавались на базе мэйнфреймов, данные вводились пакетным образом через перфокарты. Преимущество в увеличении эффективности использования процессора, в ущерб эффективности работы пользователей. Подобный подход был обоснован высокой ценой машинного времени.
- **Терминальный доступ** (60-е гг.). У каждого пользователя в распоряжении устройства ввода и вывода информации, подключенные к одной вычислительной машине. Благодаря этому складывалось впечатление, что каждый единолично владеет ЭВМ. Удаленные терминалы можно было соединять через модем по коммутируемым линиям.
- **Компьютерные сети** (с конца 60-ч гг.). С появлением технологии соединения типа «компьютер-компьютер» для соединения удаленных мэйнфреймов друг с другом.

Компьютерная сеть (вычислительная сеть, сеть передачи данных) – это совокупность компьютеров, соединенных линиями связи. Линии связи образованы кабелями, сетевыми адаптерами и другими коммуникационными устройствами. Все сетевое оборудование работает под управлением системного и прикладного программного обеспечения.

Сети являются частным случаем распределенных систем:

- **Мультипроцессорные машины**. Вычислительная машина имеет несколько процессоров, взаимодействие осуществляется общей операционной системой через общую память. Достоинства: производительность и отказоустойчивость. Недостатки: дороговизна, низкая масштабируемость, зависимость от операционной системы.
- **Многомашинные системы**. Вычислительный комплекс состоит из нескольких ЭВМ соединенных напрямую друг с другом. Управляются разными ОС. Собственный набор внешних устройств и памяти. Комплекс предназначен для выполнения задач слабо связанных по данным. Достоинства: простота организации, низкая зависимость от операционной системы. Недостатки: низкая масштабируемость.
- **Вычислительные сети**. Программные и аппаратные связи в такой системе еще слабее, связь по данным в задачах должна быть слабее для большей эффективности. Преимущество: универсальность системы.

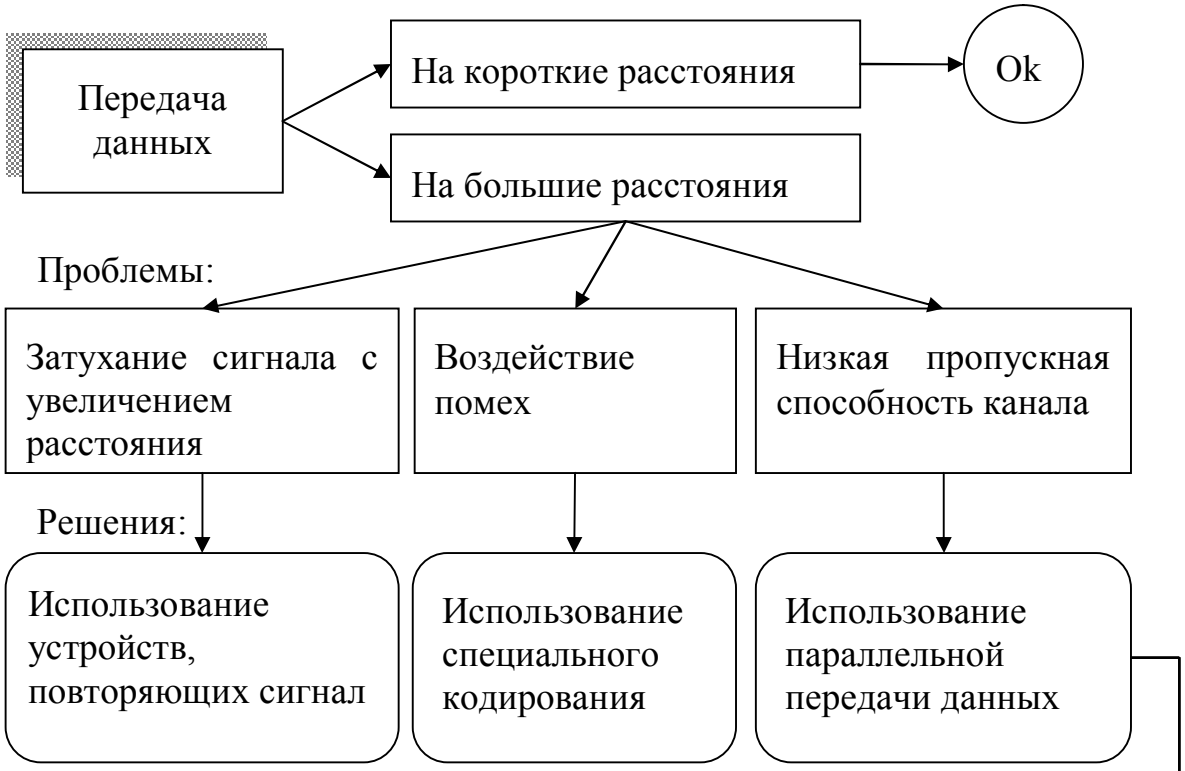
Преимущества компьютерной сети:

- Параллельные вычисления
- Отказоустойчивость системы, Распределение системы, Совместное использование данных
- Лучший показатель производительность/стоимость системы

1 Основы построения сетей

1.1 Проблемы построения сетей

1.1.1 Проблема физической передачи данных



Проблемы при параллельной передаче данных:

1.1.2 Проблемы выбора топологии

Топология сети — это структура графа, в узлах которого находятся конечные узлы сети, а ребрам соответствуют физические или информационные связи.

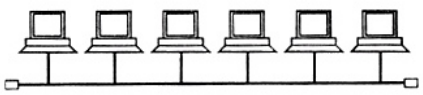
Физическая топология – определяет физическое соединение узлов линиями передачи данных.

Логическая топология – определяет пути информационных потоков в каналах передачи данных.

В общем случае физическая и логическая топологии в сети могут НЕсовпадать

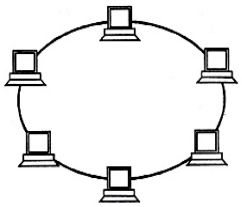
Поскольку невозможно каждый узел сети связать отдельными линиями связи со всеми остальными, необходимо выбирать одну из нескольких топологий

Общая шина



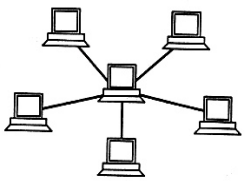
- Низкая стоимость проводника
- Низкая надежность
- Высокая расширяемость
- Низкая масштабируемость

Кольцо



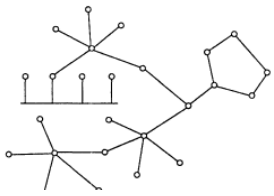
- Замкнутая шина
- Б*о*льшая надежность, чем в шине

Звезда



- Высокая надежность
- Высокая масштабируемость.
- Высокая стоимость проводника.
- Зависимость от центрального узла

Смешанная



- Соединение разнородных сетей
- Каждый сегмент устроен оптимальным для него образом

1.1.3 Проблема совместного использования линий связи

Редко когда возможно предоставить каждому компьютеру для соединения с другими собственную линию связи. Поэтому между компьютерами обычно прокладывают одну линию связи, вместе с этим возникает необходимость регулирования использования разделяемой линии связи. Данная проблема отсутствует только в полносвязной топологии.

Особо остро эта проблема стоит в топологиях шина и кольцо, где вся линия связи является разделяемой. Обычно управлением занимается либо сам узел или специальное устройство *арбитр шины*.

Типы линий связи

Разделяемая (shared) линия связи

Линия, которая совместно используется различными узлами для передачи данных.

Необходимо использовать различные виды коммутации клиентов (см.п.1.6, стр.38).

Используются в **локальных** сетях

Индивидуальная линия

От каждого узла существует индивидуальная линия связи с другим узлом или другим коммуникационным оборудованием.

Под узлом рассматривается порт коммуникационного устройства, а не конечный пользователь.

Используются в **глобальных** сетях

1.1.4 Проблема адресации узлов

Проблема адресации возникает при объединении 3-х и более узлов

Свойства идеальной адресации

- *Уникальность* — адрес уникально идентифицирует узел в сети
- *Иерархичность* — позволяет строить деревообразную структуру
- *Оптимальность* — минимальный размер, фиксированная длинна
- *Информативность* — адрес удобен для использования людьми

Типы адресации

Аппаратная адресация. В основе лежит адрес сетевого адаптера, назначаемый производителем.

05-34-F3-A7-ED-87

Составная числовая адресация. В основе лежит число, которое записывается в определенным образом через разделители.

195.197.112.204

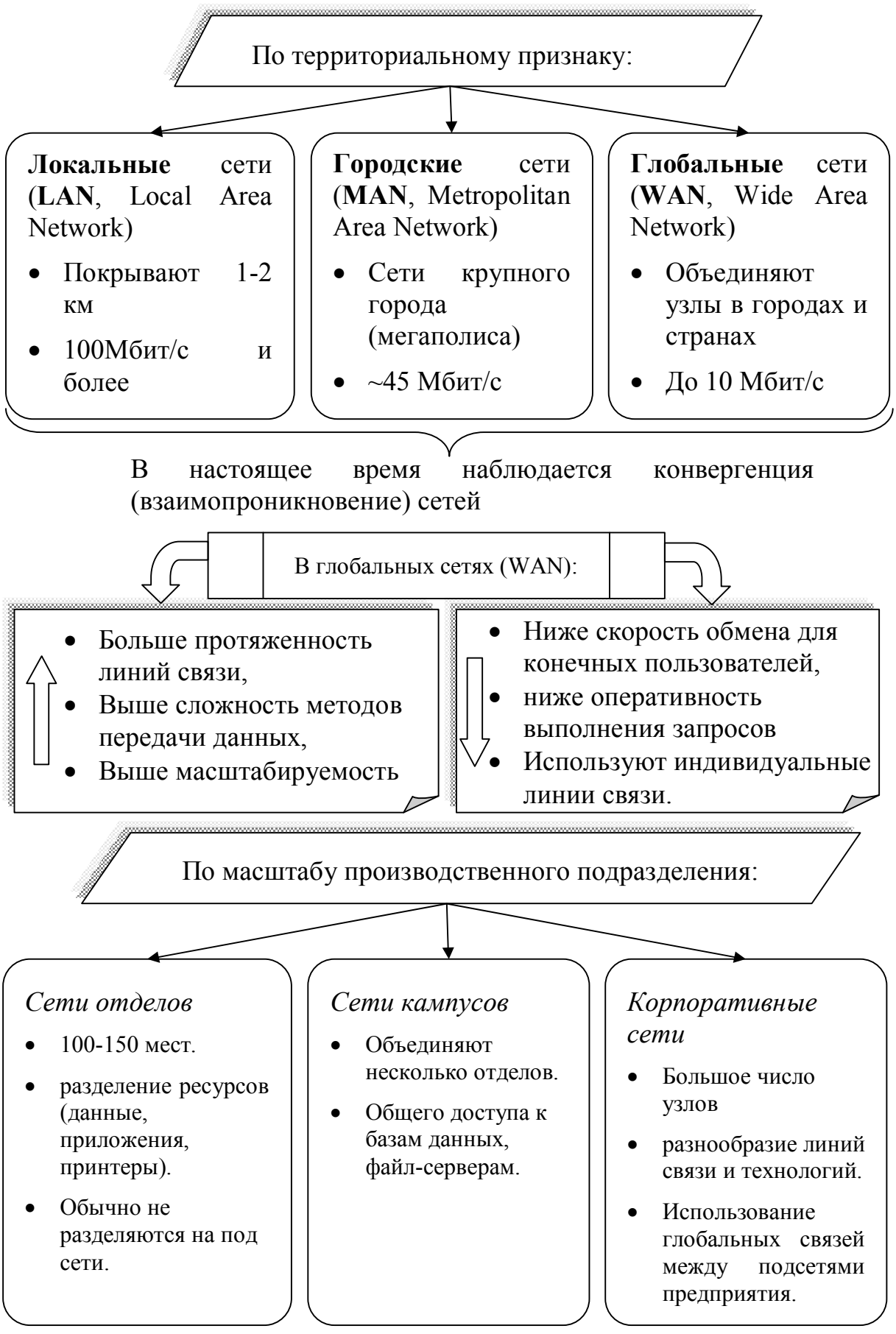
Символьная адресация. В основе лежит набор символов, записываемый с помощью разделителей.

www.opensee.ru

Критерий	**Аппаратная**	**Числовая**	**Символьная**
Уникальность	Да	Да	Да
Иерархичность	Нет	Да	Да
Информативность	Нет	Нет	Да
Оптимальный размер	Да	Да	Нет

Идеальной адресации НЕ существует, поскольку к ней предъявляются противоречивые требования.

1.2 Классификация компьютерных сетей



1.3 Основы передачи дискретных данных

Линии связи в зависимости от среды передачи данных:

Проводные (воздушные)

провода без изолирующих или экранирующих оплеток, проложенные между столбами по воздуху.

Кабельные (медные и волоконно-оптические)

состоят из проводников, заключенных в один или несколько слоев изоляции.

Радиоканалы наземной и спутниковой связи

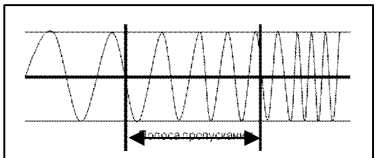
образуются с помощью передаваемых электромагнитных волн в атмосфере.

Основные характеристики линии связи:

Амплитудно-частотная характеристика – зависимость затухания мощности сигнала на выходе по сравнению с входом для всех возможных частот.

$$АЧХ = \mathrm{F}(\mathrm{A}_{\text{вых}}, \mathrm{A}_{\text{вх}})$$

Полоса пропускания (bandwidth) – это непрерывный диапазон частот, для которого отношение амплитуды выходного сигнала ко входному не менее 0.5.



Затухание (attenuation) – выличина относительного уменьшение амплитуды или мощности сигнала при передаче по линии сигнала определенной частоты.

$$A = 10 \cdot \log_{10}\left(\frac{P_{вых}}{P_{вх}}\right)$$

Пропускная способность (throughput) – характеризует максимально возможную скорость передачи данных по линии связи.

Формула Клода Шеннона:

$$C = F \log_2\left(1 + \frac{P_c}{P_ш}\right)$$

C –пропускная способность (бит/с)

F – ширина полосы пропускания

Формула Найквиста:

$$C = 2F \log_2 M$$

M – число различимых состояний сигнала

F – ширина полосы пропускания

1.4 Стандарты кабелей

При стандартизации кабелей принят протокольно-независимый подход.

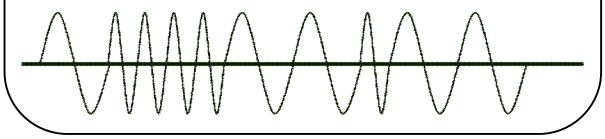
В стандарте оговариваются только электрические, оптические и механические характеристики типа кабеля или соединительного изделия.

<table>
<tr><td rowspan="10">Типы кабельных систем</td><td rowspan="2">коаксиальные кабели (coaxial cable)</td><td colspan="2">тонкий (thin) кабель, диаметр около 0,5 см, более гибкий.</td><td rowspan="2">Основное применение:
• кабельное телевидение,
• сети на основе топологии шина.</td></tr>
<tr><td colspan="2">толстый (thick) кабель, диаметр около 1 см, более жесткий.</td></tr>
<tr><td rowspan="8">кабели на основе витых пар проводов (twisted pair)</td><td rowspan="8">экранированные (shielded twisted pair, STP)
неэкранированные (unshielded twisted pair, UTP)</td><td>Кат. 1. Полоса частот до 4 КГц</td><td rowspan="4">Сейчас, практически не используются.</td></tr>
<tr><td>Кат. 2. Полоса 1 МГц</td></tr>
<tr><td>Кат. 3. Полоса 16 МГц</td></tr>
<tr><td>Кат. 4. Полоса 20 МГц</td></tr>
<tr><td>Кат. 5. Полоса 100 МГц.</td><td rowspan="4">Применение в: T1, 10BASE-T, 100BASE-T, 1000BASE-T (Gigabit Ethernet),
Token Ring 4/16 Мб/с, FDDI,
ATM 51/155 Мб/с, TP-PMD 100 Мб/с, 100VG-AnyLAN</td></tr>
<tr><td>Кат. 6. Полоса 200 МГц.</td></tr>
<tr><td>Кат. 7. Полоса 600 МГц.</td></tr>
<tr><td rowspan="3">оптоволоконные кабели (fiber optic)</td><td rowspan="2">Многомодовый (MM)
Траектории световых лучей имеют заметный разброс.
Более дешевый, но менее качественный.</td><td>Со ступенчатым показателем преломления.
Расстояние до 1 км; скорость до 100МБайт/с; длина волны 0,85 мкм</td></tr>
<tr><td>С градиентным показателем преломления.
Расстояния до 5 км; скорость до 100МБайт/с; длина волны 0,85 и 1,35 мкм; диаметр сердцевины 50 и 62,5 мкм.</td></tr>
<tr><td colspan="2">Одномодовый (SM) — все лучи проходят один и тот же путь, форма сигнала почти не искажается. Более дорогой, но имеет лучшие характеристики.
Расстояние до 50 км; скорость до 2,5 Гбит/с; длина волн 1,31 и 1,55 мкм; диаметр сердцевины 9 мкм.</td></tr>
</table>

1.5 Методы передачи данных на физическом уровне

Для передачи дискретных данных применяют два типа кодирования

На основе *синусоидального* несущего сигнала (более сложная аппаратура, скорость выше)



На основе *прямоугольных* импульсов (простота реализации, широкий спектр, скорость передачи ниже)

Методы аналоговой модуляции

Методы цифрового кодирования

Методы логического кодирования для улучшения цифровых кодов

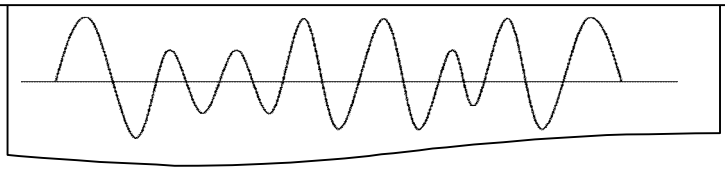
Требования, предъявляемые к методам кодирования

- Наименьшая ширина спектра результирующего импульса
- Способность распознавать ошибки
- Низкая стоимость реализации
- Самосинхронизация между передатчиком и приемником

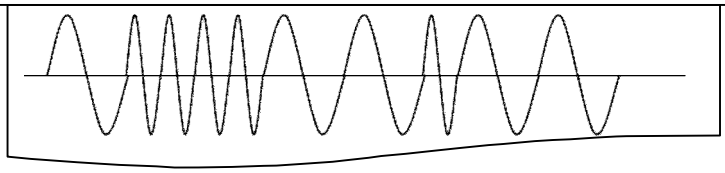
1.5.1 Методы аналоговой модуляции

Модуляция (лат. modulatio — мерность, размерность) — процесс изменения одного или нескольких параметров высокочастотного несущего колебания по закону низкочастотного информационного сигнала (сообщения).

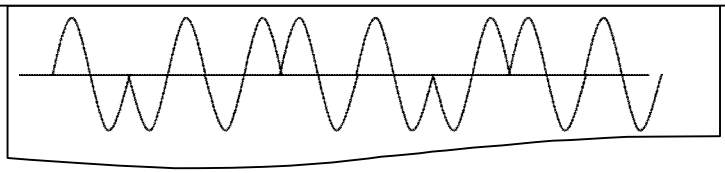
Амплитудная модуляция (BASK, Binary Amplitude Shift Keying) – для кодирования выбираются различные уровни несущей частоты. Недостаток – низкая помехоустойчивость.



Частотная модуляция (BFSK, Binary Frequency Shift Keying) – для логических состояний выбираются различные частоты.



Фазовая модуляция (BPSK, Binary Phase Shift Keying) – для логических состояний выбираются волны с различной фазой.



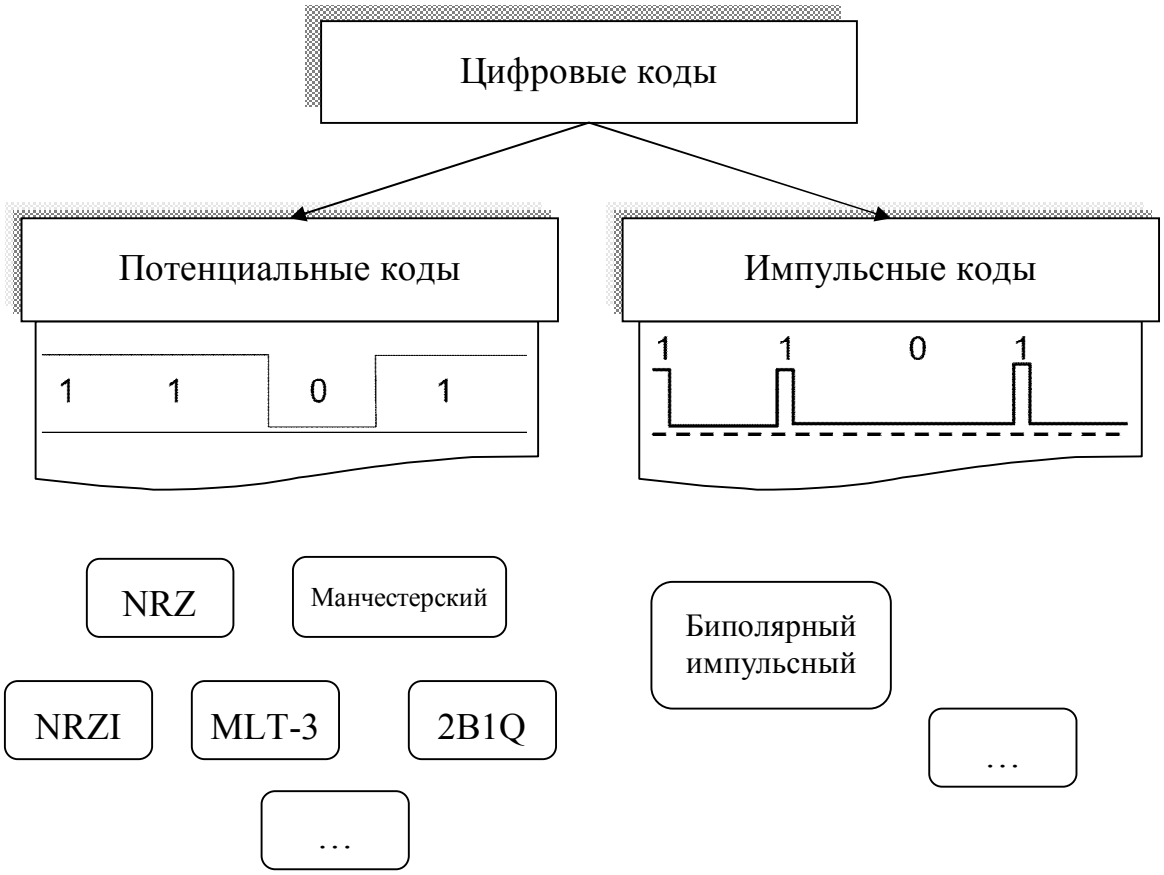
Смешанная

MFSK (Multilevel FSK) – многоуровневая частотная модуляция. Передается n бит за такт.

QPSK (Quadrature PSK) – квадратурная фазовая модуляция. Используется 4 фазы, передается 2 бита за такт.

QAM (Quadrature Amplitude Modulation) – сочетание фазовой и амплитудной модуляции. (QAM16, QAM32, QAM64)

1.5.2 Методы цифрового кодирования



Существует ряд цифровых кодов. Наиболее распространенные из них:

1) **Потенциальный код без возвращения к нулю (NRZ)**. При смене двоичного значения потенциал меняется на противоположный.

+ Прост в реализации, хорошо распознаются ошибки.

– Не обладает самосинхронизацией.

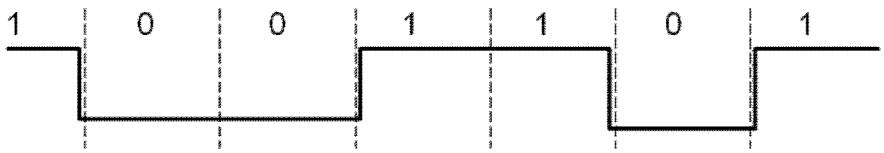


Рис. 12. Код NRZ

2) **Потенциальный код с инверсией при единице (NRZI)** 0 – предыдущий потенциал, 1 – инвертированный потенциал.

+ похож на AMI, но используется, когда третье состояние не желательно

– снижает полезную пропускную способность при длинных единицах

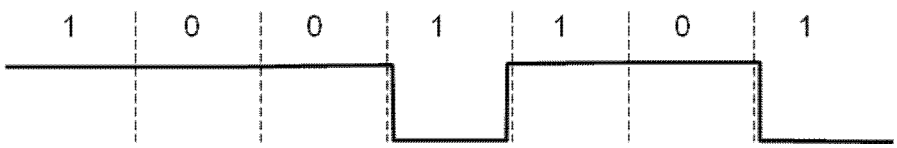


Рис. 13. Код NRZI

3) **Биполярный с альтернативной инверсией (AMI)**. Используется три уровня потенциала. 0 – нулевой потенциал. Каждая новая 1-а – кодируется положительным или отрицательным потенциалом.

+ узкий диапазон частот, наличие синхронизации при длительных последовательностях 1, отсутствие постоянной составляющей спектра.

– отсутствует самосинхронизация при длительных последовательностях 0, необходимо дополнительное увеличение мощности для распознавания 3х различных уровней.

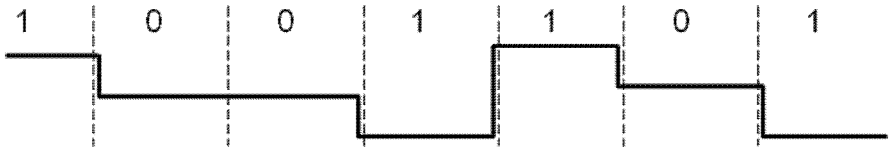


Рис. 14. Код AMI

4) **Биполярный импульсный код**: 1 – кодируется импульсом положительной полярности, 0 – отрицательной полярности. Каждый импульс длится половину такта.

+ обладает самосинхронизацией, прост в реализации.

– широкий диапазон частот, присутствие постоянной составляющей в спектре при длинных последовательностях одинаковых символов.

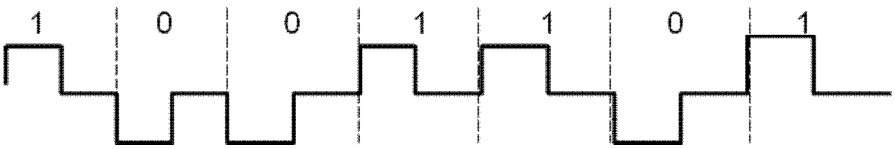


Рис. 15. Биполярный импульсный код

5) **Манчестерский код**: используется перепад потенциалов. 1 – перепад от низкого уровня к высокому, 0 – от высокого к низкому. В начале каждого такта может происходить служебный перепад сигнала, если нужно представить несколько 0 и 1.

+ самосинхронизация, отсутствие постоянной составляющей в спектре.

– более широкая полоса используемых частот.

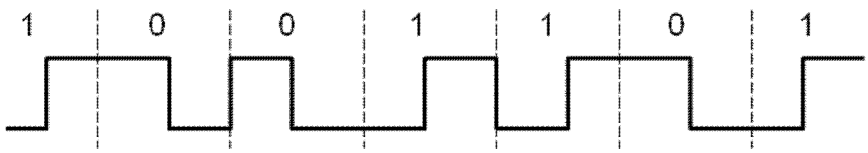


Рис. 16. Манчестерский код

6) **Трехуровневый код MLT-3**: используется 3 уровня сигнала, при 1 происходит переход на следующий уровень, при 0 уровень сохраняется.

+ более узкий, чем NRZ, диапазон частот, самосинхронизация при длительных последовательностях 1, отсутствие постоянной составляющей.

– отсутствие самосинхронизации при длинных последовательностях 0, необходим более мощный передатчик для распознавания 3х уровней.

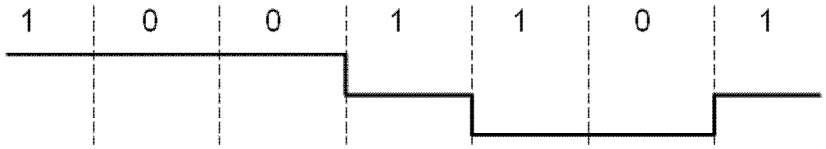


Рис. 17. Трехуровневый код MLT-3

7) **Потенциальный код 2B1Q**: каждые 2 бита кодируются одним из 4-х импульсов и передаются за 1 такт. Паре бит 00 соответствует потенциал – 2,5В, 01 – -0,8В, 10 – +0,8В, 11 – +2,5В.

+ еще более узкая полоса частот.

– наличие постоянной составляющей, отсутствие самосинхронизации.

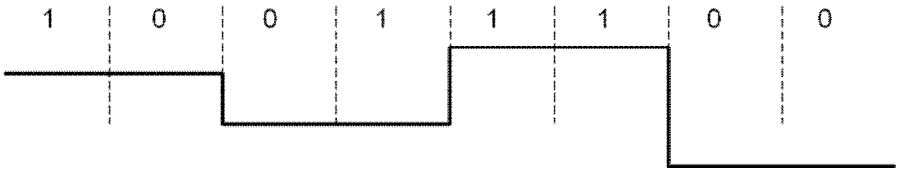
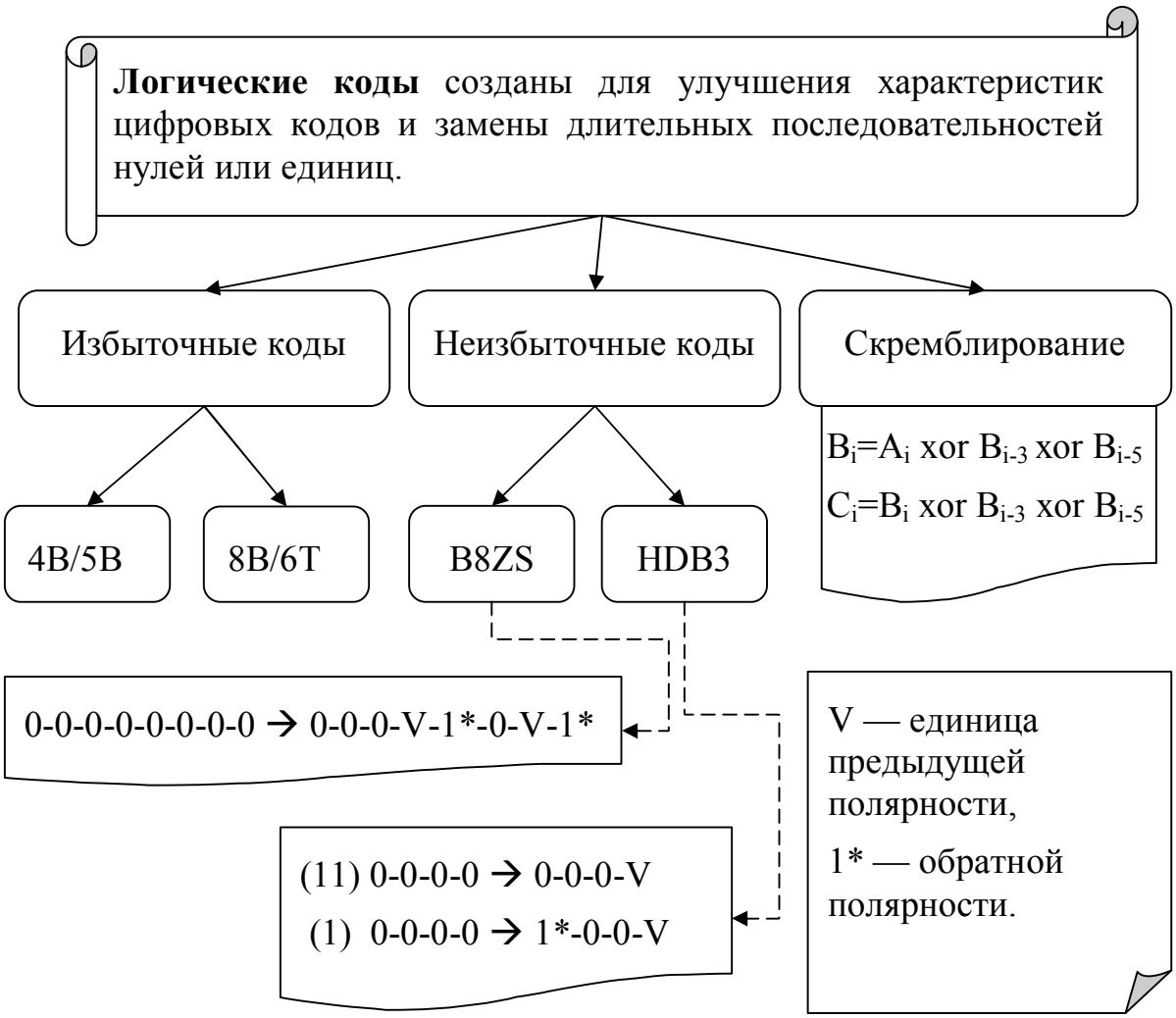
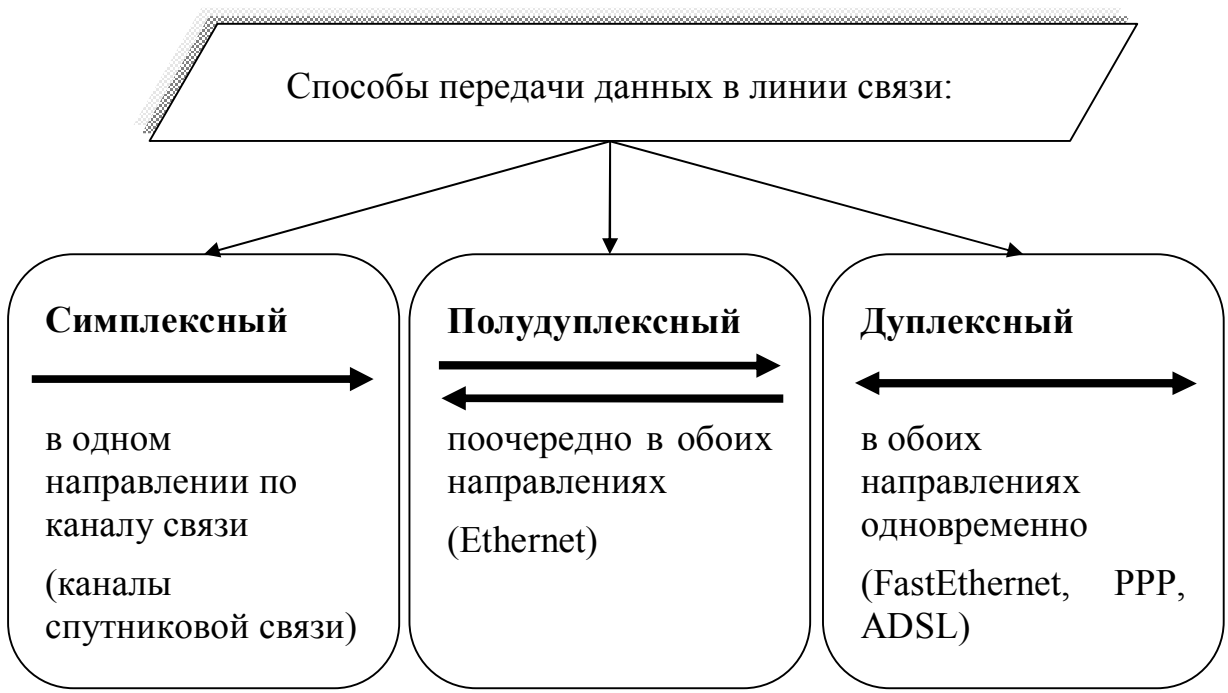


Рис. 18. Потенциальный код 2B1Q

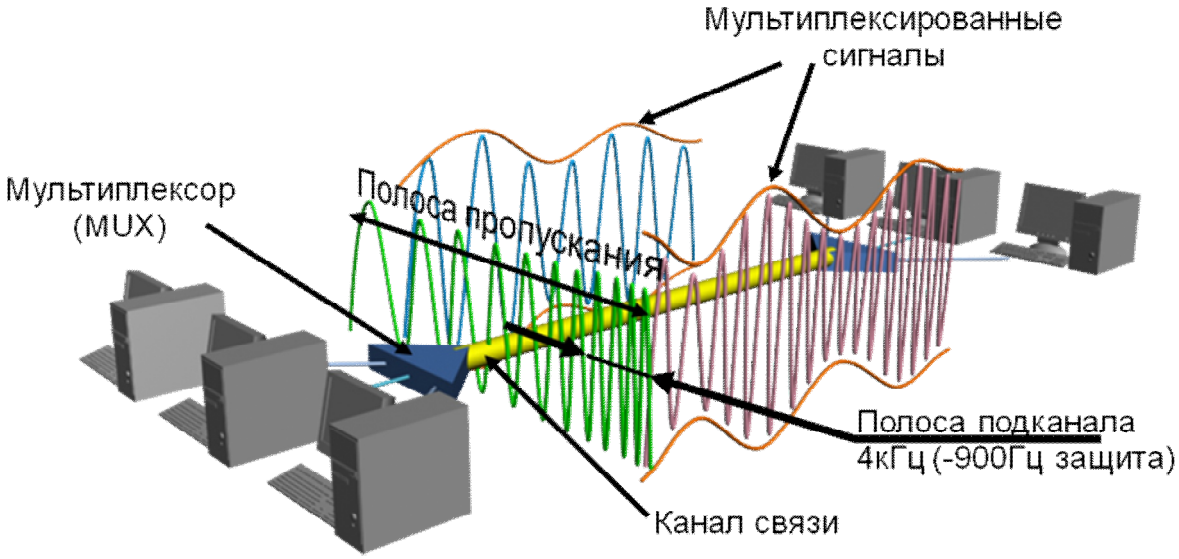


1.6 Методы коммутации



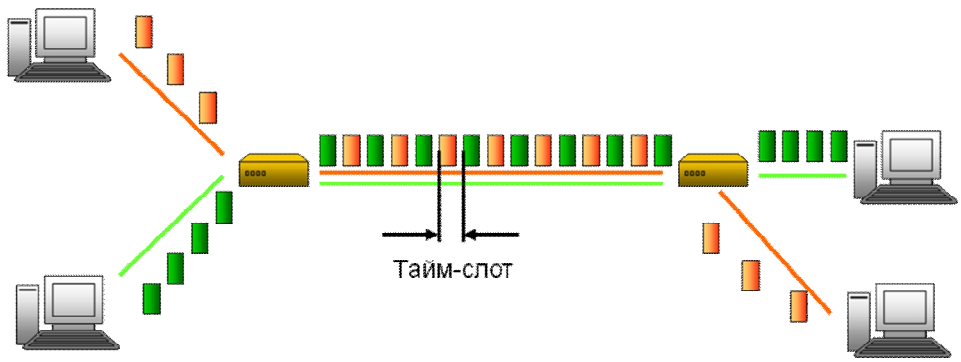
1.6.1 Коммутация каналов

Коммутация каналов на основе частотного мультиплексирования (FDM). Разработана для телефонных сетей. Применяется также для кабельного телевидения.



Коммутация каналов на основе разделения времени (TDM). Разработана при переходе на цифровые формы данных.

Мультиплексоры, коммутаторы и демультиплексоры работают в режиме разделения времени, поочередно обслуживая абонентские каналы. Цикл обычно равен 125мкс.

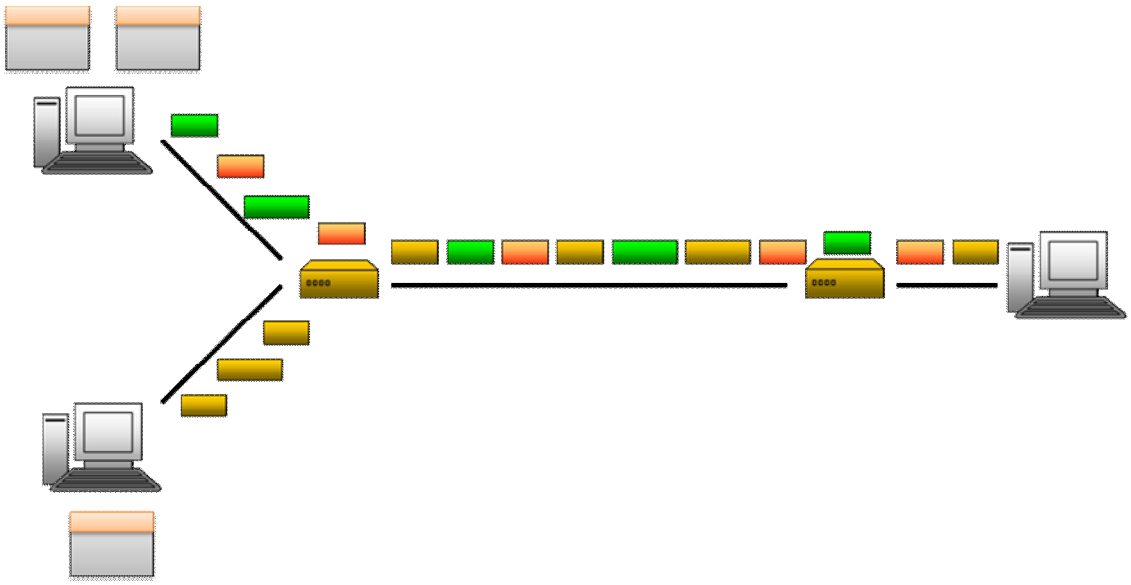


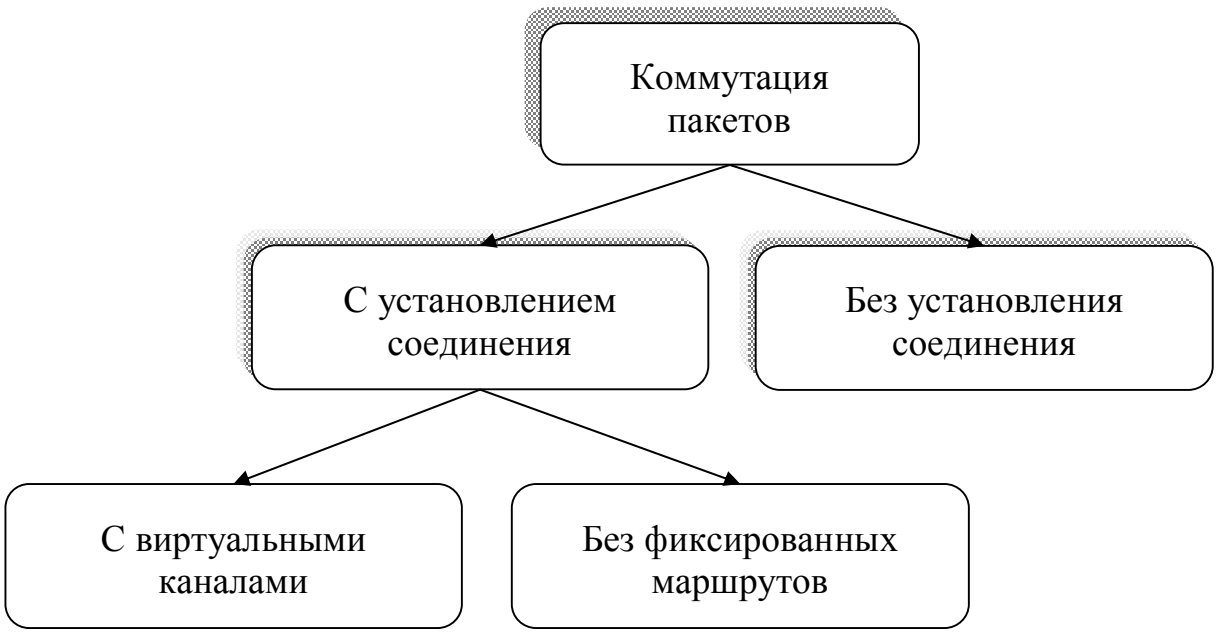
Коммутация на основе разделения по длине волны (WDM). Появилась в оптико-волоконных линиях. По одному физическому каналу связи данные передаются на волнах с разной длиной.

1.6.2 Коммутация пакетов

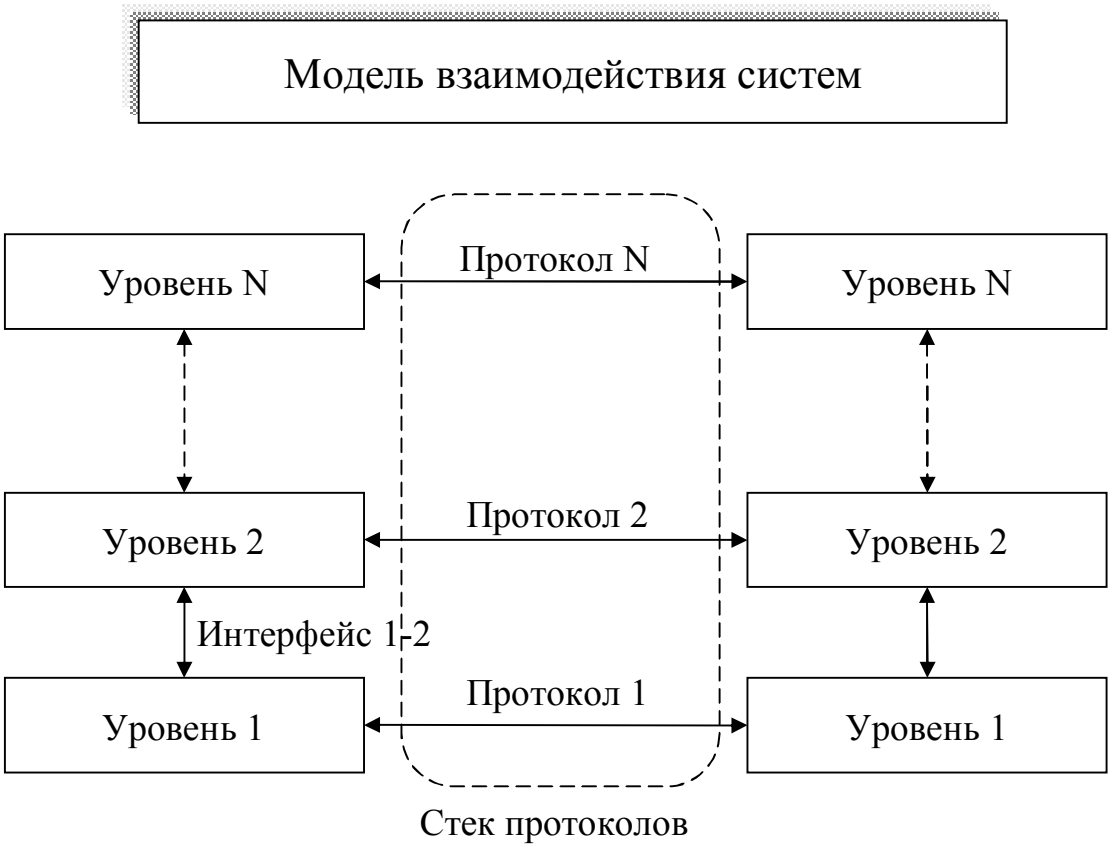
Леонард Клейнрок (Leonard Kleinrock), MIT	**1961** год

- Данные разбивают на пакеты
- Пакет снабжают заголовком
- Пакеты могут иметь переменную длину от 46 до 1500 байт.
- Коммутаторы пакетов, в отличии от коммутаторов каналов, буферизуют пакет (однако, существуют и другие режимы работы коммутаторов, см. п.2.2.5)





1.7 Модель открытых систем



Протокол — это правила, с помощью которых описывают форматы сообщений, которыми обмениваются одноуровневые компоненты в разных узлах.

Интерфейс — правила с помощью, которых стандартизуются форматы сообщений, передаваемых от одного уровня другому в одном узле.

Стек коммуникационных протоколов — это иерархический список протоколов, достаточный для организации взаимодействия.

1.7.1 Модель OSI ISO

International Organization for Standardization (ISO) – международная организация по стандартизации

Модель взаимодействия открытых систем (Open System Interconnection, OSI) разработана организацией ISO и др. в 70-е годы на основе опыта создания сетей.

- В модели OSI определены **7** базовых уровней.
- Каждый уровень реализует свои функции.
- В конкретном стеке протоколов может быть больше или меньше уровней

- Стандарты фирм (IBM, Microsoft, Adobe)
- Стандарты комитетов и объединений (ATM Forum, Fast Ethernet Alliance)
- Национальные стандарты, (IEEE, ANSI)
- Международные стандарты (ISO, ITU)

Приложения обращаются к реализации протоколов прикладного уровня. К ним относятся, например, HTTP, FTP и т.д.

На нижнем уровне — физическом — осуществляется передача сигналов. Например, с использованием кодов 4B/5B + манчестерский код по витой паре.

2 Сетевые протоколы и технологии

2.1 Физический уровень модели OSI

2.1.1 Свойства физического уровня

Физический уровень определяет передачу битов по физическим каналам связи.

На физическом уровне определяют характеристики сред:

- полоса пропускания, скорость передачи
- волновое сопротивление, напряжение тока сигнала
- тип кодирования

- Прикладной
- Представительный
- Сеансовый
- Транспортный
- Сетевой
- Канальный
- **Физический**

Передаются сигналы прямоугольной или синусоидальной формы

Кодирование (например): 4B/5B + скремблирование+ манчестерский код

Типы сред: Коаксиальный кабель, экранированная или неэкранированная витая пара, оптоволокно, радиоэфир

10 Base - T

Скорость передачи (Мбит/с): 10, 100, 1000, 10G и т.д.

Base (от baseband) – без мультиплексирование

2 – тонкий коаксиал

5 – толстый коаксиал

T – витая пара

F – оптоволокно

Физические ограничения:

Тонкий коаксиал – 185 м.

Толстый коаксиал – 500 м.

Витая пара – 100 м.

Оптоволокно – 5 км и более.

2.1.2 Оборудование физического уровня

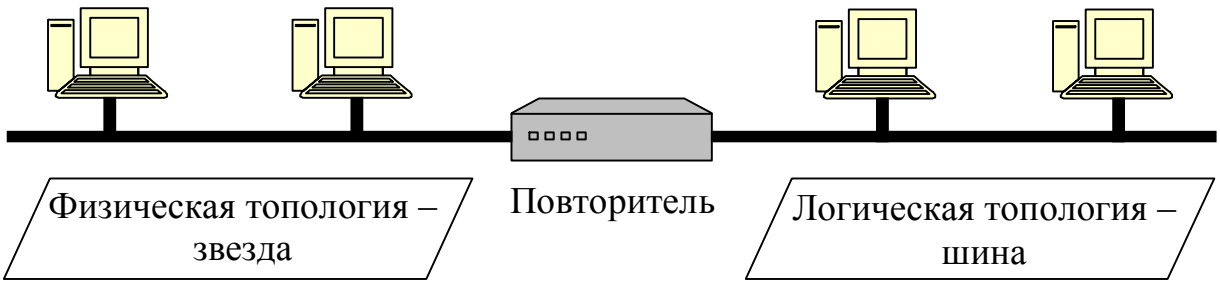
Повторитель (репитер, хаб, repeter, hub) – сетевое устройство, которое повторяет данные с одного порта на все остальные.

Повторитель имеет несколько портов для подключения узлов, обычно, 10, 16, 24 и т.д.

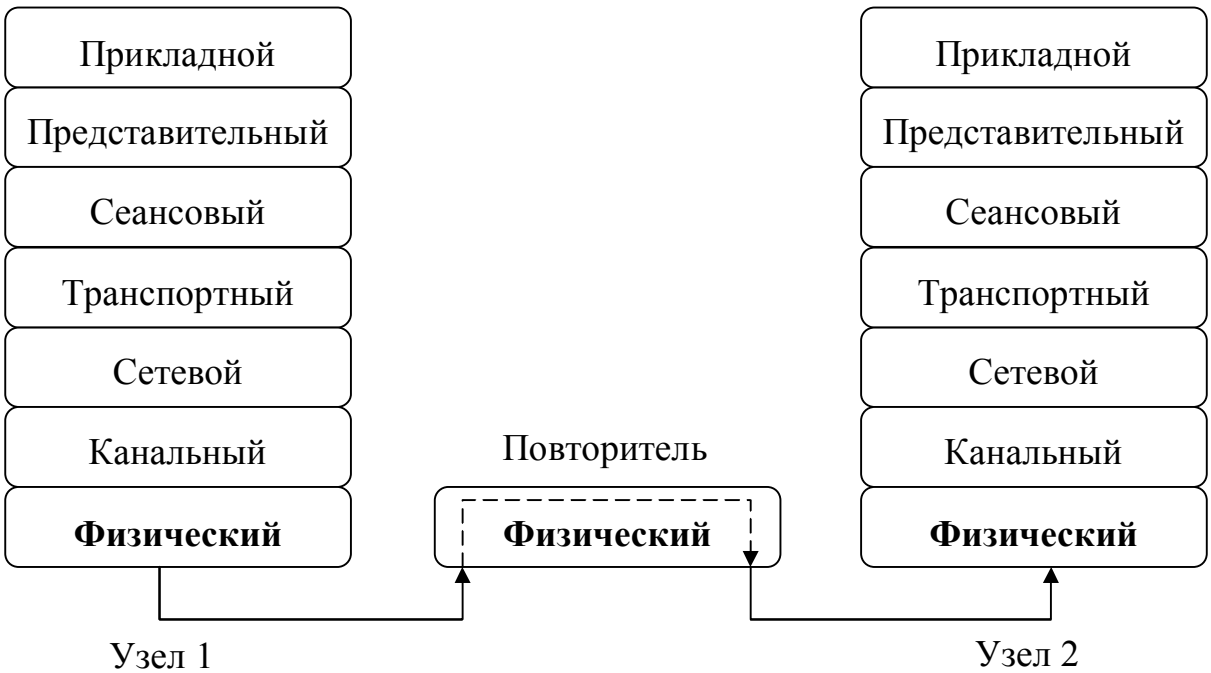
Повторители характерны для технологии Ethernet.

Повторитель меняет только физическую топологию.

Логическая топология НЕ изменяется.



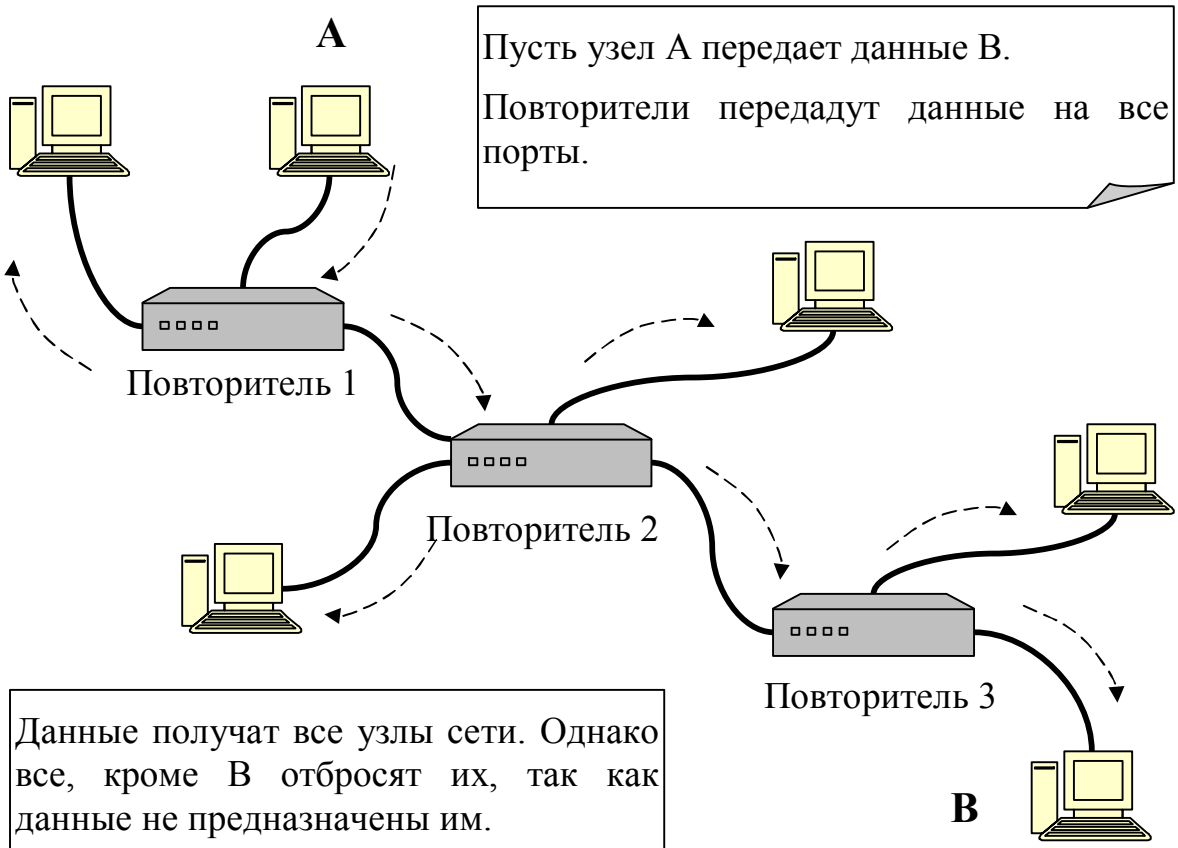
Место повторителя в стеке OSI



2.1.3 Проблемы масштабирования на физическом уровне

При организации сетей с помощью повторителей трафик каждого клиента передается до всех остальных.

То есть данные, предназначенные только одному клиенту, достигают всех других, подключенных к этой сети.



При увеличении числа клиентов (до 10-20) объем «лишнего» трафика увеличивается на столько, что дальнейшее наращивание числа клиентов и служб в сети невозможно.

Противоречие. При построении сети на основе повторителей возникает высокая конкуренция за разделяемую среду передачи данных.

Следовательно, необходима локализация трафика и логическая структуризация сети на более высоком уровне.

2.2 Канальный уровень модели OSI

Канальный уровень обеспечивает доставку кадра между двумя узлами сети определенной топологии, и реализованной по определенной технологии.

- Проверка доступности среды
- Захват среды для передачи данных.
- Из каждого блока передаваемых данных формируется кадр, который передается по сети.

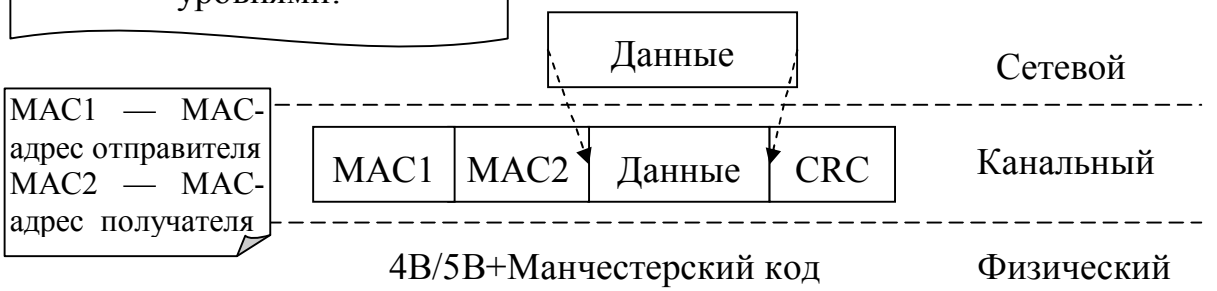
Прикладной
Представительный
Сеансовый
Транспортный
Сетевой
Канальный
Физический

Единица данных: **кадр (frame)**

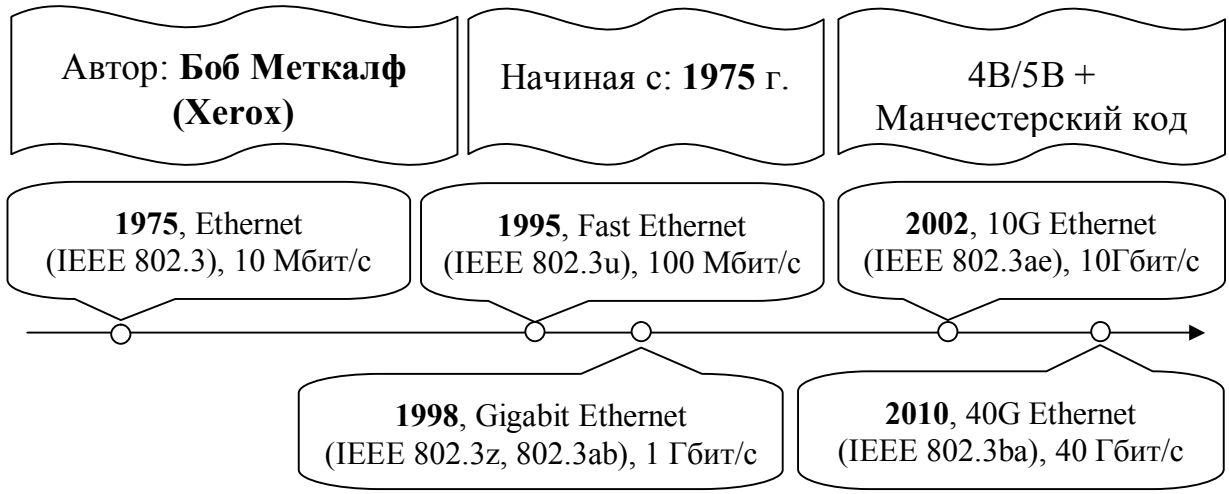
Тип адресов: **MAC-адреса**

Протоколы: **Ethernet (IEEE 802.3), Token Ring (IEEE 802.5), FDDI, SLIP, 100VG-Any_LAN, X.25, ATM, LAP-B, PPP, Wi-Fi, Bluetooth и др.**

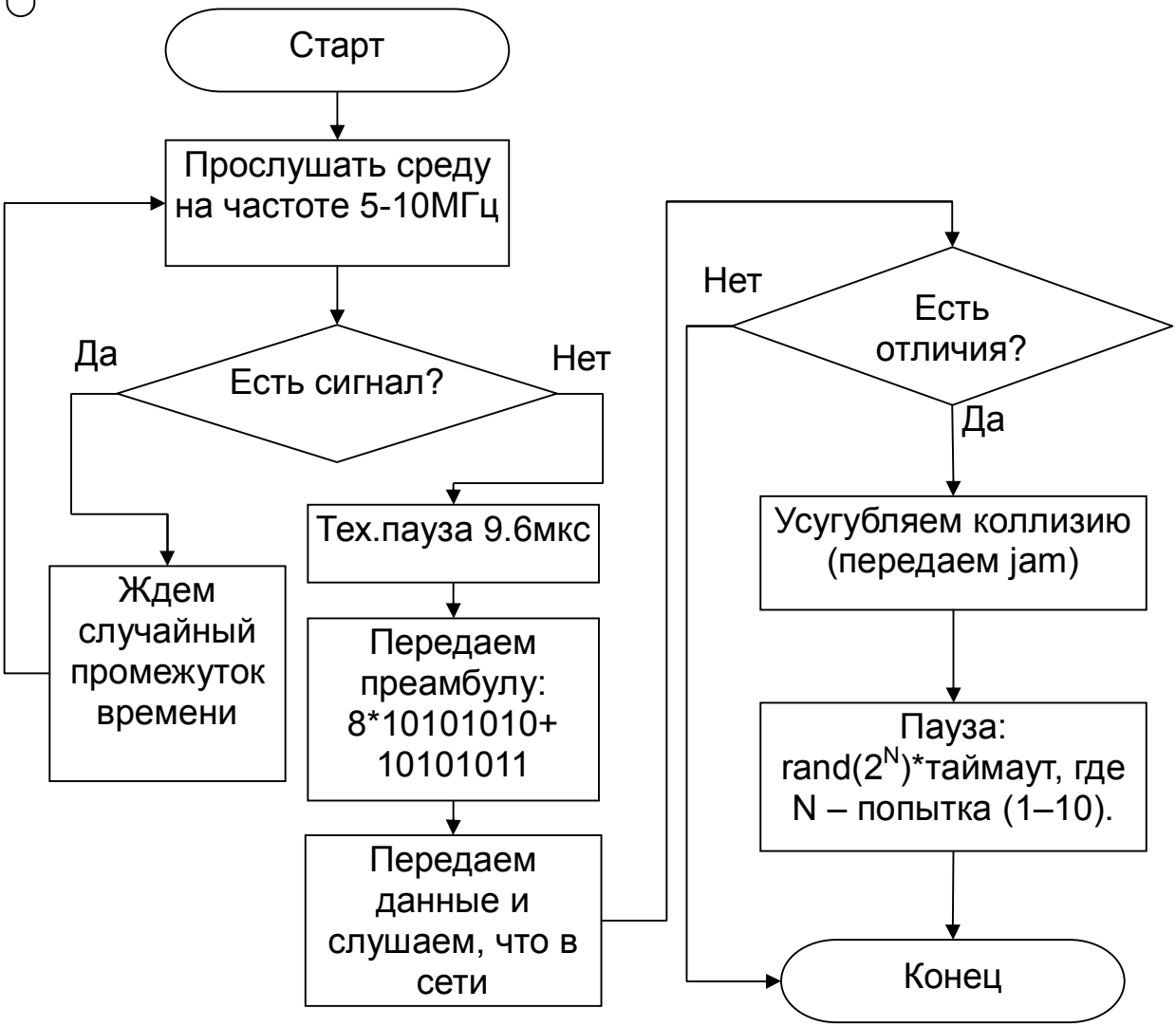
Передача данных между уровнями:



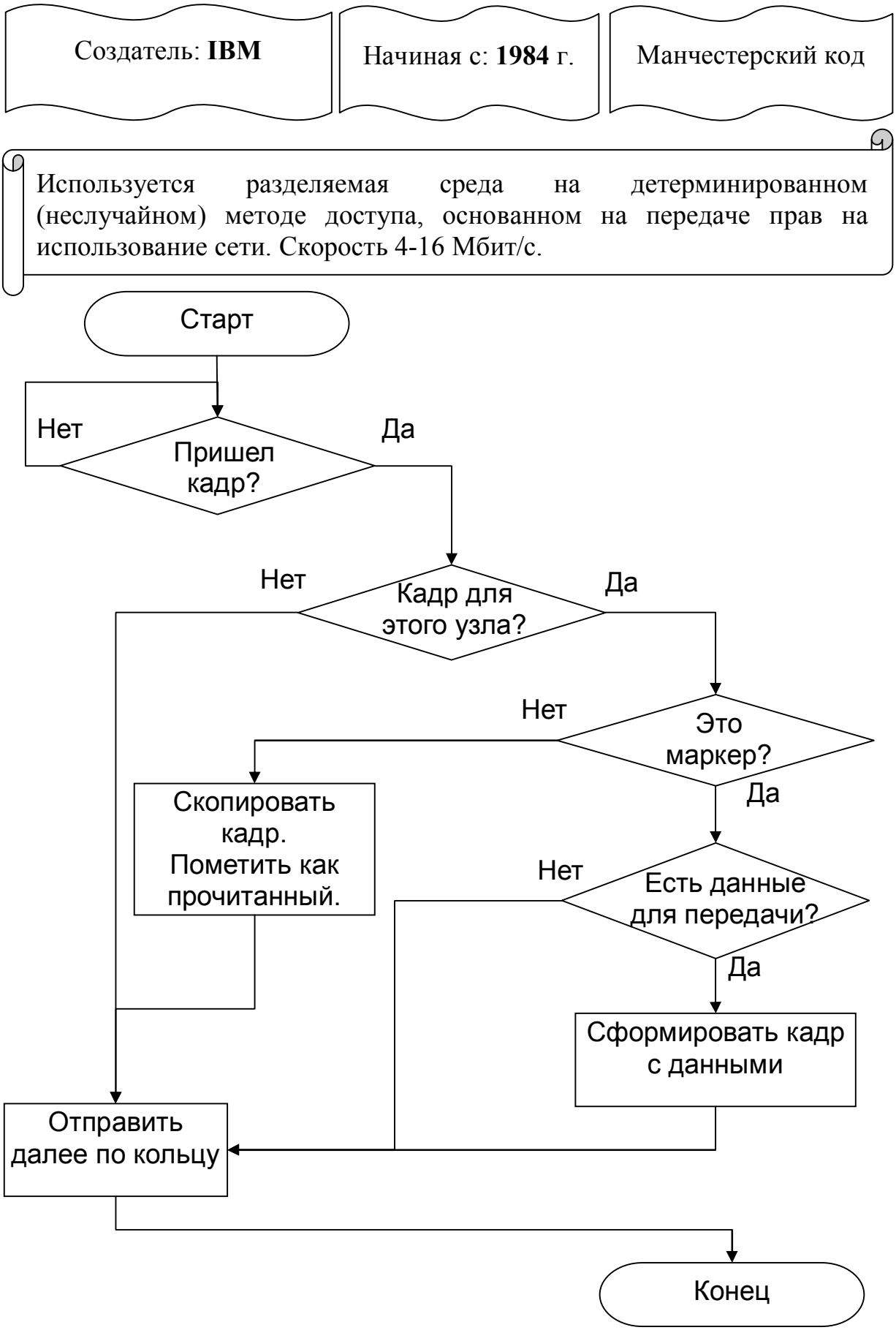
2.2.1 Протокол Ethernet (IEEE 802.3)



В основе работы метод коллективного доступа с опознаванием несущей и обнаружением коллизий **CSMA/CD** (Carrier sense multiply access with collision detection). Случайный (стохастический) доступ.



2.2.2 Протокол TokenRing (IEEE 802.5)



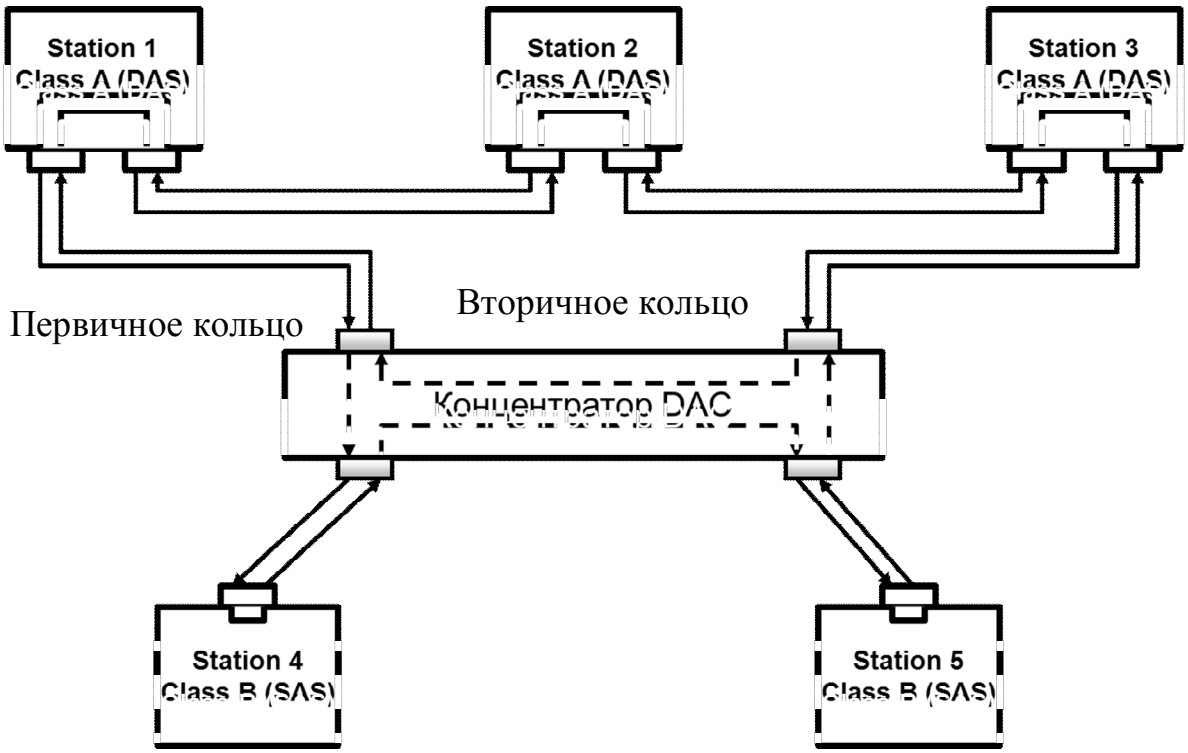
2.2.3 Протокол FDDI

Автор: **ANSI**	Начиная с: **1986** г.	4B/5B + NRZI

Используется разделяемая среда на детерминированном методе доступа схожем с TokenRing. Используется двойное кольцо, что повышает надежность при обрывах. Скорость до 100 Мбит/с.

Сеть работает в двух режимах:

- **Первый Thru (сквозной)**, нормальный – данные проходят через все узлы по первичному кольцу, второе не используется.
- **Второй Wrap (свернутый)**, при обрыве кольца – кольца сворачиваются средствами концентраторов или сетевых адаптеров.



SAS (Single Attachment Station),

DAS (Dual Attachment Station),

SAC (Single Attachment Concentrator),

DAC (Dual Attachment Concentrator)

2.2.4 Оборудования канального уровня. Мост и коммутатор

Сетевые адаптеры,

Network Interface Card (NIC) – устройство, позволяющее передавать в сеть и принимать данные из сети.

Мост (bridge), коммутатор 2-го уровня – устройство, предназначенное для объединения сегментов компьютерной сети.

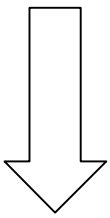
Коммутатор, свитч (switch) – многопортовый мультипроцессорный мост.

Функции сетевого адаптера:

- Захват сети
- Передача кадра с данными
- Распознавание кадров среди битов данных
- Проверка кадров

Цели структуризации сети с помощью мостов и коммутаторов:

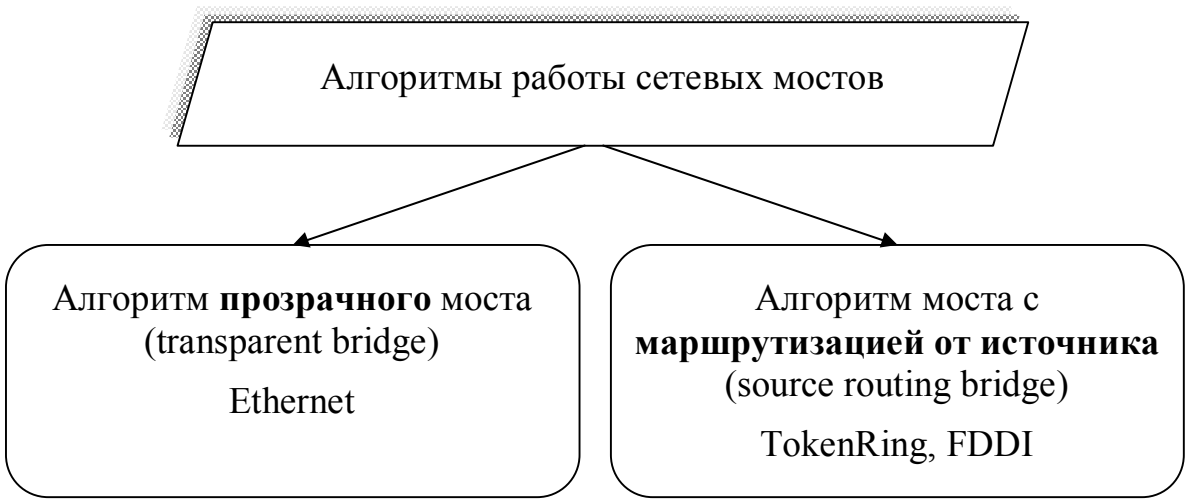
Локализация трафика и ограничение домена коллизий



Положительные следствия:

1. Увеличение пропускной способности сети, приходящейся на один узел
2. Увеличение гибкости сети
3. Повышение безопасности данных

2.2.5 Режимы коммутации и алгоритмы работы



2.2.6 Алгоритм прозрачного моста

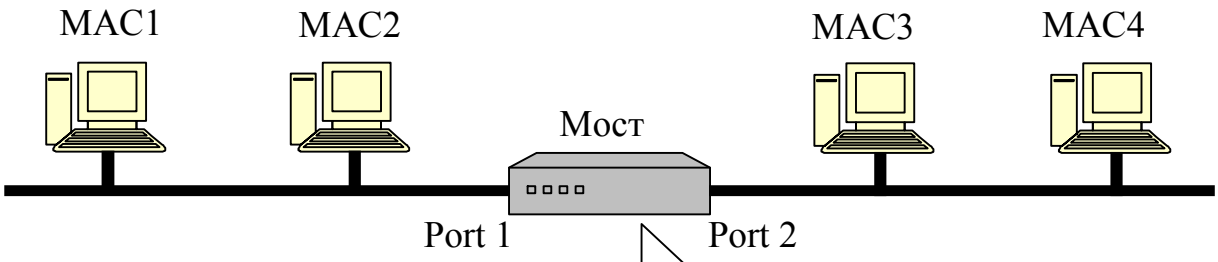
Проблема: Требуется автоматически определить порт, на котором находится клиент-получатель.

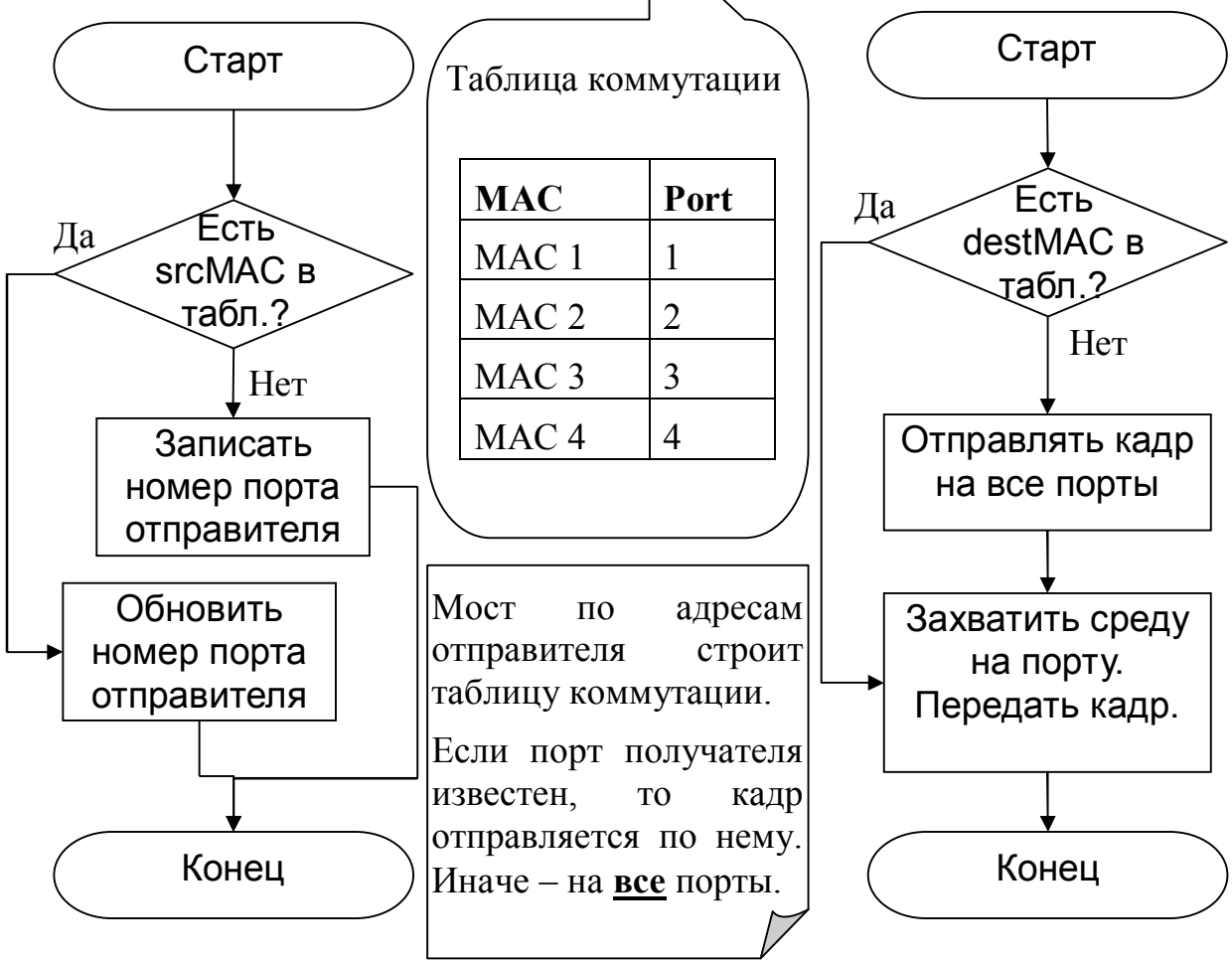
Задача: Необходимо составить таблицу соответствия «MAC-адрес — Порт».

Требуется автоматически.

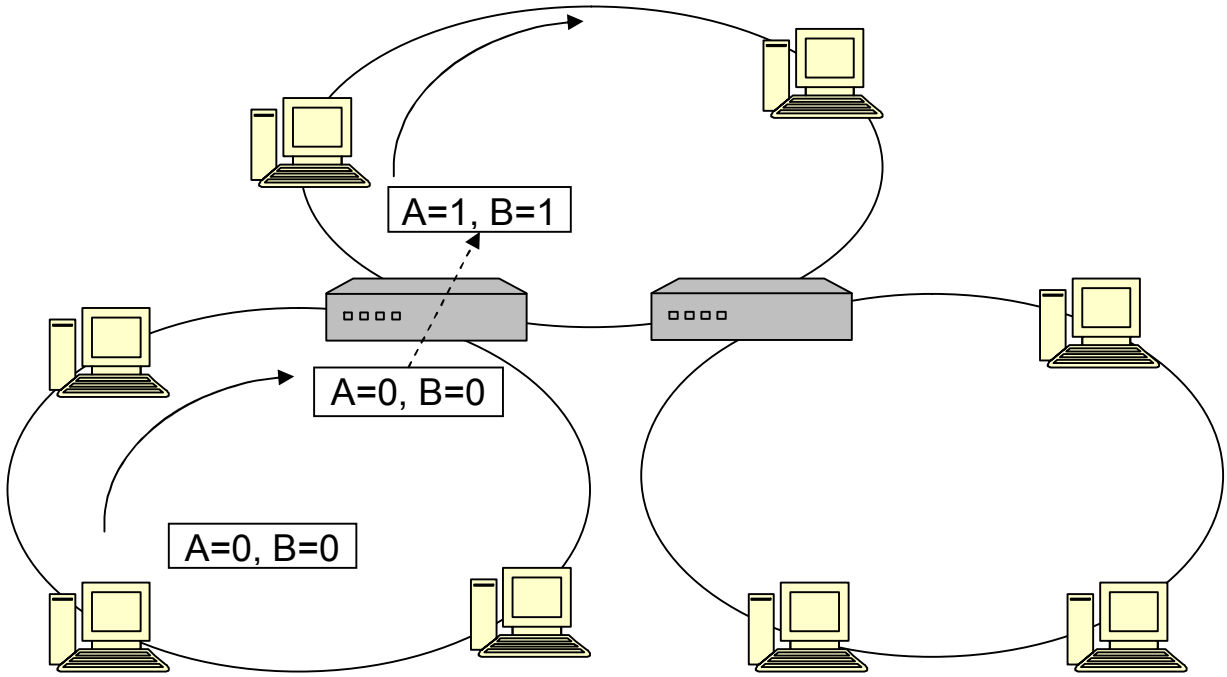
Посылка: На порт моста поступает кадр с данными, в котором есть MAC-адреса отправителя и получателя.

Решение: MAC-адреса отправителя – известны самим отправителям – значит, их следует использовать для построения таблицы.





2.2.7 Алгоритм моста с маршрутизацией от источника

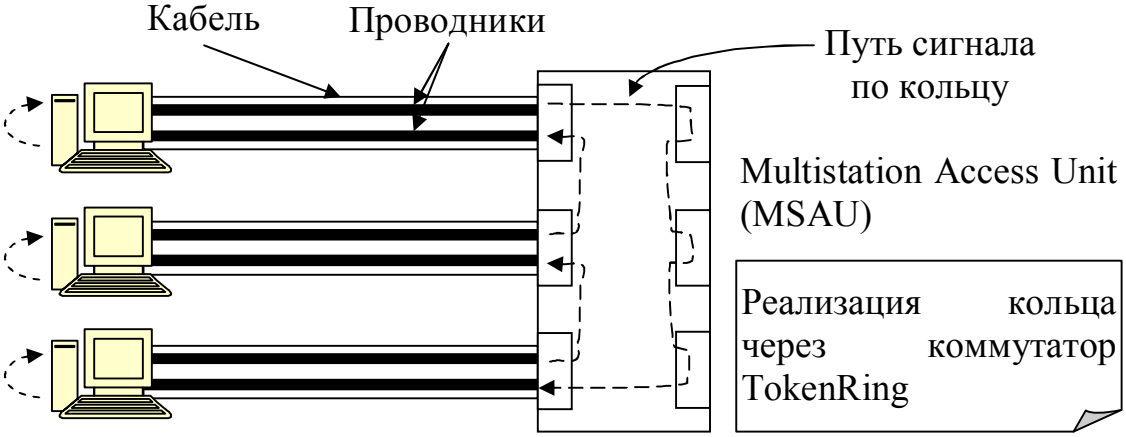


1. Станция-отправитель помещает в кадр всю адресную информацию о промежуточных мостах и кольцах.

2. Мост ищет свой идентификатор и идентификатор следующего кольца.

3. Мост меняет биты «распознан» (A) и «скопирован» (B), затем отправляет в другое кольцо.

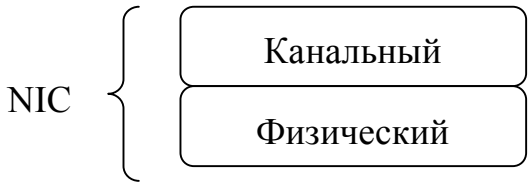
Преимущества	Недостатки
Более рациональные маршруты	Более дорогие сетевые адаптеры, принимающие участие в маршрутизации
Проще и дешевле — не нужно строить таблицы фильтрации	Сеть непрозрачна — кольца имеют номера
Более высокая скорость — не нужно просматривать таблицы фильтрации	Увеличивается трафик за счет широковещательных пакетов



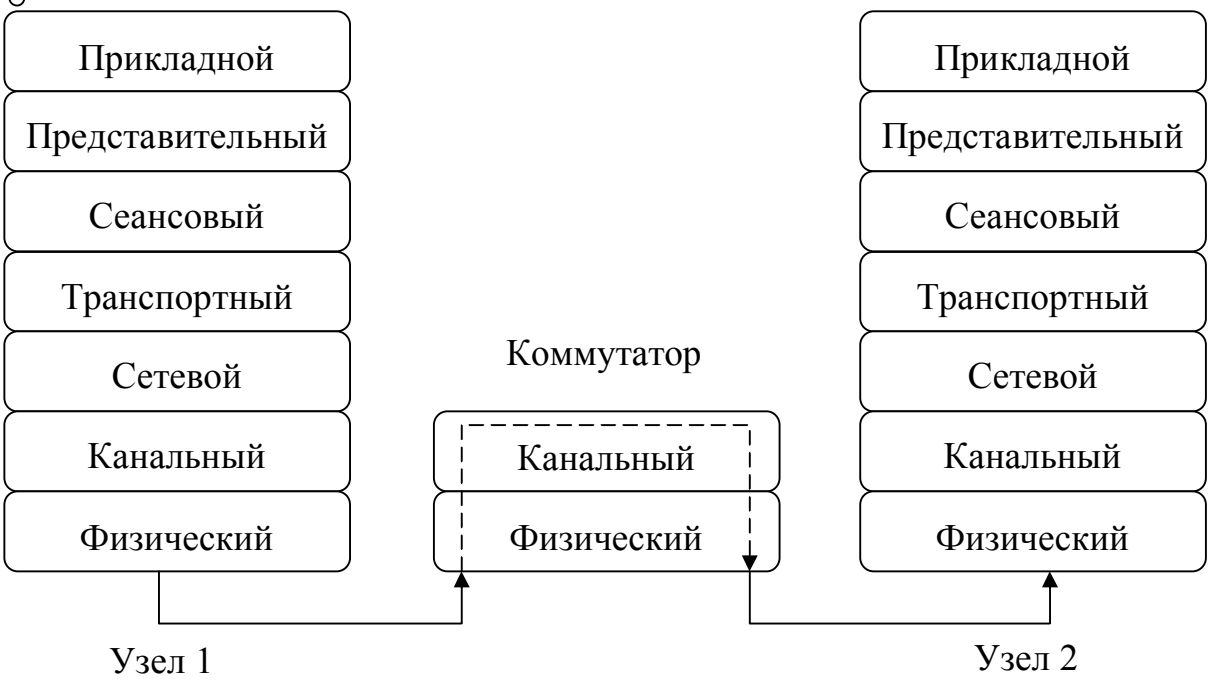
2.2.8 Оборудование канального уровня и его место в стеке OSI

Сетевые адаптеры, Network Interface Card (NIC) вместе с драйвером на компьютере реализуют физический и канальный уровень.

MAC-адрес — атрибут сетевого адаптера.



Место коммутатора в стеке OSI



С помощью хабов и свитчей **нельзя** создавать:

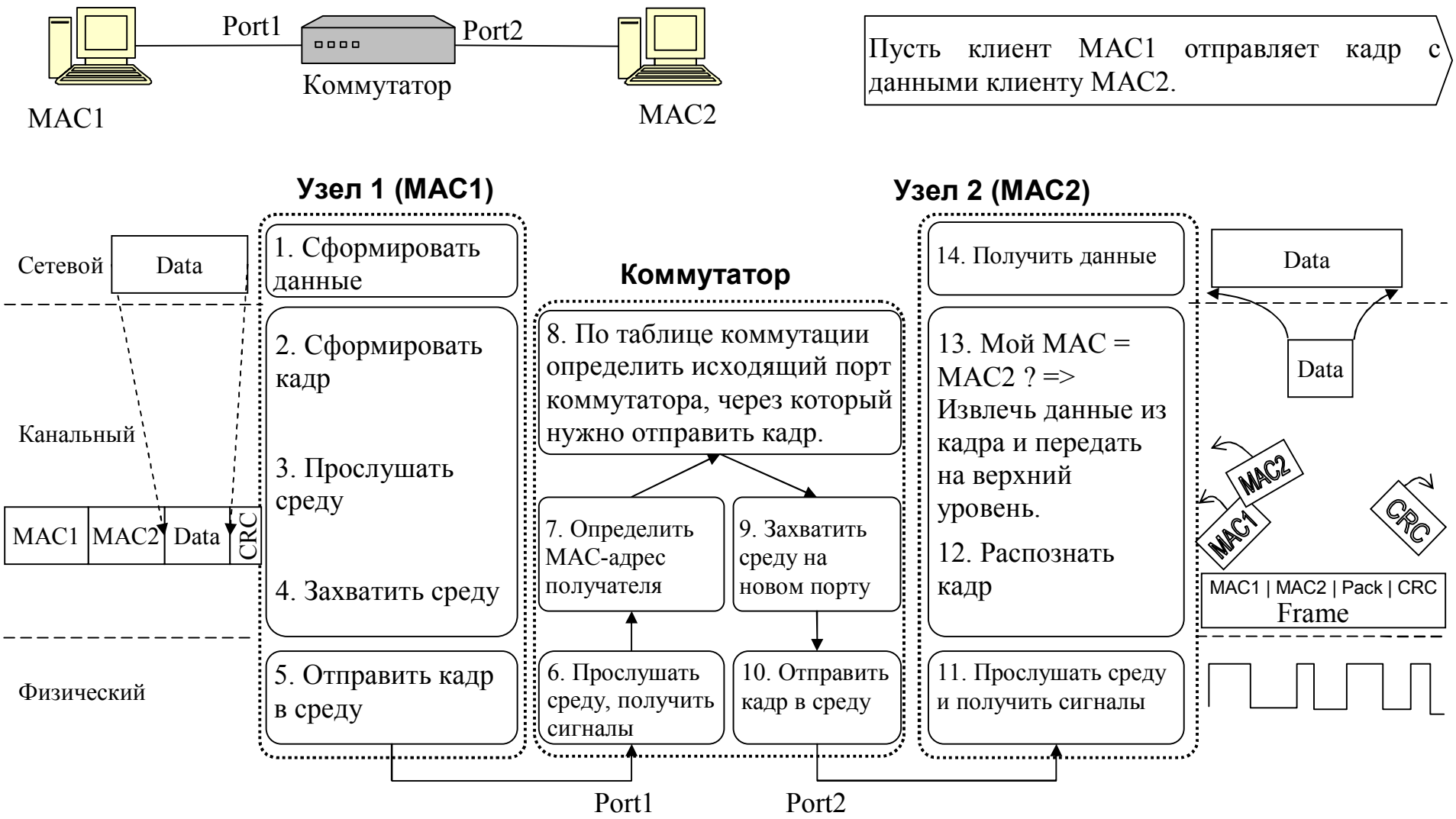
дублирующие линии связи и **циклы**,

иначе в сети произойдет «стоячая коллизия».

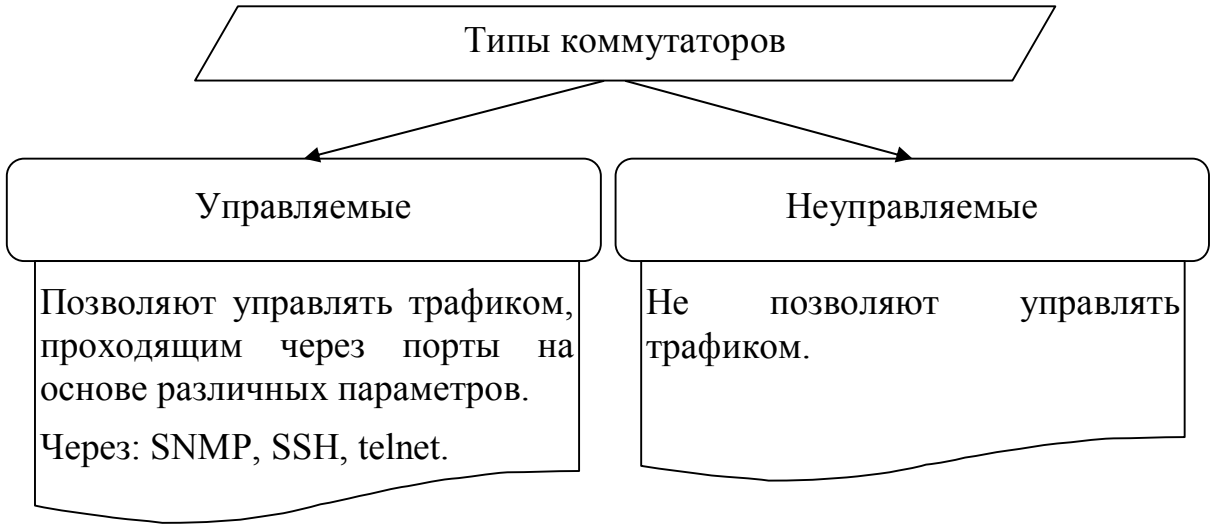
Технология **Spanning Tree Algorithm** (STA) на канальном уровне позволяет автоматически определять древовидную конфигурацию связей и резервировать дублирующие линии с помощью свитчей.

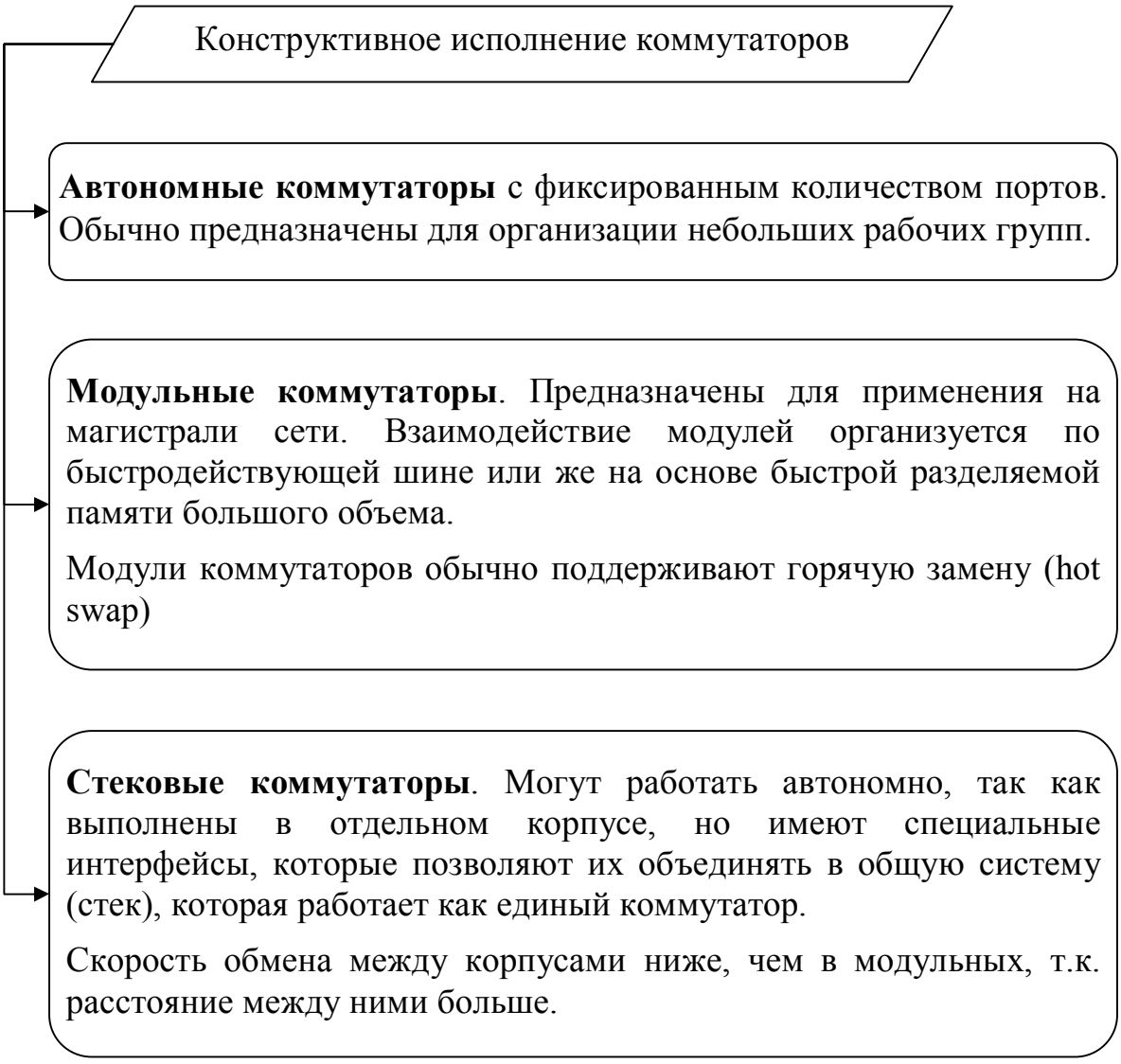
Противоречие. Невозможность создания дублирующих линий связи и необходимость объединения разнородных сетей требует реализации более высокоуровневых функций.

2.2.9 Передача данных на канальном уровне



2.2.10 Типы коммутаторов





2.2.11 Характеристики коммутаторов

Характеристиками коммутатора

Скорость фильтрации (filtering) [кадров/сек] – скорость выполнения этапов:

1. прием кадра в буфер,
2. просмотр адресной таблицы и нахождение порта,
3. уничтожение кадра, т.к. порт назначения = порту источника.

Скорость продвижения (forwarding) [кадров/сек] – скорость выполнения этапов:

1. прием кадра в буфер,
2. просмотр адресной таблицы и нахождение порта,
3. передача кадра в сеть через порт назначения.

Пропускная способность [байт/сек] — количество переданных данных через его порты в единицу времени.

Задержка передачи кадра [мкс] — время, прошедшее с момента прихода первого байта кадра на входной порт коммутатора до момента появления этого байта на выходном порту коммутатора.

- При коммутации «на лету» — 10-40 мкс
- При полной буферизации — 50-200 мкс

Максимальная емкость адресной таблицы [шт.] — максимальное количество MAC-адресов, с которыми может одновременно оперировать коммутатор для одного порта.

Внутренняя буферная память [байт на порт] — для временного хранения кадров данных в тех случаях, когда их невозможно немедленно передать на выходной порт. Буфер предназначен для сглаживания кратковременных пульсаций трафика.

2.2.12 Технология VLAN (IEEE 802.1q)

Часто в сети на канальном уровне возникает широковещательный трафик. Он может быть создан определенными протоколами (ARP, RARP, RIP-IP).

На канальном уровне кадр, предназначенный всем узлам сети, имеет адрес назначения состоящий только из 1 (FF:FF: FF:FF: FF:FF).

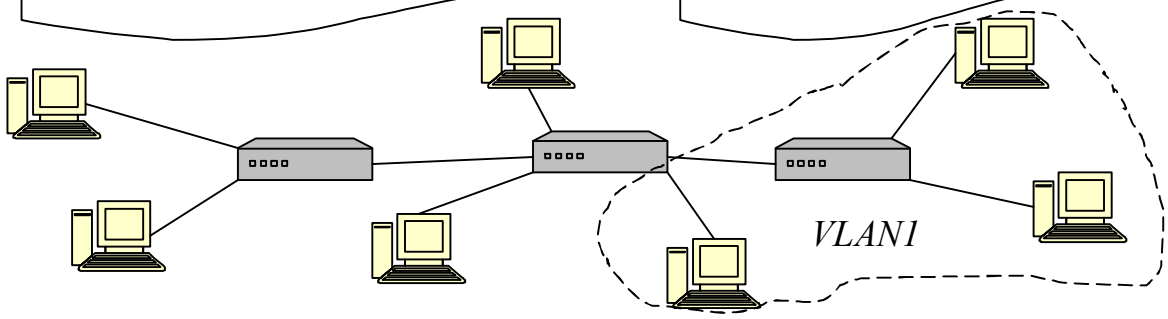
Широковещательный трафик передается всем сегментам сети и снижает ее пропускную способность.

Часто узлы сети физически могут находиться в различных местах, однако, при этом гибкость администрирования (переподключение, управление широковещательным трафиком) снижается.

VLAN (Virtual Local Area Network) — виртуальная локальная компьютерная сеть, представляет собой группу хостов с общим набором требований, которые взаимодействуют так, как если бы они были подключены к широковещательному домену, независимо от их физического местонахождения.

VLAN имеет те же свойства, что и физическая LAN, но позволяет конечным станциям группироваться вместе, даже если они не находятся в одной физической сети.

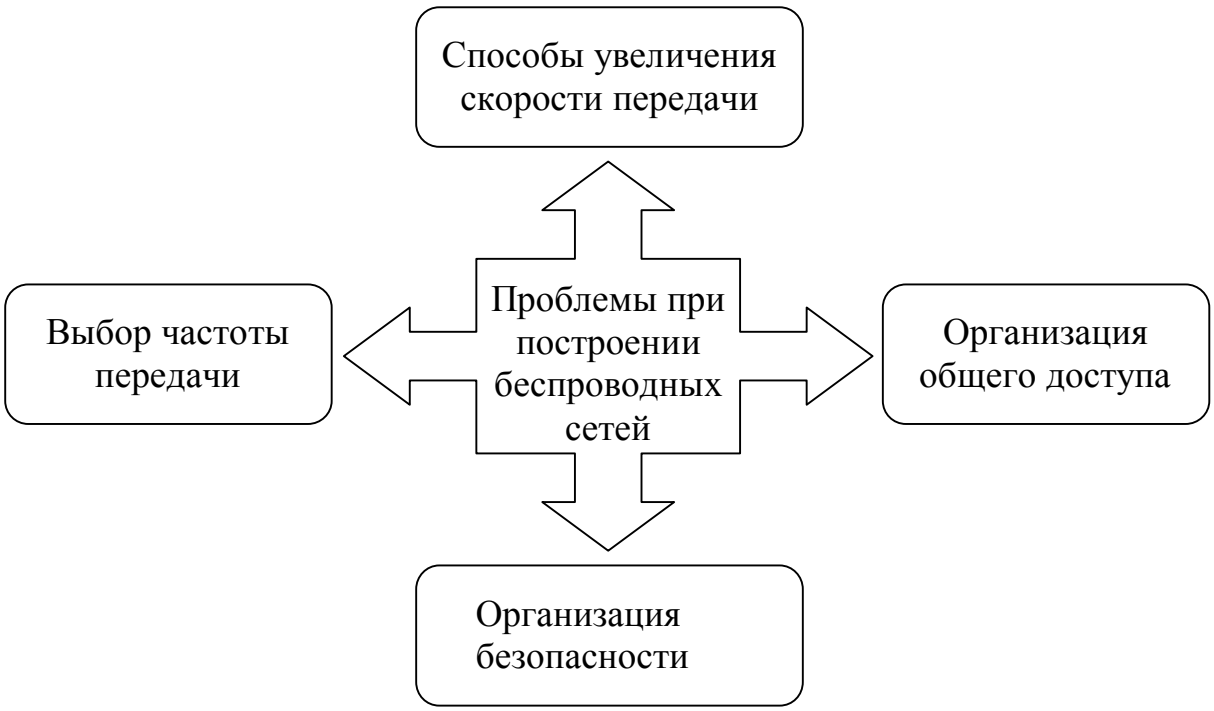
Реорганизация может быть сделана на основе программного обеспечения вместо физического перемещения устройств.



- Облегчается перемещение, добавление и изменение их соединений.
- Гибкость административного контроля
- Уменьшается потребление полосы пропускания.
- Сокращается непроизводственное использование CPU.
- Предотвращаются петли и широковещательные штормы.

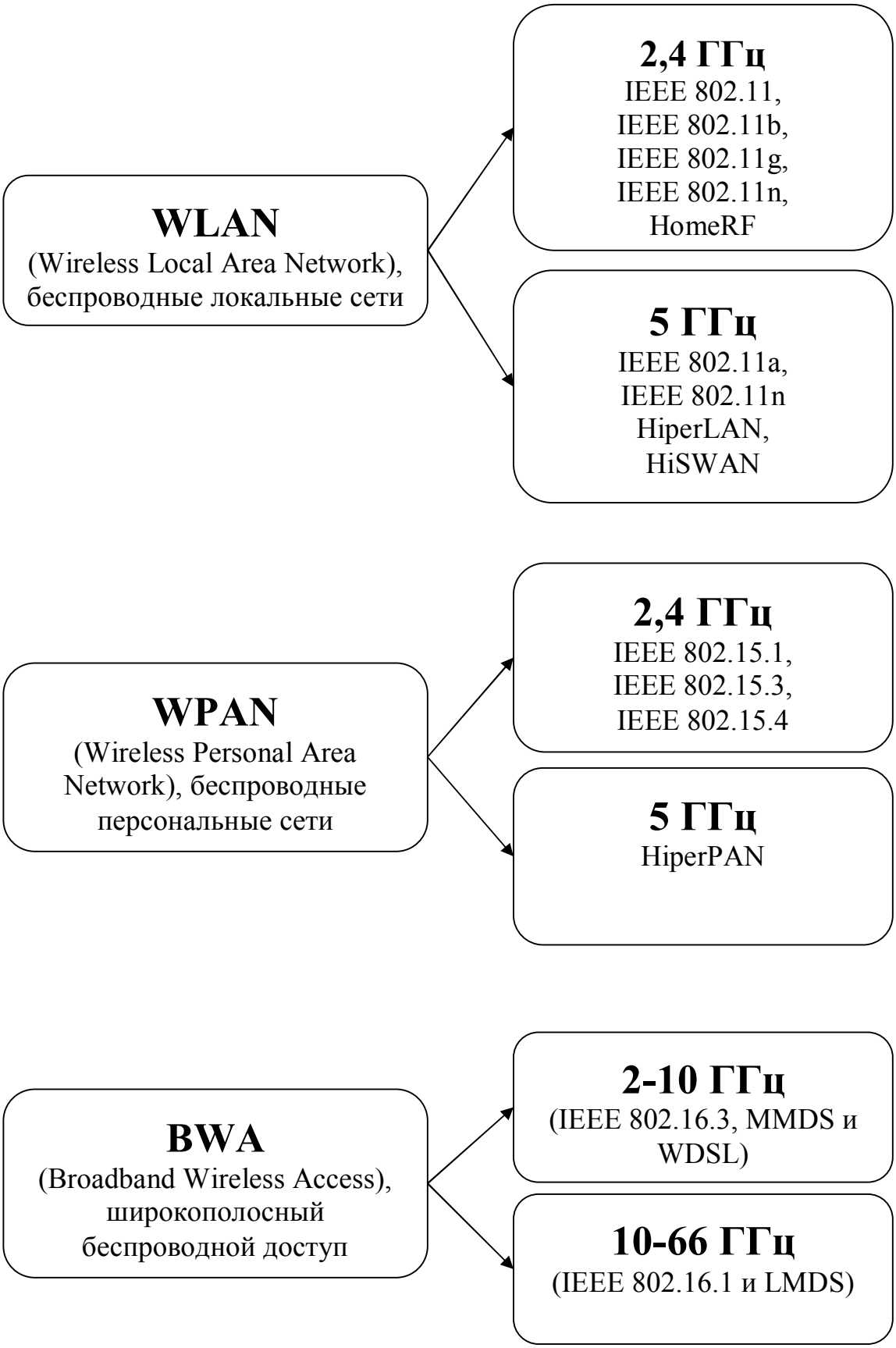
3 Беспроводные сети (IEEE 802.11)

3.1 Общие понятия

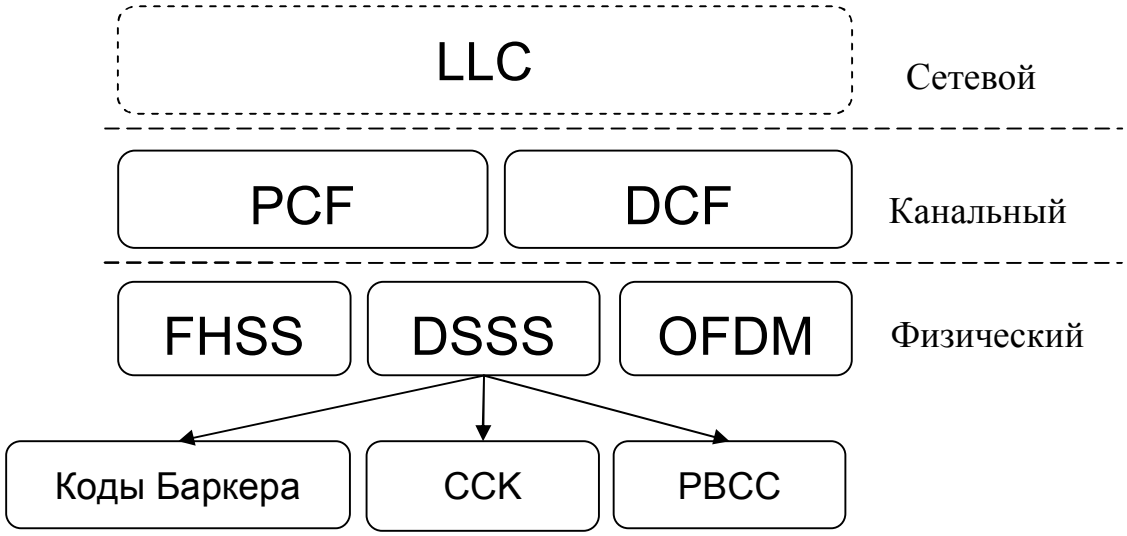


1G	1984г. Мобильные сети первого поколения (аналоговые сети)
2G	1991г. Цифровые сети. Услуга SMS. Разработка стандарта GSM1900. Основан на стандарте CDMA/TDMA. Скорость 9,6кбит/с.
2.5G	Переходное поколение. Основывается на технологиях 2G. Введение пакетной передачи данных GPRS. Скорость 384кбит/с.
3G	Международный стандарт. Скорость до 2048кбит/с, подвижных абонентов 384кбит/с.
3.5G	HSDPA (High-Speed Downlink Packet Access), до 3 Мбит/с
3.75G	HSUPA (High Speed Uplink Packet Access), до 5,8 Мбит/с
4G	Технологии Wi-Fi и WiMax, подразумевающие интеграцию в единую беспроводную сеть широкого спектра устройств.

3.1.1 Типы беспроводных сетей



3.1.2 Технологии физического и канального уровня



Методы основаны на автокорреляционных функциях.

Функция автокорреляции – функция, отражающая степень подобия самой себе через определенный временной интервал.

$$p(f(t), f(t+\tau)) = 1$$

3.2 Кодирование на физическом уровне

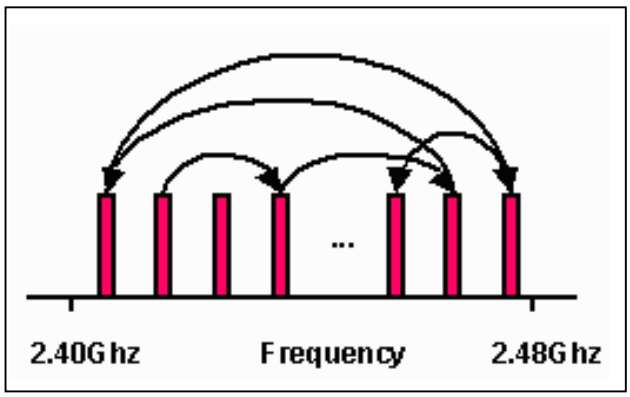
3.2.1 Метод широкополосной модуляции со скачкообразным изменением частоты FHSS

FHSS (Frequency-Hopping Spread Spectrum) – данные передаются последовательно по случайным каналам, выбранным по некоторому шаблону (pattern).

Всего 79 подчастот

Всего 22 шаблона

Смена 1600 раз в секунду



- Используется в Buetooth
- Устройства заранее выбирают шаблон при соединении

3.2.2 Технология уширения спектра DSSS

Технология уширения спектра DSSS (Direct Sequence Spread Spectrum) — работает за счет встраивания в информационный бит последовательности чипов.

Чип — прямоугольный импульс, с длительностью в N раз меньше информационного бита.

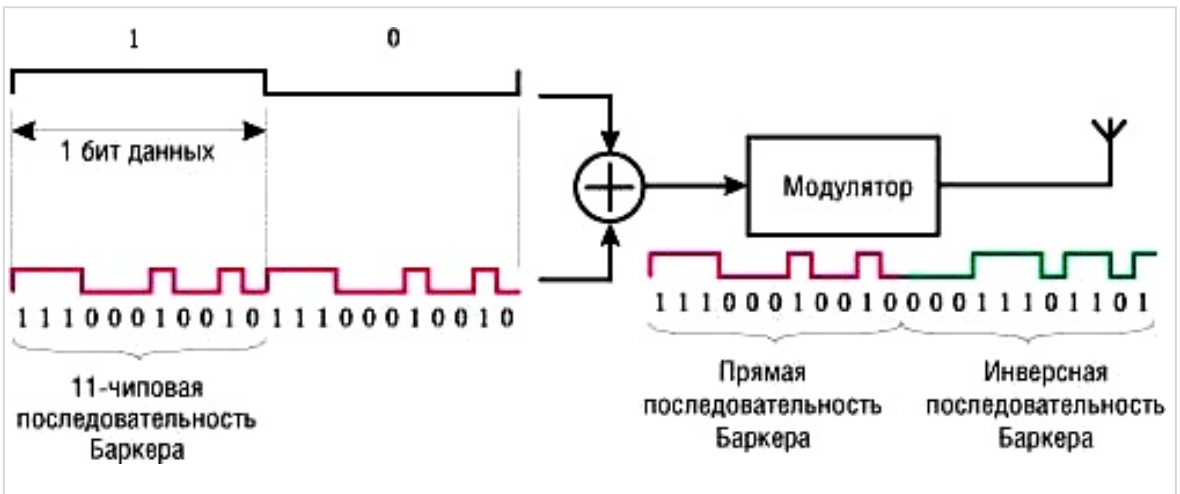
Коды Баркера – чиповые псевдослучайные последовательности, отвечающие требованиям автокорреляции.

Комплементарные коды (Complementary Code Keying, CCK) – кроме уширения спектра позволяют кодировать в одном символе больше, чем один бит

Чип	Коэф.ушир.(дБ)
110	4,77
1110/1101	6,02
11101	6,99
1110010	8,45
11100010010	10,41

Благодаря DSSS спектр в N раз шире,

а амплитуда в N раз меньше.

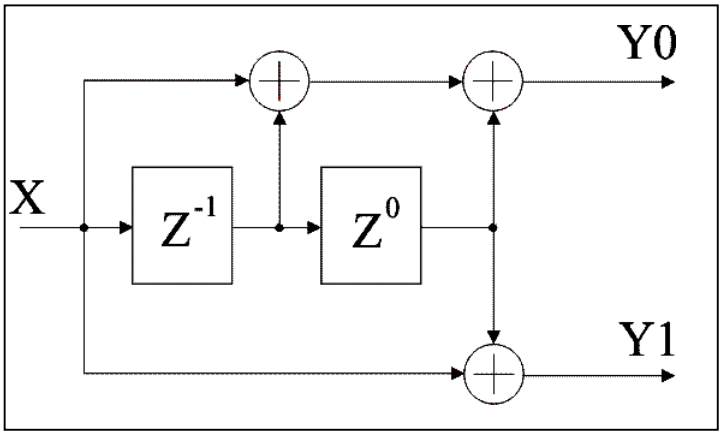


3.2.3 Двоичное пакетное сверточное кодирование PBCC

Двоичное пакетное сверточное кодирование PBCC (Packet Binary Convolutional Coding) преобразовывает входной поток бит так, чтобы одному входному биту соответствовало более одного выходного.

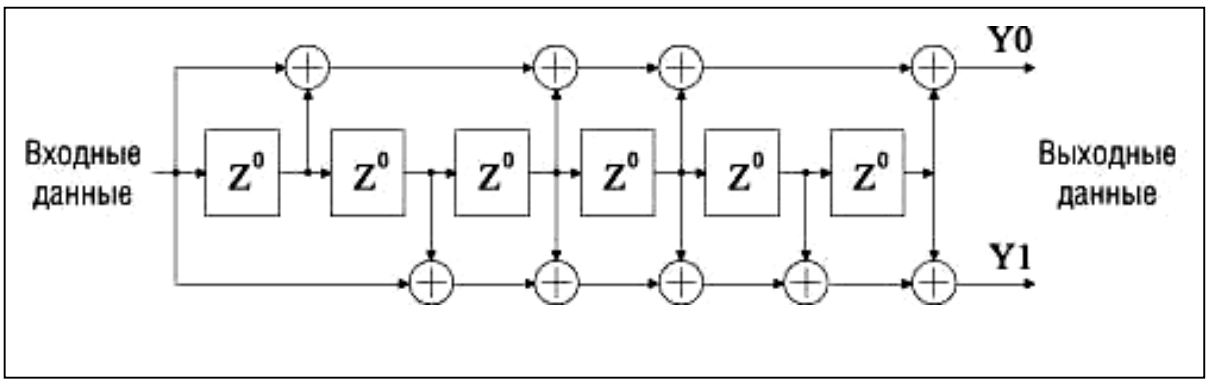
Сверточный кодер состоит из последовательно связанных запоминающих ячеек и логических элементов XOR

Существуют кодеры с коэффициентом сворачивания r = 1/2 (1 бит в 2 бита), 2/3, 3/4 и т.д.



Простейший кодер на три состояния.

Y0 и Y1 зависят от 3 значений: текущего и двух предыдущих



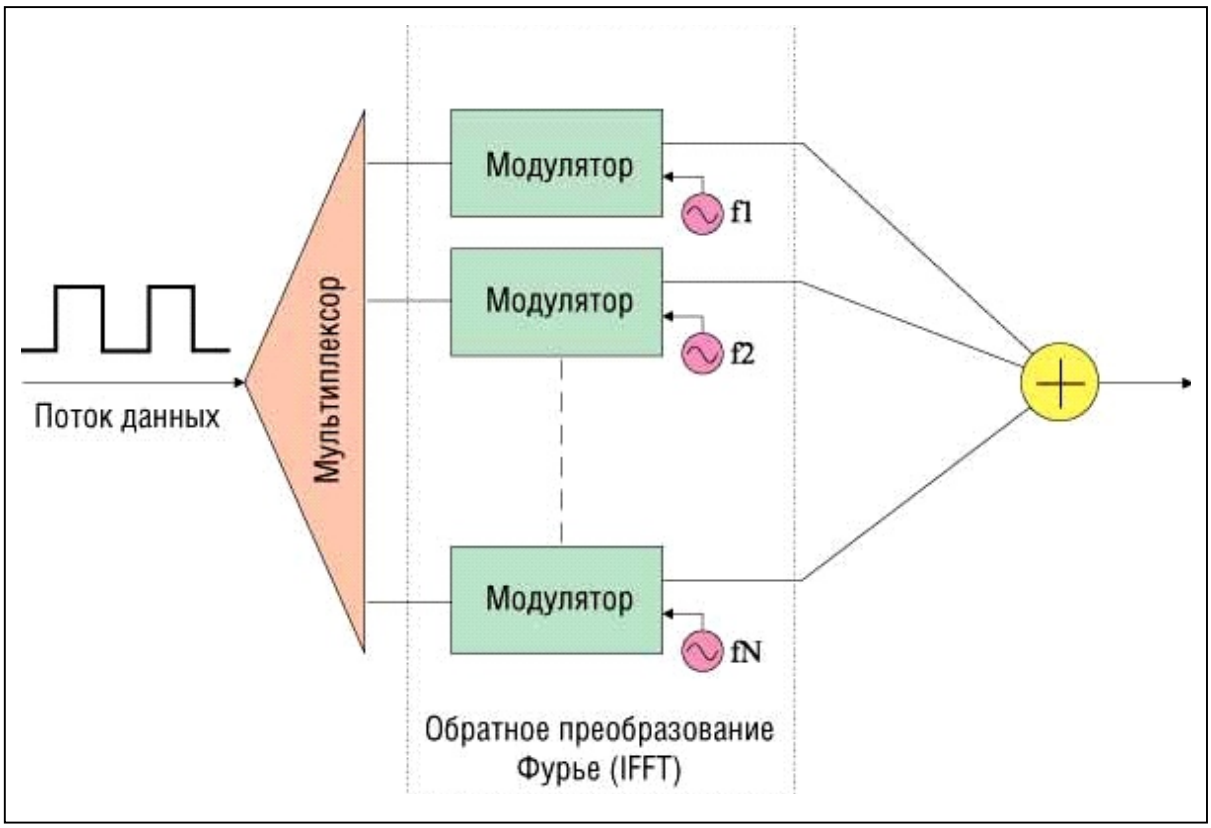
Кодер на семь состояний (K = 7) со скоростью r = 1/2

3.2.4 Ортогональное частотное разделение каналов с мультиплексированием OFDM

OFDM (Orthogonal Frequency Division Multiplexing) – использует большое количество близко расположенных ортогональных поднесущих.

Высокая скорость передачи достигается за счет одновременной передачи данных нескольким подканалам

В передатчиках сигнал мультиплексируется, затем к N-каналам применяется обратное преобразование Фурье (IFFT)



OFDM – позволяет бороться с интерференцией волн и искажением сигнал из-за нее.

Применяется в:

- ADSL и VDSL
- DVB-C2
- Wi-Fi, WiMAX

3.3 Координация на канальном уровне

Координация основана на методе коллективного доступа с обнаружением несущей и механизмом избежания коллизий **CSMA/CA** (Carrier Sense Multiple Access/Collision Avoidance)

Типы координаций на канальном уровне в беспроводных сетях

Распределенная координация (Distributed Coordination Function, DCF)

Централизованная координация (Point Coordination function, PCF)

Избежание коллизий происходит за счет использования обязательных промежутков ожидания.

DIFS (Distributed InterFrame Space**)** – обязательный промежуток при распределенной координации

PIFS (Point InterFrame Space**)** – обязательный промежуток при централизованной координации

SIFS (Simple InterFrame Space**)** – простейший промежуток

$$SIFS < PIFS < DIFS$$

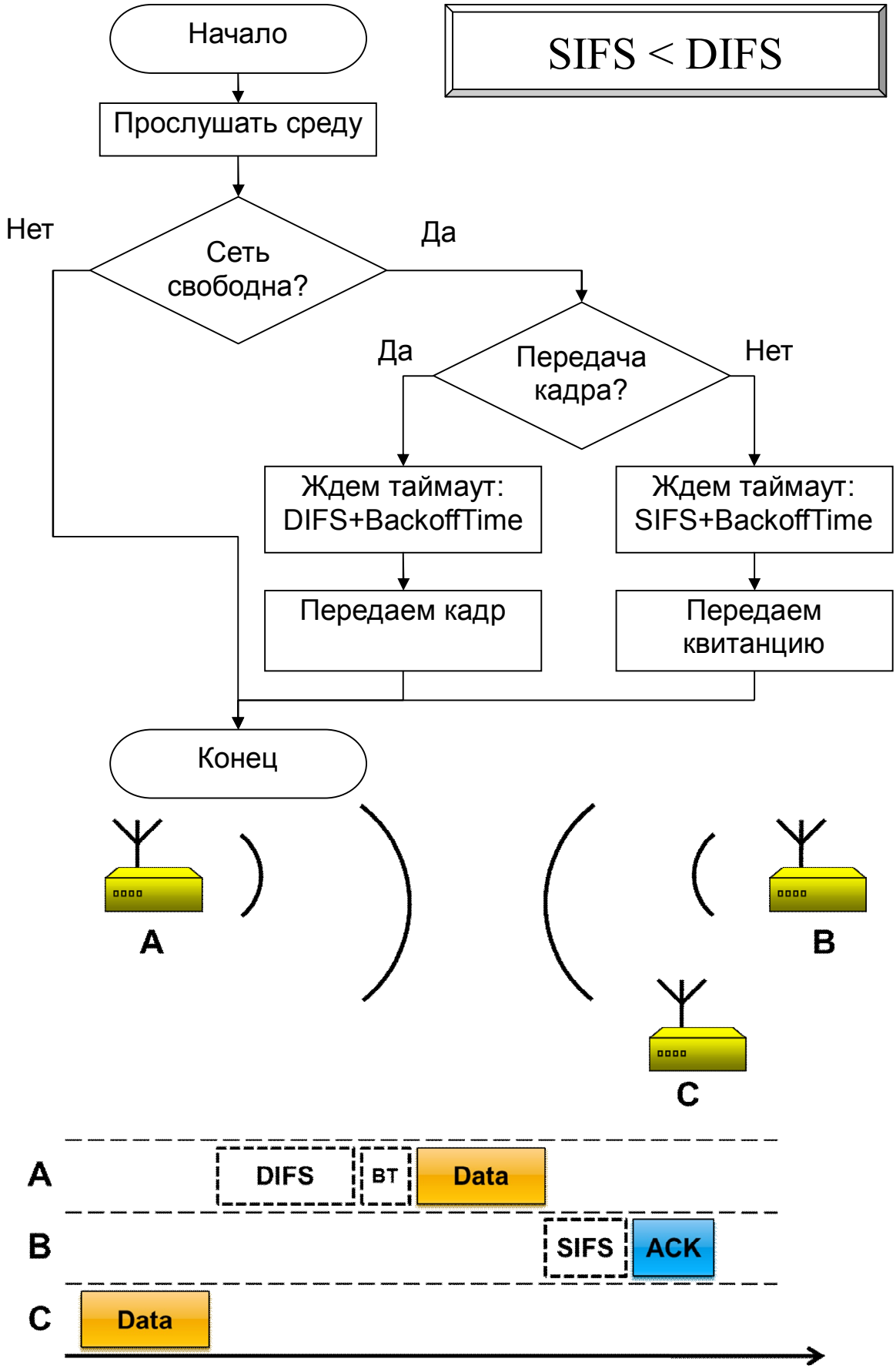
BackoffTime – изменяемый интервал.
TimeSlot – элементарный промежуток
CW (Contention Window) – размер окна от 31 до 1023 тайм-слотов.

$$\text{Backoff time} = \text{Random}(CW_{min}, CW_{max})*\text{SlotTime}$$

Если эфир был **занят** → то в следующий раз **уменьшить** окно $CW_{i+1} < CW_i$

Если кадр **неудачно принят** → то в следующий раз **увеличить** окно $CW_{i+1} > CW_i$

3.3.1 Распределенная координация DCF



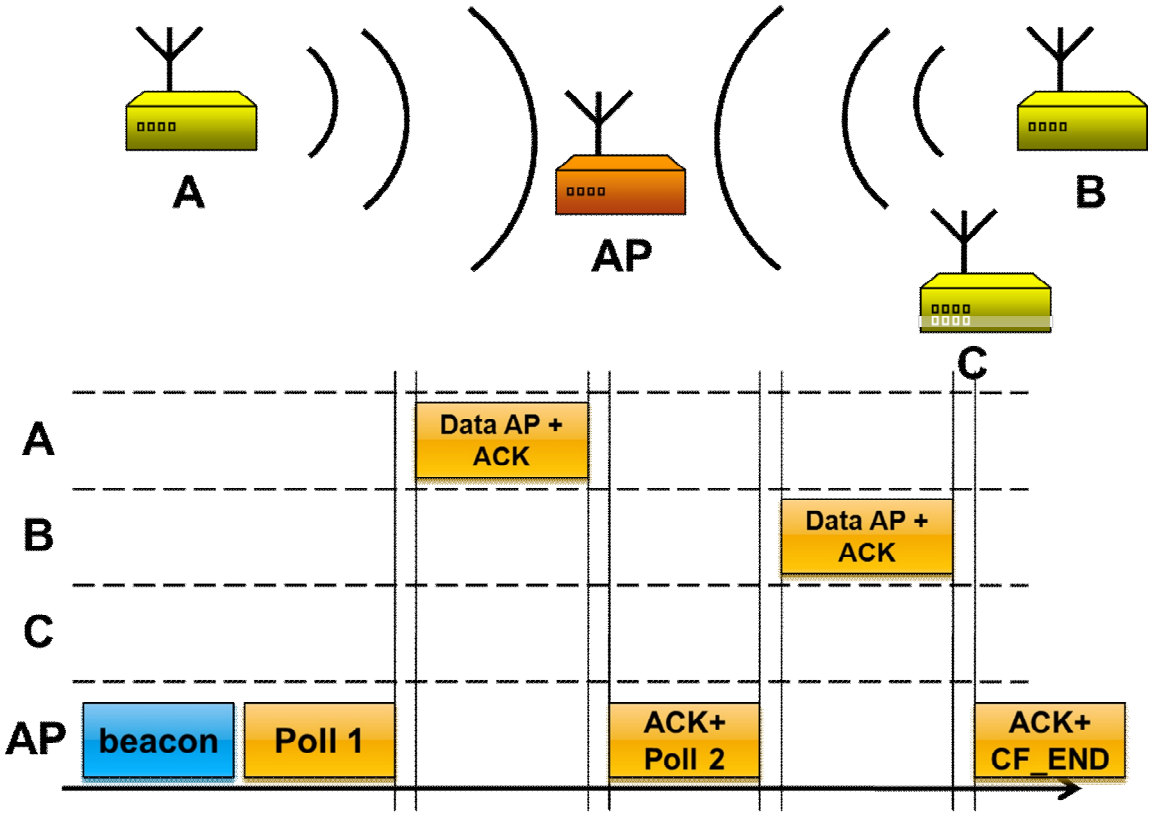
3.3.2 Централизованная координация PCF

Режим PCF реализуется точкой доступа (Access Point, AP).

Управляет на основе алгоритма опроса или исходя из приоритетов узлов сети.

Координатор ожидает промежуток времени PIFS.

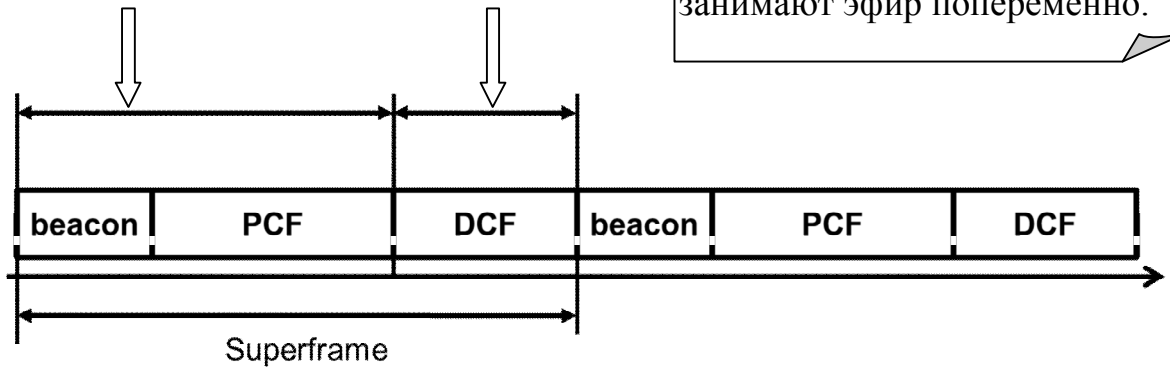
Так как **SIFS < PIFS < DIFS**, то координатор захватывает среду раньше, чем кто либо; но после любого, кто отправляет квитанцию.



CFP (Contention-Free Period) – бесконкурентный доступ

CP (Contention Period) – конкурентный доступ

Для возможности совмещения координации DCF и PCF, фреймы обоих координаций занимают эфир попеременно.



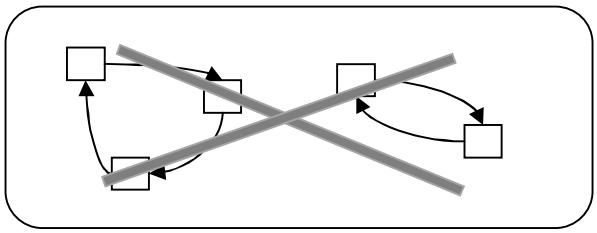
3.3.3 Характеристики стандартов беспроводных сетей

Таблица 1 – Сводная таблица характеристик стандартов беспроводных сетей

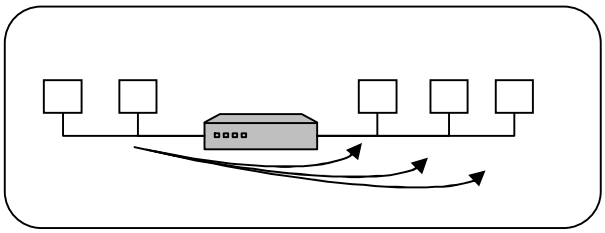
		Bluetooth	**Wi-Fi**		**Wi-MAX**			**GSM**
	802.11	*802.15.x*	*802.11b/g*	*802.11a/n*	*802.16*	*802.16a/b*	*802.16e*	*GPRS*
Назначение		Связь с мобильными устройствами	Организация беспроводных сетей в пределах здания		Организация сетей в масштабах города			Передача данных на большие расстояния
Частоты (ГГц)	2.4-24.8	**2.4 – 2.48**	**2,4 – 2,4835**	**5,15 – 5,350**	**10-66**	**2-11**	**2-6**	**1,8-1,9**
Радиус действия (м)		**10 (100*)**	**100**		**2-4км**	**4-6 (15-20) км**	**4-6 км**	**1000**
Скорость передачи (Мбит/с)	**1-2**	**~1 (2*)**	**5(11)/54**	**>100**	**32-134**	**1-75**	**<15**	**64-128**
Тип кодирования	**DSSS**	**FHSS**	**Баркера, CCK / CCK (PBCC)**	**CCK**	**коды Рида-Соломона, CCK**			**CDMA/ TDMA**
Модуляция	**BPSK**	**GFSK**	**BPSK, QPSK / OFDM**	**OFDM, QAM**	**QPSK**	**OFDM 256, OFDMA QAM16, QAM64**		**GMSK**
Количество устройств	*	**7**	**30**		*			**1000**
Шифрование		**128 IDEA, динамические ключи**	**WEP 64 и 104, TKIP, MIC**		**DES/3DES**			**A5/1**

4 Организация взаимодействия между сетями

Проблемы протоколов канального уровня:



Невозможно организовать циклы и дублирующие каналы



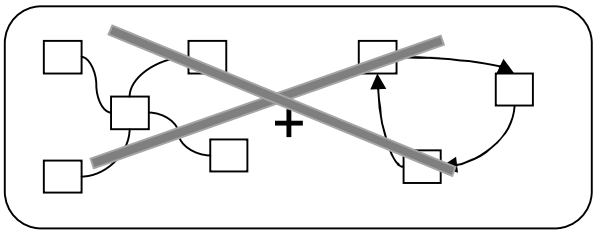
Логические сегменты подвержены широковещательным атакам

MAC1 MAC3

…

MAC2 MAC100500

Жесткая одноуровневая адресация по MAC адресам



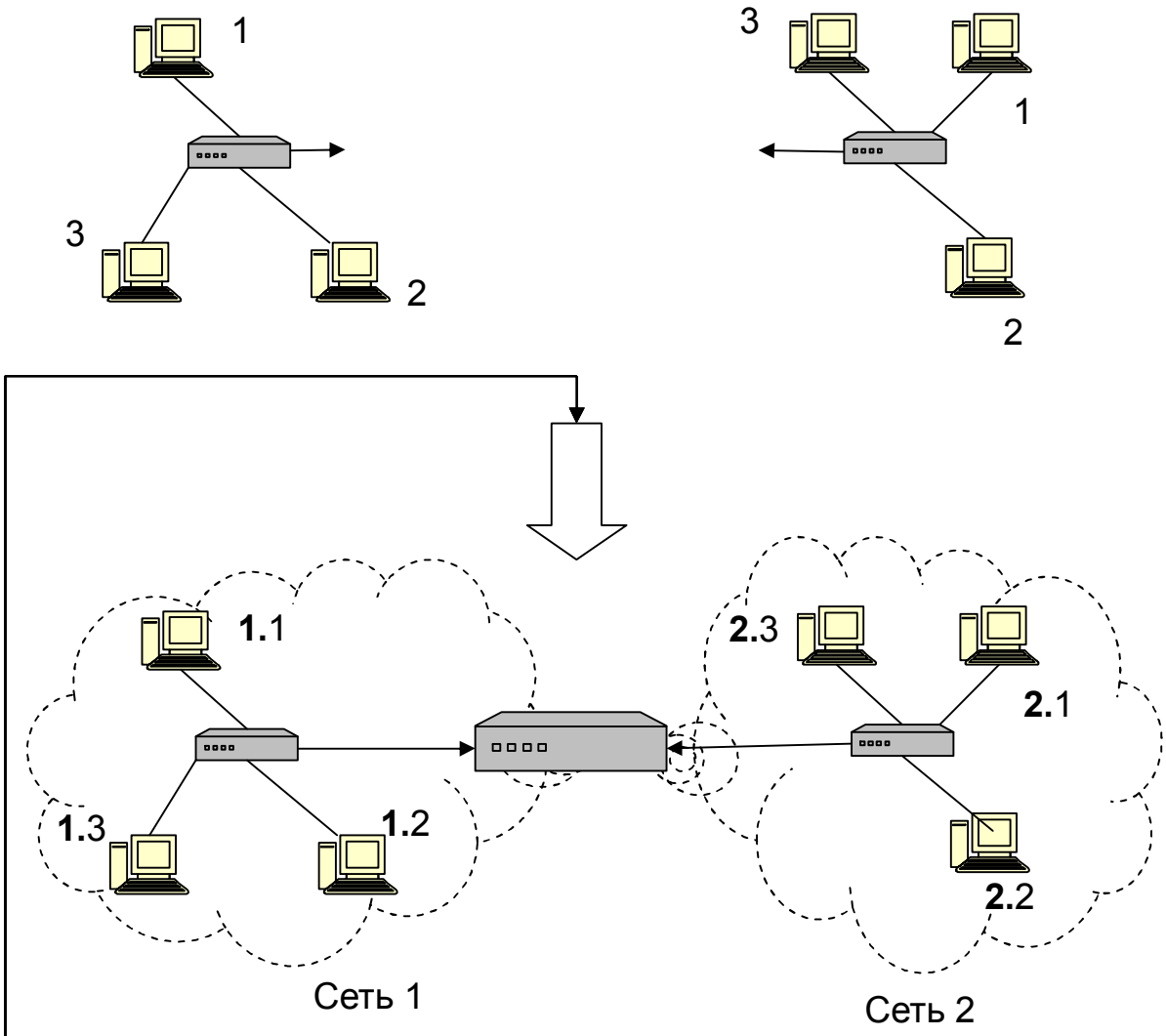
Невозможно объединить сети разных топологий, отсутствует масштабируемость

Необходимо реализовать высокоуровневые функции.

Ввести специальную систему адресации

Обеспечить взаимосвязь с канальным уровнем

Обеспечить продвижение единиц данных сквозь сети



- Разработать систему адресации
- Разработать протокол взаимодействия узлов
- Разработать механизм продвижения данных сквозь сети
- Реализовать протоколы в специальных коммуникационных устройствах, соединяющих сети.

4.1 Сетевой уровень модели OSI

На сетевом уровне объединяют несколько сетей в составную сеть или интерсеть (internetwork, internet)

- Передача пакетов между конечными узлами в составных сетях
- Выбор наилучшего маршрута по определенному критерию
- Согласование протоколов канального уровня в отдельных подсетях.

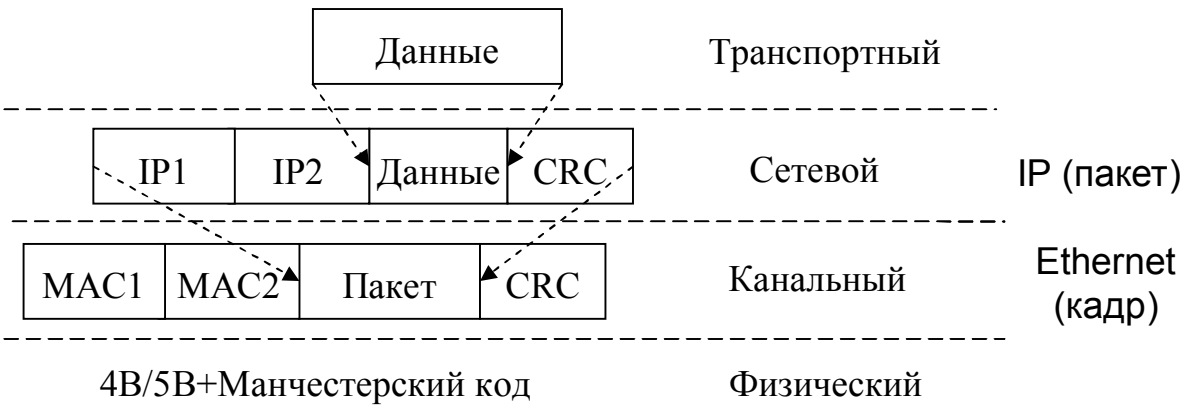
Прикладной
Представительный
Сеансовый
Транспортный
Сетевой
Канальный
Физический

Единица данных: **пакет (packet)**

Тип адресов: **IP, IPX и др.**

Протоколы: **IP, IPX, RIP, NLSP, ES-ES, IS-IS**

Передача данных между уровнями:



Адрес *сетевого* уровня называют **логическим** адресом.

Адрес *канального* уровня называют **физическим** адресом.

4.1.1 Протокол IP

- Протокол IP (internet protocol) – протокол сетевого уровня
- Стек TCP/IP
- Основная задача передавать пакеты между сетями
- Протокол передает пакеты как независимую единицу без связи с другими пакетами

В протоколе IP используются IP-адреса.

IP-адрес это 4-х (в IPv4):

IPv4: **192.168.1.1**

или 16-и байтное число в IPv6: **2001:0db8:11a3:09d7:1f34:8a2e:07a0:765d**

Формат заголовка IP-пакета:

Протокол IP используется для негарантированной доставки данных.

Однако при этом выявляются ошибки в пакетах, вычисляется контрольная сумма, поврежденные и «плохие» пакеты не передаются дальше.

4.1.2 IP-адрес и классы сетей

IP-адрес характеризует одно сетевое соединение

С сетевым адаптером может быть связано несколько IP-адресов

192.168.15.120

Часть адреса – номер сети

Другая часть – номер узла в сети

~

По аналогии с почтовым адресом:

ул. Мира, 120

Так как в качестве номера сети могут быть первый байт, первые 2 или 3, то для определенности, ввели классы сетей.

Деление на классы происходит на основе первых бит первого октета:

Противоречие. С помощью классов невозможно гибко структурировать сети. Поэтому для более гибкого определения номера сети введена *маска сети* (network mask).

4.1.3 Маска сети

Маска сети — это число, используемое в паре с IP-адресом, ее двоичная запись содержит непрерывную последовательность единиц в тех разрядах, которые должны в IP-адресе интерпретироваться как номер сети.

Маска сети скрывает то, что касается номера узла и оставляет только разряды номера сети.

! С помощью маски — сети гибко структурируются.

При использовании маски сети, понятие класс сети теряет смысл.

IP: 195.112.228.95
Mask: 255.255.255.0

В десятичном виде

ИЛИ

195.112.228.95/24

В виде количества разрядов через слешь (/)

Процесс наложение маски на IP-адрес:

```
   192.168.  1.7 = 11000000.10101000.00000001.00000111
AND
   255.255.255.0 = 11111111.11111111.11111111.00000000
   -----------------------------------------------------
   192.168.  0.0 = 11000000.10101000.00000001.00000000
```

В результате наложения маски сети на IP-адрес получается — адрес сети.

⇩

! Поэтому в качестве номера узла нельзя использовать одни нули.

~ По аналогии с почтовым адресом. В качестве адреса проживания не используется информация лишь об улице

ул. Мира, ???

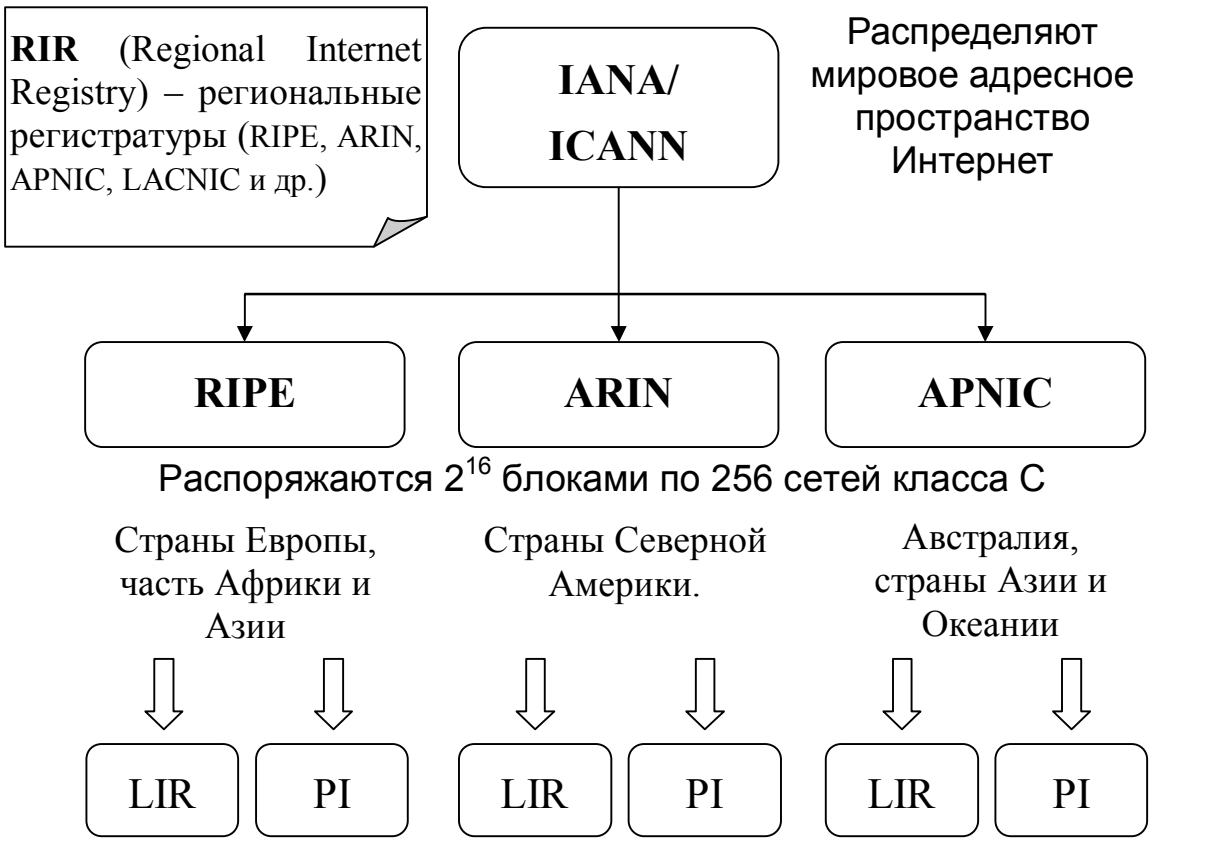
⊘ 198.206.12.0/24 = **11000110. 11001110.00001100.**00000000
✓ 175.206.12.0/16 = **10101111. 11001110.**00001100.00000000
⊘ 195.16.39.224/28 = **11000011.00010000.00100111. 1110**0000
✓ 195.16.39.225/28 = **11000011.00010000.00100111. 1110**0001

4.1.4 Распределение IP-адресов в мире

Internet Assigned Numbers Authority (IANA) – отвечает за распределение всех зарезервированных имён и номеров, которые используются в протоколах, определённых в RFC.

База IANA: http://www.iana.org/assignments/ipv4-address-space/

Internet Corporation for Assigned Names and Numbers (ICANN) – международная некоммерческая организация, регулирующая вопросы, связанные с доменными именами, IP-адресами и прочими аспектами функционирования Интернета.



RIPE – Европейский региональный регистратор

ARIN – Американский регистратор интернет-номеров

APNIC – Азиатско-Тихоокеанский сетевой информационный центр

LACNIC – Латиноамериканский и карибский информационный центр

LIR (Local Internet Registry) – локальный регистратор, занимающийся распределением адресного пространства пользователям сетей (сервис-провайдерам и их абонентам).

PI (Provider Independent) – провайдеро-независимые IP адреса, получаемые из блоков RIR.

4.1.5 Локальное распределение IP-адресов

Получение IP-адреса в адресном пространстве Интернет затратно и требует обращения в сторонние организации.

Поэтому выделены специальные блоки адресов доступные для локального использования без специальной регистрации.

в классе A — это сеть 10.0.0.0

в классе B — это диапазон из 16 сетей: 172.16.0.0 – 172.31.0.0

в классе C — это диапазон из 255 сетей: 192.168.0.0 – 192.168.255.0

Эти адреса называются **приватными** (private) или **серыми**.

Все остальные **публичными** (public) или **белыми**.

Серые адреса не требуют регистрации, поэтому их используют в локальных сетях.

Противоречие. Из-за наличия серых адресов следует, что IP-адрес не является уникальным идентификатором узла в сети? Тогда как связать два узла с серыми адресами через Интернет? (см. стр.103, п.4.2.9)

4.1.6 Соглашения об IP-адресах

Все нули (0.0.0.0) – обозначает адрес того узла, который сгенерировал этот пакет (используется в ICMP, DHCP)

Номер сети из 0 (0.0.0.126/24) – узел назначения принадлежит той же сети, что и отправитель.

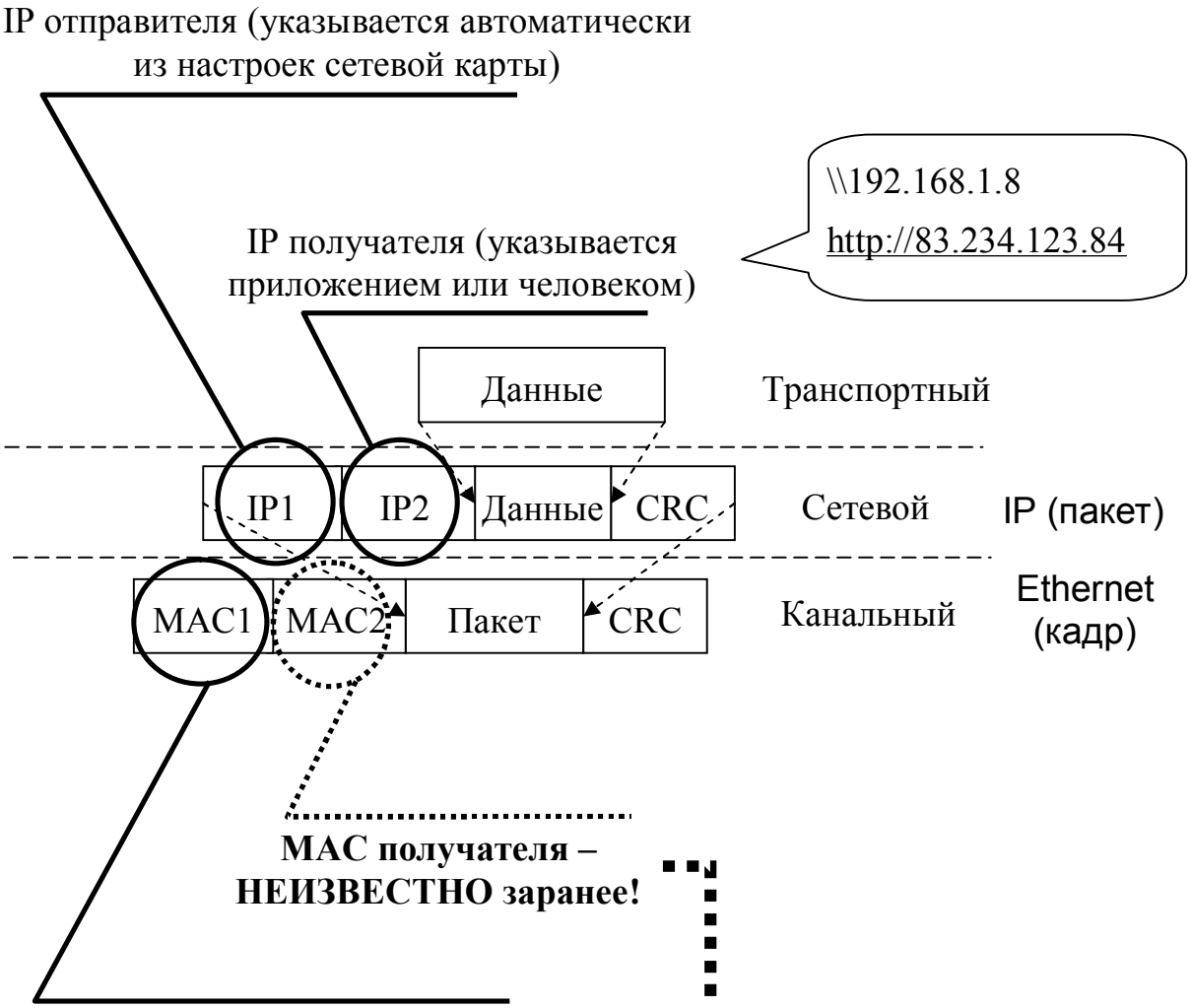
Все 1 (255.255.255.255) – пакет предназначен всем узлам сети отправителя (limited broadcast, ограниченный широковещательный адрес).

Номер узла из 1 (129.127.255.255/16) – пакет рассылается всем узлам сети с заданным номером сети (broadcast, широковещательные адрес).

Адрес 127.0.0.1 – пакет предназначен самому себе (loopback). С нижнего уровня возвращается наверх. Других адресов в сети 127.0.0.0/8 – нет.

4.1.7 Разрешение адресов

В ходе рассмотрения схемы передачи данных между уровнями обнаруживается **противоречие**:



Необходимо предусмотреть процедуру определения MAC адреса получателя, зная его IP адрес. То есть

IP2 → MAC2

~

По аналогии «в организации по должности найти человека».

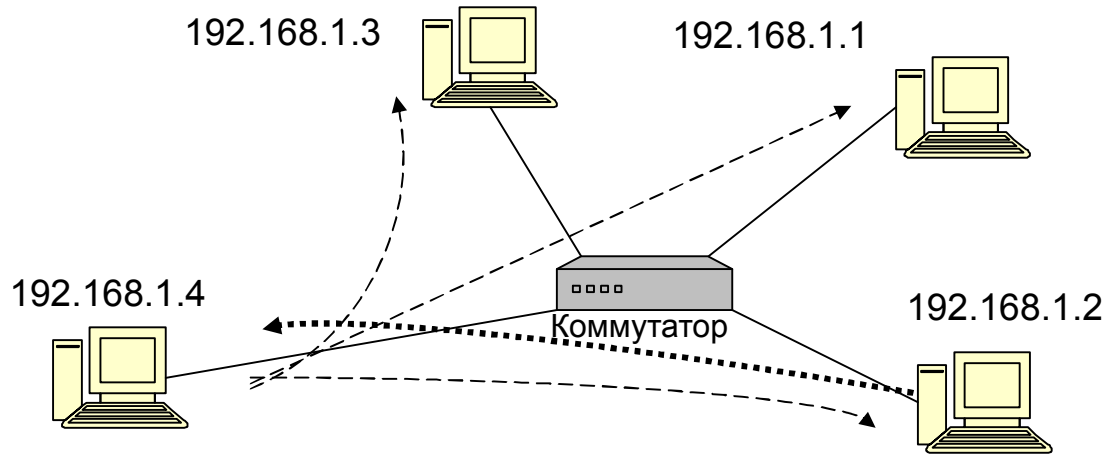
Должность (Директор)

Человек (Петя)

4.1.8 Протокол ARP

Протокол разрешения адресов (Address Resolution Protocol, ARP) – предназначенный для определения адреса канального уровня (MAC) по известному адресу сетевого уровня (IP).

Принцип работы ARP:



1 Узел широковещательно посылает ARP-запрос: «какой MAC с таким IP?»

ARP-запрос:

Свой MAC1	FF: FF:FF: FF: FF:FF
IP отправителя (192.168.1.4). IP получателя (192.168.1.2).	

2 Узел, у которого совпал IP, возвращает ARP-ответ со своим MAC адресом

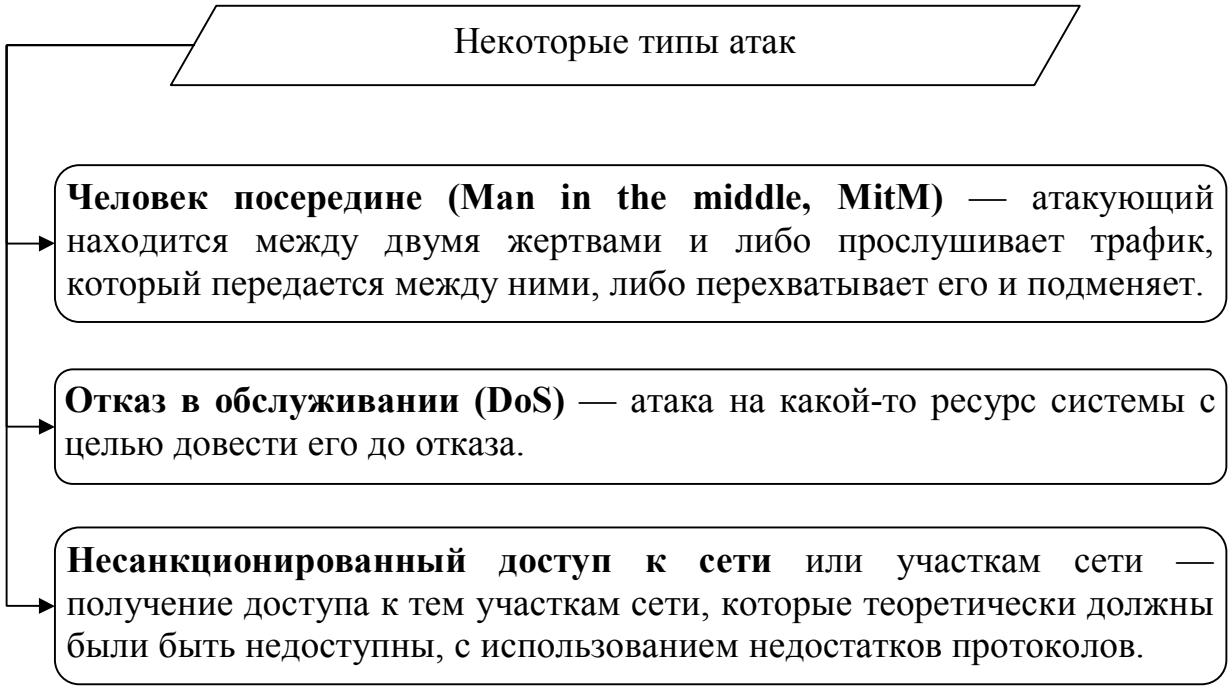
ARP-ответ:

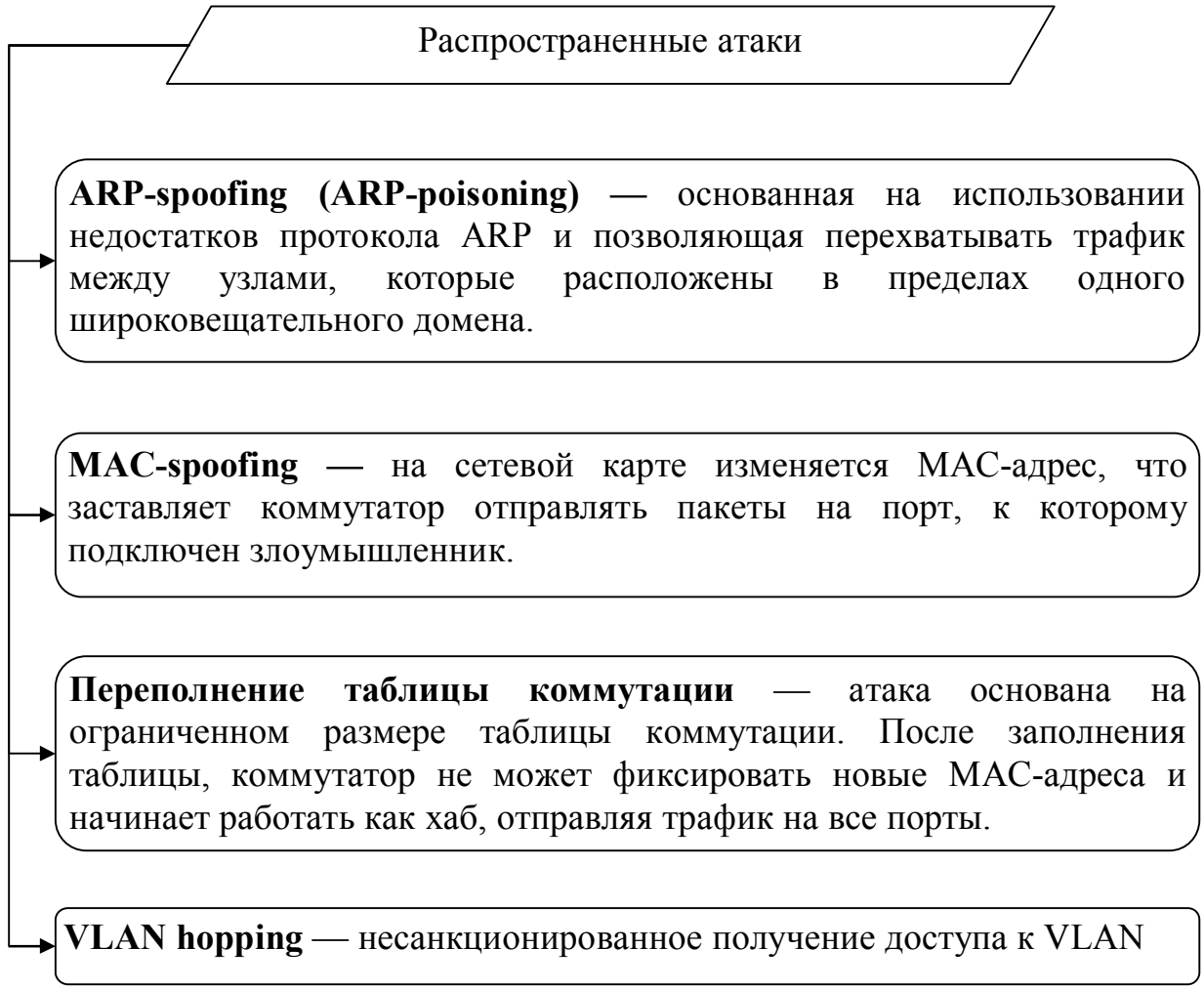
Свой MAC2	Его MAC1
IP отправителя (192.168.1.2). IP получателя (192.168.1.4).	

3 Получив ответ, узел временно сохраняет его в ARP-таблице, чтобы не «спрашивать» постоянно.

IP	**MAC**
192.168.1.2	23:BA:3F:12:11:C2
192.168.1.1	4C:EA:E5:21:B8:2D

4.1.9 Безопасность на канальном уровне

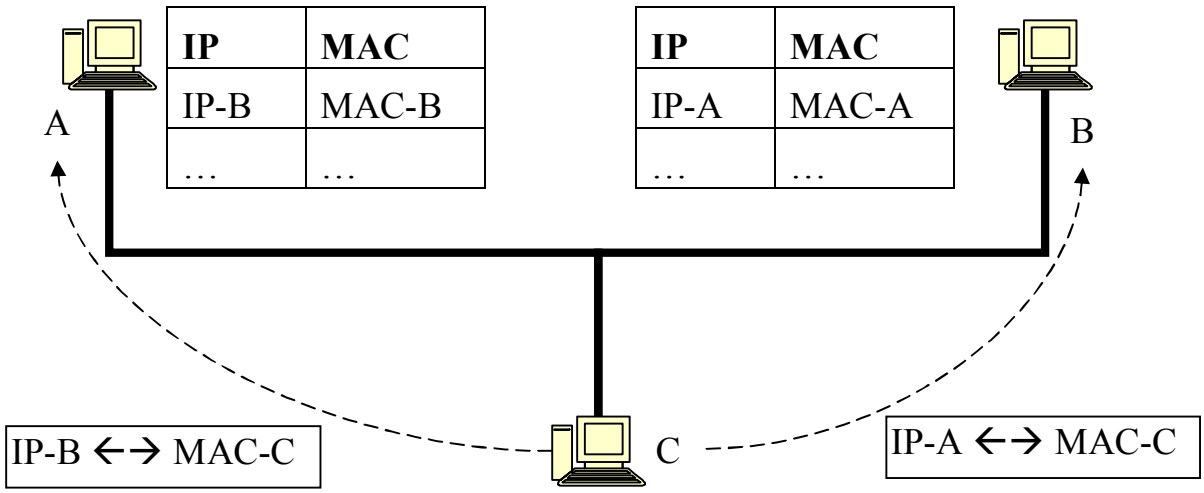




4.1.10 ARP-спуфинг (ARP-spoofing)

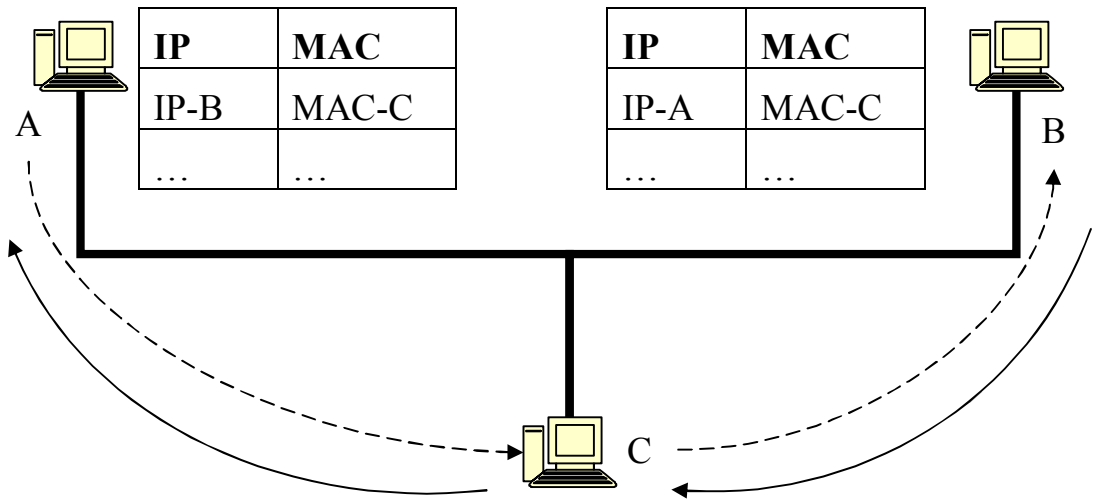
Атакуя A и B, компьютер C отправляет без запроса ARP-ответы:

- узлу A: с IP-адресом узла B и MAC-адресом узла C;
- узлу B: с IP-адресом узла A и MAC-адресом узла C.



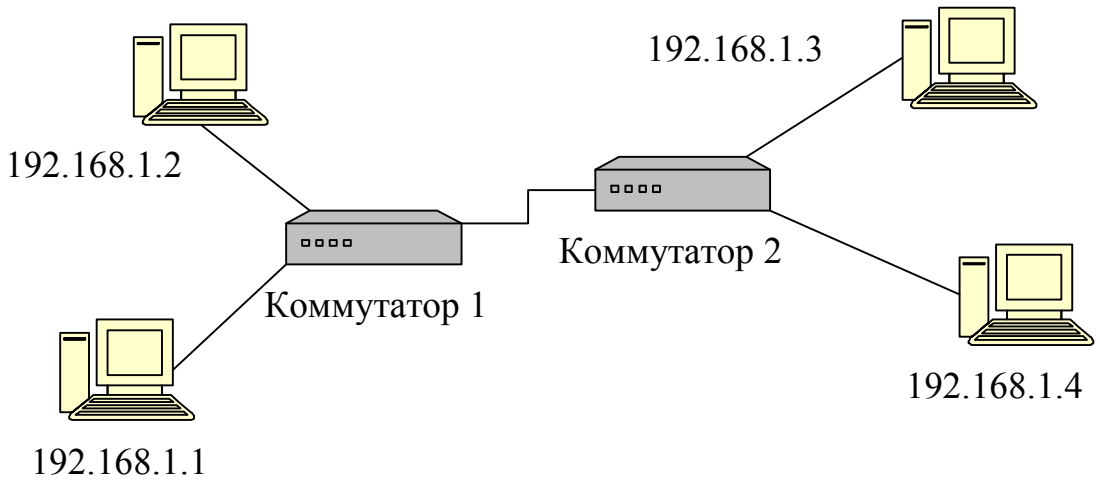
После обновления таблиц, и компьютер A, отправляя данные B, и компьютер B, отправляя данные A, в кадрах указывают MAC-адрес C.

Компьютер C прослушивает и ретранслирует трафик.

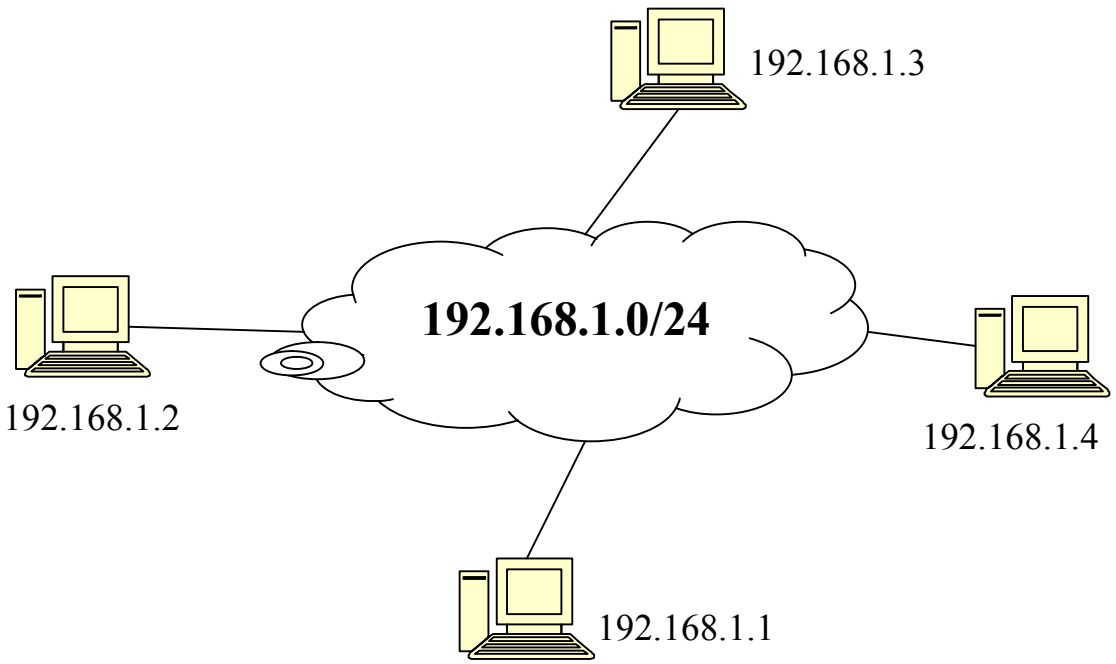


4.1.11 Абстрагирование на сетевом уровне

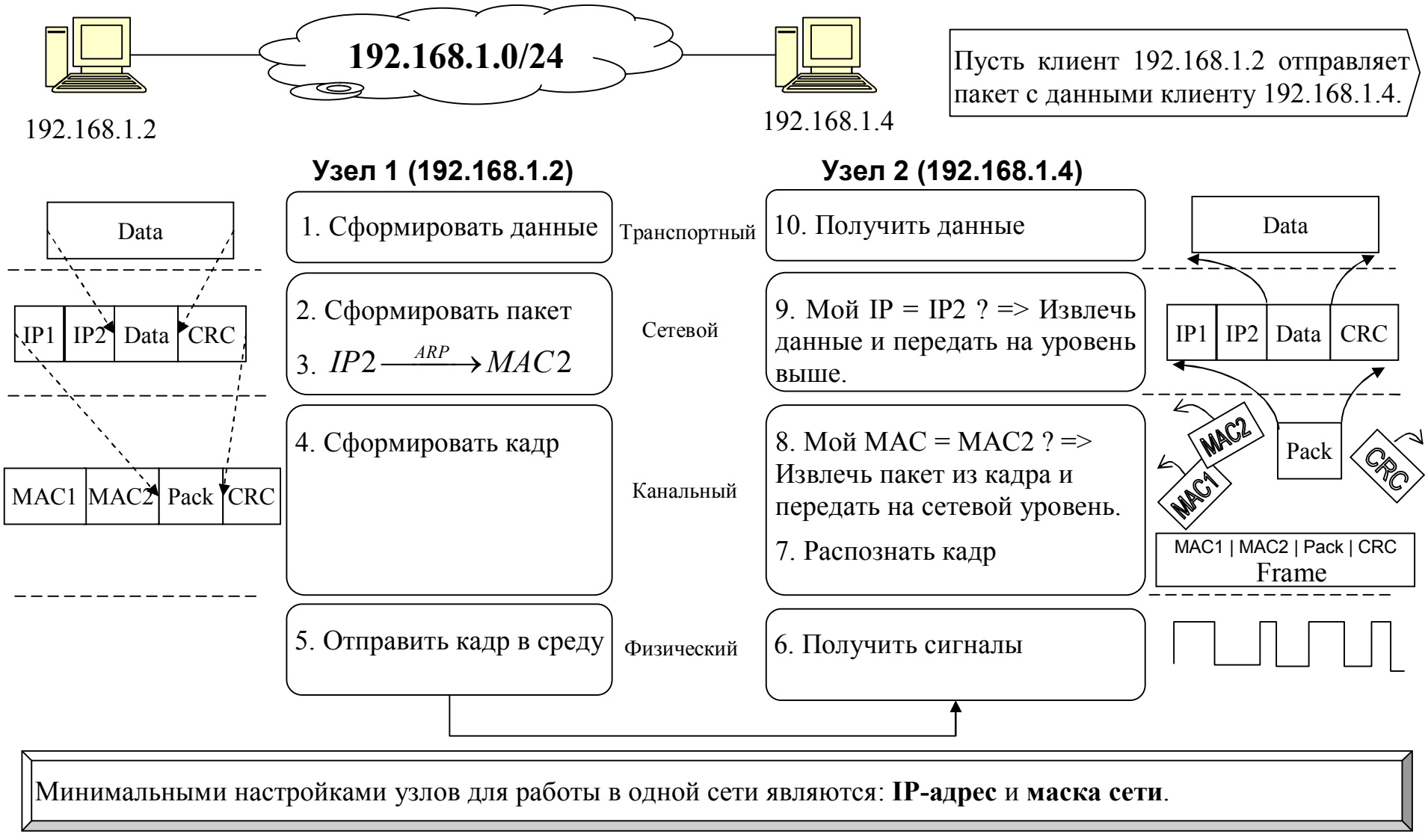
На сетевом уровне организация взаимодействия канального уровня перестаёт интересовать. То есть безразлично, как физически, какими линиями связи, через какие устройства (повторители, коммутаторы) соединены узлы сети.



Поэтому сети на сетевом уровне часто отображают в виде «облаков» с их адресом.



4.1.12 Взаимодействие на сетевом уровне в одной подсети



Минимальными настройками узлов для работы в одной сети являются: **IP-адрес** и **маска сети**.

4.1.13 Оборудование сетевого уровня. Маршрутизатор

Поскольку реализация обмена пакетами между сетями функция более высокого уровня, то эти задачи должны выполнять специальные устройства.

Маршрутизатор (router) — устройство, позволяющее объединять сети между собой, выбирать оптимальный маршрут продвижения пакетов между ними по определенным критериям.

Маршрутизатор

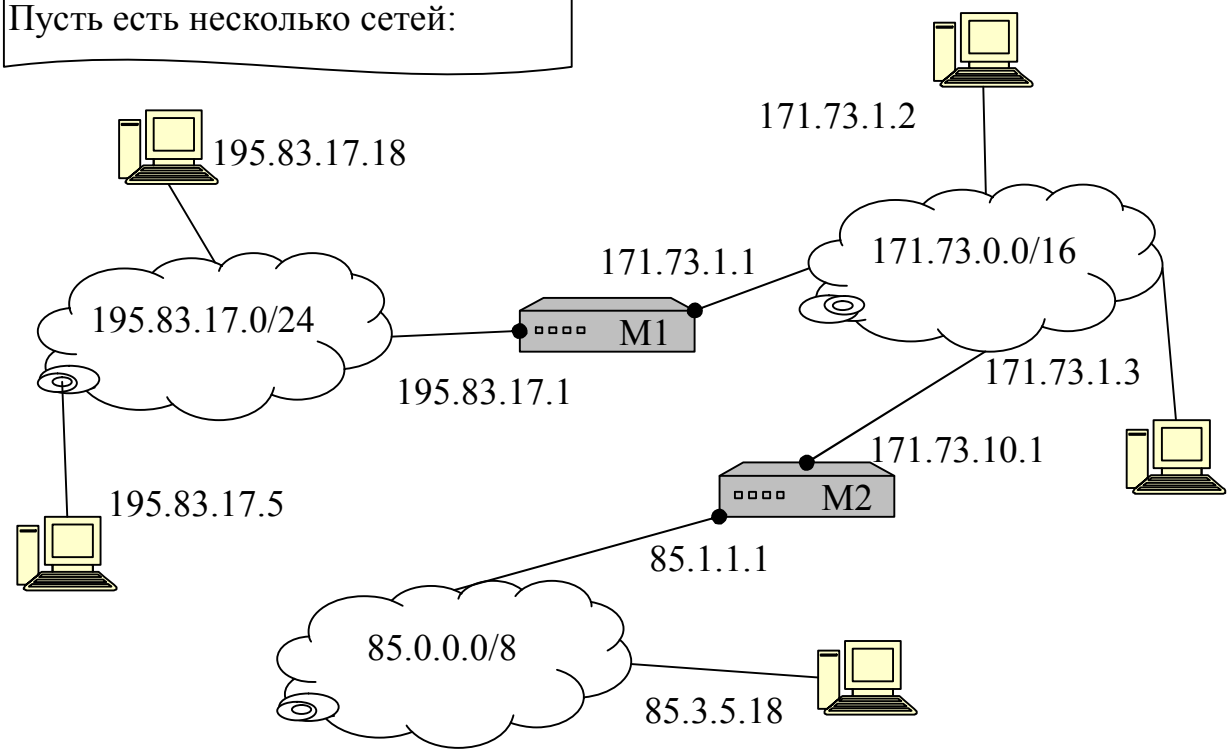
— коммуникационное устройство.

— имеет несколько портов.

— работает под управлением операционной системой (часто UNIX-подобной).

! Каждый порт рассматривается как узел сети и имеет MAC-адрес и сетевой адрес той сети, к которой подключен.

Пусть есть несколько сетей:



Противоречие. Каким образом маршрутизатору продвигать пакеты сквозь сети, используя только IP адреса отправителя и получателя?

4.1.14 Маршрутизация

Маршрутизация — процесс определения маршрута следования пакета (продвижения пакетов из одной сети в другую).

В пакете есть только IP-адрес отправителя и получателя.

Использовать алгоритм как в протоколе ARP невозможно, поскольку широковещательная рассылка между сетями недопустима.

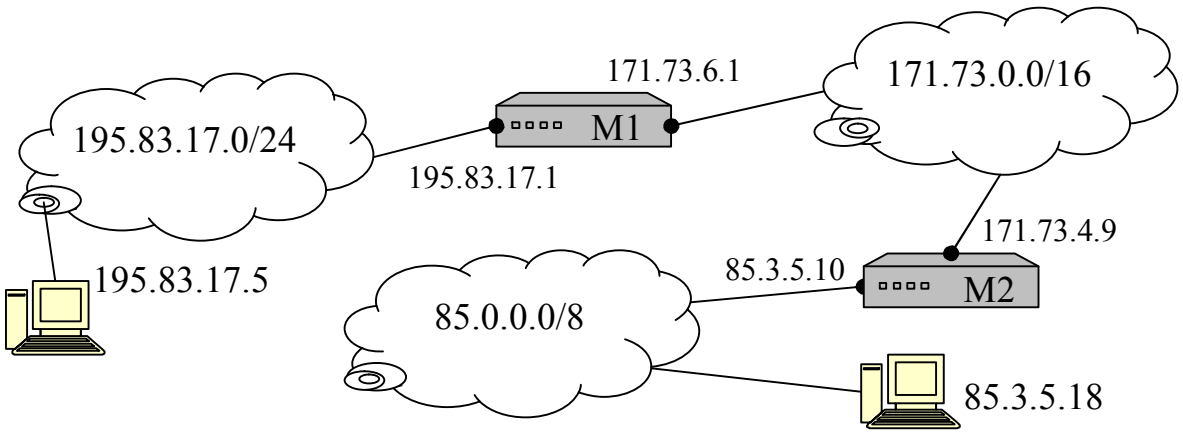
Следовательно, нужно использовать априорную информацию о маршруте, по которому передавать пакет.

При этом необходимо иметь возможность создавать маршруты вручную, а также автоматически самими устройствами.

~ Метафора: продвижение пакетов и путешествие по городам страны.

Город — Сеть, Вокзал — Маршрутизатор

Чтобы из Красноярска попасть в Большой театр в Москве, нужно сначала приехать в аэропорт, самолетом до аэропорта в Москве. Пересесть на метро и доехать до места назначения.



Пусть клиент 195.83.17.5 отправляет пакет клиенту 85.3.5.18.

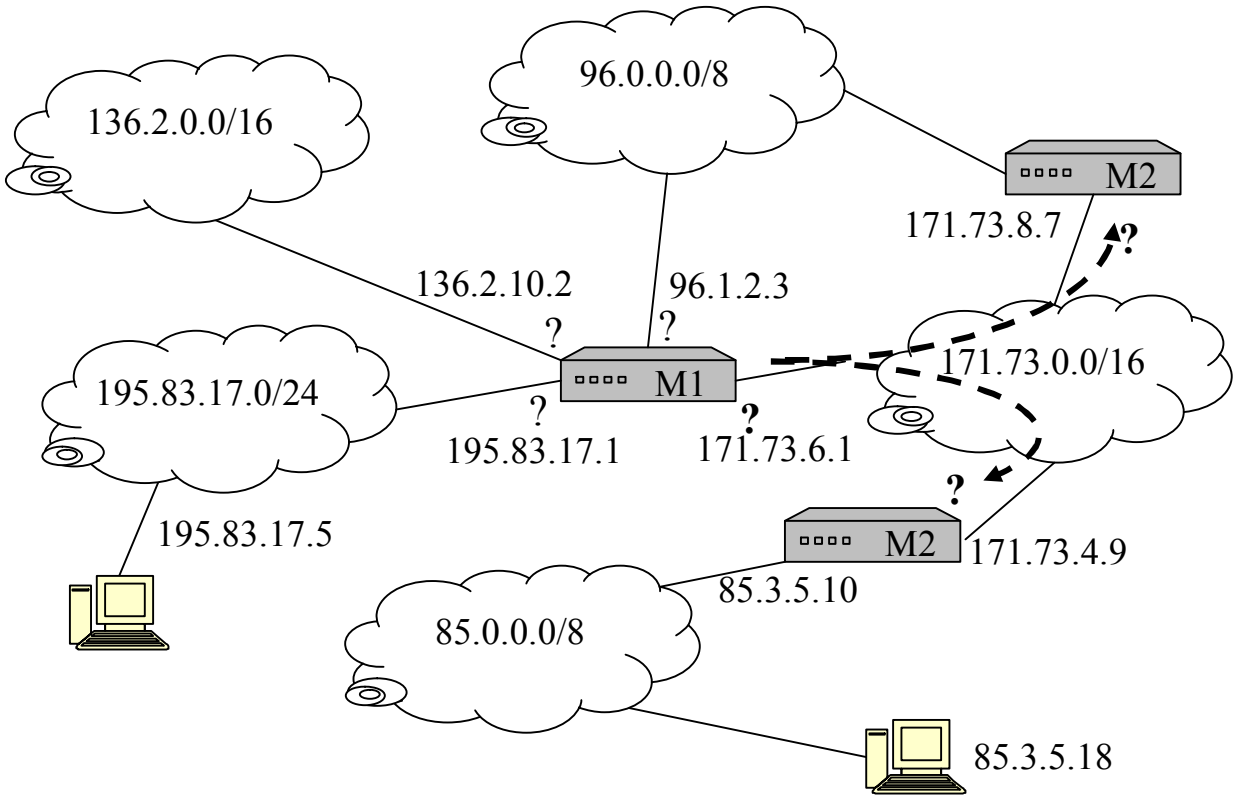
Так как IP-адрес получателя и отправителя находятся в разных подсетях, клиенту нужно заранее знать, куда направить пакет.

Пакет нужно направить тому узлу, который может дальше передать этот пакет. Здесь такой узел — порт 195.83.17.1 маршрутизатора M1.

~ Аналогично, путешествуя в большом городе и расспрашивая дорогу у других жителей города, часто на вопрос «Как доехать из X в Y?» наблюдается такая форма ответа: «Сначала из X доедите до Z, а там спросите как до Y».

! Чтобы передать пакет в другую сеть, его необходимо направить на порт ближайшего маршрутизатора.

Маршрутизатор должен решить, куда дальше следует направить пакет.



Маршрут — это последовательность маршрутизаторов, которые должен пройти пакет от отправителя до узла назначения.

При передаче пакета узлом или маршрутизатором в другую сеть существует две неопределенности:

Через какой свой порт? и **На какой порт следующего маршрутизатора?**

направить пакет.

Чтобы снять неопределенность у **всех** узлов и маршрутизаторов имеется специальная таблица — **таблица маршрутизации**.

4.1.15 Таблица маршрутизации

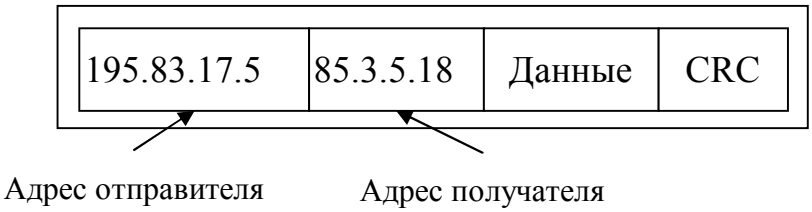
Таблица маршрутизации — ставит в соответствие сетям назначения исходящий интерфейс и адрес порта следующего маршрутизатора.

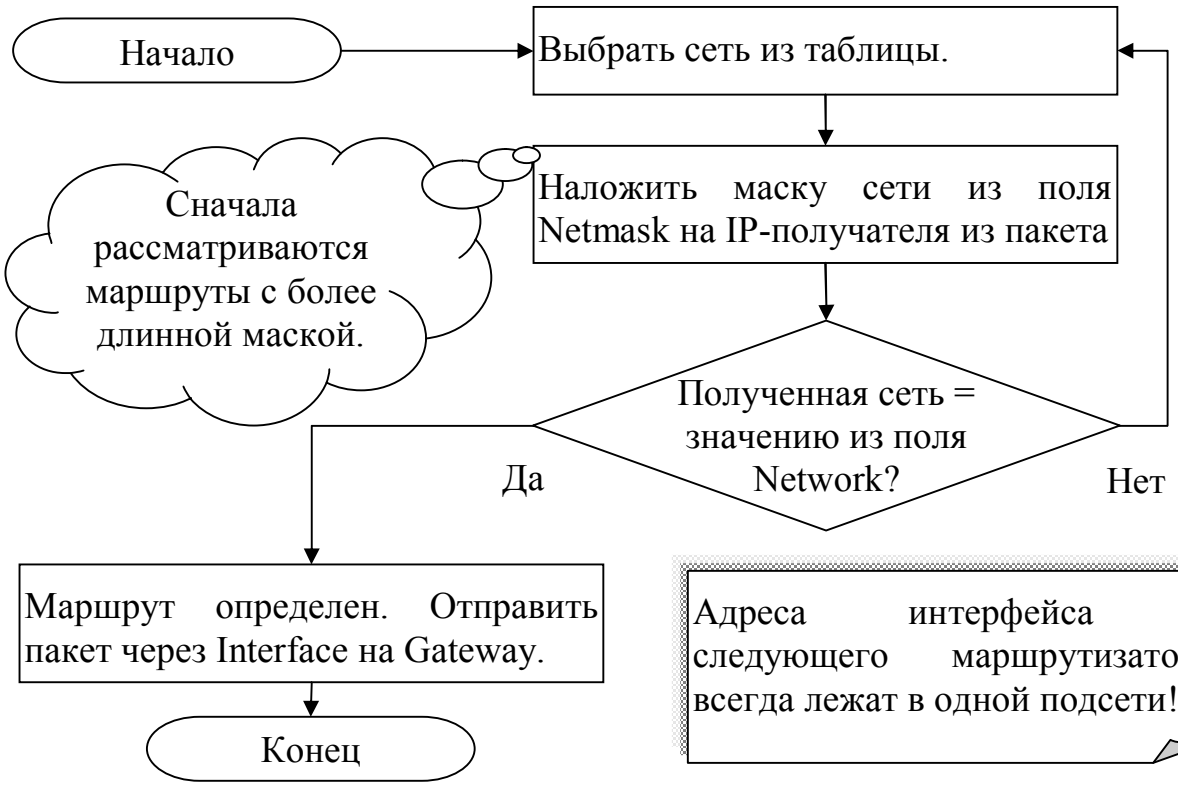
Таблица маршрутизации — атрибут каждого маршрутизатора и узла сети

Например, таблица маршрутизации для М1:

Network Address (адрес сети назначения)	**Netmask (Маска сети этой сети назначения)**	**Interface (Интерфейс отправки)**	**Gateway (Адрес порта следующего маршрутизатора)**	**Metric (Метрика)**
136.2.0.0.	255.255.0.0	136.2.10.2	—	1
96.0.0.0	255.0.0.0	96.1.2.3	—	1
171.73.0.0	255.255.0.0	171.73.6.1	—	2
85.0.0.0	255.0.0.0	171.73.6.1	171.73.4.9	1
195.83.17.0	255.255.255.0	195.83.17.1	—	1

Алгоритм определения маршрута.





Адреса интерфейса и следующего маршрутизатора всегда лежат в одной подсети!

4.1.16 Межсетевое взаимодействие. Маршрутизация

Противоречие. Как передать пакет ближайшему маршрутизатору, если адрес получателя в IP-пакете изменять нельзя?

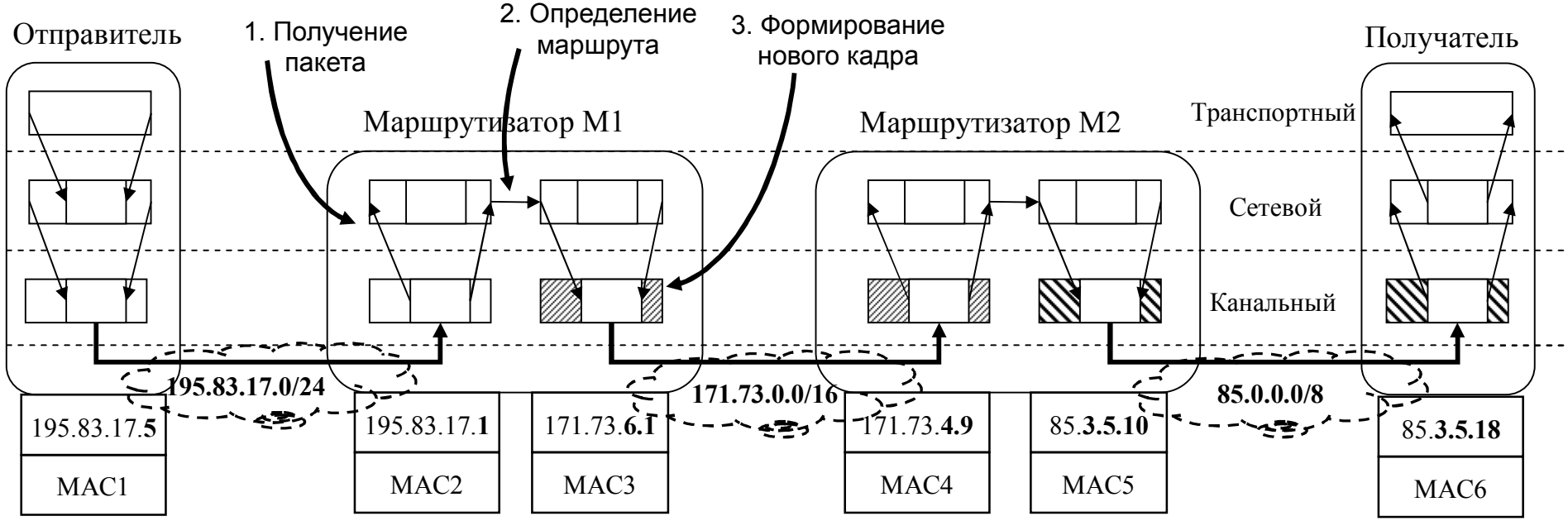
Свой MAC1	MAC2 след.маршр.	195.83.17.5	85.3.5.18	Данные	CRC	CRC

Таблица маршрутизации для M1:

Network	Netmask	Interface	Gateway	Metric
…	…	…	…	…
85.0.0.0	255.0.0.0	171.73.6.1	171.73.4.9	2
…	…	…	…	…

Таблица маршрутизации для M2:

Network	Netmask	Interface	Gateway	Metric
…	…	…	…	…
85.0.0.0	255.0.0.0	85.3.5.10	—	1
…	…	…	…	…



4.1.17 Шлюз по умолчанию (gateway)

Обычно в тупиковых сетях и их узлах существует только один путь в другие сети.

Поэтому следует упростить процесс маршрутизации в этих узлах.

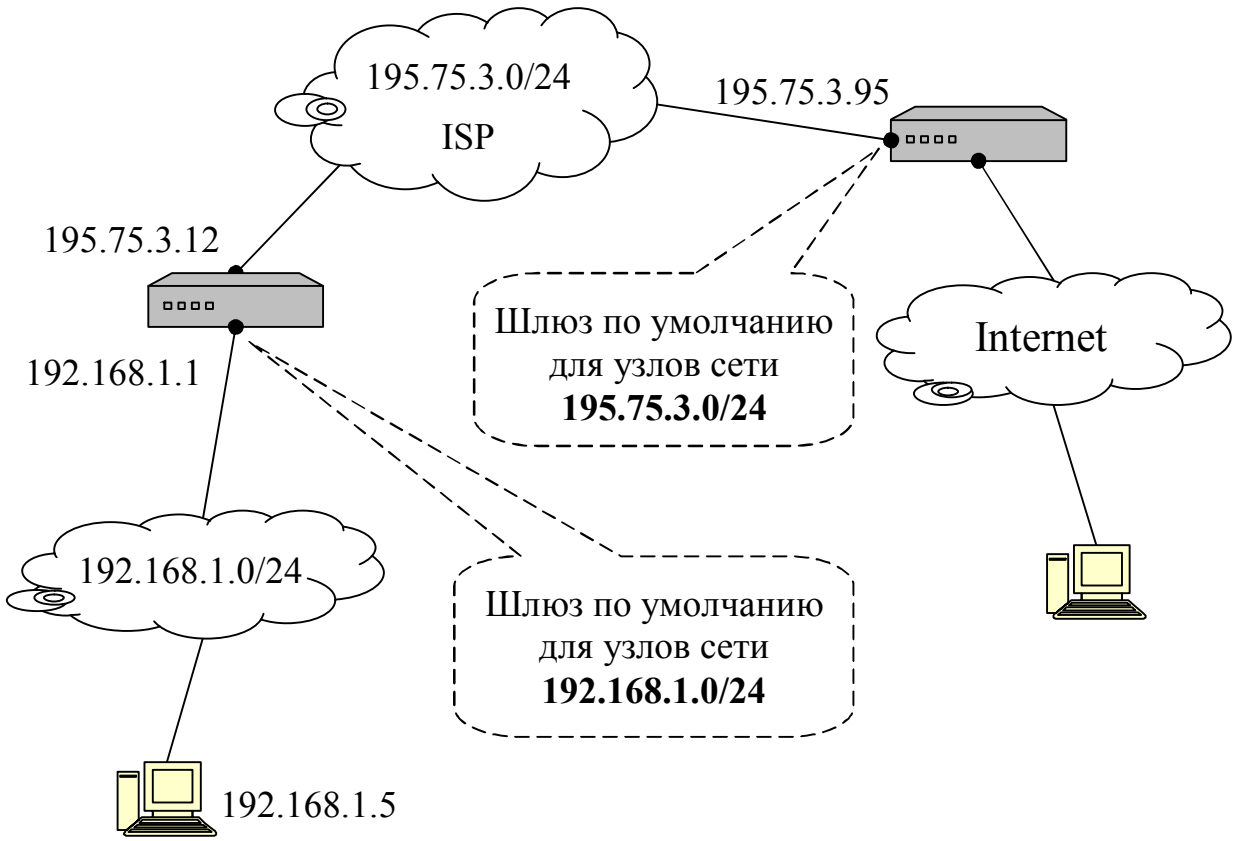
Шлюз по умолчанию — адрес порта маршрутизатора, на который отправляется трафик (пакеты), для которого невозможно определить маршрут по адресу получателя.

Шлюз по умолчанию в ОС и ПО может быть определен как отдельная настройка.

Шлюз по умолчанию в таблице маршрутизации

network	netmask	interface	gateway
0.0.0.0	0.0.0.0	<IP>	GW IP

Т.к. при наложении маски 0.0.0.0 на любую сеть получится адрес сети 0.0.0.0.



Минимальными настройками узлов для взаимодействия с различными сетями являются: **IP-адрес**, **маска сети** и **шлюз по умолчанию**.

4.1.18 Протокол сообщений ICMP

Неверный IP-адрес

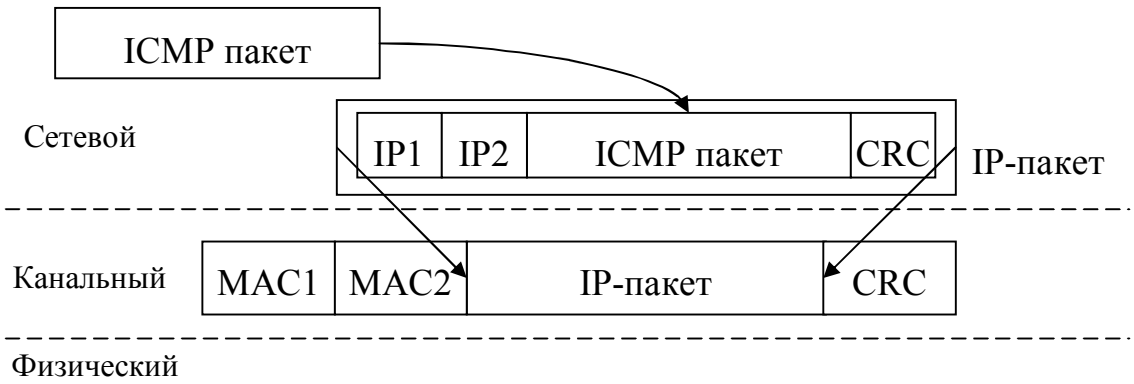
Неверный маршрут в таблице маршрутизации.

Проблемы на физическом или канальном уровне сети.

Противоречие. Необходимы определение проблем при передаче данных и обратная связь для отправителя.

ICMP (Internet Control Message Protocol, протокол межсетевых управляющих сообщений) — сетевой протокол, стека TCP/IP. Используется для передачи сообщений об ошибках и других исключительных ситуациях, возникших при передаче данных.

Пакеты ICMP при передаче инкапсулируются в IP-пакеты.



ICMP пакет имеет поля:

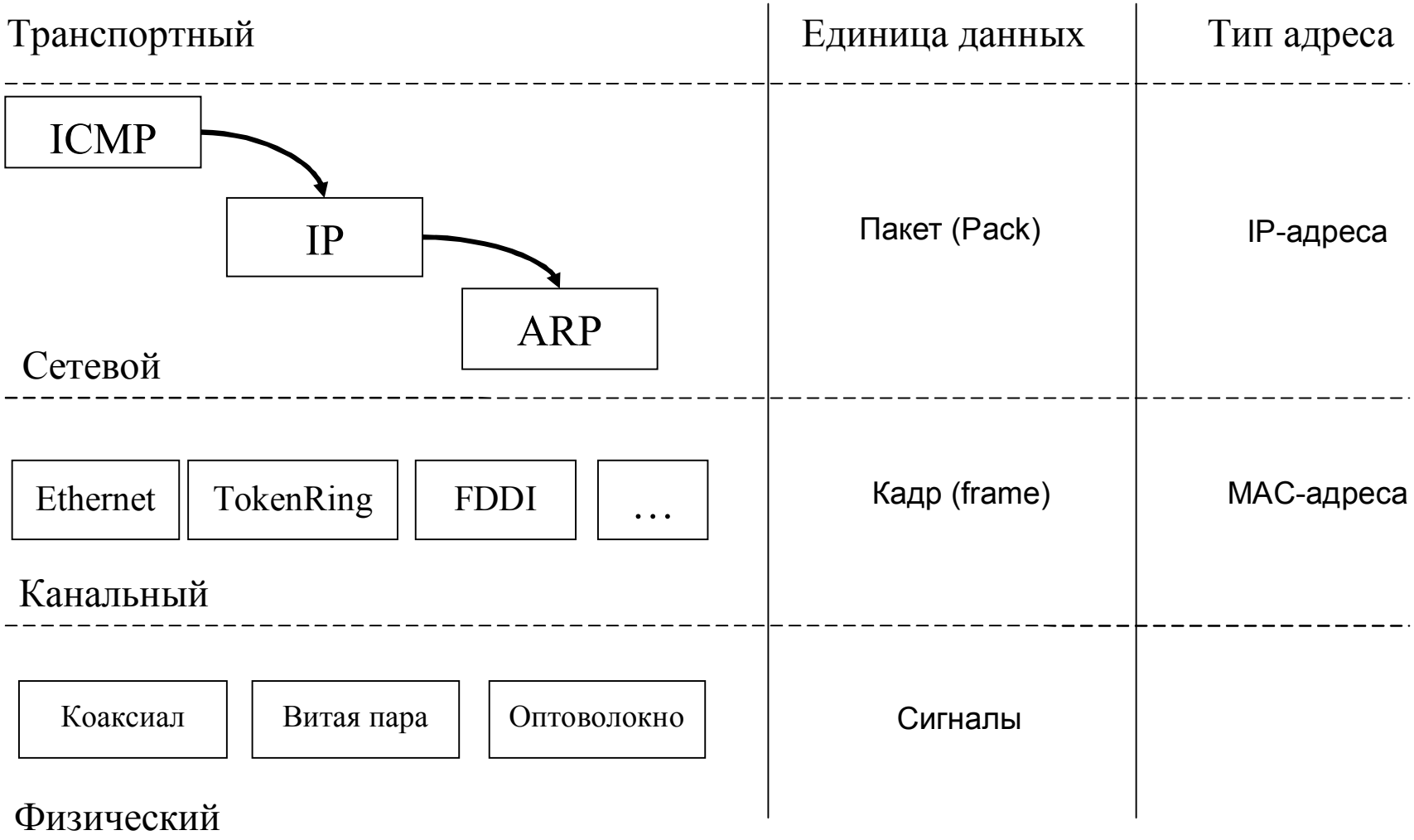
- Тип сообщения (эхо-ответ, адресат недоступен и др.)
- Код – зависит от типа (Сеть недостижима и т.д.)
- Контрольная сумма

Например. При прохождении пакета через маршрутизатор, он изменяет в IP-пакете поле TTL (TTL = TTL – 1)

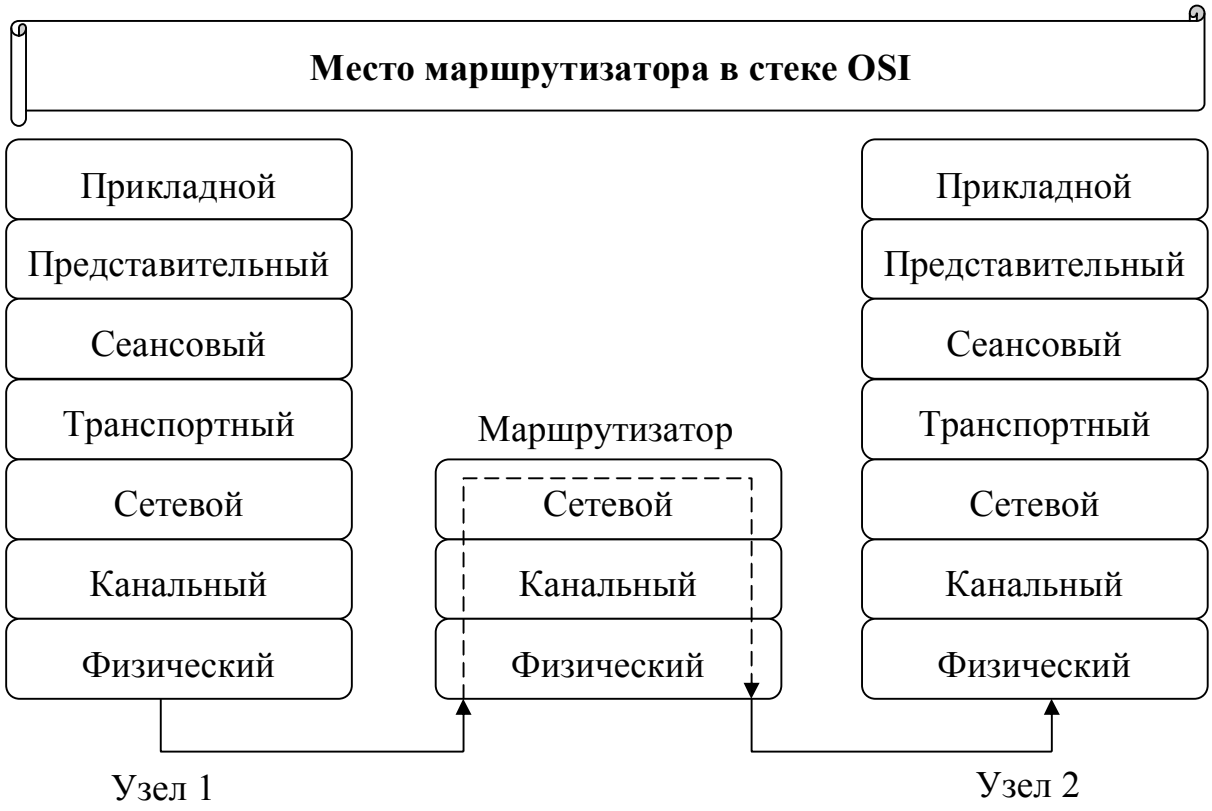
Когда TTL = 0, пакет уничтожается, а отправителю отправляется ICMP сообщение с кодом 11 (превышен TTL).

! ICMP не используются в приложениях напрямую.
Исключения — утилиты **ping** и **traceroute**.

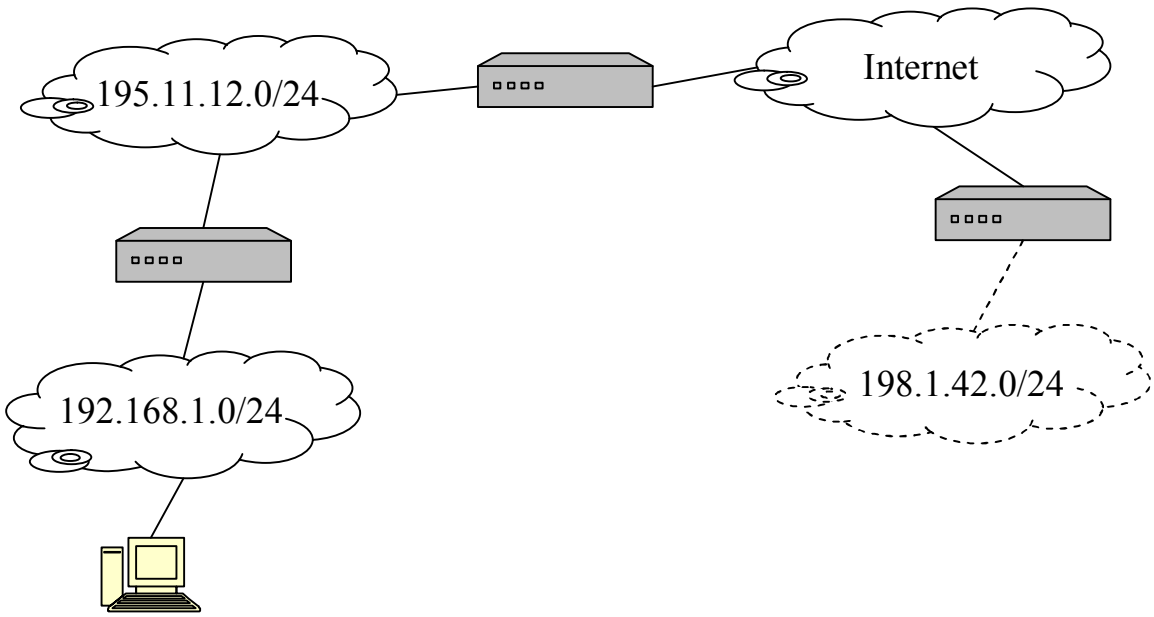
4.1.19 Протоколы и уровни OSI (канальный, сетевой)



4.1.20 Маршрутизатор и его место в стеке OSI

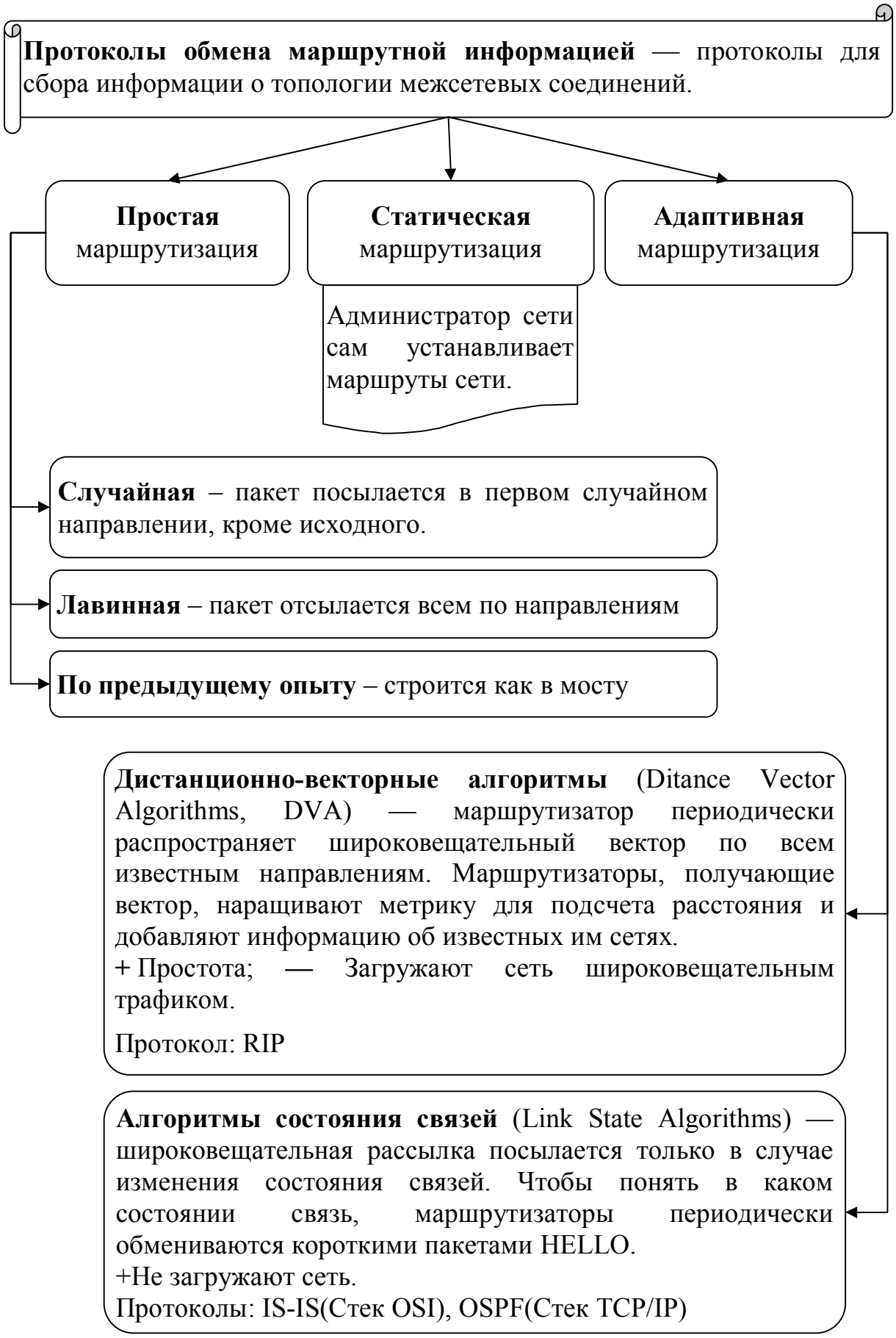


Пусть есть очень большая сеть, к которой где-то добавляется новая.



Противоречие. Необходимо обновить информацию о маршрутах на очень большом количестве маршрутизаторов, чтобы новая сеть стала доступна из всех остальных сетей.

4.1.21 Протоколы обмена маршрутной информацией



4.1.22 Проблемы маршрутизации и передачи данных на сетевом уровне

Проблемы маршрутизации:

Для обновления маршрутов на всех маршрутизаторах сети объективно требуется время.

Новая сеть не сразу будет доступна из любых других сетей. В Интернете на это может уйти сутки.

Статическая маршрутизация не исключает ошибки, в том числе и зацикливание пакетов.

В IP-пакете существует числовое поле TTL, которое уменьшается на 1 после маршрутизации, а при достижении 0 пакет удаляется.

При реальной передаче разные пакеты одного сообщения могут пройти разный путь.

На принимающей стороне необходимо собрать пакеты в правильном порядке.

Проблемы передачи данных на сетевом уровне:

Не гарантирована надежность. На сетевом уровне никак не контролируется надежность доставки сообщений. То есть отправитель никак не информируется об удачной или неудачной доставке пакета.

Отсутствует мультиплексирование прикладных потоков. На сетевом уровне не обеспечивается доставка данных между приложениями. Так как пакет доставляется узлу, а не прикладному приложению.

Нет управления загрузкой сети. На сетевом уровне не регулируется загруженность сети. Никак не регламентируется, сколько пакетов за один раз следует отправлять в сеть.

Для решения обозначенных проблем, необходимо реализовать более высокоуровневые функции.

4.2 Транспортный уровень модели OSI

На транспортном уровне организуется транспортная система между прикладными процессами.

- Мультиплексирование нескольких соединений между прикладными протоколами.
- Обеспечение надежности доставки сообщений.
- Управление загрузкой сети.

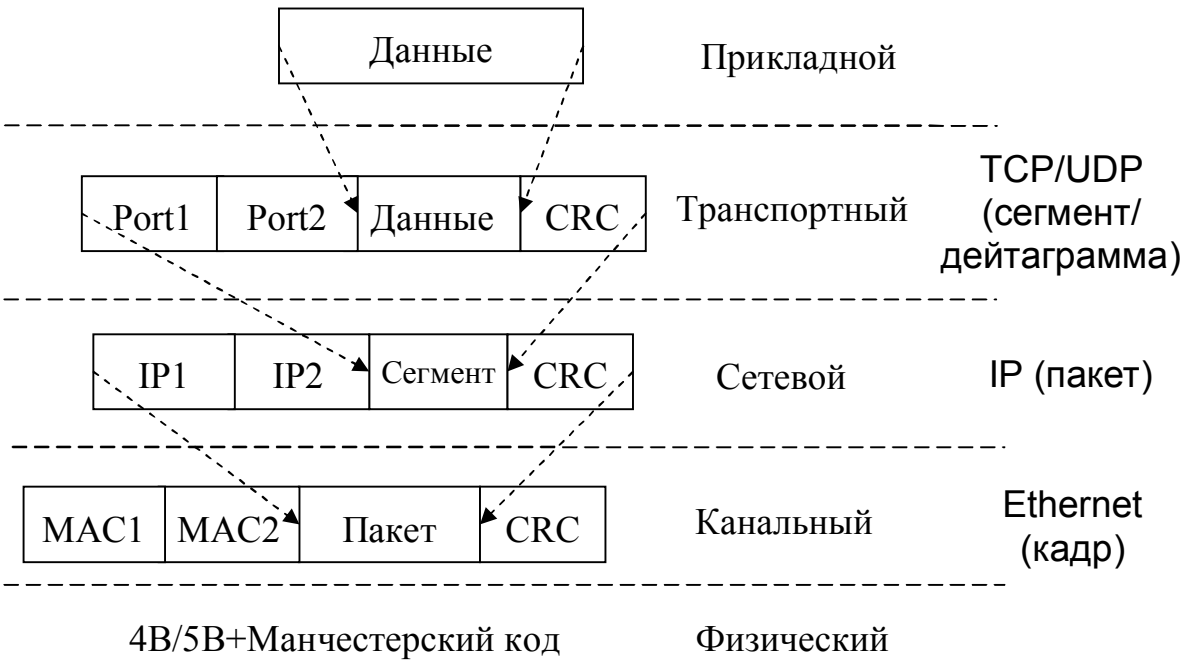
Прикладной
Представительный
Сеансовый
Транспортный
Сетевой
Канальный
Физический

Единица данных: **сегмент (segment), дейтаграмма (datagram)**

Тип адресов: **порт** (**port**)

Протоколы: **TCP, UDP, SPX и др.**

Передача данных между уровнями:



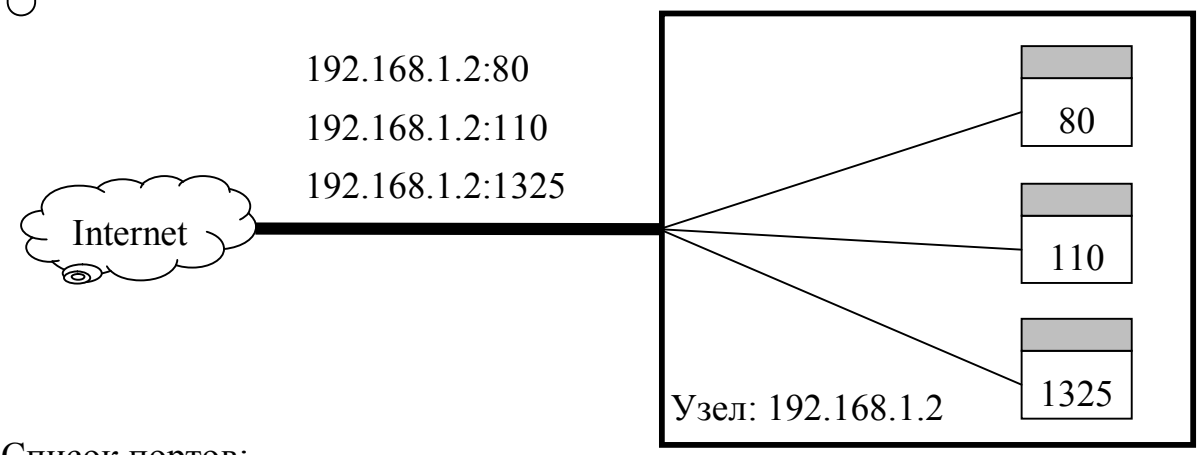
4.2.1 Порты

IP-адрес отражает одно сетевое соединение.

Как же взаимодействовать отдельным приложениям друг с другом?

Для идентификации приложений ввели **порты**.

Порт (**port**) — двухбайтное число, используемое на транспортном уровне для идентификации прикладного процесса.



Список портов:

0 . . . 1023	**Стандартизированные** – зарегистрированы для общеизвестных приложений.
1024 . . . 5000	**Временные** – для использования в различных приложениях.
5001 . . . 65535	**Пользовательские** – для использования в различных приложениях.

Порты некоторых служб:
0 – Зарезервировано
20 – FTP (data)
21 – FTP (command)
22 – SSH
23 – Telnet
25 – SMTP
67, 68 – DHCP
80 – WWW
53 – DNS
110 – POP3
137-139 – NetBIOS
443 – SSL
1433 – MS SQL
3389 – RDP
5190 – ICQ
5500 – VNC

4.2.2 Протокол TCP

- Протокол TCP (Transmission Control Protocol) – протокол транспортного уровня
- Стек TCP/IP
- Протокол с предварительным установлением соединения.
- Протокол реализует механизм надежности доставки данных

- Единицей данных является сегмент (segment)
- Сегмент имеет порт отправителя и порт получателя
- Порт отправителя идентифицирует приложение на клиентском узле, аналогично порт получателя — приложение на узле сервера.

Формат TCP-сегмента:

Порт источника 16 бит			Порт назначения 16 бит
Номер последовательности (SEQ_NUM) 32 бита			
Номер подтверждения (ACK_NUM) 32 бита			
Смещение данных, 4 бита	Зарезервировано, 6 бит	Флаги, 6 бит	Размер окна, 16 бит
Контрольная сумма, 16 бит			Указатель важности, 16 бит
Опции (необязательно) 32 бита			
Данные			

TCP-флаги:

URG — Поле «Указатель важности задействовано»

ACK — Поле «Номер подтверждения задействовано»

PSH —инструктирует получателя протолкнуть данные, накопившиеся в приемном буфере, в приложение пользователя

RST — Оборвать соединения, сбросить буфер (очистка буфера)

SYN — Синхронизация последовательности номеров сегментов

FIN — указывает на завершение соединения

4.2.3 Протокол UDP

- Протокол UDP (User Datagram Protocol) – протокол транспортного уровня
- Стек TCP/IP
- Протокол без предварительного установления соединения
- Протокол без механизма надежной доставки сообщений

- Единицей данных является дейтаграмма (datagram)
- Дейтаграмма имеет порт отправителя и порт получателя
- Обмен данными по одному TCP-порту не мешают обмену по тому же UDP-порту.

Формат UDP-дейтаграммы:

Порт источника 16 бит	Порт назначения 16 бит
Длина дейтаграмму 16 бит	Контрольная сумма 16 бит
Данные	

Недостаточная надежность протокола при потере пакетов влечет их:

дублирование

UDP используется…

При передаче больших данных:

- потоковое видео,
- игры реального времени,
- в случае передачи больших объемов данных нескольким клиентам.

При передаче коротких сообщений:

- Когда подтверждение получения не требуется
- Когда устанавливать соединение слишком накладно

4.2.4 Установка TCP-соединения

TCP-соединение происходит по схеме «трехкратного рукопожатия» (three-way handshake).

1. Запрашивающая сторона отправляет SYN сегмент, указывая номер порта сервера и исходный номер последовательности клиента ($ISN_{клиента}$).

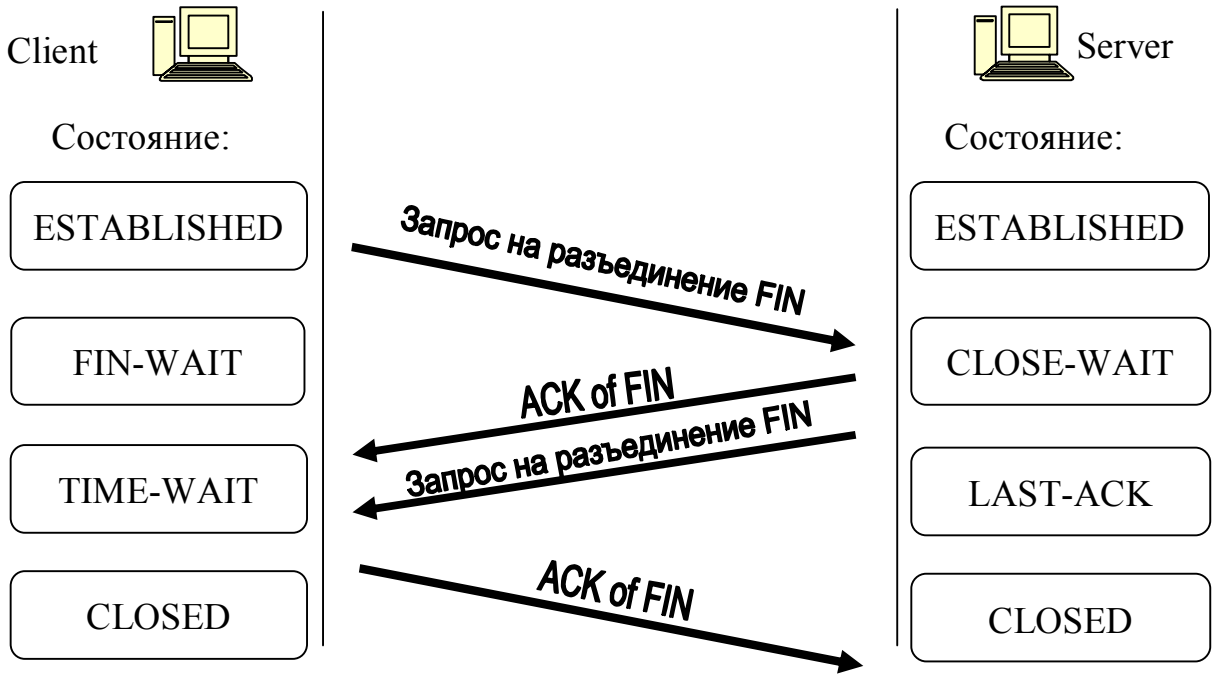
2. Сервер отвечает своим сегментом SYN, содержащим исходный номер последовательности сервера ($ISN_{сервера}$). Сервер подтверждает приход SYN клиента с использованием ACK ($ISN_{клиента}+1$).

3. Клиент должен подтвердить приход SYN от сервера с использованием ACK ($ISN_{сервера}+1$).

Уже установившееся соединение называется «**наполовину открытым**», если одна из программ TCP закрыла соединение, или отказалась от него. Причем сделала это на своем конце, не предупредив своего партнера.

4.2.5 Разрыв TCP-соединения

TCP соединение полнодуплексное, т.е. данные могут перемещаться в каждом направлении независимо от другого направления. Поэтому каждое направление должно быть закрыто независимо от другого.



1. Если, клиент завершает соединение, то он отправляет FIN-сегмент.

2. Сервер подтверждает приход FIN клиента с использованием ACK ($SEQ\_NUM_{клиента}$+1).

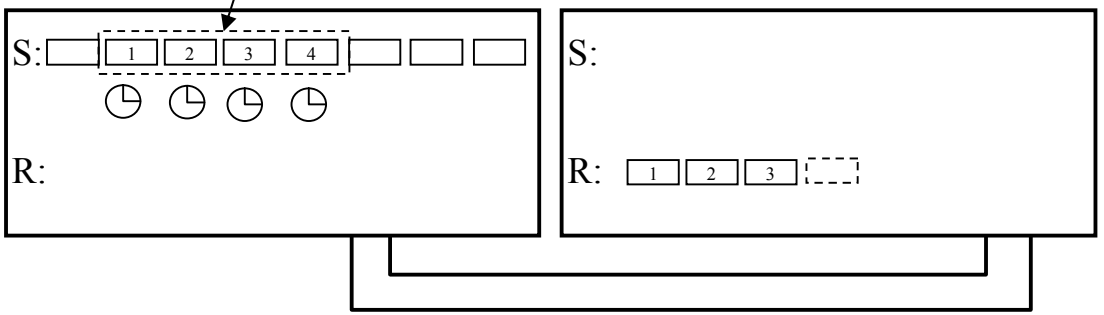
3. Сервер также завершает соединение и отправляет FIN-сегмент.

4. Клиент подтверждает приход FIN сервера с использованием ACK ($SEQ\_NUM_{клиента}$+1).

Клиенты должны сохранять уже закрытые ими для чтения информации соединения до тех пор, пока программа протокола TCP не сообщит им, что такой информации больше нет.

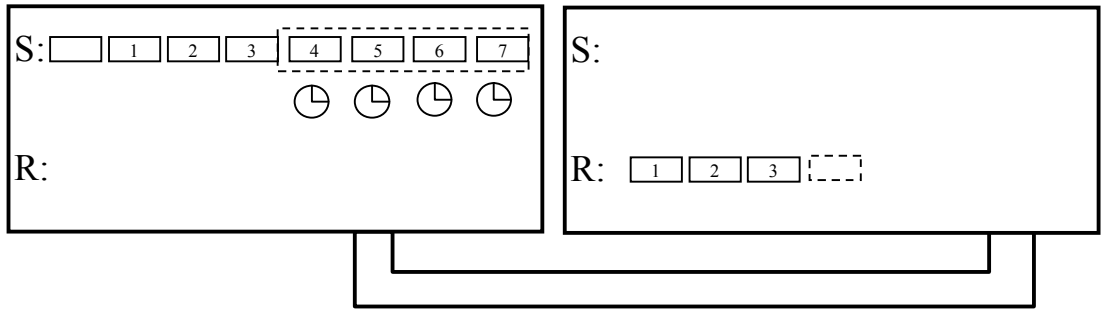
4.2.6 Скользящее окно (Sliding window)

Скользящее окно (Sliding window) – это алгоритм, определяющий количество байт, которые будут отправляется в сеть за один раз.



1. Модуль TCP в рамках окна нарезает поток байт на сегменты. Отправляет сегменты в сеть и ждет тайм-аут.

2. Получатель, получив сегменты, отправляет квитанцию об успешном приеме



3. Отправитель смещает окно и отправляет новые сегменты и повторяет непринятые.

Если для получения требуется большой тайм-аут, отправитель может снизить размер окна.

Если сеть загружена на маршрутизаторах, то они посылают сообщения об уменьшении окна.

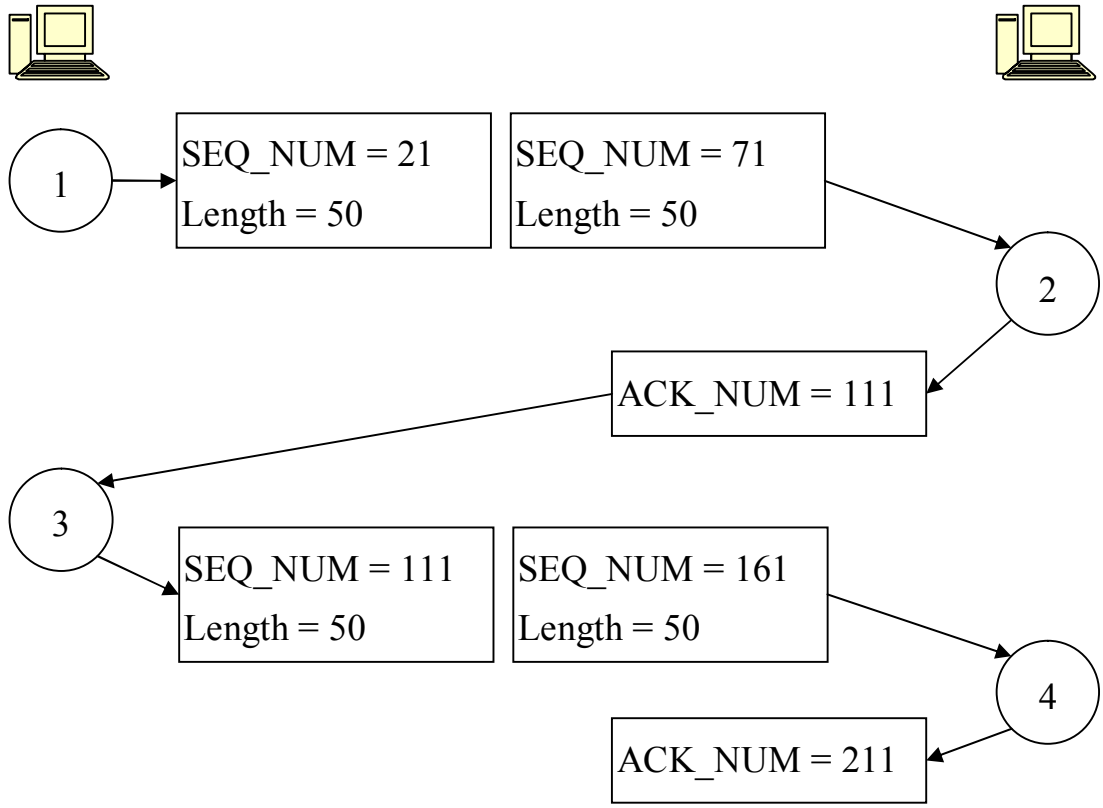
! Скользящее окно несет функцию управления загрузкой сети!

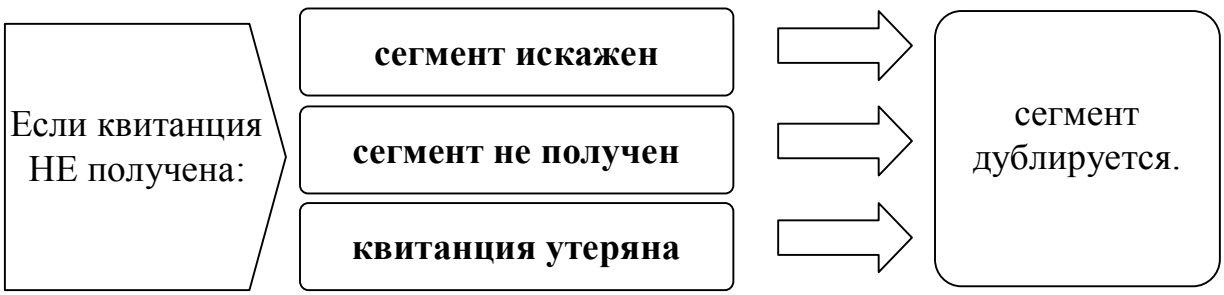
4.2.7 Механизм квитирования

Квитирование – механизм обеспечения надёжности доставки сообщений, который обязывает получателя посылать отправителю короткое сообщение (квитанцию) об успешном получении сообщения.

Чтобы не отправлять отдельные квитанции на все принятые сегменты, получатель отправляет квитанцию с запросом следующего за успешно принятым байтом.

ACK_NUM = SEQ_NUM+1





4.2.8 Межсетевой экран (firewall)

Противоречие. При увеличении количества сетей, служб и пользователей, возникает проблема появления вредоносного трафика.

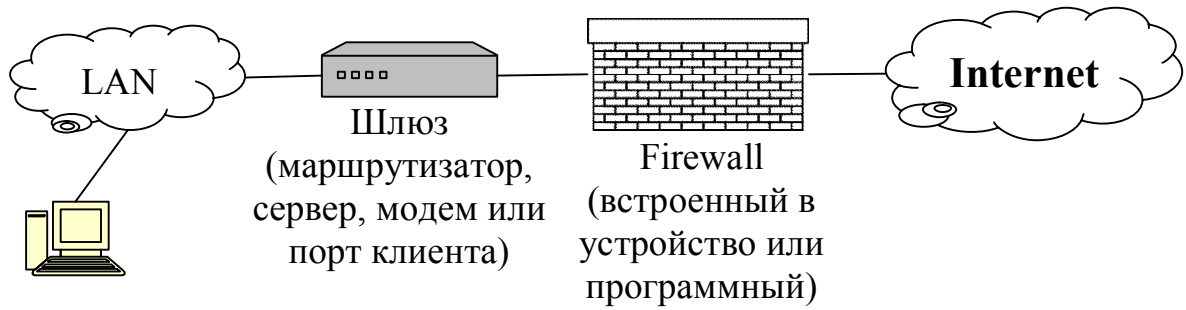
Межсетевой экран (firewall, межсетевой фильтр, файервол, брандмауэр) — это аппаратное или программное обеспечение, осуществляющее контроль и фильтрацию проходящих через него сетевых пакетов в соответствии с заданными правилами.

Файервол устанавливается:

- На внешний порт шлюза в другие сети.
- На порты клиентов сети.

Файервол фильтрует трафик по критериям:

- По направлению (входящий/исходящий)
- По протоколу (TCP/UDP)
- По IP адресам отправителя и получателя
- По портам отправителя и получателя
- По любому полю заголовков TCP/IP и др.



Настройки файервола описываются в виде правил трех видов:

Allow – разрешить

Deny – запретить

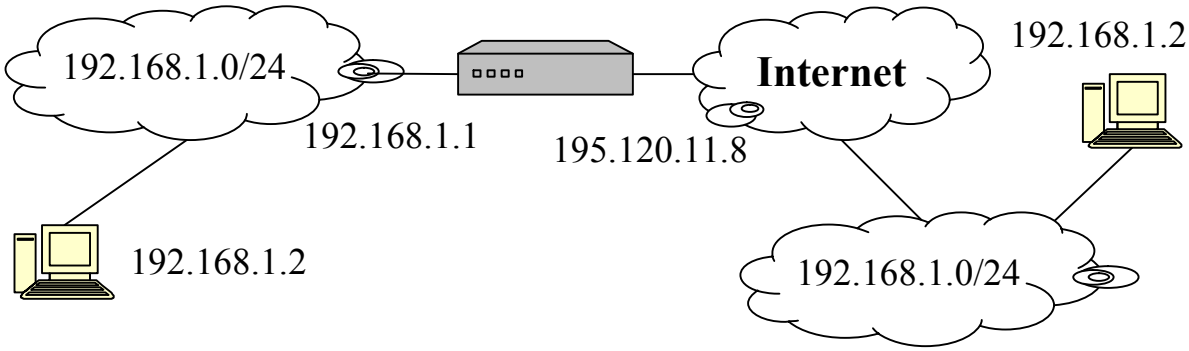
Diver – перенаправить

N	(Allow, Deny, Diver) <пакеты> from IP1:port1 to IP2:port2 <options>
1	Allow TCP from any to IP1:80 setup
2	Allow TCP from any to any
3	Divert IP2:80 TCP from any to IP1:80
4	Deny all from any to any

Файервол для: фильтрации доступа к заведомо незащищенным службам; контроля доступа к узлам сети; регистрации доступа как извне, так и изнутри; регламентирования порядка доступа к сети; уведомления о подозрительной деятельности.

4.2.9 Сетевая трансляция адресов (NAT)

Пусть есть две LAN-сети, подключенных к Интернет:

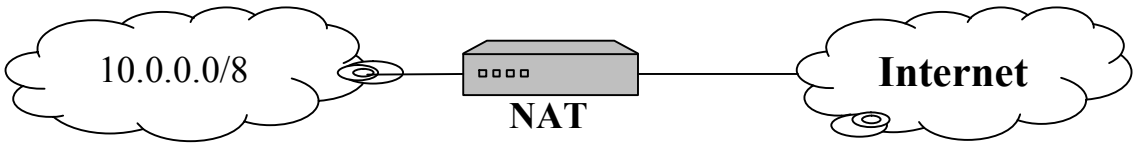


Иногда предприятию не удается получить нужное количество IP-адресов (дефицит, стоимость).

Противоречие. Как обеспечивать связь между узлами сетей с «серыми» IP-адресами, используя только один внешний IP-адрес?

Скомпенсировать **неуникальность** IP-адреса — **уникальным** портом.

Сетевая трансляция адресов (Network Address Translation, Network Masquerading) — механизм в сетях TCP/IP, позволяющий преобразовывать IP-адреса транзитных пакетов.



NAT –реализуют аппаратно или программно на маршрутизаторе или сервере.

Local IP	Local Port	Glob. IP	Glob. Port
10.0.0.3	8089	85.32.1.8	80
10.0.0.3	80		2165
10.0.0.7	3786		1354
10.0.0.7	6875		8726

4.2.10 SNAT и DNAT

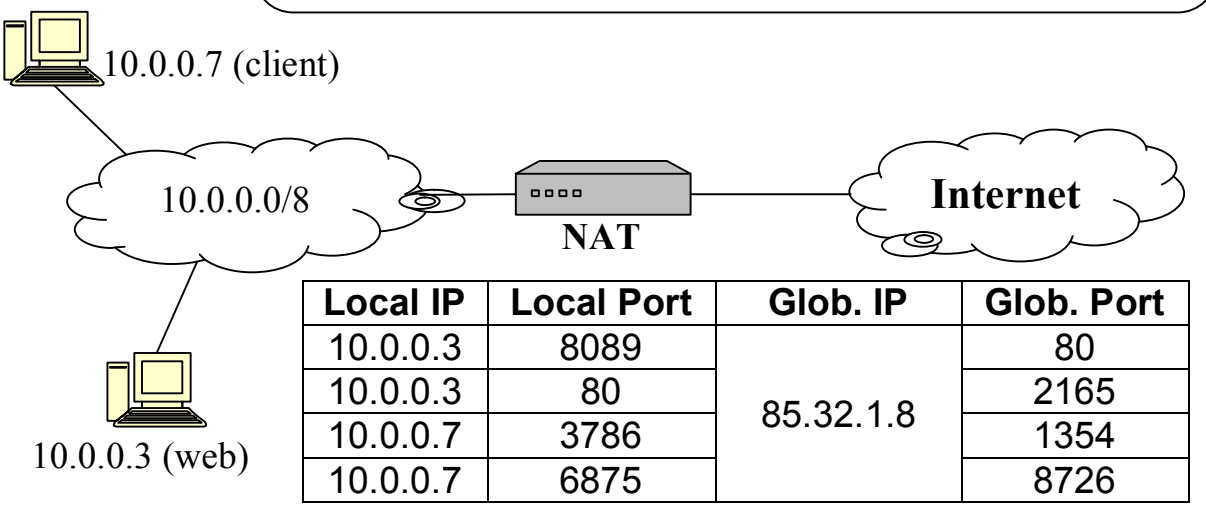
NAT

- прозрачен для клиента и для сервера
- не устанавливает разрыва соединения
- скрывает информацию о частной сети
- увеличивает адресное пространство частной сети

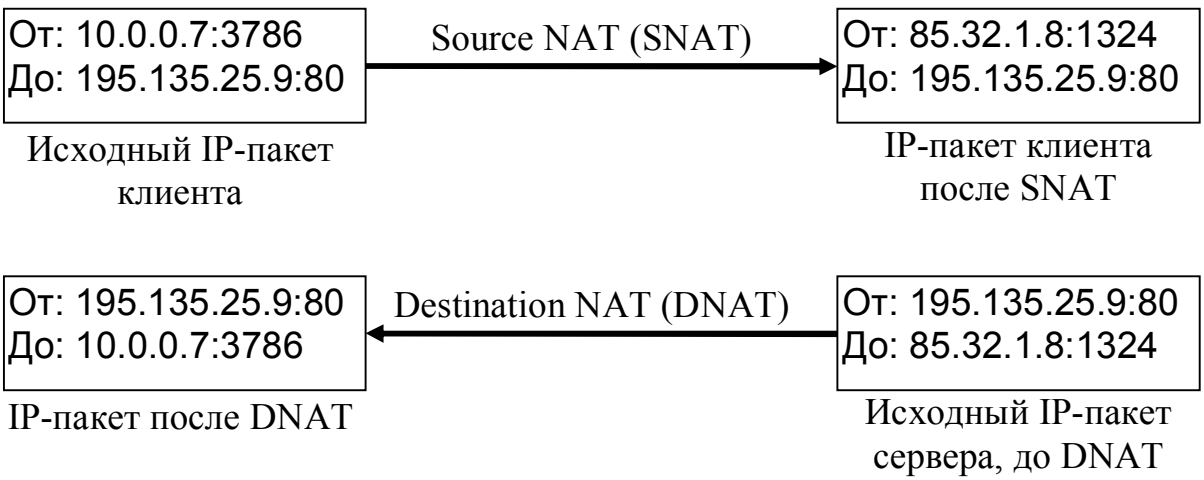
NAT

Source NAT (SNAT) – преобразование IP-адреса источника (отправителя). Позволяет клиентам частной сети выходить в Интернет.

Destination NAT (DNAT) – преобразование IP-адреса назначения (получателя). Позволяет клиентам получать данные из Интернета, а также публиковать внутренние службы сети.



Local IP	Local Port	Glob. IP	Glob. Port
10.0.0.3	8089	85.32.1.8	80
10.0.0.3	80		2165
10.0.0.7	3786		1354
10.0.0.7	6875		8726



4.2.11 IP-спуфинг (IP-spoofing)

В IP-пакете присутствует адрес отправителя и адрес получателя.

Злоумышленник может создать пакет, в котором адрес получателя будет изменен с некоторой целью.

IP-spoofing — вид атаки, заключающийся в использовании чужого IP-адреса с целью обмана системы безопасности.

Сравнительно легко атака осуществима через UDP протокол

В некоторых случаях возможна в TCP-соединениях.

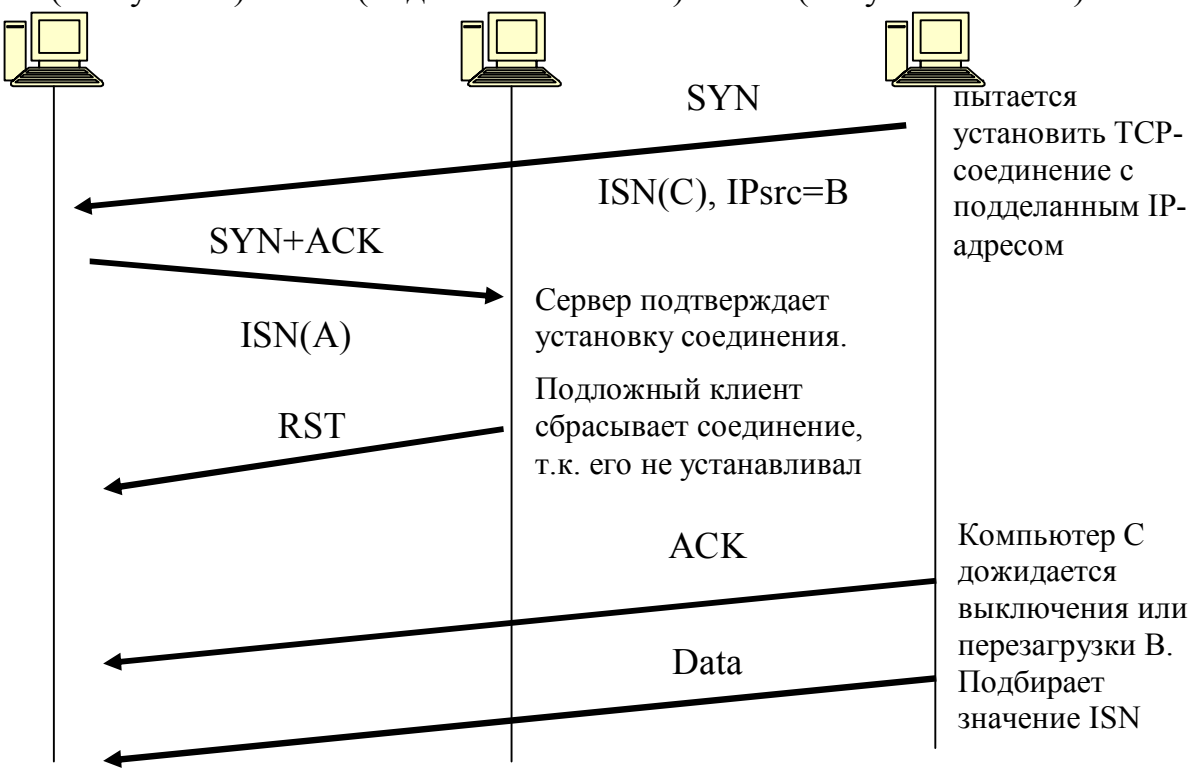
На сетевом уровне атака частично предотвращается с помощью фильтра пакетов.

Он не должен пропускать пакеты, пришедшие через те сетевые интерфейсы, откуда они прийти не могли.

Например, с внешнего интерфейса с «серым» адресом отправителя.

IP-спуфинг через TCP

Основан на подборе SeqNum. Возможно лишь в старых ОС. Современные ОС пытаются предотвращать угадывание.



5 Прикладные протоколы и службы

5.1 Сеансовый уровень

Сеансовый уровень обеспечивает поддержание сеанса связи, позволяя приложениям взаимодействовать между собой длительное время.

- управляет активностью сторон,
- синхронизацией,
- позволяет устанавливать контрольные точки в длинные сообщения.

Функции данного уровня часто реализуются протоколами других уровней.

Протоколы: TCP/UPD, NetBIOS, NCP, SAP**, Сеансовый протокол OSI**

5.2 Представительный уровень

Представительный уровень организует определенную форму представления информации.

- Производит кодирование и декодирование клиентских данных.
- Осуществляет шифрование данных.

Функции данного уровня часто реализуются протоколами других уровней.

Протоколы: **SSL, Представительный протокол OSI,** SMB, NCP, SAP

5.3 Прикладной уровень

Прикладной уровень обеспечивает взаимодействие пользовательских приложений с сетью.

- позволяет приложениям использовать сетевые службы
- формирует запросы к уровню представления

Прикладной
Представительный
Сеансовый
Транспортный
Сетевой
Канальный
Физический

Единица данных**: сообщение**

Протоколы: **HTTP, FTP, SMTP, POP3, SNMP, DNS, SMB, NCP, SAP, X.400 X.500 FTAM**

Прикладной
Представительный
Сеансовый
} Сете**не**зависимые протоколы

Транспортный — Промежуточный уровень.

Сетевой
Канальный
Физический
} Сетезависимые протоколы – зависят от технической реализации сети и используемого оборудования.

Необходимо реализовать протоколы прикладного уровня для каждого типа прикладных задач.

5.3.1 Символьная адресация

Противоречие. MAC и IP адресации неинформативны и неудобны для использования человеком, необходима символьная адресация.

Использовать механизм соответствия символьного адреса и логического адреса (сетевого уровня, например, IP).

Comp1 = 192.168.1.10

opensee.ru = 83.234.19.3

mail.ngs.ru = 195.93.186.193

Текстовый файл на локальном компьютере.

Файл: HOSTS

Активное применение до 80-х гг.

Сейчас в целях совместимости и особых ситуаций.

Специальная служба в локальной сети.

Такой принцип в сетях Microsoft и протоколе NetBIOS. За сопоставлении символьного имени NetBIOS и IP-адреса отвечает служба WINS в сети.

Используется для службы «Файлов и принтеров» в сетях Microsoft.

Специальная глобальная служба.

Принцип в службе доменных имен DNS.

За преобразование имен (разрешение имен) отвечают серверы в глобальной сети.

Начиная с 80-х годов.

Наиболее эффективный и популярный подход.

В глобальной сети:

- широковещательные запросы не применимы
- нет возможности администраторам договориться друг с другом

Необходима специальная служба

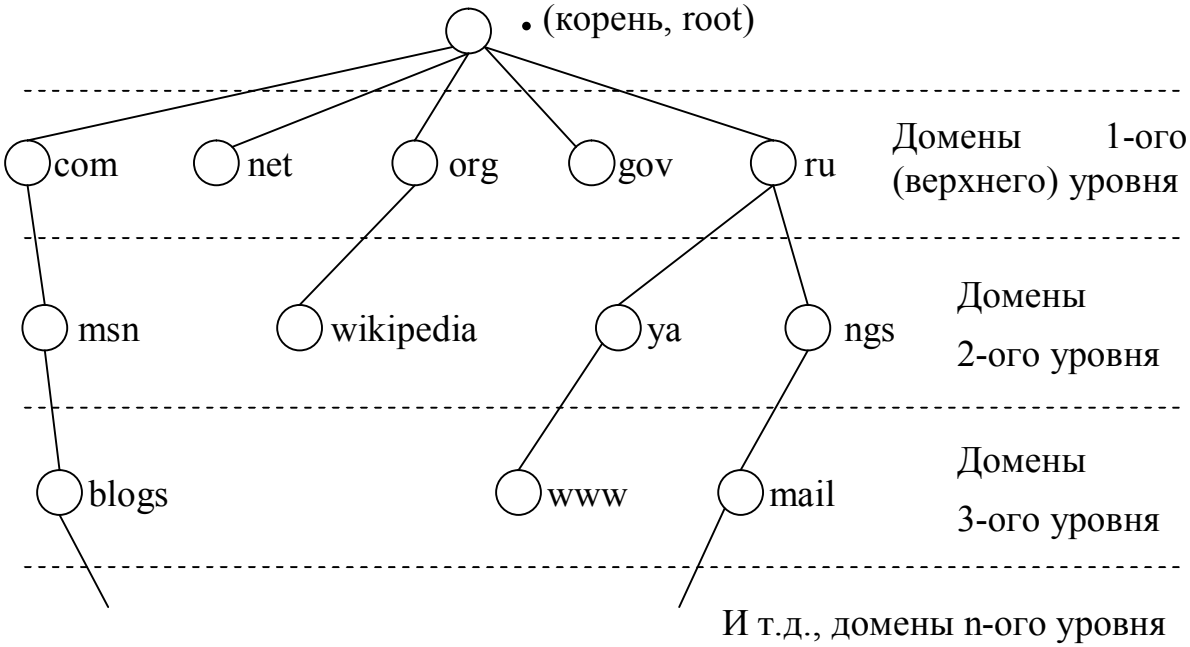
5.3.2 Служба доменных имен DNS

Служба доменных имен (Domain Name System, DNS) — это централизованная служба, основанная на распределенной базе отображений «доменное имя — IP-адрес».

Автор:
Пол Мокапетрис

С 1983 г., стандарт с 1987 г.

Порт:
53/UDP



Полное имя записывается с доменов нижнего уровня к верхнему:

mail.ngs.ru

DNS — служба **централизованная**, т.е. есть некий **центр** – это корневой сервер с именем «**.**» (точка, корень).

Управляется организацией InterNIC.

Зона ответственности — поддомен, которому делегированы права за обработку DNS-запросов данного поддомена.

За разрешение символьного имени в IP адрес в каждой зоне отвечает DNS-сервер.

Домены верхнего уровня распределятся по организационной основе или национальной принадлежности.

5.3.3 Протокол системы DNS

При обращении к узлу по доменному имени, это имя следует преобразовать в IP-адрес.

То есть, необходимо послать специальный запрос DNS-серверу.

Следовательно, IP-адрес DNS-сервера должен быть известен заранее.

Поэтому информация о DNS-сервере является отдельной настройкой ПО или ОС.

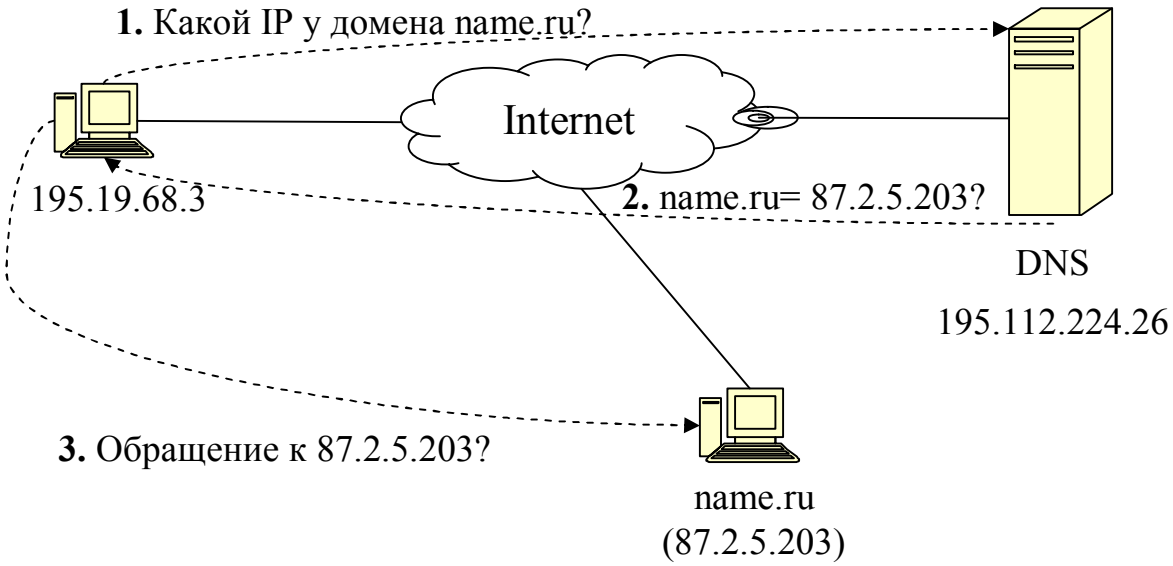
Клиент формирует запрос в стиле:

«**Какой IP у домена name.ru?**»

и посылает DNS-серверу.

Если DNS-сервер знает IP-адрес домена, то возвращает ответ с информацией об IP:

name.ru = 87.2.5.203



DNS-запросы и DNS ответы очень короткие.

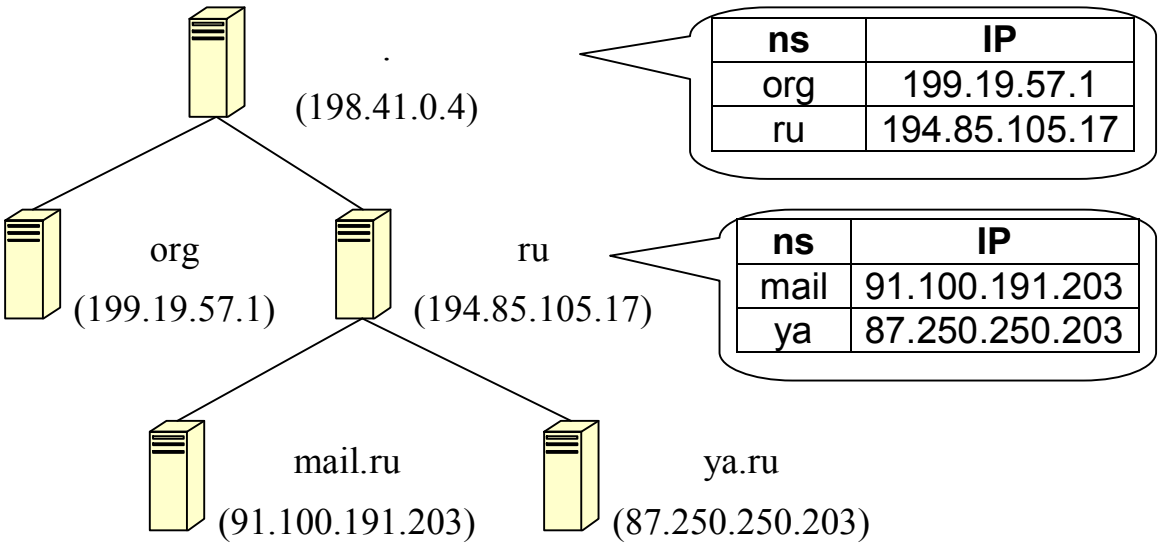
Следовательно, нерационально использовать установку и разрыв соединения с DNS-сервером.

! Поэтому разрешение запросов DNS работает через протокол UDP

Минимальные настройки узлов для взаимодействия их с узлами других сетей по имени: **IP-адрес**, **маска сети, шлюз по умолчанию, DNS-сервер**.

5.3.4 DNS-серверы, DNS-записи, DNS-алгоритмы

Как правило, DNS-серверы хранят информацию об IP-адресах своих ближайших дочерних серверах.



Типы DNS-серверов

Первичный (master) – основной сервер. Всегда один для каждой зоны.

Вторичный (slave) – вспомогательный, если выйдет из строя первичный. Обычно несколько для зоны.

Скрытый (stealth) – предназначен для обновления зоны из закрытой сети.

Информация о доменах на DNS-серверах хранится в виде **DNS-записей**. Существуют несколько типов DNS-записей.

Алгоритмы разрешения DNS-имен

Итеративная

Рекурсивная

5.3.5 Типы DNS-записей

SOA-запись (Start Of Authority) – имя первичного DNS-сервера (Primary Name Server), адрес для контактов, TTL и т.д.

```
имя    [TTL]    SOA    Данные
```

NS-запись (Name Server) – описывают DNS-серверы для данного домена.

```
домен  [TTL]   NS  имя_хоста
test.ru. 86400   NS ns3.test.ru.
test.ru. 86400   NS ns4.test.ru.
```

A-запись – устанавливает соответствие между именем хоста в домене и его IP-адресом.

```
имя_хоста [TTL]  A  IP-адрес
info.test.ru. 1d  A 194.85.61.44
```

MX-запись (Mail Exchange) – определяет почтовый сервер.

```
домен [TTL] MX приоритет сервер
test.ru. 1d MX 10 relay2.test.ru.
test.ru. 1d MX 20 relay3.test.ru.
```

CNAME-запись (Canonical Name) – присваивает хосту псевдоним. На псевдоним нельзя делать записи других типов.

```
мнемоимя  [TTL] CNAME  имя_хоста
ftp.test.ru. 1d CNAME arh.test.ru.
```

PTR-запись (Pointer) – для выполнения обратного преобразования IP-адресов в имена хостов

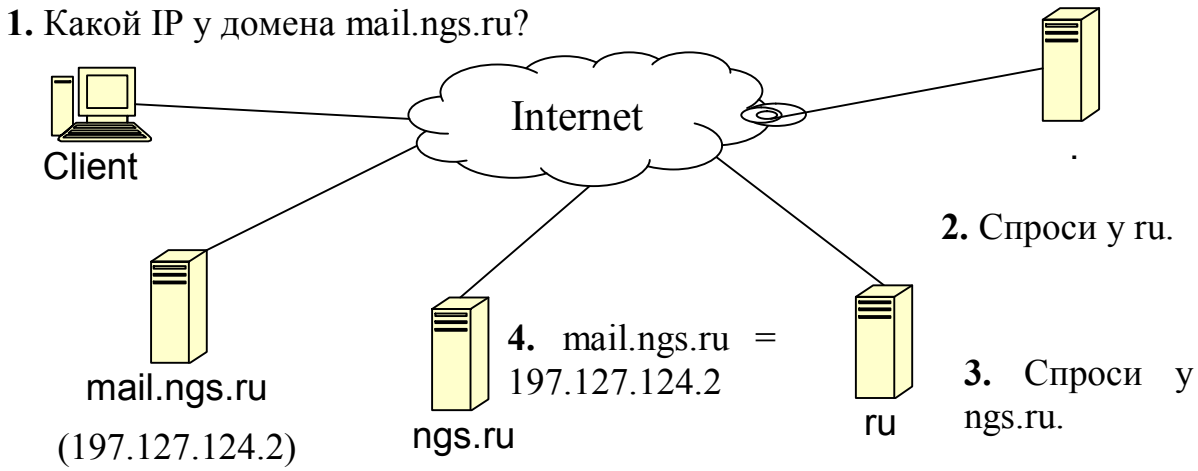
```
адрес  [TTL]  PTR  имя_хоста
42  PTR  www.my-domain-name.ru.
42.61.85.194.in-addr.arpa.   86400   PTR  www.my-domain-name.ru.
```

TXT-запись – текстовое описание доменного имени

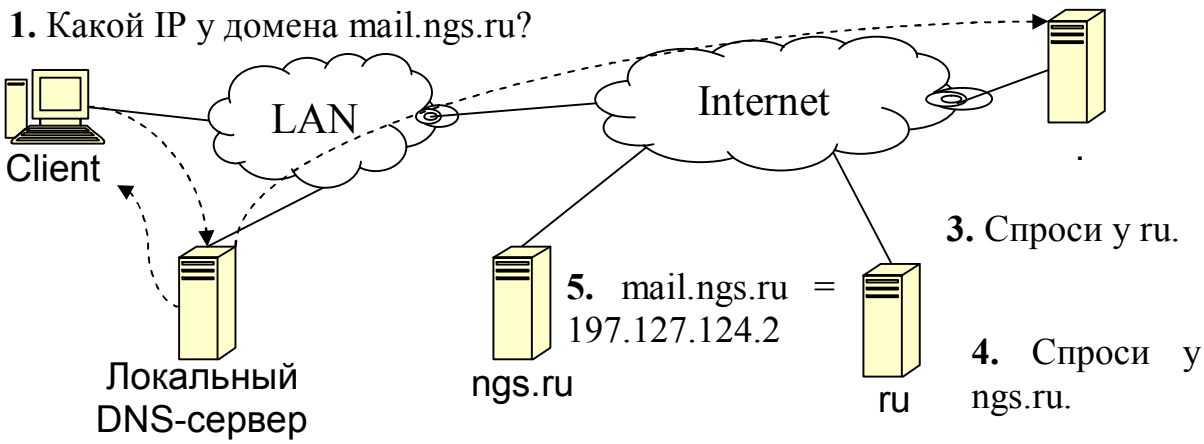
```
имя   [TTL]   TXT   текст
test.ru.  TXT "the domain..."
```

5.3.6 Алгоритмы разрешения DNS-имен

Итеративная схема – клиент делает DNS-запросы, начиная с корневого сервера. Если корневой не знает ответа, то отправляет его на уровень ниже. Так происходит до тех по пока не найдется DNS-сервер, отвечающий за данную зону.



Рукурсивная схема – клиент обращается к DNS-серверам своей локальной сети (сети провайдера). Если сервер не знает ответа, то спрашивает сам у другого сервера, при этом клиент ждет ответа. Когда сервер узнает ответ, он перешлет его клиенту.



2. Сейчас узнаю.

Какой IP у mail.ngs.ru?

6. mail.ngs.ru = 197.127.124.2

- Для клиента схема рекурсивная.
- Для локального DNS-сервера – итеративная.

5.3.7 Internationalised Domain Names

DNS допускает только ASCII в доменном имени

Национальные языки используют Unicode

Поэтому разработан Punycode (Пюникод)

Национальные домены конвертируются в Punycode и разрешаются обычным образом

Регистрация доменов на национальных языках запущена с 2003 г.

Пример конвертации:

`пример.испытание` →

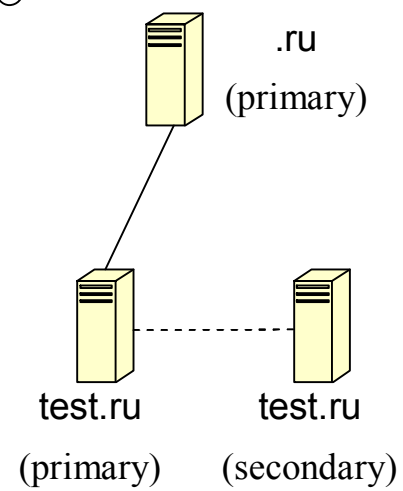
`xn--e1afmkfd.xn--80akhbyknj4f`

5.3.8 Безопасный DNS (DNSSEC)

Система DNS не защищена от искажения информации, из-за чего возможны атаки с подменой ответов DNS-серверов.

- При пересылке копии данных о домене вторичному DNS-серверу
- При кэшировании записей на DNS-сервере провайдера
- Искажение информации при обратном DNS-преобразовании

DNSSEC (Domain Name System Security Extensions) — набор спецификаций IETF, обеспечивающих безопасность информации, предоставляемой средствами DNS в IP-сетях.



1. Информация о домене (test.ru) шифруется закрытым ключом.

2. Закрытый ключ используется для подписи зоны (test.ru).

3. Цифровая подпись ключа (DS-запись) передается администратору родительской зоны (.ru)

4. Администратор зоны .ru подписывает своим закрытым ключом подпись ключа test.ru и хранит у себя.

Т.о. образуется цепочка доверия, т.к. зная открытый ключ родительской зоны, можно проверить валидность любой дочерней зоны.

5.3.9 Протокол динамического конфигурирования хостов DHCP

В больших локальных сетях неудобно использовать ручное назначение IP-адресов клиентским узлам. Некоторые узлы работают редко и резервировать для них постоянный IP-адрес не эффективно.

Протокол динамического конфигурирования хостов (Dynamic Host Configuration Protocol, DHCP) — протокол, позволяющий компьютерам автоматически получать IP-адрес и другие параметры, необходимые для работы в сети TCP/IP.

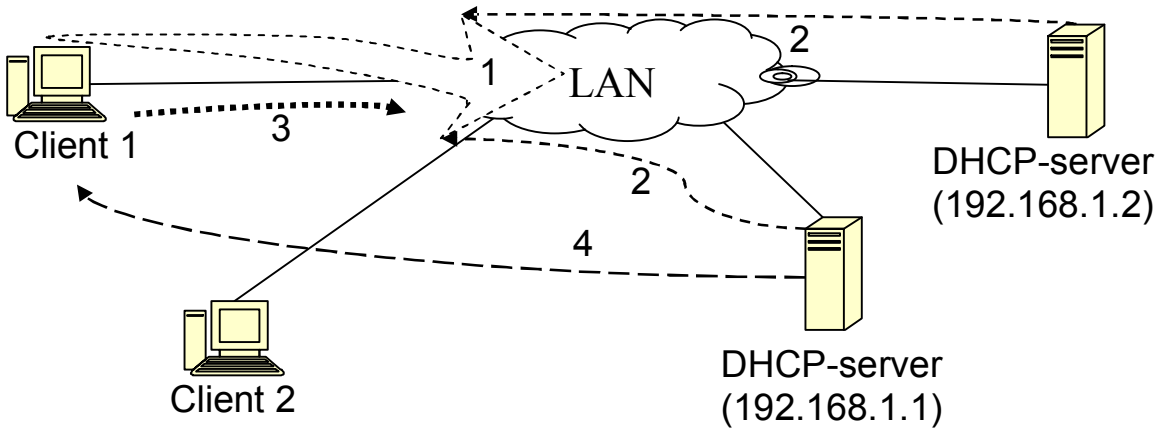
Автор: **IETF**	С 1990 г.	Порт: **67/UDP, 68/UDP**

Преимущества протокола DHCP

- Протокол позволяет создавать сети с количеством узлов больших, чем IP-адресов в данной подсети.
- При подключении узла в сеть все настройки (IP, Netmask, Gateway, DNS-server и т.д.) выдаются автоматически.
- Ни администратор, ни клиент не принимают участия в настройке IP протокола, и получении настроек.
- Один и тот же IP-могут использовать разные клиенты в разное время, так как IP-адрес выдается на время (время аренды, lease)

При запросе IP адреса у DHCP сервера IP-адрес **сервера** заранее неизвестен.	Поэтому клиент посылает широковещательный запрос (т.е. IP получателя = 255.255.255.255).
При запросе IP адреса у DHCP сервера IP-адрес **клиента** заранее неизвестен.	Поэтому в качестве обратного адреса клиент указывает адрес 0.0.0.0.
При DHCP запросах используются короткие сообщения от клиентов без IP-адресов.	Поэтому протокол DHCP использует в качестве транспорта протокол **UDP**.

5.3.10 Алгоритм работы DHCP



1. Клиент посылает широковешательный пакет с обратным адресом 0.0.0.0 с запросом (DHCPDISCOVER) на выделение IP-адреса.

Src: 0.0.0.0 | Dest: 255.255.255.255

Какой у меня IP?

2. DHCP-сервер(ы) получают запрос и высылают широковещательно свои предложения (DHCPOFFER) с IP-адресом.

Src: 192.168.1.1 | Dest: 255.255.255.255

IP=192.168.1.37?

3. Клиент получает предложения DHCP-серверов, выбирает одну конфигурацию и отправляет пакет (DHCPREQUEST) широковещательно с опцией, включающей IP-адрес, выбранного DHCP-сервера.

Src: 0.0.0.0 | Dest: 255.255.255.255

Опция DHCP 54: 192.168.1.1

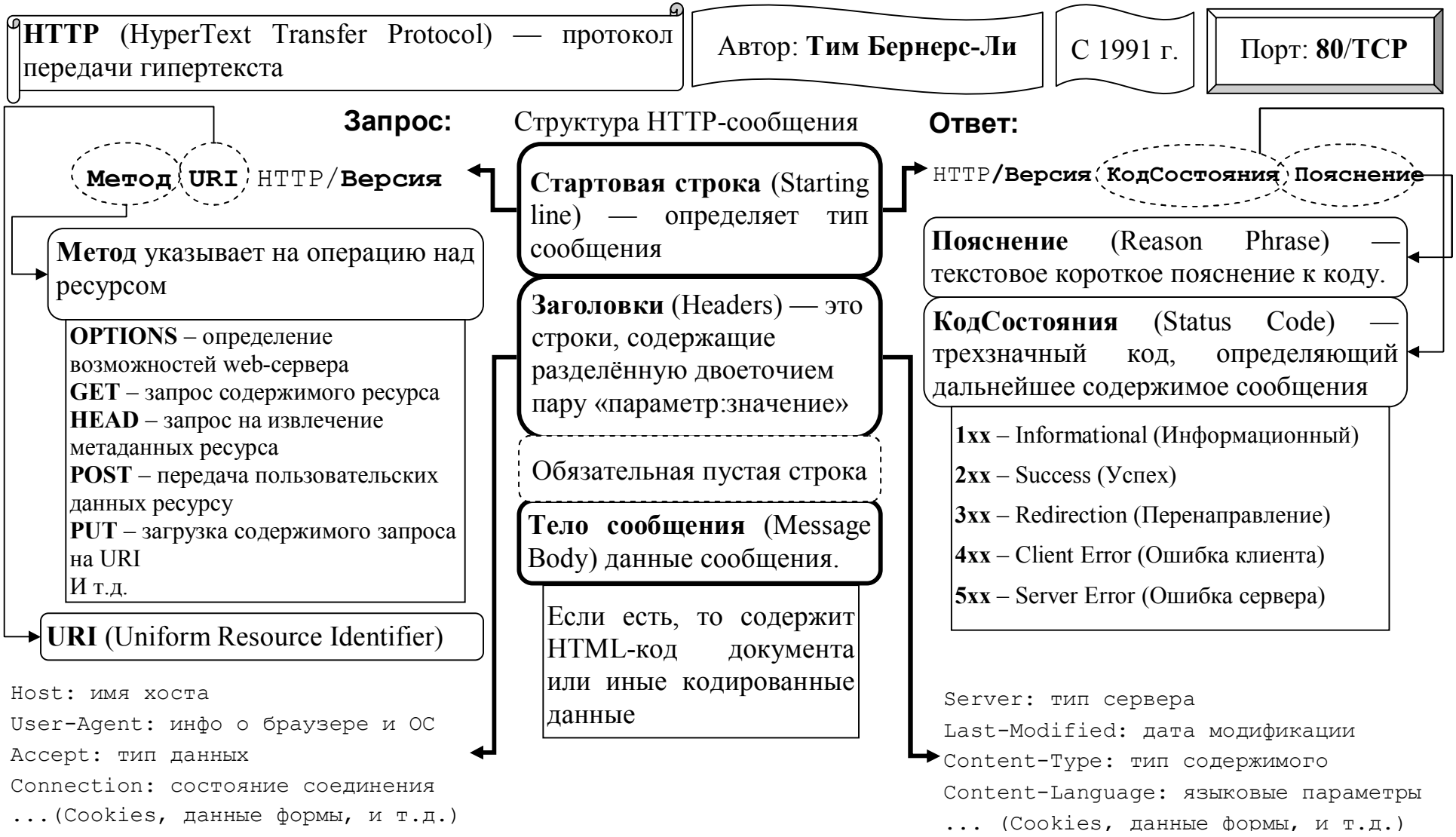
4. DHCP-сервер выделяет IP-адрес и другие настройки, подтверждает запрос пакетом (DHCPACK). Клиент, получив квитанцию, настраивает сетевой интерфейс.

Src: 192.168.1.1 | Dest: 255.255.255.255

Настройки…

Так как в сети могут быть несколько клиентских DHCP-запросов, в пакете клиент указывает опцию xid – уникальный идентификатор транзакции, по которому отличают запросы разных клиентов.

5.3.11 Протокол передачи гипертекста HTTP



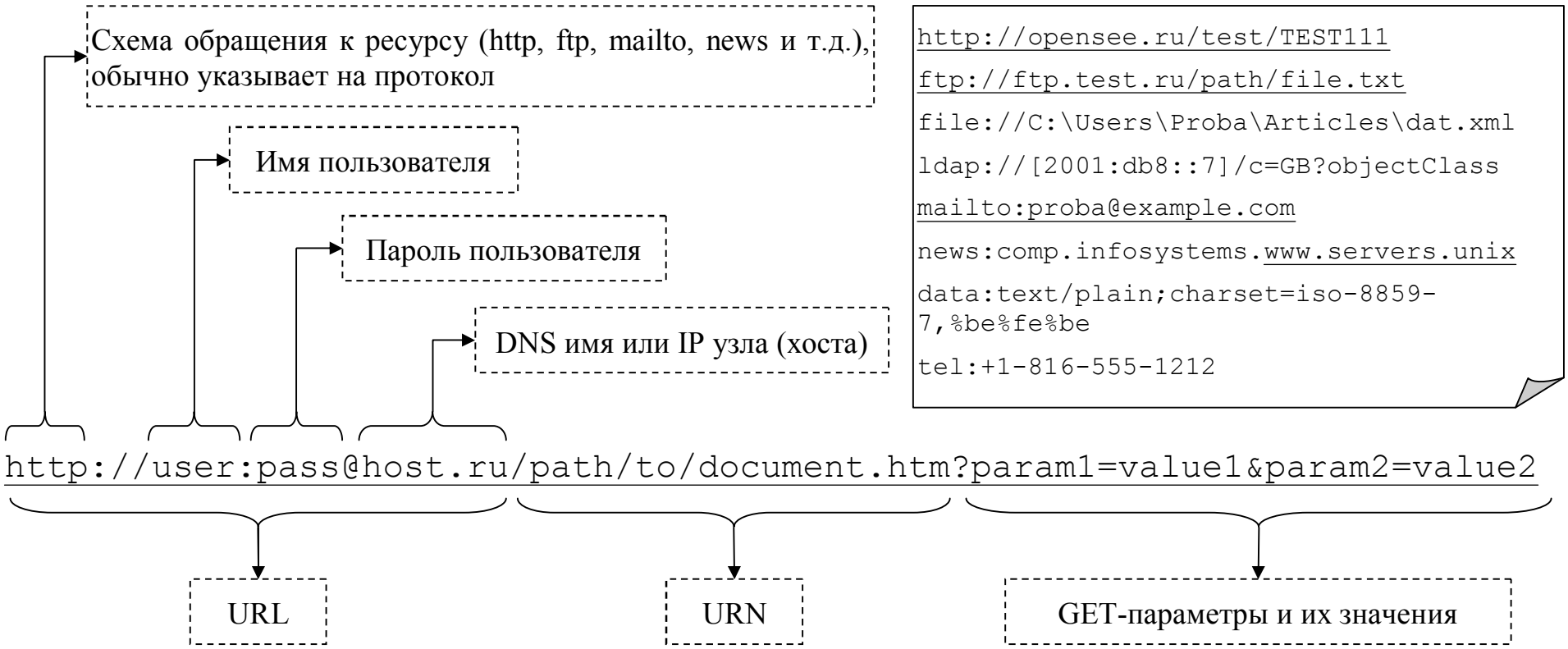
5.3.12 Структура URI

URI (Uniform Resource Identifier) – это символьная строка, идентифицирующая ресурс.

URI = URL + URN

URL (Uniform Resource Locator) – часть URI, которая определяет адрес хоста сетевого ресурса.

URN (Uniform Resource Name) – часть URI, которая определяет имя ресурса на хосте.



5.3.13 Протокол FTP

FTP (File Transfer Protocol) — протокол передачи файлов в сетях.

Автор: **Джон Постел**

С 1971 г.

Порт: **21/TCP** (для команд)
20/TCP (для данных)

Типы FTP-режимов работы

Активный режим – сервер сам подключается к клиенту для передачи данных.

Существует проблема в работе, когда клиент находится за NAT

Пассивный режим – сервер передает адрес и порт, к которому клиент должен, чтобы забрать данные.

Запрос клиента состоит из команды и параметров.

`USER` — Имя пользователя для входа на сервер
`PASS` – Пароль пользователя для входа
`TYPE` — Установить тип передачи файла (Бинарный, текстовый)
`CWD` — Сменить директорию.
`MKD` — Создать директорию.
`PWD` — Возвращает текущую директорию.
`LIST` — Возвращает список файлов директории
`PASV` — Войти в пассивный режим. Сервер вернет адрес и порт
`PORT` — Войти в активный режим.
`RETR` — Скачать файл.
`QUIT` — Отключиться

Ответ сервера имеет формат: `КодСостояния Описание`

Код состояния: XYZ

`1yz` – положительный предварительный ответ

`2yz` – успешное завершение операции

`3yz` – положительный промежуточный ответ

`4yz` – временно отрицательный ответ

`5yz` – постоянный отрицательный ответ

`x0z` – о синтаксисе

`x1z` – информация

`x2z` – о соединении

`x3z` – об авторизации и аутентификации

`x4z` – пока не используется

`x5z` – о файловой системе

`z` – разное, зависит от первых двух цифр.

Например:
200 — команда ok.
220 — сервис готов
426 — соединение закрыто
500 — ошибка синтаксиса
И др…

5.3.14 Протоколы электронной почты e-mail (SMTP, POP3)

Электронная почта использует для работы два протокола.

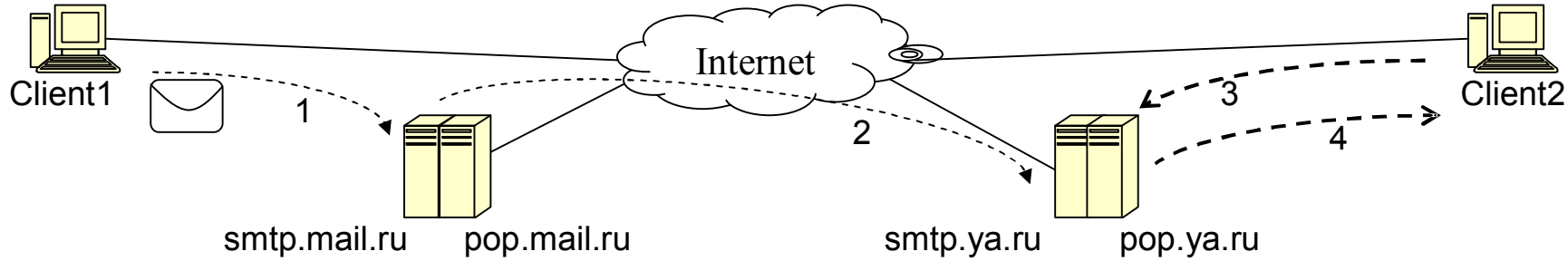
SMTP (Simple Mail Transfer Protocol) — простой протокол передачи почты от пользователей к серверам и между серверами к получателю.

POP3 (Post Office Protocol v.3) — протокол почтового отделения, используется для получения почты.

Порт: **25/TCP**

Протоколы начали развиваться с 1982 г.

Порт: **110/TCP**



1. Клиент по протоколу SMTP подключается к SMTP-серверу (smtp.mail.ru) и отправляет почтовое сообщение.

2. SMTP-сервер анализирует DNS-имя в e-mail адресе получателя. По MX-записи (или A-записи) в DNS находит SMTP-сервер получателя и отправляет ему письмо клиента.

3. Клиент-получатель подключается к своему POP-серверу и авторизуется.

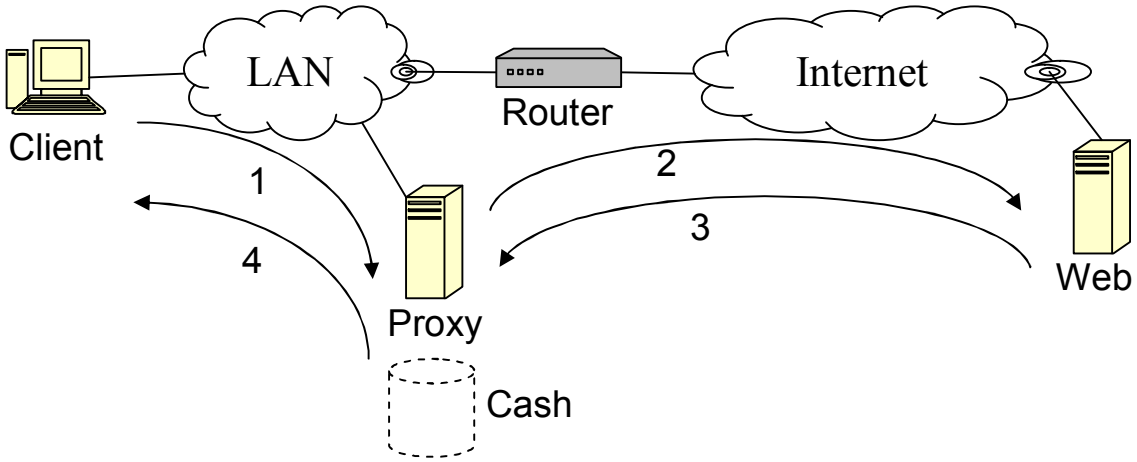
4. POP-сервер передает информацию о состоянии почтового ящика и передает ему почту.

5.3.15 Проксирование запросов. Прокси-сервер (proxy)

Противоречие. В сети необходимо контролировать трафик не только по протоколам и портам, но и по контенту. Например, контролировать загрузку ресурсов по типам (jpg, mp3, zip и т.д.) или по доменному имени и url-адресу.

Противоречие. В сети возможны частые обращения к одним и тем же ресурсам (например, страницы web), поэтому нерационально загружать их повторно, если они еще не устарели.

Прокси-сервер (proxy-server) — служба в сети, позволяющая клиентам выполнять косвенные запросы к другим сетевым службам.



1. Клиент отправляет запрос proxy-серверу. Proxy-сервер проверяет, есть ли в кэше информация и не устарела ли она?

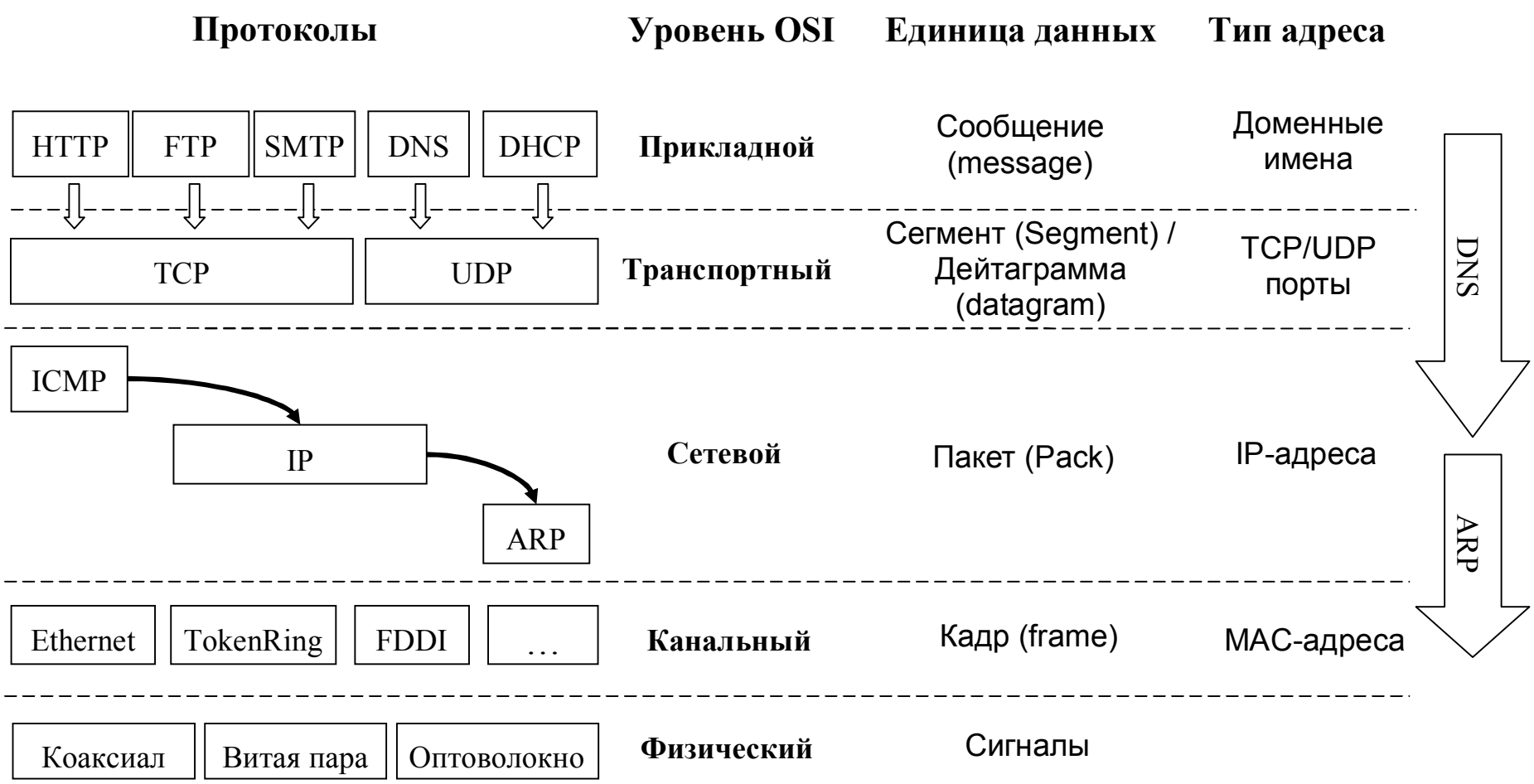
4. Proxy-сервер отправляет информацию.

2. Proxy-сервер сам запрашивает информацию у других служб.
3. Он получает ее и кэширует

Прокси

- образует два соединения: клиент-прокси и прокси-служба.
- производит более высокоуровневую обработку данных: по содержимому и по URL адресу.
- имеет кэш для повышения эффективности.
- осуществляет доступ через одну точку в сети

5.3.16 Межсетевое взаимодействие в TCP/IP: схема и протоколы



5.3.17 Стандартные протоколы коммуникационных стеков

Таблица 2 – Стандартные стеки коммуникационных протоколов

<table>
<tr><th>Модель OSI</th><th>IBM/Microsoft</th><th>TCP/IP</th><th>Novell</th><th>Стек OSI</th></tr>
<tr><td>Прикладной</td><td rowspan="2">SMB</td><td rowspan="2">Telnet, HTTP, FTP, SMTP, POP3, SNMP / DNS, RIP, OSPF</td><td rowspan="3">NCP, SAP</td><td>X.400 X.500 FTAM</td></tr>
<tr><td>Представительный</td><td>Представительный протокол OSI</td></tr>
<tr><td>Сеансовый</td><td rowspan="2">NetBIOS</td><td rowspan="2">TCP/UDP</td><td>Сеансовый протокол OSI</td></tr>
<tr><td>Транспортный</td><td>SPX</td><td>Транспортный протокол OSI</td></tr>
<tr><td>Сетевой</td><td>—</td><td>IP</td><td>IPX, NLSP</td><td>ES-ES IS-IS</td></tr>
<tr><td>Канальный</td><td colspan="4">Ethernet (IEEE 802.3), Token Ring (IEEE 802.5), FDDI, Fast Ethernet, SLIP, 100VG-Anyl_AN, X.25, ATM, LAP-B, LAP-D, PPP</td></tr>
<tr><td>Физический</td><td colspan="4">Коаксиал, витая пара, оптоволокно, радиоволны</td></tr>
</table>

Стек OSI — международный, независимый от производителей стандарт. Его поддерживает правительство США.

Стек TCP/IP был разработан по инициативе Министерства обороны США в 80-е гг. для связи экспериментальной сети ARPAnet с другими сетями

Стек IPX/SPX – стек является оригинальным стеком протоколов фирмы Novell, разработанным для сетевой операционной системы NetWare еще в начале 80-х годов.

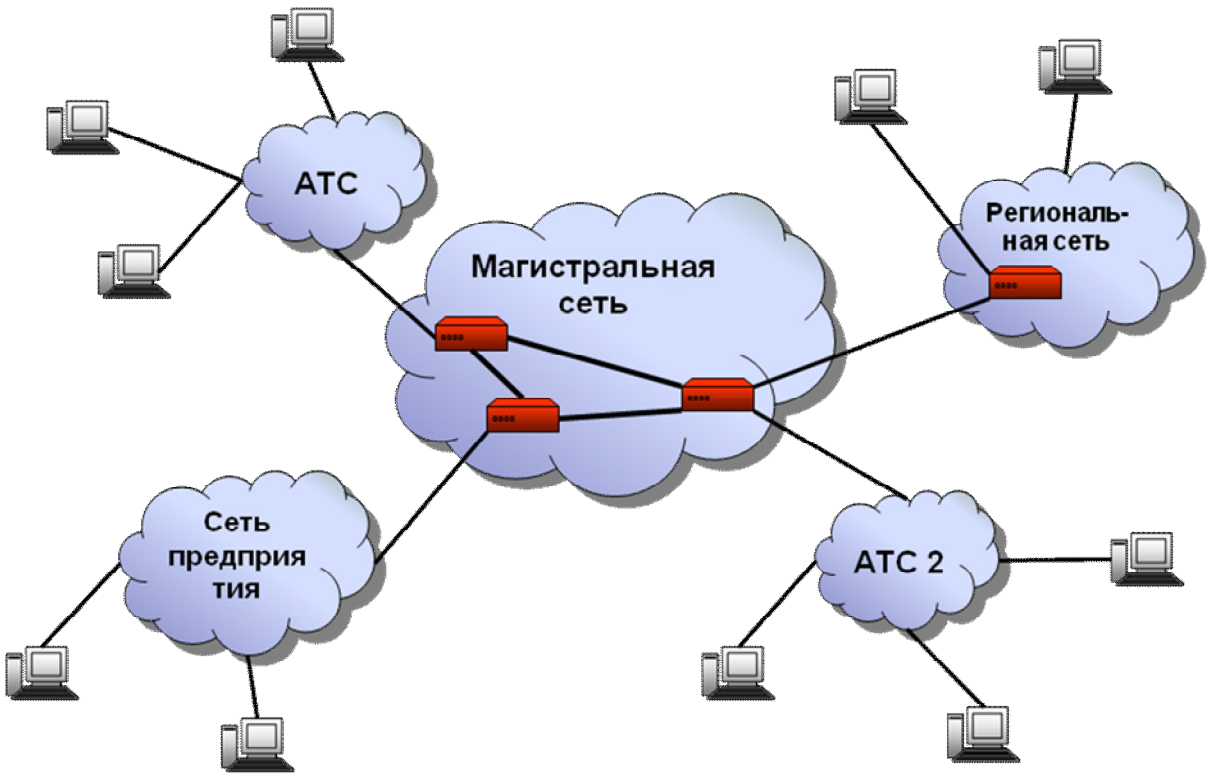
Стек NetBIOS/SMB – стек разработан IBM и Microsoft. Протокол NetBIOS (Network Basic Input/Output System) появился в 1984 году как сетевое расширение стандартных функций базовой системы ввода/вывода (BIOS) IBM PC.

6 Глобальные сети и абонентский доступ

Сети WAN (Wide area network) созданы для предоставления сервисов большому количеству конечных абонентов, разбросанных по большим

Основные абоненты глобальных сетей — локальные сети.

Разделение между глобальными и локальными сетями достаточно условное. В настоящее время наблюдается конвергенция сетей (взаимопроникновение).



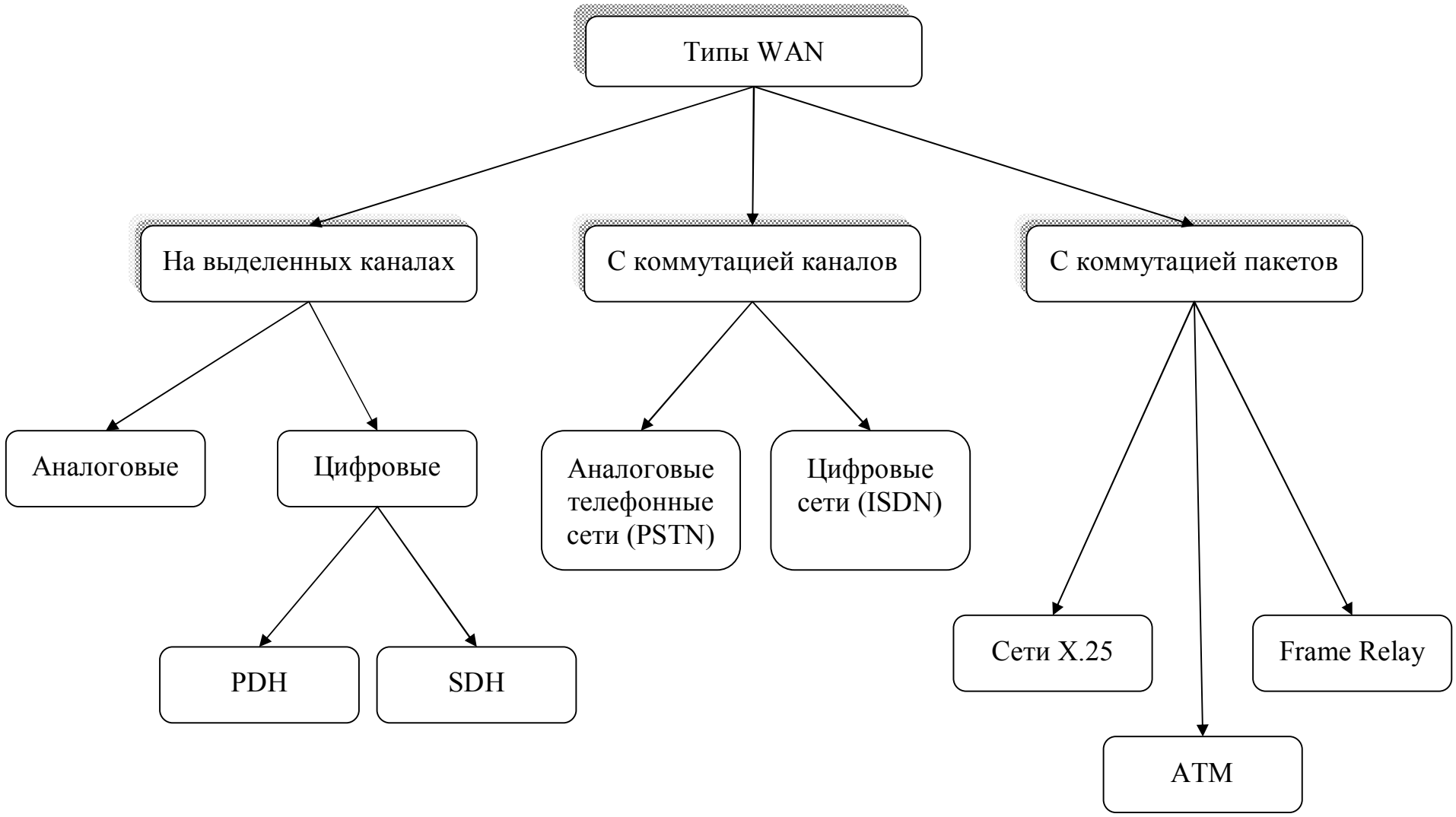
Компоненты WAN

Магистральная сеть – объединяющая отдельные сети доступа, обеспечивающая транзит трафика между ними по высокоскоростным каналам.

Сеть доступа – сеть, концентрирующая информационные потоки от многочисленных каналов связи клиентов в небольшом количестве узлов магистральной сети.

Информационные центры – реализуют информационные услуги сети (веб-порталы, файловые серверы, серверы электронной почты…).

6.1 Типы глобальных сетей



6.1.1 Сети на выделенных каналах

Выделенный канал – канал, с фиксированной полосой пропускания или фиксированной пропускной способностью, постоянно соединяющий двух абонентов.

Приобретаются у телекоммуникационных или телефонных компаний.

Абонентами могут быть отдельные устройства или сети.

PDH – Технология плезиохронной цифровой иерархии (Plesiochronic Digital Hierarchy)

- Разработана компанией AT&T в 60-х.
- Используется аппаратура T1 (в Европе E1). Она мультиплексирует каналы, передает данные и коммутирует абонентов.
- Голос цифруется с частотой 8кГц по PCM (Pulse Code Modulation).
- T1 может иметь до 24 аб, со скорость до 64Кбит/с каждый. 4 канала T1 образуют канал T2 (6.3Мбит/с). 7 каналов T2 образуют канал T3 (44.7Мбит/с). Может передавать голос, телеизображение, факс, компьютерный трафик.

SDH/SONET – Синхронная цифровая иерархия (Synchronous Digital Hierarchy) / Синхронные оптические сети(Synchronous Optical NETs).

- SONET разработана Bellcore в 1984г, затем стандартизирована комитетом T1 ANSI.
- Цель стандарта объединить каналы T1-T3 и E1-E3 в высокоскоростные магистральные линии на оптоволокне. В результате появился SDH (спецификации G.707 – G.709).
- Имеет свой стек протоколов.

6.1.2 Стек и протоколы канального уровня SDH/SONET

Канальный

Уровень тракта (path) – отвечает за доставку данных между двумя пользователями.

Уровень линии (line) – отвечает за передачу между двумя мультиплексорами.

Уровень секций (section) – поддерживает физическую целостность сети.

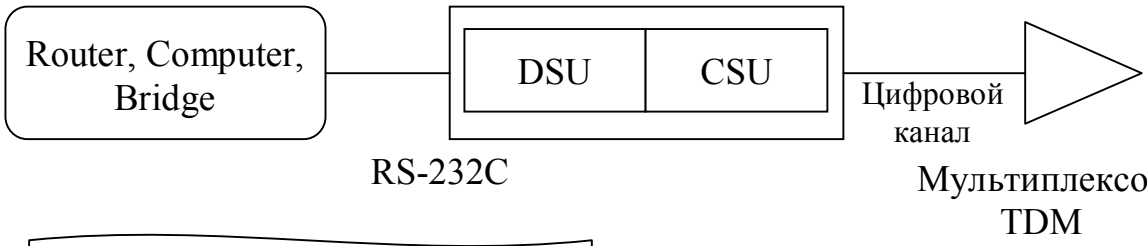
Физический

Оптическая модуляция + NRZ

Связь компьютера или маршрутизатора с цифровой линией осуществляется через пару устройств, часто в одном корпусе:

обслуживающие данные (DSU, Data Service Unit) и

обслуживающие канал (CSU, Channel Service Unit).



Протоколы канального уровня:

SLIP (Serial Line IP) – протокол выполняет единственную функцию – в потоке бит распознает начало и конец IP-пакета. Первый стандарт, позволяющий устройствам, соединиться последовательной линией связи и работать по протоколам TCP/IP. Создан в 80-е годы, в 1984 встроен в Unix.

PPP (Point to Point Protocol) – разработан для передачи кадров информации по последовательным глобальным каналам связи взамен SLIP. Поддерживает переговорную процедуру, в результате которой передаются параметры (качество линии, протокол аутентификации и т.д.). Поддерживает несколько сетевых протоколов (IP, IPX, OSI)

6.1.3 Глобальные сети на основе коммутации каналов

Аналоговые телефонные сети (PSTN, Public Switched Telephone Network) – средняя пропускная способность 9,6кбит/с, максимальная 56Кбит/с (V.90). Для соединения «пользователь — сеть».

Цифровые сети с интеграцией услуг ISDN (Integrated Services Digital Network) – первоначально были созданы для передачи цифрового голоса (1959 г), затем, добавлена возможность передавать компьютерные данные, факсимильную связь, телетекст, голосовую почту.

6.1.4 Глобальные сети на основе коммутации пакетов

Сети X.25 — семейство протоколов канального уровня, разработанных для ненадежных линий.

Сетевой	**X.25/3**
Канальный	**LAP-B**
Физический	**X.21** для оборудования DSU/CSU

Frame Relay — семейство протоколов канального уровня, разработаны в начале 1990-х для замены X.25 для быстрых надёжных каналов связи.

Сетевой	**Q.933**
Канальный	**LAP-F/LAP-D (Q.921)**
Физический	**I.430/431**

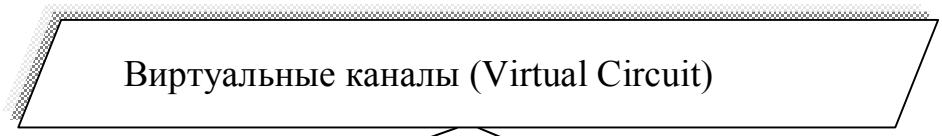
ATM (Asynchronous Transfer Mode — асинхронный способ передачи данных) — сетевая высокопроизводительная технология коммутации и мультиплексирования.

6.1.5 Технология ATM

ATM (Asynchronous Transfer Mode) — асинхронный способ передачи данных.

Основная цель – совместная передача данных, видео и голоса

Основана на передаче данных в виде ячеек (cell) фиксированного размера (53 байта) через виртуальные каналы.



PVC (Permanent Virtual Circuit) постоянный виртуальный канал.

Виртуальный канал прокладывается **администратором**.

Используются только таблицы коммутации.

Таблица маршрутизации не используются, так как идентификаторы каналов выбираются администратором.

Протоколы: FrameRelay, ATM

SVC (Switched Virtual Circuit) коммутируемый виртуальный канал.

Виртуальный канал прокладывается **автоматически** оборудованием.

При прокладывании канала используются таблицы маршрутизации,

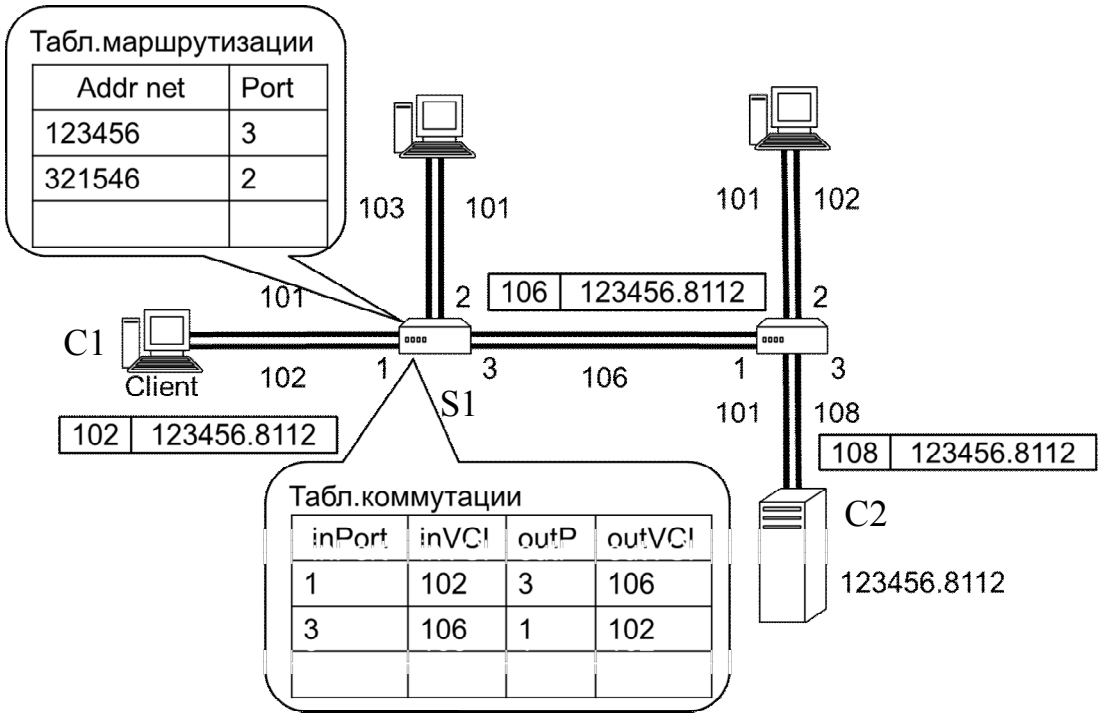
при передаче данных по каналу – таблицы коммутации.

Протоколы: ATM, позже FrameRelay.

VPI (virtual path identificator) — идентификатор виртуального пути (номер канала), используется в PVC.

VCI (virtual connect identificator) — идентификатор виртуального соединения (номер соединения), используется в SVC.

6.1.6 Установление виртуального канала



1. Установление виртуального канала (C1 — C2). Узел-инициатор (C1) генерирует пакет-запрос.

Начальное значение VCI (локальный идентификатор виртуального канала)

102 | 123456.8112

Многоразрядный адрес узла (до 16 байт)

2. В коммутаторе (S1) по таблице маршрутизации определяется выходной порт пакета (3).

3. На коммутаторе (S1) генерирует номер виртуального канала (VCI) для данного участка сети (106)

4. Одновременно коммутатор создает таблицу коммутации для продвижения пакетов в прямом и обратном направлении.

5. Аналогичная процедура (пп.2-4) установления соединения продолжается на других коммутаторах.

6. После установления соединения конечные узлы пользуются проложенным виртуальным каналом.

- При прокладке SVC канала используется **маршрутизация** по глобальным адресам.
- После установки соединения сеть работает с использованием локальных меток (VCI) на основе таблиц **коммутации**.

6.1.7 Протокол PPP (Point-to-point protocol)

80-е годы XX в. Отсутствие стандартного протокола канального уровня для передачи дейтаграмм через последовательные линии связи.

Как следствие, малое число каналов связи TCP/IP с непосредственным соединением

PPP — протокол обеспечивает метод передачи дейтаграмм через последовательные каналы связи с непосредственным соединением типа «точка-точка» (point-to-point). Первоначально передача IP-пакетов. Сейчас поддерживается инкапсуляция многих других.

Компоненты:

- Метод инкапсуляции дейтограмм при передаче по последовательным коммуникационным каналам
- Протокол LCP (Link Control Protocol) для установления, конфигурирования и тестирования информационных каналов
- Набор протоколов NCP (Network Control Protocols), для установки и конфигурирования различных протоколов сетевого уровня.

Решаемые задачи:

Присвоение и управление адресами IP	Асинхронное и синхронное формирование пакетов	Обнаружение ошибок

Конфигурация канала связи	Проверка качества канала	Проверка качества канала	Согласование способа сжатия

Общий алгоритм:

1) Инициатор отправляет пакеты LCP для выбора конфигурации и проверки канала передачи данных.

2) Инициатор отправляет пакеты NCP, чтобы выбрать и определить конфигурацию одного или нескольких протоколов сетевого уровня.

3) Отправка данных через созданный канал.

4) Разрыв соединения через пакеты LCP или NCP, или внешнее событие.

Преимущества:

— одновременная работа по различным сетевым протоколам

— проверка целостности данных по контрольной сумме

— динамический обмен адресами IP;

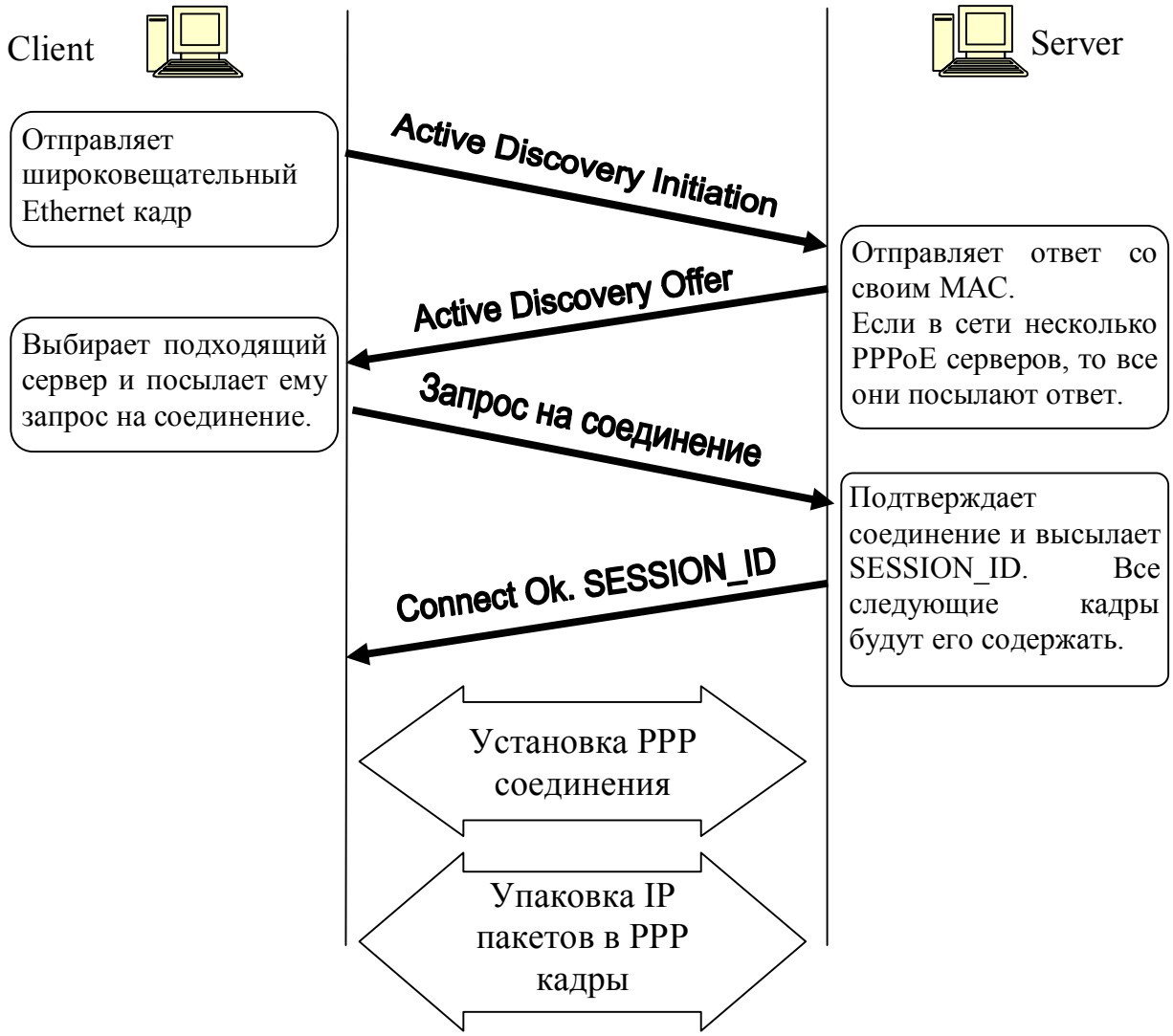
— сжатие заголовков IP- и TCP-пакетов.

6.1.8 Протокол PPPoE (Point-to-point protocol over Ethernet)

PPPoE —протокол передачи кадров PPP через Ethernet.
Это туннелирующий протокол (tunneling protocol), который позволяет передавать IP-пакеты, или другие, которые инкапсулируются в PPP, через соединения Ethernet, но с программными возможностями PPP соединений.

Установка соединения через PPPoE.

Пусть существует Ethernet-среда:



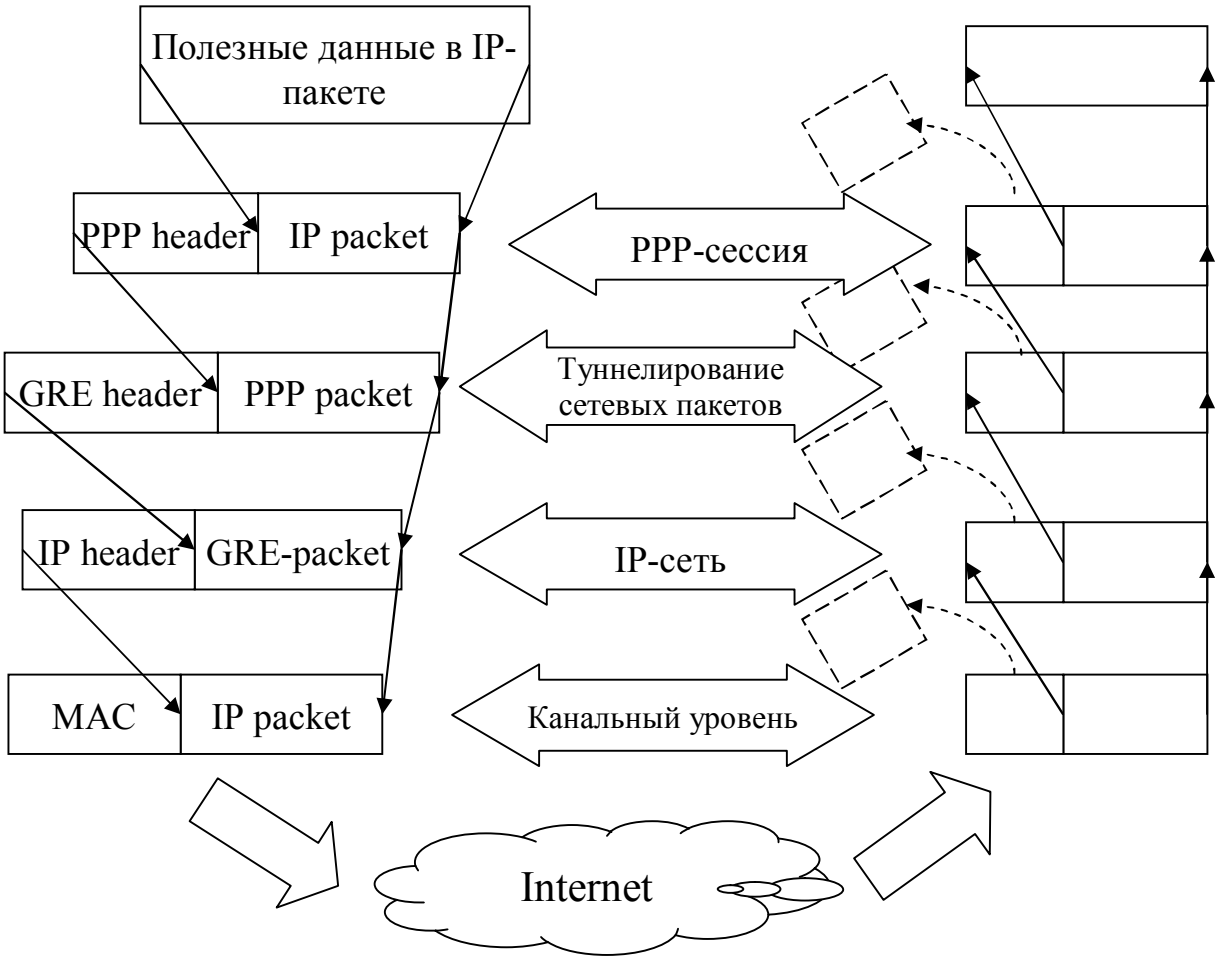
Виртуальный канал между клиентом и сервером идентифицируется по ID-сессии (SESSION_ID) и MAC-адресам клиента и сервера.

6.1.9 Протокол PPTP

PPTP (Point-to-point tunneling protocol) — туннельный протокол типа точка-точка, позволяющий компьютеру устанавливать защищённое соединение с сервером за счёт создания специального туннеля в стандартной, незащищённой, сети.

PPTP инкапсулирует кадры PPP в IP-пакеты для передачи по глобальной IP-сети.

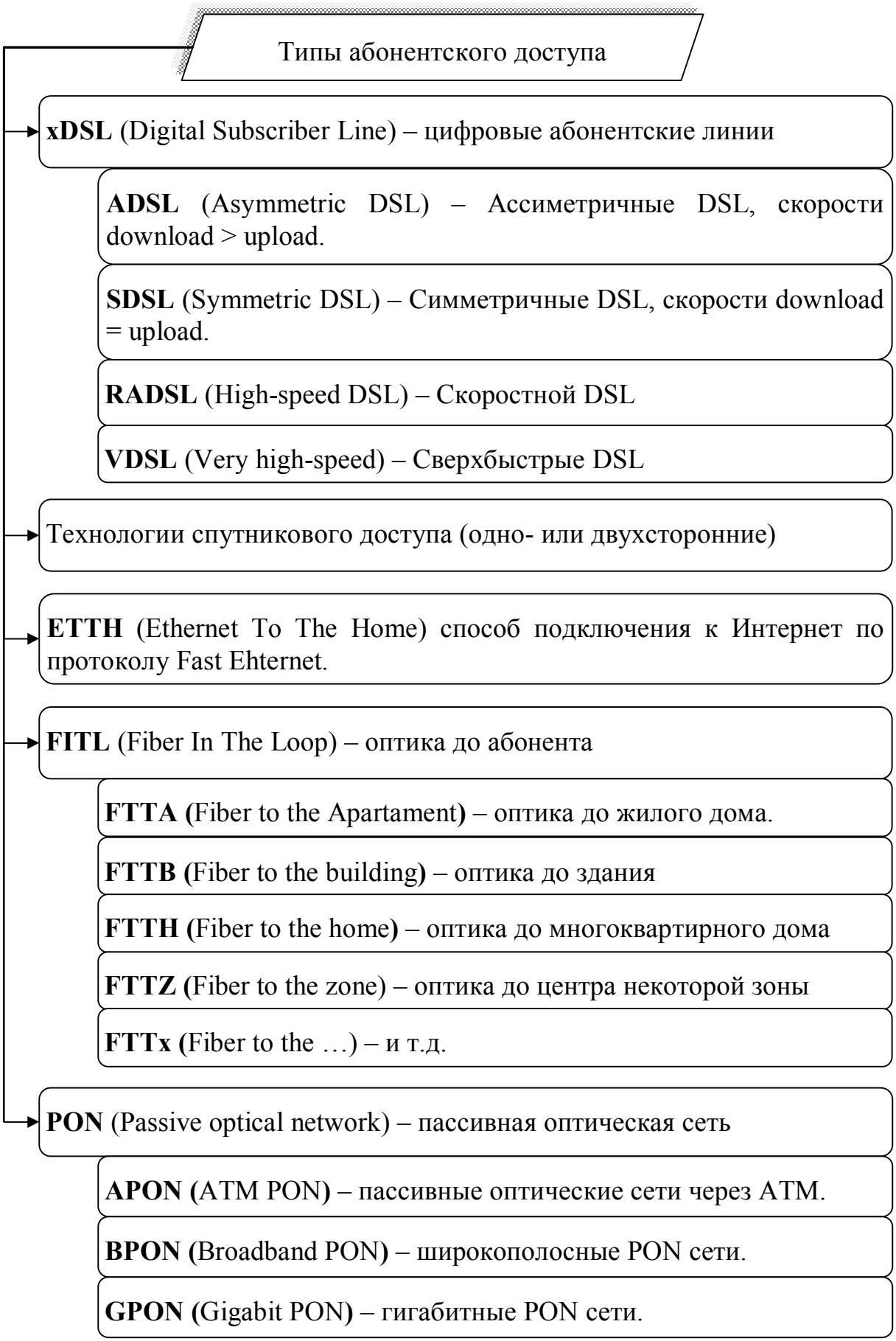
PPTP может также использоваться для организации туннеля между двумя локальными сетями.



GRE (Generic Routing Encapsulation) — протокол туннелирования сетевых пакетов от Cisco Systems. Его основное назначение — инкапсуляция пакетов сетевого уровня сетевой модели OSI в IP пакеты.

PPTP-трафик может быть зашифрован с помощью MPPE. Для аутентификации клиентов могут использоваться MS-CHAPv2 и EAP-TLS.

6.2 Технологии абонентского доступа

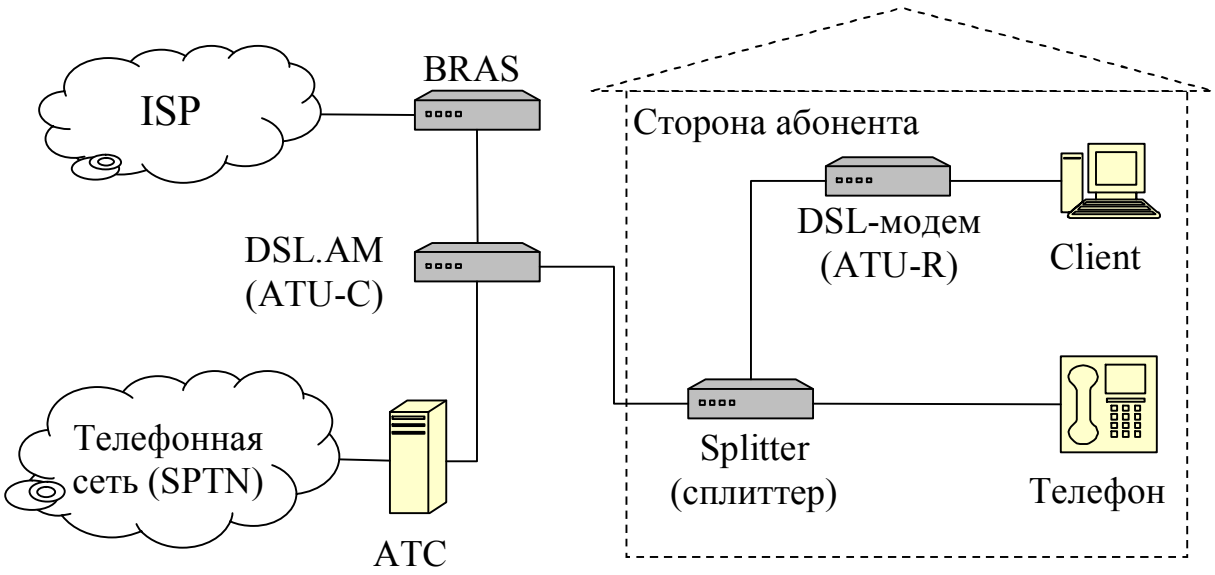


6.2.1 Цифровые абонентские линии xDSL

Технология xDSL рассчитана на получение высокоскоростного доступа к Интернет с помощью 2х-проводного телефонного кабеля и модема через ближайшую АТС

Компания Bellcore	С 1987 г.	Полоса частот: 1 МГц (новые стандарты до 2 МГц)

Цифровое кодирование: **2B1Q, PAM16, PAM32, др.**	Модуляция: **QAM, OFDM, CAP, DMT**



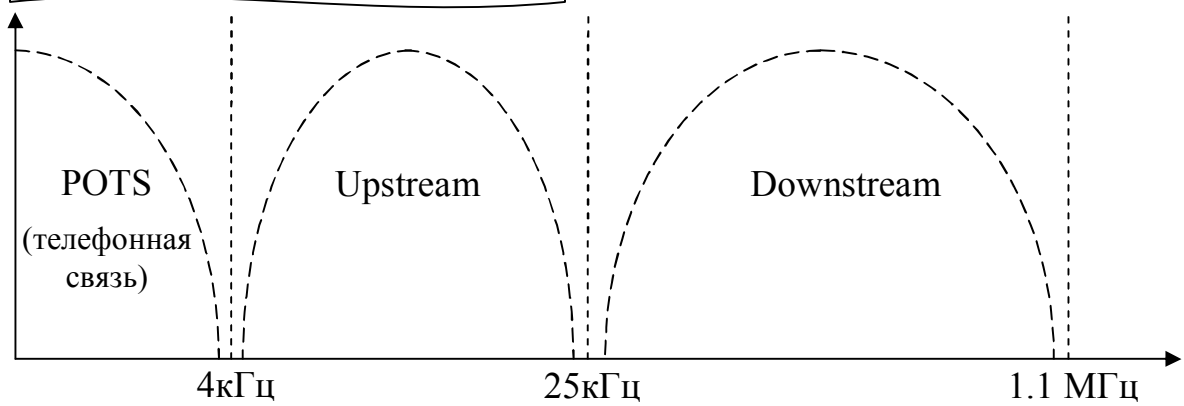
DSL.AM – DSL Access Multiplexer

ATU-C – ADSL Transceiver Unit станционный

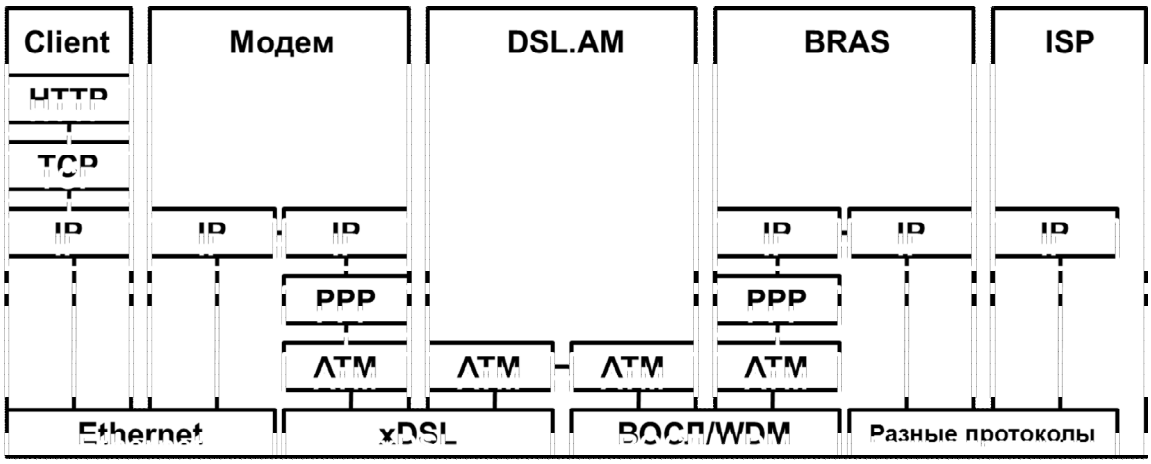
ATU-R – ADSL Transceiver Unit удаленный

Максимальное расстояние до абонента: 3,5-5,5км при сечении 0,5мм

Распределение частот в ADSL:

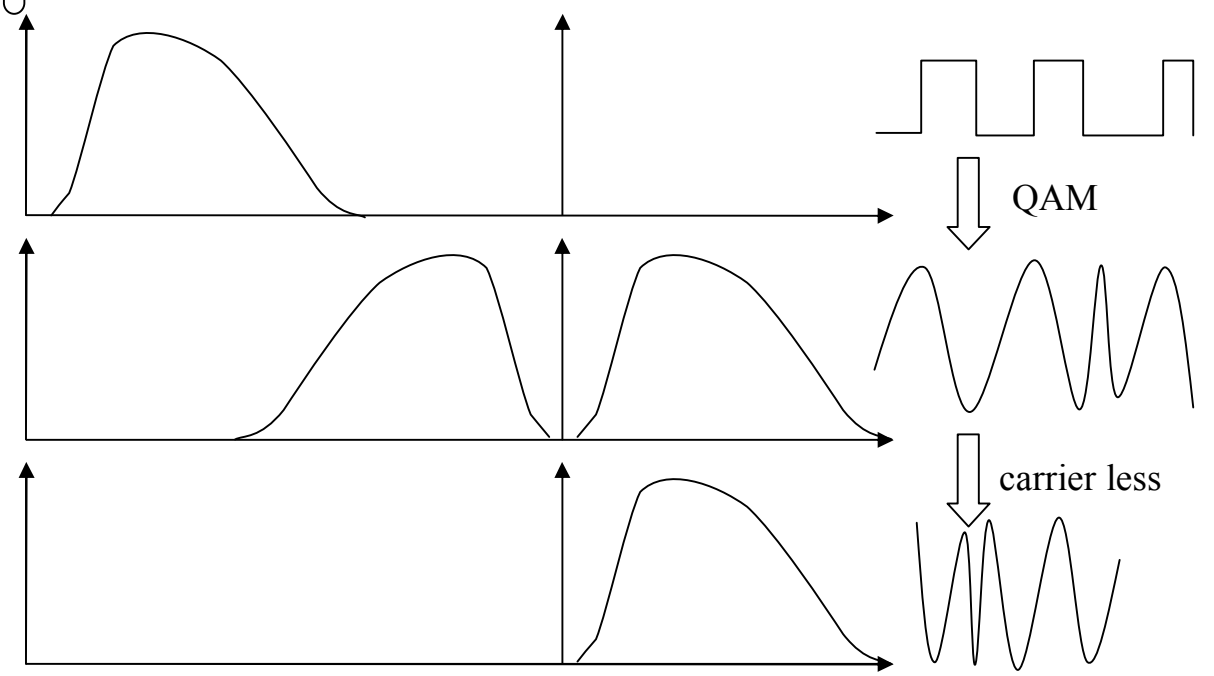


6.2.2 Межсетевое взаимодействие через xDSL



6.2.3 Типы модуляции CAP

CAP (carrier less amplitude modulation/phase modulation) Амплитудно-фазовая модуляция с подавлением несущей.



\+
- Относительная простота
- Спектральная эффективность
- Передача сигнала на большие расстояния

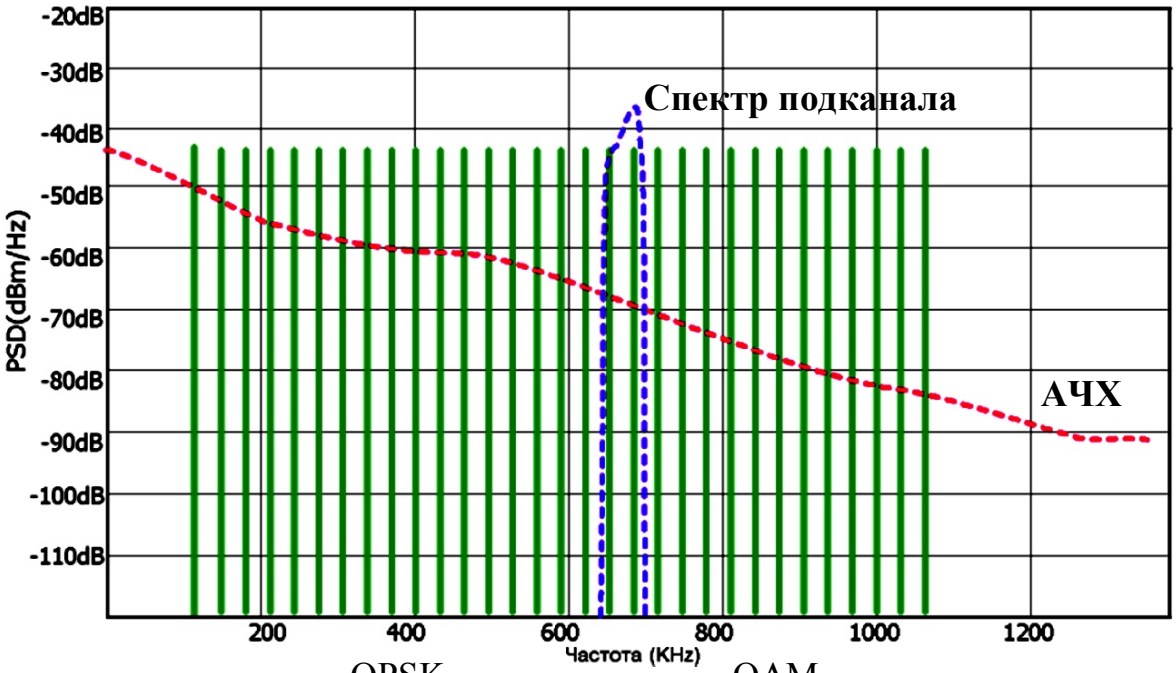
—
- Отсутствие стандартизирующего документа
- Низкая гибкость лицензионной политики владельца патента GlobeSpan

6.2.4 Типы модуляции DMT

DMT (discrete multi ton) – многочастотный алгоритм

Компания Amati Communication

С 1990-х гг., в 1993 принят ANSI



QPSK QAM

В соответствии с диапазоном адаптивно выбирается тип модуляции.
Используется QPSK (где больше помех) и QAM (где меньше помех).

\+
- Высокая скорость передачи сигнала
- Высокая надежность
- Динамическая адаптация к помехам
- Стандартизация в ANSI

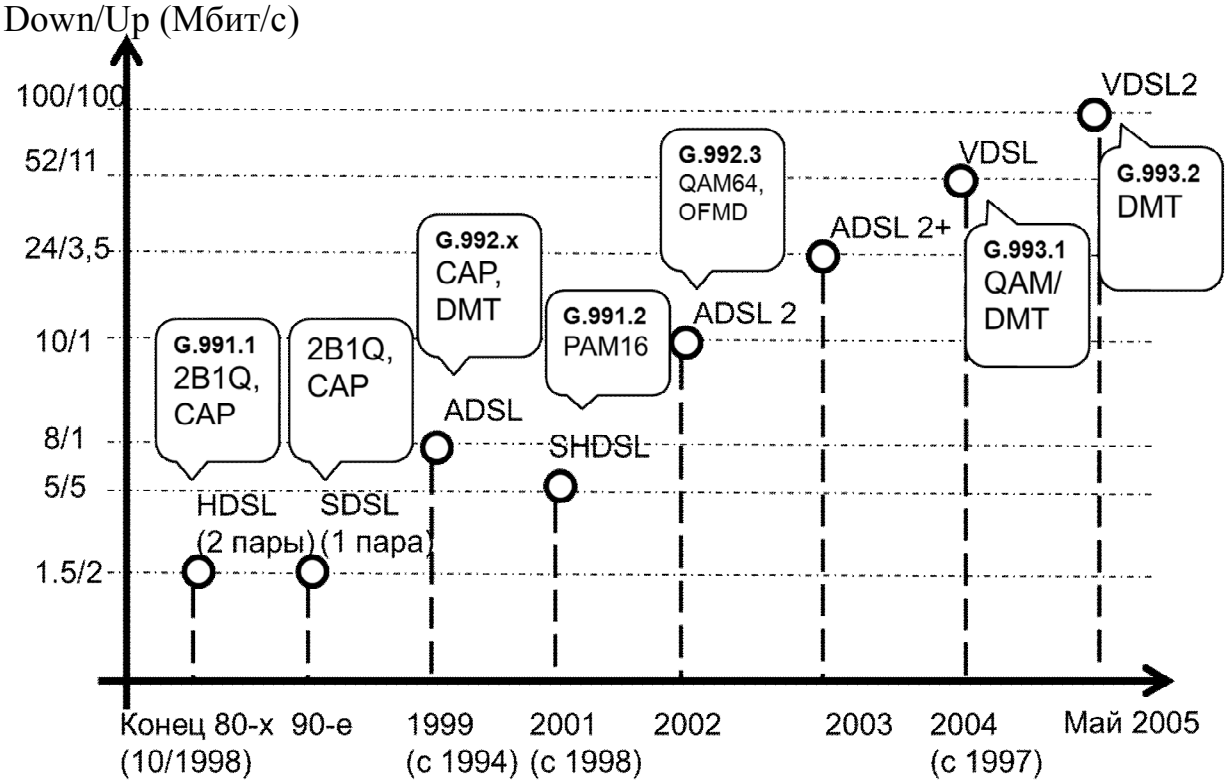
—
- Громоздкость реализации
- Сложность аппаратной реализации

6.2.5 Стандарты xDSL (ITU G.992.x)

Основные скоростные характеристики:

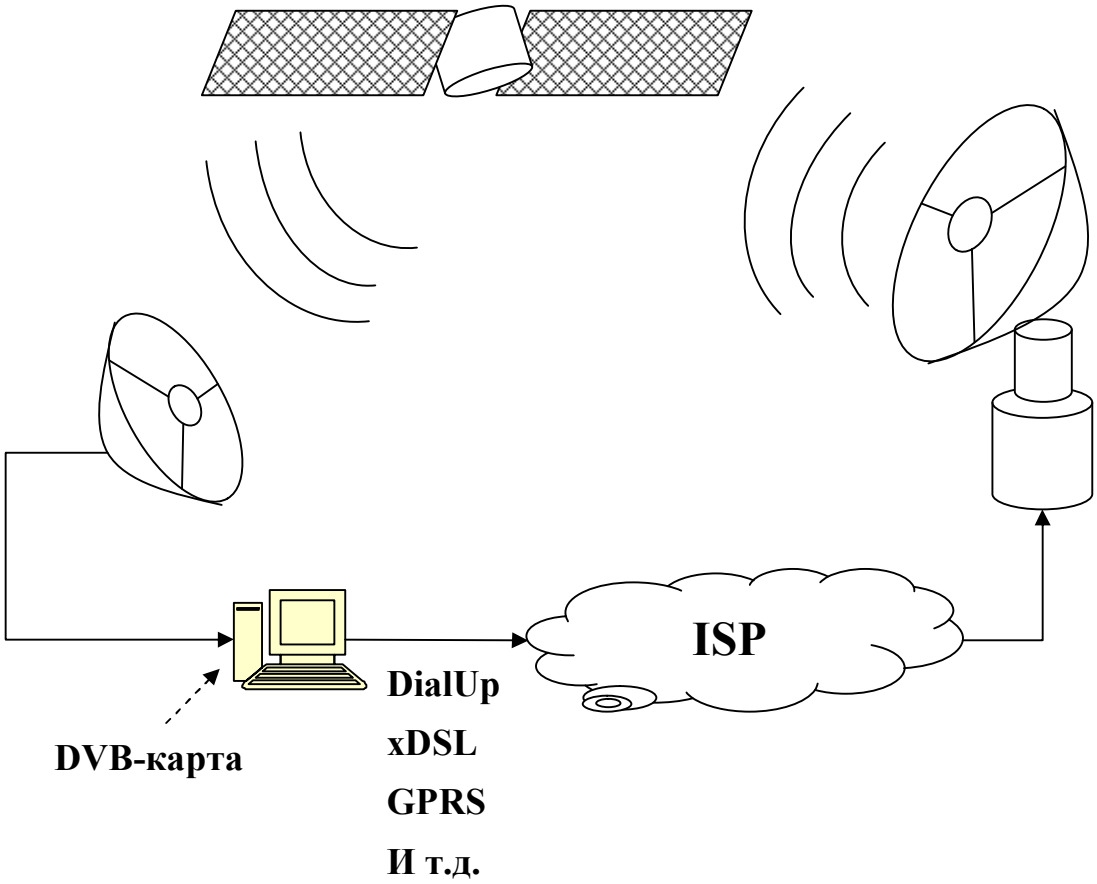
	Download	Upload (Мбит/с)
• **ADSL**		
– ANSI T1.413 Issue 2	8	1
– ITU G.992.1 (G.DMT)	8	1
– ITU G.992.2 (G.Lite)	1,5	0,5
• **ADSL2**		
– ITU G.992.3/4	12	1,0
– ITU G.992.3/4 Annex J	12	3,5
– ITU G.992.3/4 Annex L	5	0,8
• **ADSL2+**		
– ITU G.992.5	24	1,0
– ITU G.992.5 Annex L	24	1,0
– ITU G.992.5 Annex M	24	3,5

Стандарты xDSL в историческом развитии, типы модуляции и скорости:



6.2.6 Спутниковый доступ к Интернет

Односторонний (one-way) — для приёма данных используется спутниковый канал, а для передачи — доступные наземные каналы.



Двухсторонний (two-way) — для приёма, и для передачи используются спутниковые каналы.

Требует:

- особого качества антенну
- высокочастотный приемный и передающий блок, устанавливаемый на облучатель антенны
- спутниковый терминал

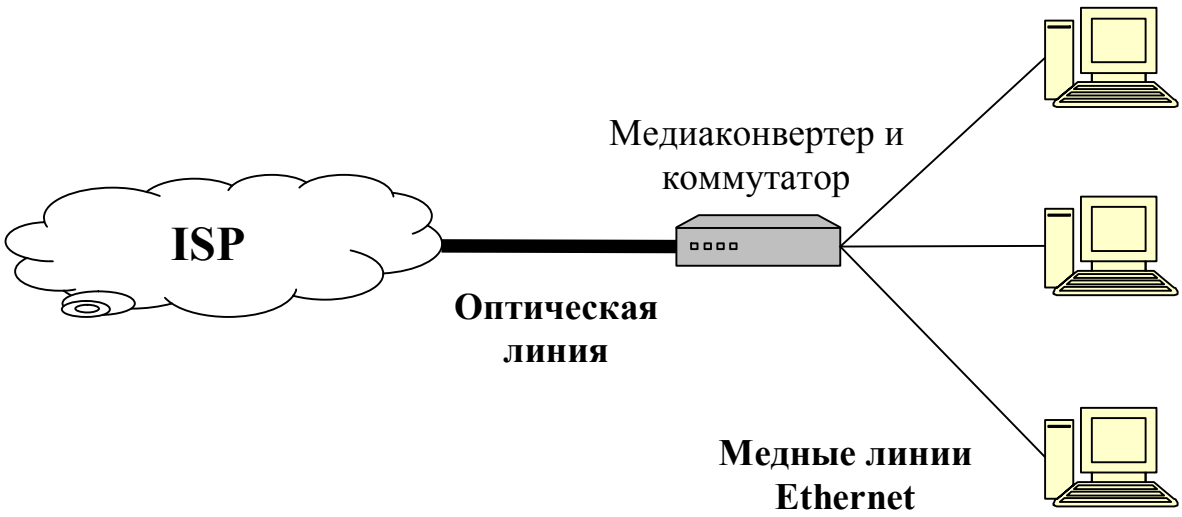
Клиентские настройки не отличаются от настроек обычного проводного доступа к Интернет (например, xDSL).

6.2.7 Технология ETTH

ETTH (Ethernet To The Home) способ подключения к Интернет по протоколы Fast Ehternet.

Разработка Teleste Corporation и Tratec Telecom B.V.

100 Мбит/с или 1 Гбит/с



Причины появления:

- Большая распространенность сетей Ethernet в качестве LAN.
- Удорожание стоимости технического решения «последней мили» (отрезок от оборудования провайдера до клиента).

Преимущества ETTH:

- Прозрачность доступа к сети Интернет (не требуется специального оборудования типа модемов).
- Высокая как нисходящая (download), так и восходящая (upload) скорость.
- Большая протяженность линии до клиента (против нескольких километров в xDSL).

Следствия появления и распространения ETTH:

- Упрощение подключения.
- Вытеснение xDSL с рынка технологий абонентского доступа.

6.2.8 Технология xPON

PON (Passive optical network) — технология пассивных оптических сетей.

С **1995** года

Смысл PON в том, что между центральным узлом, обеспечивающим подключение к магистрали (SDH/ATM), и абонентскими узлами создается полностью пассивная оптическая сеть древовидной топологии.

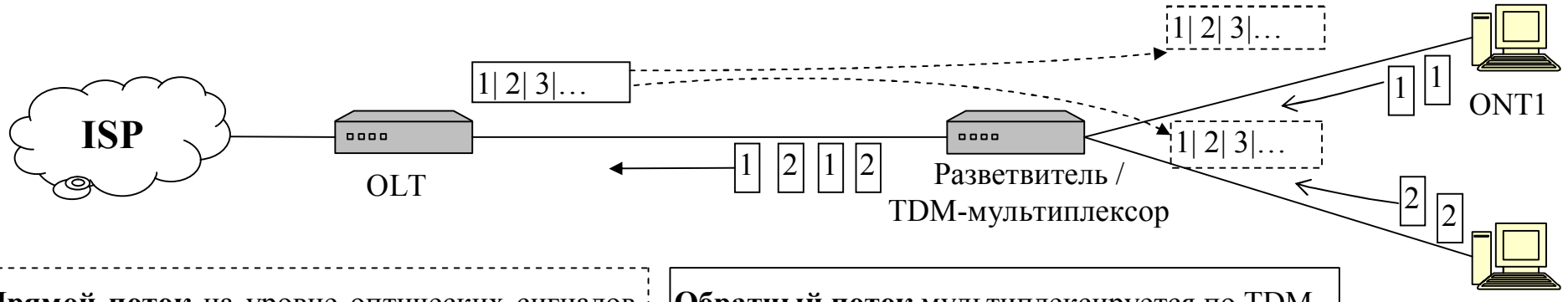
British Telecom, France Telecom, Deutsche Telecom, NTT, KPN, Telefonica и Telecom Italia

Идея архитектуры PON — используется один приёмопередающий модуль в оптическом терминале OLT для передачи информации множеству абонентских устройств ONT/ONU и приёма информации от них.

ONT (optical network terminal), термин ITU

ONU (optical network unit), термин IEEE

OLT (optical line terminal)



Прямой поток на уровне оптических сигналов, является широковещательным.

Каждый абонентский узел ONT, читая адресные поля, выделяет из этого общего потока предназначенную только ему часть информации.

Обратный поток мультиплексируется по TDM.

Все абонентские узлы ONT ведут передачу на одной и той же длине волны, используя концепцию множественного доступа с временным разделением (TDM).

7 Элементы безопасности

Информационная безопасность информационной системы — это состояние ИС, при котором она, с одной стороны, способна противостоять дестабилизирующему воздействию внешних и внутренних информационных угроз, а с другой – ее наличие и функционирование не создает информационных угроз для элементов самой системы и внешней среды.

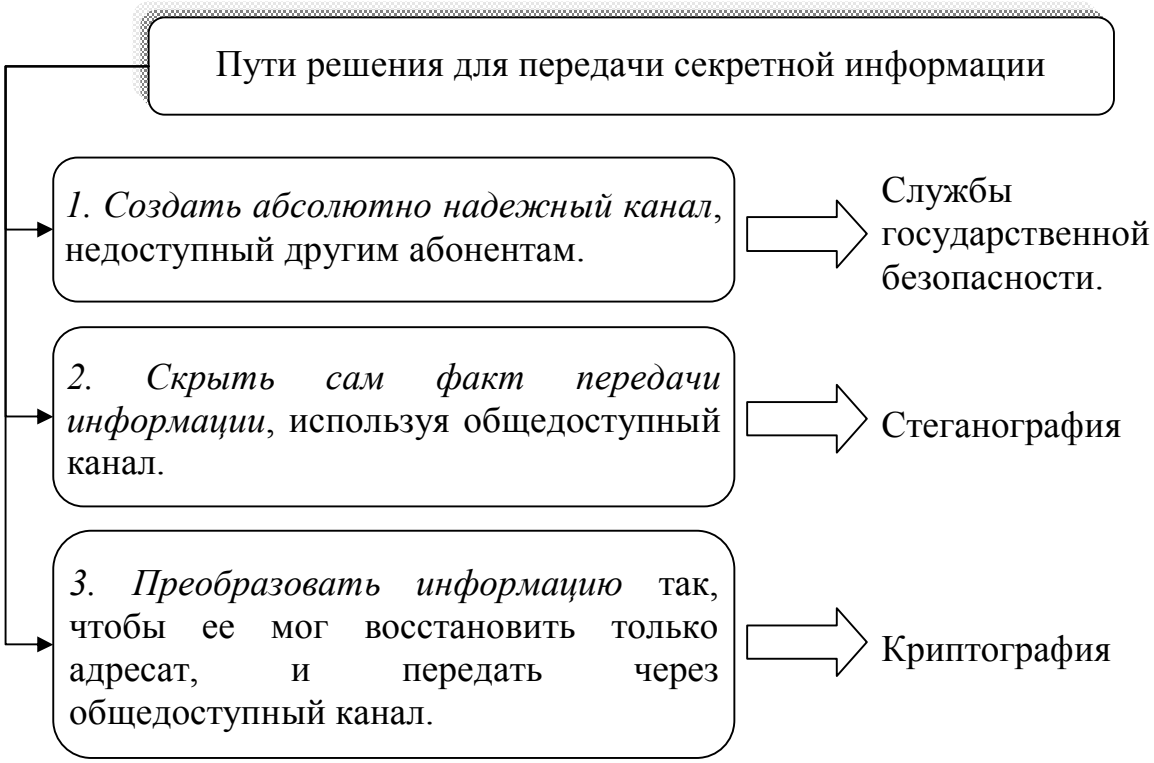
Три вида угроз безопасности

Угроза нарушения конфиденциальности – информация становится известной тому, кто не располагает полномочиями доступа к ней.

Угроза нарушения целостности – любое несанкционированное умышленное изменение информации, хранящейся в вычислительной системе или передаваемой из одной системы в другую.

Угроза отказа служб – блокировка доступа к некоторому ресурсу вычислительной системы в результате преднамеренных действий пользователем или злоумышленником.

7.1 Передача секретной информации



Стеганография — это наука, изучающая такие методы организации передачи секретных сообщений, которые скрывают сам факт передачи информации.

Акростих	Запись данных между дорожками на магнитных дисках
Письмо через трафарет	Запись данных в картинку, звук или видео изменением наименьшего значащего бита LSB (Least Signification Bits).

Криптография — это наука о преобразовании (шифровании) информации с целью ее защиты от незаконных пользователей.

Криптоанализ — анализ с помощью математических методов с целью нарушения конфиденциальности и аутентичности информации без знания шифра и его ключа.

7.2 Элементы криптографии

Шифрование — процесс преобразования информации по заданному алгоритму (шифру) с использованием некоторого ключа (сменный элемент шифра).

Криптографическая система или **шифр** — это семейство обратимых преобразований открытого сообщения (текста) в шифрованный.

Ключ — это значение параметров алгоритма криптографического шифра, т.е. его сменный элемент.

Типы криптографических систем или шифров

- Симметричные шифры
- Асимметричные шифры

Условия абсолютной стойкости шифра

1) **полная случайность** (равновероятность) ключа (ключ нельзя вырабатывать с помощью детерминированного устройства)

2) **равенство** длины ключа и длины открытого текста

3) **однократность** использования ключа

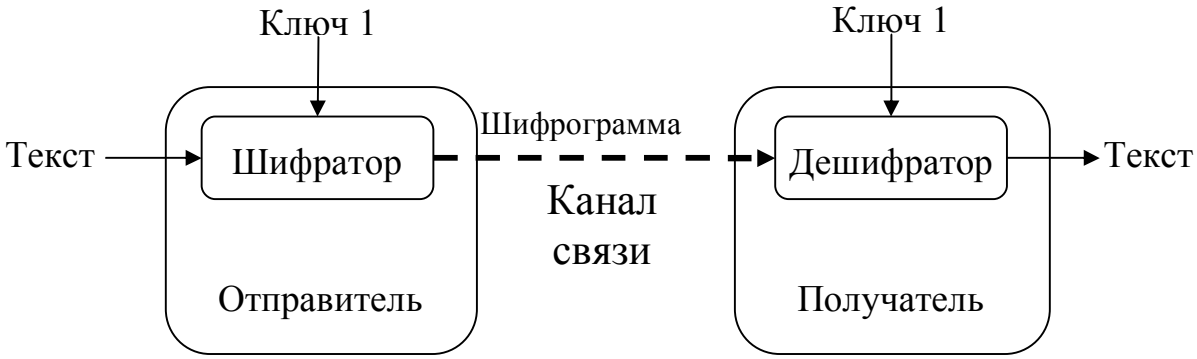
Анализ криптограмм

- Частотный анализ символов
- Цепи Маркова

7.2.1 Симметричные шифры

Симметричные шифры — такие шифры, в которых для шифрования и для расшифровывания используется один и тот же ключ.

~ По аналогии, чтобы передать секретную информацию абонент №1 покупает замок с ключом. Копию ключа он передает абоненту №2 по некоторому защищенному каналу связи. Затем закрывает информацию с помощью замка на этот ключ, и отправляет зашифрованные данные абоненту №2, который открывает их полученным ключом.



Виды симметричных шифров

Подстановки — заключается в замене символов исходного текста на другие (того же алфавита) по более или менее сложному правилу (Цезаря, Вижинера и т.д.).

Перестановки — перестановка символов по известному ключу

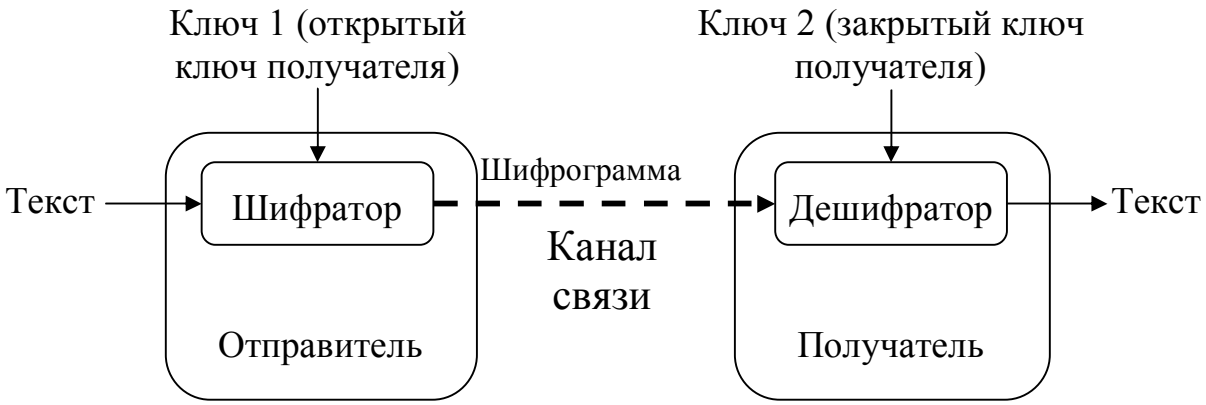
Гаммирование — заключается в генерации гаммы шифра с помощью датчика псевдослучайных чисел и наложении полученной гаммы на открытые данные обратимым образом (например, используя сложение по модулю 2)

Блочные шифры — похож на подстановки, но открытый текст шифруется блоками по несколько символов за одну итерацию.

7.2.2 Ассиметричные шифры

Ассиметричные шифры – такие шифры, в которых для шифрования и для расшифровывания используются различные ключи.

~ По аналогии, чтобы абонент 1 передал абоненту 2 секретную информацию абонент №2 покупает замок с ключом и абоненту №1 отправляет незакрытый замок без ключа. Абонент №1 закрывает информацию на этот захлопывающийся замок и отправляет абоненту №2. Так как ключ от замка был только у него (у 2) и никому никогда не передавался, то такой способ более надежный. При этом восстановление ключа по замку должно быть очень сложной задачей.



Криптографические системы с открытым ключом используют так называемые необратимые или односторонние функции.

$f(x) \rightarrow y$ легко

$y \rightarrow x$ очень сложно

Типы необратимых преобразований

- Разложение больших чисел на простые множители
- Вычисление логарифма в конечном поле
- Вычисление корней алгебраических уравнений

7.2.3 Ассиметричный шифр RSA

Шифр RSA разработан в 1977 году и получил название в честь его создателей: Рона Ривеста, Ади Шамира и Леонарда Эйдельмана.

В настоящее время алгоритм RSA используется во многих стандартах, среди которых протоколы SSL и HTTPS, в продуктах PGP, встроен в IE и Netscape.

Создание открытого и закрытого ключа:

1. Пу**с**ть $n = pq$, где p и q - различные простые числа.
2. Находим $\varphi(n) = (p - 1)(q - 1) = n - p - q + 1$.
3. Выбираем такое e и d, что $e\ d \bmod \varphi(n) = 1$.
4. Теперь e и n – открытый ключ, d и n – закрытый

Использование открытого и закрытого ключа:

Расшифровывание заключается в действии:

$$y^d \bmod n = x$$

Шифрование заключается в действии:

$$x^e \bmod n = y$$

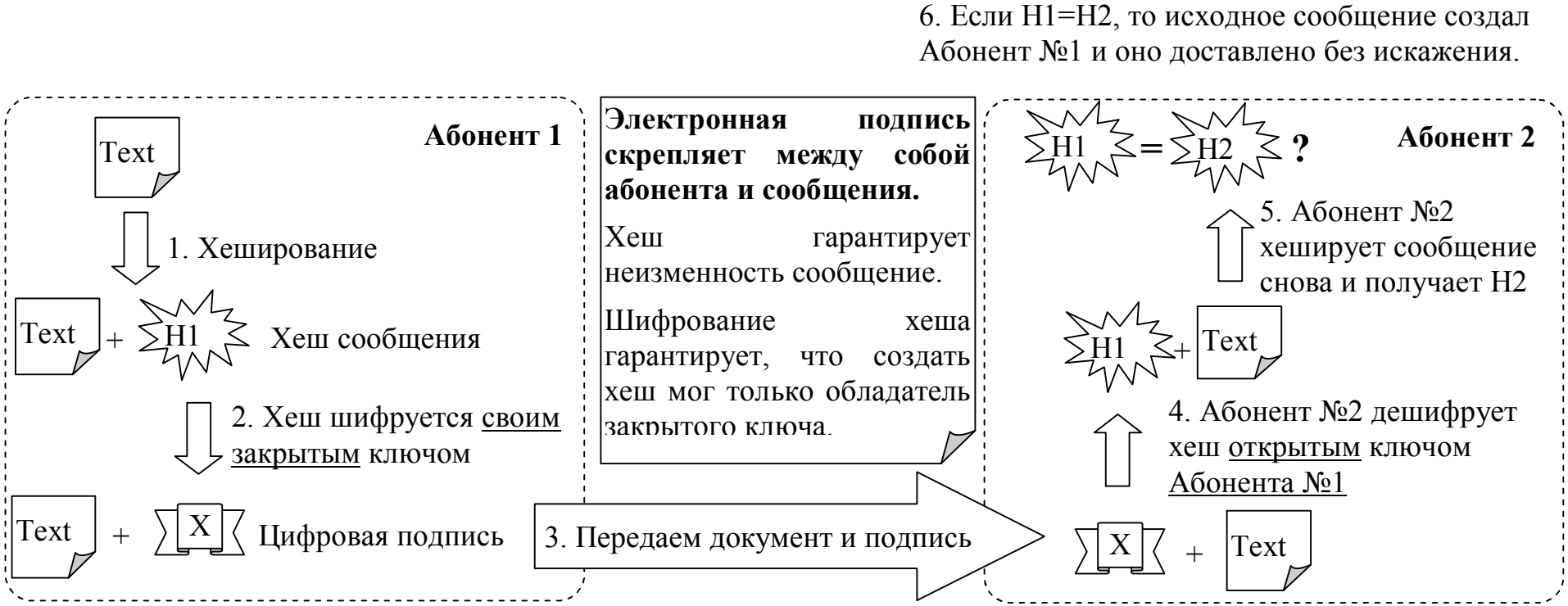
7.2.4 Цифровая подпись

Цифровая подпись — реквизит документа, предназначенный для аутентификации абонента, то есть определения того, действительно ли он является тем за кого себя выдает.

Основан на алгоритме RSA и хеш-функциях.

Хеш-функция — функция, которая отображает бесконечное множество значений во множество *p*, фиксированной мощности *n*.

Алгоритмы хеширования: MD4, MD5, SHA, и т.д.



Практические работы

В каждой работе используется ряд виртуальных машин. Установка операционных систем не входит в объем времени работы, поэтому следует заранее подготовить виртуальные машины и операционные системы для них.

Рекомендуется заранее установить один экземпляр клиентской ОС Windows (Windows XP), серверной (Windows Server 2003) и серверной ОС Linux (openSuSE). Никакие дополнительные настройки ОС кроме установки по умолчанию не предполагаются. Необходимо сделать копию виртуальных дисков этих машин, так как потребуется создавать по две различные виртуальные машины с серверной и клиенткой операционной системой, которые будут использоваться в различных практических работах.

Выбор дистрибутива Linux может быть различным. Дистрибутив SLAX Linux Live CD выбран так, как имеет весьма небольшой размер (230Мб) и работает при 64 Мб доступной оперативной памяти, что весьма удобно в работах с большим числом виртуальных машин.

Виртуальные машины с серверной операционной системой постепенно, от одной работе к другой, будут наращиваться прикладными службами, поэтому лучше использовать одни и те же образы. Применение в разных работах одинаковых названий узлов сети означает использование одних тех же виртуальных машин.

Многие работы связаны друг с другом и учитывается, что они выполняются последовательно. Учитывается, что настроив однажды DHCP-сервер, в последующих работах клиенты могут получать через него IP-адреса. Вместе с тем, выполнять работы можно и в произвольном порядке, так как в инструкциях присутствуют необходимые ссылки на другие работы или имеются иные компенсирующие действия.

Практические работы содержат сведения о базовых настройках прикладных служб и безопасности, достаточные для понимания и усвоения принципов их работы, а также технологий и протоколов связанных с ними. За дополнительной информацией следует обращаться к документации по продуктам, а также специальным руководствам.

Практическая работа 1. Обжатие витой пары

Цель работы: Изучить процесс оконечивания кабелей типа витой пары. Получить навыки работы с инструментом обрезки, обжимным инструментом (кримпер), инструментом обжимки на кросс, кабельным тестером.

Объем времени: 1 ч. (7-12 чел)

Инструменты: инструмент обрезки, обжимной инструмент (кримпер), инструмент обжимки на кросс, кабельный тестер, коннекторы RJ-45.

Необходимые сведения:

Порядок цветных жил в коннекторе:

	T586A	T586B	Кроссовая разводка	
			Конец 1	Конец 2
1	Бело-Зеленый.	Бело-Оранж.	Бело-Оранж.	Бело-Зеленый.
2	Зеленый	Оранжевый	Оранжевый	Зеленый
3	Бело-Оранжев.	Бело-Зеленый	Бело-Зеленый	Бело-Оранжев.
4	Голубой	Голубой	Бело-Голубой	Бело-Голубой
5	Бело-Голубой	Бело-Голубой	Голубой	Голубой
6	Оранжевый	Зеленый	Зеленый	Оранжевый
7	Бело-Коричн.	Бело-Коричн.	Бело-Коричн.	Бело-Коричн.
8	Коричневый	Коричневый	Коричневый	Коричневый

В стандарте Fast Ethernet используются проводники: 1, 2, 3, 6.

Последовательность действий:

1. Отрежьте необходимый сегмент кабеля
2. Инструментом обрезки снимите 2-3 см изоляции с кабеля, проверьте, не повредились ли жилы проводников.
3. Расположите цветные жилы в правильном порядке и выпрямите их так, чтобы они находились рядом друг с другом.
4. Ровно обрежьте жилы кабеля инструментом обрезки.
5. Вставьте жилы кабеля в коннектор до упора.
6. Проверьте порядок цветных жил в коннекторе. При расположении коннектора контактами вперед с обратной стороны коннектора (где нет лапки) должен наблюдаться прямой порядок цветных жил (слева направо).
7. Вставьте коннектор с жилами кабеля в обжимной инструмент (кримпер) и зажмите рукоятку инструмента до упора.

8. Извлеките обжатый коннектор из инструмента.
9. То же самое проделайте с другим концом кабеля.
10. Проверьте кабель с помощью кабельного тестера, проверьте правильность прохождения сигнала по парам кабеля.

Результаты: оконеченный с обоих концов кабель типа «витая пара», готовый к использованию в локальной сети.

Практическая работа 2. Подключение узлов, использование коммутаторов

Цель работы: Изучить функционирование повторителей и коммутаторов, рассмотреть процесс подключения узлов к коммуникационному оборудованию. Получить навыки подключения узлов к коммутаторам. Научиться планировать расположение коммуникационных устройств в помещении и подключать узлы к ним.

Объем времени: 1 ч. (7-12 чел)

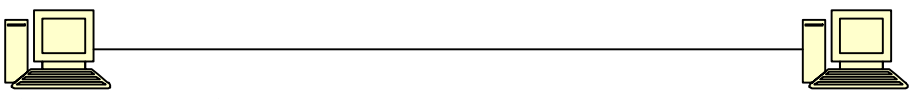
Аппаратное обеспечение: 1(пп.1-13) или 2(пп.1-22) коммутатора, 2 компьютера (узел 1: 192.168.1.15, узел 2: 192.168.1.20), 4 сегмента кабеля, 1 сегмент кабеля с кроссовой разводкой.

Необходимые команды:

`ipconfig /all`	Выводит информацию обо всех сетевых интерфейсах компьютера (ОС Windows)
`ifconfig -a`	Выводит информацию обо всех сетевых интерфейсах компьютера (ОС Linux)
`ping <хост>`	Посылает сообщение ICMP с эхо-запросом узлу с указанным IP-адресом или DNS-именем.

Последовательность действий:

1. Соедините оба компьютера сегментом кабеля с кроссовой разводкой.

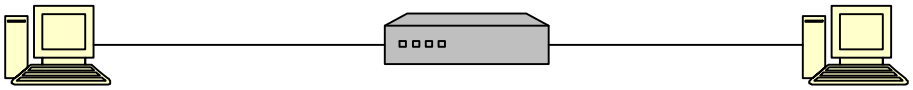


Узел 1 (192.168.1.15) Узел 2 (192.168.1.15)

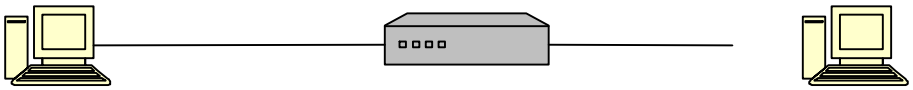
2. Проверьте настройки сетевого адаптера командой. С узла с IP-адресом 192.168.1.15 пошлите эхо-запрос другому узлу (с IP-адресом 192.168.1.20)

```
>ping 192.168.1.15
```

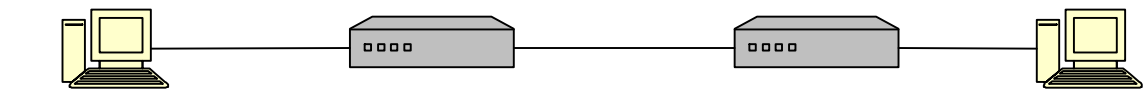
3. Убедитесь в прохождении эхо-запросов между узлами. Сохраните снимок экрана с эхо-ответом.
4. Отсоедините кабель с кроссовой разводкой от узлов.
5. Подключите узлы друг к другу через кабель с прямой разводкой.
6. Проверьте прохождения эхо-запросов командой `ping` с обоих узлов.
7. Сравните результат эхо-запроса с результатами из пункта 3.
8. Подключите оба компьютера к коммутатору через отдельные сегменты кабелей к коммутатору.



9. Проверьте прохождение эхо-запроса между узлами.
10. Отключите кабель от сетевого интерфейса узла №2.



11. Проанализируйте результат эхо-запроса с узла №1 до узла №2.
12. Проанализируйте результат эхо-запроса с узла №2 до узла №1.
13. Сравните эхо-ответы из п.11 и п.12.
14. Подключите каждый узел к отдельному коммутатору. Соедините коммутаторы друг с другом через свободные порты сегментом кабеля.



15. Проверьте успешное прохождение эхо-запросов в обе стороны.
16. Отключите сегмент кабеля между коммутаторами.
17. Проверьте прохождение эхо-запросов в обе стороны. Сравните результаты эхо-ответа с п.15.
18. Подключите коммутаторы друг к другу двумя различными сегментами кабеля до образования цикла (дублирующей связи).

19. Проверьте прохождение эхо-запросом между двумя узлами. Если коммутаторы поддерживают технологию Spaning Tree, попробуйте выполнить п.18 с включенной и отключенной опцией Spaning Tree.
20. Восстановите схему подключения как в п.14.
21. Перезагрузите коммутаторы отключением питания.
22. Проверьте прохождения эхо-запросов.

Результаты: Сравните результаты эхо-запросов в пп.3, 6, 11, 12, 17, 19.

Практическая работа 3. Настройка сетевых адаптеров (Windows)

Цель работы: Изучить технологию настройки сетевых адаптеров для подключения узлов в локальную сеть в ОС Windows. Получить навыки настройки сетевых адаптеров и устранения типичных неисправностей в ОС Windows.

Объем времени: 1 ч. (с учетом настройки виртуальных машин, но без учета установки ОС. Рекомендуется заранее установить ОС в виртуальную машину и копировать файл виртуального диска на рабочие места).

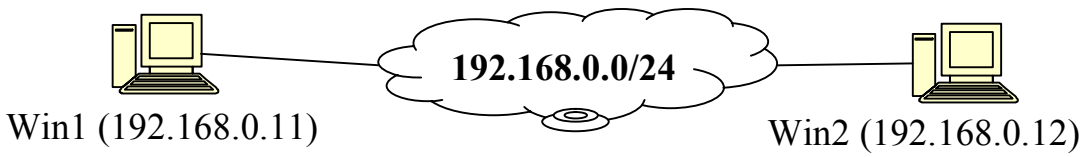
Программное обеспечение: ISO-образ ОС Windows XP.

Виртуальные машины:

Win1 (HDD: 4Gb; RAM: 128Mb; LAN0: internal),

Win2 (HDD: 4Gb; RAM: 128Mb; LAN0: internal)

Схема сети:



Необходимые команды:

`ipconfig /all`	Выводит информацию обо всех сетевых интерфейсах компьютера (ОС Windows)
`ping <хост>`	Посылает сообщение ICMP с эхо-запросом узлу с указанным IP-адресом или DNS-именем.

Последовательность действий:

1. Включите узел Comp1

2. Проверьте сетевые настройки с помощью команды `ipconfig`.

   ```
   > ipconfig -all
   ```

3. Откройте окно настройки сетевого интерфейса: **Пуск | Панель управления | Сетевые подключения**. Выберите локальный сетевой интерфейс. Правым щелчком мыши по пункту сетевого интерфейса вызовите контекстное меню, выберите пункт **Свойства**.

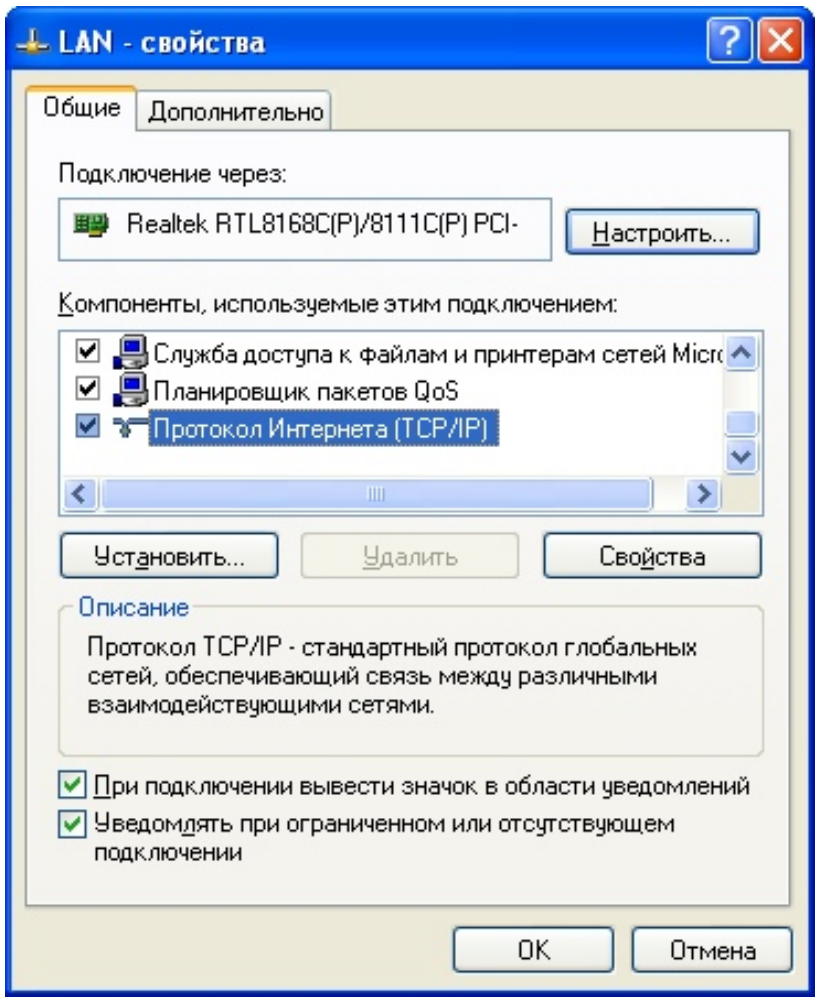


Рис. 19. Окно свойств сетевого интерфейса

4. Выберите в списке протоколов **Протокол Интернета (TCP/IP)** и щелкните по кнопке **Свойства**.
5. Установите переключатель **Использовать следующий IP-адрес**.
6. Укажите в настройках IP адрес узла.
7. Щелкните по кнопке **Ok** во всех окнах, и по кнопке **Закрыть** в окне свойств сетевого подключения.
8. Проверьте сетевые настройки с помощью команды `ipconfig -all`.
9. Проверьте прохождение эхо-запросов командой `ping`.

Если эхо-запросы не проходят, разрешите в Windows firewall входящие эхо-запросы.

Для этого в окне брандмауэра Windows перейдите на вкладку **Дополнительно** (**Advanced**), выберите сетевой интерфейс, щелкните по кнопке **Настройки**… (**Settings**), на вкладке **ICMP,** установите флажок **Разрешить входящий эхо-запрос** (**Allow incoming echo request**). Во всех открытых окнах щелкните по кнопке **Ок**.

Практическая работа 4. Настройка сетевых адаптеров (Linux)

Цель работы: Изучить технологию настройки сетевых адаптеров для подключения узлов в локальную сеть в ОС Linux. Получить навыки настройки сетевых адаптеров и устранения типичных неисправностей в ОС Linux.

Объем времени: 1 ч. (с учетом настройки виртуальных машин, установка ОС Linux не входит).

Программное обеспечение: ISO-образ ОС SLAX 6.0.6 Linux Live CD (в этой и других работах дистрибутив Linux может быть изменен. Однако, следует учитывать, что команды и расположение конфигурационных файлов могут быть иными), ISO-образ ОС openSuSE 11.4.

Виртуальные машины:

Lin2 (HDD: нет; RAM: 128Mb; LAN0: internal; ОС: SLAX),

LinOS (HDD: 8 Gb; RAM: 256Mb; LAN0: internal; ОС: openSuSE).

Схема сети:

Необходимые команды:

`ifconfig -a`	Выводит информацию обо всех сетевых интерфейсах компьютера (ОС Linux)
`ifconfig <ethX> <IP>/<MASK> up`	Назначение IP-адреса сетевому интерфейсу ethX
`ip addr show`	Выводит информацию о настройках сетевых интерфейсах. Работает без root-доступа.
`ip addr add`	Привязывает адрес к сетевому интерфейсу.

`<ip_addres>/<mask> dev <ethX>`	
`ip addr del <ip>/<mask> dev <ethX>`	Удаляет привязку IP-адреса на указанном интерфейсе.
`ping <хост>`	Посылает сообщение ICMP с эхо-запросом узлу с указанным IP-адресом или DNS-именем.
`su [`*`username`*`]`	Смена текущего пользователя. По умолчанию на root. После подачи команды будет запрошен пароль к учетной записи указанного пользователя.
`cd` *`directory`*	Переход в указанную директорию на диске.
`vim` *`filename`*	Открыть указанный файл из текущего каталога в текстовом редакторе vim.

Последовательность действий:

1. Запустите ОС Linux с LiveCD (SLAX Linux) на виртуальной машине Lin2.
2. Откройте консоль, смените пользователя на root командой `su`.

   ```
   user# su
   password:
   ```

3. Просмотрите информацию об интерфейсах

   ```
   #ifconfig –a
   ```

4. Отключите интерфейс командой:

   ```
   #ifconfig eth0 down
   ```

5. Поднимите интерфейс с нужными настройками сети командой

   ```
   #ifconfig eth0 10.0.0.1 netmask 255.0.0.0 up
   ```

6. Настройте сетевой интерфейс на узле LinOS аналогично Lin2.
7. Проверьте работоспособность сети командой `ping` с узла Lin2.

   ```
   #ping 10.0.0.1
   ```

8. Проверьте работоспособность сети командой `ping` с узла LinOS.
9. Перезагрузите узел LinOS (openSuSE). Убедитесь, что настройки сети сбросились.
10. Воспользуемся утилитой ip, чтобы настроить аналогичные сетевые настройки на узле LinOS. Просмотрите список подключений командой:

    ```
    #ip link show
    ```

11. Просмотрите сетевые настройки интерфейсов с помощью команды:

```
#ip addr show
```

12. Смените пользователя на `root` командой `su`.
13. Добавьте настройку IP-адреса к сетевому интерфейсу eth0 командой:

```
#ip addr add 10.0.0.1/8 dev eth0
```

14. Просмотрите сетевые настройки интерфейсов с помощью команды:

```
#ip addr show
```

15. Проверьте прохождение эхо-запросов между узлами LinOS и Lin2
16. Добавьте на узле LinOS к интерфейсу eth0 еще один IP-адрес командой:

```
#ip addr add 10.0.0.3/8 dev eth0
```

17. Проверьте прохождение эхо-запросов с узла Lin2 по обоим IP-адресам (10.0.0.1 и 10.0.0.3).
18. Удалите один из IP-адресов с узла LinOS командой:

```
#ip addr del 10.0.0.3/8 dev eth0
```

19. Перезагрузите узел LinOS. Проверьте сброс сетевых настроек интерфейсов.
20. Настройте постоянную конфигурацию сети с помощью конфигурационного файла. Найдите файл (в различных дистрибутивах и версиях путь и имя файла может отличаться, а вместо X может быть как MAC адрес карты, так и символьное имя интерфейса, например eth0):

```
/etc/sysconfig/network/ifcfg-eth-id-XX:XX:XX:XX:XX)
```

21. Откройте файл в консольном текстовом редакторе `VIM` от имени root командой:

```
#su
password:
#cd /etc/sysconfig/network/
#vim ifcfg-eth-id-XX:XX:XX:XX:XX
```

22. Нажмите клавишу **i**, для перехода в режим редактирования. Отредактируйте файл:

```
BOOTPROTO=static
BROADCAST=10.255.255.255
IPADDR=10.0.0.2
NETMASK=255.0.0.0
NETWORK=10.0.0.0
ONBOOT=yes
```

23. Нажмите на клавиатуре клавишу **Esc**, чтобы перейти в режим команд. Нажмите на клавиатуре клавишу **Z**, **Z**, для сохранения файла.
24. Проверьте прохождение эхо-запросов командой `ping`.
25. Перезагрузите узел LinOS (openSuSE).
26. Проверьте прохождение эхо-запросов командой `ping`.

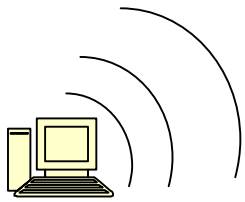
Практическая работа 5. Подключение узлов по протоколу Wi-Fi в режиме Ad-Hoc

Цель работы: Изучить способы подключение узлов с помощью беспроводной сети Wi-Fi. Получить навыки подключения узлов в сети Wi-Fi в режиме компьютер-компьютер (Ad-Hoc).

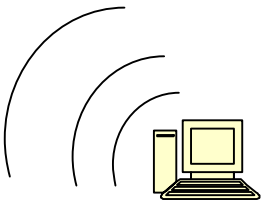
Объем времени: 1 ч.

Аппаратное обеспечение: 2 компьютера с беспроводными сетевыми адаптерами Wi-Fi.

Схема сети:



Узел 1 (192.168.1.15)



Узел 2 (192.168.1.20)

Последовательность действий:

1. Проверьте установку драйверов беспроводных сетевых адаптеров и настройки сетевого протокола.
2. Установите настройки сетевого адаптера. Выберите частотный канал (например, 11, или канал AdHoc, если есть).
3. Придумайте или сгенерируйте специальной программой ключ WEP.
4. Выберите тип беспроводной сети Open, шифрование WEP, укажите ключ шифрования. Примените настройки.
5. Повторите настройки на втором узле.
6. Включите оба узла, проверьте подключение.

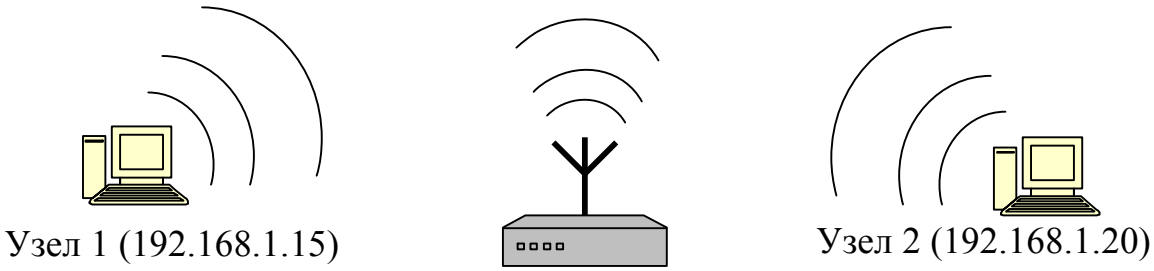
Практическая работа 6. Подключение Wi-Fi через точку доступа

Цель работы: Изучить способы подключение узлов с помощью беспроводной сети Wi-Fi. Получить навыки подключения узлов в сети Wi-Fi через точку доступа (Access Point).

Объем времени: 1 ч.

Аппаратное обеспечение: 2 компьютера с беспроводными сетевыми адаптерами Wi-Fi, точка доступа (Access Point).

Схема сети:



Общая схема действий:

1. Подключить узел и точку доступа непосредственно друг к другу.
2. Настроить точку доступа через веб-интерфейс
3. Отключить проводной сегмент и настроить узлы через точку доступа.

Последовательность действий:

1. Проверьте установку драйверов беспроводных сетевых адаптеров и настройки сетевого протокола.
2. Подключите один узел и точку доступа к одному коммутатору.

3. Подключитесь к точке доступа по веб-интерфейсу через IP-адрес, который указан в документации к точке доступа.
4. Укажите настройки точки доступа: SSID (например, ProbaWiFI), Частотный канал (например, 11).
5. Придумайте или сгенерируйте специальной программой ключ WEP/WPA/WPA2 или др.

6. Выберите тип беспроводной сети Open, тип шифрования WEP/WPA/WPA2, укажите ключ шифрования.
7. Примените настройки.
8. Отключите проводной сегмент между узлом и точкой доступа.
9. Подключите к беспроводной сети сначала один узел. Для этого добавьте новую беспроводную сеть на клиенте. Укажите SSID сети, введите ключ шифрования и сохраните настройки.
10. Установите аналогичные настройки на втором узле.
11. Проверьте подключение на обоих узлах командой `ping`.

Практическая работа 7. Исследование работы ARP (Windows или Linux)

Цель работы: Изучить работу протокола ARP. Получить навыки анализа и модификации ARP-таблицы.

Объем времени: 1 ч. (с учетом настройки виртуальных машин, установка ОС Linux/Windows не входит).

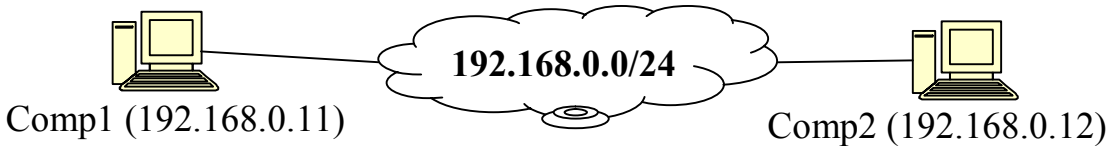
Программное обеспечение: ISO-образ ОС SLAX 6.0.6 Linux Live CD (в этой и других работах дистрибутив Linux может быть изменен. Однако, следует учитывать, что команды и расположение конфигурационных файлов могут быть иными), ISO-образ ОС openSuSE 11.4.

Виртуальные машины:

Win1 (HDD: 4Gb; RAM: 128Mb; LAN0: internal),

Win2 (HDD: 4Gb; RAM: 128Mb; LAN0: internal)

Схема сети:



Необходимые команды:

`arp -a`	Выводит записи ARP таблицы.
`ip neigh show`	(Linux) Выводит записи ARP таблицы (Ethernet-соседи). Не требует root-доступа.
`arp -d` *IP*	Удаляет запись из ARP-таблицы.

`arp –s` *`IP MAC`*	Добавляет статическую запись в ARP таблицу с IP и указанным MAC адресом.
`ip neighbor add <ip> lladdr <mac> dev <ethX> nud permanent`	Добавляет статическую запись в ARP таблицу с IP и указанным MAC адресом.
`ip neighbor del <ip> lladdr <mac> dev <ethX> nud permanent`	Удаляет запись из ARP-таблицы со специфическим IP, MAC адресом для выбранного интерфейса.
`ping <`*`хост`*`>`	Посылает сообщение ICMP с эхо-запросом узлу с указанным IP-адресом или DNS-именем.

Последовательность действий:

1. Просмотрите ARP-таблицу узла Comp1:

   `>arp –a`

   Просмотреть список записей в ОС Linux возможно аналогичной командой, а также с помощью утилиты ip:

   `#ip neigh show`

2. Зафиксируйте ее (сделайте снимок экрана)
3. Пошлите эхо-запрос с помощью команды `ping` по IP адресу узла Comp2 (его не должно быть в ARP-таблице).
4. Просмотрите таблицу ARP.
5. Определите MAC-адрес узла, которому в п.3 послан эхо-запрос.
6. Удалите запись из ARP-таблицы:

   `>arp –d IP`

7. Проверьте отсутствие записи в таблице
8. Добавьте статическую запись в ARP-таблицу узла Comp1 на IP адрес узла Comp2 с фиктивным MAC-адресом (реально несуществующим в данной сети):

   `>arp –s 192.168.1.12 12-34-56-78-9A-BC`

   Для ОС Linux добавление через утилиту arp будет выглядеть аналогично, а через утилиту ip следующим образом:

   `#ip neigh add 192.168.1.12 lladdr 00:11:22:33:44:55 dev eth0 nud permanent`

9. Проверьте наличие добавленной записи
10. Пошлите эхо-запрос по IP адресу Comp2

11. Удалите из таблицы фиктивную запись. Для ОС Linux через утилиту ip, команда будет следующей:

```
#ip neighbor del 192.168.1.12 lladdr 00:11:22:33:44:55 dev eth0 nud permanent
```

12. Пошлите эхо-запрос по IP адресу Comp2
13. Проанализируйте результаты эхо-ответов в п.10 и п.12.

Практическая работа 8. Изучение работы протокола TCP

Цель работы: Изучить принципы маршрутизации пакетов. Получить навыки настройки маршрутизации пакетов в ОС Linux, а также настройки основного шлюза сетевых интерфейсов для доступа в другие сети.

Объем времени: 1-2 ч.

Программное обеспечение: сниффер пакетов.

Последовательность действий:

Сделайте запрос в браузере к странице http://ya.ru, и проанализируйте ответ.

1. Запрос на разрешение доменного имени:

```
(UDP)192.168.1.5:3890->195.112.224.126:53 ,51 Bytes
DNS queries:1 (ya.ru; QTYPE:1; CLASS:1)
```

2. Ответ DNS сервера

```
(UDP)195.112.224.126:53->192.168.1.5:3890 ,126 Bytes
DNS queries:1 (ya.ru; QTYPE:1; CLASS:1)
  answers:1 (ya.ru; … TTL (H:M:S):0:30:37; addr:213.180.204.8)
```

3. Запрос на установление соединения

```
(TCP)192.168.1.5:3894->213.180.204.8:80 ,48 Bytes
flags (SYN); seq_number:103688453; ack_number:0
```

4. Установление синхронизации

```
(TCP)213.180.204.8:80->192.168.1.5:3894 ,48 Bytes
flags (SYN+ACK); seq_number:356094 8111;
ack_number:103688454
```

(..453+1) (..8111+1)

5. Подтверждение синхронизации

```
(TCP)192.168.1.5:3894->213.180.204.8:80 ,40 Bytes
ack :1; seq_number:103688454; ack_number:356094 8112
```

6. Запрос HTTP

(TCP)192.168.1.5:3894->**213.180.204.8:80 ,377** Bytes

flags (ACK, PUSH); seq_number:103688**454**; ack_number:356094**8112**

HTTP **GET / HTTP/1.1**;

User-Agent: Opera/9.63 (Windows NT 5.1; U; ru) Presto/2.1.1;

Host: ya.ru;

Accept: text/html, application/xml;q=0.9, application/xhtml+xml, image/png, image/jpeg, image/gif, image/x-xbitmap, */*;q=0.1;

Accept-Language: ru-RU,ru;q=0.9,en;q=0.8;

Accept-Charset: iso-8859-1, utf-8, utf-16, *;q=0.1

7. Ответ web-сервера

(TCP)213.180.204.8:80->192.168.1.5:3894 **,287** Bytes

flags (ACK, PUSH); seq_number:356094**8112**; ack_number:103688**791**

(791–454=337), т.е.: ans(ack) – req(seq) = req(total size) – 2*head(size)

HTTPHTTP/1.1 200 OK;

Content-Type: text/html; charset=windows-1251;

Content-Length: 4848;

8. Данные от web-сервера

(TCP)**213.180.204.8:80**->192.168.1.5:3894 ,1400 Bytes

Flags; seq_number:356094**8359**; ack_number:103688**791**

8359–8112=**247**+20+20=**287**

HTTP; DATA(1360 byte) (HTML документ)

9719–8359=**1360**

9. Подтверждение клиента

(TCP)192.168.1.5:3894->**213.180.204.8:80** ,40 Bytes

Flags; seq_number:103688**791**; ack_number:356094**9719**

10. Продолжение 1 ответа сервера с HTML документом

(TCP)**213.180.204.8:80**->192.168.1.5:3894 ,1400 Bytes

Flags; seq_number:356094**9719**; ack_number:103688**791**

HTTP: DATA:1360 Bytes

11. Продолжение 2 ответа сервера с HTML документом

(TCP)**213.180.204.8:80**->192.168.1.5:3894 ,1400 Bytes

Flags; seq_number:35609**51079**; ack_number:103688**791**;

HTTP: DATA:1360 Bytes

51079–49719=1360

12. Подтверждение клиента

(TCP)192.168.1.5:3894->213.180.204.8:80 ,40 Bytes

Flags; seq_number:103688**791**; ack_number:35609**52439**;

13. Завершение отправки данных

52439–51079= 1360

(TCP)213.180.204.8:80->192.168.1.5:3894 ,808 Bytes

flags (FIN); seq_number:35609**52439**; ack_number:103688**791**

14. Подтверждение клиента

(TCP)192.168.1.5:3894->213.180.204.8:80 ,40 Bytes

Flags (ACK); seq_number:103688**791**; ack_number:35609**53208**

53208–52439=769

1. Пошлите запрос google.com
2. Найдите три строки, касающиеся установления соединения. Впишите начальные значения **SEQ_NUM** клиента и сервера.
3. Найдите первый ответ сервера после запроса клиента корневой страницы.
4. Найдите разницу между **ACK_NUM** из ответа сервера и **SEQ_NUM** из запроса клиента. Число должно быть равно **total_size** запроса клиента за двукратным вычетом размера заголовка **header_size**.
5. Найдите запрос клиента к корневой странице сервера.
6. Выпишите/сохраните текст запроса, а именно, строку запроса и заголовок клиента.
7. Найдите строки с завершением ответа сервера и подтверждением клиента.
8. Найдите разницу между **ACK_NUM** из подтверждения клиента и **SEQ_NUM** из данных сервера. Разница должны быть равна объему данных HTTP.
9. Пошлите в окне сниффера запрос на разрешение доменного имени. Просмотрите ответ.
    a. Если в сети установлен прокси-сервер, узнайте его IP-адрес и направьте запрос DNS на него.
10. Просмотрите кэш DNS командой:

```
>ipconfig –displaydns
```

11. Сбросьте кэш командой:

```
>ipconfig –flushdns
```

12. Просмотрите кэш DNS.

Практическая работа 9. Изучение функционирования сетевого экрана (Linux)

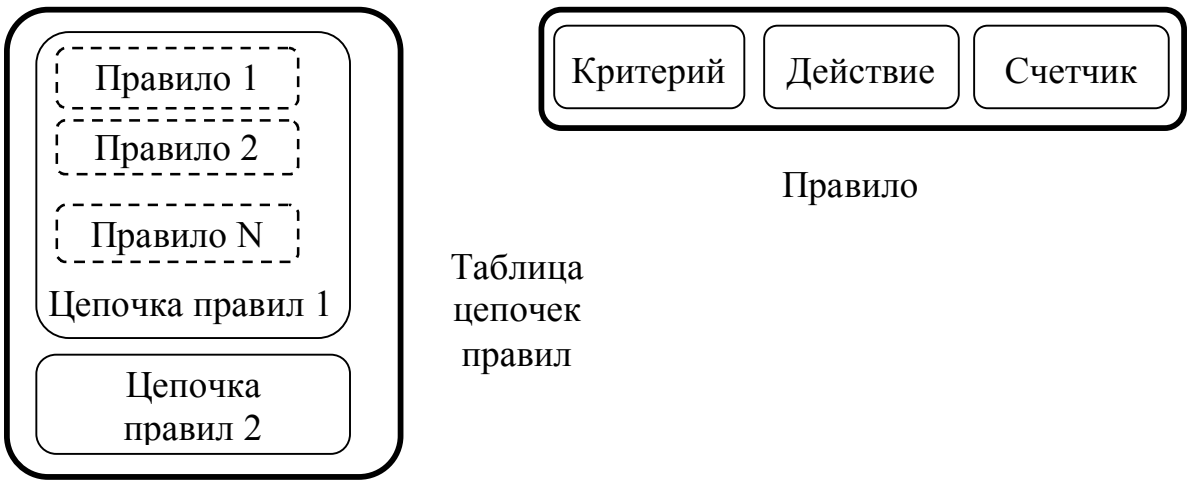
Цель работы: Изучить функционирование сетевого экрана в ОС Linux. Получить навыки настройки сетевого экрана в ОС Linux на примере протокола SSH. Получить навыки открытия и закрытия портов для определенного вида трафика на примере 22/TCP для демона sshd.

Объем времени: 1-2 ч.

Программное обеспечение: putty, ISO-образ ОС openSuSE, ISO-образ ОС Windows.

Необходимые сведения и команды:

Настройки сетевого экрана в ОС Linux производятся через утилиту iptables.



Основные понятия:

Правило — это запись, которая описывает критерии трафика и устанавливает действие над ним.

Если пакет соответствует критерию, к нему применяется действие, и он учитывается счетчиком. Критерий необязательный параметр, если его нет, то действие распространяется на все пакеты. Указание действия тоже не обязательно, в его отсутствии правило будет работать только как счетчик.

Критерий — логическое выражение, анализирующее свойства пакета и/или соединения и определяющее, подпадает ли данный конкретный пакет под действие текущего правила.

Действие — описание действия, которое нужно проделать с пакетом и/или соединением в том случае, если они подпадают под действие этого правила.

Счетчик — компонент правила, обеспечивающий учет количества пакетов, которые попали под критерий данного правила. Также счетчик учитывает суммарный объем таких пакетов в байтах.

Цепочка — упорядоченная последовательность правил. Цепочки делятся на пользовательские и базовые.

Базовая цепочка — цепочка, создаваемая по умолчанию при инициализации таблицы. Каждый пакет, в зависимости от того, предназначен ли он самому хосту, сгенерирован им или является транзитным, должен пройти положенный ему набор базовых цепочек различных таблиц. Базовая цепочка отличается от пользовательской наличием «действия по умолчанию» (default policy). Это действие применяется к тем пакетам, которые не были обработаны другими правилами этой цепочки и вызванных из нее цепочек. Имена базовых цепочек всегда записываются в верхнем регистре (PREROUTING, INPUT, FORWARD, OUTPUT, POSTROUTING).

Пользовательская цепочка — цепочка, созданная пользователем. Может использоваться только в пределах своей таблицы. Рекомендуется не использовать для таких цепочек имена в верхнем регистре, чтобы избежать путаницы с базовыми цепочками и встроенными действиями.

Таблица — совокупность базовых и пользовательских цепочек, объединенных общим функциональным назначением. Имена таблиц (как и модулей критериев) записываются в нижнем регистре, так как в принципе не могут конфликтовать с именами пользовательских цепочек. При отсутствии явного указания таблицы в правиле, используется таблица filter.

Правила в каждой цепочке применяются сверху вниз и действие (Action) применяется от того правила, которое сработало первым. Если ни одно правило не отработало, то применяется правило по умолчанию — обычно «пропустить» (ACCEPT) пакет.

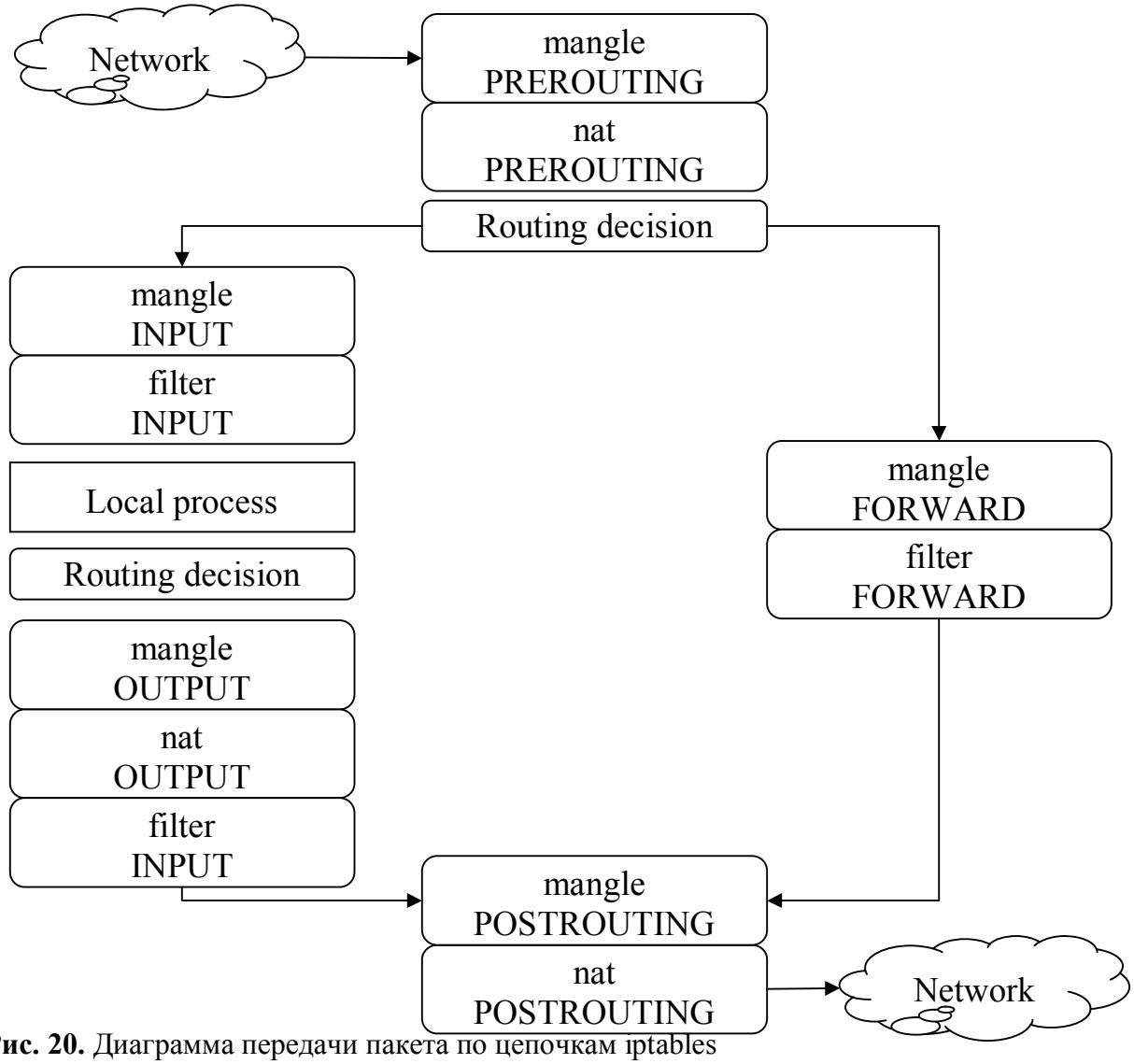


Рис. 20. Диаграмма передачи пакета по цепочкам iptables

Порядок прохождения цепочек и таблиц пакетами от локальных процессов:

Шаг	Таблица	Цепочка	Примечание
1			Локальный процесс (т.е., программа-сервер или программа-клиент).
2			Принятие решения о маршрутизации. Здесь решается, куда пойдет пакет дальше – на какой адрес, через какой сетевой интерфейс и пр.
3	mangle	OUTPUT	Здесь производится внесение изменений в заголовок пакета. Выполнение фильтрации в этой цепочке может иметь негативные последствия.
4	nat	OUTPUT	Эта цепочка используется для трансляции сетевых адресов (NAT) в пакетах, исходящих от локальных процессов брандмауэра.
5	Filter	OUTPUT	Здесь фильтруется исходящий траффик.

6	mangle	POSTROUTING	Цепочка POSTROUTING таблицы mangle в основном используется для правил, которые должны вносить изменения в заголовок пакета перед тем, как он покинет брандмауэр, но уже после принятия решения о маршрутизации. В эту цепочку попадают все пакеты, как транзитные, так и созданные локальными процессами брандмауэра.
7	nat	POSTROUTING	Здесь выполняется Source NAT. Не следует в этой цепочке производить фильтрацию пакетов во избежание нежелательных побочных эффектов. Однако и здесь можно останавливать пакеты, применяя политику по умолчанию DROP.
8			Сетевой интерфейс (например, eth0)
9			Кабель (т.е., Internet)

Порядок прохождения цепочек и таблиц пакетами, предназначенных для локального приложения:

Шаг	Таблица	Цепочка	Примечание
1			Кабель (т.е. Интернет)
2			Входной сетевой интерфейс (например, eth0)
3	mangle	PREROUTING	Обычно используется для внесения изменений в заголовок пакета, например для установки битов TOS и пр.
4	nat	PREROUTING	Преобразование адресов (Destination NAT). Фильтрация пакетов здесь допускается только в исключительных случаях.
5			Принятие решения о маршрутизации.
6	mangle	INPUT	Пакет попадает в цепочку INPUT таблицы mangle. Здесь вносятся изменения в заголовок пакета перед тем как он будет передан локальному приложению.
7	filter	INPUT	Здесь производится фильтрация входящего трафика. Помните, что все входящие пакеты, адресованные нам, проходят через эту цепочку, независимо от того с какого интерфейса они поступили.
8			Локальный процесс/приложение (т.е., программа-сервер или программа-клиент)

Порядок прохождения цепочек и таблиц транзитных пакетов:

Шаг	Таблица	Цепочка	Примечание
1			Кабель (т.е. Интернет)
2			Сетевой интерфейс (например, eth0)

3	mangle	PREROUTING	Обычно эта цепочка используется для внесения изменений в заголовок пакета, например, для изменения битов TOS и пр..
4	nat	PREROUTING	Эта цепочка используется для трансляции сетевых адресов (DNAT). SNAT выполняется позднее, в другой цепочке. Любого рода фильтрация в этой цепочке может производиться только в исключительных случаях
5			Принятие решения о дальнейшей маршрутизации, т.е. в этой точке решается куда пойдет пакет – к локальному приложению или на другой узел сети.
6	mangle	FORWARD	Далее пакет попадает в цепочку FORWARD таблицы mangle, которая должна использоваться только в исключительных случаях, когда необходимо внести некоторые изменения в заголовок пакета между двумя точками принятия решения о маршрутизации.
7	Filter	FORWARD	В цепочку FORWARD попадают только те пакеты, которые идут на другой хост. Вся фильтрация транзитного трафика должна выполняться здесь. Не забывайте, что через эту цепочку проходит трафик в обоих направлениях, обязательно учитывайте это обстоятельство при написании правил фильтрации.
8	mangle	POSTROUTING	Эта цепочка предназначена для внесения изменений в заголовок пакета уже после того как принято последнее решение о маршрутизации.
9	nat	POSTROUTING	Эта цепочка предназначена в первую очередь для SNAT. Не используйте ее для фильтрации без особой на то необходимости. Здесь же выполняется и маскарадинг (Masquerading).
10			Выходной сетевой интерфейс (например, eth1).
11			Кабель (пусть будет LAN).

Таблица Mangle предназначена для внесения изменений в заголовки пакетов (mangle – искажать). Т.е. в этой таблице можно устанавливать биты TOS (Type Of Service) и т.д. Действия: TOS, TTL, MARK.

Таблица Nat используется для выполнения преобразований сетевых адресов NAT. Только первый пакет из потока проходит через цепочки этой таблицы, трансляция адресов или маскировка применяются ко всем последующим

пакетам в потоке автоматически. Для этой таблицы характерны действия: DNAT, SNAT, MASQUERADE.

Таблица Filter содержит наборы правил для выполнения фильтрации пакетов. Пакеты могут пропускаться далее, либо отвергаться (действия ACCEPT и DROP соответственно), в зависимости от их содержимого.

Сохранение правил

```
iptables-save [-c] [-t table]
iptables-save -c > /home/save.txt
```

`-c` – сохраняет счетчики пакетов

Восстановление правил

```
iptables-restore [-c] [-n]
cat /home/save.txt | iptables-restore -c
```

`-c` – восстановление счетчика пакетов

`-n` – сохранение существующих правил

Построение правил

```
iptables [-t table] command [match] [target/jump]
```

Таблицы (`table`) — `nat`, `mangle`, `filter`.

Команды (`command`):

-A – Добавляет новое правило в конец заданной цепочки.

-D – Удаление правила из цепочки по критерию или по номеру.

```
iptables -D INPUT 1.
```

-R — Замена правила другим.

```
iptables -R INPUT 1 -s 192.168.0.1 -j DROP
```

-I — Вставить правило на определенное место.

```
iptables -I INPUT 1 --dport 80 -j ACCEPT
```

-L — Вывод списка правил в заданной цепочке.

```
iptables -L INPUT
```

-F — Сброс (удаление) всех правил из заданной цепочки (таблицы).

-N — Создается новая цепочка с заданным именем в заданной таблице

-X — Удаление заданной цепочки из заданной таблицы

-P — Задает политику по умолчанию для заданной цепочки.

```
iptables -P INPUT DROP
```

-E — выполняет переименование пользовательской цепочки.

```
iptables -E allowed disallowed
```

Критерии (`match`):

-p — типа протокола (TCP, UDP и ICMP)

-s — IP-адрес(а) источника пакета. Знак ! перед IP означает, все кроме IP.

-d — IP-адрес(а) получателя.

-i — Интерфейс, с которого был получен пакет.

-o — Задает имя выходного интерфейса. Только в цепочках OUTPUT, FORWARD и POSTROUTING.

-f — Правило распространяется на все фрагменты фрагментированного пакета.

--sport — Исходный порт, с которого был отправлен пакет.

```
iptables -A INPUT -p tcp --sport 22
```

--dport — Порт или диапазон портов, на который адресован пакет.

--tcp-flags — Определяет маску и флаги tcp-пакета. SYN, ACK, FIN, RST, URG, PSH, ALL и NONE.

```
iptables -p tcp --tcp-flags SYN,FIN,ACK SYN.
```

--icmp-type — Тип сообщения ICMP определяется номером или именем.

```
iptables -A INPUT -p icmp --icmp-type 8
```

--mac-source — MAC адрес сетевого узла, передавшего пакет.

```
iptables -A INPUT -m mac --mac-source 00:00:00:00:00:01
```

--uid-owner — Производится проверка «владельца» по User ID (UID). Например, для блокировки отдельных пользователей.

```
iptables -A OUTPUT -m owner --uid-owner 500
```

--gid-owner — Производится проверка «владельца» пакета по Group ID (GID).

--pid-owner — Производится проверка «владельца» пакета по Process ID (PID).

--state — Проверяется признак состояния соединения (state).

Действия (`target`):

ACCEPT – пакет прекращает движение по цепочке (и всем вызвавшим цепочкам) и считается принятым (т.е. пропускается), тем не менее, пакет

продолжит движение по цепочкам в других таблицах и может быть отвергнут там.

DROP – «сбрасывает» пакет. Пакеты прекращают свое движение полностью, т.е. они не передаются в другие таблицы. После этого действия могут оставаться незакрытые сокеты.

LOG – служит для журналирования отдельных пакетов и событий. В журнал могут заноситься заголовки IP пакетов и другая информация.

SNAT – используется для изменения исходящего IP адреса в IP заголовке пакета. Ключ `--to-source IP`. Только в таблице nat, в цепочке POSTROUTING.

DNAT – используется для преобразования адреса места назначения в IP заголовке пакета. Ключ `--to-destination IP`. Только в цепочках PREROUTING и OUTPUT

MARK – Используется для установки меток для определенных пакетов. `-j MARK --set-mark 2`

REDIRECT – Выполняет перенаправление пакетов и потоков на другой порт той же самой машины. К примеру, можно пакеты, поступающие на HTTP порт перенаправить на порт HTTP proxy.

```
iptables -t nat -A PREROUTING -p tcp --dport 80 -j
REDIRECT --to-ports 8080
```

REJECT – в тех же самых ситуациях, что и DROP, но в отличие от DROP, команда REJECT выдает сообщение об ошибке на хост, передавший пакет. Может использоваться только в цепочках INPUT, FORWARD и OUTPUT (и во вложенных в них)

```
iptables -A FORWARD -p TCP --dport 22 -j REJECT --
reject-with tcp-reset
```

Ключ --reject-with указывает, какое сообщение необходимо передать в ответ, если пакет совпал с заданным критерием. Значение tcp-reset передает сегмент с флагом RST, что позволяет закрыть соединение.

RETURN – прекращает движение пакета по текущей цепочке правил и производит возврат в вызывающую цепочку. Если цепочка лежит на верхнем уровне (например, INPUT), то к пакету будет применена политика по умолчанию (ACCEPT или DROP).

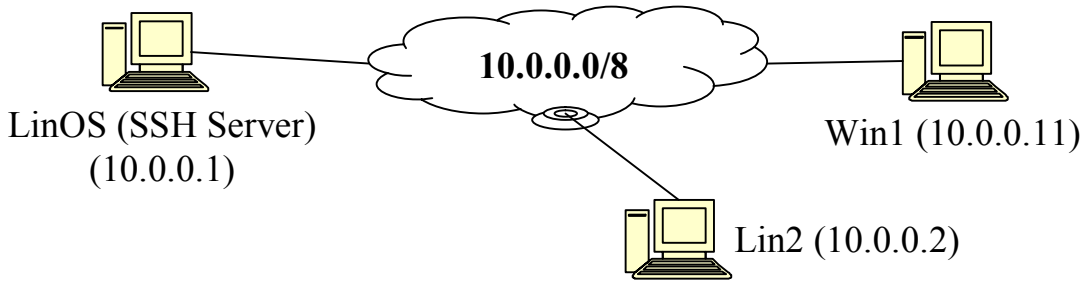
Виртуальные машины:

Win1 (HDD: 4Gb; RAM: 128Mb; LAN0: internal; ОС Windows XP)

Lin2 (HDD: –; RAM: 64Mb; LAN0: internal; ОС SLAX Linux)

LinOS (HDD: 8 Gb; RAM: 256Mb; LAN0: internal; ОС: openSuSE)

Схема сети:



Последовательность действий:

1. Включите виртуальную машину LinOS и Win1.
2. Настройте сетевые интерфейсы обоих машин.
3. Проверьте прохождение эхо-запросов между обоими узлами.
4. На машине LinOS проверьте установку пакетов ssh:

   ```
   #rpm –qa | grep openssh
   ```

5. Установите на машине LinOS демон SSHD, для доступа к узлу по протоколу ssh.

   ```
   #apt-get install openssh-*
   ```

   Или

   ```
   #zypper install openssh-*
   ```

6. Запустите демон SSHD:

   ```
   #/etc/init.d/sshd start
   ```

7. Подключитесь к LinOS локально через протокол ssh. Для этого в консоли на LinOS введите команду `ssh`, замените username на имя пользователя в ОС и hostname, на имя машины в сети. При запросе соединения без использования ключа ответьте yes. При запросе пароля введите пароль от учетной записи username:

   ```
   #ssh username@hostname
   #The authenticity of host (…) connection (yes/no)yes
   #password:
   ```

8. Убедитесь в успешном входе. Узнайте текущий каталог, просмотрите его содержимое командой `dir`.

9. Закройте подключение по протоколу ssh, дав команду exit:

   ```
   #exit
   ```

10. Установите утилиту putty или любой другой клиент ssh на узел Win1.
11. Укажите параметры подключения

    Host Name (or IP address): 10.0.0.1

    Port: 22

    Connection type: SSH

    Window | Translation | Remote character set: UTF-8

12. Щелкните по кнопке **Open**.
13. Проанализируйте результат подключения.
14. Откройте порт на машине LinOS для демона SSHD путем добавления правила, разрешающего принятие tcp-трафика по 22 порту.

    ```
    #iptables –I INPUT 1 –p tcp --dport 22 –j ACCEPT
    ```

    Или (с использованием дополнительной цепочки):

    ```
    #iptables –i eth0 –j input_ext
    ```
    (все, что на интерфейс eth0 пропускаем через цепочку input_ext)

    ```
    #iptables –I input_ext 1 –p tcp --dport 22 –j ACCEPT
    ```
    (вставляем правило на первое место. Так как последнее правило отбрасывает все.)
15. Подключитесь через ssh-клиент putty с машины Win1 к узлу LinOS, аналогично пункту 11.
16. Проверьте успешность установления соединения.
17. Перезагрузите машину LinOS
18. Просмотрите список правил iptables командой:

    ```
    #iptables-save
    ```

19. Убедитесь в отсутствии ранее добавленного правила.
20. Откройте на редактирование файл /etc/sysconfig/SuSEfirewall2:

    ```
    #vim /etc/sysconfig/SuSEfirewall2
    ```

21. Добавьте параметр, описывающий имя демонов, к которым разрешен трафик с недоверенных сетей (untrusted networds):

    ```
    FW_CONFIGURATIONS_EXT = "sshd"
    ```

22. Сохраните файл (Esc, Z, Z).
23. Перезагрузите машину LinSrv
24. Подключитесь к узлу LinOS по протоколу ssh с узла Win1.

25. Откройте для редактирования файл `/etc/sysconfig/SuSEfirewall2`:

```
#vim /etc/sysconfig/SuSEfirewall2
```

26. Восстановите параметр `FW_CONFIGURATIONS_EXT` в исходное состояние.

27. Добавьте параметр `FW_CUSTOMRULES`, в котором укажите путь к файлу с пользовательскими правилами iptables.

```
FW_CUSTOMRULES =
"/etc/sysconfig/scripts/SuSEfirewall2-custom"
```

28. Откройте для редактирования файл с пользовательскими правилами iptables:

```
#vim /etc/sysconfig/scripts/SuSEfirewall2-custom
```

29. Отредактируйте файл путем добавления в секцию правила, открывающего порт 22:

```
fw_custom_before_port_handling() {
    iptables -I INPUT 1 -p tcp --dport 22 -j ACCEPT
}
```

30. Сохраните файл (Esc, Z, Z).

31. Перезагрузите машину LinOS

32. Подключитесь к узлу LinOS по протоколу ssh с узла Win1.

33. Подключите Lin2 в сеть.

34. Проверьте подключение к LinOS по протоколу ssh с узла Lin2.

Практическая работа 10. Настройка DHCP-сервера (Windows)

Цель работы: Изучить функционирование DHCP-сервера в ОС Window Server. Получить навыки настройки DHCP-сервера в ОС Windows Server, а также сетевых адаптеров на клиентских узлах через протокол DHCP.

Объем времени: 1-2 ч. (с учетом настройки виртуальных машин, но без учета установки ОС. Рекомендуется заранее установить ОС в виртуальную машину и копировать файл виртуального диска на рабочие места).

Программное обеспечение: ISO-образ ОС Windows XP, ISO-образ ОС Windows Server 2003, ISO-образ ОС SLSX Linux Live CD.

Виртуальные машины:

WinSrv (HDD: 4Gb; RAM: 128Mb; LAN0: internal; ОС: Windows Server 2003),

Win2 (HDD: 4Gb; RAM: 128Mb; LAN0: internal; ОС Windows XP),

LinOS (HDD: 8 Gb; RAM: 256Mb; LAN0: internal; ОС: openSuSE).

Lin2 (HDD: нет; RAM: 128Mb; LAN0: internal; ОС: SLAX),

Схема сети:

Необходимые команды:

`ipconfig /all`	Выводит информацию обо всех сетевых интерфейсах компьютера (ОС Windows)
`ipconfig /release`	Освобождение аренды IP-адреса
`ipconfig /renew`	Принудительное обновление аренды
`ifconfig -a`	Выводит информацию обо всех сетевых интерфейсах компьютера (ОС Linux)
`ping <хост>`	Посылает сообщение ICMP с эхо-запросом узлу с указанным IP-адресом или DNS-именем.

Последовательность действий:

1. Включите виртуальную машину WinSrv, включите брандмауэр Windows.
2. Настройте сетевой адаптер сервера, аналогично действиям в предыдущей работе (Настройка сетевых адаптеров (Windows), см. стр.154)
3. Добавьте серверу роль DHCP-сервер, чтобы установить в ОС специальные службы и компоненты. Щелкните **Пуск** | **Панель управления** | **Администрирование | Управление данным сервера** | **Добавить роль** | **Особая конфигурация | Далее | DHCP-сервер | Далее**. При установке, возможно, потребуется CD-диск с дистрибутивом операционной системы. Для этого в виртуальной машине примонтируйте ISO-образ с дистрибутивом.
4. Если после установки открылся мастер настройки области, перейдите к следующему шагу.

a. Чтобы самостоятельно открыть мастер создания области откройте оснастку DHCP (**Управление DHCP**: **Пуск** | **Панель управления** | **Администрирование** | **DHCP**).

b. В левой части окна выберите сервер.

c. В контекстном меню сервера выберите команду **Создать область…** .

5. В мастере создания области:

   a. Щелкните по кнопке **Далее**, чтобы начать создание новой области.

   b. Введите имя области, например, `clients`. Щелкните по кнопке **Далее**.

   c. Введите начальный и конечный IP-адрес пула адресов, а также маску сети. Щелкните по кнопке **Далее**.

   d. На следующей странице при необходимости можете указать исключения IP-адресов из указанного пула. Щелкните по кнопке **Далее**.

   e. Укажите срок аренды для IP-адресов. Например, 1 день. Щелкните по кнопке **Далее**.

   f. Выберите пункт **Настроить эти параметры позже**, поскольку в данной работе не используется шлюз и DNS-сервера. Щелкните по кнопке **Далее**.

   g. Щелкните во всех окна **Готово**.

6. В дереве в левой части окна в DHCP-сервере выделите созданную область IP-адресов. В контекстном меню области выберите пункт **Активировать**.

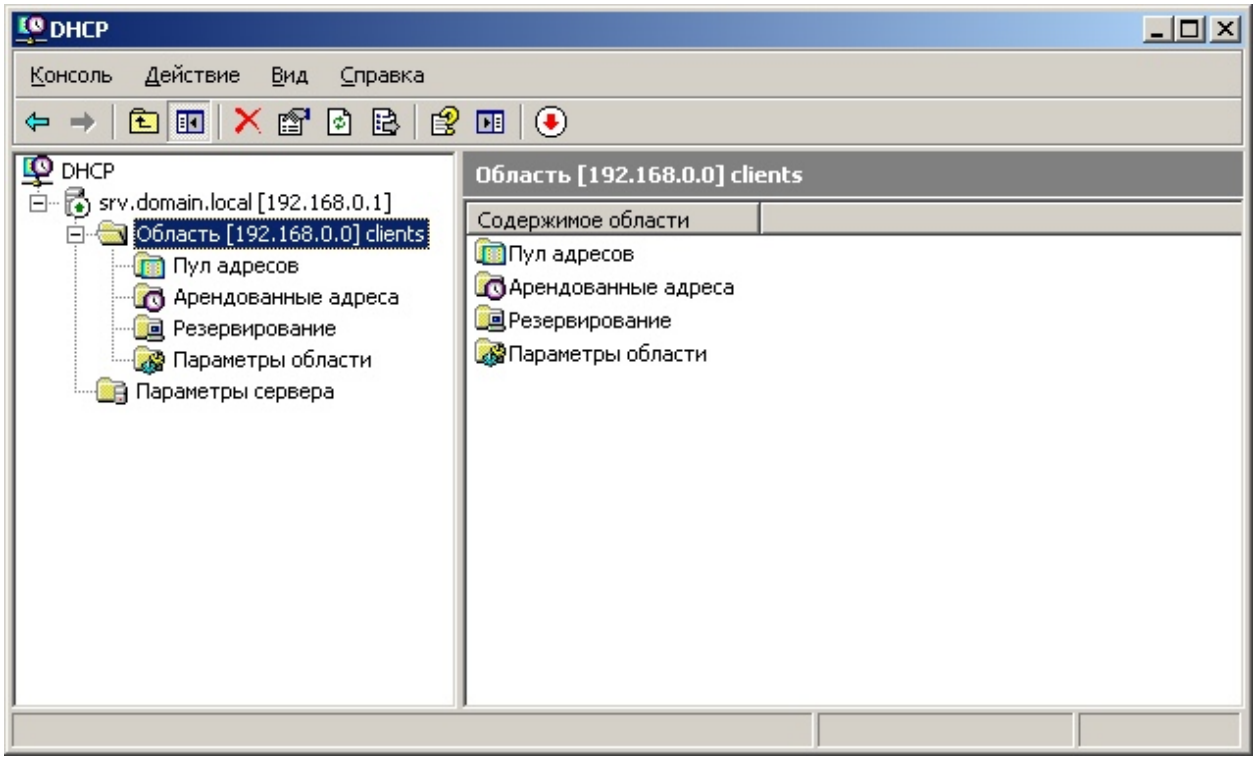


Рис. 21. Окно оснастки DHCP-сервера

7. Запустите клиентский узел Win2.
8. Проверьте текущие настройки сетевых интерфейсов командой `ipconfig`.
9. Откройте окно настройки сетевого интерфейса: **Пуск | Панель управления | Сетевые подключения**. Выберите локальный сетевой интерфейс. Правым щелчком мыши по пункту сетевого интерфейса вызовите контекстное меню, выберите пункт **Свойства**.
10. Выберите в списке протоколов **Протокол Интернета (TCP/IP)** и щелкните по кнопке **Свойства**.
11. Установите переключатель **Автоматически получить IP-адрес**.
12. Щелкните в окнах Ок, а в окне свойств сетевого интерфейса **Закрыть**.
13. Проверьте настройки сетевых интерфейсов командой `ipconfig`.
14. Откройте в брандмауэре порты 67/UDP и 68/UDP.
    a. Откройте окно настройки сетевого экрана (**Панель управления | Брандмауэр Windows**).
    b. Перейдите на вкладку **Исключения**.
    c. Щелкните по кнопке **Добавить порт…** .
    d. Введите символическое название, например, dhcp67.
    e. Укажите порт 67, установите переключатель UDP.
    f. Проделайте аналогичные действия для порта 68/UDP.
    g. Щелкните по кнопке **Ок** во всех открытых окнах.

15. Обновите аренду адреса и проверьте настройки сетевых интерфейсов командой `ipconfig`. Убедитесь в том, что IP-адрес был выдан DHCP-сервером.

```
>ipconfig /renew
>ipconfig /all
```

16. На узле WinSrv, откройте оснастку **Управление DHCP**.
17. Откройте раздел **Арендованные адреса**.
18. Убедитесь, что сервер выдал IP-адрес в аренду клиенту. Убедитесь, что MAC адрес на сервере соответствует MAC адресу клиента Win2.
19. Проверьте прохождение эхо-запросов с сервера до клиента и наоборот.
20. Освободите аренду на клиенте Win2

```
>ipconfig /release.
```

21. Проверьте настройки сетевых интерфейсов командой `ipconfig`.
22. Обновить аренду IP-адреса командой:

```
>ipconfig /renew.
```

23. Запустите клиент Lin2.
24. Проверьте сетевые настройки командой `ifconfig`:

```
#ifconfig -a
```

25. Убедитесь, что DHCP-сервер выдал IP-адрес.
26. Выключите клиент Lin2.
27. Убедитесь, что аренда адреса сохранилась.
28. Включите клиент LinOS.
29. Проверьте настройки сетевого интерфейса LinOS командой `ifconfig`.
30. Настройте постоянную конфигурацию сети с помощью конфигурационного файла. Найдите файл (в различных дистрибутивах и версиях путь и имя файла может отличаться, а в место X может быть как MAC адрес карты, так и символьное имя интерфейса, например eth0):

```
/etc/sysconfig/network/ifcfg-eth-id-XX:XX:XX:XX:XX)
```

31. Откройте файл в консольном текстовом редакторе VIM от имени root командой:

```
#cd /etc/sysconfig/network/
#vim ifcfg-eth-id-XX:XX:XX:XX:XX
```

32. Нажмите клавишу **i**, для перехода в режим редактирования. Отредактируйте файл:

```
ONBOOT=yes
```

```
BOOTPROTO=dhcp
```

33. Нажмите на клавиатуре клавишу **Esc**, чтобы перейти в режим команд.
34. Нажмите на клавиатуре клавишу **Z, Z**, для сохранения файла.
35. Перезагрузите узел LinOS (openSuSE).
36. Проверьте настройки сетевых интерфейсов командой `ifconfig`.
37. Убедитесь в успешной выдаче IP-адреса узлу. Убедитесь в том, что адрес является новым и не совпадает с тем, который был выдан Lin2.
38. Проверьте прохождение эхо-запросов между всеми клиентами и сервером.

Практическая работа 11. Настройка DHCP-сервера (Linux)

Цель работы: Изучить функционирование DHCP-сервера в ОС Linux. Получить навыки настройки DHCP-сервера в ОС Linux, а также сетевых интерфейсов на клиентских узлах через протокол DHCP.

Объем времени: 1-2 ч.

Программное обеспечение: ISO-образ ОС Windows XP, ISO-образ ОС SLAX Linux Live CD, ISO-образ ОС openSuSE.

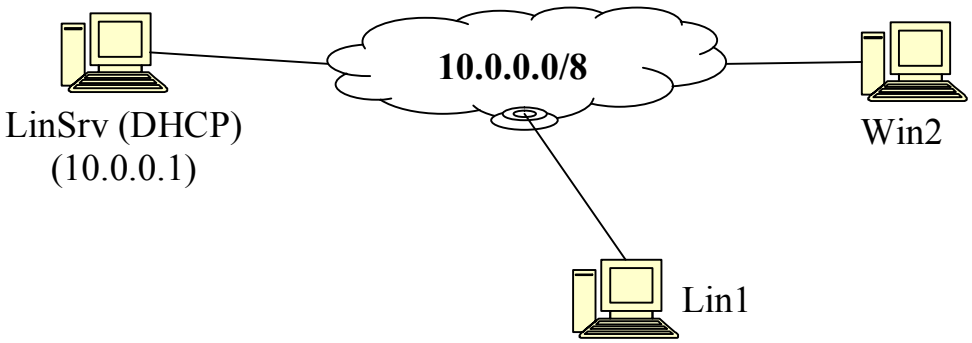
Виртуальные машины:

Win2 (HDD: 4Gb; RAM: 128Mb; LAN0: internal; ОС Windows XP),

Lin1 (HDD: нет; RAM: 128Mb; LAN0: internal; ОС: SLAX),

LinSrv (HDD: 8 Gb; RAM: 256Mb; LAN0: internal; ОС: openSuSE), является копией виртуальной машины LinOS.

Схема сети:



Необходимые команды:

`ipconfig /all`	Выводит информацию обо всех сетевых интерфейсах компьютера (ОС Windows)

`ipconfig /release`	Освобождение аренды IP-адреса
`ipconfig /renew`	Принудительное обновление аренды
`ifconfig -a`	Выводит информацию обо всех сетевых интерфейсах компьютера (ОС Linux)
`ping <хост>`	Посылает сообщение ICMP с эхо-запросом узлу с указанным IP-адресом или DNS-именем.
`apt-get install` *`имя_пакета`*	Установка пакета из имеющихся репозиториев (ОС Linux)
`zypper install` *`имя_пакета`*	Установка пакета с помощью менеджера пакетов zypper из имеющихся репозиториев (ОС Linux openSuSE)

Последовательность действий:

1. Установите демон `dhcpd` на узел LinSrv. Для этого в консоли от имени root дайте команды:

   ```
   #apt-get install dhcpd
   ```

   Или

   ```
   #zypper install dhcpd
   ```

2. Откройте файл `/etc/sysconfig/dhcpd` в редакторе vim:

   ```
   #vim /etc/sysconfig/dhcpd
   ```

3. Добавьте информацию о том, на каком интерфейсе работает DHCP, отредактировав параметр:

   ```
   DHCPD_INTERFACE=”eth0”
   ```

4. Откройте на редактирование файл `/etc/dhcpd.conf` в редакторе vim:

   ```
   #vim /etc/dhcpd.conf
   ```

5. Установите параметры DHCP-сервера:

   ```
   #глобальный параметры:
   option domain-name “example.com”
   option domain-name-servers ns1.example.com
   option subnet-mask 255.0.0.0
   default-leases-time 600 #время аренды по умолчанию
   max-lease-time 7200 #максимальное время аренды
   ```

6. Определите пул адресов 10.0.0.11–10.0.0.20 в секции:

   ```
   subnet 10.0.0.0 netmask 255.0.0.0 {
     range 10.0.0.11 10.0.0.20
   ```

```
    option broadcast-address 10.255.255.255
    option routers 10.0.0.1
}
```

7. Нажмите на клавиатуре клавишу Esc, чтобы перейти в режим команд
8. Нажмите на клавиатуре Z, Z для сохранения файла.
9. Перезапустите демон DHCPD

```
#/etc/init.d/dchpd restart
```

10. Добавьте правила iptables разрешающие UDP-трафик на порты 67 и 68.

```
#iptables –I INPUT 1 –p udp --dport 67 –j ACCEPT
#iptables –I INPUT 1 –p udp --dport 68 –j ACCEPT
```

11. Запустите машину Lin1
12. Проверьте успешность выдачи IP адреса командой `ifconfig`.
13. Проверьте выдачу аренда IP-адреса на DHCP-сервере в файле `/var/lib/dhcp/db/dhcpd.leases`.
14. Проверьте прохождение эхо-запросов между всеми клиентами и DHCP-сервером.

Практическая работа 12. Маршрутизация пакетов (Windows)

Цель работы: Изучить принципы маршрутизации пакетов. Получить навыки настройки маршрутизации пакетов в ОС Windows, а также настройки сетевых интерфейсов для доступа в другие сети.

Объем времени: 2-4 ч.

Программное обеспечение: ISO-образ ОС Windows XP.

Виртуальные машины:

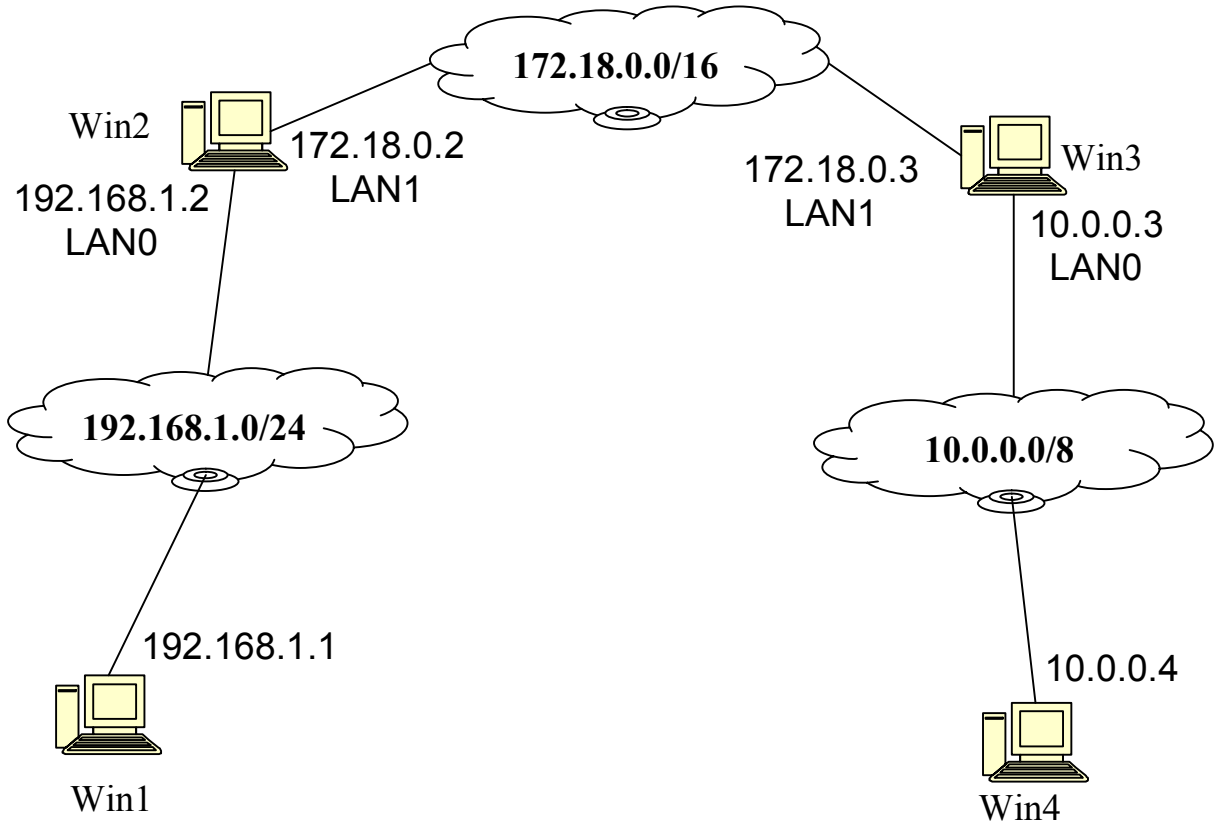
Win1 (HDD: 4Gb; RAM: 128Mb; LAN0: internal; ОС Windows XP)

Win2 (HDD: 4Gb; RAM: 128Mb; LAN0, LAN1: internal; ОС Windows XP),

Win3 (HDD: 4Gb; RAM: 128Mb; LAN0, LAN1: internal; ОС Windows XP),

Win4 (HDD: 4Gb; RAM: 128Mb; LAN0: internal; ОС Windows XP),

Схема сети:



Необходимые команды:

`ipconfig /all`	Выводит информацию обо всех сетевых интерфейсах компьютера (ОС Windows)
`tracert` *IP*	Трассировка маршрута с выводом ICMP-ответов промежуточных маршрутизаторов.
`ping <хост>`	Посылает сообщение ICMP с эхо-запросом узлу с указанным IP-адресом или DNS-именем.
`route print`	Выводит таблицу маршрутизации узла (ОС Windows)
`route add <`*IP*`> mask <`*mask*`> <`*GW_IP*`> IF <`*интерфейс*`>`	Добавляет маршрут в сеть *IP* с маской *mask* на шлюз *GW_IP* через *интерфейс*.

Последовательность действий:

1. Включите виртуальные машины Win1 и Win2.
2. Настройте сетевые интерфейсы Win1(LAN0) и Win2(LAN0).
3. Проверьте прохождение эхо-запросов между узлами Win1 и Win2.
4. Включите виртуальную машину Win3.
5. Настройте сетевые интерфейсы Win2 (LAN1) и Win3 (LAN1).

6. Проверьте прохождение эхо-запросов между узлами Win2 и Win3.
7. Проверьте прохождение эхо-запросов между узлами Win1 и Win3. Убедитесь в том, что эхо-запросы не проходят.
8. Проверьте трассировку маршрута между узлами Win1 и Win3.

   ```
   >tracert 172.18.0.3
   ```

9. Настройте на клиенте Win1 шлюз по умолчанию.
   a. Для этого в свойствах сетевого интерфейса (**Пуск | Панель управления | Сетевые подключения**, в контекстном меню сетевого подключения выберите пункт **Свойства**).
   b. Откройте окно настройки протокола IP.
   c. В пункте **Основной шлюз** укажите: 192.168.1.2.
   d. Щелкните по кнопке **Ок**, и в окне настроек сетевого подключении по кнопке **Закрыть**.
10. Проверьте прохождение эхо-запросов с узла Win1 до Win3. Убедитесь в том, что эхо-запросы не проходят.
11. Проверьте трассировку маршрута между узлами Win1 и Win3.

    ```
    >tracert 172.18.0.3
    ```

12. Просмотрите на узле Win2 таблицу маршрутизации с помощью команды `route`:

    ```
    >route print
    ```

13. Убедитесь в наличии маршрутов для ближайших сетей (192.168.1.0/24 и 172.18.0.0/16).
14. Включите на узле Win2 службу маршрутизации и удаленного доступа.
    a. Откройте окно служб: **Пуск | Панель управления | Администрирование | Службы**.
    b. Найдите в списке службу **Маршрутизация и удаленный доступ** (**Routing and Remote access**). Двойным щелчком по пункту в списке откройте окно настройки запуска службы.
    c. В окне настройки службы на вкладке **Общая** в пункте **Тип запуска** в выпадающем меню выберите типа запуска **Авто**.
    d. Щелкните в окне службы по кнопке **Пуск**.
    e. Закройте окно настройки служб.
15. Проверьте прохождение эхо-запросов из одной сети в другую.
16. Проверьте трассировку маршрута между узлами Win1 и Win3.

    ```
    >tracert 172.18.0.3
    ```

17. Сравните результаты трассировки с результатами из пункта 11.

18. Добавьте маршрут для сети 192.168.1.0/24 в таблицу маршрутизации узла Win3:

```
> route add 192.168.1.0 mask 255.255.255.0 172.18.0.2 IF 2
```

19. Проверьте прохождение эхо-запросов из одной сети в другую.
20. Проверьте трассировку маршрута между узлами Win1 и Win3. Проанализируйте значения TTL.
21. Включите четвертый узел Win4
22. Настройте на Win4 сетевой интерфейс
23. Проверьте прохождение эхо-запросов до Win3:

```
>ping 10.0.0.3
```

24. Настройте шлюз по умолчанию на узле Win4. Шлюз по умолчанию 10.0.0.3.
25. Включите службу маршрутизации и удаленного доступа на узле Win3.
26. Проверьте прохождение эхо-запросов между узлами Win1 и Win2.
27. Проверьте трассировку маршрута между узлами Win1 и Win2.
28. Добавьте маршрут для сети 10.0.0.0/8 в таблицу маршрутизации узла Win2:

```
> route add 10.0.0.0 mask 255.0.0.0 172.18.0.3 IF 2
```

29. Проверьте прохождение эхо-запросов между узлами Win1 и Win2.
30. Проверьте трассировку маршрута между узлами Win1 и Win2.
31. Проверьте прохождение эхо-запросов между узлами Win1 и Win4.
32. Проверьте трассировку маршрута между узлами Win1 и Win4.
33. Проанализируйте значения TTL в эхо-ответах.

Практическая работа 13. Маршрутизация пакетов (Linux)

Цель работы: Изучить принципы маршрутизации пакетов. Получить навыки настройки маршрутизации пакетов в ОС Linux, а также настройки основного шлюза сетевых интерфейсов для доступа в другие сети.

Объем времени: 2-4 ч.

Программное обеспечение: ISO-образ ОС SLAX Linux Live CD.

Виртуальные машины:

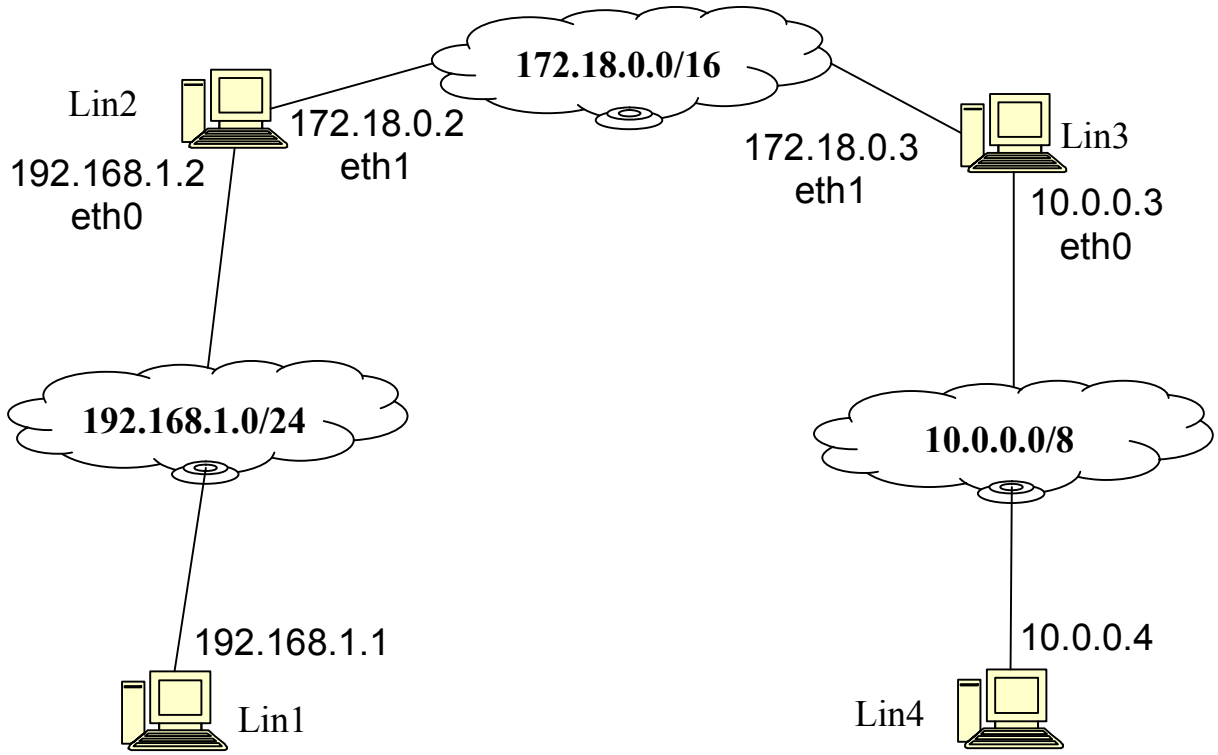
Lin1 (HDD: –; RAM: 64Mb; LAN0: internal; ОС SLAX Linux)

Lin2 (HDD: –; RAM: 64Mb; LAN0, LAN1: internal; ОС SLAX Linux),

Lin3 (HDD: –; RAM: 64Mb; LAN0, LAN1: internal; ОС SLAX Linux),

Lin4 (HDD: –; RAM: 64Mb; LAN0: internal; ОС SLAX Linux).

Схема сети:



Необходимые команды:

`ifconfig -a`	Выводит информацию обо всех сетевых интерфейсах компьютера (ОС Windows)
`traceroute` *IP*	Трассировка маршрута с выводом ICMP-ответов промежуточных маршрутизаторов.
`ping <хост>`	Посылает сообщение ICMP с эхо-запросом узлу с указанным IP-адресом или DNS-именем.
`netstat -rn`	Выводит статистику сетевых подключений
`route`	Выводит таблицу маршрутизации узла
`ip route show`	Выводит таблицу маршрутизации узла. Не требует root-привилегий.
`route add default gw` *<IP> <ethX>*	Добавляет шлюз по умолчанию с *IP* для интерфейса *ethX*.
`route -v add -net` *<NET_ADDR>/<NETMASK>* `gw` *<GW_IP>* `dev`	Добавляет маршрут в сеть с адресом *NET_ADDR* с маской *NET_MASK* (в десятичном виде) на шлюз *GW_IP* через

<ethX>	интерфейс *ethX*.

Последовательность действий:

1. Включите виртуальные машины Lin1 и Lin2.
2. Настройте сетевой интерфейс Lin1(eth0)

   ```
   #ifconfig eth0 192.168.1.1 netmask 255.255.255.0 up
   ```
3. Настройте сетевой интерфейс Lin2(eth0).

   ```
   #ifconfig eth0 192.168.1.2 netmask 255.255.255.0 up
   ```
4. Проверьте прохождение эхо-запросов между узлами Lin1 и Lin2.
5. Включите виртуальную машину Lin3.
6. Настройте сетевой интерфейс Lin2 (eth1)

   ```
   #ifconfig eth1 172.18.0.2 netmask 255.255.0.0 up
   ```
7. Настройте сетевой интерфейс Lin3 (eth1)

   ```
   #ifconfig eth1 172.18.0.3 netmask 255.255.0.0 up
   ```
8. Проверьте прохождение эхо-запросов между узлами Lin2 и Lin3.
9. Проверьте прохождение эхо-запросов между узлами Lin1 и Lin3. Убедитесь в том, что эхо-запросы не проходят.
10. Проверьте трассировку маршрута между узлами Lin1 и Lin3.

    ```
    #traceroute 172.18.0.3
    ```
11. Настройте на клиенте Lin1 шлюз по умолчанию:

    ```
    #route add default gw 192.168.1.2 eth0
    ```

    Или через утилиту ip:

    ```
    #ip route add default via 192.168.1.2
    ```
12. Проверьте прохождение эхо-запросов с узла Lin1 до Lin3. Убедитесь в том, что эхо-запросы не проходят.
13. Проверьте трассировку маршрута между узлами Lin1 и Lin3.

    ```
    #traceroute 172.18.0.3
    ```
14. Просмотрите на узле Lin2 таблицу маршрутизации с помощью команды `route`:

    ```
    #route
    ```

    Или утилитой ip:

    ```
    #ip route show
    ```
15. Убедитесь в наличии маршрутов для ближайших сетей (192.168.1.0/24 и 172.18.0.0/16).
16. Включите на узле Lin2 маршрутизацию пакетов.

a. Вариант 1. Включить продвижение пакетов в системном конфигурационном файле `/etc/sysctl.conf`, добавив параметр:

   ```
   net.ipv4.ip_forward = 1
   ```

b. Вариант 2. Включить продвижение пакетов в параметрах ядра:

   ```
   #echo '1' > /proc/sys/net/ipv4/ip_forward
   ```

   Проверьте включение маршрутизации командой:

   ```
   #cat /proc/sys/net/ipv4/ip_forward
   ```

17. Проверьте прохождение эхо-запросов из одной сети в другую.
18. Проверьте трассировку маршрута между узлами Lin1 и Lin3.

    ```
    #traceroute 172.18.0.3
    ```

19. Сравните результаты трассировки с результатами из пункта 10.
20. Добавьте маршрут для сети 192.168.1.0/24 в таблицу маршрутизации узла Lin3:

    ```
    #route –v add –net 192.168.1.0/24 gw 172.18.0.2 dev eth1
    ```

    Или через утилиту ip:

    ```
    #ip route add 192.168.1.0/24 via 172.18.0.2
    ```

    Чтобы удалить неправильный маршрут используйте команду:

    ```
    #route –v del –net 10.0.0.0/8
    ```

    Аналогично, чтобы удалить маршрут через утилиту ip:

    ```
    #ip route del 192.168.1.0/24
    ```

21. Проверьте прохождение эхо-запросов из одной сети в другую.
22. Проверьте трассировку маршрута между узлами Lin1 и Lin3. Проанализируйте значения TTL в эхо-ответах.
23. Включите четвертый узел Lin4
24. Настройте на Lin4 сетевой интерфейс

    ```
    #ifconfig eth0 10.0.0.4 netmask 255.0.0.0 up
    ```

25. Проверьте прохождение эхо-запросов до Lin3:

    ```
    #ping 10.0.0.3
    ```

26. Настройте шлюз по умолчанию на узеле Lin4. Шлюз по умолчанию 10.0.0.3.

    ```
    #route add default gw 10.0.0.3 eth0
    ```

27. Включите службу маршрутизации и удаленного доступа на узле Lin3.
28. Проверьте прохождение эхо-запросов между узлами Lin1 и Lin2.
29. Проверьте трассировку маршрута между узлами Lin1 и Lin2.

30. Добавьте маршрут для сети 10.0.0.0/8 в таблицу маршрутизации узла Lin2:

```
#route -v add -net 10.0.0.0/8 gw 172.18.0.3 dev eth1
```

31. Проверьте прохождение эхо-запросов между узлами Lin1 и Lin2.
32. Проверьте трассировку маршрута между узлами Lin1 и Lin2.
33. Проверьте прохождение эхо-запросов между узлами Lin1 и Lin4.
34. Проверьте трассировку маршрута между узлами Lin1 и Lin4.
35. Проанализируйте значения TTL в эхо-ответах.

Практическая работа 14. Организация сетей Microsoft Network на основе рабочих групп (Windows)

Цель работы: Изучить принципы организации сетей Microsoft Network на основе рабочих групп (workgroup). Получить навыки настройки сетей Microsoft Network в ОС Windows по протоколу NetBIOS/SMB.

Объем времени: 1 ч.

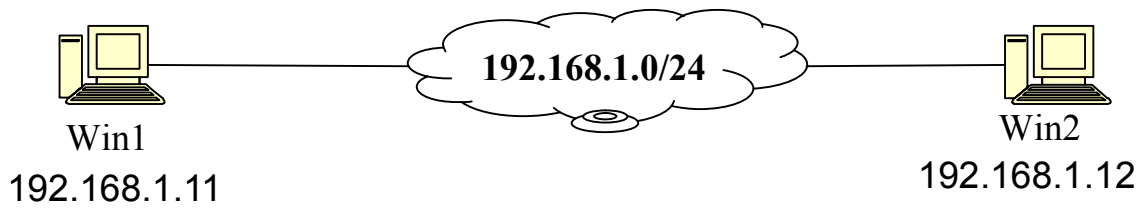
Программное обеспечение: ISO-образ ОС Windows XP.

Виртуальные машины:

Win1 (HDD: 4Gb; RAM: 128Mb; LAN0: internal; ОС Windows XP)

Win2 (HDD: 4Gb; RAM: 128Mb; LAN0: internal; ОС Windows XP)

Схема сети:



Последовательность действий:

1. Запустите машину Win1 и Win2
2. Настройте сетевые интерфейсы обоих узлов (IP и маску подсети). Проверьте успешное прохождение эхо-запросов.
3. Запустить мастер **Network Setup Wizard** на машине Win1. Щелкните по кнопке **Далее** для старта мастера.
4. Щелкните по кнопке **Далее** для продолжение настройки сети.

5. В окне **Выбора метода подключения**, выберите пункт **Этот компьютер подключен к Интернет другой компьютер в моей сети**. Щелкните по кнопке **Далее**.
6. Укажите имя компьютера Win1 и его описание. Щелкните по кнопке **Далее**.
7. Укажите название рабочей группы Workgroup. Щелкните по кнопке **Далее**.
8. Установите переключатель на пункте **Включить службу доступа к файлам и принтерам**. Щелкните по кнопке **Далее**.
9. Просмотрите указанные настройки на странице еще раз. Щелкните по кнопке **Далее**. Дождитесь завершения работы мастера.
10. На последней странице мастера выберите пункт **Просто завершить работу мастера**. Щелкните по кнопке **Далее**.
11. Проделайте аналогичную настройку сети на машине Win2.
12. Проверить функционирование общего доступа, для этого откройте окно Сетевое окружение, выберите пункт **Вся сеть** (**Entire Network**). Откройте рабочую группу Workgroup. Убедитесь в наличии в сети двух машин.
13. Создайте папку `Общая` на машине Win1 на диске C.
14. Откройте общий доступ к этой папке, щелкнув по пункту **Доступ…** в контекстном меню каталога в файловом менеджере.
15. Убедитесь в отображении ресурсов машин Win1 в сетевом окружении с машины Win2.
16. На машине Win1 просмотрите открытые порты с помощью команды:

```
>netstat –a
```

17. На машине Win1 отключить Службу доступа к файлам и принтерам (**Панель управления | Администрирование | Службы | Службу доступа к файлами и принтерам**, установите в пункте **Типа запуска**: **Отключено**).
18. Проверить отображение общих ресурсов этого компьютера с машины Win2.
19. Просмотрите открытые порты на Win1 с помощью команды netstat.

```
>netstat –a
```

20. Включите службу доступа к файлам и принтерам на машине Win1.
21. На машине Win1 отключите порты Службы доступа к файлам и принтерам в брандмауэре (**Панель управления | Брандмауэр Windows**, вкладка **Исключения**, снимите флажок **Служба доступа к файлам и принтерам**).
22. Проверьте отображение узлов в сетевом окружении и доступ к общим ресурсам к Win1 с машины Win2.

23. На машине Win1 откройте порты Службы доступа к файлам и принтерам в брандмауэре.
24. На машине Win1 откройте окно настройки сетевого интерфейса (**Панель управления** | **Сетевые подключения**, в контекстном меню сетевого подключения выберите пункт **Свойства**).
25. Отключите клиент для сетей Microsoft на данном интерфейсе снятием соответствующего флажка (**Клиент для сетей Microsoft**).
26. Проверьте отображение компьютеров в сети с машины Win2.
27. Включите клиент для сетей Microsoft на машине Win1.

Практическая работа 15. Настройка службы Samba (Linux)

Цель работы: Изучить особенности работы службы Samba и клиента службы Samba для организации взаимодействия с сетями Microsoft Network. Получить навыки настройки клиента Samba и сервера Samba для подключения к сетям Microsoft Network по протоколу NetBIOS/SMB.

Объем времени: 1 ч.

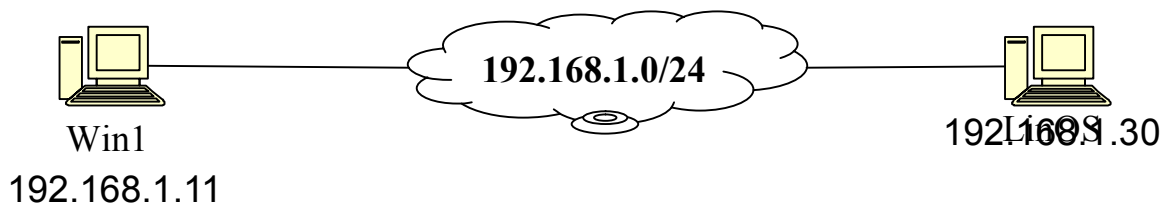
Программное обеспечение: ISO-образ ОС openSuSE, ISO-образ ОС Windows XP.

Виртуальные машины:

Win1 (HDD: 4Gb; RAM: 128Mb; LAN0: internal; ОС Windows XP)

LinOS (HDD: 8 Gb; RAM: 256Mb; LAN0: internal; ОС: openSuSE)

Схема сети:



Последовательность действий:

1. Включите Win1, настройте сетевой интерфейс и клиент сетей Microsoft Network.

2. Включите машину с ОС Linux, настройте сетевой интерфейс.
3. Установите на машину с ОС Linux пакеты samba-common, samba-client, samba:

```
#apt-get install samba samba-*
```

или

```
#zypper install samba samba-*
```

4. На машине с ОС Linux откройте на редактирование файл `/etc/samba/smb.conf`.

```
#vim /etc/samba/smb.conf
```

5. Отредактируйте файл, включив настройки о рабочей группе и аккаунте для входа на другие узлы:

```
workgroup = WORKGROUP
guest account = guest
security = share
```

6. Сохраните файл (Esc, Z, Z)
7. Перезагрузите сервер samba командой

```
#/etc/init.d/smb restart
```

8. Откройте порты (TCP 139, 445; UDP 137, 138) в сетевом экране на машине с ОС Linux для доступа к службе Samba извне.
    a. Вариант 1. **YaST2** | **Безопасность и пользователи** | **Брандмауэр** | **Разрешенные службы** | **Netbios server** | **Добавить**. Примените все настройки в окнах.
    b. Вариант 2. Через iptables:

```
#iptables –I INPUT 1 –p tcp --dport 139 –j ACCEPT
#iptables –I INPUT 1 –p tcp --dport 445 –j ACCEPT
#iptables –I INPUT 1 –p udp --dport 137 –j ACCEPT
#iptables –I INPUT 1 –p udp --dport 138 –j ACCEPT
```

9. Проверьте отображение ресурсов Windows в Linux, для этого введите в адресную строку файлового менеджера адрес:

```
smb://192.168.1.11/
```

10. Проверьте отображение ресурсов Linux из Windows, введя в файловый менеджер адрес:

```
\\192.168.1.30
```

11. Создайте общую папку на машине с ОС Linux.
12. Откройте папку в общий доступ на чтение и на запись как для сервера samba, так и для NFS:

a. В контекстном меню каталога: **Свойство папки** | **Сделать общими** | **Общие файлы.**

b. Установите флажок **Разрешить общий доступ в локальной сети** , **Расширенный режим**, **Использовать SAMBA** и **Использовать NFS**.

c. Щелкните по кнопке **Добавить каталог**, укажите каталог, который необходимо представить в общий доступ.

d. В новом окне установите флажки: **сделать общей через NFS, Открытая**, **Записываемая**.

e. Установите флажки: **Сделать общей через SAMBA**, **Открытая**, **Записываемая**.

13. Проверьте отображения ресурсов Linux с машины с Windows.

Практическая работа 16. Установка и настройка web-сервера (IIS в Windows)

Цель работы: Изучить этапы установки службы Internet Information Service (IIS) в ОС Windows. Получить навыки настройки сервера IIS, создания виртуальных каталогов и организации доступа к серверу с других узлов сети.

Объем времени: 2 ч.

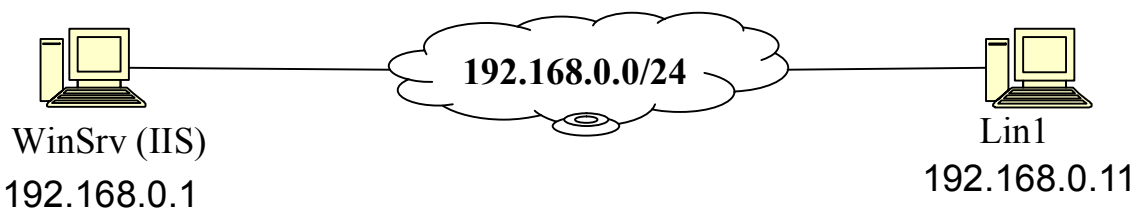
Программное обеспечение: ISO-образ ОС Windows Server 2003, ISO-образ ОС SLAX Linux.

Виртуальные машины:

WinSrv (HDD: 4Gb; RAM: 128Mb; LAN0: internal; ОС Windows Server 2003)

Lin1 (HDD: -; RAM: 128Mb; LAN0: internal; ОС Linux SLAX)

Схема сети:



Последовательность действий:

1. Включите машину WinSrv, настройте сетевой интерфейс. Включите брандмауэр Windows.
2. Включите машину Lin1, настройте сетевой интерфейс.

3. Проверьте прохождение эхо-запросов между узлами.
4. На машине WinSrv установите службу IIS.
    a. Откройте окно **Управления сервером** (**Администрирование** | **Управление сервером**)
    b. Щелкните по кнопке **Добавить или удалить роль**. Щелкните по кнопке **Далее**.
    c. Выберите пункт **Сервер приложений (IIS, ASP.NET)**. Щелкните по кнопке **Далее**.
    d. Щелкните по кнопке **Далее**, для установки сервера.
    e. Щелкните по кнопке **Далее** и дождитесь завершения работы мастера.
    f. Во всех окнах щелкните по кнопкам **Готово**.
5. На машине WinSrv откройте консоль IIS (**Панель управления | Администрирование | Диспетчер служб IIS**).
6. В древовидном списке найдите пункт **Веб-узлы**. В нем выделите **Веб-узел по умолчанию**.
7. Откройте контекстное меню элемента **Веб-узел по умолчанию**. Выберите пункт Свойства.
8. В окне настроек веб-узла перейдите на вкладку **Домашний каталог**.
9. Определите расположения домашнего каталога веб-сервера.
10. Проверьте работоспособность web-сервера, для этого откройте браузер на машине WinSrv, и введите адрес:

    ```
    http://localhost/
    ```

11. В файловом менеджере на машине WinSrv найдите домашний каталог веб-сервера (как правило, c:\Inetput\wwwroot). Этот каталог есть корневая директория веб-узла. Ознакомьтесь с его содержимым.
12. Создайте в домашнем каталоге веб-сервера (c:\Inetput\wwwroot) каталог `megaproba`.
13. Создайте в каталоге `megaproba` файл `test.html`. Со следующим содержимым:

    ```
    <html>
    <head>
    <title>Проба</title>
    </head>
    <body>
    <h1>Привет.</h1>
    ```

```
Текст...
</body>
```

14. В контекстном меню узла выберите пункт **Создать** | **Создать виртуальный каталог…**.
15. В открывшемся окне мастера создания виртуального каталога щелкните по кнопке **Далее**.
16. Укажите псевдоним каталога, например, `proba`. Щелкните по кнопке **Далее**.
17. Укажите путь к реальному каталогу на диске c:\Inetput\wwwroot\megaproba. Щелкните по кнопке **Далее**.
18. Установкой соответствующих флажков разрешите чтение каталога, выполнение ASP и CGI скриптов. Щелкните по кнопке **Далее**.
19. На последней странице мастера щелкните по кнопке **Готово**.
20. Убедитесь в создании нового виртуального каталога.
21. Откройте браузер на машине WinSrv, и введите адрес:

    `http://localhost/proba/test.html`
22. В браузер на машине WinSrv, и введите адрес:

    `http://localhost/megaproba/test.html`
23. Скопируйте файл test.html в корневой каталог веб-сервера (c:\Inetput\wwwroot) и переименуйте его в index.html.
24. В браузере на машине WinSrv, и введите адрес:

    `http://localhost/`
25. На машине WinSrv в диспетчере служб IIS щелкните правой кнопкой мыши по пункту **Веб-узле по умолчанию** и выберите пункт **Свойства**.
26. В открывшемся окне перейдите на вкладку **Документы**.
27. В группе **Задать страницу содержания по умолчанию** щелкните кнопку **Добавить…**

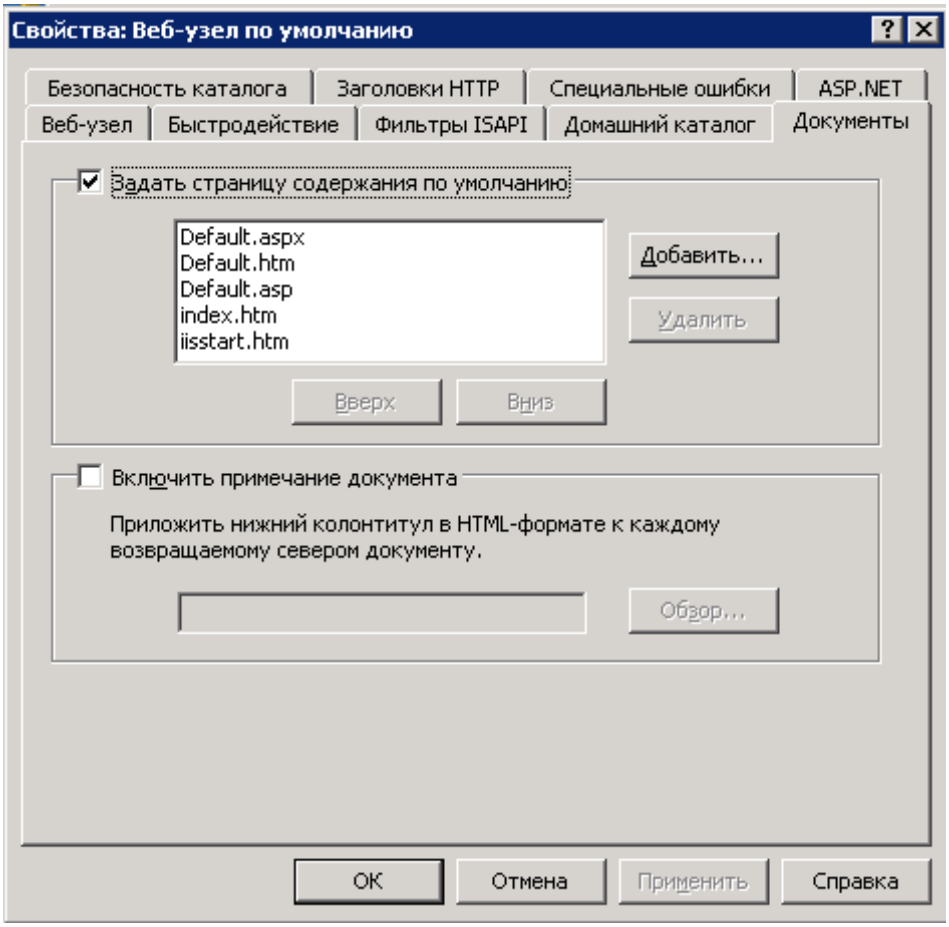


Рис. 22. Окно свойств веб-узла по умолчанию в оснастке IIS

28. В новом окне введите имя страницы index.html и щелкните по кнопке **Ок**.
29. Вернувшись в предыдущее окно кнопкой **Вверх** переместите пункт `index.html` на начало списка.
30. Закройте окно щелчком по кнопке **Ок**.
31. В браузере на машине WinSrv, и введите адрес:

    `http://localhost/`
32. Создайте в корневом каталоге веб-сервера файл index.php со следующим содержимым.

    `<?php phpinfo(); ?>`
33. Обратитесь к файлу через браузер по адресу:

    `http://localhost/index.php`
34. Загрузите на узел WinSrv дистрибутив интерпретатора PHP.
35. Распакуйте его в каталог c:\php.
36. Откройте Диспетчер служб IIS на машине WinSrv.
37. В древовидном списке откройте раздел **Расширения веб-службы**.
38. В меню **Действие** выберите пункт **Добавить новые расширения веб-службы…** .

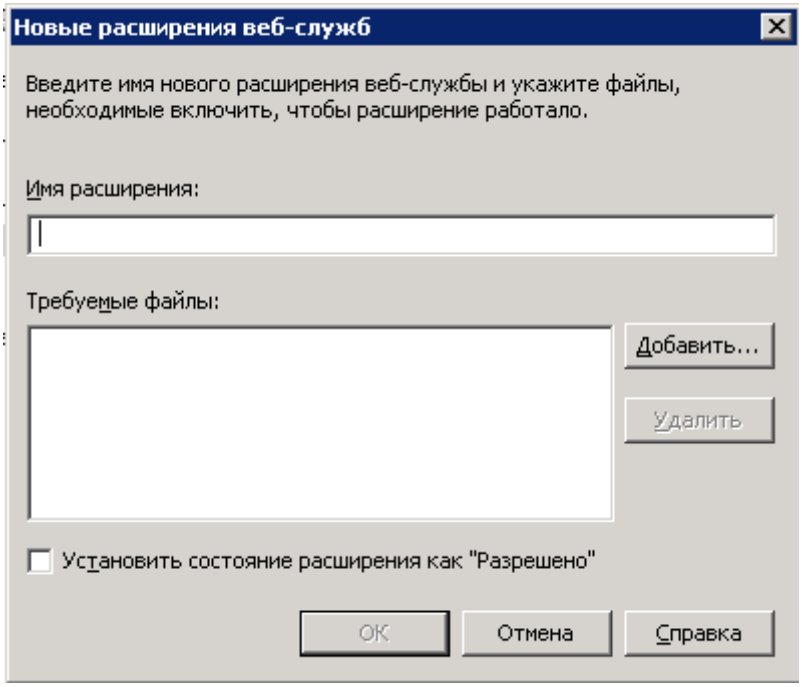


Рис. 23. Окно создания расширений веб-служб

39. Введите имя расширения, например, PHP5 и укажите путь до файла c:\php\php-cgi.exe.
40. Установите флажок **Установите состояние расширения как «Разрешено»**.
41. Щелкните по кнопке **Ок**.
42. Откройте окно свойств Веб-узла по умолчанию через контекстное меню соответствующего пункта в древовидном списке.
43. На вкладке **Домашний каталог** щелкните по кнопке **Настройка…** .
44. В новом окне на вкладке **Сопоставления** щелкните по кнопке **Добавить...** .

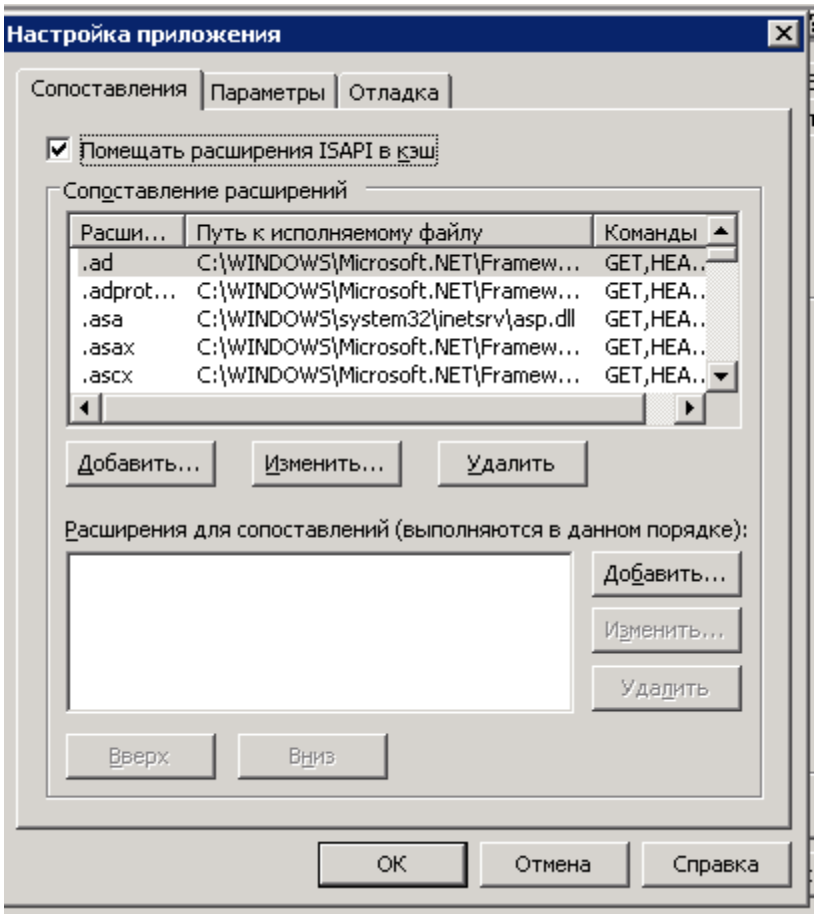


Рис. 24. Окно настройки сопоставления расширениям

45. В новом окне укажите путь до файла c:\php\php-cgi.exe и укажите обрабатываемые им расширения .php.
46. Щелкните по кнопке **Ок** во всех открытых окнах.
47. Обратитесь к файлу index.php через браузер по адресу:

    `http://localhost/index.php`
48. Убедитесь в корректной обработке скрипта.
49. Запустите на машине Lin1 браузер и обратитесь по IP-адресу узла к веб-серверу WinSrv:

    `http://192.168.0.1/`
50. Откройте порт 80 в брандмауэре Windows для прохождения запросов к веб-серверу.
    a. Откройте окно настройки сетевого экрана (**Панель управления | Брандмауэр Windows**).
    b. Перейдите на вкладку **Исключения**.
    c. Щелкните по кнопке **Добавить порт…** .
    d. Введите символическое название, например, web.
    e. Укажите порт 80, установите переключатель TCP.

f. Щелкните по кнопке **Ок** во всех открытых окнах.

51. На машине Lin1 в браузере и обратитесь по IP-адресу узла к веб-серверу WinSrv:

    http://192.168.0.1/

52. Также попробуйте отобразить другие страницы:

    http://192.168.0.1/index.php

    http://192.168.0.1proba/test.html

53. Убедитесь в успешном отображении страниц.

Практическая работа 17. Установка и настройка web-сервера (Apache в Linux)

Цель работы: Изучить этапы установки службы Internet Information Service (IIS) в ОС Windows. Получить навыки настройки сервера IIS, создания виртуальных каталогов и организации доступа к серверу с других узлов сети.

Объем времени: 2 ч.

Программное обеспечение: ISO-образ ОС openSuSE, ISO-образ ОС Windows XP.

Виртуальные машины:

LinSrv (HDD: 8Gb; RAM: 256Mb; LAN0: internal; ОС openSuSE)

Win1 (HDD: 4; RAM: 128Mb; LAN0: internal; ОС Windows XP)

Схема сети:

Последовательность действий:

1. Включите машину LinSrv, настройте сетевой интерфейс.
2. Включите машину Win1, настройте сетевой интерфейс.
3. Проверьте прохождение эхо-запросов между узлами.
4. Установите на машину LinSrv веб-сервер Apache.

```
#apt-get install apache2
```

Или

```
#zypper install apache2
```

5. Запустите сервер Apache командой:

```
#/etc/init.d/apache2 start
```

6. Откройте в браузере на машине LinSrv адрес:

```
http://localhost/
```

7. Проанализируйте файл `/etc/apache2/default-server.conf`, открыв его для чтения:

```
#cat /etc/apache2/default-server.conf
```

8. По параметру DocumentRoot определите корневой каталог веб-сервера.
9. Создайте в корневом каталоге сервера (`/srv/www/htdocs`) текстовый файл `index.htm` с содержимым:

```
<html>
<head>
<title>Проба</title>
</head>
<body>
<h1>Привет.</h1>
Текст...
</body>
```

10. Обратитесь через браузер на машине LinSrv по адресу:

```
http://localhost/
```

11. Откройте на редактирование файл `/etc/apache2/default-server.conf`:

```
#vim /etc/apache2/default-server.conf
```

12. Добавьте в файл настройку об индексных файлах:

```
DirectoryIndex index.htm
```

13. Перезагрузите веб-сервер Apache командой:

```
#/etc/init.d/apache2 restart
```

14. Проверьте отображение корневой страницы по адресу:

```
http://localhost/
```

15. Установите интерпретатор PHP:

```
#apt-get install php5 apache2-mod_php5
```

Или

```
#zypper install php5 apache2-mod_php5
```

16. Перейдите в корневую директорию веб-сервера

```
#cd /srv/www/htdocs
```

17. Откройте редактор vim

```
#vim
```

18. Нажмите на клавиатуре **i** для перехода в режим редактирования. Наберите содержимое файла:

```
<?php phpinfo(); ?>
```

19. Нажмите Esc, чтобы выйти в режим команд. Нажмите на клавиатуре двоеточие (:), наберите команду sav и имя файла index.php. На экране должна образоваться строка:

```
:sav index.php
```

20. Нажмите Enter для сохранения файла, а затем выйдите из редактора, нажав на клавиатуре Z, Z.

21. Дайте команду `dir` и убедитесь в создании файла index.php.

22. Обратитесь к файлу через браузер на машине LinSrv:

http://localhost/index.php

23. Перезагрузите веб-сервер apache командой:

```
#/etc/init.d/apache2 restart
```

24. Обратитесь к файлу через браузер на машине LinSrv:

http://localhost/index.php

25. Обратитесь к веб-серверу по IP-адресу LinSrv с машины Win1.

http://10.0.0.1/

26. Откройте порт в сетевом экране на машине LinSrv:

    a. Вариант 1. **YaST2** | **Безопасность и пользователи** | **Брандмауэр** | **Разрешенные службы** | **HTTP сервер** | **Добавить**. Примените все настройки в окнах.

    b. Вариант 2. Через iptables:

```
#iptables -I INPUT 1 -p tcp --dport 80 -j ACCEPT
```

27. Обратитесь к веб-серверу по IP-адресу LinSrv с машины Win1.

http://10.0.0.1/

http://10.0.0.1/index.php

Практическая работа 18. Настройка DNS сервера (Windows)

Цель работы: Изучить этапы установки DNS-сервера в ОС Windows, изучить разрешение DNS-запросов. Получить навыки настройки DNS-сервера, настройки клиентских узлов для обращения к DNS-серверу для разрешения DNS-имен.

Объем времени: 2 ч.

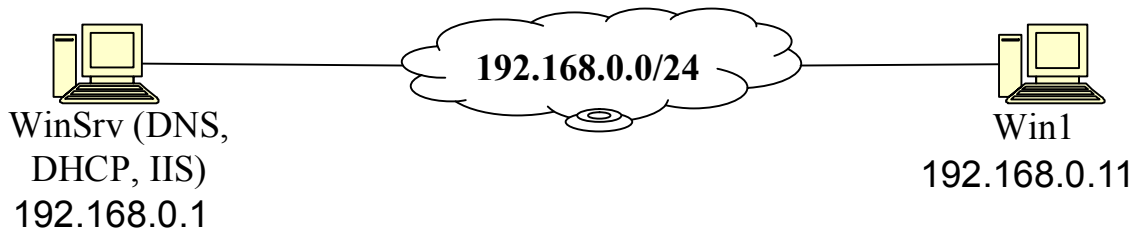
Программное обеспечение: ISO-образ ОС Windows Server 2003, ISO-образ ОС Windows XP.

Виртуальные машины:

WinSrv (HDD: 4Gb; RAM: 128Mb; LAN0: internal; ОС Windows Server 2003)

Win1 (HDD: 4Gb; RAM: 128Mb; LAN0: internal; ОС Windows XP)

Схема сети:



Последовательность действий:

1. Запустите обе виртуальные машины. Настройте сетевые интерфейсы. Проверьте прохождение эхо-запросов.
2. Откройте окно **Управление сервером** (**Панель управления | Администрирование | Управление данным сервером**).
3. Щелкните по кнопке **Добавить или удалить роль**.
4. Выберите **DNS-сервер**.
5. Щелкните по кнопке **Далее**. На следующей странице убедитесь, что мастер будет устанавливать DNS-сервер, а затем запустит мастер настройки. Щелкните по кнопке **Далее**. В ходе установки может потребоваться дистрибутивный диск. При запросе примонтируйте в виртуальной машине ISO образ дистрибутивного диска Windows Server 2003.
6. В открывшемся окне настройки мастера щелкните по кнопке **Далее**, чтобы начать первичную настройку DNS-сервера.
7. На странице Выбор действия установите переключатель **Создать зону прямого просмотра**. Щелкните по кнопке **Далее**.

8. На странице **Размещение сервера** выберите пункт **Управление зоной выполняется этим сервером**. Щелкните по кнопке **Далее**.
9. Введите имя зоны, например, proba.ru. Щелкните по кнопке **Далее**.
10. На следующей странице подтвердите файл зоны и щелкните по кнопке **Далее**.
11. На странице **Динамическое обновление** выберите пункт **Запретить динамическое обновление**. Щелкните по кнопке **Далее**.
12. На странице **Серверы** пересылки выберите пункт **Нет, не пересылать запросы**. Щелкните по кнопке **Далее**.
13. После завершения работы мастера щелкните по всех окнах по кнопкам **Готово**.
14. На странице создания домена выберите пункт **Создать новый домен в новом дереве доменов**. Щелкните по кнопке **Далее**.
15. На странице имени домена укажите имя будущего домена, например, `proba`. Щелкните по кнопке **Далее**.
16. На странице создания леса, выберите пункт **Создать новое дерево доменов в новом лесу**. Щелкните по кнопке **Далее**.
17. На итоговой странице проверьте все указанные настройки, и щелкните по кнопке **Далее**.
18. Дождитесь завершения работы мастера.
19. Откройте оснастку DNS (**Панель управления | Администрирование | DNS**)

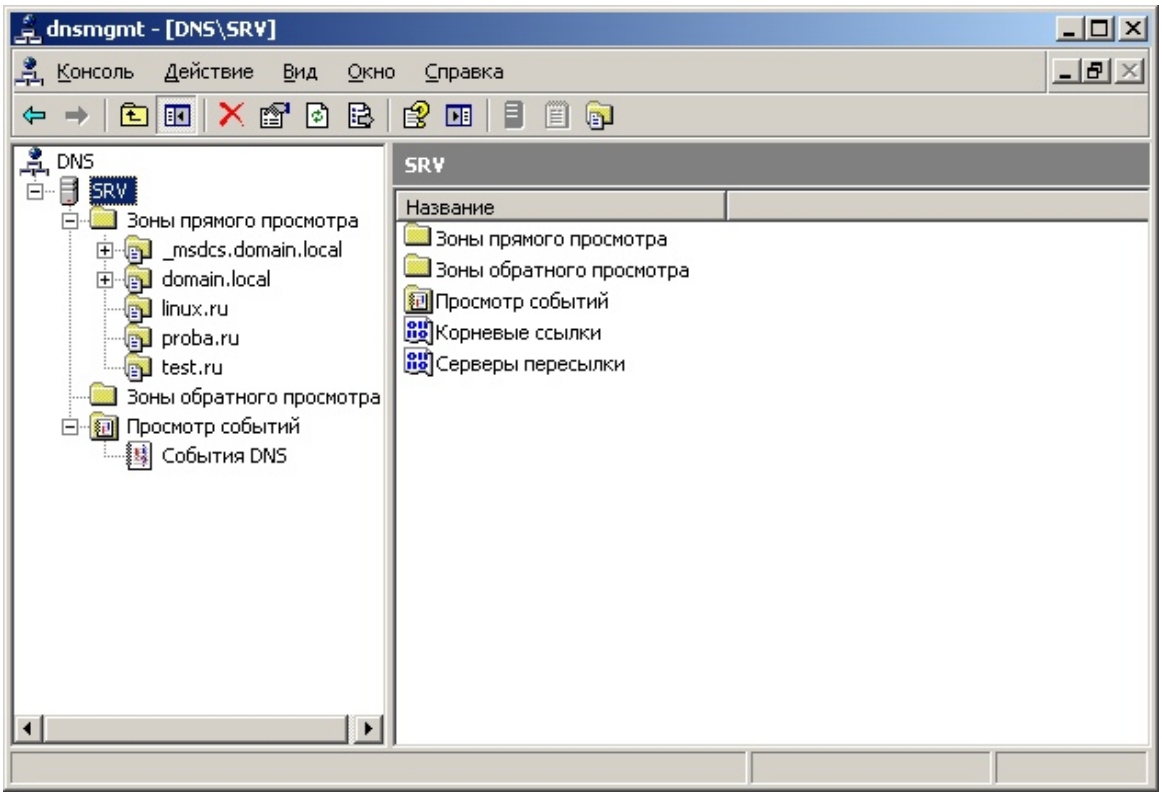


Рис. 25. Окно оснастки DNS-сервера

20. В древовидном списке сервера выберите пункт **Зоны прямого просмотра**.
21. На машине WinSrv в оснастке настройки DNS выберите новую зону `proba.ru` в древовидном списке DNS-сервера.
22. В контекстном меню зоны выберите пункт **Создать узел (A)…** .
23. В открывшемся окне имя узла оставьте пустым, а в качестве IP-адреса укажите адрес WinSrv. Щелкните по кнопке **Добавить узел**,
24. Создайте еще один узел с имя узла `www` и IP-адресом таким же, как WinSrv.
25. В качестве IP-адреса узла укажите IP-адрес машины WinSrv.
26. Щелкните по кнопке **Добавить узел**, затем закройте окно.
27. Убедитесь в появлении новой записи в зоне `proba.ru`.
28. Откройте консоль (**Пуск | Выполнить…**, введите `cmd` и щелкните по кнопке **Ок**).
29. Пошлите эхо-запрос на хост с именем proba.ru.

```
>ping www.proba.ru
```

30. Откройте браузер, перейдите по адресу http://www.proba.ru.
31. Откройте в брандмауэре порт 53/UDP.
    a. Откройте окно настройки сетевого экрана (**Панель управления | Брандмауэр Windows**).
    b. Перейдите на вкладку **Исключения**.
    c. Щелкните по кнопке **Добавить порт…** .
    d. Введите символическое название, например, dns.
    e. Укажите порт 53, установите переключатель UDP.
    f. Щелкните по кнопке **Ок** во всех открытых окнах.
32. На машине Win1 в качестве DNS-сервера укажите IP-адрес WinSrv.
    a. Откройте окно настройки сетевого интерфейса: **Пуск | Панель управления | Сетевые подключения**. Выберите локальный сетевой интерфейс. Правым щелчком мыши по пункту сетевого интерфейса вызовите контекстное меню, выберите пункт **Свойства**.
    b. Выберите в списке протоколов **Протокол Интернета (TCP/IP)** и щелкните по кнопке **Свойства**.
    c. В новом окне установите переключатель **Использовать следующие адреса DNS-серверов**.
    d. В поле **Предпочитаемый DNS-сервер** укажите IP-адрес DNS-сервера WinSrv (192.168.0.1).

33. На машине Win1 откройте браузер, перейдите по адресу http://www.proba.ru.
34. Убедитесь в успешном отображении данных веб-сервера, настроенного в предыдущей работе (Установка и настройка web-сервера (IIS в Windows), см. стр. 194).
35. Просмотреть кэш DNS

```
>ipconfig /displaydns
```

36. Очистите кэш DNS-записей:

```
ipconfig /displaydns
```

37. На машине WinSrv дополните настройки DHCP сервера, чтобы он выдавал адрес DNS сервера клиентам.
    a. Откройте оснастку **DHCP**.
    b. Найдите в древовидном списке пункт с ранее созданной областью и раскройте список
    c. В древовидном списке области перейдите в пункт **Параметры области**.
    d. В контекстном меню выберите пункт **Настроить параметры**.
    e. Установите в списке флажок напротив пункта **006 DNS-серверы**.
    f. Введите в поле IP-адрес адрес данного сервера и щелкните по кнопке **Добавить**.
    g. Щелкните по кнопке **Ок**, чтобы закрыть окно.
38. На машине Win1 в свойствах сетевого соединения выберите пункт переключатель **Получать адреса DNS-серверов автоматически**.
39. Обновите аренду адреса, чтобы Win1 обновил аренду и получил новые настройки вместе с DNS-сервером.

```
>ipconfig /renew
```

40. На машине Win1 откройте браузер, перейдите по адресу http://www.proba.ru.
41. Просмотреть кэш DNS

```
>ipconfig /displaydns
```

42. Убедитесь в успешном отображении кэшированных DNS-записей.

Практическая работа 19. Настройка DNS сервера (Linux)

Цель работы: Изучить этапы установки DNS-сервера bind в ОС Linux, изучить разрешение DNS-запросов. Получить навыки настройки DNS-

сервера, настройки клиентских узлов для обращения к DNS-серверу для разрешения DNS-имен в ОС Linux.

Объем времени: 2 ч.

Программное обеспечение: ISO-образ ОС openSuSE, ISO-образ ОС SLAX Linux.

Виртуальные машины:

LinSrv (HDD: 8Gb; RAM: 256Mb; LAN0: internal; ОС openSuSE)

Lin1 (HDD: -; RAM: 128Mb; LAN0: internal; ОС Linux SLAX)

Схема сети:

Последовательность действий:

1. Установите пакеты bind, bind-*.

   ```
   #apt-get install bind bind-*
   ```

   Или

   ```
   #zypper install bind bind-*
   ```

2. Откройте файл `/etc/named.conf` для чтения

   ```
   #cat /etc/named.conf
   ```

3. Найдите инструкцию `options` и параметр `directory`, в нем указано расположение зонных файлов. Например, `/var/lib/named`.

4. Откройте файл `/etc/named.conf` в редакторе vim

   ```
   #vim /etc/named.conf
   ```

5. Создайте в файле зону:

   ```
   zone "domain.ru" {
     type master;
     file "master/domain.ru"; #относительно directory
   };
   ```

6. В каталоге зонных файлов в указанном месте (`/var/lib/named/master/`) создайте зонный файл `domain.ru` с содержимым:

```
$TTL 2d
@ IN SOA  compname.site.       root.compname.site. (
  2009052400;  serial
  3h; период обновление зоны для вторичных DNS
  1h; период между попытками, если первичный не дал обновление
  1w; период устаревания, если первичный так и не отвечает
  1d ) ; минимальное TTL для отрицательных ответов

domain.ru. 86400   NS  compname.site.;compname.site замените на имя компьютера LinSrv
domain.ru.     86400   A   10.0.0.1
www            86400   A   10.0.0.1
```

7. Запустите демон named командой:

```
#/etc/init.d/named start
```

8. Перезагрузите зону командой:

```
#rndc reload domain.ru
```

9. Проверьте информацию о зоне командой dig:

```
dig @domain.ru version.bind txt chaos
```

10. Откройте порт в брандмауэре linux порт 53/UDP:

```
#iptables –A INPUT 1 –p udp --dport 53 –j ACCEPT
```

11. На машине Lin1 настройте DNS-сервера для разрешения имен. Для этого откройте в редакторе vim файл /etc/resolv.conf.

```
#vim /etc/resolv.conf
```

12. Добавьте в него строчку

```
nameserver 192.168.1.11
```

13. Сохраните файл (Esc, Z, Z)

14. На машине Lin1 пошлите эхо-запрос

```
#ping domain.ru
#ping www.domain.ru
```

15. На машине Lin1 откройте браузер и перейдите по адресу

```
http://www.domain.ru
```

16. Убедитесь в правильном отображении содержимого веб-страницы, ранее настроенного веб-сервера.

Практическая работа 20. Настройка почтового сервера (Windows)

Цель работы: Изучить основы работы почтовых протоколов SMTP и POP3, этапы установки почтовой службы в ОС Windows. Получить навыки настройки почтового сервера и создания учетных записей в ОС Windows.

Объем времени: 1 ч.

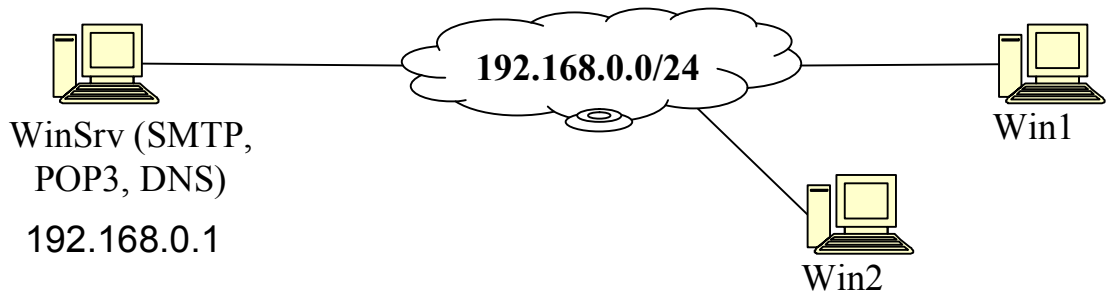
Программное обеспечение: ISO-образ ОС Windows Server 2003, ISO-образ ОС Windows XP.

Виртуальные машины:

WinSrv (HDD: 4Gb; RAM: 128Mb; LAN0: internal; ОС Windows Server 2003)

Win1 (HDD: 4Gb; RAM: 128Mb; LAN0: internal; ОС Windows XP)

Схема сети:



Последовательность действий:

1. Включите машину WinSrv, настройте сетевой интерфейс. Включите брандмауэр Windows.
2. Включите машину Win1, настройте сетевой интерфейс.
3. Проверьте прохождение эхо-запросов между узлами.
4. На машине WinSrv установите службу SMTP.
    a. Откройте окно **Управления сервером** (**Администрирование** | **Управление сервером**)
    b. Щелкните по кнопке **Добавить или удалить роль**. Щелкните по кнопке **Далее**.
    c. Выберите пункт **Почтовый сервер (POP3, SMTP)**. Щелкните по кнопке **Далее**.
    d. Выберите метод проверки подлинности **Локальные учетные записи** и введите доменное имя, для которого сервер будет получать почту, например, proba.ru. Щелкните по кнопке **Далее.**

e. Щелкните по кнопке **Далее** и дождитесь завершения работы мастера.

f. Щелкните по кнопке **Готово**.

5. Откройте оснастку **Служба POP3** (**Панель управления | Администрирование | Служба POP3**).
6. В древовидном списке в левой части окна выберите домен (proba.ru).
7. В контекстном меню домена выберите пункт **Создать | Почтовый ящик…** .
8. В новом окне укажите название почтового ящика, например, info, и укажите пароль. Щелкните по кнопке **Ок**.
9. Аналогично создайте почтовый ящик для пользователя support.
10. Выберите в древовидном списке текущий почтовый сервер (WinSrv), в контекстном меню выберите пункт Свойства.
11. В окне свойств поставьте флажок **Требует безопасная проверка пароля (SPA)**. Щелкните по кнопке **Ок**.
12. На машине Win1 запустите почтовую программу, например, Outlook Express.
13. В меню **Сервис (Tools) | Учетные записи… (Accounts…)** на вкладке **Почта (Mail)** щелкните по кнопке **Добавить… (Add…)**.
14. В открывшемся мастере учетных записей введите символическое имя для почтового ящика, например, Mr.Info. Щелкните по кнопке **Далее**.
15. Укажите почтовый адрес учетной записи (info@proba.ru). Щелкните по кнопке **Далее**.
16. Укажите SMTP и POP3 серверы, в данном случае они совпадают и имеют имена proba.ru. Щелкните по кнопке **Далее**.
17. Укажите имя пользователя и пароль к учетной записи и установите флажок **Использовать безопасную проверку пароля (SPA)** . Щелкните по кнопке **Далее**.
18. На машине Win1 в OutlookExpress создайте письмо для support@proba.ru и отправьте его.
19. На машине WinSrv в оснастке POP3 Service проверьте, что в почтовом ящике есть одно сообщение.
20. На машине Win2 настройте OutlookExpress аналогичным образом для учетной записи support@proba.ru.
21. Получите почту для учетной записи support@proba.ru. Ответьте на это письмо.
22. Получите ответ на Win1 для учетной записи info@proba.ru

Практическая работа 21. Настройка почтового сервера (Linux)

Цель работы: Изучить основы работы почтовых протоколов SMTP и POP3, этапы установки почтовой службы в ОС Linux. Получить навыки настройки почтового сервера и создания учетных записей в ОС Linux.

Объем времени: 1 ч.

Программное обеспечение: ISO-образ ОС openSuSE, ISO-образ ОС SLAX Linux.

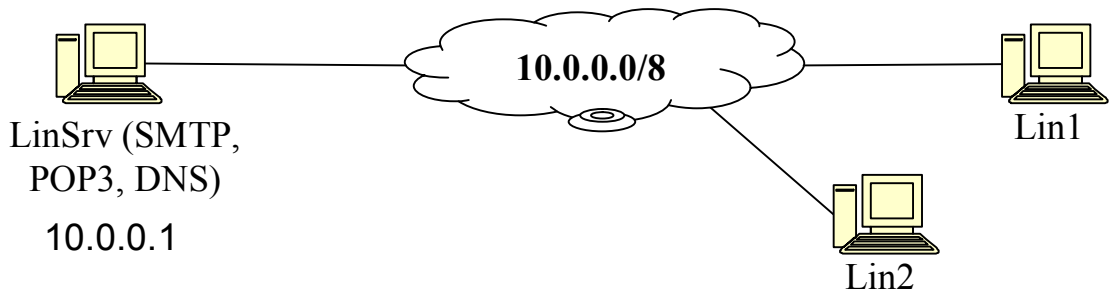
Виртуальные машины:

LinSrv (HDD: 8Gb; RAM: 256Mb; LAN0: internal; ОС openSuSE)

Lin1 (HDD: –; RAM: 128Mb; LAN0: internal; ОС SLAX Linux)

Lin2 (HDD: –; RAM: 128Mb; LAN0: internal; ОС SLAX Linux)

Схема сети:



Последовательность действий:

1. Запустите машины. Проверьте выдачу DHCP сервером IP-адресов или настройте сетевые интерфейсы. Проверьте прохождение эхо-запросов.
2. На машине LinSrv установите пакет fetchmail

   ```
   #apt-get install fetchmail
   ```

   или

   ```
   #zypper install fetchmail
   ```
3. Проверьте установку пакета postfix:

   ```
   #rpm –qa | grep postfix
   ```
4. Запустите fetchmail:

   ```
   #rcfetchmail start
   ```
5. Запустите postfix:

   ```
   #/etc/init.d/postfix start
   ```

6. Убедитесь, что сервер прослушивает 25 порт командой telnet:

```
# telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 domain.ru ESMTP Postfix
```

7. Откройте на редактирование файл `/etc/postfix/main.cf`:

```
#vim /etc/postfix/main.cf
```

8. Проверьте следующие настройки, и правильность указания доменных имен сервера:

```
myhostname = linsrv.domain.ru
mydomain = domain.ru
mydestination = $myhostname, $mydomain, localhost.$mydomain
```

9. Добавьте ip-адреса вашей локальной сети:

```
mynetworks = 192.168.0.0/24, 127.0.0.0/8
```

10. Добавьте строки для закрытия open relay (доставки почты любых клиентов любым адресатам)

```
relay_domains = $mydestination
relay_domains_reject_code = 554
```

11. Закройте файл (Esc, Z, Z).

12. Перезапустите postfix:

```
#/etc/init.d/postfix restart
```

13. В консоли от имени пользователя создайте письмо для другого пользователя в ОС через консольный почтовый клиент `mailx`:

```
#mailx user_login
Subject: введите тему письма
Введите текст письма
.
```

14. Откройте клиент KMail. Настройте учетную запись пользователя для локального ящика и получите почту.

Практическая работа 22. Настройка FTP-сервера (Windows)

Цель работы: Изучить основы работы протокола FTP, этапы установки и первоначальной настройки службы ftp в ОС Windows. Получить базовые навыки настройки и запуска ftp-сервера в ОС Windows.

Объем времени: 1 ч.

Программное обеспечение: ISO-образ ОС Windows Server 2003, ISO-образ ОС Windows XP, ISO-образ ОС SLAX Linux.

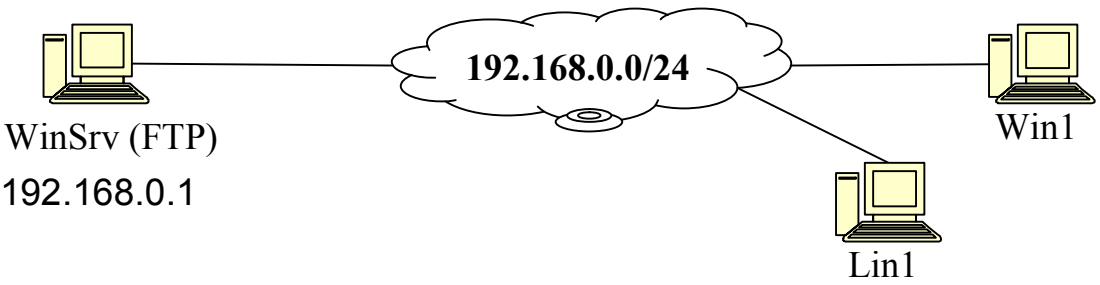
Виртуальные машины:

WinSrv (HDD: 8Gb; RAM: 256Mb; LAN0: internal; ОС openSuSE)

Lin1 (HDD: –; RAM: 128Mb; LAN0: internal; ОС SLAX Linux)

Win1 (HDD: 4Gb; RAM: 128Mb; LAN0: internal; ОС Windows XP)

Схема сети:



Последовательность действий:

1. Запустите машины. Проверьте выдачу DHCP сервером IP-адресов или настройте сетевые интерфейсы. Проверьте прохождение эхо-запросов.
2. Откройте окно установки компонентов Windows (**Панель управления | Установка и удаление программ | Установка компонентов Windows**).
3. Выберите раздел **Сервер приложений**, щелкните по кнопке **Состав…**, выберите пункт **Службы IIS…**, щелкните по кнопке **Состав…** . Поставьте флажок напротив компонента **Служба FTP**.
4. Во всех окна щелкните по кнопке **Ок**.
5. Дождитесь установки компонента.
6. Откройте оснастку **Диспетчер служб IIS** (**Панель управления | Администрирование | Диспетчер служб IIS**)
7. В древовидном списке сервера найдите раздел Узлы FTP. Раскройте список.
8. Выберите пункт **FTP-узел по умолчанию**. В контекстном меню пункта выберите пункт Свойства.

9. На вкладке **Домашний каталог** определите корневой каталог FTP-сервера.
10. В файловом менеджере создайте в домашнем каталоге FTP-сервера текстовый файл с произвольным содержимым.
11. Откройте порты в брандмауэре Windows.
12. Подключитесь к FTP-серверу с машины Lin1 через веб-браузер.

    `ftp://192.168.0.1/`
13. Подключитесь к FTP-серверу с машины Win1 через любой FTP-менеджер.
14. Попробуйте загружать и отправлять файлы на FTP-сервер.

Практическая работа 23. Настройка FTP-сервера (Linux)

Цель работы: Изучить основы работы протокола FTP, этапы установки и первоначальной настройки службы ftp в ОС Linux. Получить базовые навыки настройки и запуска ftp-сервера в ОС Linux.

Объем времени: 1 ч.

Программное обеспечение: ISO-образ ОС openSuSE, ISO-образ ОС SLAX Linux, ISO-образ ОС Windows XP.

Виртуальные машины:

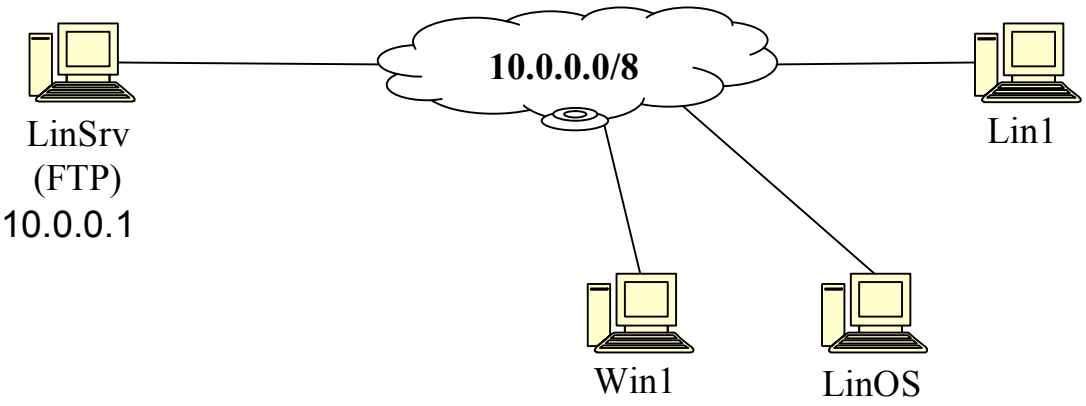
LinSrv (HDD: 8Gb; RAM: 256Mb; LAN0: internal; ОС openSuSE)

LinOS (HDD: 8Gb; RAM: 256Mb; LAN0: internal; ОС openSuSE)

Lin1 (HDD: –; RAM: 128Mb; LAN0: internal; ОС SLAX Linux)

Win1 (HDD: 4Gb; RAM: 128Mb; LAN0: internal; ОС Windows XP)

Схема сети:



Последовательность действий:

15. Запустите машины. Проверьте выдачу DHCP сервером IP-адресов или настройте сетевые интерфейсы. Проверьте прохождение эхо-запросов.
16. На машине LinSrv установите пакет `vsftpd`:

```
#apt-get install vsftpd
```

или

```
#zypper install vsftpd
```

17. Откройте на редактирование файл /etc/vsftpd.conf:

```
#vim /etc/vsftpd.conf
```

18. Отредактируйте параметр `write_enabled`, чтобы клиенты имели возможность отправлять файлы на сервер:

```
write_enabled = YES
```

19. Сохраните файл (Esc, Z, Z)
20. На машине LinSrv запустите службу FTP.

```
#/etc/init.d/vsftpd start
```

21. Откройте в сетевом экране необходимые порты (20, 21 и для пассивного режима: 30000-30100)

```
#iptables –I INPUT 1 –p tcp --dport 20:21 –j ACCEPT
#iptables –I INPUT 1 –p tcp --dport 30000:30100 –j ACCEPT
```

22. На машине LinSrv создайте в домашнем каталоге пользователя текстовый файл test.txt.
23. На машине Lin1 в браузере введите адрес:

ftp://*имя_пользователя*@10.0.0.1

24. Просмотрите содержимое каталогов.
25. На машине LinOS в консоли запустите консольный ftp клиент:

```
#ftp
```

26. В приглашении наберите команду подключения к ftp-серверу:

```
ftp>open 10.0.0.1
```

27. Наберите логин пользователя, имеющегося на машине LinSrv. Нажмите на клавиатуре Enter.
28. Введите пароль учетной записи и нажмите Enter.
29. Введите команду `dir`, чтобы просмотреть содержимое текущего каталога.
30. Убедитесь в том, что в списке присутствует тестовый файл test.txt.

31. В приглашении ftp-клиента дайте команду на загрузку файла:

```
ftp>get test.txt
```

32. Проверьте на машине LinOS, что файл загрузился в текущий каталог (тот каталог, который был открыт в консоли до того как был запущен ftp клиент)

33. Выйдите из ftp-клиента, введя команду `exit`:

```
ftp>exit
```

34. Создайте в текущем каталоге на машине LinOS файл proba.txt с произвольным содержимым.

```
#vim proba.txt
```

35. Сохраните файл (Esc, Z, Z)

36. Откройте консольный ftp-клиент. Зайдите на ftp-сервер LinSrv.

37. Передайте файл proba.txt на ftp-сервер командой:

```
ftp>send proba.txt
```

38. Выполните команду `dir`, чтобы просмотреть содержимое каталога ftp-сервера. Убедитесь, что в нем появился файл proba.txt.

39. На машине Win1 подключитесь к ftp-серверу через веб-браузер аналогично пункту 9.

Практическая работа 24. Настройка NAT (Windows)

Цель работы: Изучить принципы работы сетевой трансляции адресов в ОС Windows. Получить навыки настройки сетевой трансляции адресов NAT для обеспечения доступа клиентов к внешним ресурсам, настройки клиентских узлов для обращения к внешним ресурсам.

Объем времени: 2 ч.

Программное обеспечение: ISO-образ ОС Windows Server 2003, ISO-образ ОС Windows XP, ISO-образ ОС openSuSE.

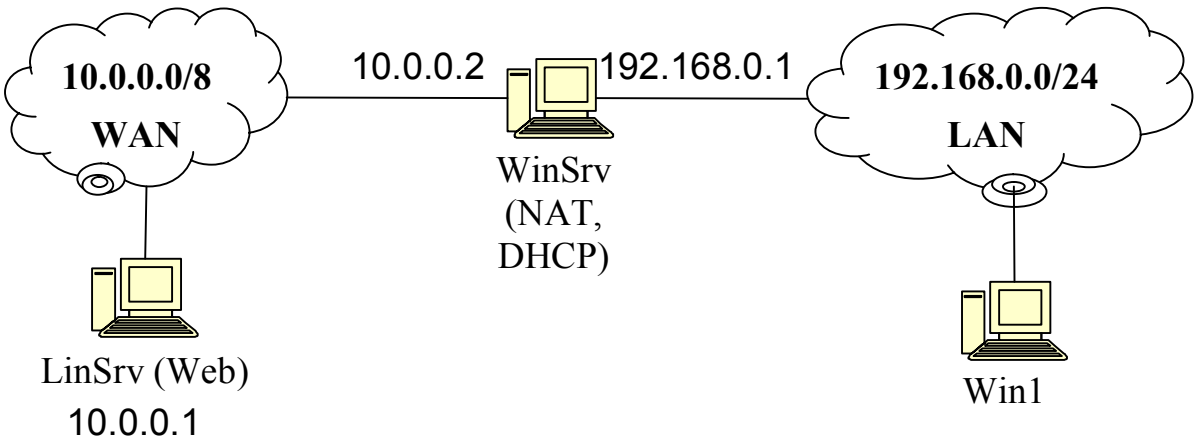
Виртуальные машины:

WinSrv (HDD: 4Gb; RAM: 128Mb; LAN0, LAN1: internal; ОС Windows Server 2003)

Win1 (HDD: 4Gb; RAM: 128Mb; LAN0: internal; ОС Windows XP)

LinSrv (HDD: 8Gb; RAM: 256Mb; LAN0: internal; ОС openSuSE)

Схема сети:



Последовательность действий:

1. Включите WinSrv и Win1.
2. Проверьте аренду IP-адреса узлом Win1, или настройте сетевой интерфейс узла статически.
3. Проверьте прохождение эхо-запросов.
4. На машине WinSrv в настройках DHCP-сервера включите информацию о шлюзе по умолчанию, так как далее узлу Win1 необходимо обращаться в другие сети.
    a. Откройте оснастку **DHCP**
    b. Выберите в древовидном списке сервер и область адресов.
    c. Перейдите в пункт **Параметры области**.
    d. В контекстном меню выберите **Настроить параметры…** .
    e. Поставьте флажок напротив параметра **003 Маршрутизатор**.
    f. Укажите в поле IP-адрес адрес шлюза по умолчанию (192.168.0.1), т.е. текущий адрес сервера для клиентов сети 192.168.0.0/24.
    g. Щелкните по кнопке **Добавить**.
    h. Щелкните по кнопке **Ок**.
5. На машине Win1 обновите аренду

    ```
    >ipconfig /renew
    ```

6. В выводе утилиты `ipconfig` убедитесь в присутствие параметра «шлюз по умолчанию».

    ```
    >ipconfig /all
    ```

7. Настройте на машине WinSrv второй (новый) сетевой интерфейс. Укажите для него IP-адрес 10.0.0.2 и маску подсети 255.0.0.0.
8. Включите машину LinSrv.

9. Проверьте прохождение эхо-запросов между LinSrv и WinSrv.
10. На машине WinSrv отключите службы Брандмауэр Windows, так как служба RRAS не работает совместно со службой ICS.
    a. Откройте окно служб **Панель управления | Администрирование | Службы** | **Брандмауэр Windows**
    b. Выберите тип состояния **Отключено**
    c. Щелкните по кнопке **Стоп**, и щелкните по кнопке **Ок**
11. На машине WinSrv включите службу Маршрутизации и удаленного доступа (Routing and Remote Access Service) (**Панель управления | Администрирование | Службы**).
    a. Откройте оснастку службы RRAS (**Панель управления | Администрирование | Маршрутизация и удаленный доступ**)
    b. В списке в левой части окна выберите данный сервер. В контекстном меню пункта выберите команду **Настроить и включить маршрутизацию и удаленный доступ**.
    c. В появившемся мастере щелкните **Далее**, для начала настройки.
    d. Выберите тип конфигурации **Сетевая трансляция адресов (NAT)** и щелкните по кнопке **Далее**.
    e. Выберите общедоступный интерфейс, то есть тот интерфейс, который подключен к сети 10.0.0.0/8. Щелкните по кнопке **Далее**.
    f. Щелкните по кнопке **Готово** для завершения работы мастера.

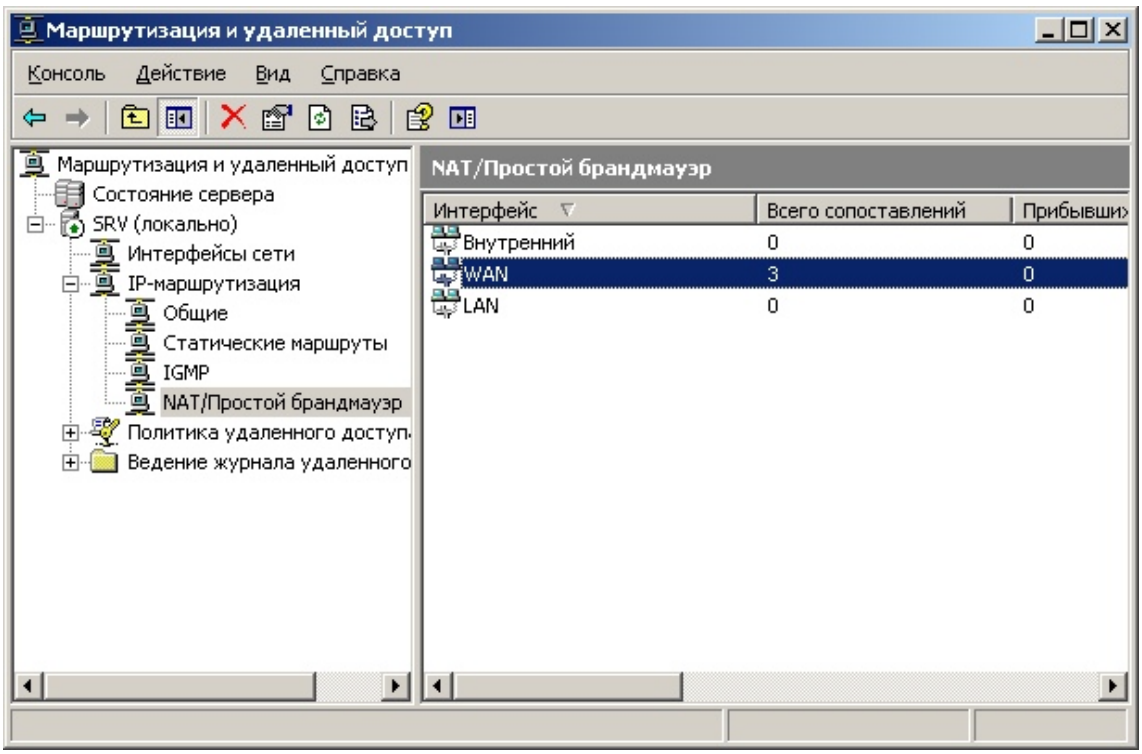


Рис. 26. Окно оснастки службы маршрутизации и удаленного доступа

12. По желанию на машине WinSrv в DNS создайте зону linux.ru и A-запись `www` со ссылкой на 10.0.0.1.
13. Обратитесь на машине WinSrv в браузере по адресу `http://10.0.0.1/index.php` или `http://www.linux.ru/index.php` (если DNS-зона была настроена в п.9).
14. Найдите в списке переменных значение переменной `REMOTE_ADDR`. Проверьте, что ее значение равно 10.0.0.1, т.е. IP-адрес WinSrv.
15. Обратитесь на машине Win1 в браузере по адресу `http://10.0.0.1/index.php` или `http://www.linux.ru/index.php` (если DNS-зона была настроена в п.9).
16. Найдите в списке переменных значение переменной `REMOTE_ADDR`. Проверьте, что ее значение равно 10.0.0.1, т.е. IP-адрес WinSrv. Это означает, что веб-серверу действительно пришел пакет с обратным адресом WinSrv, а не Win1.
17. Отключите службу RRAS и включите ICS, чтобы настроить NAT через службу ICS.
    a. Откройте окно службы **Панель управления | Администрирование | Маршрутизация и удаленный доступ**.
    b. В контекстном меню сервера выберите пункт **Отключить маршрутизацию и удаленный доступ**. Согласитесь с тем, что конфигурация будет утеряна.
    c. Откройте окно службы **Панель управления | Администрирование | Службы | Маршрутизация и удаленный доступ**.
    d. Выберите тип состояния **Отключено.**
    e. Щелкните по кнопке **Стоп**, и щелкните по кнопке **Ок.**
    f. Откройте окно службы **Панель управления | Администрирование | Службы** | **Брандмауэр Windows.**
    g. Выберите тип состояния **Авто**
    h. Щелкните по кнопке **Старт**, и щелкните по кнопке **Ок.**
18. Откройте окно **Сетевые подключения** (**Панель управления | Сетевые подключения**).
19. Откройте контекстное меню внешнего сетевого интерфейса, того, который подключен к сети 10.0.0.0/8.
20. Перейдите на вкладку Дополнительно и поставьте флажок в пункте **Разрешить другим пользователям сети использовать подключение к Интернет данного компьютера**.
21. В появившемся диалоговом окне согласитесь с тем, что в данном случае локальному интерфейсу будет присвоен адрес 192.168.0.1.

22. Щелкните по кнопке **Ок**.

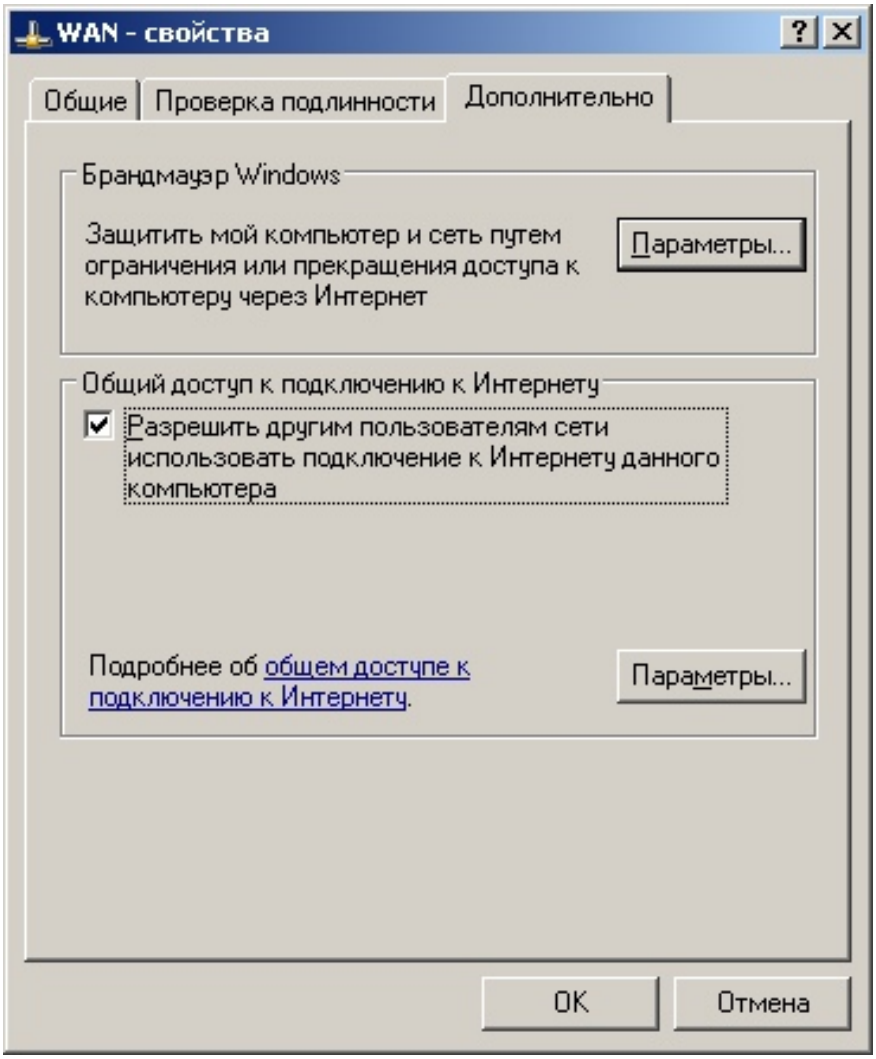


Рис. 27. Окно настройки общего доступа к интерфейсу

23. Обратитесь на машине Win1 в браузере по адресу `http://10.0.0.1/index.php` или `http://www.linux.ru/index.php` (если DNS-зона была настроена в п.9).
24. Найдите в списке переменных значение переменной `REMOTE_ADDR`. Проверьте, что ее значение равно 10.0.0.1, т.е. IP-адрес WinSrv.

Практическая работа 25. Настройка NAT (Linux)

Цель работы: Изучить принципы работы сетевой трансляции адресов в ОС Linux. Получить навыки настройки сетевой трансляции адресов NAT для обеспечения доступа клиентов к внешним ресурсам, настройки клиентских узлов для обращения к внешним ресурсам.

Объем времени: 1 ч.

Программное обеспечение: ISO-образ ОС Windows Server 2003, ISO-образ ОС SLAX Linux LiveCD, ISO-образ ОС openSuSE.

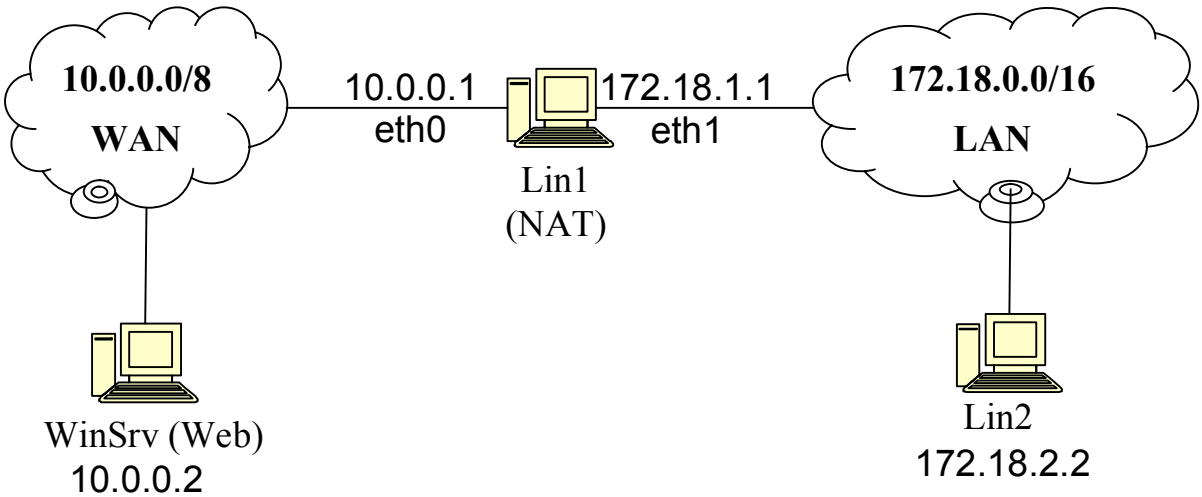
Виртуальные машины:

WinSrv (HDD: 4Gb; RAM: 128Mb; LAN0, LAN1: internal; ОС Windows Server 2003)

Lin1 (HDD: –; RAM: 128Mb; LAN0, LAN1: internal; ОС SLAX Linux)

Lin2 (HDD: –; RAM: 128Mb; LAN0: internal; ОС SLAX Linux)

Схема сети:



Последовательность действий:

1. Включите WinSrv, настройте внешний интерфейс.
2. Включите Lin1 настройте интерфейс eth0. Проверьте прохождение эхо-запросов между обоими узлами.
3. Настройте на Lin1 интерфейс eth1.
4. Настройте на Lin2 сетевой интерфейс. Проверьте прохождение эхо-запросов.
5. Настройте маршрутизацию пакетов на Lin1. Проверьте прохождение эхо-запросов между WinSrv и Lin2.
6. На машине WinSrv создайте в корневом каталоге веб-сервера страницу Default.asp, со следующим содержимым для отображения IP-адреса клиента:

```
Your host is:
<%= Trim(Request.ServerVariables("REMOTE_HOST")) %>
```

7. Включите расширения Acrive Server Pages на веб-сервере.
8. Проверьте отображение IP адреса клиента при обращении с машины Lin2. Для этого в браузере обратитесь по адресу:

http://10.0.0.2/Default.asp

9. Настройте перенаправление tcp-трафика с внутреннего интерфейса eth1 на внешний eth0 с использованием диапазона адресов (в данном случае только один адрес 10.0.0.1):

```
iptables -t nat -A POSTROUTING -p tcp -o eth0 -j SNAT
--to 10.0.0.1
```

10. Обновите страницу браузера у клиента Lin2. Убедитесь, что значение внешнего IP адреса 10.0.0.1, т.е. LinSrv.

Практическая работа 26. Перенаправление портов и публикация локальных служб (Windows)

Цель работы: Изучить принципы работы сетевой трансляции адресов в ОС Windows. Получить навыки публикации внутренних серверов для доступа внешних клиентов к внутренним ресурсам сети.

Объем времени: 1-2 ч.

Программное обеспечение: ISO-образ ОС Windows Server 2003, ISO-образ ОС Windows XP, ISO-образ ОС SLAX Linux.

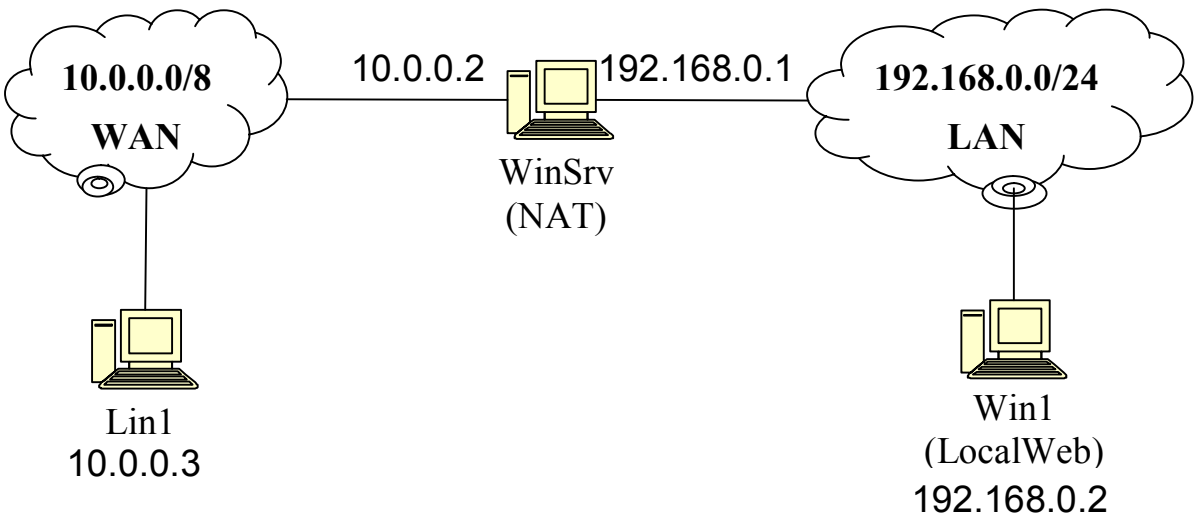
Виртуальные машины:

WinSrv (HDD: 4Gb; RAM: 128Mb; LAN0, LAN1: internal; ОС Windows Server 2003)

Win1 (HDD: 4Gb; RAM: 128Mb; LAN0: internal; ОС Windows XP)

Lin1 (HDD: –; RAM: 128Mb; LAN0: internal; ОС SLAX Linux)

Схема сети:



Последовательность действий:

1. Запустите машину Win1, WinSrv, Lin1, настройте сетевые интерфейсы. Настройте маршрутизацию на WinSrv. Настройте NAT через ICS (аналогично финальному состоянию в работе Настройка NAT (Windows), стр. 216)
2. Установите на машину Win1 службу IIS
    a. Откройте окно установки компонентов Windows (**Панель управления | Установка и удаление программ | Установка компонентов Windows**)
    b. Установите флажок напротив пункта **Internet Information Servicies (IIS)** и щелкните по кнопке **Далее**.
    c. Дождитесь завершения установки (во время установки потребуется диск с дистрибутивом Windows, примонтируйте при необходимости ISO образ дистрибутива).
    d. Создайте в корневом каталоге веб-сервера текстовый файл с именем index.html со следующим содержимым:

    ```
    <h1>Local server</h1>
    ```

    e. В файловом менеджере Проводник откройте окно **Сервис | Свойства папки…** .
    f. На вкладке **Вид** снимите флажок **Использовать простой общий доступ к файлам**.
    g. Откройте контекстное меню папки C:\Inetput\wwwroot и выберите пункт **Свойства**.
    h. В новом окне на вкладке **Безопасность** в списке **Пользователи и группы** выберите группу **Гостевая учетная запись Интернет (*Имяхост*\IUSR_*имяхоста*)**.
    i. Дайте права на чтение каталога и выполнение файлов для этой учетной записи выбором в списке права: **Чтение и выполнение**, **Список содержимого папки** и **Чтение**.

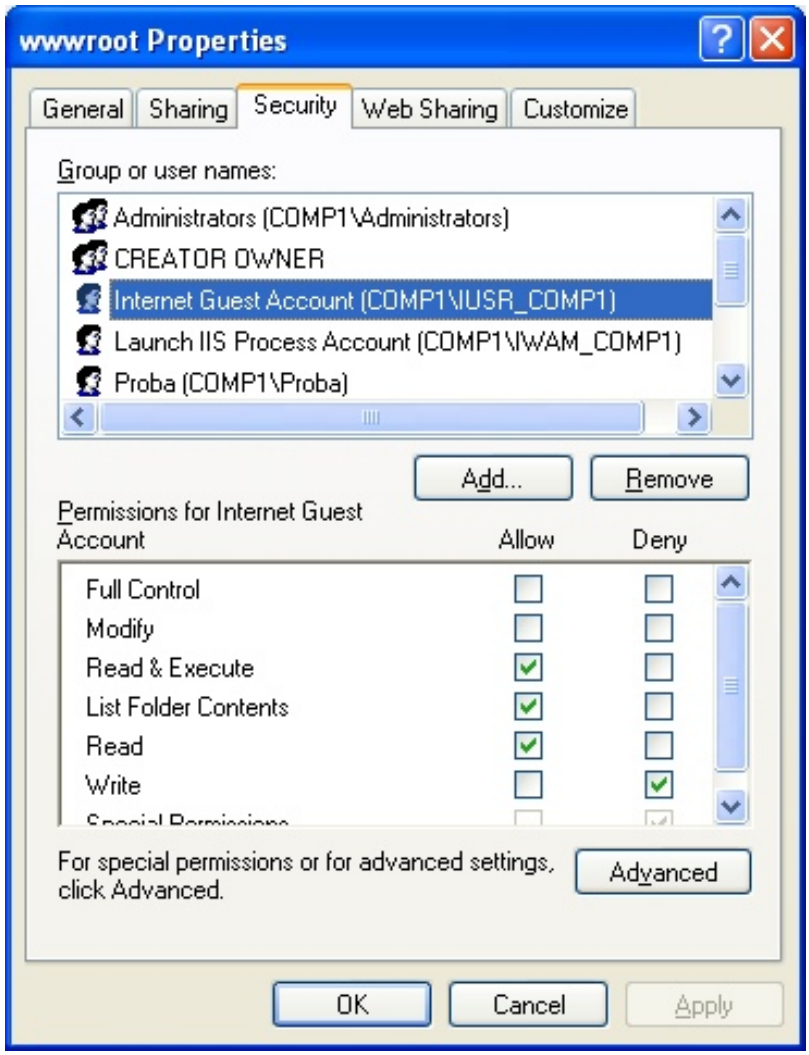


Рис. 28. Окно настройки прав доступа к каталогу

j. Щелкните по кнопке **Ок**.

k. В браузере проверьте успешное отображение содержимого страницы по адресу http://localhost/index.html.

3. На машине WinSrv откройте окно брандмауэра Windows (**Панель управления | Брандмауэр Windows**).
4. На вкладке **Дополнительно** выберите внешний интерфейс и щелкните по кнопке **Параметры**.
5. Установите флажок напротив службы **Web**.
6. Щелкните по кнопке **Изменить…**.

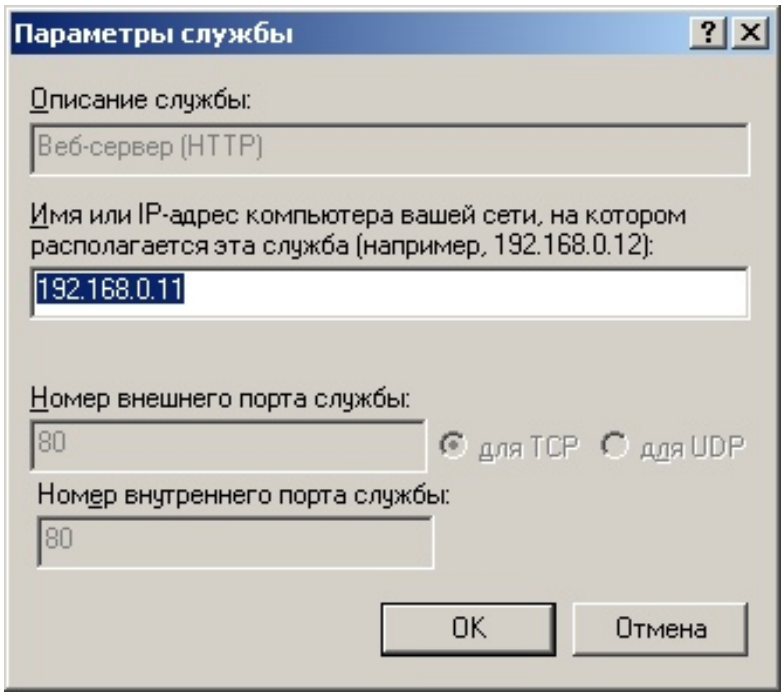


Рис. 29. Окно настройки перенаправления трафика

7. Укажите IP-адрес машины Win1, чтобы изменить адрес службы в локальной сети.
8. Щелкните во всех окнах **Ок**.
9. На машине Lin1 откройте в браузере адрес http://10.0.0.2/index.html. Убедитесь в отображении страницы локального веб-сервера Win1.
10. Отключите службы **Брандмауэр Windows (ICS)** и включите службы **RRAS**, чтобы реализовать публикацию веб-сервера через эту службу. Настройте сетевую трансляцию адресов.
11. В оснастке **Маршрутизация и удаленный доступ** в разделе **IP-маршрутизация** выберите пункт **NAT/Простой брандмауэр**.
12. В правой части окна выберите внешний сетевой интерфейс, в контекстном меню интерфейса выберите пункт **Свойства**.
13. В открывшемся окне на вкладке службы и порты поставьте флажок напротив пункта **Веб-сервер (HTTP)**.
14. В новом окне в поле IP-адреса укажите адрес машины Win1.
15. Во всех окнах щелкните по кнопке **Ок**.
16. На машине Lin1 откройте в браузере адрес http://10.0.0.2/index.html. Убедитесь в отображении страницы локального веб-сервера Win1.

Практическая работа 27. Перенаправление портов и публикация локальных служб (Linux)

Цель работы: Изучить принципы работы сетевой трансляции адресов в ОС Linux. Получить навыки публикации внутренних серверов для доступа внешних клиентов к внутренним ресурсам сети.

Объем времени: 1 ч.

Программное обеспечение: ISO-образ ОС SLAX Linux LiveCD, ISO-образ ОС openSuSE.

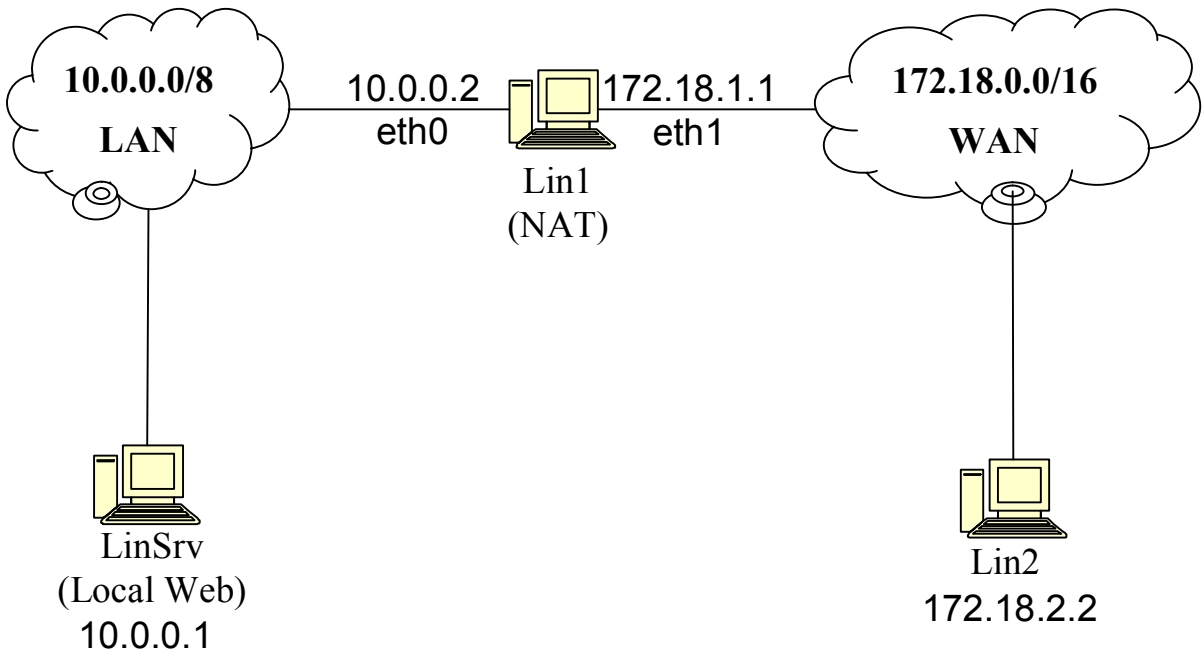
Виртуальные машины:

LinSrv (HDD: 4Gb; RAM: 128Mb; LAN0, LAN1: internal; ОС openSUSE)

Lin1 (HDD: –; RAM: 128Mb; LAN0, LAN1: internal; ОС SLAX Linux)

Lin2 (HDD: –; RAM: 128Mb; LAN0: internal; ОС SLAX Linux)

Схема сети:



Последовательность действий:

1. Включите все машины Lin1 и LinSrv, настройте сетевые интерфейсы.
    a. Проверьте прохождение эхо-запросов между Lin1 и LinSrv.
    b. Настройте шлюз по умолчанию только для LinSrv

       ```
       #route add default gw 10.0.0.2
       ```

    c. Включите машину Lin2, настройте сетевой интерфейс.

d. Включите маршрутизацию на Lin1.

e. Проверьте, что эхо-запросы проходят между Lin1 и Lin2.

f. Проверьте, что эхо-запросы НЕ проходят между узлами LinSrv и Lin2.

g. Проверьте с машины Lin1 функционирование веб-сервера LinSrv.

2. На машине Lin2 обратитесь через браузер по внешнему адресу маршрутизатора http://172.18.1.1/index.php. Убедитесь, что служба веб не отвечает.

3. Установите правило для DNAT и проброса порта с внешнего адреса (172.18.1.1) по протоколу tcp с порта 80, на внутренний адрес веб-сервера (10.0.0.1).

```
#iptables -t nat -A PREROUTING -p tcp -d 172.18.1.1 --dport 80 -j DNAT --to 10.0.0.1
```

4. Проверьте функционирование веб-сервера при обращении из внешней сети по внешнему адресу маршрутизатора. Для этого на машине Lin2 в браузере откройте адрес http://172.18.1.1/index.php.

5. Найдите в списке параметров переменную `REMOTE_ADDR`, убедитесь, что ее значение равно текущему IP-адресу Lin2.

Практическая работа 28. Проброс TCP-порта через SSH

Цель работы: Изучить принципы организации протоколов TCP/IP, основы работы службы SSH и организацию передачи TCP-трафика через протокол SSH. Получить навыки настройки клиента и службы SSH таким образом, чтобы на удаленном клиенте получить доступ к приватной службе в закрытой сети только через опубликованную службу SSH.

Объем времени: 2 ч.

Программное обеспечение: ISO-образ ОС SLAX Linux LiveCD, ISO-образ ОС openSuSE.

Виртуальные машины:

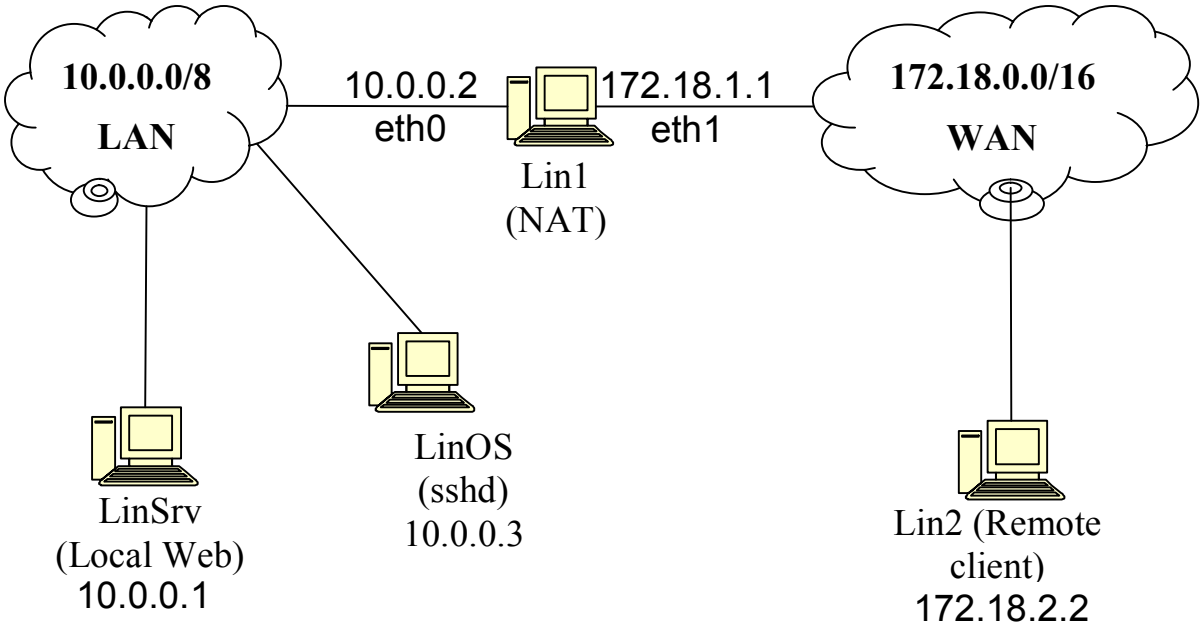
LinSrv (HDD: 4Gb; RAM: 128Mb; LAN0, LAN1: internal; ОС Windows Server 2003)

LinOS (HDD: 4Gb; RAM: 128Mb; LAN0, LAN1: internal; ОС Windows Server 2003)

Lin1 (HDD: –; RAM: 128Mb; LAN0, LAN1: internal; ОС SLAX Linux)

Lin2 (HDD: –; RAM: 128Mb; LAN0: internal; ОС SLAX Linux)

Схема сети:



Последовательность действий:

1. Включите все машины, настройте сетевые интерфейсы как в предыдущей работе.
    a. Проверьте прохождение эхо-запросов между Lin1 и LinSrv.
    b. Настройте шлюз по умолчанию для LinSrv и LinOS.

    ```
    #route add default gw 10.0.0.2
    ```

    a. Проверьте, что эхо-запросы НЕ проходят между узлами LinSrv и Lin2.
2. На машине LinOS установите и запустите службу sshd. Проверьте ее функционирование.
    a. На машине LinOS, откройте порт в файерволе.
    b. Проверьте успешность подключения к машине LinOS по протоколу ssh с LinSrv.
3. На машине Lin1 опубликуйте только одну внутреннюю службу — ssh, находящуюся на машине LinOS:

    ```
    #iptables -t nat -A PREROUTING -p tcp -d 172.18.1.1 --dport 22 -j DNAT --to 10.0.0.3
    ```
4. Проверьте успешность подключения к опубликованной службе sshd с машины Lin2.
5. На машине LinOS откройте файл конфигурации /etc/ssh/sshd_config в редакторе vim:

    ```
    #vim /etc/ssh/sshd_config
    ```

6. Включите перенаправление TCP:

   ```
   AllowTcpForwarding yes
   ```

7. Сохраните изменения в файле. Перезагрузите службу sshd на машине LinOS.
8. На машине Lin2 подключитесь к Lin1, перенаправив трафик с 80 порта LinSrv частной сети на свой произвольный локальный порт:

   ```
   #ssh –f –N –L 4080:10.0.0.1:80 root@172.18.1.1
   ```

9. Откройте на машине Lin2 браузер. Наберите в строке адреса URL:

   ```
   http://localhost:4080
   ```

10. Проверьте успешное отображение контента, находящегося на веб-сервере в закрытой сети LAN.

Практическая работа 29. Настройка контроллера домена и Active Directory

Цель работы: Изучить принципы организации компьютеров в сети с Active Directory. Получить навыки установки и первичной настройки Active Directory на контроллер домена. Получить базовые навыки организации безопасности в сети.

Объем времени: 2 ч.

Программное обеспечение: ISO-образ ОС Windows Server 2003, ISO-образ ОС Windows XP, ISO-образ ОС SLAX Linux.

Виртуальные машины:

WinSrv (HDD: 4Gb; RAM: 128Mb; LAN0: internal; ОС Windows Server 2003)

Win1 (HDD: 4Gb; RAM: 128Mb; LAN0: internal; ОС Windows XP)

Win1 (HDD: 4Gb; RAM: 128Mb; LAN0: internal; ОС Windows XP)

Схема сети:

Последовательность действий:

1. Включите машины. Проверьте выдачу IP-адресов DHCP сервером на WinSrv или настройте сетевые интерфейсы. Проверьте прохождение эхо-запросов.
2. Запустите установку Active Directory на машине WinSrv.
    a. Вариант 1. **Панель управления | Администрирование | Управление данным сервером | Добавить или удалить роль**. В открывшемся окне выберите пункт **Контроллер домена (Active Directory)**, щелкните по кнопке **Далее**. Для запуска мастера на следующей странице щелкните по кнопке **Далее**.
    b. Вариант 2. В командной строке дайте команду dcpromo.

    ```
    >dcpromo
    ```
3. В окне мастера щелкните по кнопке **Далее** для начала установки Active Directory. Ознакомившись с информацией на следующей странице также щелкните по кнопке **Далее**.
4. Выберите пункт **Контроллер домена в новом домене**. Щелкните по кнопке **Далее**.
5. Выберите пункт **Новый домен в новом лесу**. Щелкните по кнопке **Далее**.
6. Введите имя домена, например, domain.local. Щелкните по кнопке **Далее**.
7. Подтвердите NetBIOS имя домена, или введите иное. Щелкните по кнопке **Далее**.
8. Укажите пути к базам данных и журналу AD. Щелкните по кнопке **Далее**.
9. Укажите путь к системному тому. Щелкните по кнопке **Далее**.
10. На странице диагностики DNS выберите пункт **Установить и настроить DNS-сервер на этот компьютер и выбрать этот DNS-сервер в качестве предпочитаемого DNS-сервера**. Щелкните по кнопке **Далее**.
11. На следующей странице укажите совместимость, указав параметр **Разрешения, совместимые только с Windows 2000 или Windows Server 2003**. Щелкните по кнопке **Далее**.
12. Задайте пароль для восстановления Active Directory. Щелкните по кнопке **Далее**.
13. На следующей странице подтвердите готовность к установки Active Directiory на сервер. Щелкните по кнопке **Далее**.
14. Дождитесь завершения работы мастера. Щелкните по кнопке **Готово**.
15. Перезагрузите сервер.
16. При входе под учетной записью обратите внимание на изменения окна входа. Теперь там доступен выбор домена для входа.

17. Откройте оснастку **Active Directory – пользователи и компьютеры** (**Панель управления | Администрирование**).
18. В древовидном списке выберите текущий сервер, в нем раздел Пользователи (Users).

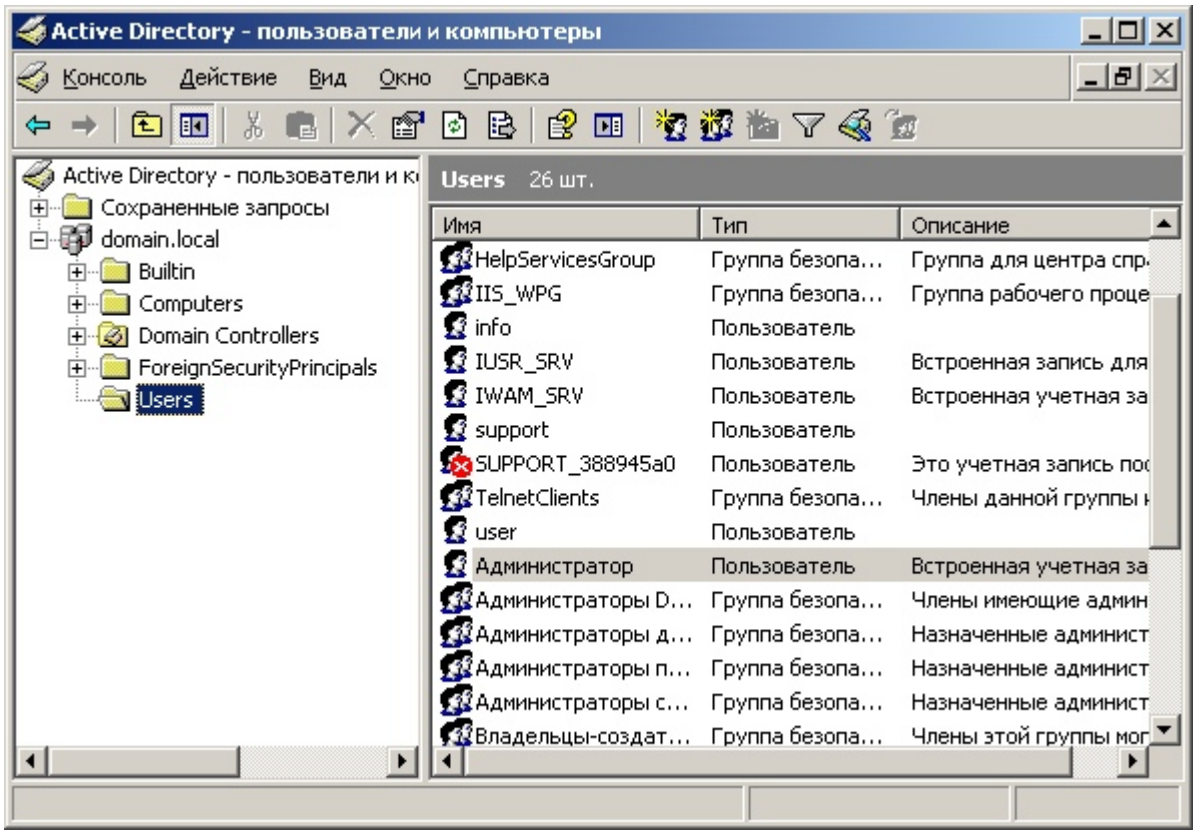


Рис. 30. Окно Active Directory – пользователи и группы

19. Командой **Действия | Создать | Пользователь**, вызовите окно создания нового доменного пользователя.
20. В новом окне укажите имя пользователя, его учетной записи для входа (например, admin). Щелкните по кнопке **Далее**.
21. Задайте пароль учетной записи и выставите при необходимости дополнительные опции. Щелкните по кнопке **Далее**.
22. Аналогичными действиями создайте пользователя user.
23. Войдите на машину Win1.
24. Откройте окно **Система** (**Панель управления | Система**).
25. Перейдите на вкладку **Имя компьютера**.
26. Добавьте машину Win1 в домен domain.
27. Щелкните по кнопке **Изменить…** .
28. В группе **Является членом** установите переключатель **Домена**.
29. Введите имя домена domain.
30. По необходимости введите новое имя компьютера в домене.

31. Щелкните по кнопке **Ок**.
32. В новом окне введите имя пользователя, имеющего права на добавление компьютера в домен. В качестве имени введите полное имя пользователя, включая домен: domain\Администратор, и введите пароль. Щелкните по кнопке **Ок**.
33. Дождитесь сообщения об успешном добавлении узла в домен. Щелкните по кнопке **Ок**.
34. Не перезагружайте компьютер.
35. Добавьте учетную запись на машину Win1 для доменного пользователя admin
    a. Откройте окно со списком пользователей (**Панель управления | Пользователи и роли**)
    b. Щелкните по кнопке **Добавить…** .
    c. Введите имя пользователя admin и домен domain. Щелкните по кнопке **Далее**.
    d. Укажите переключателем группу пользователя **Администраторы**. Щелкните по кнопке **Готово**.
36. Закройте окно **Пользователи и роли**. Перезагрузите компьютер Win1.
37. При входе под учетной записью заметьте изменения окна входа. Теперь указывается имя учетной записи и домен для входа (или ***имя_комьютера*** **(этот компьютер)** в случае локального входа). Укажите имя учетной записи admin и выберите домен DOAMIN. Щелкните по кнопке **Ок**.
38. Создайте на диске `C:` папку `Общая`.
39. В свойствах папки на вкладке Доступ добавьте доменного пользователя admin и user в список. Для этого:
    a. Щелкните по кнопке **Добавить…** .
    b. Введите через точку с запятой имена учетных записей (admin; user) и щелкните по кнопке **Проверить**.
    c. Убедитесь, что система правильно определила доменные учетные записи и щелкните по кнопке **Ок**.
    d. Установите полные права для этих двух пользователей. Щелкните по кнопке **Ок**.
40. Включите машину Win2. Добавьте ее в домен. Создайте на ней учетную запись для доменного пользователя user, добавив его в группу **Ограниченные пользователи**.
41. Зайдите на машину Win2 под доменным пользователем user.
42. Откройте сетевое окружение, найдите домен domain. Найдите в нем компьютер Win1. Откройте его ресурсы.

43. Зайдите в папку `Общая`, и попытайтесь создать в ней каталог или файл.
44. На машине Win1 откройте свойства каталога `Общая`, на вкладке **Безопасность** определите права для доменных пользователей admin и user аналогично пункту 39.
45. На машине Win2 попробуйте зайти в папку `Общая` на машине Win1 и создать в ней каталоги и документы.
46. На машине Win2 выйдите из-под учетной записи user и войдите под учетной записью admin.
47. Через сетевое окружение зайдите в папку `Общая` на машине Win1 и создайте в ней каталоги и документы.
48. Попробуйте создать на машине Win2 в папке `Program Files` собственный каталог.
49. Создайте учетную запись на машине Win2 для доменного пользователя admin, добавив в локальную группу `Администраторы`.
50. Выйдите и войдите снова под учетной записью admin на машину Win2.
51. Попробуйте создать на машине Win2 в папке `Program Files` собственный каталог.
52. На машине Win1 под учетной записью admin создайте в корневом каталоге диска `C:` папку `Distrib`.
53. Откройте папку `Distrib` в общий доступ, назначив полные права пользователю admin и права на чтения пользователю user. Удалите из списка доступа группу **Все**.
54. Авторизуйтесь на машине Win2 под пользователем user. Для этого войдите на данный компьютер под этим пользователем, или запустите файловый менеджер от имени данного пользователя (в контекстном меню файла или ярлыка запуска выберите пункт **Запустить от имени…**).
55. Откройте в сетевом окружении компьютер Win1. Перейдите в папку `Distrib`.
56. На машине Win1 на вкладке **Безопасность** окна **Свойства** каталога `Distrib` установите полные права для пользователя admin, и поставьте полный запрет для пользователя user.
57. На машине Win2 под пользователем user откройте в сетевом окружении компьютер Win1. Перейдите в папку `Distrib`.
58. Проанализируйте результаты обращений к каталогу.

Практическая работа 30. Настройка proxy-сервера (Windows)

Цель работы: Изучить принципы работы службы proxy в ОС Windows. Получить навыки настройки службы proxy (на пример TrafficInspector) для обеспечения доступа клиентов к внешним ресурсам, настройки клиентских узлов для обращения к внешним ресурсам через службу proxy.

Объем времени: 2 ч.

Программное обеспечение: ISO-образ ОС Windows Server 2003, ISO-образ ОС Windows XP, ISO-образ ОС openSuSE.

Виртуальные машины:

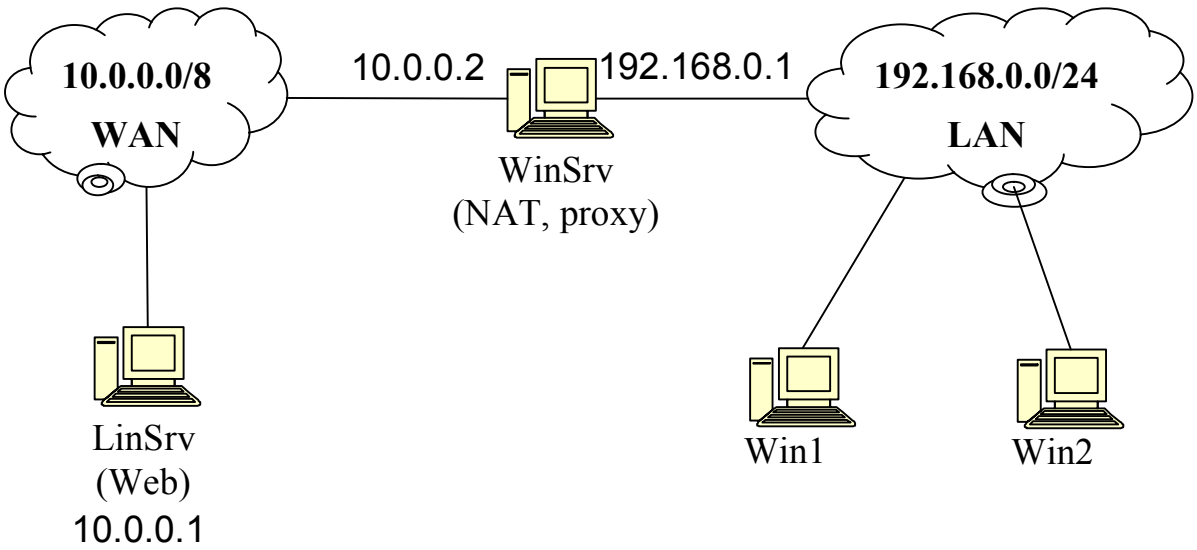
WinSrv (HDD: 4Gb; RAM: 128Mb; LAN0, LAN1: internal; ОС Windows Server 2003)

Win1 (HDD: 4Gb; RAM: 128Mb; LAN0: internal; ОС Windows XP)

Win2 (HDD: 4Gb; RAM: 128Mb; LAN0: internal; ОС Windows XP)

LinSrv (HDD: 8Gb; RAM: 256Mb; LAN0: internal; ОС openSuSE)

Схема сети:



Последовательность действий:

1. Организуйте сеть как показано на схеме. Реализуйте сетевую трансляцию адресов через службу RRAS, аналогично работе Настройка NAT (Windows) (стр. 216)
2. Установите на машину WinSrv приложение TrafficInspector 2.0. После установки приложение будет работать в демо-режиме с доступом для трех пользователей.

3. В разделе TrafficInspector запустите мастер конфигурации сервера щелчком по кнопке `Конфигуратор`. В ходе ответов на вопросы мастера укажите внешний и внутренний интерфейсы.
4. Откройте раздел **Биллинг | Клиенты**.
5. В контекстном меню раздела выберите пункт **Добавить клиента…**

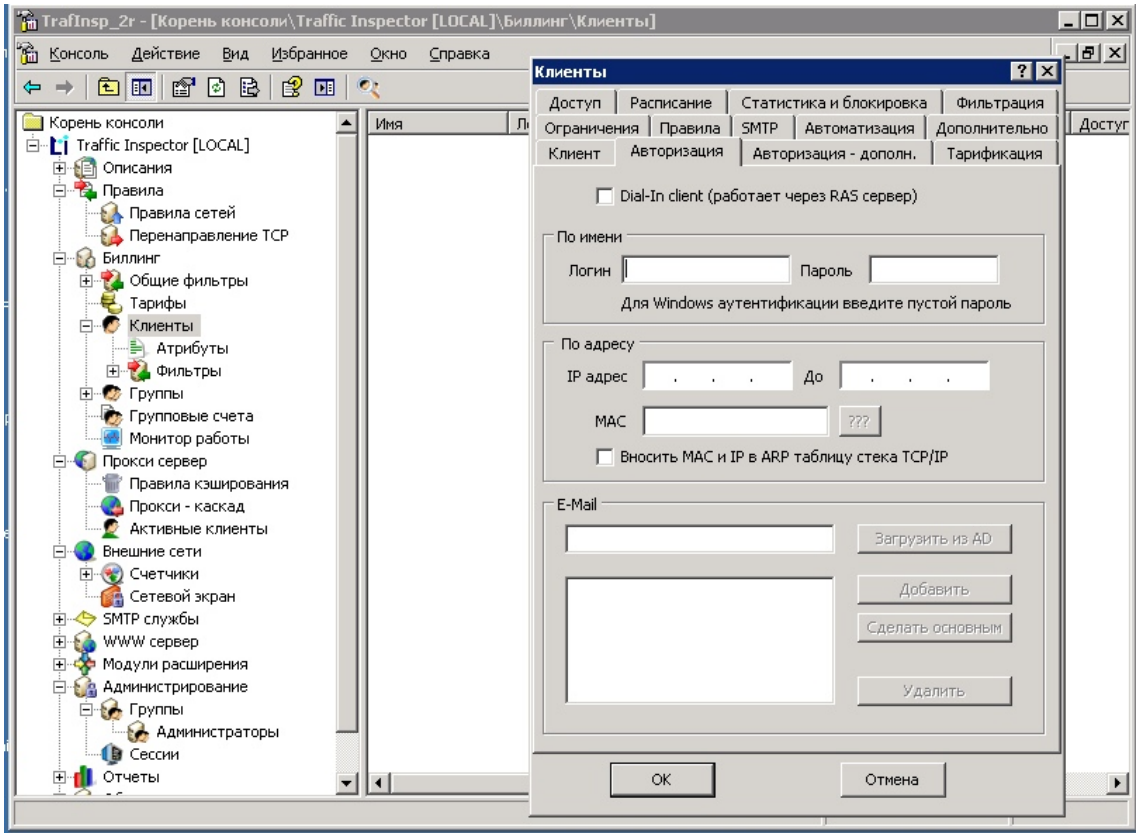


Рис. 31. Главное окно программы TrafficInspector

6. Создайте клиента с авторизацией по IP-адресу с неограниченным доступом по трафику:
    a. Введите имя клиента.
    b. В качестве авторизации клиента укажите его IP-адрес.
    c. На вкладке **Доступ** выберите пункт **Безлимитный**.
    d. Щелкните по кнопке **Ок**.
7. На машине Win1 под пользователем admin в настройках прокси-сервера Internet Explorer (**Сервис | Свойства обозревателя… | Подключения | Настройка LAN…**) укажите адрес и порт прокси-сервера (192.168.0.1:3128). Во всех окнах щелкните по кнопке **Ок**.
8. На машине Win1 в браузере обратитесь по адресу http://10.0.0.1/index.php
9. Проверьте значение переменной `REMOTE_ADDR`. Убедитесь, что она имеет значение внешнего IP-адреса прокси-сервера (10.0.0.2).
10. Сделайте аналогичные настройки на Win2. Попробуйте перейти в браузере по адресу http://10.0.0.1/index.php.

11. На машине Win1 под пользователем user настройте адрес прокси-сервера. Перейдите в браузере по адресу http://10.0.0.1/index.php.
12. На машине WinSrv в настройках клиента прокси-сервера TrafficInspector измените авторизацию клиента на доменную, указав в качестве авторизации имя доменного пользователя (DOMAIN\admin), предварительно удалив IP-адрес из соответствующего поля.
13. Перейдите в браузере по адресу http://10.0.0.1/index.php на машине Win1 под доменным пользователем admin и user. Аналогичную проверку проделайте на машине Win2 под доменными пользователями. Проанализируйте результаты.

Практическая работа 31. Настройка proxy-сервера (Linux)

Цель работы: Изучить принципы работы службы proxy в ОС Linux. Получить навыки настройки службы proxy на примере пакета squid для обеспечения доступа клиентов к внешним ресурсам, настройки клиентских узлов для обращения к внешним ресурсам через службу proxy.

Объем времени: 1 ч.

Программное обеспечение: ISO-образ ОС Windows Server 2003, ISO-образ ОС SLAX Linux LiveCD, ISO-образ ОС openSuSE.

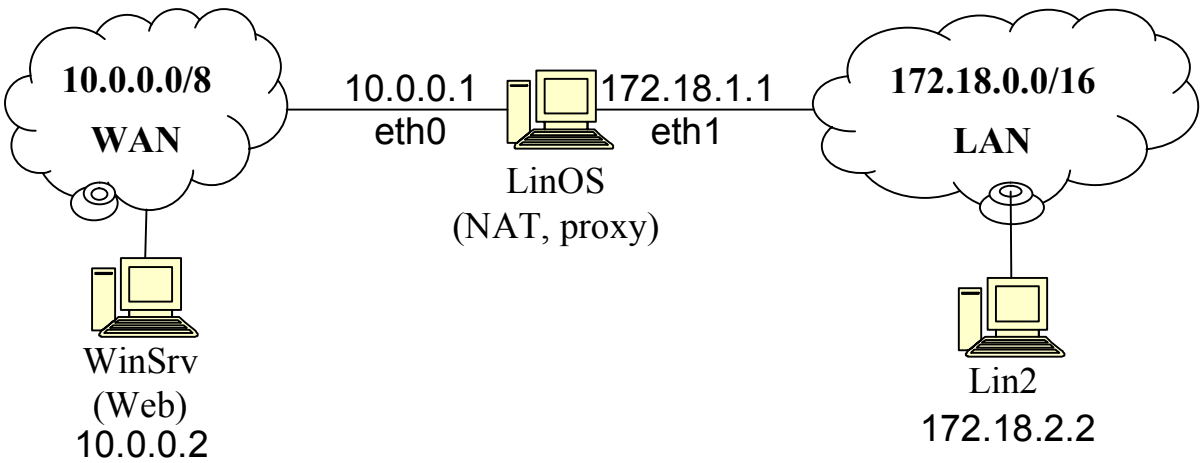
Виртуальные машины:

WinSrv (HDD: 4Gb; RAM: 128Mb; LAN0, LAN1: internal; ОС Windows Server 2003)

LinOS (HDD: –; RAM: 128Mb; LAN0, LAN1: internal; ОС SLAX Linux)

Lin2 (HDD: –; RAM: 128Mb; LAN0: internal; ОС SLAX Linux)

Схема сети:



Последовательность действий:

1. Запустите машины Lin2 и LinOS. Настройте сетевые интерфейсы.
2. Настройте шлюз по умолчанию для машины Lin2.
3. Включите WinSrv. Настройте сетевой интерфейс. Проверьте прохождение эхо-запросов между WinSrv и LinOS.
4. Настройте веб-сервера данного узла (WinSrv), откройте порт через брандмауэр Windows или службу RRAS.
5. Проверьте корректное отображение ресурсов веб-сервера с LinOS.
6. Установите пакет `squid` на машину LinSrv.

   ```
   #apt-get install squid
   ```

   Или

   ```
   #zypper install squid
   ```

7. Отредактируйте файл `/etc/squid/squid.conf`

   a. Укажите порт для прокси

   ```
   http_port 3128
   ```

   b. Добавьте тип доступа в ACL

   ```
   acl myproba src 10.0.0.2/8 #доступ с IP
   ```

   c. Установите разрешения для ACL

   ```
   http_access allow myproba #разрешаем этому IP
   ```

8. Запустите службу squid:

   ```
   #/etc/init.d/squid restart
   ```

9. Откройте порт прокси в брандмауэре и закройте доступ напрямую к 80 порту, чтобы удостовериться, что HTTP трафик идет через прокси

   ```
   #iptables –I INPUT 1 –p tcp --dport 3128 –j ACCEPT
   ```

10. Настройте прокси на клиентской машине Lin2. Укажите настройки адреса прокси-сервера и его порта в системе или в браузере.
11. Откройте на машине Lin2 браузер и обратитесь по адресу http://10.0.0.2/index.php.
12. Проверьте значение переменной `REMOTE_ADDR`. Убедитесь, что она имеет значение внешнего IP-адреса прокси-сервера (10.0.0.1).

Библиографический список

1. Christopher Negus Linux Bible 2008 Edition: Boot Up to Ubuntu, Fedora, KNOPPIX, Debian, openSUSE, and 11 Other Distributions, Wiley Publishing, Inc., Canada, 2008, 891p.
2. Sander van Vugt The Definitive Guide to SUSE Linux Enterprise Server, APRESS, NY, 2006, 714p.
3. Stewart S. Miller Wi-Fi Security, NY: McGraw-Hill, 2003, 311p.
4. Адамс Б., Ченг Э. Руководство по междоменной многоадресатной маршрутизации. : Пер. с англ. — М.: Издательский дом «Вильямс», 2004. — 320 с.: ил.
5. Айвенс К. Компьютерные сети. Хитрости. — СПб.: Питер, 2006. — 298 с.
6. Амато Вито Основы организации сетей Cisco, том 1.: Пер. с англ. — М. : Издательский дом «Вильямс», 2002. — 512 с.
7. Амато Вито Основы организации сетей Cisco, том 2.: Пер. с англ. — М. : Издательский дом «Вильямс», 2002. — 464 с.: ил.
8. Бакланов И.Г. Технологии ADSL/ADSL2+ теория и практика применения. — М.: Метротек, 2007. — 384 с.
9. Бигелоу С. Сети: поиск неисправностей, поддержка и восстановление: Пер. с англ. — СПб.: БХВ-Петербург, 2005. — 1200 с.
10. Бруй В. В. , Карлов С. В. LINUX-сервер: пошаговые инструкции инсталляции и настройки. — М.: Изд-во СИП РИА, 2003. – 572 с.
11. Вишневский В.М. Широкополосные беспроводные сети передачи информации / В.М. Вишневский, А.И. Ляхов, С.Л. Портной, И.В. Шахнович, М.: Техносфера, 2005, 592с.
12. Гальперович Д. Инфраструктура кабельных сетей / http://ksaa.edu.ru/book/galper/galper.htm
13. Гарипова К.Г. Компьютерные сети как пример новых информационных технологий / Методическая разработка Гарипова К.Г., г. Сасово, 2004 г.
14. Гейер Дж. Беспроводные сети. Первый шаг : Пер. с англ. — М. : Издательский дом «Вильяме», 2005, 192 с.: ил.
15. Далхаймер К,, Уэлш М. Запускаем Linux, 5-е издание. — Пер. с англ. —ы СПб.: Символ-Плюс, 2008. — 992 с, ил.
16. Досталек Л., Кабелова А. TCP/IP и DNS в теории и на практике. Полное руководство / Пер.с чеш. Рус. Изд.под ред. М. В. Финкова и А.В. Анисимова. СПб.: Наука и Техника, 2006, 206 с.
17. Еаррет Д. Дж. Linux: основные команды. Карманный справочник / Пер. с англ. — М.: КУДИЦ-ОБРАЗ, 2005. — 288 с.

18. Жеретинцева Н.Н. Курс лекций по компьютерным сетям, Владивосток: ДВГМА, 2000 г., 158 с.

19. Колисниченко Д.Н. Ubuntu Linux. Краткое руководство пользователя. — СПб.: БХВ-Петербург, 2007. — 304 с.

20. Колисническо Д.Н. Сделай сам компьютерную сеть. Монтаж, настройка, обслуживание. СПб.: Наука и Техника, 2004, 400 с.

21. Леинванд Аллан Конфигурирование маршрутизаторов Cisco, 2-е изд.: Пер. с англ. — М. : Издательский дом «Вильямс», 2001. — 368 с. : ил.

22. Ленников А. Строим локальную сеть / http://www.neowin.net.ru

23. Лукацкий А. Безопасность без проводов // КомпьютерПресс. — 2004. — №5

24. Мелехи В.Ф. Вычислительные машины, системы и сети: учебник для студ.высш.учеб.заведений / В.Ф. Мелехин, Е.Г. Павловский. — 2-е изд.,стер. — М.: Издательский центр «Академия», 2007. — 560с.

25. Немет Эви, Снайдер Гарт, Трент Р. Хайн Руководство администратор Linux, 2-е издание.: Пер. с англ. — М.: ООО «И.Д. Вильямс», 2007. — 1072 с.

26. Оглтри Т. Firewall. Практическое применение межсетевых экранов: Пер. с англ. — М.: ДМК Пресс, 2001. — 400с.

27. Оглтри Т. Firewalls. Практическое применение межсетевых экранов: Пер. с англ., М.: ДМК Пресс, 2001, 400с.: ил.

28. Олифер В.Г. Компьютерные сети. Принципы, технологии, протоколы / В.Г. Олифер, Н.А. Олифер, СПб.: Питер, 2001, 672 с.: ил.

29. Олифер В.Г. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. / В.Г. Олифер, Н.А. Олифер, 3-е изд., СПб.: Питер, 2006, 958 с.: ил.

30. Палмер М. Проектирование и внедрение компьютерных сетей. Учебный курс. / Майкл Палмер, Роберт Брюс Синклер., 2е изд., СПб.: БХВ-Петербург, 2005г.

31. Пахомов С. Механизмы коллективного доступа в сетях 802.11 // КомпьютерПресс. — 2004. — №5

32. Пескова С.А. Сети телекоммуникации: учеб. Пособие для студ.высш.учеб.заведений / С.А. Пескова, А.В. Кузин, А.Н. Волков. — 2-е изд., стер. — М.: Издательский центр «Академия», 2007. — 352 с.

33. Поляк-Баргинский А.В. Администрирование сети на примерах. — СПб.: БХВ-Петербург, 2005. — 320 с.

34. Рыжов К.В. Сто великих изобретений, М.: «Вече», 1999, 527 с.: ил.

35. Семенов А. Стандарт 802.11n — путь к новому поколению WLAN // КомпьютерПресс. — 2005. — №5

36. Семенов А.Б. Проектирование и расчет структурированных кабельных систем и их компонентов. –М.: ДМК Пресс; М.: Компания АйТи, 2003. — 416+16 с.: ил.

37. Скловская С.Л. Команды Linux. Справочник, 3-е изд., перераб. и доп. / С.Л. Скловская. — СПб.: ООО «ДиаСофтЮП», 2004. — 848 с.

38. Скляр Б. Цифровая связь. Теоретические основы и практическое применение. Изд. 2-е, испр.: Пер. с англ. – М.: Издательский дом «Вильямс», 2003. – 1104 с.

39. Стахнов А. А. Сеть для офиса и Linux-сервер своими руками. — СПб.: БХВ-Петербург, 2006. — 320 с : ил.

40. Степанов В., Интернет в профессиональной информационной деятельности / http://www.libs.ru/doc/textbook/recomend.html

41. Столлингс В. Передача данных. 4-е изд, СПб.: Питер, 2004, 750 с: ил.

42. Таненбаум Э. Компьютерные сети, 4-е изд., СПб.: Питер, 2003, 992 с.: ил.

43. Таненбаум Э., Вудхалл А. Операционные системы: разработка и реализация. — СПб.: Питер, 2006. — 576 с.

44. Уильям Р. Станек. Internet Information Services 5.0. Справочник администратора. / Пер. с англ. — М.: Издательско-торговый дом «Русская Редакция», 2002. — 464 с.: ил.

45. Хелеби С., Мак-Ферсон Д. Принципы маршрутизации в Internet, 2-е издание. : Пер. с англ. М. : Издательский дом «Вильямс», 2001. — 448 с. : ил.

46. Холмогоров В. Компьютерная сеть своими руками. Самоучитель / В. Холмогоров. — СПб.: Питер, 2003. — 171 с.: ил.

47. Шредер К. Linux. Сборник рецептов. — СПб.: Питер, 2006. — 432 с: ил.

Буторин Денис Николаевич, кандидат педагогических наук, имеет большой опыт преподавания информационных технологий в филиале Красноярского государственного педагогического университета им. В.П. Астафьева в г. Ачинске. Участник ряда всероссийских и международных конференций. Автор более 30 научных работ, монографии “Машинная реализация методики проблемного обучения студентов информатике в программной среде”, учебного пособия “Компьютерные сети. Практической курс”, учебно-методического пособия “Методика использования в педагогической практике открытой научной образовательной среды openSEE” и книги “MS Agent и Speech API в Delphi” (издательство БХВ-Петербург).

В учебном пособии:

- ☑ **Имеется подробная историческая справка о появлении и развитии компьютерных сетей.**
- ☑ **Вся теоретическая информация представлена в виде схем и диаграмм.**
- ☑ **Материал представлен в высоко визуализированной форме для его более качественного понимания.**
- ☑ **Содержание построено на основе диалектического и проблемно-ориентированного подхода.**
- ☑ **Теоретические сведения активно и быстро погружают в предметную область.**
- ☑ **Представлено 30 практических работ, покрывающих большинство разделов настройки локальных сетей и прикладных служб.**
- ☑ **Все работы имеют кросс-платформенную составляющую и ориентированы на платформы Windows и Linux**
- ☑ **Практические работы интенсивно вовлекают обучаемого в познавательную и поисковую деятельность.**