ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ

Ордена Трудового Красного Знамени федеральное государственное
бюджетное образовательное учреждение высшего образования
Московский технический университет связи и информатики

Кафедра технологий электронного обмена данными

УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ

ОСНОВЫ ТЕХНОЛОГИЙ СЕТИ ИНТЕРНЕТ

Москва 2019

УДК 004.77
ББК 32.971.353

Учебно-методическое пособие
Основы технологий сети Интернет
Направление подготовки: 11.03.02 – Инфокоммуникационные технологии и системы связи

Составители: А.С. Кремер, А.В. Иванюк, И.В. Захаров, К.А. Севрук, А.А. Малюк, А.В. Смирнов, М.И. Кухаренко, Г.В. Курячий, И.И. Решенина, А.А. Волков, В.С. Малиночкин, М.Р. Магафуров, Д.А. Ермолаев, Т.В. Черствов.

Издание утверждено советом факультета Сети и системы связи Московского технического университета связи и информатики.
Протокол № 10 от 21 мая 2019 года.

ISBN

Рецензент: В.А. Ерёменко, к.т.н., доцент

Содержание

Введение 5

1. Общие вопросы функционирования сети Интернет 7

1.1. Анализ архитектурных принципов построения, а также действующего порядка управления сетью Интернет 7

1.2. Совершенствование принципов обеспечения информационной безопасности функционирования сети Интернет 14

1.3. Порядок регулирования сети Интернет на международном уровне 16

1.4. Нормативное правовое регулирование сети Интернет на национальном уровне 28

1.5. Оценка внутренней связности российского сегмента сети Интернет .. 38

1.6. Принципы обеспечения интеграции российского сегмента сети Интернет в национальную и международную ИКТ-инфраструктуру. Оценка экономического и политического значения такой интеграции 42

1.7. Гуманитарные аспекты работы в сети Интернет 49

1.8. Свободные лицензии и их ограничения. Преимущества и недостатки открытой модели разработки программного обеспечения. 68

2. Протоколы сети Интернет 88

2.1. Модель взаимодействия открытых систем и стек протоколов TCP/IP 88

2.2. Стандарты Ethernet 94

2.3. Интернет-протокол версии 4 102

2.4. Интернет-протокол версии 6 109

2.5. Протокол преобразования адресов ARP 112

2.6. Протокол динамического конфигурирования хоста DHCP 116

2.7. Трансляция сетевых адресов NAT 121

2.8. Протокол передачи команд и сообщений ICMP 125

2.9. Маршрутизация 128

2.10. Протоколы маршрутизации RIP, OSPF, BGP 133

2.11. Протокол управления передачей TCP 152

2.12. Протокол пользовательских дейтаграмм UDP 161

2.13. Система доменных имён DNS 163

2.14. Протокол виртуального терминала TELNET 172

2.15. Протоколы передачи файлов FTP и TFTP 176
2.16. Протоколы электронной почты: SMTP, POP, IMAP 182
2.17. Гипертекстовый протокол HTTP 187
2.18. Протокол безопасного доступа SSH 191
Список литературы 195

Введение

В процессе происходящей в настоящее время в Российской Федерации цифровой трансформации подавляющее большинство систем принятия решений и бизнес-процессов в ключевых отраслях экономики и сфере государственного управления реализуются или планируются к реализации с использованием информационно-коммуникационных технологий сети Интернет. В различных информационных системах, доступ к которым осуществляется с использованием сети Интернет, уже сейчас хранятся и обрабатываются значительные объемы информации, в том числе касающейся вопросов государственной политики и обороны, финансовой и научно-технической сферы, частной жизни граждан. Пользователями российского сегмента сети Интернет в 2017 году являлись более 85 млн. человек.

Глобализация современных информационно-коммуникационных сетей и информационных систем, вынужденное применение при их построении иностранного оборудования и заимствованного программного обеспечения, а также существенное увеличение количества автоматизированных систем управления производственными и технологическими процессами в условиях интенсивного совершенствования средств и методов применения информационных и коммуникационных технологий в противоправных целях, формируют новые угрозы безопасности Российской Федерации, усложняют решение задач по обеспечению защиты интересов граждан и государства в информационной сфере. В этих условиях особое значение должно уделяться подготовке квалифицированных кадров, способных разрабатывать и внедрять отечественные информационно-коммуникационные технологии.

На содержание этой деятельности существенное влияние оказывают агрессивная политика крупнейших иностранных вендоров-производителей программного обеспечения и оборудования, а также недостаточное внимание, уделяемое технологическому оснащению процесса обучения. В большинстве случаев используются демонстрационные возможности обычного компьютера с комментариями, основанными на доминирующей в информационном пространстве информации о зарубежных технических решениях. Последствием сложившейся ситуации является повсеместное внедрение в учебный процесс изучения продукции иностранных производителей, сформировавших развитую экосистему продвижения своих решений. При этом не уделяется достаточного внимания изучению

возможностей и перспектив внедрения отечественных технологий. Существующая организация учебного процесса не способствует решению важнейших задач подготовки специалистов в сфере развития национальной ИКТ-инфраструктуры с использованием отечественных технологий, а также обеспечения безопасности и доверия при её использовании. При этом такой мощный ресурс, как образование, используется недостаточно эффективно и целенаправленно для содействия решению задач обеспечения технологической независимости и информационной безопасности Российской Федерации.

Важным направлением решения этой проблемы является создание готовых унифицированных решений, которые могут быть использованы при формировании и реализации образовательных программ. Такой подход обеспечит повышение качества изучения интернет-технологий и будет способствовать повышению привлекательности и конкурентоспособности использующих такие решения образовательных организаций. Изучение интернет-технологий станет целенаправленным, то есть появятся основания для планомерного повышения технической грамотности и активности специалистов по разработке отечественных решений в области интернет-технологий и их применению в условиях цифрового развития.

С этой целью на базовой кафедре общественного-государственного объединения «Ассоциация документальной электросвязи» подготовлен и введен в эксплуатацию типовой отечественный модуль изучения интернет-технологий (ТОМИИТ), который стал основой для формирования профессиональных знаний и компетенций в области построения, развития и управления сетью Интернет. Важнейшей составляющей данного проекта является оснащение модуля российскими программно-аппаратными средствами. Неотъемлемой частью ТОМИИТ являются лабораторный практикум по курсу «Основы технологий сети Интернет», а также настоящее учебно-методическое пособие.

В первой главе пособия даются общие сведения об истории развития сети Интернет, принципах его регулирования, гуманитарных аспектах работы в сети Интернет, а также информация об открытой модели разработки программного обеспечения.

Во второй главе пособия приведена информация о наиболее распространённых протоколах сети Интернет.

Авторы благодарят АНО «Координационный центр национального домена сети Интернет» за оказанную помощь в реализации проекта.

1. Общие вопросы функционирования сети Интернет

1.1. Анализ архитектурных принципов построения, а также действующего порядка управления сетью Интернет

Идея объединения компьютерных сетей появилась раньше, чем такие сети были созданы. Но только после того, как несколько компьютерных сетей были построены реально, проблема разработки архитектуры межсетевого взаимодействия – интернет – стала превращаться в центральную для разработчиков.

Созданные разными коллективами отдельные компьютерные сети в течение нескольких лет работали независимо друг от друга, но в исследовательском сообществе быстро осознали важность межсетевой передачи пакетов данных.

Создание первой компьютерной сети, известной как ARPANET, финансировало Агентство передовых оборонных исследовательских проектов (DARPA) Министерства обороны США. И уже вскоре, в ходе разработки двух дополнительных сетей передачи данных (наземной радиосети и спутниковой сети), в DARPA было принято решение работать над задачей объединения сетей, что и привело, в конечном счете, к созданию глобальной информационной сети, известной сегодня как Интернет.

Изначально архитектура сети Интернет была призвана обеспечить взаимную связь систем с различными типами пакетов передаваемых данных и согласованную работу различных компьютеров в этих сетях. Каждая входящая в состав Интернета сеть и ее компьютеры имели уникальные идентификаторы в данной архитектуре. Основной идеей было определение простого процесса, в котором биты, отправляемые с одного компьютера в виде пакетов данных, были бы надежно и эффективно доставлены на любой другой компьютер по его уникальному для всех объединенных сетей имени.

Предполагалось, что отдельные сети различных типов могут иметь различные размеры пакетов, интерфейсы, методы защиты от ошибок, правила маршрутизации и организацию управления. Этот подход позволил встроить в архитектуру новые сети, которые могли быть созданы и разработаны в будущем. Хотя компонентами сети Интернет должны были стать сети, предоставляемые одним или несколькими провайдерами и службами предоставления вычислительных услуг (изначально это были

научные работники и их организации), в конечном итоге сюда вошли и всевозможные устройства и программные службы провайдеров.

Сущностью первичной архитектуры Интернет стало специфицированное множество протоколов и процедур, позволяющих компонентам объединенных сетей взаимодействовать. Данная архитектура опиралась на систему идентификации, представленную IP адресами и соответствующим интернет-протоколом (IP). Каждый передаваемый между сетями пакет данных должен был содержать, как минимум, IP адрес компьютера-получателя. В первичной архитектуре не было указано, как данная сеть должна маршрутизировать такие пакеты данных, хотя было ясно, что потребуется средство маршрутизации по общей системе сетей. Для этого, а также для обеспечения согласования между сетями, было введено понятие «шлюз». Такой шлюз, ныне известный как маршрутизатор, должен был принимать пакеты данных из сети, обрабатывать IP-адрес, содержащийся в пакете и определять, к какому маршрутизатору пакет данных должен быть направлен. Этот процесс должен был повторяться, пока пакет не поступал по назначению.

Разрабатывались также процессы фрагментации данных в случае превышения сетевых ограничений для размеров пакетов данных, объединения таких фрагментов в точке назначения, защиты от ошибок и работы с возникающими копиями пакетов. Некоторые из этих задач частично выполнялись маршрутизаторами, но изначально их решали соединяемые компьютеры, использующие протокол управления передачей (протокол TCP). Изначально, именно протокол TCP, помимо прочих задач, должен был генерировать IP-пакеты, а также осуществлять их отправку и получение. В ходе дальнейшего развития функции обработки IP были отделены, и протокол TCP превратился в TCP/IP.

В сущности, объединение компьютеров в единую сеть Интернет было описано протоколами и процедурами, позволяющими взаимодействовать различным сетям и компьютерам. Отдельные организации могли разрабатывать и использовать свои собственные сети, вычислительные сервисы и применять устройства различных типов, если они отвечали требованиям соответствующих протоколов. Рост сети Интернет происходил естественным путем, и его общее техническое руководство было независимо от менеджмента какой-либо входящей в него сети или сервиса. Именно это и сыграло решающую роль в глобальном расширении сети Интернет, поскольку каждый участник мог нести ответственность за управление собственными ресурсами.

В то же время существовал ряд причин, по которым должна была осуществляться координация управления работой этой глобальной информационной системы. Главной из них являлась необходимость присвоения IP и сетевых адресов таким образом, чтобы исключить их конфликты. Началась работа над стандартами сети Интернет, которую в конце 1983 года возглавила Рабочая группа по стандартам для сети Интернет (IETF). В 1998 году, надзор над некоторыми процессами перешел к вновь созданной организации, названной Корпорация по присвоению имен и адресов в Интернете (ICANN), которая до настоящего времени контролируется правительством США. О ней будет рассказано позже.

В начале 1970-х, когда началась работа над протоколами Интернет, многие полагали, что интерес к взаимосвязанным компьютерным сетям будет ограниченным. Коммерческие системы были все еще очень дороги, а персональные компьютеры еще не вышли на рынок. Дешевые микропроцессоры находились на ранней стадии развития. Тем не менее, за очень короткое время исследовательскому сообществу стали доступны первые компактные рабочие станции и эффективные локальные сети, такие, как Ethernet. Это привело к тому, что представление о том, что может существовать только небольшое количество компьютеров и сетей связи, было опровергнуто быстрым ростом числа объединяемых компьютеров и их совершенствованием. При этом базовая архитектура сети Интернет оказалась способной к масштабированию.

Изначально IP адрес (называемый IPv4) имел длину 32 бита, из которых только 8 бит использовались для идентификации сети, а остальные 24 бита - для определения хоста в этой сети. В то время такой размер адресного пространства казался просто огромным, но вскоре стало очевидно, что могут появиться тысячи отдельных локальных сетей Ethernet и еще большее число персональных компьютеров и рабочих станций.

В исследовательском сообществе появились планы изменить интерпретацию IPv4 адресов, чтобы создать большие, средние и малые сети, названные сетями класса A, B и C. Это повлекло за собой необходимость создания средств определения класса IP адреса и соответствующей интерпретации битов. Могло существовать небольшое количество сетей класса A, каждая из которых имела большое количество IP адресов, и большое количество сетей класса C, каждая из которых имела малое количество IP адресов. Сети класса B находились посередине.

Со временем стало ясно, что необходимы специальные сетевые имена, чтобы пользователям не приходилось запоминать отдельные IP адреса. Первоначальный способ присвоения имен был принят в сети ARPANET, когда отдельным компьютерам или хост-машинам присваивались 16-битные ARPANET адреса. Имена выбирались специалистами DARPA совместно с разработчиками сайтов, а преобразование имен в 16-битные адреса осуществлялось в NIC – Сетевом Информационном Центре Стэнфордского Исследовательского Института (SRI). Время от времени сетевые сайты загружали файл, называемый «host.txt», отправляемый NIC для получения самых последних преобразований. Когда произошел переход на протокол TCP/IP, стала использоваться конструкция «имя домена-точка-arpa» для различения этих двух блоков. Например, имя ucla предназначалось для обозначения 16-битного адреса в файле host.txt, а имя ucla.arpa использовалось для 32-битного адреса (который изначально включал 16-битный адрес с добавлением идентификатора сети).

Когда число компьютеров значительно возросло и стало ясно, что файл host.txt становится слишком большим для разово загружаемого файла, появилась «система доменных имен». Вместо домена одного верхнего уровня (т.е. «точка arpa»), были введены еще семь базовых доменов (т.е. gTLDs), а именно: .com, .net, .org, .edu, .mil, .gov, и .int. Каждый из этих доменов был доступен в интерактивном режиме для получения информации об именах отдельных доменов под общим именем gTLD. Далее были введены домены с кодами стран, и с тех пор дополнительные домены gTLDs стали официально назначаться ICANN.

Позднее появилась WWW - всемирная паутина («World Wide Web»), которая стала одной из самых важных и наиболее популярных прикладных систем, использующих для своего функционирования Интернет. Благодаря предложенной комбинации имен доменов с именами файлов, которая получила название унифицированного указателя ресурса (URL), процесс доступа к файлам, содержащим структурированные данные по всей сети Интернет, был радикально упрощен. Вместо того, чтобы узнавать и вызывать процедуру доступа к различным файлам на удаленных компьютерах, все необходимые процессы инициировались автоматически по умолчанию при клике на URL. Стандартизованное структурирование данных позволило эффективно отображать информацию с адаптацией ее к размерам окна пользователя. Постоянно продолжающееся совершенствование WWW направлено на развитие ее

функциональности и на настоящий момент она остается главным средством доступа к информации в сети Интернет.

К концу 1990 годов влияние сети Интернет стало ощутимым по всему земному шару. Многие страны строили планы по ее использованию в самых разных целях, начиная от здравоохранения, образования и распространения информации, до стратегических коммуникаций и бизнеса. Поскольку потенциал Интернет является без сомнения огромным, стали задавать вопрос «а кто руководит Сетью?» Последовала серия консультаций, которая в свою очередь привела к двум собраниям Всемирногосаммита по вопросам информационного общества (WSIS) в Женеве в 2003 г. и в Тунисе в 2005 г. для обсуждения связанных с Интернетом проблем.

Саммит обнажил недовольство текущим положением дел со стороны многих правительств. Например, часть правительств развивающихся стран хотели бы видеть ICANN в рамках МСЭ, что позволило бы им участвовать в принятии решений равноправно с правительством США.

Конечно, в рамках WSIS решения такого рода не могли быть приняты. Самое существенное, о чем удалось договориться, — это проведение ежегодных конференций, где различные заинтересованные стороны, включая бизнесменов, пользователей, правительства и техническое сообщество, могли бы неформально обсуждать вопросы управления, координации Интернета и принятия решений. Эти конференции получили название «Форум по управлению Интернетом» (Internet Governance Forum, IGF). Важной особенностью IGF является то, что, действуя под эгидой ООН, форум не имеет мандата принятия решений и резолюций. С одной стороны, это делает работу форума менее формальной, позволяет фокусироваться на реальных проблемах, а не на политической интриге. С другой стороны, отсутствует реальный стимул поиска общей точки зрения и возможных компромиссов, что значительно снижает эффективность этих мероприятий.

В августе 1990 г. было внесено предложение о создании Сетевого Координационного центра RIPE (RIPE NCC) со следующими задачами:

— создание и обслуживание Европейской IP-регистратуры в рамках архитектуры, предложенной в RFC 1174;

— информационное обслуживание сетей;

— административная поддержка RIPE.

Уже в сентябре 1992 г. RIPE NCC начал обслуживать запросы на получение адресного пространства от европейских организаций.

Осенью 1996 года вышел документ «Политика и процедуры европейской интернет-регистратуры» (RIPE-140), который представил 4 основных принципа распределения адресного пространства.

1. Уникальность — основополагающее требование для глобальной системы распределения адресов. Каждый присвоенный адрес должен быть уникальным в глобальной сети Интернет.

2. Агрегируемость — иерархическое распределение адресов, позволяющее оптимизировать глобальную систему маршрутизации. Распределение адресных ресурсов, учитывающее топологию сети и взаимоотношения провайдер-клиент, позволяет оптимизировать маршруты и, как следствие, уменьшить нагрузку на глобальную систему маршрутизации.

3. Сохранение — распределение ресурсов «по потребностям», минимизация неиспользуемых запасов.

4. Регистрация — регистрация распределенных и присвоенных адресов в общедоступной базе данных для поддержки уникальности и решения сетевых проблем на любом уровне.

Эти принципы остались неизменными до сегодняшнего дня. Основные новации в политике распределения адресов с тех пор касались выбора параметров, позволяющих обеспечить оптимальный баланс между противоречащими принципами агрегируемости и сохранения. По существу этих параметров три:

1. Минимальный размер распределяемого пространства. Чем больше минимальный размер, тем выше вероятность неиспользуемых запасов.

2. Временной интервал планирования, используемый для демонстрации потребности в ресурсах. Чем больше этот интервал, тем больше последовательного адресного пространства может получить сервис-провайдер, тем меньше различных блоков необходимо анонсировать провайдеру, тем меньше записей в глобальной таблице маршрутизации. В настоящее время потребность определяется на базе последующих 12 месяцев.

3. Процент использования распределенных ресурсов (утилизация). Для получения последующего блока адресов провайдер должен продемонстрировать, что 80% адресного пространства присвоено и используется. В случае IPv6 параметр утилизации немного более сложный, но суть его та же. Чем больше этот параметр, тем меньше запасы и скорость потребления адресных ресурсов.

Протокол IP в сетевой модели TCP/IP не напрасно называется уровнем Интернета. Можно сказать, что он приводит к общему знаменателю всю структуру Всемирной сети. Именно на уровне протокола IP взаимодействуют разнородные по своей архитектуре технологии н топологии сети, а на более высоком уровне на плодородной почве IP бурно развиваются и транспортные протоколы, и особенно протоколы приложений. Протокол IP является единственным универсальным требованием для подключения к Интернету. «Подключения» в широком смысле этого слова, ведь, подключаясь к Интернету, сеть или устройство по определению становится частью Интернета, расширяя его связность и функциональность.

IP можно по праву назвать универсальным коннектором. Современные технологии цифровой передачи данных обеспечивают немыслимую ранее пропускную способность, каждый день нас удивляют новейшие приложения для ПК и мобильных устройств — по все эти впечатляющие изменения происходят на уровнях ниже и выше IP. При этом новая версия протокола IPv6, открывающая новые возможности роста и инноваций, внедряется недостаточно быстро. Это неудивительно: обновление фундамента, да еще в такой самоорганизующейся среде, как Интернет, — задача чрезвычайно сложная, требующая усилий многих заинтересованных сторон, а также огромных затрат времени.

Рассказывая об архитектуре построения и порядке управления Интернета необходимо отметить роль глобальной системы доменных имен – DNS (Domain Name System). Глобальная система доменных имен является фундаментальным элементом Интернета. Эта система позволяет клиенту получить информацию, связанную с запрашиваемым доменным именем. Доменное имя — лучше запоминаемый, мнемонический идентификатор ресурса Сети (например, веб-сервера), в отличие от IP-адреса, записываемого в числовом виде. Наиболее распространенным запросом, обслуживаемым DNS, является получение IP-адреса устройства, связанного с именем. Поэтому функцию DNS также называют трансляцией имен в адреса.

DNS определяется набором протоколов, разработанных в IETF и опубликованных в документах RFC (Request for Comments). С 1987 г., когда была завершена работа над основной спецификацией сегодняшней DNS (RFC 1034 и RFC 1035), до настоящего времени было выпущено более 500 RFC, определяющих дополнительные функции системы или так или иначе связанные с ее работой. Но DNS также и глобальная

распределенная база данных, хранящая сотни миллионом имен и связанных с ними ресурсов. Развиваясь вместе с самим Интернетом, DNS сегодня обслуживается более чем 16 млн серверов, обрабатывая несколько десятков миллионов запросов в секунду. Подробнее о DNS вы узнаете из раздела 2.13 пособия.

1.2. Совершенствование принципов обеспечения информационной безопасности функционирования сети Интернет

Актуальность задач обеспечения доверия и безопасности при использовании сети Интернет повышается по мере расширения ее использования в целях цифровой трансформации государственного управления, промышленного производства, науки, образования, здравоохранения, улучшения условий и качества жизни граждан.

Можно выделить три основных направления обеспечения информационной безопасности сети Интернет:

- нормативное правовое (законодательное) регулирование;
- техническое регулирование;
- образовательная, просветительская и культурологическая деятельность.

Законодательство регулирует отношения, возникающие при использовании сети Интернет. Важность законодательства состоит в обязательности исполнения его требований. Учитывая, что сеть Интернет является международной, угрозы ее безопасному функционированию могут иметь международный характер. Противодействие таким угрозам требует формирования и применения международного законодательства, что является значительно более сложным и трудоемким по сравнению с формированием и применением законодательства на национальном уровне.

В связи с указанными ограничениями законодательного регулирования обеспечения доверия и безопасности при использовании сети Интернет все большее значение приобретает техническое регулирование (стандартизация). Стандарты регулируют технологические, организационные и процедурные аспекты обеспечения информационной безопасности. Хотя применение стандартов носит добровольный характер, их внедрение стимулируется стремлением использовать эффективные, совместимые и апробированные решения обеспечения доверия и

безопасности при использовании сети Интернет. Российским законодательством предусмотрена возможность изменения добровольного статуса стандарта на обязательный путем включения ссылки на стандарт в текст нормативного правового акта. Учитывая международный характер Интернета, особое значение приобретает участие в разработке международных стандартов и, в первую очередь, в рамках международного союза электросвязи (МСЭ). Преимуществом принимаемых в МСЭ стандартов является предусмотренная процедура их согласования с администрациями стран – членов МСЭ. Такая процедура необходима при принятии стандартов информационной безопасности и позволяет оценить отсутствие противоречий между проектом стандарта и национальным законодательством. Участие в международной стандартизации обеспечивает следующие преимущества:

– получение актуальной информации, опирающейся на лучшие мировые практики, по противодействию угрозам информационной безопасности;

– возможность отстаивать национальные интересы, а также формировать и поддерживать благоприятный имидж страны путем внесения профессиональных предложений;

– участие в международном разделении труда и продвижении отечественных средств информационной безопасности на рынки развивающихся стран.

Образовательная деятельность необходима для подготовки специалистов, готовых и способных решать задачи обеспечения доверия и безопасности при работе в сети Интернет на основе использования отечественных средств информационной безопасности. Просветительская и культурологическая деятельность содействует осознанному выполнению требований нормативного правового и технического регулирования, поддержанию и выполнению нормам кодексов поведения при работе в сети Интернет, формированию культуры информационной безопасности.

К базовым принципам регулирования информационной безопасности следует отнести.

1. На уровне национальной юрисдикции с целью обеспечения технологической независимости и информационной безопасности Российской Федерации принимаются законодательные и технические нормы регулирования и переход на преимущественное использование отечественных средств обеспечения информационной безопасности. При этом анализируются и учитываются международные законодательные и

технические нормы и решения на предмет минимизации рисков фрагментации и изоляции российского сегмента сети Интернет от международной ИКТ-инфраструктуры.

2. На международном уровне с целью содействия интеграции российского сегмента сети Интернет в международную ИКТ-инфраструктуру при обеспечении технологической независимости и информационной безопасности Российской Федерации обеспечивается полноценное и профессиональное участие в организациях и форумах, обсуждающих и принимающих правовые и технические нормы международного регулирования, осуществляется содействие распространению отечественных решений в сфере обеспечения информационной безопасности.

3. На национальном и международном уровнях обеспечивается ведение просветительской, культурологической и образовательной деятельности для подготовки научно-технических специалистов, заинтересованных и способных работать в сфере обеспечения информационной безопасности Российской Федерации.

1.3. Порядок регулирования сети Интернет на международном уровне

Система принятия решений и управления Интернетом в международном обиходе называется Internet Governance. До второй половины 90-х гг. прошлого века принятие решений в Интернете осуществляло техническое сообщество ,ядро которого составляли участники IAB и IETF. Ключевой фигурой являлся научный сотрудник Института информатики (ISI) Университета Южной Калифорнии (USC) Джон Постел, который руководил IANA — организацией по присвоению IP-адресов.Ситуации начала стремительно меняться с расширением Интернета и его коммерциализацией. В центре этой коммерциализации находилась DNS, а ключевую роль начала играть частная компания, связанная с оборонными агентствами США, — Network Solutions, Incorporated, или NSI.

До 1993 г., когда NSF (Национальный научный фонд, National Science Foundation — независимое агентство при правительстве США, отвечающее за развитие науки и технологий), принял на себя финансирование «гражданской» части Интернета, поддержка центральной регистратуры, сопровождавшей корневую зону DNS (DDN-NIC), осуществлялась

Министерством обороны США. Функции DDN-NIC начиная с 1991 г. выполнял подрядчик — компания NSI.

В 1993 г. NSF объявила тендер на осуществление регистрационных услуг DDN-NIC, победителем которого стала все та же компания NSI. Контракт предоставил NSI монопольные права на регистрацию доменов второго уровня в .COM, .ORG и .NET в порядке очередности получения заявок. Важно отметить, что NSI получила широкие операционные полномочия, но все-таки вопросы политики, в частности относительно регистрации доменов верхнего уровня, оставались за IANA.

Изначально регистрация производилась бесплатно, за счет гранта NSF. Однако по мере роста числа регистраций на повестку дня встали вопросы масштабирования. В 1995 г. NSF изменила положения договора, позволив NSI установить плату в $50 в год за доменное имя второго уровня. Бум приватизации и взрывного развития Интернета набирал обороты, и регистрация приняла глобальный характер. Соответственно, росли и доходы NSI.

Вопросы администрирования корневой зоны затрагивали интересы растущего числа различных групп — начиная от новообразованных компаний, которым требовалось значащее имя в Интернете, до правообладателей торговых марок, видевших в DNS как возможности, так и опасности. Правительства государств постепенно понимали, что эти вопросы граничат с их суверенными интересами.

Формально NSI не имела права вносить изменения в корневую зону, но компания по существу монопольно контролировала корневой уровень и чрезвычайно прибыльный бизнес доменов второго уровня. Это положение вещей доставляло серьезный дискомфорт всем остальным заинтересованным лицам. Для противодействия монополии в этой области можно было, например, создать дополнительные домены верхнего уровня. Но политика создания новых доменов отсутствовала — и было непонятно, кто же формально контролирует корень DNS.

Ситуация осложнялась тем фактом, что пятилетний договор между NSI и NSF истекал в 1998 г. В отсутствии NSF будущее управления корневой зоной было по меньшей мере непонятным. Для разрешения этой ситуации Джоном Постелом в 1996 году был предложен проект в группе под названием International Ad Hoc Committee (IAHC). Эта группа была создана под эгидой ISOC, IAB, IANA, ITU, INTA и WIPO для разработки альтернативного предложения, с которым она вышла годом позже. На нем стоит остановиться подробнее.

В предложении IAHC (и связанном с ним Протоколе о взаимопонимании gTLD-MOU) была представлена структура управления корневым уровнем DNS. В этой структуре не предусматривалось создание множества новых доменов верхнего уровня и связанных с ними регистратур-регистраторов, работающих по модели NSI, но конкурирующих между собой, как это было предложено Постелом в его проекте. Вместо этого комитет IAHC предлагал разделить функции регистратуры и регистратора, а также стимулировать конкуренцию на уровне регистраторов.

В рамках предложения предполагалось создание всего нескольких дополнительных доменов верхнего уровня, свободных от монополии NSI. Как было сказано в предложении, «пространство имен верхнего уровня обеспечивает перераспределение избытка имен через структуру национальных доменов». А создание новых доменов верхнего уровня «неизбежно приведет к дублированию регистрации, только усугубив существующие проблемы, связанные с полезностью и жизнеспособностью структуры DNS Интернета». В отношении национальных доменов признавался суверенитет государств в определении политики. Регистраторы получали равноправный доступ ко всем доменам верхнего уровня. Они могли устанавливать собственные расценки за свои услуги, конкурируя между собой. Все регистраторы являлись членами ассоциации регистраторов CORE (Council of Registrars, Совет регистраторов), отвечающей за разработку правил, а также за обеспечение необходимой координации, организационную и юридическую поддержку. По замыслу IAHC, CORE следовало зарегистрировать как некоммерческую организацию в швейцарском городе Женеве.

Разногласия в отношении доменных имен решались путем установления 60-дневного периода ожидания, во время которого возможные споры должны были урегулировать специальные комитеты Всемирной организации по интеллектуальной собственности (ВОИС, WIPO), Administrative Challenge Panels.

Официальная церемония подписания Протокола gTLD-MOU состоялась в мае 1997 г. в Женеве. Более 200 организаций подписали протокол, включая Международный союз электросвязи (МСЭ), ВОИС и Всемирный почтовый союз (ВПС).

Тем не менее протокол вызвал критику ряда организаций, которые видели в нем угрозу своим интересам. В первую очередь это была NSI,

которая после окончания договора с NSF могла оказаться на уровне регистратора с полной потерей своей монопольной позиции.

Протокол вызвал и серьезную озабоченность правительства США, которое видело в нем угрозу передачи контроля над глобальной DNS межгосударственным организациям, например МСЭ. В июле 1997 г. телекоммуникационное агентство NTIA Министерства торговли США опубликовало проект приватизации DNS для публичного обсуждения. Этот проект был призван напомнить о де-юре контроле правительства США за корневым уровнем и IANA, а также перехватить инициативу создания новой модели управления Интернетом.

Проект во многом основывался на модели IAHC и содержал эскиз будущей некоммерческой организации, которой правительство США предполагало передать функции IANA и которая была призвана обеспечивать координацию распределение имен, номеров и адресов. Также был предложен план демонополизации NSI, начиная с предоставления доступа к доменам .COM, .ORG, .NET желающим регистраторам на равной основе и заканчивая передачей контроля за корневой зоной и ее мастер-сервером.

Значение этого проекта, как и последующего «Зеленого документа», не всеми было понято. Комитет IAHC продолжал действовать согласно изначальному плану, и в октябре 1997 г. была создана ассоциация CORE. За несколько месяцев до окончания контракта NSF-NSI Джон Постел решил сделать «небольшой шаг в направлении» новой управляющей модели Интернета, когда «редакция и публикация корневой зоны будут осуществляться непосредственно IANA. Этот «тест» не вызвал одобрения у правительства США.

После этого события развивались стремительно, но уже под контролем правительства США. За «Зеленой книгой» последовала «Белая книга», организация различных комитетов и форумов по разработке новой модели управления. Наиболее выдающимися были IANA Transition Advisors Group ITAG (Группа советников по трансформации IANA) под предводительством Постела и анти-IAHC группа, под координацией которой прошла серия так называемых International Forum on the White Paper, IFWP (Международный Форум по «Белой книге»). В рамках форума организовывались встречи в различных точках земного шара и стимулировалось обсуждение вопросов новой модели управления между членами интернет-сообщества, но ключевую роль в процессе становления новой компании, которая получила название ICANN (Internet Corporation

for Assigned Names and Numbers — Корпорация Интернета по распределению адресов и номеров), по-прежнему играла коалиция IAHC во главе с Джоном Постелом и NTIA. В сентябре 1998 г. ICANN была зарегистрирована как некоммерческая корпорация в штате Калифорния, США.

С созданием ICANN страсти не улеглись. ICANN было необходимо завоевать поддержку других важных заинтересованных групп — конечно же, IETF, региональных интернет-регистратур и регистратур национальных доменов. Основную проблему представляла собой NSI, которой по плану предстояло покинуть свою исключительную позицию монополиста общих доменов верхнего уровня. Без поддержки правительства США здесь было не обойтись.

Несколько ключевых соглашений и договоров поддержали молодую организацию ICANN в ее становлении. Протокол о взаимопонимании между Министерством торговли США и ICANN зафиксировал ответственность сторон в соответствии с требованиями «Белой книги». Предполагалось, что ICANN может обрести самостоятельность уже в сентябре 2000 г. В реальности Протокол пережил несколько продлений и ревизий, пока, наконец, не был заменен в 2009 году другим документом — Affirmation of Commitment (Подтверждение обязательств). Он формально провозглашал ICANN независимым от правительства США.

Демонополизацию NSI осуществляли два договора. Первый — бывший договор NSI с NSF, теперь перешедший в руки NTIA, по которому NSI, а впоследствии Verisign, купившая NSI, выполняли функции технического обслуживания корневой зоны — внесение изменений и предоставление зоны корневым серверам, — а также функцию регистратуры доменов .COM, .NET и .ORG. Этот договор сегодня содержит более 30 поправок, отражающих постепенно менявшуюся роль Verisign от монополиста до аккредитованного регистратора и регистратуры доменов .COM и .NET.

Второй договор был между ICANN и NSI — это Договор Регистратуры, по которому NSI сохраняла права обслуживания регистратур .COM, .ORG и .NET, но при основном условии: регистрация доменов второго уровня будет приниматься только от регистраторов аккредитованных ICANN. NSI подписала договор аккредитации, став также и регистратором на уровне этих доменов.

В начале 2000 г. между NTIA и ICANN был заключен договор на передачу функций IANA новой организации — ICANN. Договор

постфактум легитимировал поглощение IANA, а также восстанавливал контроль правительства США за этими ключевыми функциями. Он перезаключался четыре раза и пережил значительное число различных поправок.

Кратко остановимся на самой IANA. Как набор функций IANA существовала с начала 70-х гг. прошлого столетия в рамках проекта ARPANET, предтечи Интернета. Физически этот акроним был тождественен Джону Постелу — он придумал имя и выполнял все функции. В то время IANA являлась каталогом уникальных идентификаторов протоколов. За время своего существования IANA превратилась в центральную регистратуру различных параметров Интернета в трех областях:

1. Протоколы Интернета: здесь IANA отвечает за присвоение различных параметров (операционных кодов, номеров портов и протоколов, идентификаторов объектов), которые используются разнообразными протоколами.

2. Система доменных имен DNS: здесь IANA отвечает за содержимое корневой зоны и обслуживание запросов на ее изменение.

3. Адресное пространство IP: здесь IANA обслуживает глобальный пул, часть которого распределяется между региональными интернет-регистратурами (РИР), часть предназначается для системы мультикаст, а часть зарезервирована IETF для будущего использования.

Таким образом, в независимой децентрализованной культуре Интернета IANA отвечает за три централизованные, иерархические и чрезвычайно важные базы данных и связанные с ними услуги.

Чтобы помочь ICANN встать на ноги, но в то же время уберечь ее от неосторожных шагов, правительство США сформировало сеть договоров и протоколов о взаимопонимании, тем самым все больше формализуя свою уникальную роль в централизованных функциях Интернета — координации и поддержке корневой зоны DNS, распределении адресных и номерных ресурсов, регистрации уникальных параметров протоколов. Это, безусловно, вызывало озабоченность правительств других государств, которые также хотели видеть себя полноправными участниками принятия решений, затрагивающих глобальный Интернет.

Полномочия Government Advisory Committee (GAC, Правительственный консультативный комитет), который существовал в рамках структуры ICANN, были расширены в 2002 г., но роль его по-прежнему была консультативной. Для влияния на разработку и

утверждение политик и правил представители правительств должны были участвовать в общественном процессе на равных. Недовольство росло еще и потому, что ICANN не являлась не только межгосударственной, но даже и «международной» организацией.

01 октября 2016 года IANA была зарегистрирована под названием «Организация по открытым техническим идентификаторам» (PTI) и уже выполняет функции IANA от имени ICANN. В предложении Координационной группы по передаче координирующей роли в исполнении функций IANA (ICG) сообщество отрасли доменных имен указало на необходимость создания нового юридического лица — дочерней компании ICANN — для выполнения функции IANA по присвоению имен. В целях обеспечения согласованности при выполнении функций IANA и общей организации деятельности, предложение ICG также содержало требование о том, чтобы PTI выполняла функции IANA, связанные с ресурсами нумерации и параметрами протоколов, на условиях субподряда. ICANN разрабатывала учредительный договор PTI в сотрудничестве со Сквозной рабочей группой сообщества по функциям, связанным с именами (CWG-координирующая роль), и ее внешним юрисконсультом. В учредительном договоре PTI указано назначение PTI, роль ICANN как единственного участника, а также другие юридические характеристики, связанные с корпоративным и некоммерческим статусом PTI.

Учредительный договор PTI был опубликован для 30-дневного периода общественного обсуждения, который начался 01 июля 2016 года. На совещании 09 августа 2016 года Правление ICANN утвердило формирование PTI и поручило генеральному директору ICANN перейти к регистрации PTI в качестве юридического лица. 10 августа 2016 года в канцелярию штата Калифорния был подан и принят учредительный договор PTI под наименованием «Организация по открытым техническим идентификаторам» (PTI). 28 сентября 2016 года Правление PTI утвердило Учредительный договор.

Чтобы обеспечить наличие надлежащих требований к управлению PTI, ICANN разрабатывала устав PTI в сотрудничестве с группой CWG-координирующая роль и ее внешним юрисконсультом. Устав PTI определяет требования к Правлению PTI и годовому бюджету PTI и одновременно отвечает требованиям сообщества отрасли доменных имен, сформулированным в предложении ICG, и требованиям нового Устава ICANN.

Устав PTI был опубликован для 30-дневного периода общественного обсуждения 12 июля 2016 года. Прежде чем закрыть период общественного обсуждения, ICANN и юрисконсульт группы CWG-координирующая роль согласовали надлежащую формулировку для решения оставшихся проблем, вызвавших озабоченность у группы CWG-координирующая роль. Отчет персонала и окончательная версия устава были опубликованы 18 августа 2016 года. 15 сентября 2016 года Правление ICANN утвердило Устав PTI. 28 сентября Правление PTI утвердило Устав.

Хотя предложение ICG не содержало такого требования, ICANN сделала PTI организацией с четкими методами управления. В документе Национального управления по телекоммуникациям и информации США (NTIA) с результатами оценки и рекомендациями Комитета организаций-спонсоров комиссии Тредуэя (COSO) содержится несколько рекомендаций, подчеркивающих необходимость разработки базового набора документов, регулирующих указания, позиции и поведение участников PTI и поощряющих соблюдение важных этических норм.

Чтобы выполнить и поставленную ICANN цель ввести для PTI высокие стандарты управления, и сформулированные NTIA в ходе оценки COSO рекомендации, ICANN составила проекты следующих документов для PTI: политика в области предотвращения конфликта интересов, кодекс поведения Правления и стандарты ожидаемого поведения. В основу каждого из указанных документов лег текст аналогичного документа, используемого в ICANN.

Все три документа были опубликованы для 30-дневного периода общественного обсуждения с 8 июля по 7 августа 2016 года. После устранения замечаний, поступивших в процессе общественного обсуждения, эти документы были опубликованы 12 августа 2016 года в окончательной редакции. Эти документы были утверждены Правлением PTI 29 сентября 2016 года.

Проект контракта на исполнение функций IANA был создан на основе составленного юрисконсультом группы CWG-координирующая роль проекта регламента оказания услуг, включенного в состав предложения ICG. Проект данного контракта 15 июля 2016 года был передан группе CWG-координирующая роль и ее внешнему юрисконсульту на рассмотрение и обсуждение, результатом которых стало обновление документа для решения вызвавших озабоченность проблем.

10 августа 2016 года пересмотренный проект контракта на исполнение функций IANA был опубликован для проведения 30-дневного периода общественного обсуждения. После устранения замечаний, полученных в период общественного обсуждения, ICANN опубликовала 15 сентября 2016 года окончательный текст контракта на исполнение функций IANA. В тот же день ICANN утвердила контракт. 28 сентября этот контракт утвердило Правление PTI. 20 сентября 2016 года ICANN и PTI подписали контракт. Этот контракт, который уже вступил в силу, позволяет PTI выполнять функцию IANA по присвоению имен от имени ICANN.

Сообщество отрасли доменных имен включило в состав предложения ICG требование ввести новую совокупность ожиданий в отношении уровня обслуживания (SLA), которая заменит установленные в договоре на исполнение функций IANA, срок действия которого истек, стандарты качества работы IANA при исполнении функции присвоения имен. Согласно новым SLA, Организация по открытым техническим идентификаторам будет обязана измерять, регистрировать и сообщать дополнительные подробные данные о сроках выполнения транзакций при обработке запросов на внесение изменений в корневую зону.

С целью введения SLA в действие ICANN внесла изменения в систему управления корневой зоной (RZMS), чтобы собирать необходимые для новых SLA данные. После внесения изменений сбор данных осуществлялся примерно три с половиной месяца, чтобы установить для каждого SLA предлагаемые пороговые значения.

ICANN провела обсуждение с группой CWG-координирующая роль, чтобы проанализировать и скорректировать предложенные пороги. ICANN и CWG-координирующая роль согласовали пороговые значения для всех SLA, которые были включены в состав окончательного контракта на исполнение функций IANA, утвержденного ICANN и Правлением PTI.

Хотя это и не является требованием сообщества отрасли доменных имен, в предложение ICG указано на необходимость заключить субподрядные соглашения с PTI на исполнение функций IANA, относящихся к параметрам протоколов и ресурсам нумерации интернета, в целях согласованного исполнения функций IANA и общей организации работы.

ICANN составила проекты субподрядных соглашений на исполнение функций, относящихся к параметрам протоколов и ресурсам нумерации интернета, и передала эти проекты соответствующим сообществам для

анализа и комментирования. Оба документа утверждены Правлением PTI 28 сентября 2016 года, подписаны ICANN и PTI и вступили в силу.

Чтобы обеспечить наличие у PTI финансов и ресурсов для исполнения функций IANA, ICANN подготовила проект соглашения об оказании услуг, в котором изложены схемы оказания прямых и совместных услуг, которые ICANN будет предлагать PTI для обеспечения функционирования этой организации и исполнения функций IANA.

Проект соглашения об оказании услуг был представлен на рассмотрение трех оперативных сообществ 12 августа 2016 года и приведен к окончательному виду 15 сентября 2016 года. В тот же день ICANN утвердила этот документ, который также был утвержден Правлением PTI 28 сентября 2016 года. 30 сентября 2016 года ICANN и PTI подписали это соглашение, и оно вступило в силу.

Как следует из изложенного выше, управление ресурсами сети Интернет продолжает оставаться привилегией группы частных компаний, основное влияние на деятельность которых оказывает администрация США. Изменить ситуацию, действуя внутри этой сложившейся и самоуправляемой структуры, не представляется возможным ввиду огромных финансовых и политических привилегий, которые она приносит своим участникам. Изменить её возможно только действуя извне путем наращивания международного давления и активности стран, не согласных со сложившейся ситуацией в управлении Интернетом.

Такое давление и активность выражаются, в основном, через деятельность Международного союза электросвязи (МСЭ). Перечислим основные Резолюции, принятые членами МСЭ в рамках Полномочных конференций, Всемирных конференций по развитию электросвязи и Всемирных конференций по стандартизации электросвязи, относящиеся к управлению сетью Интернет и обеспечению безопасности и устойчивости ее функционирования.

Резолюции полномочных конференций МСЭ

- Рез.101 Сети, основанные на интернет-протоколе
- Рез.102 Роль МСЭ в вопросах международной государственной политики, касающихся Интернета и управления ресурсами Интернета, включая наименования доменов и адреса
- Рез.133 Роль администраций государств-членов в управлении интернационализированными (многоязычными) наименованиями доменов

- Рез 180 Содействие переходу с IPv4 на Ipv6

Резолюции всемирных конференций МСЭ по развитию электросвязи

- Рез. 20 Недискриминационный доступ к современным средствам электросвязи/информационно-коммуникационным технологиям, услугам и соответствующим приложениям
- Рез. 30 Роль сектора развития электросвязи МСЭ в осуществлении решений Всемирной встречи на высшем уровне по вопросам информационного общества с учетом повестки дня в области устойчивого развития на период до 2030 года
- Рез. 63 Распределение IP-адресов и содействие переходу к развертыванию Ipv6 в развивающихся странах

Резолюции всемирных конференций МСЭ по стандартизации электросвязи

- Рез. 20 Процедуры распределения и управления международными телекоммуникационными ресурсами нумерации, наименования, адресации и идентификации
- Рез. 47 Доменные имена верхнего уровня с кодами стран
- Рез. 48 Интернационализированные (многоязычные) доменные имена
- Рез. 60 Учет эволюции систем идентификации/нумерации и их конвергенции с системами и сетями, работающими на основе протокола IP
- Рез. 64 Распределение интернет-адресов и содействие переходу на Ipv6 и его развертыванию
- Рез. 69 Недискриминационный доступ к использованию интернет-ресурсов
- Рез. 75 Реализация решений Всемирной встречи на высшем уровне по вопросам информационного общества с учетом новой повестки дня в области устойчивого развития на период до 2030 года.

Для повышения эффективности международного правового регулирования сети Интернет необходимо конкретизировать головную роль МСЭ за развитие ИКТ-инфраструктуры и обеспечение доверия и безопасности при использовании ИКТ в отношении сети Интернет (пункты С2 и С5 Плана действий Всемирной встречи на высшем уровне по вопросам развития информационного общества). Учитывая, что Интернет

является неотъемлемой составной частью международной ИКТ-инфраструктуры (а национальные сегменты сети Интернет – неотъемлемой составной частью национальных ИКТ-инфраструктур) необходимо формализовать действия пунктов С2 и С5 в отношении сети Интернет.

Рассмотрим порядок регулирования обеспечения информационной безопасности сети Интернет на международном уровне должно осуществляться по трем основным направлениям. Обеспечения информационной безопасности.

1. Международное нормативное правовое регулирование.

Вопросы международного права, относящиеся к сети Интернет, регулируют:

– обеспечение недискриминационного доступа к сетевым и информационным ресурсам;

– запрет на нарушение функционирования или уничтожение сетевых и информационных ресурсов;

– запрет на вмешательство в вопросы, регулируемые на уровне национальных юрисдикций;

– обеспечение функционирования доменов верхнего уровня на национальных языках.

Для повышения эффективности международного правового регулирования сети Интернет необходимо конкретизировать головную роль МСЭ (определенную поручением ООН) за развитие ИКТ-инфраструктуры и обеспечение доверия и безопасности при использовании ИКТ в отношении сети Интернет (пункты С2 и С5 Плана действий Форума ООН на высшем уровне по вопросам развития информационного общества).

2. Международное нормативное техническое регулирование (стандартизация).

Вопросы международного нормативного технического регулирования сети Интернет относятся к технологическим, организационным и процедурным аспектам обеспечения безопасного и устойчивого функционирования сети. Эти вопросы, в частности, включают:

– архитектуру обеспечения информационной безопасности;

– управление информационной безопасностью;

– безопасность Интернета вещей;

– защиту персональных данных;

– управление идентификацией;

– безопасность облачных вычислений;

– безопасность инфраструктуры открытых ключей;

– криптографическую защиту.

Технические решения по вопросам развития инфраструктуры и обеспечения доверия и безопасности при работе в сети Интернет принимаются в рамках ICANN и в организациях, разрабатывающих технические стандарты (IETF, W3C, МСЭ и другие). Распределение адресного пространства сети Интернет осуществляют уполномоченные региональные организации. В плане совершенствования международного технического регулирования сети Интернет необходимо расширять участие российских экспертов в деятельности указанных организаций и проводить экспертизу обсуждаемых и принимаемых документов.

3. Международная образовательная, просветительская и культурологическая деятельность по вопросам управления Интернетом.

Данная деятельность ведется в рамках IGF, ISOC, академии МСЭ, а также во многих университетах и академиях. В плане совершенствования данного направления необходимо расширять участие представителей гражданского общества в деятельности и образовательных программах указанных организаций. Необходимо также совершенствовать учебные программы российских университетов, обеспечивая подготовку специалистов, готовых и способных разрабатывать и внедрять отечественное оборудование для реализации интернет-технологий и технологий информационной безопасности.

1.4. Нормативное правовое регулирование сети Интернет на национальном уровне

В Российской Федерации правовое регулирование сети Интернет осуществляется посредством системы нормативных правовых актов, среди которых важно особо отметить федеральные законы «О связи», «Об информации, информационных технологиях и о защите информации», «О персональных данных» и «О безопасности критической информационной инфраструктуры Российской Федерации».

Функционирование сети Интернет обеспечивается следующими услугами связи: телематические услуги связи (предоставление доступа к информационно-телекоммуникационной сети Интернет) и услуги связи по передаче данных. Соответственно, положения Федерального закона «О

связи» (далее – Закон о связи) вносят существенный вклад в правовое регулирование сети Интернет.

В Законе о связи определены основные понятия, в том числе такие понятия, как: «оператор связи», «организация связи», «пользователь услугами связи», «сеть связи», «средства связи», «трафик», «управление сетью связи», «услуга связи», «услуга присоединения», «услуга по пропуску трафика», «контентные услуги».

Согласно статье 29 Закона о связи деятельность юридических лиц и индивидуальных предпринимателей по возмездному оказанию услуг связи осуществляется только на основании лицензии на осуществление деятельности в области оказания услуг связи. Перечень наименований услуг связи, вносимых в лицензии, и соответствующие перечни лицензионных условий установлены постановлением Правительством Российской Федерации от 18.02.2005 № 87. Процедура лицензирования в полном объеме урегулирована на уровне Закона о связи (глава 6), что исключило необходимость принятия подзаконных актов по вопросу лицензирования и является основанием для заявления о том, что в отношении лицензирования услуг связи Закон о связи действительно является законом прямого действия.

В соответствии с пунктом 1 статьи 44 Закона о связи на территории Российской Федерации услуги связи оказываются операторами связи пользователям услугами связи на основании договора об оказании услуг связи, заключенного в соответствии с гражданским законодательством и правилами оказания услуг связи. Правительством Российской Федерации утверждены Правила оказания телематических услуг связи (постановление Правительства РФ от 10.09.2007 N 575) и Правила оказания услуг связи по передаче данных (постановление Правительства РФ от 23.01.2006 N 32).

Отношения по присоединению сетей электросвязи и их взаимодействию также подпадают под сферу действия Закона о связи. В частности, в статье 18 Закона о связи установлено следующее:

– операторы связи и владельцы сетей связи специального назначения имеют право на присоединение своих сетей электросвязи к сети связи общего пользования на основании договоров о присоединении сетей электросвязи;

– операторы сети связи общего пользования обязаны оказывать услуги присоединения иным операторам связи в соответствии с правилами присоединения сетей электросвязи и их взаимодействия, утвержденными

Правительством Российской Федерации (постановления Правительства Российской Федерации от 28.03.2005 № 161 и от 13.12.2006 № 760);

– если иное не предусмотрено Законом о связи, цены на услуги присоединения и услуги по пропуску трафика определяются оператором связи самостоятельно, исходя из требований разумности и добросовестности.

Применительно к сетям передачи данных не установлены требования к построению сети связи и порядку пропуска трафика.

Согласно статье 27 Закона о связи федеральный государственный надзор в области связи осуществляется уполномоченными федеральными органами исполнительной власти в соответствии с их компетенцией в порядке, установленном Правительством Российской Федерации. Постановлением Правительства РФ от 05.06.2013 N 476 утверждено Положение о федеральном государственном надзоре в области связи, в соответствии с которым государственный надзор осуществляется Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций и ее территориальными органами с применением риск-ориентированного подхода.

Важно отметить также следующие вопросы, подпадающие под сферу действия Закона о связи:

– тайна связи;

– обязанности операторов связи и ограничение прав пользователей услугами связи при проведении оперативно-розыскных мероприятий, мероприятий по обеспечению безопасности Российской Федерации и осуществлении следственных действий;

– управление сетями связи в чрезвычайных ситуациях и в условиях чрезвычайного положения;

– подтверждение соответствия средств связи и услуг связи.

Применительно к международному сотрудничеству Российской Федерации в области связи предусмотрено, что такое сотрудничество осуществляется на основе соблюдения общепризнанных принципов и норм международного права, а также международных договоров Российской Федерации (статья 69 Закона о связи).

В соответствии со статьей 70 Закона о связи для оказания услуг связи в пределах мировых информационно-телекоммуникационных сетей на территории Российской Федерации является обязательным:

– создание российских сегментов глобальных, региональных спутниковых сетей связи в целях обеспечения взаимодействия с единой

сетью связи Российской Федерации и обеспечение управления российскими сегментами глобальных, региональных спутниковых сетей связи с территории Российской Федерации;

– создание российских операторов связи, отвечающих требованиям, предъявляемым к ним Законом о связи;

– обеспечение экономической, общественной, оборонной, экологической, информационной и иных видов безопасности.

Следующим законодательным актом, имеющим важное значение для нормативно-правового регулирования сети Интернет, является Федеральный закон «Об информации, информационных технологиях и о защите информации» (далее – Закон об информации).

В статье 15 Закона об информации установлено, что на территории Российской Федерации использование информационно-телекоммуникационных сетей осуществляется с соблюдением требований законодательства Российской Федерации в области связи, Закона об информации и иных нормативных правовых актов Российской Федерации. При этом регулирование использования информационно-телекоммуникационных сетей, доступ к которым не ограничен определенным кругом лиц, осуществляется в Российской Федерации с учетом общепринятой международной практики деятельности саморегулируемых организаций в этой области.

В Законе об информации раскрывается ряд основных понятий: «информационные технологии», «информационная система», «информационно-телекоммуникационная сеть», «электронное сообщение», «электронный документ», «оператор информационной системы», «сайт в сети Интернет, «страница сайта в сети Интернет, «доменное имя», «сетевой адрес», «владелец сайта в сети Интернет, «провайдер хостинга», «поисковая система».

Закреплен правовой статус ряда субъектов права: владелец сайта в сети Интернет, оператор информационной системы, провайдер хостинга, организатор распространения информации (в том числе, организатор сервиса мгновенных сообщений), новостной агрегатор, владелец аудиовизуального сервиса. Так, установлено, что владелец сайта в сети Интернет обязан разместить на принадлежащем ему сайте информацию о своих наименовании, месте нахождения и адресе, адресе электронной почты для направления заявления, указанного в статье 15.7 Закона об информации, а также вправе предусмотреть возможность направления

этого заявления посредством заполнения электронной формы на сайте в сети Интернет.

В отношении организатора распространения информации в сети Интернет установлена обязанность по хранению на территории Российской Федерации:

1) информации о фактах приема, передачи, доставки и (или) обработки голосовой информации, письменного текста, изображений, звуков, видео- или иных электронных сообщений пользователей сети Интернет и информацию об этих пользователях в течение одного года с момента окончания осуществления таких действий;

2) текстовых сообщений пользователей сети Интернет, голосовой информации, изображений, звуки, видео-, иных электронных сообщений пользователей сети Интернет до шести месяцев с момента окончания их приема, передачи, доставки и (или) обработки.

В статье 3 Закона об информации установлены принципы правового регулирования отношений в сфере информации, информационных технологий и защиты информации, в частности, следующие принципы:

– свобода поиска, получения, передачи, производства и распространения информации любым законным способом;

– установление ограничений доступа к информации только федеральными законами;

– обеспечение безопасности Российской Федерации при создании информационных систем, их эксплуатации и защите содержащейся в них информации;

– недопустимость установления нормативными правовыми актами каких-либо преимуществ применения одних информационных технологий перед другими, если только обязательность применения определенных информационных технологий для создания и эксплуатации государственных информационных систем не установлена федеральными законами.

Согласно статье 16 Закона об информации государственное регулирование отношений в сфере защиты информации осуществляется путем установления требований о защите информации, а также ответственности за нарушение законодательства Российской Федерации об информации, информационных технологиях и о защите информации. Также предусмотрено, что федеральными законами могут быть установлены ограничения использования определенных средств защиты

информации и осуществления отдельных видов деятельности в области защиты информации.

Основной акцент регулирования сделан в Законе об информации на распространении информации в сети Интернет. В частности, определена форма открытых данных применительно к информации, размещаемой ее обладателями в сети Интернет в формате, допускающем автоматизированную обработку без предварительных изменений человеком в целях повторного ее использования (общедоступная информация).

Предусмотрены меры для предотвращения и пресечения нарушений в сети Интернет. Так, закреплен порядок создания единого реестра всех доменных имен, страниц сайтов в сети Интернет и сетевых адресов, позволяющих идентифицировать сайты в сети Интернет, содержащие информацию, распространение которой в Российской Федерации запрещено. Постановлением Правительства РФ от 26.10.2012 № 1101 утверждены Правила создания, формирования и ведения единой автоматизированной информационной системы «Единый реестр доменных имен, указателей страниц сайтов в информационно-телекоммуникационной сети Интернет и сетевых адресов, позволяющих идентифицировать сайты в информационно-телекоммуникационной сети Интернет, содержащие информацию, распространение которой в Российской Федерации запрещено», Правилами принятия уполномоченными Правительством Российской Федерации федеральными органами исполнительной власти решений в отношении отдельных видов информации и материалов, распространяемых посредством информационно-телекоммуникационной сети Интернет, распространение которых в Российской Федерации запрещено.

Также определяется порядок ограничения доступа к подобным сайтам (статьи 15.1 – 15.3, 15.5 – 15.6.1 Закона об информации), согласно которому оператор связи, оказывающий услуги по предоставлению доступа к информационно-телекоммуникационной сети Интернет, обязан ограничить доступ к незаконно размещенной информации в соответствии со сведениями, полученными таким оператором связи от Роскомнадзора по системе взаимодействия, что исключает возможность прямого обращения к оператору связи различных государственных органов, третьих лиц.

Установлено так называемое «право на забвение», согласно которому оператор поисковой системы, распространяющий в сети Интернет рекламу, которая направлена на привлечение внимания потребителей,

находящихся на территории Российской Федерации, по требованию гражданина (физического лица) обязан прекратить выдачу сведений об указателе страницы сайта в сети Интернет, позволяющих получить доступ к информации о заявителе, распространяемой с нарушением законодательства Российской Федерации, являющейся недостоверной, а также неактуальной, утратившей значение для заявителя в силу последующих событий или действий заявителя, за исключением информации о событиях, содержащих признаки уголовно наказуемых деяний, сроки привлечения к уголовной ответственности по которым не истекли, и информации о совершении гражданином преступления, по которому не снята или не погашена судимость.

Представляется необходимым отметить, что в Законе об информации установлены гарантии для лиц, выполняющих законные требования об ограничении доступа к информации, распространяемой посредством сети Интернет. Так, согласно статье 17 Закона об информации провайдер хостинга, оператор связи и владелец сайта в сети Интернет не несут ответственность перед правообладателем и перед пользователем за ограничение доступа к информации и (или) ограничение ее распространения в соответствии с требованиями Закона об информации. Кроме того, предусмотрено, что в случае, если распространение определенной информации ограничивается или запрещается федеральными законами, гражданско-правовую ответственность за распространение такой информации не несет лицо, оказывающее услуги:

1) либо по передаче информации, предоставленной другим лицом, при условии ее передачи без изменений и исправлений;

2) либо по хранению информации и обеспечению доступа к ней при условии, что это лицо не могло знать о незаконности распространения информации.

Закон об информации регулирует и порядок применения внесудебных мер по прекращению нарушения авторских и (или) смежных прав в информационно-телекоммуникационных сетях, в том числе в сети Интернет, принимаемые владельцем сайта по заявлению правообладателя (статья 15.7 Закона об информации).

Также определены меры, направленные на противодействие использованию на территории Российской Федерации информационно-телекоммуникационных сетей и информационных ресурсов, посредством которых обеспечивается доступ к информационным ресурсам и информационно-телекоммуникационным сетям, доступ к которым

ограничен на территории Российской Федерации (статья 15.8 Закона об информации).

Федеральным законом от 31.12.2017 № 482-ФЗ «О внесении изменений в отдельные законодательные акты Российской Федерации» в Закон об информации включена статья 14.1 «Применение информационных технологий в целях идентификации граждан Российской Федерации», согласно которой государственные органы, банки и иные организации в случаях, определенных федеральными законами, после проведения идентификации при личном присутствии гражданина Российской Федерации с его согласия на безвозмездной основе размещают в электронной форме:

1) сведения, необходимые для регистрации гражданина Российской Федерации в единой системе идентификации и аутентификации, и иные сведения, если такие сведения предусмотрены федеральными законами, - в единой системе идентификации и аутентификации;

2) биометрические персональные данные гражданина Российской Федерации - в единой информационной системе персональных данных, обеспечивающей обработку, включая сбор и хранение биометрических персональных данных, их проверку и передачу информации о степени их соответствия предоставленным биометрическим персональным данным гражданина Российской Федерации.

В систему законодательных актов по регулированию сети Интернет входит и Федеральный закон «О безопасности критической информационной инфраструктуры Российской Федерации» (далее – Закон о КИИ), которым к объектам критической информационной инфраструктуры отнесены информационные системы, информационно-телекоммуникационные сети и автоматизированные системы управления.

Согласно статье 2 Закона о КИИ субъектами критической информационной инфраструктуры являются государственные органы, государственные учреждения, российские юридические лица и (или) индивидуальные предприниматели, которым на праве собственности, аренды или на ином законном основании принадлежат информационные системы, информационно-телекоммуникационные сети, автоматизированные системы управления, функционирующие в сфере здравоохранения, науки, транспорта, связи, энергетики, банковской сфере и иных сферах финансового рынка, топливно-энергетического комплекса, в области атомной энергии, оборонной, ракетно-космической, горнодобывающей, металлургической и химической промышленности,

российские юридические лица и (или) индивидуальные предприниматели, которые обеспечивают взаимодействие указанных систем или сетей.

В Законе о КИИ определены требования к государственной системе обнаружения, предупреждения и ликвидации последствий компьютерных атак на информационные ресурсы Российской Федерации; полномочия Президента Российской Федерации и органов государственной власти Российской Федерации в области обеспечения безопасности критической информационной инфраструктуры Российской Федерации; предусмотрено категорирование объектов КИИ и ведение реестра значимых объектов КИИ; установлены права и обязанности субъектов КИИ; определены требования к системе безопасности и по обеспечению безопасности значимого объекта КИИ. Также предусмотрены оценка безопасности КИИ РФ и государственный контроль в области обеспечения безопасности значимых объектов КИИ.

Уголовный кодекс Российской Федерации дополнен статьей 274.1. «Неправомерное воздействие на критическую информационную инфраструктуру Российской Федерации».

Также стоит отметить особо Федеральный закон «О персональных данных» (далее – Закон о ПД), требования которого обязательны и при обработке персональных данных в сети Интернет.

Так, согласно статье 18 Закона о ПД при сборе персональных данных, в том числе посредством информационно-телекоммуникационной сети Интернет, оператор обязан обеспечить запись, систематизацию, накопление, хранение, уточнение (обновление, изменение), извлечение персональных данных граждан Российской Федерации с использованием баз данных, находящихся на территории Российской Федерации, за исключением случаев, указанных в пунктах 2, 3, 4, 8 части 1 статьи 6 Закона о ПД.

В целях удостоверения личности и полномочий контрагента при совершении сделок в Интернете применяются положения Федерального закона «Об электронной подписи». Указанный Закон содержит немаловажное для виртуальных сделок положение, в соответствии с которым информация в электронной форме, подписанная квалифицированной электронной подписью, признается электронным документом, равнозначным документу на бумажном носителе, подписанному собственноручной подписью.

Положения, касающиеся нормативно-правового регулирования оказания услуг в Интернет-среде, содержатся и в других федеральных

законах, в том числе в Законе РФ «О средствах массовой информации», Федеральном законе «О рекламе», Федеральном законе «О защите конкуренции», Федеральном законе «Об основах охраны здоровья граждан в Российской Федерации» (в части применения телемедицинских технологий), Федеральном законе «О защите детей от информации, причиняющей вред их здоровью и развитию».

Для обеспечения усиления роли государства в развитии национального сегмента сети Интернет и обеспечении доверия и безопасности при его использовании целесообразно рассматривать российский сегмент сети Интернет как неотъемлемую составную часть национальной ИКТ-инфраструктуры. Такой подход обеспечивает применение лицензионных требований к деятельности операторов связи и интернет-сервис-провайдеров, включая безопасность присоединения, обеспечение СОРМ, приоритезацию доступа (включая ограничение) к сетевым ресурсам при чрезвычайных ситуациях (ЧС) и в условиях чрезвычайного положения (ЧП), управление качеством связи и т.д.

Следует также отметить, что изменение законодательных норм должно предусматривать целый ряд важнейших системных задач обеспечения государственного регулирования, относящихся как к российскому сегменту сети Интернет, так и к ССОП в целом:

- запрет на передачу проблем безопасности за пределы национальной юрисдикции;
- обеспечение функционирования систем управления сетями связи в пределах национальной юрисдикции;
- ограничение на проведение мониторинга и обслуживания сетей связи организациями, находящимися за пределами национальной юрисдикции;
- ограничение на использование находящихся за пределами национальной юрисдикции баз знаний угроз информационной безопасности;
- выполнение базовых требований безопасности при присоединении сетей связи;
- обеспечение приоритезации (включая ограничение) доступа к сетевым информационным ресурсам при ЧС и в условиях ЧП, а также в других установленных законодательством случаях;
- управление безопасностью в контексте управления качеством связи.

1.5. Оценка внутренней связности российского сегмента сети Интернет

В Российской Федерации имеется 5 магистральных операторов с общефедеральным покрытием, которые, в основном, обеспечивают транзит данных по территории РФ:

- Вымпелком;
- Мегафон;
- МТС;
- Ростелеком;
- Транстелеком.

К этим операторам (как правило, одновременно к нескольким из соображений надёжности) подключены большинство операторов, оказывающих услуги доступа в Интернет конечным потребителям. Все российские сервисы, предоставляющие услуги через сеть Интернет и контент ресурсы, также имеют прямые стыки с упомянутыми большими операторами, а также многими операторами доступа.

Большие магистральные операторы имеют множество распределенных по стране межсетевых стыков, за счет которых трафик российских пользователей замыкается внутри этой иерархии по кратчайшим путям. Множественность стыков обеспечивает многократное резервирование маршрутов прохождения трафика через разнесенные магистральные сети и множество стыков верхнего уровня.

Прочие операторы связи РФ, а также поставщики услуг в сети Интернет и контент-провайдеры (организаторы распространения информации – далее ОРИ) с целью минимизации пути пробега трафика и расходов на его транспортировку устанавливают между своими сетями прямые (пиринговые) стыки или участвуют в точках обмена трафиком различного уровня.

Обобщённая схема взаимодействия сетей российских операторов связи, поставщиков услуг в сети Интернет и сетей ОРИ между собой приведена на рисунке 1.5.1.

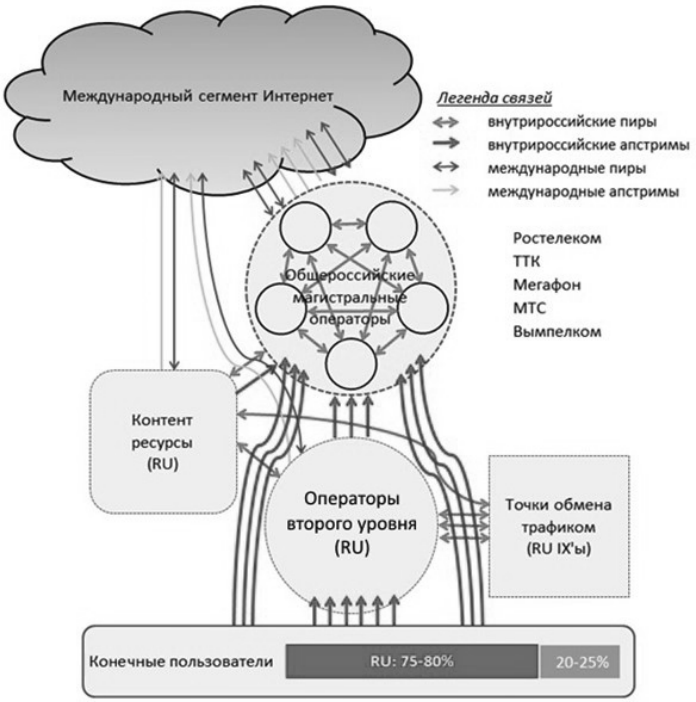

Рисунок 1.5.1 – Структура взаимодействия российских сетей, обеспечивающих доступ российских конечных пользователей в Интернет

Большое количество конкурирующих операторов обеспечивают на межоператорском уровне несколько маршрутов для передачи трафика от источников контента и сервисов к конечным пользователям по разным сетевым инфраструктурам. Это, во-первых, обеспечивает высокий уровень резервирования маршрутов для пропуска трафика различного вида, а, во-вторых, вынуждает операторов изыскивать кратчайшие пути (как в смысле BGP-маршрутизации, так и в смысле задействования инфраструктуры своих сетей), чтоб выиграть борьбу за клиентов за счёт, в первую очередь, качества услуг.

Основной трафик, передающийся по сетям передачи данных, – контент, запрошенный пользователями. Совокупность источников трафика по запросам пользователей называются предпочтениями пользователей по трафику. Предпочтения пользователей по трафику – спектр распределения межоператорского трафика по Автономным Системам (далее – АС).

Предпочтения пользователей по трафику – устойчивая, медленно меняющаяся во времени и по территории Российской Федерации характеристика, которой операторы не управляют. Предпочтения пользователей формируют структуру трафика. Неискаженные предпочтения пользователей могут быть замерены по достаточно большой группе конечных пользователей. На большинстве межоператорских стыков картина смещена из-за множества связей и балансировки трафика.

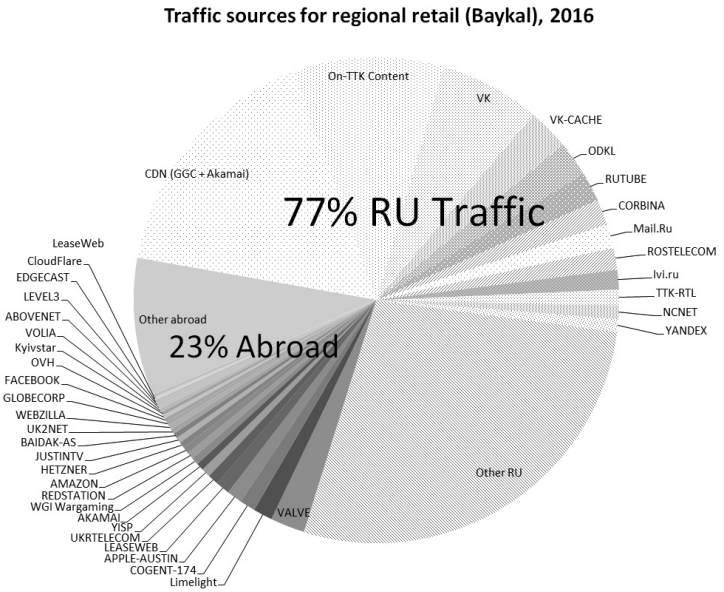

Рисунок 1.5.2 – Распределение предпочтений конечных пользователей для одного из сибирских регионов по измерениям в 2016 году

Представленная на рисунке 1.5.2 структура трафика является типичной – зарубежные ресурсы в трафике пользователей составляют, примерно, 20-25 %. Остальной трафик поступает с контент-ресурсов или кэш-серверов, подключенных непосредственно к сети оператора связи, а также с ОРИ ВКонтакте, Одноклассники, Майл.ру, Яндекс или с сетей других российских операторов через прямые пиринговые или клиентские стыки.

Обеспечение получения трафика клиентом по кратчайшему пути, то есть исключение петель через других операторов (в первую очередь, зарубежных) – приоритетная задача политики маршрутизации для любого

оператора сети Интернет. Это позволяет оптимизировать связность (в терминах протокола BGP) и минимизировать себестоимость пропуска трафика. Без подобной оптимизации экономическая целесообразность ведения магистрального бизнеса исчезает.

Зарубежные стыки, обеспечивающие международную связность российского сегмента сети Интернет, организованы всеми магистральными операторами и другими крупными интернет-игроками для получения трафика популярных международных сервис- и контент-ресурсов (Google, Microsoft, Apple, игровые площадки и т.д.) и доставки российского контента до иностранных пользователей, а также для обеспечения запросов бизнеса, который имеет потребность в коммуникациях с партнёрами вне территории России.

Зарубежная связность приводит к дополнительным затратам для любого из игроков российского интернет рынка. Затраты определяются стоимостью каналов связи до зарубежных площадок обмена трафиком и до мировых Tier1 провайдеров. Емкость каналов и количество портов оборудования, приобретаемых за рубежом, непосредственно связаны с объемом трафика в соответствии с предпочтениями пользователей по трафику.

Количественная характеристика межоператорского обмена сетей передачи данных верхнего уровня (Ростелеком, ТТК, МегаФон, МТС, ВымпелКом) между собой и с крупными ОРИ России приведена в таблице 1.5.1.

Таблица 1.5.1

Оператор	Количество стыков (по bundle), шт.	Количество географически разнесенных точек (населённых пунктов) межсетевого взаимодействия, шт.	Общая портовая емкость пиринговых стыков, Гбит/с
ТТК	62	21	1620
Ростелеком	108	18	4100
МегаФон	65	13	2025
ВымпелКом	118	17	2800
МТС	61	18	2200

Приведённые в таблице 1.5.1 данные демонстрируют высокий уровень связности российских магистральных сетей передачи данных для передачи интернет-трафика.

Операторы второго уровня принимают участие в точках обмена трафиком и устанавливают взаимодействие между своими сетями. Например, полоса обмена для MSK-IX по семейству пиринговых точек в Российской Федерации составляет около 2,5 Тбит/с. Ориентировочно, еще столько же трафика замыкается через другие российские точки обмена трафиком. В сумме (с исключением из данных таблицы 1 взаимообмена между пятеркой крупнейших операторов – чтобы исключить двойной счет), получаем оценку в 14 Тбит/с полосы внутрироссийского обмена трафиком. К этому следует добавить межоператорские полосы портов, которые покупают операторы второго уровня (рисунок 1.5.1).

Суммарный трафик в сторону клиентов мобильного и фиксированного широкополосного доступа по совокупности всех операторов Российской Федерации оценивается в ~18 Тбит/с (28 млн пользователей х 0,5 Мбит/с потребления в среднем на каждого + 4 Тбит/с в сумме от пользователей мобильного Интернета). Доля зарубежного трафика в этом объеме составляет ~4,5 Тбит/с. На долю внутрироссийского трафика приходится ~14 Тбит/с внутрироссийских пиринговых связей по магистральным операторам и операторам второго уровня.

Таким образом, исходя из описанной структуры российских иерархически взаимоувязанных сетей передачи данных с большим числом игроков разного уровня и наличием распределенных по всей России межоператорских пиринговых стыков, отсутствуют предпосылки для обмена внутрироссийским трафиком в сколь-нибудь заметном объёме через зарубежные сети.

1.6. Принципы обеспечения интеграции российского сегмента сети Интернет в национальную и международную ИКТ-инфраструктуру. Оценка экономического и политического значения такой интеграции

Сеть Интернет является инструментом (семейством протоколов), обеспечивающим интеграцию для информационного взаимодействия различных по структуре и принципам функционирования сетей, устройств, информационных ресурсов и приложений (например, www,

электронная почта, электронные услуги и т.д.). Являясь средством обеспечения интеграции и взаимодействия, сеть Интернет может рассматриваться как неотъемлемая составная часть международной ИКТ-инфраструктуры, в то время, как национальные сегменты сети Интернет – как неотъемлемые составные части национальных ИКТ-инфраструктур. Основными механизмами обеспечения интеграции и взаимодействия в сети Интернет является поддержка функционирования систем наименований доменов и адресации.

Маршрутизация трафика российских пользователей через зарубежные сети имела место на заре российского Интернета. Для этого периода (1990-е годы) характерны следующие особенности:

объем российского контента минимален. Пользователи обращаются прежде всего к зарубежным Интернет-ресурсам;

в стране отсутствует развитая IP-инфраструктура, обеспечивающая эффективное взаимодействие российских сетей, предоставляющих доступ в Интернет;

количество пользователей Интернета в стране не велико и объем потребляемого ими трафика незначителен (доминирующая технология доступа – dial-up).

В такой ситуации, действительно, для предоставления пользователям доступа в Интернет провайдеру наиболее целесообразно организовать канал сравнительно небольшой емкости до какой-либо зарубежной точки обмена трафиком, через которую он получит всю необходимую связность и доступ к наиболее популярным Интернет-ресурсам. Небольшой в общем объеме трафика внутрироссийский трафик при такой конфигурации сетей проходит через те же зарубежные точки обмена трафиком, через которые выходят в Сеть российские маломощные IP-сети.

За 20-25 лет, прошедших с описанного выше периода, ситуация в российском сегменте Интернета изменилась радикально. Драматически выросло количество пользователей. В соответствии с численностью населения страны, Россия стала одним из крупнейших Интернет рынков Европы. При этом российская аудитория, в массе своей, не англоязычная и потребляет, прежде всего, русскоязычный контент, генерируемый и располагающийся внутри страны.

Изменились технологии доступа. Dial-up ушел в прошлое, и абоненты потребляют широкополосный доступ по фиксированным и мобильным сетям на мегабитных скоростях. В стране созданы разветвленные операторские IP-сети, конкурирующие между собой и покрывающие

практически всю населенную территорию страны, кроме малочисленных поселений, в которых проживает незначительный процент населения.

Произошли изменения и на глобальном интернет-рынке. С учетом того, что за последние десять лет на рынке имело место существенное снижение цен на IP-транзит, при сохранении достаточно высоких затрат на пиринг (рисунок 1.6.1), вхождение в клуб глобальных операторов уровня Tier1 и сохранение места в нем на текущий момент требует достаточно высоких операционных затрат. Поэтому для большинства региональных провайдеров, особенно «замкнутых региональных интернет-экосистем», вхождение в клуб глобальных Tier 1 сегодня представляется экономически нецелесообразным.

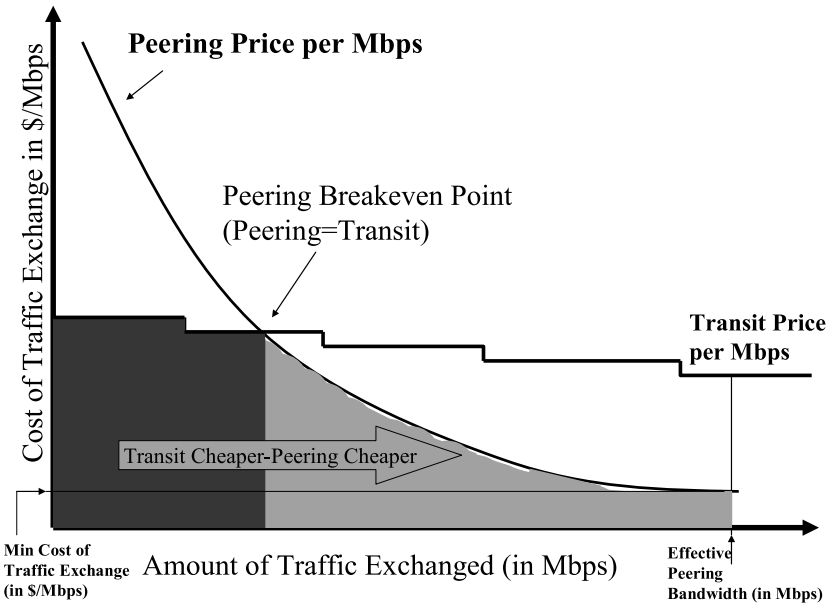

Рисунок 1.6.1 – Затраты на пиринг

Однако, с технической точки зрения пиринг имеет свои преимущества: более низкую задержку, лучший контроль маршрутизации и, следовательно, может привести к снижению потери пакетов. Для операторов пиринг имеет и финансовые выгоды, поскольку при больших емкостях стыков пиринг ощутимо дешевле IP-транзита. При этом клиенты

пользуются большей пропускной способностью, обеспечиваемой пирингом, и, следовательно, платят больше денег. В этой связи во многих регионах образовались собственные региональные интернет-экосистемы со своими региональными провайдерами уровня Tier 1 (локального). Под региональным провайдером уровня 1 понимается провайдер, который имеет доступ ко всей таблице маршрутизации в интернет-регионе исключительно через пиринговые соединения.

В случае, если региональная интернет-экосистема имеет развитых региональных Tier 1, а также существенный объем контента, генерируемого региональными ОРИ, трафик преимущественно замыкается в пределах региона, при этом потребление внешних по отношению к региону ресурсов, включая ресурсы глобальных Tier1 провайдеров, не превышает 20 – 25 %. Такие региональные интернет-экосистемы являются замкнутыми.

Региональные российские операторы уровня Tier 1 начали формироваться в начале 2000-х годов с развитием цифровой инфраструктуры магистральных каналов связи и появлением фиксированных сетей широкополосного доступа, на тот момент, преимущественно, на базе технологии ADSL.

В 2005 – 2010 годах формирование региональных российских Tier 1 было завершено. После этого многие зарубежные операторы, пытавшиеся играть (спекулировать) на продаже «петель российского трафика», в том числе Cable&Wireless, Cogent и др., вынуждены были уйти с российского рынка услуг IP-транзита и, как следствие этого, из российской интернет-экосистемы.

Точки обмена трафиком в России преимущественно используются российскими же операторами связи уровня Tier 2 и российскими ОРИ для сокращения длины маршрута и улучшения качества услуг, предоставляемых конечным клиентам. Объем трафика в точках обмена трафиком не превышает 20-25% от общего объема трафика российской интернет-экосистемы. Фактически, точки обмена трафиком в РФ сегодня находятся на уровне Tier 2 по отношению к региональным Tier 1.

Архитектура российской интернет-экосистемы и ее место в глобальной интернет-экосистеме представлены на рисунке 1.6.2.

Попробуем сопоставить затраты оператора связи (провайдера доступа в Интернет) при обеспечении связности по внутрироссийскому трафику внутри страны по сравнению с организацией «петли» для внутрироссийского трафика через зарубежные точки обмена трафиком.

На текущий момент стоимость полосы объемом 1 Мбит/с для услуги IP-транзит при общем объеме закупки от 10 Гбит/с в городах Москве и Санкт-Петербурге не превышает 25 – 30 рублей в месяц. Таким образом, стоимость полосы 10 Гбит/с в городах Москве и Санкт-Петербурге не превышает 250 000 – 300 000 рублей в месяц, а с учетом подключения через точки обмена трафиком внутри страны для оператора составит 320 000 – 385 000 руб/мес.

Для закупки интернет-трафика у зарубежных операторов с целью организации «зарубежной петли российского трафика» потребуются следующие затраты: оплата услуг связи по предоставлению в пользование международного канала связи на участке Москва – Франкфурт–на–Майне (Хельсинки, Стокгольм), Санкт-Петербург – Франкфурт–на–Майне (Хельсинки, Стокгольм), оплата за кроссировку в зарубежном телехаусе, оплата услуги IP-транзита у международного оператора.

В таблице 1.6.1 приведен расчет затрат на организацию «зарубежной петли трафика» объемом 10 Гбит/с.

Таблица 1.6.1

Средняя стоимость услуги связи по предоставлению в пользование международного канала связи 10 Гбит/с, евро в месяц	Средняя стоимость кроссировки в телехаусе, евро в месяц	Средняя стоимость услуги IP-транзит, евро в месяц
2100	150	5500
Итого в месяц, евро	7750	
Итого в месяц, рубли	542 500 (при курсе 70 рублей за евро)	

Для оператора, работающего вне двух крупнейших городов страны, затраты могут быть несколько выше по обоим сравниваемым вариантам, что не меняет картину в целом.

С учетом произведенного расчета затраты на организацию «зарубежной петли российского трафика» в среднем в 2 раза превышают затраты на закупку услуг IP-транзита через российских операторов.

Таким образом, систематическое использование пропуска внутрироссийского трафика через зарубежные сети связи не имеет ни

технического, ни экономического смысла для российского интернет-провайдера. Более того, такие «петли» делают провайдера неконкурентоспособным на высококонкурентном российском рынке ни по цене услуги, ни по ее качественным показателям.

В таблице 1.6.2 приведен расчет дополнительных затрат отрасли на организацию «петель российского трафика», исходя из предположения, что 60 % трафика российской интернет-экосистемы - «зарубежные петли внутрироссийского трафика».

Таблица 1.6.2

Объем трафика (из расчета 60 % от общего объема трафика российской интернет-экосистемы), Гбит/с	Затраты на услуги связи по предоставлению в пользование международных каналов связи, евро в месяц	Средняя стоимость кроссировки в телехаусе, евро в месяц	Средняя стоимость услуги IP-транзит, евро в месяц
7000	700*2100 = 1 470 000	700 * 150 = 105 000	700 * 5500 = 3 850 000
9000	900*2100 = 1 890 000	900* 150 = 135 000	900 * 5500 = 4 950 000
Итого в месяц, евро		5 425 000 – 6 975 000	
Итого в месяц, рубли		379 750 000 – 488 250 000	

Приведенный в таблице 1.6.2 расчет показывает, что дополнительные затраты на организацию «зарубежных петель российского трафика» в указанном объеме 60 % потребуют от участников рынка дополнительных затрат не менее 4,6 миллиардов рублей в год при ухудшении качества услуг и отсутствии дополнительных доходов от такого решения по пропуску трафика. Для отрасли это представляется экономически нецелесообразным и неприемлемым. Кроме того, емкости существующих зарубежных каналов, организованных российскими операторами связи за рубеж для пропуска интернет-трафика, технически не могут обеспечить передачу 60 % внутрироссийского трафика. Намеренная организация «зарубежных петель российского трафика» представляется экономически

нецелесообразной, так как в разы увеличивает текущие затраты всех участников рынка.

С учетом приведенного анализа можно констатировать, что сегодня в России сформирована одна из наиболее автономных замкнутых (самоподдерживающихся и самодостаточных) интернет-экосистем в Европе (рисунок 1.6.2).

Интеграция на национальном и международном уровне обеспечивает формирование единой национальной и международной ИКТ-инфраструктуры. Фрагментация и изоляция противоречат экономическим и политическим интересам Российской Федерации

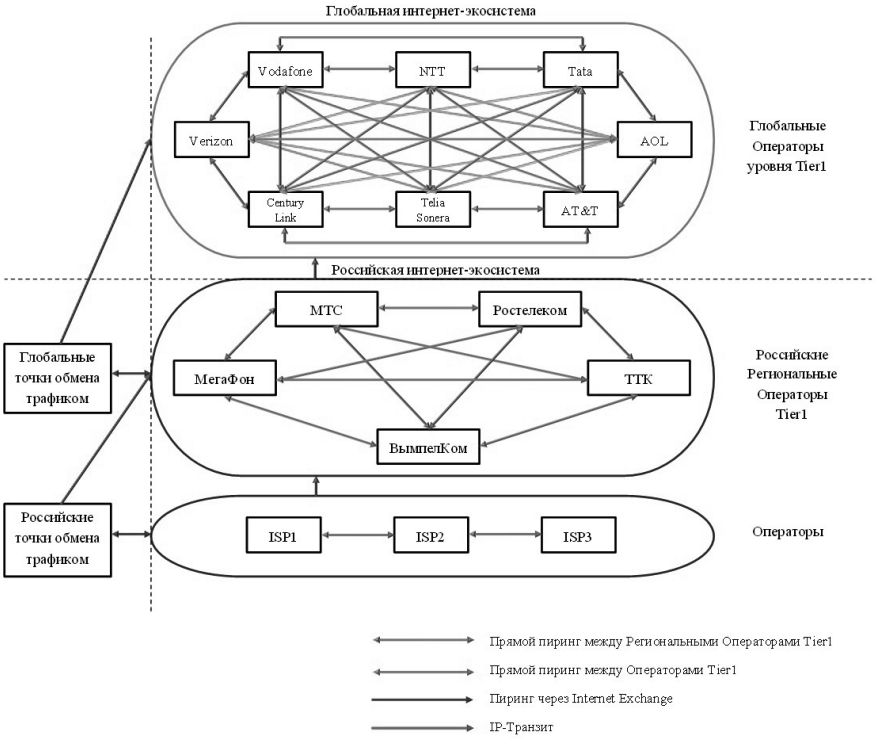

Рисунок 1.6.2. – Архитектура российской интернет-экосистемы и ее место в глобальной интернет-экосистеме

1.7. Гуманитарные аспекты работы в сети Интернет

1.7.1. Гуманитарные проблемы информационной безопасности

Как определено в Доктрине информационной безопасности Российской Федерации, под информационной безопасностью следует понимать состояние защищенности личности, общества и государства от внутренних и внешних информационных угроз, при котором обеспечиваются реализация конституционных прав и свобод человека и гражданина, достойные качество и уровень жизни граждан, суверенитет, территориальная целостность и устойчивое социально-экономическое развитие Российской Федерации, оборона и безопасность государства. В это определение, как видим, изначально вкладывается комплексное представление данного понятия. И только используя комплексный подход (по целям, средствам и организационным решениям), можно добиться прогресса в обеспечении безопасности.

Подобный взгляд на проблему информационной безопасности четко просматривается и в утвержденных Советом Безопасности Российской Федерации Основных направлениях научных исследований в области обеспечения информационной безопасности Российской Федерации. Указанные направления четко делятся на три группы: гуманитарные проблемы, технические проблемы и проблемы кадрового обеспечения. Причем удельный вес гуманитарных проблем только по их количеству достигает 80%.

Конкретно Советом Безопасности обозначены следующие приоритетные проблемы.

***Общеметодологические проблемы обеспечения информационной безопасности*:**

- развитие информационной сферы как системообразующего фактора жизни общества;
- разработка методологии обеспечения информационной безопасности как междисциплинарной отрасли научного знания;
- развитие системы обеспечения безопасности информационного (постиндустриального) общества;
- использование информационной сферы для решения задач конкурентоспособного развития России на современном этапе;
- информационное обеспечение государственной политики;
- сохранение культурно-нравственных ценностей российского народа.

***Проблемы развития нормативного правового и нормативного технического обеспечения информационной безопасности*:**

- развитие информационного права;
- нормативное правовое и нормативное техническое обеспечение безопасности интересов личности и общества в информационной сфере;
- нормативное правовое регулирование отношений в области развития системы массовой информации и коммуникации, информационного обеспечения государственной политики;
- нормативное правовое регулирование отношений в области создания и использования современных информационных технологий, индустрии информационных услуг;
- нормативное правовое обеспечение безопасности информационных и телекоммуникационных систем;
- нормативное правовое регулирование отношений в области борьбы с преступлениями в сфере информационно-коммуникационных технологий;
- нормативное правовое регулирование отношений в области обеспечения международной информационной безопасности.

***Проблемы обеспечения безопасности индивидуального, группового и массового сознания*:**

- обеспечение безопасности личности, общества и государства от деструктивных информационных воздействий;
- противодействие злоупотреблениям свободой распространения массовой информации, в том числе в сети Интернет.

К числу наиболее острых на сегодняшний день гуманитарных проблем информационной безопасности, по мнению большинства экспертов, можно отнести:

- обеспечение баланса интересов личности, общества и государства в информационной сфере, баланса между потребностью в свободном обмене информацией и допустимыми ограничениями ее распространения;
- национальные интересы России и информационное противостояние в современном мире;
- место и роль средств массовой информации в решении задач информационного обеспечения государственной политики Российской Федерации;

- ценностная ориентация личности, ее информационное обоснование и информационная безопасность;
- информационная безопасность и проблемы информационной культуры и этики;
- сохранение культурно-нравственных ценностей российского народа;
- информационное пространство и проблема целостности российского государства;
- социально-психологические последствия внедрения и широкого распространения современных информационных технологий;
- обеспечение информационно-психологической безопасности личности и общества;
- формирование системы международной информационной безопасности.

1.7.2. Формирование информационной культуры общества, этика в сфере использования информационных технологий

Остановимся далее более подробно на ключевой гуманитарной проблеме – формировании информационной культуры современного общества и связанным с этим обеспечением культуры информационной безопасности.

Здесь уместно привести высказывание Патриарха Московского и Всея Руси Кирилла, который в одном из своих выступлений говорил:

«… Невозможно сегодня защитить нашу молодежь и детей от информации, которую они потребляют. Только внутри самого человека должны быть выстроены эти рубежи обороны. Человек должен быть достаточно открыт к восприятию того, что несет ему современный мир, и одновременно должен быть способным защитить самые сокровенные глубины своей жизни, сохраняя свою национальную, духовную, религиозную, культурную самобытность, а вместе с этой самобытностью сохраняя нравственную систему ценностей…».

Развитие информационно-коммуникационных технологий порождает новые контексты проблем нравственности и правил поведения. Проблема правил поведения в «киберпространстве» выглядит чрезвычайно сложной, если учитывать предоставляемые здесь возможности анонимности и неконтролируемости с одной стороны, а с другой – многообразие групп (различаемых по возрасту, роду занятий, социальным позициям, национальной принадлежности и другим основаниям). В данном контексте

особое значение приобретают различия в ценностных ориентирах таких групп, в представлениях о хорошем и плохом, о допустимом и недопустимом, разные способы обращения, как говорят, со «своими» и «чужими». Нельзя не принимать во внимание и достаточно широко распространенного мнения, что в виртуальной среде должны утрачивать силу этические предписания, действующие в реальности.

Надежды на решение многих общественно важных проблем в информационной сфере, сегодня зачастую связывают с совершенствованием правового регулирования. Однако, если последнее сконцентрировано лишь на позитивном праве, писаных законах, игнорирующих сложившиеся представления о справедливости, социальных ценностях и этически приемлемых формах поведения, то расширение сферы его действия способно привести скорее к усилению конфликтов, чем к формированию устойчивого порядка. Ключевое значение в данных условиях приобретают вопросы этические, ибо, во-первых, по-настоящему действенным может быть лишь закон, базирующийся на прочных нравственных основаниях, а во-вторых, невозможно подвести под юридические нормы и проконтролировать все аспекты деятельности человека.

Каждая эпоха дает свои импульсы развитию нравственного сознания и создает трудности для такого развития. Вызовы, характерные для начала XXI века, не в последнюю очередь связаны с формированием глобального информационного пространства, позволяющего массам людей расширять свои представления о допустимых и рекомендуемых формах поведения. Аудио- и видеопродукция (в том числе, игровая индустрия) вносят свой вклад в широкое распространение образцов поведения, формально признаваемого не только безнравственным, но и противоправным. Это способствует распространению морального релятивизма – представления об относительном характере любых этических норм и неправомерности выдвижения абсолютных моральных императивов. Возведение финансово-экономической реальности в ранг основной, если не единственной, социальной реальности способствует тому, что разговоры о нравственных основаниях общества воспринимаются как несерьезные и даже неприличные. На фоне призывов к «толерантности» и «пониманию другого» задача «понимания себя» и общности, образуемой «мной» с «другими» выглядит в лучшем случае третьестепенной. Надежды на то, что распространение компьютерных технологий по всему миру позволит выработать согласованные стандарты поведения, которые могли бы стать

основой глобальной этики, выглядят скорее наивными, чем имеющими веские основания. Таким образом, вся сложность состоит в том, что глобализация экономики и создание «глобального информационного общества» сопровождаются скорее разрушением ранее сложившихся этических систем, чем формированием глобального нравственного сознания.

Серьезные расхождения в представлениях о поведении подобающем и недопустимом затрудняют сегодня совместную деятельность людей. Все чаще возникает необходимость договариваться о правилах, формулировать установки, которые должны разделяться всеми членами той или иной организации или сообщества, принимать этические кодексы и устанавливать санкции за их нарушение.

В итоге, по мере развития информационного общества происходит и формирование новой ступени информационной культуры. Педагогика и психология рассматривают информационную культуру личности как своеобразную подсистему, обеспечивающую должный уровень реализации ряда важнейших процессов ее жизнедеятельности. К этим процессам могут быть отнесены:

- генерация зрелых личностных смыслов и, тем самым, формирование адекватной и динамичной картины мира;
- эффективный информационный обмен, обеспечиваемый формированием ряда информационных умений: оценки полезности и истинности получаемой информации, отбора личностно значимой информации, поиска необходимой информации, в том числе о методах ее переработки, коммуникативных и языковых умений (восприятия и передачи), информационно-психологической самозащиты;
- выработка и совершенствование индивидуально-эффективных способов сохранения и усвоения информации;
- информационная психогигиена (экология) – саморегуляция информационных процессов в соотнесении их с актуальным состоянием организма;
- информационная нравственность, регулирующая вопросы доступа к чужой информации, использования информации для корыстных целей или целей давления на личность, ограничения доступа других к полезной информации.

Очевидно, в содержание понятия информационной культуры можно включить и такие необходимые процессы и умения, как способность к концентрации внимания на предмете, способность к логической и

ценностной обработке информации, способность увидеть новые комбинации свойств в отражаемых явлениях, т. е. способности творческого восприятия.

Таким образом, под информационной культурой общества в целом на современном уровне развития следует понимать способность его членов эффективно использовать доступные информационные ресурсы, средства информационных коммуникаций, а также передовые достижения в области информатизации и информационных технологий. Информационная культура - это понимание внутренних информационных механизмов, управляющих поведением человека и развитием общества.

К основным факторам, влияющим на уровень информационной культуры современного общества, можно отнести:

- состояние системы образования (определяет общий уровень интеллектуального развития людей, их материальные и духовные потребности);

- состояние информационной инфраструктуры общества (обеспечивает возможность получать, передавать и использовать необходимую человеку информацию, оперативно осуществлять те или иные информационные коммуникации);

- уровень демократизации общества (обеспечивает правовые гарантии доступа людей к необходимой им информации);

- экономическую состоятельность страны (гарантирует возможность получения ее гражданами необходимого образования, а также приобретения и использования ими современных продуктов ИТ-индустрии).

Важным направлением формирования информационной культуры общества является соблюдение этических норм в сфере информационных технологий.

Этические нормы (правила) зародились в недрах Сети и в 1994 году опубликованы в книге Вирджинии Ши Netiquette (Netiquette - "сетевой" этикет, правила, принятые в Сети или киберпространстве), в которой сформулированы десять основных правил, следование которым существенно облегчит жизнь участникам информационного обмена.

Правило 1: Помните, что говорите с человеком.

Одно из самых очевидных и все же самое часто нарушаемое правило в Сети. Очень многие забывают, что их собеседник — не компьютер, а живой человек, на которого оказывается реальное воздействие.

Правило 2: Придерживайтесь тех же стандартов поведения, что и в реальной жизни.

Интернет создает ощущение анонимности и возникает заблуждение, что в Сети правила поведения не так строги. Необходимо соблюдать этику общения, оставаясь в рамках закона как в реальном, так и в виртуальном пространстве.

Правило 3: Помните, что находитесь в киберпространстве.

Если ведете активную сетевую жизнь (посещаете несколько сообществ и форумов), важно помнить, какого стиля общения придерживаются ваши собеседники. В большинстве сложившихся сетевых коллективов есть свои правила, которыми с удовольствием поделятся с новичком. Универсальным является совет — первое время посмотреть и послушать.

Правило 4: Уважайте время и возможности других.

Если готовы поделиться со всем миром гениальной новостью, подумайте, а всем ли эта новость важна. Не следует также ожидать мгновенной реакции на сообщения.

Правило 5: Сохраняйте лицо.

Репутация в Интернете значит ничуть не меньше, чем в реальной жизни.

Правило 6: Помогайте другим там, где можете.

У пользователей могут возникнуть вопросы, ответов на которые в Сети нет. В таком случае, вся надежда на добрых людей, которые могут помочь. Мир становится лучше, когда мы помогаем другим, а мы сами — счастливее.

Правило 7: Не ввязывайтесь в конфликты.

Желание отличиться выдает новичка, а страстный спор может потрепать нервы, в первую очередь тому, кто его спровоцировал.

Правило 8: Уважайте право на частную переписку.

Не читайте чужие письма, не распространяйте в Сети личную информацию других людей (реальные имена, адреса, телефоны, фотографии) без их согласия.

Правило 9: Не злоупотребляйте своими возможностями.

Виртуальное пространство предполагает различный доступ к тем или иным ресурсам, различный уровень знаний в тех или иных вопросах. Обладая преимуществами над другими пользователями, не следует направлять их против людей.

Правило 10: Учитесь прощать другим их ошибки…

…или хотя бы время от времени вспоминайте, сколько ошибок сделали вы.

8 ноября 2016г. в Москве подписан Кодекс добросовестных практик (Кодекс этической деятельности (работы) в сети Интернет), разработанный в рамках инициированного Роскомнадзором проекта «Цифровой дом», цель которого — создание безопасной и комфортной цифровой среды. Под Кодексом поставили подписи представители около 30 организаций, подтвердившие свою готовность содействовать обеспечению безопасного информационного пространства в сети Интернет на основе требований законодательства Российской Федерации, положений международных договоров, рекомендаций уполномоченных органов государственной власти, а также создания, развития и внедрения мероприятий по формированию культуры безопасного поведения в Сети. Кодекс опубликован на сайте Федеральной службы по надзору в сфере связи, информационных технологий и массовых коммуникаций. Кодекс открыт для присоединения к нему любой заинтересованной стороны.

1.7.3. Глобальная культура информационной безопасности

Уровень информационной культуры непосредственно зависит от важнейших характеристик общественного развития и может служить не только интегральным показателем состояния общества, но и важнейшим фактором его развития. Именно поэтому вопросы информационной культуры и, в частности культуры информационной безопасности, в последние годы становятся предметом особого внимания влиятельных международных организаций.

Так, Генеральной Ассамблеей ООН в декабре 2002 года принята Резолюция, утверждающая принципы создания глобальной культуры кибербезопасности, которых должны придерживаться все участники глобального информационного общества (государственные органы, предприятия, организации и индивидуальные пользователи), которые создают информационные системы и сети, поставляют их, владеют и управляют ими, обслуживают или используют их.

Глобальная культура кибербезопасности в соответствии с принципами, утвержденными Генеральной Ассамблеей ООН, формируется на основе девяти взаимодополняющих элементов.

1) ***Осведомленность.*** Участники глобального информационного общества должны быть осведомлены о необходимости обеспечения

безопасности информационных систем и сетей и о том, что они могут для этого сделать.

2) ***Ответственность.*** Участники отвечают за безопасность информационных систем и сетей сообразно с ролью каждого из них. Они должны регулярно пересматривать свои политики, практику, меры и процедуры безопасности и оценивать их соответствие среде применения.

3) ***Реагирование.*** Участники должны принимать своевременные и совместные меры по предупреждению инцидентов, затрагивающих безопасность, их обнаружению и реагированию на них. Они должны обмениваться в надлежащих случаях информацией об угрозах и факторах уязвимости и прибегать к оперативному и эффективному сотрудничеству в деле предупреждения, обнаружения таких инцидентов и реагирования на них.

4) ***Этика.*** Поскольку информационные системы и сети используются в современном обществе повсюду, участникам необходимо учитывать законные интересы других сторон и признавать, что их действия или бездействие могут причинить вред другим.

5) ***Демократия.*** Безопасность должна обеспечиваться так, чтобы это соответствовало ценностям, которые признаются демократическим обществом, включая свободу обмена мыслями и идеями, свободный доступ к информации, конфиденциальность информации и коммуникации, надлежащую защиту информации личного характера, открытость и гласность.

6) ***Оценка риска.*** Все участники должны периодически оценивать потенциальный риск, чтобы выявлять угрозы и факторы уязвимости, анализировать ключевые внутренние и внешние факторы, сказывающиеся на безопасности, определять допустимую степень риска, выбирать надлежащие инструменты контроля, позволяющие регулировать риск потенциального ущерба информационным системам и сетям с учетом характера и значимости защищаемой информации.

7) ***Проектирование и внедрение средств обеспечения безопасности.*** Участники должны рассматривать соображения безопасности в качестве важнейшего элемента планирования и проектирования, эксплуатации и использования информационных систем и сетей.

8) ***Управление обеспечением безопасности.*** Участники должны применять комплексный подход к управлению обеспечением безопасности, опираясь на динамичную оценку риска, охватывающую все уровни деятельности участников и все аспекты их операций.

9) ***Переоценка.*** Участники должны подвергать вопросы безопасности информационных систем и сетей пересмотру и переоценке и вносить надлежащие изменения в политику, практику, меры и процедуры обеспечения безопасности, учитывая при этом появление новых и изменение прежних угроз и факторов уязвимости.

Генеральная Ассамблея ООН предложила всем соответствующим международным организациям и государствам-членам организации учитывать эти элементы в рамках их усилий по развитию в обществе культуры кибербезопасности. Сегодня компетентность в сфере информационных технологий и информационной безопасности становится необходимым условием успешной социализации личности в новой информационной среде общества.

1.7.4. Всеобуч в области культуры информационной безопасности

Очевидно, что одним из наиболее важных механизмов повышения компетентности и формирования культуры информационной безопасности является массовое обучение людей тому, как ценить безопасность, ответственно использовать компьютерные технологии, как реагировать на инциденты, связанные с нарушением безопасности, как восстанавливать компьютерные системы и информацию после таких инцидентов, как обращаться с доказательствами, которые могут потребоваться во время судебного расследования компьютерных преступлений, как и кому сообщать об инцидентах, связанных с нарушением информационной безопасности. Практика показывает, что обучение основам информационной безопасности и преподавание этики использования компьютерных технологий больше способствуют укреплению безопасности, чем какие-либо другие меры. Без знания и соблюдения норм нравственности и этики, особенно молодыми людьми, по всей видимости, не будут преодолены проблемы компьютерной и сетевой безопасности.

Таким образом, не вызывает никаких сомнений высокая актуальность проблемы формирования информационной культуры у подрастающего поколения. Еще раз подчеркнем, что это вытекает, с одной

стороны, из универсальной онтологической значимости информации в бытии, а с другой стороны, связано с повышением функционального значения информации, информационной культуры в жизни человека в современном информационном обществе, в котором информация стала системообразующей ценностью.

Обучение должно идти по двум направлениям: *профессиональное* и *массовое* обучение. Профессиональное обучение ориентировано на целевую аудиторию (студентов средних специальных и высших учебных заведений, слушателей курсов повышения квалификации или центров переподготовки и сертификации специалистов в области информационной безопасности). Профессиональное обучение может быть *основным* и *дополнительным*.

Основное обучение должно обеспечиваться всем спектром образовательных учреждений (начального, среднего и высшего профессионального образования). Современная система образования, помимо подготовки профессионалов в области обеспечения информационной безопасности, должна участвовать в «информационном всеобуче», в воспитании активных и информированных граждан, в формировании новой культуры информационной безопасности, которая соответствовала бы современному уровню развития информационных технологий и способствовала бы органичному вхождению людей в информационное общество. Обучение морали, этике и ответственному использованию информации и информационных технологий, интегрированное в гуманитарные и естественнонаучные дисциплины призвано выполнять уникальную функцию подготовки школьников и студентов к жизни в информационном пространстве. Сегодня, для того чтобы выпускник образовательного учреждения (средней школы, лицея, колледжа или вуза) не чувствовал себя человеком второго сорта в информационном обществе, а хорошо в нем ориентировался и имел бы достаточно высокие шансы найти удовлетворительно оплачиваемую работу, он должен уже в процессе своего образования получить достаточно высокую компетентность в сфере информационных технологий и информационной безопасности.

Современная жизнь требует от всех членов информационного общества постоянного повышения квалификации, непрерывного обновления знаний, освоения новых видов деятельности. В идеале повышение образовательного уровня человека должно продолжаться в течение всей жизни. В связи с этим на смену парадигме

«поддерживающего» или «просветительского» образования, пришла инновационная парадигма образования, важнейшей составляющей которой стала идея «образования в течение всей жизни» или непрерывного образования.

Реализация этой парадигмы связана с *дополнительным обучением* на тренингах, семинарах или курсах повышения квалификации с последующим получением соответствующего сертификата. Повышением квалификации специалистов в области информационной безопасности сегодня занимаются не только профильные вузы, но и учебные центры дополнительного образования (как правило, негосударственные), созданные компаниями и организациями, активно работающими на рынке средств и услуг, связанных с защитой информации.

Цель *массового обучения* состоит в том, чтобы вовлечь в процесс обучения как можно больше людей и добиться максимального эффекта при ограниченных ресурсах. Западный опыт использования армии добровольцев для организации местных, региональных и национальных мероприятий и благотворительных акций свидетельствует о том, что подход, аналогичный подходу к решению серьезных социальных проблем (таких, как распространение СПИДа и других угрожающих жизни заболеваний), может быстро и эффективно обеспечить повышение осведомленности населения о проблемах информационной безопасности и соответствующих превентивных мерах.

Наблюдающийся в последнее время взрывной рост преступлений, совершаемых с использованием Интернета, побудил государственные органы ряда стран создать *центры, информирующие о киберпреступлениях*. Эти центры занимаются сбором информации о компьютерных инцидентах в киберпространстве и доведением ее до широкой общественности. Все эти центры можно разделить на две категории:

- центры, которые не являются правоохранительными и выявляют, регистрируют и анализируют факты всех компьютерных инцидентов, а также предоставляют о них информацию и консультируют население;

- правоохранительные центры, которые действуют как национальные информационные центры по компьютерным преступлениям, связываясь непосредственно с другими национальными и международными группами реагирования на компьютерные инциденты

для выявления потенциальных угроз и оценки рисков. Кроме того, эти центры обеспечивают подготовку кадров для правоохранительных органов, а также сотрудничают с международными и частными правоохранительными агентствами.

Роль этих центров в повышении осведомленности граждан о проблемах информационной безопасности достаточно велика. Они функционируют в качестве первого пункта контактов в тех случаях, когда происходит или предполагается, что произошел, компьютерный инцидент. Центры также консультируют тех, кто хочет больше узнать о мерах, которые используют в целях выявления и предотвращения сетевых вторжений, а также о способах восстановления систем и данных после успешных кибератак.

Помимо рассмотренных методов повышения информированности и массового обучения есть и другие, которые широко используются в западных странах, хотя и являются менее эффективными. Они попадают в категорию активистской деятельности, то есть общественно-политического движения, пропагандирующего активное вмешательство граждан в решение острых социальных и политических проблем. К этим методам относится пропаганда и создание «горячих линий». Информационно-пропагандистские группы работают с общественностью, корпорациями и правительствами для того, чтобы повысить уровень осознания обществом проблем современного компьютеризированного мира и повлиять на формирование культуры информационной безопасности. Стратегия вовлечения в компании большого числа граждан приносит плоды в виде массового осознания угроз личной, корпоративной и национальной безопасности, которое ведет к усилению общественного давления на законодателей и государственные органы, вынуждая их должным образом защищать интересы граждан и общества.

«Горячие линии» позволяют широкой общественности брать на себя инициативу в наблюдении и уведомлении о компьютерных инцидентах. В этой связи интересен пример американской Национальной стратегии безопасности киберпространства (*National Strategy for the Security of Cyberspace – NSSC*), которая одной из своих приоритетных задач считает вовлечение обычных пользователей в пропаганду идеи личной информационной безопасности, а также безопасности их сообщества и всего общества в целом. В большинстве случаев стратегия заключается в организации приема сообщений по каналам «горячей линии» от лиц-свидетелей инцидентов компьютерной безопасности. Во многих странах

ответственными за прием сообщений и принятие соответствующих мер являются правоохранительные органы и поставщики услуг Интернета.

В целом задача формирования современной культуры информационной безопасности требует использования возможностей всех звеньев системы непрерывного образования для повышения осведомленности всех членов общества о проблемах безопасности информационных систем и сетей, осознания каждым человеком своей роли и ответственности, обучения людей этике в сфере информационных технологий, целенаправленной деятельности государственных органов, больших общественных усилий по нескольким фронтам: в сфере законодательства, нормативного регулирования, а также активной деятельности граждан.

1.7.5. Информационное оружие и информационная война

При рассмотрении гуманитарных проблем информационной безопасности необходимо учитывать такие факторы как «информационное оружие» и «информационная война». Под этим понимается целенаправленное использование информационных средств и современных информационных технологий для ведения вооруженной борьбы на межгосударственном уровне. Анализ тенденций в развитии военно-силового противоборства (см. рис. 1.7.1) явно показывает, что «кибервойны» превращаются в основное средство борьбы в XXI веке.

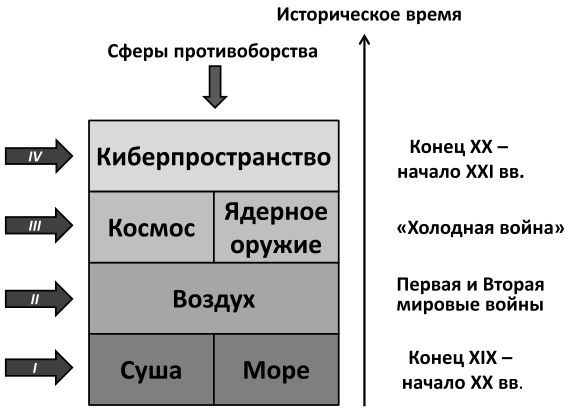

Рис. 1.7.1 Пространство военно-силового противоборства

В связи с ростом информационной зависимости всех сфер жизнедеятельности информационного общества «информационное оружие» представляет серьезную военную угрозу. Так, по оценке экспертов, нарушение работы компьютерных сетей, используемых в системах управления государственными и банковскими структурами, путем вывода из строя вычислительных и связных средств или уничтожения хранящейся в сетях информации способно нанести экономике страны настолько серьезный ущерб, что его можно сравнивать с ущербом от применения ядерного оружия.

По мнению ряда специалистов, суть информационной войны состоит в достижении какой-либо страной (или группой стран) подавляющего преимущества в информационной области, позволяющего с достаточно высокой степенью достоверности моделировать поведение «противника» и оказывать на него в явной или скрытой форме выгодное для себя влияние. Таким образом, можно предположить, что страны, проигравшие информационную войну, проигрывают ее «навсегда», поскольку их возможные шаги по изменению ситуации, которые сами по себе требуют колоссальных материальных и интеллектуальных затрат, будут контролироваться и нейтрализовываться победившей стороной.

Число стран, разрабатывающих сегодня информационное оружие (в основном с использованием сети Интернет), превышает 120, разрабатывающих оружие массового уничтожения около 30.

основные цели информационной войны:

- дезорганизация деятельности управленческих структур, транспортных потоков и средств коммуникации;
- блокирование деятельности отдельных предприятий и банков, а также целых отраслей промышленности путем нарушения многозвенных технологических связей и системы взаиморасчетов, проведения валютно-финансовых махинаций и т.п.;
- инициирование крупных техногенных катастроф на территории противника в результате нарушения штатного управления технологическими процессами и объектами, имеющими дело с большими количествами опасных веществ и высокими концентрациями энергии;
- массовое распространение и внедрение в сознание людей определенных представлений, привычек и поведенческих стереотипов;
- вызов недовольства или паники среди населения, а также провоцирование деструктивных действий различных социальных групп.

При этом объектом информационного противоборства является

любой объект, в отношении которого возможно осуществление информационного воздействия (в том числе, и применение информационного оружия) либо иного воздействия (силового, политического, экономического и т.д.), результатом которого будет модификация его свойств как информационной системы. Исходя из этого, в качестве таких объектов можно рассматривать:

- систему социальных отношений информационного общества;
- систему политических отношений информационного общества;
- систему психологических отношений информационного общества.

Таким образом, объектом информационного противоборства может стать любой сегмент информационно-психологического пространства, в том числе:

- массовое и индивидуальное сознание граждан;
- социально-политические системы и процессы;
- информационная инфраструктура;
- информационные и психологические ресурсы.

К психологическим ресурсам общества относятся следующие компоненты:

- система ценностей общества;
- психологическая толерантность системы ценностей (устойчивость системы ценностей по отношению к внешним или внутренним деструктивным воздействиям);
- индивидуальное и массовое сознание граждан;
- психологическая толерантность сознания граждан (устойчивость сознания граждан к манипулятивному воздействию и вовлечению в противоправную деятельность манипулятивными методами тайного принуждения личности);
- психическое здоровье граждан;
- толерантность психического здоровья граждан (устойчивость психического здоровья по отношению к внешним или внутренним деструктивным воздействиям).

Следует отметить, что информационное противоборство в области социально-политических отношений имеет глубокие исторические корни. Примеры использования такого рода информационного оружия и борьбы с

ним мы можем найти и в нашей истории. Так, на рис. 1.7.2 приведен документ, относящийся к XVIII веку.

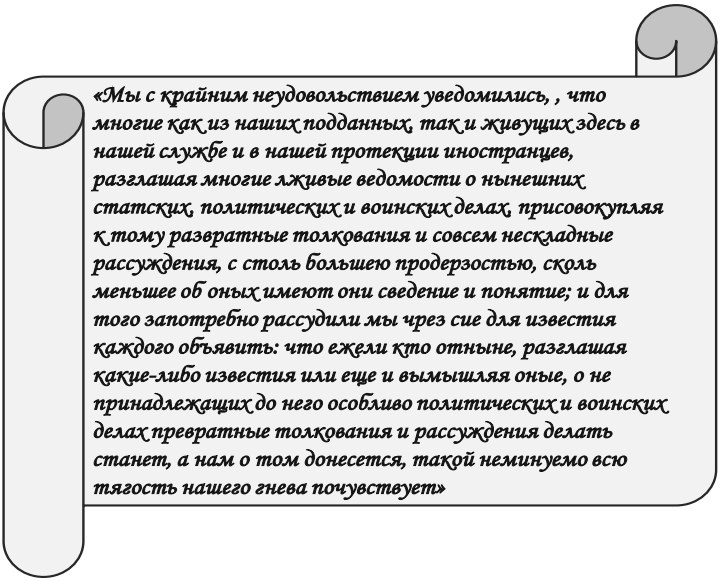

Рис. 1.7.2 Указ Императрицы Елизаветы Петровны
(СПб ведомости, 1750 г., №46)

С тех пор возможности информационного противоборства, конечно же, значительно расширились. Основными объектами применения информационного оружия в современных условиях являются:

- компьютерные и связные системы, используемые государственными и правительственными организациями при выполнении своих управленческих функций;
- военная информационная инфраструктура, решающая задачи управления войсками и боевыми средствами, сбора и обработки информации в интересах вооруженных сил;
- информационные и управленческие структуры банков, транспортных и промышленных предприятий;
- средства массовой информации, и в первую очередь электронные (радио, телевидение, Интернет и т.д.).

Определив объекты информационного противоборства, необходимо

обозначить и те субъекты, которые могут использовать те или иные виды современного информационного оружия. Ими могут являться:

- государства, их союзы и коалиции;
- международные организации;
- негосударственные (в том числе международные) незаконные вооруженные формирования и организации террористической, экстремистской, радикальной политической, радикальной религиозной направленности;
- транснациональные корпорации;
- виртуальные социальные сообщества;
- медиа-корпорации;
- виртуальные коалиции.

Такие субъекты обладают определенными признаками, указывающими на заинтересованность и возможность осуществлять информационное противоборство. К этим признакам можно отнести:

- наличие у субъекта в информационно-психологическом пространстве собственных интересов;
- наличие в составе субъекта специальных сил (структур), функционально предназначенных для ведения информационного противоборства или уполномоченных на его ведение;
- обладание и/или разработка информационного оружия, средств его доставки и маскировки;
- наличие под контролем субъекта определенного сегмента информационного пространства, в пределах которого он обладает преимущественным правом устанавливать нормы регулирования информационно-психологических отношений (на правах собственности, закрепленных нормами национального и международного законодательства) или государственным суверенитетом;
- существование в официальной идеологии положений, допускающих участие субъекта в информационном противоборстве.

Особо следует выделить роль сетевых корпораций, играемую ими в информационно-психологической борьбе, которую можно охарактеризовать следующим образом.

1. Транснациональные корпорации в глобальном информационном обществе практически обладают всеми признаками суверенного государства – территорией, определяемой ареалом распространения их сетевой инфраструктуры, стратегическими ресурсами

(информационными потоками в их информационных и телекоммуникационных системах), «населением» (штатом сотрудников) и относительно полным суверенитетом.

2. Транснациональные корпорации, разрабатывая новые информационно-коммуникационные технологии, развивая свои информационные и телекоммуникационные системы и сети, контролируя циркулирующие по ним потоки, создают среду в которой могут разворачиваться боевые действия между участниками информационно-психологического противоборства.

Можно считать, что информационная война ведется субъектами информационного противоборства в сфере, искусственно создаваемой человеком в результате разработки новых средств воздействий (информационных технологий) и средств доступа к уязвимым объектам нападения (сетевой инфраструктуры), т.е., фактически, в условиях и по законам, определяемым разработчиками и владельцами сетей и технологий.

Как можно противостоять информационному воздействию? Помимо технологических средств защиты информации и обеспечения информационной безопасности (противодействия так называемым киберугрозам), важнейшая роль отводится здесь готовности общества к информационному противоборству и способности его противостоять различного рода манипуляциям общественным и личным сознанием граждан.

Из анализа военных и международных аспектов проблемы информационной войны можно сделать сегодня следующие основные выводы:

- ряд стран стремится получить преимущество в создании систем и средств ведения информационной войны, что может представлять серьезную угрозу национальной безопасности России;
- создание целостного комплекса средств и методов ведения информационной войны может осуществляться постепенно, по мере развития в мире базовых информационных технологий, что позволяет осуществлять мониторинг этого процесса;
- тема информационного оружия и информационной войны, в силу своей чрезвычайной важности для безопасности страны, требует комплексной проработки ее военно-стратегических, правовых, разведывательных и контрразведывательных аспектов, а также координации усилий всех заинтересованных ведомств России.

1.8. Свободные лицензии и их ограничения. Преимущества и недостатки открытой модели разработки программного обеспечения.

1.8.1. Роль свободного программного обеспечения в развитии отрасли ИТ

Свободным называется программное обеспечение, права на использование которого определены простой (неисключительной) лицензией (см. статью 1236 Гражданского кодекса Российской Федерации — далее по тексту ГК), разрешающей пользователю:

использовать программу для ЭВМ в любых, не запрещенных законом целях;

получать доступ к исходным текстам (кодам) программы как в целях изучения и адаптации, так и в целях переработки программы для ЭВМ;

распространять программу (бесплатно или за плату, по своему усмотрению);

вносить изменения в программу для ЭВМ (перерабатывать) и распространять экземпляры измененной (переработанной) программы с учетом возможных требований наследования лицензии.

Программное обеспечение с открытыми исходными кодами (англ. open source software) – программное обеспечение, исходные коды которого свободно доступны. Одна лишь доступность кода, тем или иным способом, не даёт оснований считать его свободным, поскольку не влечет передачи права свободного (неограниченного) распространения, модификации и права распространения модифицированного кода.

Основные положения в области создания, распространения и использования свободного программного обеспечения, в том числе для государственных и муниципальных нужд, отражены в ГОСТ Р 54593-2011 «Информационные технологии. Свободное программное обеспечение».

Свободное программное обеспечение защищено авторским правом при помощи свободных лицензий. Лицензия свободного ПО (англ. free software licence) — это лицензия на программное обеспечение, которая предоставляет получателям права копировать, модифицировать и повторно распространять программу. В случае, так называемого «проприетарного ПО», эти действия, как правило, запрещены.

Существует несколько разновидностей свободных лицензий. В основном, они подразделяются на два вида: требующие распространенения модифицированных версий ПО на тех же условиях, на

которых были предоставлены права использования (например, лицензия GNU GPL), и разрешающие изменять права использования модифицированных версий (например, лицензия BSD).

Свободные лицензии разработаны также для различного контента (например, Creative Commons), аппаратного обеспечения и т.д.

1.8.2. Использование свободного программного обеспечения в сети Интернет

Самым масштабным примером успешного использования свободного ПО является Интернет, который основан на стеке TCP/IP, веб-сервере Apache, серверах DNS, ftp и др. - всё это СПО, базирующееся на открытых стандартах. Накоплен богатый опыт внедрения СПО в государственных структурах, например: в парламенте и ряде ключевых министерств Франции, в испанской провинции Естремадура, в странах Латинской Америки, в Китае. Анализ этих внедрений показывает, что они успешны как правило, там, где удалось наладить взаимодействие с сообществом разработчиков и обеспечить открытость проекта. Там же, где пытаются воспользоваться свободными программами, не возвращая собственные наработки в проекты, происходит обособление и потеря динамики; именно так значительно затормозилась разработка китайского RedFlag Linux.

Разработкой свободных программ за рубежом занимается значительное число крупных фирм, из разработчиков дистрибутивов Linux: Американские Red Hat, и Novell, SuSE, Французско-Бразильская Mandriva, Южно-Африканская Ubuntu. При этом на свободную разработку переходят такие крупнейшие «традиционные» производители: IBM (среда разработки Eclipse, участие в различных свободных проектах), Sun (открывает коды Solaris, Java, приобрел и открыл OpenOffice).

Многие фирмы, производящие мобильные и встроенные устройства, начали широко использовать Linux, среди них Nokia, Motorolla, Siemens. Особенно интересно решение Siemens о полном переходе для всех своих устройств на ОС Linux в течение нескольких лет.

Во многих странах разработка свободного ПО поддерживается на государственном уровне. Особый интерес представляют: проект FLOSS Программы по технологиям информационного общества Европейской комиссии, программа OSOR (the European Open Source Repository), проект разработки Национальной защищенной ОС в Казахстане, законы провинций Испании и Италии и ряда штатов Индии, государственная поддержка Red Flag Linux в Китае.

Большое внимание уделяется переходу на использование открытых стандартов: проект e-GIF (e-Government Interoperability Framework) в Англии, SAGA в Германии, широкий переход на открытый формат документов ODF.

Таким образом, даже при наличии крупных коммерческих фирм, ориентированных на СПО, за рубежом проекты свободного ПО давно уже находят государственную поддержку, а результаты разработки рекомендуются к широкому внедрению.

В России СПО имеет более широкое распространение, чем кажется на первый взгляд. Например, в российской зоне Интернет преобладает СПО: более 80% серверов работает под ОС Linux или FreeBSD, более 90% веб-серверов занимает свободный Apache, при этом доля использования свободного ПО даже выше среднемирового показателя.

Среди систем, получивших в России государственную сертификацию много свободных: Atlix производства НПО «Атлас», ИВК-Кольчуга – межсетевой экран на базе СПО сертифицированный на работу с гостайной, различные дистрибутивы Альт производства «Базальт СПО», сертифицированные ФСТЭК России, ОС Astra Linux производства АО «НПО РусБИТех», линейка операционных систем НТЦ ИТ РОСА .

В России уже накоплен определенный опыт внедрения свободных программ: Linux используется в Центральном Банке России и в суперкомпьютерных кластерах СКИФ, интересные внедрения в городе Новгороде и в городе Алексине (Тульская область) в рамках ФЦП «Электронная Россия». В ФГБУ НИИ «Восход» установлено около двух сотен серверов «Эльбрус» с российской ОС Альт, на которых развернута система, обеспечивающая паспорта нового поколения.

В сфере образования имеется опыт поставок Linux в школы начиная с 2004 года. Свободные программы широко используются в ведущих вузах: МГУ, МИФИ, МФТИ, МГТУ, МПГУ, Петербургский политехнический институт, Ижевский и Нижегородский университеты. Ежегодно в январе в Переславле-Залесском проходит конференция «Свободные программы в высшей школе».

В то же время существует проблема масштабных внедрений, связанная с тем, что крупные системные интеграторы только осваивают решения на базе СПО, а отечественные разработчики СПО часто не имеют опыта масштабных внедрений. Для решения этой проблемы необходима консолидация усилий системных интеграторов и отечественных разработчиков СПО.

В целом, сообщество разработчиков СПО в России пока немногочисленно, как и отечественные компании-разработчики. Их доля на рынке значительно меньше, чем в США и странах Европы, процесс консолидации находится в начальной стадии.

1.8.3. Предпосылки для выхода России на передовые позиции в мире в области производства СПО

Преодоление технологического отставания состоит не только в значительном снижении доли заимствованных технологий, но и в выходе России на мировой рынок как поставщика передовых технологических решений. В области разработки СПО предпосылками для выхода России на передовые позиции являются низкий материальный порог вхождения, отсутствие необходимости закупки дорогих лицензий и патентов, возможность быстрого наращивания необходимого интеллектуального потенциала. Важным фактором является наличие команд разработчиков, интегрированных в международные проекты разработки свободного ПО.

Основными индикаторами развития СПО в стране могут стать:

состояние нормативно-правовой базы;

ориентация на открытые стандарты (наличие профилей стандартов);

открытая инфраструктура разработки;

наличие опыта внедрения и техподдержки;

доля использования СПО в образовании.

1.8.4. Правовой фундамент развития программного обеспечения (в том числе свободного ПО) в Российской Федерации

Стремясь обрести технологический суверенитет, органы государственной власти предпринимают меры по поддержке развития СПО в России. Правовым фундаментом импортозамещения программного обеспечения стала нормативная база, включающая федеральные законы, распоряжения правительства, стандарты и методические рекомендации, направленнные на повышение эффективности, унификацию и стандартизацию информационных технологий. Эти правовые акты регламентируют разработку, внедрение и применение свободного программного обеспечения во всех сферах жизни общества: государственном управлении, экономике, культуре, быту. Одновременно государственная политика предусматривает повышение квалификации персонала и правовую охрану результатов интеллектуальной деятельности.

С 1 января 2016 года, согласно постановлению Правительства Российской Федерации от 16.11.2015 г. № 1236 «Об установлении запрета на допуск программного обеспечения, происходящего из иностранных государств, для целей осуществления закупок для обеспечения государственных и муниципальных нужд», при закупке программного обеспечения органы государственной власти обязаны отдавать приоритет российским разработкам, включенным в Единый реестр российских программ для электронных вычислительных машин и баз данных. Госкорпорациям также было рекомендовано при закупке ПО отдавать предпочтение российскому программному обеспечению.

Распоряжением Правительства Российской Федерации от 27 июля 2016 года № 1588 «Об утверждении плана перехода органов исполнительной власти и государственных внебюджетных фондов на использование отечественного программного обеспечения» был утверждён соответсвующий план. В марте 2017 уточнены требования к отечественному офисному ПО, включенному в реестр российского программного обеспечения (постановление Правительства Российской Федерации от 23 марта 2017 № 325). К офисному ПО, согласно этого постановления, относятся операционная система, коммуникационное ПО, почтовые приложения, интернет-браузер, системы электронного документооборота, средства антивирусной защиты, файловые менеджеры, редакторы презентаций, табличный редактор, текстовый редактор, справочно-правовая система и др.

1.8.5. Единый реестр российских программ для электронных вычислительных машин и баз данных

Знаковым событием для государственной программы импортозамещения стало создание в начале 2016 года единого реестра российских программ для электронных вычислительных машин и баз данных (далее – Реестр), который размещен на сайте Минкомсвязи России (https://reestr.minsvyaz.ru/). Он позволил отрегулировать процедуру госзакупок, обеспечил госведомствам и федеральным органам исполнительной власти понятный и прозрачный механизм выбора софтверных решений.

Одновременно Реестр стал индикатором активности российских разработчиков программных решений: заявки поступают в реестр ежедневно, в стране создаются тысячи российских продуктов.

Согласно 44-ФЗ от 5 апреля 2013 г. «О контрактной системе в сфере закупок товаров, работ, услуг для обеспечения государственных и муниципальных нужд» правительство может вводить дополнительные ограничения при закупке ПО. Такие ограничения реализованы в постановлении Правительства Российской Федерации от 16.11.2015 г. № 1236: федеральные органы государственной власти и внебюджетные фонды, приобретая ПО для собственных нужд, обязаны отдавать предпочтение отечественным разработкам. Отечественным считается ПО из Реестра. Если в Реестре есть ПО нужного класса, то покупатель обязан объявить конкурс без допуска зарубежного ПО. Исключением является случай, когда всё имеющееся в классе ПО не соответствует по функциональным, техническим или эксплуатационным характеристикам. Обоснование публикуется в составе конкурсной документации.

Вопрос о включении ПО в Реестр решается экспертным советом (создан приказом Минкомсвязи России от 30 декабря 2015 г. № 615). При проведении экспертизы оценивается российское «происхождение» программного продукта. Вопрос о качестве ПО не рассматривается: качество определяет сам потребитель, исходя из конкретных задач.

Ориентируясь на программы из Реестра при проведении конкурсов на закупку ПО, потребителю не надо многократно проводить экспертизу на его «отечественность», чтобы не отягощать заказчиков и поставщиков многократными экспертизами.

Критерии отбора ПО в Реестр совершенствуются. Так, осенью 2017 года вступило в силу постановление Правительства Российской Федерации от 23 марта 2017 г. № 325, в котором обозначены дополнительные требованиях к ПО для федеральных органов исполнительной власти и внебюджетных фондов, сведения о котором включены в реестр российского программного обеспечения. В качестве одного из новых требований зафиксирована совместимость прикладной программы не только с MS Windows, но и с двумя другими ОС из Реестра, в том числе ОС базе Linux. Это требование разрывает «замкнутый круг», который не дает организациям перейти на импортонезависимое ПО. На сегодняшний день ИТ-инфраструктуры подавляющего большинства российских организаций и предприятий построена на программном обеспечении, работающем только под операционной системой MS Windows. С одной стороны, замена ОС в такой ситуации равносильна остановки деятельности целого госведомства или корпорации. С другой – есть требования регулятора, Методические рекомендации Минкомсвязи России, где

сказано, каких показателей по импортозамещению к каким срокам необходимо достигнуть. Введение дополнительных требований к реестровому ПО снимает это противоречие и позволяет плавно провести миграцию на отечественное ПО.

1.8.6. Операционные системы на базе СПО – системообразующий элемент ИТ-инфраструктуры российских организаций и предприятий

Операционная система (ОС) служит для управления работой всего оборудования и программного обеспечения, включенного в состав цифровой инфраструктуры организации или предприятия.

Принципиально важно, чтобы жизненный цикл ОС обеспечивался отечественной инфраструктурой: технологической, организационной, образовательной, чтобы ОС:

была произведена в России, и ее дальнейшее развитие контролировалось из российской юрисдикции;

позволяла управлять цифровой инфраструктурой практически неограниченного масштаба и сложности;

обладала высокой отказоустойчивостью и способностью к работе в заданных штатных режимах даже в условиях экстремальных нагрузок;

имела требуемый уровень защиты, подтвержденный сертификатами ФСТЭК России;

обеспечивалась гарантированной технической поддержкой на всей территории страны.

1.8.7. Отечественная инфраструктура разработки СПО, репозиторий

Операционные системы на основе СПО, как правило, выпускаются в виде дистрибутивов – наборов программ, необходимых для решения комплексной задачи потребителя (физического или юридического лица). Организация сразу получает все приложения, необходимые сотрудникам в работе – их не надо дополнительно приобретать или искать в Интернете. Например, в состав дистрибутива операционной системы могут быть включены программы, необходимые для управления ИТ-инфраструктурой организации или предприятия: сама ОС, инструментарий службы каталогов, средства виртуализации и построения облачных ресурсов, программа для обеспечения коллективной работы пользователей, веб-браузер, почтовая программа, средства для работы с мультимедиа, офисный пакет (текстовый, графический и табличный редакторы, редакторы презентаций и формул) и др.

При разработке и модернизации дистрибутивов ОС по государственному заказу учитывается специфика требований заказчика: в состав ОС могут включаться специализированные компоненты, повышающие уровень ее защищенности. Такие дистрибутивы могут быть сертифицированы во ФСТЭК России в соответствии с требованиями к их использованию.

Все дистрибутивы ОС создаются компаниями-разработчиками на инфраструктуре сборки (репозитории) – своем или чужом. Сегодня в мире есть несколько наиболее крупных публичных репозиториев: у американской фирмы RedHat (Fedore Core), французской Mandriva, международный проект Debian, российский проект Сизиф (держатель – компания «Базальт СПО»), который содержит более 18 000 пакетов программ.

Репозиторий представляет собой «фабрику» по производству программного обеспечения, объединяющую инфраструктуру разработки и сборки, банк (хранилище) программ и сообщество разработчиков (как правило, международное).

Для динамичного развития отечественных разработок на базе СПО необходимо, чтобы в российской юрисдикции находился хотя бы один репозиторий, в состав которого входят:

инфраструктура для реализации полноценного жизненного цикла программного обеспечения: его разработки, модификации и сопровождения на весь период эксплуатации потребителем, его взаимосвязь и целостность разработки;

единый банк программных решений под свободными лицензиями, включая типовые программные комплексы для решения задач разных категорий потребителей: государственных органов и муниципалитетов, вузов и школ, поликлиник и больниц, крупных корпораций и пр. (для программных решений должна фиксироваться информация о лицензии);

система контроля зависимостей между пакетами (при формировании пакетов программ, разработчик указывает разного рода зависимости с другими пакетами (по выполнению, по сборке, и др.); автоматическая система контроля анализирует как исполняемые бинарные файлы, так и программы на скриптовых языках в целях обнаружения неразрешенных зависимостей или паразитных зависимостей; такая проверка обеспечивает высокий уровень интеграции всех пакетов в единый репозитарий, на основании которого можно создавать целостные решения);

эталонная сборочная среда для СПО (она должна включать как исходные тексты программ со всеми зависимостями времени исполнения и времени сборки, так и все необходимые средства сборки бинарных пакетов (компиляторы, системные библиотеки и др.), настраиваемые системы генерации и установки дистрибутива; наличие такой среды дает возможность независимой фирме или потребителю СПО самостоятельно продолжить его разработку; эталонная сборочная среда должна обеспечивать автоматический контроль зависимостей и целостность системы, она должна иметь возможность функционировать в распределенном режиме, с созданием локальных репозиториев в ведомствах или фирмах, с возможностью их синхронизации; главное требование к ней – публичность и открытость);

единый подконтрольный государству держатель эталонной сборочной среды, который реализует единый регламент функционирования, обновления и доступа к инфраструктуре разработки и ПО.

Важно отметить, что технологическую независимость обеспечивает не любое СПО. Ряд российских операционных систем создается на основе западных репозиториев. Политику их развития определяют зарубежные разработчики, поэтому в эти репозитории сложно включить, например, компоненты для работы с серверами и десктопами на базе процессоров российского производства. На сегодняшний день единственным технологически независимым отечественным репозиторием является Sisyphus - один из четырех крупнейших в мире репозиториев свободных программ. Политику его развития с учетом специфики российского законодательства и требований регуляторов, совместимости с отечественным ПО и аппаратными платформами определяет российская компания-держатель репозитория. На базе Sisyphus реализуется полный технологический цикл подготовки, выпуска и поддержки дистрибутивов ОС Альт. Благодаря стопроцентному контролю над репозиторием, разработчик ОС регулярно выпускает обновления ОС, а обнаруженные уязвимости закрываются в день их объявления, операционная система устойчиво функционирует на процессорах как зарубежного, так и отечественного производства. На базе репозитория Sisyphus формируется сообщество российских разработчиков прикладного ПО, полностью совместимого с ОС. Разработанные прикладные программы включаются в реестр отечественного ПО. Отечественные потребители автоматизируют разнообразные бизнес-процессы, приобретая готовое ПО или заказную разработку.

1.8.8. Структура разработки и сборки ОС Альт. Участие Базальт СПО в проектах разработки ПО для ОС LInux

Разработка ОС Альт тесно связана с открытыми проектами разработки ПО под вободными лицензиями (верхняя область рисунка 1.8.1). Во многих этих проектах участвуют и сотрудники компании Базальт СПО. Например в проектах strace, glibc, libreoffice, samba, wine разработчики Базальт СПО принимают самое непосредственное участие. Ниже приведена сводная информация об участии Базальт СПО в открытых проектах разработки ПО (см. таблицу 1).

Исходные тексты этих проектов являются основой для сборки соответствующих пакетов в нестабильном рабочем репозитории «Сизиф» (рисунок 1.8.1).

Рисунок 1.8.1 – российский репозиторий «Сизиф» как сборочная среда

Таблица 1.8.1 — Участие Базальт СПО в проектах разработки СПО

Проект	**Участие «Базальт СПО»**
glibc	Основная системная библиотека. Выпускающий последней международной версии – сотрудник «Базальт СПО»
Kernel.org	Ядро Linux. Вклад «Базальт СПО» – модуль LSM, контролирующий запуск скриптовых приложений; включение поддержки крипто, соответствующего российским ГОСТам
Харденинг (усиление безопасности)	участие в проектах Kernel, Glibc, libssl, GCC и др. В частности – в проекте Open Wall Linux (OWL) (среди ключевых участников проекта двое – сотрудники «Базальт СПО»).
Chroot (изолированное окружение для пакетов)	использование изолированного окружения (chroot) для пакетов, которые могут быть атакованы извне
обеспечение совместимости с MS Active Directory	«Базальт СПО» создала патчи масштабирования систем для Samba DC
защищенный терминальный доступ	участие в проекте libssl («Базальт СПО» - один из ключевых разработчиков)
Strace – отслеживание системных вызовов между процессом и операционной системой (ядром)	Базальт СПО» - один из основных разработчиков проекта. При проверках ПО для сертификации во ФСТЭК многие отечественные компании используют трассировщик Strace
Apt RPM – контроль цепочек зависимостей на основе RPM	«Базальт СПО» ведет проект
Xcat – управление узлами, используемыми в суперкомпьютерах	реализована возможность загрузки разнородных образов на разные узлы
Zabbix – система мониторинга	обеспечена возможность иерархического сбора информации
LSM (Linux Security Module)	создан дополнительный модуль

SE	«Базальт СПО» создал и развивает собственные политики
PVE - управление средой, в которой выполняются виртуальные окружения	собран специальный дистрибутив для разворачивания и управления виртуализацией
Технологии по сетевой ОС	создана специальная версия ОС для управления высокоскоростными пакетными процессорами коммутаторов маршрутизаторов (в частности, для сетевого устройства на базе Mellanox)
Alterator - модульная система управления	собственная разработка «Базальт СПО»

Репозиторий Сизиф

Репозиторий Сизиф представляет собой набор пакетов, для каждого из которых определен перечень зависимостей. Зависимость означает необходимость в определенной сущности для успешной установки пакета в систему. Таковые сущности могут быть различных типов:

пакет;

файл, находящийся в определенном месте файловой системы;

наличие символичесвого имени в библиотеке;

наличие ABI (application binary interface) с определенным наобром параметров (характеризуется определенным цифровым отпечатком).

Репозиторий Сизиф удовлетворяет требованиям ГОСТ Р 54593-2011 «Информационные технологии. Свободное программное обеспечение. Общие положения».

Репозиторий переводится в очередное состояние путем выполнения т.н. *сборочного задания*, которое представляет собой определенную операцию над исходными текстами ПО, спецификациями сборки, пакетами и еще много над чем. Сборочное задание может изменить сразу несколько пакетов. Сборочное задание будет принято и переведет репозиторий в новое состоняие, если после такого перехода количество неудовлетворенных зависимостей уменьшается. Такая политика управления репозиторием приводит в итоге к тому, что репозиторий замкнут относительно зависимостей.

Пакеты в репозиторий Сизиф собираются из исходных текстов членами Альт Линукс Team, в котоую входят как сотрудники Базальт СПО, так и другие разработчики. Вся информации о репозитории Сизиф является открытой, включая спецификации сборки, исходные тексты программ и вс прочую информацию , представляя собой , таким образом открытую платформу для

разработки ПО, коорое затем может быть включено в дистрибутивы Альт. Подробнее см packages.altlinux.org

На данном ресурсе представлены все пакеты по всем бранчам и платформам.

Важно отметить, что репозиторий Сизиф не зависит от других репозиториев (Red Hat / CentOS, Debian), то есть пакеты не копируются оттуда, а собираются из исходных текстов в своей собственной *сборочной системе*.

Это означает, что Базальт СПО может самостоятельно планировать развитие репозитория и соответственно своих продуктов в отношении функциональности, релизов и *целевых платформ*.

На базе репозитория Сизиф происходит формирование *бранчей*.

Бранчи

В определенный момент времени, который диктуется планами развития продуктов Базальт СПО в репозитории Сизиф выделяется определенное замкнутое по зависимостям подмножество пакетов и выводится в независимую от Сизифа ветвь – "*бранч*".

Бранчи также называются *платформами*, так как на их базе формируются *дистрибутивы* продуктов Базальт СПО. Дистрибутивы, которые на рынке сейчас и описаны ниже, базируются на 8 платформе. Сейчас готовится к выпуску девятая платформа, и соответственно будут выпущены дистрибутивы 9 версии.

Дистрибутив

Дистритбутив представляет собой операционную систему – ядро Linux, собранное в Базальт СПО, набор системного ПО и набор пакетов прикадного ПО, реализующий определенный набор функциональности, то есть определенный продукт.

В зависимости от применения продукта, выпускаются следующий набор:

- Альт Сервер
- Альт Рабочая станция
- Альт Образование (универсальный дистрибутив: может быть установлен как сервер, так и рабочая станция)
- Альт 8 СП – сертификаты ФСТЭК России и Минобороны. (универсальный дистрибутив: может быть установлен как сервер, так и рабочая станция)
- Simply Linux (дистрибутив для домашнего использования, не входит в Реестр российского ПО)

Бранч, на основе которого выпускается дистрибутив, становится репозиторием для данного дистрибутива (p8, p9 и так далее). Подробнее смотри раздел про Управление репозиториями .

1.8.10. Преимущества использования свободного программного обеспечения и открытой модели разработки

Развитие разработки и использования свободного программного обеспечения с открытым кодом позволяет в масштабах государства преодолеть технологическую зависимость от заимствованных решений и сформировать собственную отрасль создания программ для ЭВМ в России.

Для государства применение СПО дает возможность обеспечить высокий уровень информационной безопасности за счёт публичного доступа к исходному коду и его независимого аудита. Проприетарные приложения нередко содержат недокументированные функции, что является потенциальной угрозой. Проводимые рядом проприетарных компаний акции по доступу государственных аудиторов к своему исходному коду подобной безопасности не обеспечивают, так как не обеспечивают главного — гарантированной собираемости программного изделия именно из тех исходных текстов, которые были предоставлены аудиторам. СПО же дает возможность не только анализа текста программ, но и их модификации, самостоятельной сборки исполняемого кода из исходных текстов в заведомо надёжной и безопасной технологической среде, позволяют обеспечить принципиально иной уровень информационной безопасности.

В стране создается широкий ассортимент программ для ЭВМ высокого качества и одновременно значительно снижается количество нарушений в сфере авторского права.

СПО не требует лицензионных выплат за каждый установленный экземпляр программы. Модель применения ПО на таких условиях очень выгодна для рынка. Например, государство может провести открытый конкурс, однократно заплатить фирме-разработчику за поставку программного обеспечения и затем тиражировать его без ограничений. Таким образом, для обычных пользователей оно будет практически бесплатным. Кроме того, прозрачная схема платежей и отсутствие сверхприбылей у разработчиков СПО позволяет существенно снизить коррупционную составляющую на государственном рынке ИКТ.

Расширяется участие российских разработчиков в выполнении работ и оказании услуг для государственных и муниципальных нужд. Тем самым обеспечиваются дополнительные инвестиции в развитие отечественного производителя.

При разработке свободного ПО возможно повторное использование всего уже разработанного массива текстов свободных программ. Это значительно повышает производительность труда программиста и снижает себестоимость разработки. Именно конкуренция более эффективной модели разработки

заставила Билла Гейтса на выступлении в Малайзии говорить, что «Свободное ПО убивает рабочие места».

Будущие специалисты-разработчики могут учиться на чужом успешном опыте: свободно изучать документированный исходный код свободных программ для ЭВМ и модифицировать его, в том числе создавать на его базе собственные разработки. Те, кто уже сегодня хочет попробовать свои силы в разработке СПО, может принять участив в работе сообществ – например, российского сообщества Sisyphus (http://sisyphus.ru/ru/project/).

Существует обширная – и, что немаловажно – бесплатная литература об СПО. Описания технологий и программных продуктов можно найти в книгах – их PDF-версии доступны для скачивания (например, библиотека «Свободные книги о свободном ПО» https://www.altlinux.org/Books:Main_page). Многие из этих книг написаны российскими разработчиками СПО и преподавателями вузов.

Важным преимуществом СПО для разработчиков является возможность полноправно участвовать в международных проектах разработки свободного ПО и получать доступ к самым современным технологиям и разработкам.

Бизнес на базе СПО, в отличие от проприетарного программного обеспечения, ориентирован не на получение лицензионных отчислений, а на оказание услуг по разработке и поддержке, поэтому он хорошо вписывается в современные бизнес-модели, например «программное обеспечение как сервис» (SaaS). Именно доступность исходных кодов программ и возможность их модификации позволяет обеспечивать высокий уровень и гибкость сервисов.

Современные тенденции развития разработки информационных технологий характеризуются новыми архитектурными решениями, основная направленность которых — создание универсальных сервисных платформ, обеспечивающих взаимодействие разных типов пользователей, прежде всего производителей и потребителей какой-либо продукции или услуги.

Характерно, что такой сервис (платформа) как Uber не владеет ни одним автомобилем, но обеспечивает услуги найма такси в двухстах городах мира, Alibaba обеспечивает куплю-продажу более миллиарда наименований товаров, не имея отношения к их производству, Airbnb предлагает аренду жилых помещений более, чем в ста странах, не владея ни гостиницами, ни апартаментами.

Для создания подобных платформ необходим анализ возможностей различных сфер деятельности и выявление тех компонентов инфраструктуры, которые могут быть оформлены как общая платформа. Однако для реализации такого проекта требуется интеграция различных технических решений,

включая ПО, а также организация взаимодействия неопределённого круга лиц, в том числе разработчиков информационных технологий.

Создание современных информационных технологий, таким образом, всё более зависит не только и не столько от написания программного кода, сколько от разработки и внедрения спецификаций разного уровня: архитектурных решений, «профилей», планирования процессов и формирования «сервисов».

Ограничения, накладываемые проприетарными лицензиями, как правило, являются непреодолимыми препятствиями для интеграции.

Так, показательным примером «платформ» являются репозитории — технологические платформы свободного ПО, в том числе российский репозиторий Сизиф. Это сложные сервисы для обеспечения жизненного цикла ПО. Их развитие вне открытой модели разработки принципиально невозможны.

Необходимо отметить, что кроме технического и технологического значения открытая модель разработки глубоко интегрирована с современным пониманием создания и развития сложных систем.

В рамках современного системного подхода общепринятой точкой зрения является понимание того, что сложная система (например, уровня системной сложности — с самоорганизацией) должна быть открытой, если подразумевается её выживание и развитие.

Открытые системы — это системы, которые обмениваются с окружающей средой веществом, энергией и информацией.

Кроме того, начиная с определённого уровня системной сложности, дальнейшее развитие системы (проекта, сообщества, платформы и т.п.) возможно только посредством целенаправленного создания и развития коллективного субъекта познавательной деятельности. Это обусловлено, в том числе и очевидным ограничением возможностей отдельного человека или группы лиц. Такое масштабное взаимодействие, как правило неопределённого круга лиц, — также принципиально невозможны вне открытой модели разработки.

Таким образом, открытая модель разработки информационных технологий — это стратегический выбор, обусловленный глубокими методологическими причинами. При этом, и сама открытая модель разработки, её принципы, подходы, технологии, методологические основания — должна совершенствоваться.

Слабые стороны открытой модели разработки правильнее назвать слабыми с точки зрения современных краткосрочных интересов бизнеса: трудность с организацией технической поддержки продукции, более сложная схема капитализации и организации бизнеса.

Однако всегда могут быть исключения: для каких-то отдельных проектов, как правило узкоспециализированных, небольших с точки зрения трудоёмкости или с коротким по времени жизненным циклом, проприетарная модель разработки может оказаться организационно и технически более эффективной. Это те самые исключения, которые как говорят — подтверждают правило: всем очевидно — это исключение, обусловленное особыми обстоятельствами или характеристиками проекта.

1.8.11. Процессы разработки и внедрения новых сервисов и протоколов

Как было отмечено выше, в рамках создания современных информационных технологий повышается значимость разработки всех видов спецификаций: не только программного кода, но и различного уровня общности структурных и поведенческих моделей разрабатываемых систем.

Деятельность проектировщиков, разработчиков всегда была связана с какими-либо методиками проектирования. Особенно интенсивно они начали развиваться в 70-80-х годах XX века.

Так, например, в рамках советской космической программы «Буран» с 1986 года разрабатывался язык проектирования «Дракон». Как известно, космический аппарат многоразового использования «Буран» был уникален по своим техническим характеристикам и, в том числе, был способен совершать посадку в полностью автоматическом режиме.

Язык проектирования «Дракон» ориентирован преимущественно на разработку, так называемых, поведенческих моделей системы.

В настоящее время после многолетней интеграции и переработки различных языков и идей в области проектирования наиболее сбалансированными можно считать язык UML/SysML и семантически эквивалентные ему нотации IDEF.

Эти языки, в отличие от языков программирования, не являются полностью формализованными: при реализации разработанных с их помощью моделей в большинстве случаев происходит и потеря информации, и её модификация или искажение. Синтаксис этих языков не является строгим и полным.

Язык проектирования представляет собой просто взаимосвязанный набор концепций и понятий, посредством которых можно описать различные модели систем. Этот набор является результатом длительного обобщения опыта и идей проектирования. Так, базовыми понятиями проектирования систем является нескольких десятков элементов моделей (класс, объект, компонент, подсистема, кооперация, интерфейс, состояние, сигнал, сообщение, событие и т. д.), их связей («род-вид», зависимость, реализация, ассоциация), а также обобщенная

классификация самих моделей. Своеобразная сложность языков проектирования заключается в основном в толковании перечисленных абстрактных понятий.

Для примера, иерархия моделей (диаграмм) актуальной версии языка UML изображена на рисунке 1.8.2.

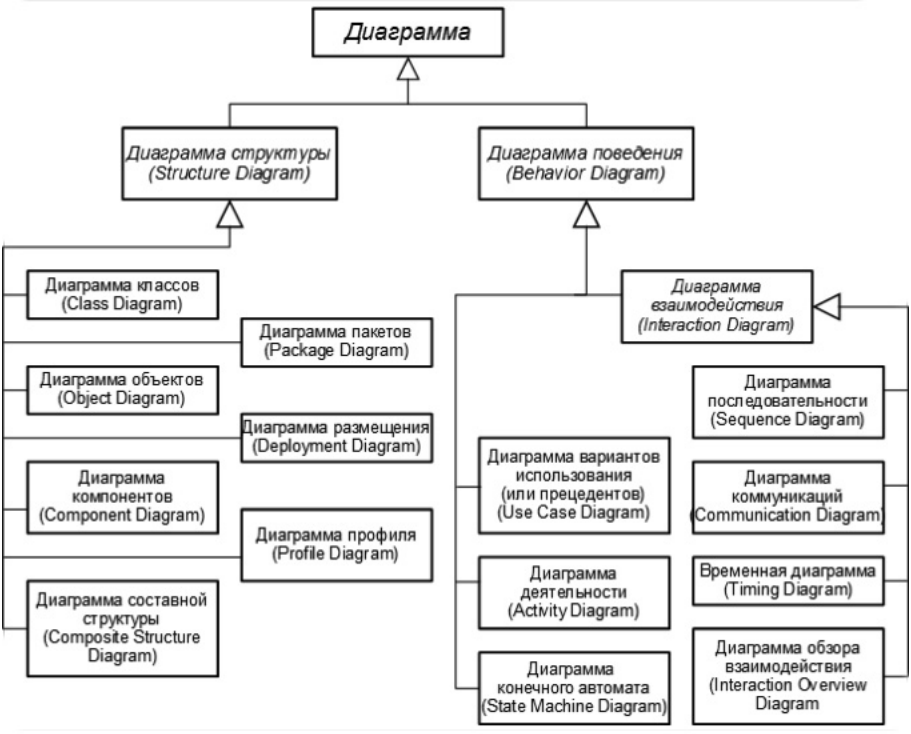

Рисунок 1.8.2 - Классификация моделей систем, применяемая в языке UML

С начала двухтысячных в связи с тематикой проектирования часто можно встретить такое понятие как «сервис-ориентированная архитектура» (SOA или СОА). Известны также такие методики как «общая архитектура для беспилотных систем» (Joint Architecture for Unmanned Systems – JAUS), «язык анализа и разработки архитектуры» (Architecture Analysis & Design Language – AADL), применяемый для разработки авионики.

По своему смыслу перечисленные методологии проектирования являются обобщением опыта моделирования определённых типов систем и библиотеками шаблонов моделей и типовых подходов, основанных на одних и тех же базовых идеях.

В основе СОА две идеи: полнятие интеграционная шина (ESB), то есть компонент-посредник, и понятие «сервисы».

Термин «СОА» впервые появился в 1996 году в статье вице-президентов «Гартнер» Роя Шульте и Ефима Натиса «Сервис-ориентированные архитектуры. Часть 1». Статья содержала определение СОА как стиля многозвенных вычислений и описание преимуществ сервис-ориентированных конфигураций. В 2007 году Томас Эрль издал книгу «Шаблоны проектирования СОА».

На основе этой работы крупнейшие корпорации (IBM, Microsoft, Oracle, Red Hat и другие) разрабатывают свои методологии, которые со временем всё более отличаются. То есть, крупнейшие ИТ-корпорации заинтересованы в основном в защите своих разработок посредством их стандартизации на уровне международных стандартов, но никак не в обобщении опыта.

Архитектура JAUS в основном предлагает обеспечить взаимодействие между компонентами через иерархическую организацию адресного пространства: идентификатор подсистемы – идентификатор узла в подсистеме – идентификатор компонента в узле, а также создание специальных компонентов-менеджеров (аналог ESB), обеспечивающих передачу сообщений между остальными компонентами и минимизирующих количество связей между ними.

Таким образом, в сфере проектирования существует большое количество разнообразной маркетинговой информации. Методологически все наработки сводятся к обобщенным в рамках UML элементам, связям и типам моделей.

Именно они являются базой для проектирования новых сервисов и различного рода спецификаций, которые затем реализуются средствами разработки, становятся кодом компьютерных программ, новыми протоколами, стандартами, инструкциями для персонала, руководствами по эксплуатации информационных систем и пр.

2. Протоколы сети Интернет

2.1. Модель взаимодействия открытых систем и стек протоколов TCP/IP

Международная организация стандартизации (International Standards Organization – ISO) разработала эталонную модель взаимодействия открытых систем (Open System Interconnection reference model – OSI), которая определяет концепцию и методологию создания сетей передачи данных. Модель описывает стандартные правила функционирования устройств и программных средств, при обмене данными между узлами (компьютерами) в открытой системе. Модель ISO/OSI включает семь уровней. На рисунке 2.1.1 показана модель взаимодействия двух устройств: узла источника (source) и узла назначения (destination). Совокупность правил, по которым происходит обмен данными между программно-аппаратными средствами, находящимися на одном уровне, называется протоколом. Набор протоколов называется стеком протоколов и задается определенным стандартом. Взаимодействие между уровнями определяется стандартными интерфейсами.

Рисунок 2.1.1 – Семиуровневая модель ISO/OSI

Взаимодействие соответствующих уровней является виртуальным, за исключением физического уровня, на котором происходит обмен данными по кабелям, соединяющим компьютеры. На рисунке 2.1.1 приведено соответствие протоколов стека TCP/IP уровням модели OSI. Взаимодействие уровней между собой внутри узла происходит через межуровневый интерфейс, каждый нижний уровень предоставляет услуги следующему верхнему уровню.

Виртуальный обмен между соответствующими уровнями узлов А и В происходит определенными единицами информации (рисунок 2.1.2). На трех верхних уровнях – это сообщения или данные (Data), на транспортном уровне – сегменты (Segment), на сетевом уровне – пакеты (Packet), на канальном уровне – кадры (Frame) и на физическом – последовательность битов.

Для каждой сетевой технологии существуют свои протоколы и технические средства, часть из которых имеет условные обозначения, приведенные на рисунках 2.1.1 и 2.1.2. Среди технических средств физического уровня следует отметить кабели, разъемы, повторители сигналов (repeater), многопортовые повторители или концентраторы (hub), преобразователи среды (transceiver), например, преобразователи электрических сигналов в оптические и наоборот. На канальном уровне – это мосты (bridge), коммутаторы (switch) и сетевые карты или адаптеры (Network Interface Card – NIC). На сетевом уровне – маршрутизаторы (router).

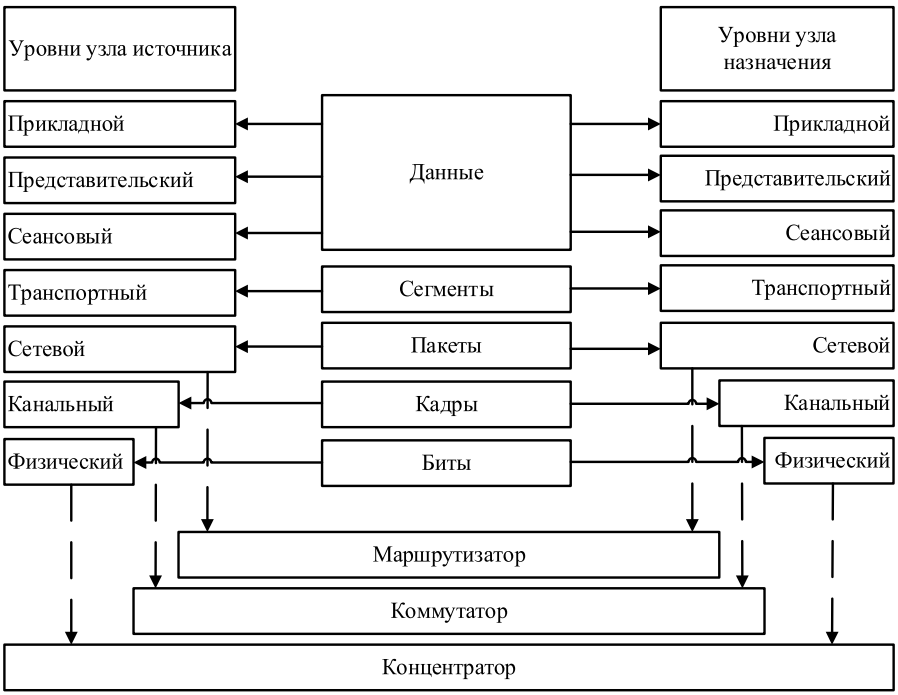

Рисунок 2.1.2 – Устройства и единицы информации соответствующих уровней

При передаче данных от источника к узлу назначения, подготовленные на прикладном уровне передаваемые данные, последовательно проходят от самого верхнего, Прикладного уровня 7 узла источника информации до самого

нижнего – Физического уровня 1, затем передаются по физической среде узлу назначения, где последовательно проходят от нижнего уровня 1 до уровня 7.

Самый верхний, Прикладной уровень (Application Layer) 7 оперирует наиболее общей единицей данных – сообщением. На этом уровне реализуется управление сетевыми службами, такими, как FTP, TFTP, HTTP, SMTP, SNMP.

Представительский уровень (Presentation Layer) 6 изменяет форму представления данных. Например, передаваемые с уровня 7 данные преобразуются в общепринятый формат ASCII. При приеме данных происходит обратный процесс. На уровне 6 также происходит шифрование и сжатие данных.

Сеансовый уровень (Session Layer) 5 устанавливает сеанс связи двух конечных узлов (компьютеров), определяет, какой компьютер является передатчиком, а какой приемником.

Транспортный уровень (Transport Layer) 4 делит большое сообщение узла источника информации на части, при этом добавляет заголовок и формирует сегменты определенного объема, а короткие сообщения может объединять в один сегмент. В узле назначения происходит обратный процесс. В заголовке сегмента задаются номера порта источника и назначения, которые адресуют службы верхнего прикладного уровня для обработки данного сегмента. Кроме того, транспортный уровень обеспечивает надежную доставку пакетов. При обнаружении потерь и ошибок на этом уровне формируется запрос повторной передачи, при этом используется протокол TCP. Когда необходимость проверки правильности доставленного сообщения отсутствует, то используется более простой и быстрый протокол дейтаграмм пользователя UDP.

Сетевой уровень (Network Layer) 3 адресует сообщение, задавая единице передаваемых данных (пакету) логические сетевые адреса узла назначения и узла источника (IP-адреса), определяет маршрут, по которому будет отправлен пакет данных, транслирует логические сетевые адреса в физические, а на приемной стороне – физические адреса в логические. Сетевые логические адреса принадлежат пользователям.

Канальный уровень (Data Link) 2 формирует из пакетов кадры данных (frames). На этом уровне задаются физические адреса устройства-отправителя и устройства-получателя данных. Например, физический адрес устройства может быть прописан в ПЗУ сетевой карты компьютера. На этом же уровне к передаваемым данным добавляется контрольная сумма, определяемая с помощью алгоритма циклического кода. На приемной стороне по контрольной сумме определяют и по возможности исправляют ошибки.

Физический уровень (Physical) 1 осуществляет передачу потока битов по соответствующей физической среде (электрический или оптический кабель,

радиоканал) через соответствующий интерфейс. На этом уровне производится кодирование данных, синхронизация передаваемых битов информации.

Протоколы трех верхних уровней являются сетенезависимыми, три нижних уровня являются сетезависимыми. Связь между тремя верхними и тремя нижними уровнями происходит на транспортном уровне.

Важным процессом при передаче данных является инкапсуляция (encapsulation) данных. Передаваемое сообщение, сформированное приложением, проходит три верхних сетенезависимых уровня и поступает на транспортный уровень, где делится на части, и каждая часть инкапсулируется (помещается) в сегмент данных (рисунок 2.1.3). В заголовке сегмента содержится номер протокола прикладного уровня, с помощью которого подготовлено сообщение, и номер протокола, который будет обрабатывать данный сегмент.

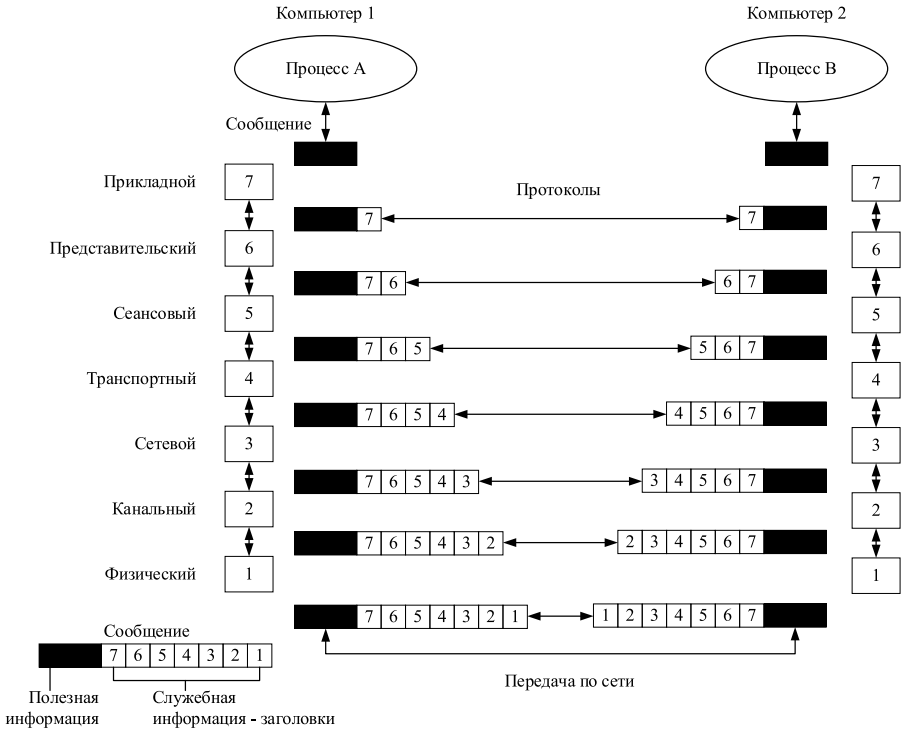

Рисунок 2.1.3 – Инкапсуляция данных

На сетевом уровне сегмент инкапсулируется в пакет данных, заголовок (header) которого содержит, сетевые (логические) адреса отправителя

информации (источника) – Source Address (SA) и получателя (назначения) – Destination Address (DA). В стеке протоколов TCP/IP это IP-адреса.

На канальном уровне пакет инкапсулируется в кадр или фрейм данных, заголовок которого содержит физические адреса узла передатчика и приемника, а также другую информацию. Кроме того, на этом уровне добавляется трейлер (концевик) кадра, содержащий информацию, необходимую для проверки правильности принятой информации. Таким образом, происходит обрамление данных заголовками со служебной информацией, т. е. инкапсуляция данных.

Название информационных единиц на каждом уровне, их размер и другие параметры инкапсуляции задаются согласно протоколу единиц данных (Protocol Data Unit – PDU). Итак, на трех верхних уровнях – это сообщение (Data), на Транспортном уровне 4 – сегмент (Segment), на Сетевом уровне 3 – пакет (Packet), на Канальном уровне 2 – кадр (Frame), на Физическом уровне 1 – последовательность бит.

Вместе с семиуровневой моделью OSI на практике применяется четырехуровневая модель стека протоколов TCP/IP (рисунок 2.1.4).

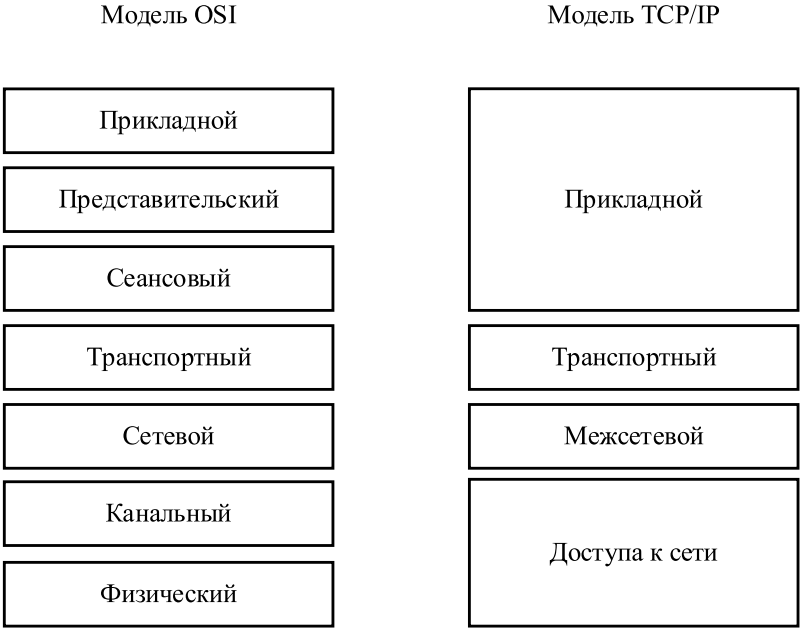

Рисунок 2.1.4 – Модели OSI и TCP/IP

Прикладной уровень модели TCP/IP по названию совпадает с названием модели OSI, но по функциям гораздо шире, поскольку охватывает три верхних сетенезависимых уровня (прикладной, представительский и сеансовый). Транспортный уровень обеих моделей и по названию, и по функциям одинаков. Сетевой уровень модели OSI соответствует межсетевому (Internet) уровню модели TCP/IP, а два нижних уровня (канальный и физический) представлены объединенным уровнем доступа к сети (Network Access).

В таблице 2.1.1 приведены сведения об основной информации, добавляемой в заголовки сообщений на разных уровнях OSI-модели.

Таблица 2.1.1 – Основная информация в заголовках сообщений

Физический уровень	Канальный уровень	Сетевой уровень	Транспортный уровень	Верхние уровни
Частотно-временные параметры и синхронизация	Физические адреса источника и назначения	Логические адреса источника и назначения	Номера порта источника и назначения	Сопряжение пользователей с сетью

На транспортном уровне в заголовке сегмента задаются номера портов приложений источника и назначения. Номера портов адресуют приложения или сервисы прикладного уровня, которые создавали сообщение и будут его обрабатывать на приемной стороне. Например, сервер электронной почты с номерами портов 25 и 110 позволяет посылать e-mail сообщения и принимать их, номер порта 80 адресует веб-сервер.

Для обмена сообщениями помимо номеров портов на сетевом уровне в заголовке пакета необходимо задать логические адреса источника и назначения. К логическим адресам относятся IP-адреса пользователей. В документации, используемой в настоящее время, версии IPv4 адреса IP отображаются в десятичной форме в виде четырех групп чисел. Каждая группа может содержать числа от 0 до 255. Группы разделены между собой точками, например, 192.168.10.21, 172.16.250.17, 10.1.10.122.

В дополнение к логическим адресам на канальном уровне в заголовке кадра задаются физические адреса устройства-источника и устройства-назначения. Наиболее широко распространенной сетевой технологией канального уровня в настоящее время является Ethernet или ее модификации (Fast Ethernet, Gigabit Ethernet, 10Gigabit Ethernet). При этом в качестве физических адресов используются MAC-адреса (Media Access Control). В документации MAC-адреса представлены в виде 12 шестнадцатеричных чисел, например, 00-05-A8-69-CD-F1. Тот же адрес может быть представлен и в

несколько другой форме 00:05:A8:69:CD:F1 или 0005.A869.CDF1. MAC-адреса компьютеров прошиты в ПЗУ сетевой карты.

На трех нижних уровнях модели OSI функционируют аппаратно-программные средства, передающие сообщения с высокой скоростью. Сообщение от одного конечного узла до другого проходит через промежуточные устройства, маршрутизаторы и коммутаторы (Рисунок 2.1.5), на которых обрабатывается средствами трех или двух нижних уровней (Рисунок 2.1.2). Транспортный уровень, обеспечивающий надежность передачи данных, функционирует только на конечных узлах сети (рисунок 2.1.5).

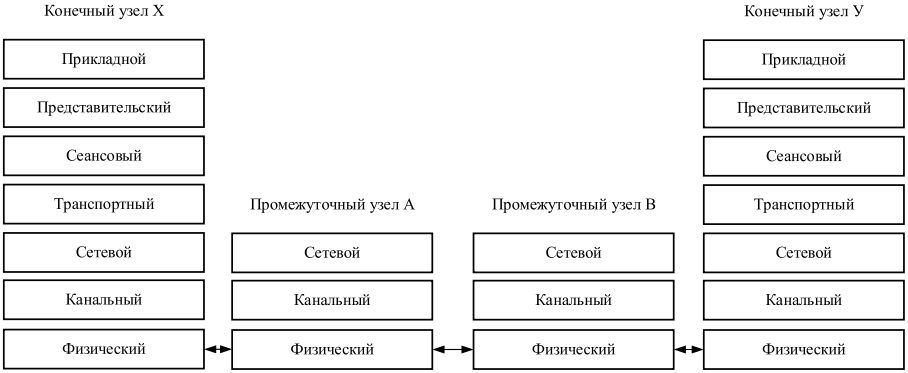

Рисунок 2.1.5 – Передача сообщения по сети

2.2. Стандарты Ethernet

2.2.1. Протокол Ethernet

Ethernet (от англ. ether “эфир”) – пакетная технология передачи данных преимущественно локальных компьютерных сетей, разработанная Робертом Метклафом в компании Xerox. Ethernet стал самой распространённой технологией ЛВС, вытеснив такие устаревшие технологии, как Arcnet, Token ring и FDDI.

Технология Ethernet разделяется на две совершенно разные технологии: классический Ethernet и коммутируемый Ethernet.

Классический Ethernet – изначально использовавшийся вариант, разрабатывался для работы в сетях с общей средой передачи данных и использовал топологию типа общая шина (Bus). Эта технологии использовалась с первого варианта Ethernet, имеющего скорость передачи 10 Мбит/с, и поддерживалась до стандарта Gigabit Ethernet.

В технологии Ethernet, работающей на разделяемой среде, одновременно только одно устройство могло осуществлять передачу, так как если два и более устройств начинали передавать свои данные по общей шине, то возникала коллизия, не позволяющая сети нормально функционировать. По этим причинам была реализована технология, которая управляла процессом передачи данных – CSMA/CD (Carrier Sense Multiple Access with Collision Detection) множественный доступ с контролем несущей и обнаружением коллизий.

Перед передачей данных устройство в сети Ethernet прослушивало среду передачи на наличие несущей, если линия была свободна, то передача начиналась. Передающие узлы, обнаружив коллизию, прекращали передачу данных, после чего повторную попытку передачи делали через случайный интервал времени (каждый через свой) после освобождения линии.

Коммутируемый Ethernet – вариант технологии, применяющийся в настоящее время и использующий топологию сетей типа точка-точка. Данная технология реализуется благодаря появлению в сетях коммутаторов. Коммутатор использует адресную информацию в заголовке кадра (MAC-адрес) и организует независимые виртуальные каналы между портами, к которым подключена пара узлов, образующая соединение точка-точка.

Благодаря коммутаторам Ethernet пропала необходимость в технологии CSMA/CD, так как для каждого устройства выделяется свой канал связи, а также появилась возможность использовать дуплексный режим передачи.

В модели взаимодействия открытых систем OSI, Ethernet функционирует на физическом и канальном уровне. Причем канальный уровень разделен на 2 подуровня: LLC (Logical Link Control) – подуровень логической передачи данных и MAC (Media Access Control) – подуровень управления доступом к среде.

На физическом уровне технология Ethernet определяет правила передачи сигналов по трем разным типам кабелей: коаксиальному кабелю, витой паре и оптоволокну. Канальный уровень модели описывает формат кадров и протоколы управления доступом к середе.

На канальном уровне для передачи данных используются кадры Ethernet. В Ethernet существует два основных формата кадров:

Ethernet II (Ethernet DIX) – фирменный стандарт Ethernet компаний DEC, Intel и Xerox. Кадр Ethernet II не отражает разделения канального уровня Ethernet на подуровни LLC и MAC, его поля поддерживают функции обоих уровней;

IEEE 802.3/LLC – юридический стандарт Ethernet. Построен в соответствии с принятым разбиением функций канального уровня на

подуровни MAC и LLC. Поэтому результирующий кадр является вложением кадра LLC, определяемого стандартом 802.2, в кадр MAC, определяемый стандартом 802.3.

7 Байт	1 Байт	6 Байт	6 Байт	2 Байта	46 – 1500 Байт	4 Байта
Preamble	SFD	Destination Address (DA)	Source Address (SA)	Type	Data	FCS

Рисунок 2.2.1 – Формат кадра Ethernet II

Preamble – Поле преамбулы, используется для того, чтобы дать время и возможность приемопередатчикам войти в устойчивый синхронизм с принимаемыми сигналами. Каждый байт содержит одну и ту же последовательность битов – 10101010;

SFD – Начальный ограничитель кадра (Start of Frame Deimiter) состоит из одного байта с набором битов 10101011. Появление этой комбинации является указанием на предстоящий прием кадра;

DA – Адрес получателя (физический адрес сетевой карты - MAC-адрес получателя). Первый бит адреса получателя – это признак того, является адрес индивидуальным (unicast) или групповым (multicast): «0» – адрес указывает на индивидуальный адрес, «1» – это групповой адрес нескольких станций сети. При широковещательной адресации все биты поля адреса устанавливаются в 1 (MAC-адрес FF:FF:FF:FF:FF:FF);

SA – Адрес отправителя, 6-ти байтовое поле, содержащее MAC-адрес отправителя. Первый бит всегда имеет значение 0. В первых трех байтах MAC-адреса содержится код производителя сетевого адаптера, присвоенный IEEE. В остальных трех байтах – адрес собственно устройства (сетевой карты);

Type – Тип пакеты, в данном поле содержится шестнадцатеричный код о типе протокола вышестоящего уровня. Значения для некоторых распространенных сетевых протоколов: 0x0800 для IP, 0x0806, 0x86DD – для IPv6;

Data – Поле передаваемых данных, содержит данные кадра. Чаще всего это информация, нужная протоколам верхнего уровня. Данное поле не имеет фиксированной длины. Если длина пользовательских данных меньше 46 байт, то это поле дополняется до минимального размера байтами заполнения. Эта операция требуется для корректной работы метода доступа Ethernet;

FCS – Поле контрольной последовательности (Frame Check Sequence) содержит значение, которое вычисляется по определенному алгоритму в процессе кодирования содержимого кадра помехоустойчивым циклическим

кодом. После получения кадра принимающая станция выполняет собственное вычисление контрольной последовательности для этого кадра, сравнивает полученное значение с принятым значением поля FCS и, таким образом, определяет, не искажен ли полученный кадр.

Как уже отмечалось, кадр Ethernet II не отражает разделения канального уровня Ethernet на уровни MAC и LLC: его поля поддерживают функции обоих уровней, например, интерфейсные функции поля Тип относятся к функциям уровня LLC, в то время как все остальные поля поддерживают функции уровня MAC.

2.2.2. Протокол Fast Ethernet

Технология Fast Ethernet (100 Мбит/с) – это развитие технологии Ethernet, работающей на скорости 10 Мбит/с. Особенности построения и требования к Fast Ethernet описаны в стандарте IEEE 802.3u, который является дополнением к существующему стандарту 802.3. Уровни MAC и LLC технологии Fast Ethernet, т. е. структура кадров и доступ к среде передачи, остались неизменными, все отличия касаются только физического уровня технологии Fast Ethernet.

Все времена передачи кадров Fast Ethernet в 10 раз меньше соответствующих времен технологии 10 Мбит/с Ethernet: межбитовый интервал составляет 10 нс вместо 100 нс, а межкадровый интервал - 0.96 мкс вместо 9.6 мкс.

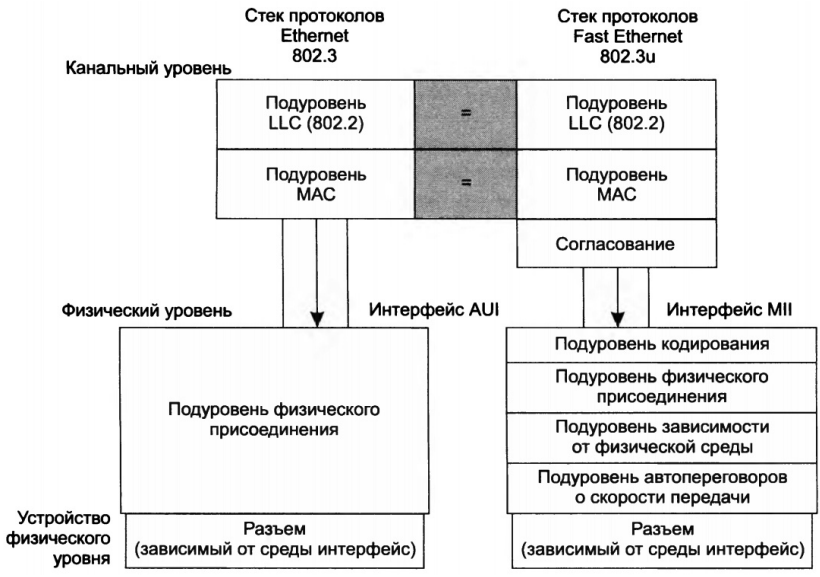

Рисунок 2.2.2 – Отличия технологий Ethernet и Fast Ethernet.

Организация физического уровня технологии Fast Ethernet является модульной. Это объясняется тем, что технология Fast Ethernet изначально была рассчитана на применение различных типов физической среды и кодирования, модульность физического уровня позволяет достичь этой цели достаточно легко. Различные же варианты физической среды Ethernet 10 Мбит/с разрабатывались разными организациями в разное время, отсюда и отсутствие гибкости в построении физического уровня. Модульный подход был впоследствии применен и во всех других более скоростных вариантах Ethernet, включая 100G Ethernet.

Физический уровень Fast Ethernet состоит из трех модулей:

Независимый от среды интерфейс (Media Independent Interface, MII). Этот интерфейс поддерживает независимый от физической среды способ обмена данными между подуровнями MAC и PHY.

Модуль согласования (Reconciliation) нужен для того, чтобы уровень MAC, рассчитанный ранее на интерфейс AUI, мог работать с физическим уровнем через интерфейс MII.

Устройство физического уровня (Physical Layer Device, PHY) состоит из нескольких подуровней:

Подуровня кодирования данных (Physical Coding Sublayer, PCS), преобразующего поступающие от уровня MAC байты в символы логического кода, например, 4B/5B;

Подуровней физического присоединения (Physical Media Attachment, PMA) и зависимости от физической среды (Physical Media Dependent, PMD), которые обеспечивают формирование сигналов в соответствии с методом физического кодирования, например, NRZI;

Подуровня автопереговоров, который позволяет двум взаимодействующим портам автоматически выбрать наиболее эффективный режим работы, например, полудуплексный или дуплексный.

Схема автопереговоров позволяет двум физически соединенным устройствам, которые поддерживают несколько стандартов физического уровня, отличающихся битовой скоростью и количеством витых пар, согласовать наиболее выгодный режим работы. Обычно процедура автопереговоров происходит при подсоединении сетевого адаптера, который может работать на скоростях 10 и 100 Мбит/с, к концентратору или коммутатору.

В технологии Fast Ethernet признаком свободного состояния среды является передача по ней символа простоя источника Idle (11111) – соответствующего избыточного кода (а не отсутствие сигналов, как в стандартах Ethernet 10 Мбит/с). Для отделения кадра Ethernet от символов

простоя источника используется комбинация символов начального ограничителя кадра — пара символов J (11000) и K (10001) кода 4B/5B, а после завершения кадра перед первым символом простоя источника вставляется символ T.

Рисунок 2.2.3 – Формат кадра Fast Ethernet

Fast Ethernet поддерживает три варианта физической среды:

- Волоконно-оптический многомодовый кабель (два волокна);
- Витая пара категории 5 (две пары);
- Витая пара категории 3 (четыре пары).

Официальный стандарт 802.3u установил три различные спецификации для физического уровня Fast Ethernet и дал им следующие названия: 100BASE-TX, 100BASE-T4 и 100BASE-FX. Все версии обладают одинаковой скоростью передачи – 100 Мбит/с, но используют разную среду передачи. Стандарты представлены в таблице 2.2.1.

Таблица 2.2.1– Различные спецификации для физического уровня Fast Ethernet

Стандарт	Тип кабеля	Максимальная длина сегмента	Метод кодирования
100Base-TX	Cat 5 UTP	100 метров	4B/5B+MLT-3
100Base-FX	Многомодовое оптоволокно 62.2 мкм	412 метров (полудуплекс) 2 км (полный дуплекс)	4B/5B+NRZI
100Base-T4	Cat 3 UTP (устаревшая спецификация)	100 метров	8B/6T

2.3. Протокол Gigabit Ethernet

В данной технологии, так же как в Fast Ethernet, была сохранена преемственность с технологией Ethernet: практически не изменились форматы кадров, сохранился метод доступа CSMA/CD в полудуплексном режиме. На логическом уровне используется кодирование 8B/10B.

Для поддержания различных физических сред физический уровень Gigabit Ethernet имеет такую же модульную структуру, как и физический уровень Fast

Ethernet, с тем отличием, что вместо интерфейса МИ в нем применяется интерфейс GMII (Gigabit МП), работающий на скорости 1 Гбит/с.

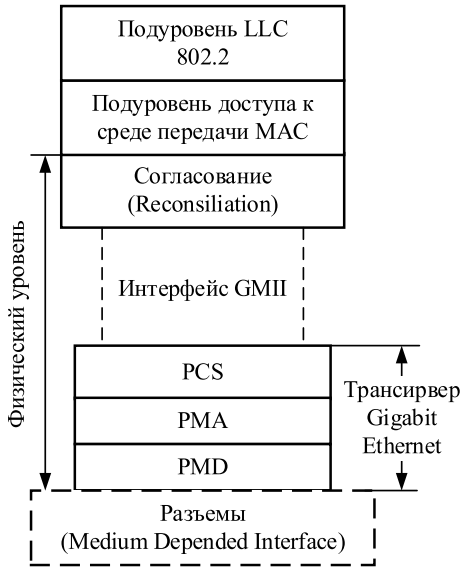

Рисунок 2.2.4 – Структура уровней стандарта Gigabit Ethernet

Поскольку скорость передачи увеличилась в 10 раз по сравнению с Fast Ethernet, то было необходимо либо уменьшить диаметр сети до 20 – 25 м, либо увеличить минимальную длину кадра. В технологии Gigabit Ethernet пошли по второму пути, увеличив минимальную длину кадра до 512 байт, вместо 64 байт в технологии Ethernet и Fast Ethernet. Диаметр сети остался равным 200 м, так же как в Fast Ethernet.

Для увеличения длины кадра до величины, требуемой в новой технологии, поле данных дополняется до длины 512 байт так называемым полем расширения носителя (Carrier Extension) размером 448 байт (512-64), представляющим собой поле, заполненное нулями. Формально минимальный размер кадра не изменился, он по-прежнему равняется 64 байт, и объясняется это тем, что поле расширения помещается после поля контрольной суммы кадра (FCS). Соответственно значение этого поля не включается в контрольную сумму и не учитывается при указании длины поля данных в поле длины. Поле расширения является просто расширением сигнала несущей частоты, необходимым для корректного обнаружения коллизий. Если размер кадра равен или превосходит 512 байт, то поле расширения носителя отсутствует.

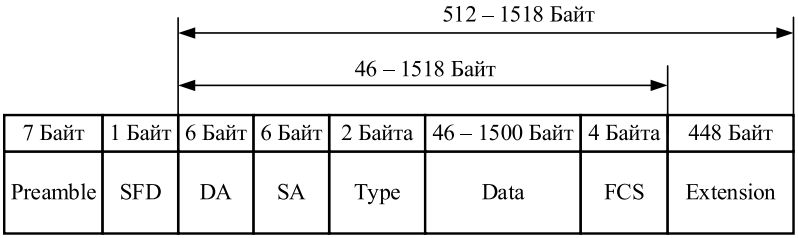

Рисунок 2.2.5 – Формат кадра Gigabit Ethernet

Современные сети Gigabit Ethernet, как правило, строятся на основе коммутаторов и работают в полнодуплексном режиме. В этом случае говорят не о диаметре сети, а о длине сегмента, которая определяется физической средой передачи данных. Gigabit Ethernet предусматривает использование следующих стандартов, представленных в таблице 2.2.2.

Для передачи данных по многомодовому волоконно-оптическому кабелю стандарт предписывает применение излучателей, работающих на двух длинах волн: 850 и 1300 нм. Применение светодиодов с длиной волны 850 нм объясняется тем, что они намного дешевле, чем светодиоды, работающие на волне 1300 нм, хотя при этом максимальная длина кабеля уменьшается, так как затухание многомодового оптоволокна на волне 850 нм более чем в два раза выше, чем на волне 1300 нм.

Таблица 2.2.2 – Стандарты использования Gigabit Ethernet

Стандарт	Тип кабеля	Максимальная длина сегмента	Метод кодирования
1000Base-SX (802.3z) Лазерный диод 850 нм	Многомодовое оптоволокно (50 мкм)	500 метров	8B/10B+NRZI
	Многомодовое оптоволокно (62.5) мкм	220 метров	
1000Base-LX (802.3z) Лазерный диод 1300 нм	Одномодовое оптоволокно (9 мкм)	5 километров	
	Многомодовое оптоволокно (50 мкм)	500 метров	
	Многомодовое оптоволокно (62.5) мкм	400 метров	
1000BASE-T (802.3ab)	Cat 5 UTP 4 неэкранированные витые пары	100 метров	4D/PAM-5

Для многомодового оптоволокна стандарт Gigabit Ethernet определяет спецификации 1000Base-SX и 1000Base-LX. В первом случае используется длина волны 850 нм (S означает Short Wavelength — короткая длина волны), а во втором — 1300 нм (L означает Long Wavelength — длинная длина волны).

Для спецификации 1000Base-LX в качестве источника излучения всегда применяется полупроводниковый лазерный диод с длиной волны 1300 нм. Спецификация 1000Base-LX позволяет работать как с многомодовым, так и с одномодовым кабелем. В спецификациях 1000Base-SX и 1000Base-LX подуровень кодирования преобразует байты уровня MAC в коды 8B/10B (а не 4B/5B, как в стандарте Fast Ethernet).

Для использования уже имеющихся симметричных кабелей UTP категории 5 был разработан стандарт 802.3ab. Поскольку в технологии Gigabit Ethernet данные должны передаваться со скоростью 1000 Мбит/с, а витая пара 5-й категории имеет полосу пропускания 100 МГц, было решено передавать данные параллельно по 4 витым парам и задействовать UTP категории 5 или 5e с шириной полосы более 125 МГц. Таким образом, по каждой витой паре необходимо передавать данные со скоростью 250 Мбит/с, что в 2 раза превышает возможности UTP категории 5e. Для устранения этого противоречия используется код 4D-PAM5 с пятью уровнями потенциала (-2, -1, 0, +1, +2). По каждой паре проводов одновременно производится передача и прием данных со скоростью 125 Мбит/с в каждую сторону. При этом происходит постоянная коллизия, при которой формируются сигналы сложной формы пяти уровней. Разделение входного и выходного потоков производится за счет использования схем гибридной развязки. В качестве таких схем применяются сигнальные процессоры. Для выделения принимаемого сигнала приемник вычитает из суммарного (передаваемого и принимаемого) сигнала собственный передаваемый сигнал.

2.3. Интернет-протокол версии 4

Адреса (IP-адреса) интернет протокола версии 4 (IPv4) являются основным типом адресов используемых на сетевом уровне модели OSI для передачи пакетов между сетями. IP-адреса состоят из четырех байт.

Присвоение IP-адресов хостам осуществляется:

- вручную, настраивается администратором вычислительной сети;
- автоматически, с использование специальных протоколов (протокол DHCP - Dynamic Host Configuration Protocol, протокол динамической настройки хостов).

Протокол IPv4 работает на межсетевом уровне стека протокола TCP/IP и на сетевом уровне OSI. Основной задачей протокола является осуществление передачи блоков данных от хоста-отправителя, до хоста-назначения, где отправителями и получателями выступают вычислительные машины, однозначно идентифицируемые адресами IP-адресами. IP-протокол осуществляет, в случае необходимости, фрагментацию отправляемых дейтаграмм для передачи данных через сети с меньшим максимальным размером пакетов.

Протокол IP функционирует без установления соединения, нет подтверждения доставки пакетов, нет обмена сообщениями с узлом назначения о готовности к приему пакетов, не осуществляется контроль корректности полученных данных с помощью контрольной суммы.

Протокол IP отправляет и обрабатывает каждую дейтаграмму как независимую порцию данных.

После отправки дейтаграммы протоколом IP в сеть, дальнейшие действия с этой дейтаграммой никак не контролируются отправителем. Если дейтаграмма, по каким-либо причинам, не может быть передана дальше по сети, то она уничтожается. Уничтоживший дейтаграмму узел, имеет возможность сообщить о этом событии по адресу отправителя с помощью протокола ICMP. Надежная доставка данных обеспечивается протоколом транспортного уровня TCP.

На сетевом уровне модели OSI работают маршрутизаторы. Основная задача протокола IP – это маршрутизация дейтаграмм, определение оптимального пути следования дейтаграмм от узла-отправителя к узлу-получателю на основе IP адреса.

2.3.1. Формат заголовка IP

Структура IP пакетов версии 4 представлена в таблице 2.3.1.

Таблица 2.3.1 – Структура IP пакетов версии 4

<table>
<tr><td>0-3</td><td>4-7</td><td>8-15</td><td>16-18</td><td>19-23</td><td>24-31</td></tr>
<tr><td>Version</td><td>IHL</td><td>ToS</td><td colspan="3">Total Length</td></tr>
<tr><td colspan="3">Identification</td><td>Flags</td><td colspan="2">Fragment Offset</td></tr>
<tr><td colspan="2">TTL</td><td>Protocol</td><td colspan="3">Header Checksum</td></tr>
<tr><td colspan="6">Source Address</td></tr>
<tr><td colspan="6">Destination Address</td></tr>
<tr><td colspan="5">Options</td><td>Padding</td></tr>
<tr><td colspan="6">Data</td></tr>
</table>

Version (Версия протокола) – для IPv4 значение поля равно 4;

IHL (IP Header Length - длина заголовка IP-пакета) – указывает количество 32-битных слов (4 байта) в заголовке IP. Поле IHL состоит из 4-х бит, следовательно, максимальная длина IP заголовка равна 64 байтам;

ToS (Type of Service - Тип обслуживания) – байт, содержащий набор критериев, определяющих тип обслуживания IP-пакетов, представлен в таблице 2.3.2.

Таблица 2.3.2 – Тип обслуживания

0	1	2	3	4	5	6	7
Precedence			D	T	R	0	0

Описание байта обслуживания побитно:

0-2 – Precedence (приоритет) данного IP-сегмента

3 – Delay (требование ко времени задержки) передачи IP-сегмента (0 – нормальная, 1 – низкая задержка)

4 – Throughput (требование к пропускной способности) маршрута, по которому должен отправляться IP-сегмент (0 – низкая, 1 – высокая пропускная способность)

5 – Reliability (требование к надежности) передачи IP-сегмента (0 – нормальная, 1 – высокая надежность)

6-7 –зарезервированы для будущего использования.

Total Length (Длина пакета) – длина пакета в октетах, включая заголовок и данные. Минимальное корректное значение для этого поля равно 20, максимальное 65535;

Identification (Идентификатор) – значение, назначаемое отправителем пакета и предназначенное для определения корректной последовательности фрагментов при сборке пакета. Для фрагментированного пакета все фрагменты имеют одинаковый идентификатор;

Flags (Флаги) – содержит два флага:

DF (Do not Fragments – не фрагментировать) – устанавливается когда IP пакет нельзя фрагментировать;

LF (Last Fragment – последний фрагмент) устанавливается, когда это последний пакет в серии фрагментных пакетов;

Fragment Offset (Смещение фрагмента) – значение, определяющее позицию фрагмента в потоке данных. Смещение задается количеством восьми байтовых блоков.

TTL (Время жизни) – число маршрутизаторов, которые должен пройти этот пакет. При прохождении маршрутизатора это число уменьшатся на единицу. Если значения этого поля равно нулю то, пакет должен быть

отброшен и отправителю пакета может быть послано ICMP сообщение Time Exceeded (код 11 тип 0);

Protocol (Протокол) – идентификатор интернет-протокола следующего уровня указывает, данные какого протокола содержит пакет;

Header Checksum (Контрольная сумма заголовка) – вычисляется в соответствии с RFC 1071;

Source/Destination Address (Адрес отправителя/получателя) – указывают IP адреса отправителя и получателя;

Options + Padding (Опции + Заполнение) – поля переменной длины для указания различных опций и выравнивания размера пакета относительно 32-х битной границы;

Data (Данные) – содержат полезную нагрузку полученную от протоколов транспортного уровня.

2.3.2. Классы IP адресов

Все IP-адреса можно разделить на две логические части – номера сети и номера узла сети (номер хоста). Чтобы определить какая именно часть IP-адреса принадлежит к номеру сети, а какая – к номеру хоста, определяется значениями первых бит адреса. Также, первые биты IP-адреса используются для того, чтобы определить к какому классу относится тот или другой IP-адрес.

В таблице 2.3.3 приведена структура IP-адреса для разных классов.

Таблица 2.3.3 – струкрута IP-адреса для различных классов

Класс	1 байт								1 байт								1 байт								1 байт							
Класс A	0	№ сети							№ узла																							
Класс B	1	0	№ сети														№ узла															
Класс C	1	1	0	№ сети																					№ узла							
Класс D	1	1	1	0	Адрес группы multicast																											
Класс E	1	1	1	1	0	Зарезервирован																										

Если адрес начинается с 0, то сеть относят к классу A и номер сети занимает один байт, остальные 3 байта номер узла в сети. Сети класса A имеют номера в диапазоне от 1 до 126. Номер 0 не используется, а номер 127 зарезервирован для специального применения. Сетей класса A немного, но количество узлов в них может достигать 2^{24}, то есть 16 777 216 узлов.

Если первые два бита адреса равны 10, то сеть относится к классу B. В сетях класса B под номер сети и под номер узла отводится по 2 байта. Таким образом, сеть класса является сетью средних размеров с максимальным числом узлов 2^{16}, что составляет 65 536 узлов.

Если адрес начинается с последовательности 110, то это сеть класса C. В этом случае под номер сети отводится 3 байта, а под номер узла – 1 байт. Сети этого класса наиболее распространены, число узлов в них ограничено 2^8, то есть 256 узлами.

Если адрес начинается с последовательности 1110, то он является адресом класса D и обозначает особый, групповой адрес – multicast. Если в пакете в качестве адреса назначения указан адрес класса D, то такой пакет должны получить все узлы, которым присвоен данный адрес.

Если адрес начинается с последовательности 11110, то это значит, что данный адрес относится к классу E. Адреса этого класса зарезервированы для будущих применений.

В таблице 2.3.4 приведены диапазоны номеров сетей и максимальное число узлов, соответствующих каждому классу сетей. Большие сети получают адреса класса A, средние – класса B, а маленькие – класса C.

Таблица 2.3.4 – Диапазоны номеров сетей и максимальное число узлов

Класс	Первые биты	Наименьший номер сети	Наибольший номер сети	Максимальное число узлов в сети
A	0	1.0.0.0	126.0.0.0	2^{24}-2
B	10	128.0.0.0	191.255.0.0	2^{16}-2
C	110	192.0.0.0	223.255.255.0	2^8-2
D	1110	224.0.0.0	239.255.255.255	Multicast
E	11110	240.0.0.0	247.255.255.255	Зарезервирован

2.3.3. Использование масок в IP адресации

Традиционная схема деления IP-адреса на номер сети (NetID) и номер узла (HostID) основана на понятии класса, который определяется значениями нескольких первых бит адреса. В настоящее время для определения номера сети и номера хоста используются маски.

Маска – это число, которое используется в паре с IP-адресом; двоичная запись маски содержит единицы в тех разрядах, которые должны в IP-адресе интерпретироваться как номер сети. Поскольку номер сети является цельной частью адреса, единицы в маске также должны представлять непрерывную последовательность.

Для стандартных классов сетей маски имеют следующие значения:

- класс A – 11111111. 00000000. 00000000. 00000000 (255.0.0.0);
- класс B – 11111111. 11111111. 00000000. 00000000 (255.255.0.0);
- класс C – 11111111. 11111111.11111111. 00000000 (255.255.255.0).

Снабжая каждый IP-адрес маской, можно отказаться от понятий классов адресов и сделать более гибкой систему адресации. Например, если адрес 185.23.44.206 ассоциировать с маской 255.255.255.0, то номером сети будет 185.23.44.0, а не 185.23.0.0, как это определено системой классов.

Расчет номера сети и номера узла с помощью маски:

	Десятичный вид	Двоичный вид	
IP	185.23.44.206	10111001.00010111.00101100.11001110	Логическое умножение
		&	
маска	255.255.255.0	11111111.11111111.11111111.00000000	
	185.23.44.0	10111001.00010111.00101100.00000000	Номер сети
	0.0.0.206	00000000.00000000.00000000.11001110	Номер узла

В масках количество единиц в последовательности, определяющей границу номера сети, не обязательно должно быть кратным 8, чтобы повторять деление адреса на байты.

Существует также короткий вариант записи маски, называемый префиксом, который содержит число двоичных единиц в маске, отсчет ведется слева направо. В таблице 2.3.5 отображается соответствие префикса с маской.

Таблица 2. 3.5 – Соответствие префикса и маски

Маска	Префикс	Количество узлов в сети
255.255.255.252	/30	4
255.255.255.248	/29	8
255.255.255.240	/28	16
255.255.255.224	/27	32
255.255.255.192	/26	64
255.255.255.128	/25	128
255.255.255.0	/24	256

Механизм масок широко распространен в IP-маршрутизации. С их помощью администратор может структурировать свою сеть. На основе этого же механизма можно объединять адресные пространства нескольких сетей путем введения «префиксов» с целью уменьшения объема таблиц маршрутизации.

Особые IP адреса

В протоколе IP существует несколько соглашений об особой интерпретации IP-адресов:

0.0.0.0 - представляет адрес шлюза по умолчанию, т.е. адрес по которому следует направлять пакеты, если не найден адрес в таблице маршрутизации;

255.255.255.255 – широковещательный адрес. Сообщения, переданные по этому адресу, получают все узлы локальной сети, содержащей источник сообщения;

«Номер сети».«все нули» – адрес сети (например 192.16.10.0);

«Все нули».«номер узла» – узел в данной сети (например 0.0.0.28). Может использоваться для передачи сообщений конкретному узлу внутри локальной сети;

Если в поле номера узла назначения стоят только единицы, то пакет, имеющий такой адрес, рассылается всем узлам сети с заданным номером сети. Например, пакет с адресом 192.16.10.255 доставляется всем узлам сети 192.16.10.0. Такая рассылка называется широковещательным сообщением (broadcast). При адресации необходимо учитывать те ограничения, которые вносятся особым назначением некоторых IP-адресов. Ни номер сети, ни номер узла не может состоять только из одних двоичных единиц или только из одних двоичных нулей. Отсюда следует, что максимальное количество узлов, приведенное в таблице для сетей каждого класса, на практике должно быть уменьшено на 2. В сетях класса С под номер узла отводится 8 бит, которые позволяют задавать 256 номеров: от 0 до 255. Однако на практике максимальное число узлов в сети класса С не может превышать 254, так как адреса 0 и 255 имеют специальное назначение.

Особый смысл имеет IP-адрес, первый октет которого равен 127.x.x.x. Он используется для тестирования программ и взаимодействия процессов в пределах одной машины. Когда программа посылает данные по IP-адресу 127.0.0.1, то образуется как бы «петля». Данные не передаются по сети, а возвращаются модулям верхнего уровня, как только что принятые. Поэтому в IP-сети запрещается присваивать машинам IP-адреса, начинающиеся со 127. Этот адрес имеет название loopback.

В протоколе IP нет понятия широковещательности в том смысле, в котором оно используется в протоколах канального уровня локальных сетей, когда данные должны быть доставлены абсолютно всем узлам. Как ограниченный широковещательный IP-адрес, так и широковещательный IP-адрес имеют пределы распространения в интерсети – они ограничены либо сетью, к которой принадлежит узел-источник пакета, либо сетью, номер

которой указан в адресе назначения. Поэтому деление сети с помощью маршрутизаторов на части локализует широковещательный шторм.

Все используемые в Интернете адреса, должны регистрироваться, что гарантирует их уникальность. Такие адреса называются реальными или публичными IP-адресами.

Для локальных сетей, не подключенных к Интернету, регистрация IP-адресов, естественно, не требуется, так как, в принципе, здесь можно использовать любые возможные адреса. Однако, чтобы не допускать возможность конфликтов при последующем подключении к сети Интернет, рекомендуется применять в локальных сетях только частные IP-адреса, представленные в таблице 2.3.6.

Таблица 2.3.6 – Частные IP-адреса

10.0.0.0 – 10.255.255.255
172.16.0.0 – 172.31.255.255
192.168.0.0 – 192.168.255.255

2.4. Интернет-протокол версии 6

Одним из недостатков интернет-протокола IPv4 является относительно небольшое количество адресов, всего около 4,23 миллиарда адресов. Использование IPv4 продолжается, благодаря широкому применению технологии преобразования сетевых адресов NAT (Network Address Translation).

Проблему нехватки сетевых адресов должна решить новая версия интернет протокола - **IPv6 (Internet Protocol version 6).**

Задачи внедрения IPv6:

- Обеспечение доступа к глобальной сети для миллиардов хостов;
- Сокращение размера таблиц маршрутизации;
- Упрощение протокола для ускорения обработки пакетов маршрутизации;
- Повышение уровня безопасности протокола;
- Упрощение работы многоадресных рассылок с помощью указания областей рассылки;
- Перспективы дальнейшего развития протокола в будущем;
- Постепенное внедрение протокола IPv6.

В протоколе IPv6 используется адреса 128 бит вместо 32 бит в IPv4.

Протокол IPv6 совместим с основными протоколами Интернета, включая TCP, UDP, ICMP.

Особенности IPv6

Протокол IPv6 имеет длину адреса 16 байт, обеспечить практически неограниченный запас IP – адресов.

Протокол IPv6 по сравнению с IPv4 имеет более простой заголовок пакета. Таким образом, маршрутизаторы могут быстрее обрабатывать пакеты, что повышает производительность.

Улучшенная поддержка необязательных параметров.

Повышен уровень безопасности, аутентификация и конфиденциальность являются ключевыми чертами протокола IPv6.

Уделено больше внимание типу представляемых услуг. Для этой цели в заголовке пакета IPv4 было отведено 8-разрядное поле.

2.4.1. Заголовок протокола IPv6

Структура IP пакетов версии 6 представлена в таблице 2.4.1.

Таблица 2.4.1 – Структура IP пакета версии 6

0-3	4-8	9-15	16-23	24-31
Version	Traffic Class	Flow Label		
Payload Length			Next Header	Hop Limit
Source Address				
Destination Address				
Data				

Version (Версия) – для IPv6 значение поля должно быть равно 6;

Traffic Class (Класс трафика) – используется для того, чтобы различать пакеты с разными требованиями к доставке в реальном времени;

Flow Label (Метка потока) – применяется для установки между отправителем и получателем псевдосоединения с определенными свойствами и требованиями. Например, поток пакетов между двумя процессами на разных хостах может обладать строгими требованиями к задержкам, что потребует резервирование пропускной способности;

Payload Length (Длина полезной нагрузки) – сообщает, сколько байт следует за 40-байтовым заголовком;

Next Header (Следующий заголовок) – сообщает, какой из дополнительных заголовков следует за основным;

Hop Limit (Максимальное число транзитных узлов) – аналог времени жизни (TTL);

Source/Destination Address (Адрес отправителя/получателя) – указывают IP адреса отправителя и получателя;

Data (Данные) – содержат полезную нагрузку, полученную от протоколов транспортного уровня.

2.4.2. Типы адресов и модель адресации протокола IPv6

В протоколе IPv6 существуют следующие типы адресов:

- **Unicast** – Идентификатор одиночного интерфейса. Пакет, посланный по уникастному адресу, доставляется интерфейсу, указанному в адресе;
- **Anycast** – Идентификатор набора интерфейсов, принадлежащих разным узлам. Пакет, посланный по этому адресу, доставляется одному из интерфейсов, указанному в адресе (ближайшему, в соответствии с мерой, определенной протоколом маршрутизации);
- **Multicast** – Идентификатор набора интерфейсов (обычно принадлежащих разным узлам). Пакет доставляется всем интерфейсам, заданным этим адресом.

В IPv6 не существует широковещательных адресов, их функции переданы Multicast -адресам.

В протоколе IPv6 действует следующая модель адресации.

IPv6 адреса всех типов ассоциируются с интерфейсами, а не узлами. Так как каждый интерфейс принадлежит только одному узлу, Unicast-адрес интерфейса может идентифицировать узел.

IPv6 Unicast-адрес соотносится только с одним интерфейсом. Одному интерфейсу могут соответствовать много IPv6 адресов различного типа. Существует два исключения из этого правила:

Одиночный адрес может приписываться нескольким физическим интерфейсам, если приложение рассматривает эти интерфейсы как единое целое при представлении на уровне Интернет.

Маршрутизаторы могут иметь ненумерованные интерфейсы (интерфейсу не присваивается IPv6 адрес) для соединений точка-точка, чтобы исключить необходимость вручную конфигурировать и объявлять эти адреса. Адреса не нужны для соединений точка-точка маршрутизаторов, если эти интерфейсы не используются в качестве точки отправления или назначения при посылке IPv6 дейтаграмм.

Формы представления адресов IPv6: шестнадцатеричные числа и двоеточия. Эта форма является предпочтительной и имеет вид n:n:n:n:n:n:n:n. Каждый знак n соответствует 4-х значному шестнадцатеричному числу (всего 8 шестнадцатеричных чисел, для каждого числа отводится 16 бит).

Например: `1FA9:FFFF:2621:ACDA:2245:BF98:3412:4167`.

Сжатая форма записи адреса. По причине большой длины адрес обычно содержит много нулей подряд. Для упрощения записи адресов используется

сжатая форма, в которой смежные последовательности нулевых блоков заменяются парами символов двоеточий (::). Однако такой символ может встречаться в адресе только один раз.

Например:

Адрес групповой рассылки FFEA:0:0:0:0:CA28:1210:4362 имеет сжатую форму FFEA::CA28:1210:4362;

Loopback-адрес 0:0:0:0:0:0:0:1 в сжатой форме вы-глядит так ::1;

Неопределенный адрес 0:0:0:0:0:0:0:0 превращается в ::.

Смешанная форма. Эта форма представляет собой сочетание адресов протоколов IPv4 и IPv6. В этом случае адрес имеет формат n:n:n:n:n:n:d.d.d.d, где каждый символ n соответствует 4-х значному шестнадцатеричному числу (6 шестнадцатеричных чисел, для каждого числа отводится 16 бит), а d.d.d.d -часть адреса, записанная в формате IPv4 (32 бита).

Например:

```
0:0:0:0:0:0:19.8.62.32
0:0:0:0:0:FFFF:111.214.2.34
```

или в сжатом виде:

```
::19.8.62.32
::FFFF:111.214.2.34
```

2.5. Протокол преобразования адресов ARP

В сети Интернет сетевое взаимодействие осуществляется на основе IP-адресов. В сети данные передаются с помощью технологий канального уровня, например, Ethernet. Сетевые устройства канального уровня коммутаторы используют физические MAC-адреса. Для получения MAC-адреса сетевого устройства в локальной сети по известному IP-адресу используется протокол преобразования адресов (ARP-Address Resolution Protocol).

Протокол ARP работает в режиме запрос-ответ. На каждом сетевом устройстве, например, сетевом адаптере хоста или коммутаторе поддерживается собственная ARP-таблица, в которой в ходе функционирования сети накапливается информация о соответствии между IP-адресами и MAC-адресами интерфейсов других устройств данной сети. Первоначально, при включении компьютера, маршрутизатора или коммутатора в сеть их ARP-таблицы пусты.

Таблица 2.5.1 – Формат ARP-таблицы

IP-адрес	MAC-адрес
192.168.0.1	08:7B:39:00:2F:C3
192.168.0.5	D8:6C:02:C1:DF:15

В процессе определения MAC-адреса по IP-адресу назначения, между взаимодействующими устройствами передаются два основных типа сообщения:

ARP запрос (ARP request). Когда устройству необходимо отправить IP пакет, оно проверяет свою ARP-таблицу, в которой ищет соответствие между MAC и IP-адресом получателя пакета. Если такого соответствия не обнаружено, то устройство формирует широковещательный ARP-запрос, в котором содержится IP-адрес получателя с просьбой сообщить его MAC-адрес. Далее устройство отправляет его на широковещательный MAC-адрес FF:FF:FF:FF:FF:FF, который получат все устройства в локальной сети.

ARP ответ (ARP reply). Все устройства в локальной сети, получившие широковещательный пакет, анализируют его и сравнивают IP-адрес назначения со своим IP-адресом. Если IP-адрес не совпадает, то пакет отбрасывается, если IP-адрес совпал с адресом, указанным в ARP-запросе, то устройство формирует ARP-ответ, в котором сообщает свой MAC-адрес и отправляет его устройству, от которого получен ARP-запрос.

Формат кадра протокола ARP представлен на рисунке 2.5.1.

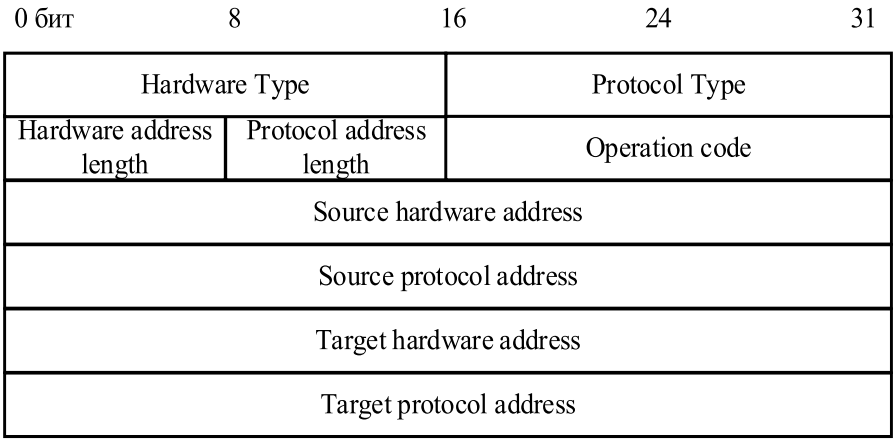

Рисунок 2.5.1 – Формат кадра протокола ARP

Поля кадра протокола ARP имеют следующие назначения:

- **Hardware Type** – тип аппаратного адреса, поле определяет тип используемого физического адреса. Для MAC-адреса в сети Ethernet это поле принимает значение 1;
- **Protocol type** – поле определяет тип сетевого адреса. Для адреса протокола IPv4 это поле принимает значение 2048;
- **Hardware address length** – длина физического адреса, определяет количество байт, выделенное для физического адреса. Для MAC-адреса в сети Ethernet имеет длину 6 байт;
- **Protocol address length** – длина сетевого адреса определяет количество байт выделенное для сетевого адреса. Для адреса протокола IPv4 имеет длину 4 байта;
- **Operation code** – gполе код операции принимает значение 1 в случае ARP-запроса и 2 в случае ARP-ответа. Последние четыре поля определяют MAC- и IP-адреса отправителя и получателя;
- **Source hardware address** – поле MAC-адреса источника кадра;
- **Source protocol address** – поле IP-адреса источника пакета;
- **Target hardware address** – поле MAC-адреса узла назначения;
- **Target protocol address** – поле IP-адреса узла назначения.

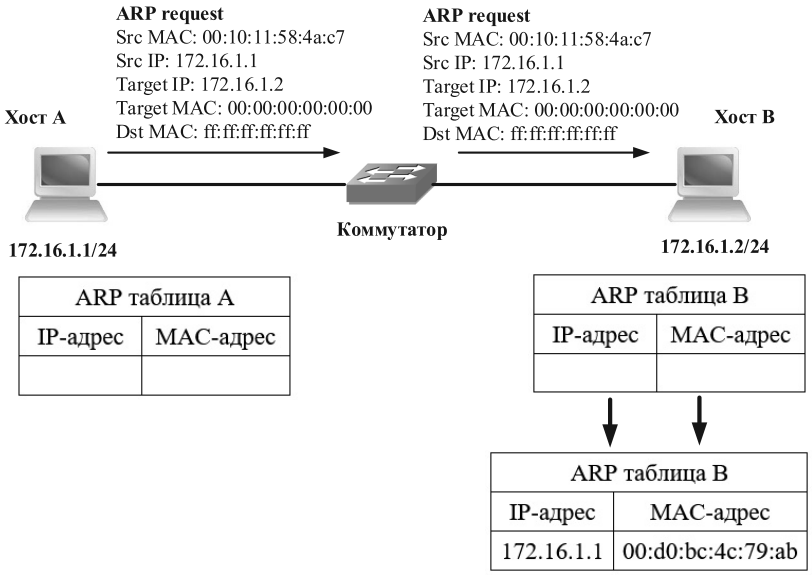

Рисунок 2.5.2 – Процесс запроса MAC-адреса с помощью ARP-request

Шаг 1: когда хост А хочет отправить свои данные на хост В, он формирует IP-пакет, который помещает в кадр Ethernet. Для отправки кадра Ethernet, необходим MAC-адрес узла назначения. Хост А обращается к своей ARP-таблице, в которой отсутствует запись для MAC-адреса хоста В. В этом случае хост А формирует широковещательный ARP-запрос, чтобы определить MAC-адреса хоста В. Запрос содержит MAC-адрес (Src MAC) и IP-адрес (Src IP) источника, а также IP-адрес (Target IP) узла назначения, при этом MAC-адрес узла назначения (Target MAC) заполняется нулями, после фрейм отправляется в сеть на широковещательный кадр ff:ff:ff:ff:ff:ff;

Шаг 2: так как фрейм широковещательный, то коммутатор принимает его и перенаправляет на все порты, кроме порта, на котором этот фрейм был получен. Если к коммутатору присоединены еще какие-либо сетевые устройства, то они также получат от него данный широковещательный фрейм. Эти устройства проанализируют полученный фрейм и добавят в свою ARP-таблицу IP-адрес и MAC-адрес хоста А, а после отбросят его, так как он не предназначен им.

Шаг 3: получив широковещательный фрейм от коммутатора, хост В обрабатывает кадр Ethernet, и анализирует его данные. Хост В видит, что это ARP-запрос, а его IP-адрес совпадает с IP-адресом назначения в ARP-запросе. После он добавляет полученные IP-адрес и MAC-адрес хоста А в свою ARP-таблицу и формирует ARP-ответ, в котором указаны его MAC-адрес и IP-адрес.

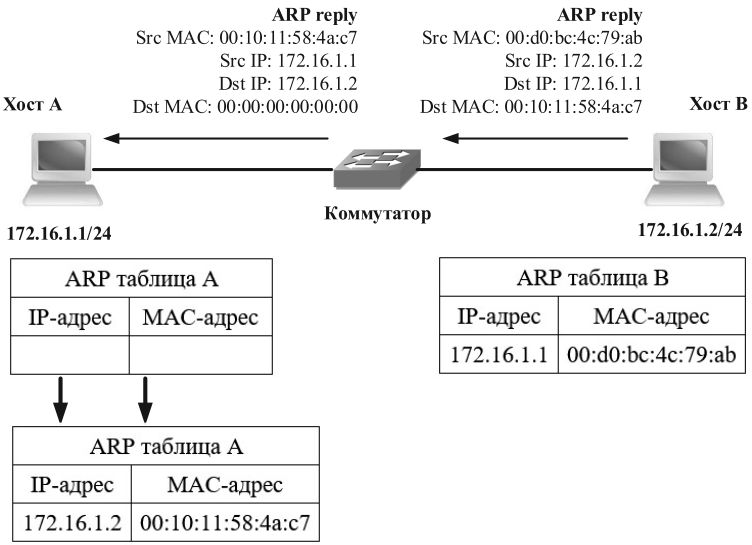

Рисунок 2.5.3 – Процесс передачи сообщения ARP-request

Шаг 4: Хост А получает ARP-ответ, добавляет в свою ARP-таблицу полученные MAC-адрес и IP-адрес хоста В. На этом работа протокола ARP завершается и начинает работу протокол IP, который инкапсулирует IP-пакет в кадр Ethernet, просматривает свою ARP-таблицу на наличие соответствующего MAC-адреса для IP-адреса 172.16.1.2, так как такая запись присутствует, то Хост А в заголовок кадра Ethernet помещает MAC-адрес хоста В и отправляет пакет в сеть к хосту В.

В ARP-таблицах существуют два типа записей: динамические и статические:

Статические записи создаются вручную и не имеют срока устаревания, существуют до тех пор, пока сетевое устройство остается включенным.

Динамические записи создаются автоматически при работе протокола ARP и должны периодически обновляться. Если запись не обновлялась в течение определенного времени, то она удаляется из таблицы. Поэтому в ARP-таблице содержатся записи не обо всех узлах сети, а только о тех, которые участвуют в сетевом взаимодействии.

2.6. Протокол динамического конфигурирования хоста DHCP

Протокол DHCP (Dynamic Host Configuration Protocol) – протокол динамической конфигурации хоста, используется для назначения хостам IP-адресов, в автоматическом режиме. Протокол DHCP выступает в качестве альтернативы ручной настройке параметров сети для компьютеров в сети. Помимо IP-адреса, протокол DHCP назначает другие важные сетевые настройки, такие как маска подсети, шлюз по умолчанию и адреса DNS-серверов.

Данный протокол работает на основе модели «Клиент-сервер». В качестве клиента выступает компьютер, который получает IP-адрес автоматически, а в качестве DHCP-сервера выступает компьютер, обеспечивающий выдачу IP-адресов другим устройствам в сети, а также ведет таблицу уже выделенных IP-адресов, чтобы избежать дублирования и последующих ошибок.

DHCP-клиент и DHCP-сервер обмениваются сообщениями в режиме запрос-ответ. В роли транспортного протокола для обмена DHCP-сообщениями выступает протокол UDP. При отправке сообщения с клиента на сервер используется 67-й порт DHCP-сервера, при передаче в обратном направлении - 68-й порт на стороне DHCP-клиента.

В процессе предоставления IP-адреса по протоколу DHCP между клиентом и сервером передаются следующие четыре основных сообщения:

DISCOVER (обнаружить). Передается клиентом DHCP при поиске подходящего сервера DHCP на широковещательный IP-адрес 255.255.255.255, адресом отправителя клиента является адрес 0.0.0.0.

- специально зарезервированный IP-адрес для использования хостами, у которых еще нет IP-адреса;
- 255.255.255.255 – адрес, зарезервированный как широковещательный адрес IPv4.

OFFER (предложение). Передается сервером DHCP как предложение клиенту IP-адреса;

REQUEST (запрос). Передается клиентом DHCP как запрос серверу на резервирование IP-адреса, указанного им в сообщении OFFER;

ACKNOWLEDGEMENT (подтверждение). Передается DHCP-сервером при назначении адреса и информировании о маске, а также IP-адресах шлюза по умолчанию и серверов DNS.

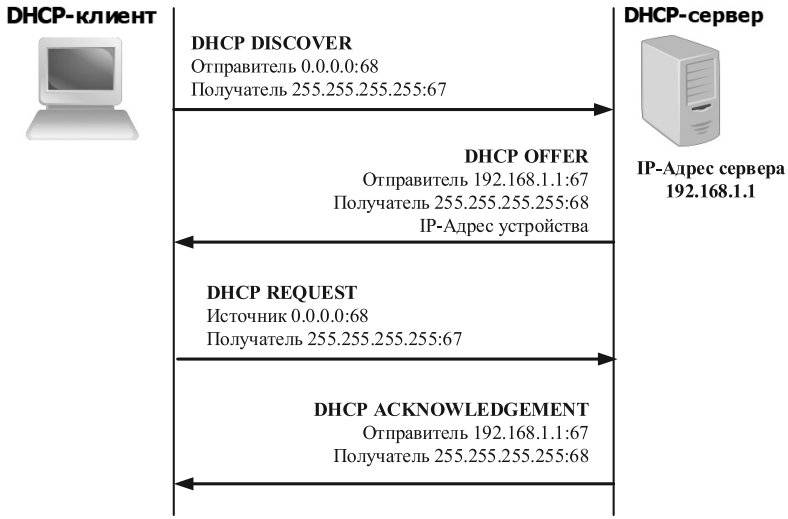

Рисунок 2.6.1 – Процесс обмена сообщениями DHCP

Процесс обмена сообщениями DHCP:

Шаг 1: когда клиент включается, у него нет никакой информации о той сети, в которой он находится. Его первая задача – узнать, где находится DHCP-сервер. Для того, чтобы найти DHCP-сервер, клиент с адреса 0.0.0.0 посылает сообщение DHCPDISCOVER на широковещательный IP-адрес 255.255.255.255, который получат все хосты в сети.

Шаг 2: сервер получает запрос DHCP DISCOVER, и отправляет в ответ сообщение DHCP OFFER, в котором содержатся все необходимые параметры конфигурации сети, запрашиваемые клиентом: IP-адрес, маска подсети, шлюз по умолчанию и IP-адреса DNS-серверов. Так как клиент все еще не имеет IP-адреса, то в качестве IP-адреса получателя сервер использует адрес 255.255.255.255, посылая его на все хосты в сети. Клиент получит предложение OFFER, а остальные устройства отбросят его.

Шаг 3: клиент формирует сообщение DHCP REQUEST, в котором посылает запрос на резервирование полученного IP-адреса от сервера, указав, что он принимает параметры конфигурации, отправленные ему. Сообщение DHCPR EQUEST по-прежнему отправляется клиентом с адреса 0.0.0.0, так как клиенту еще не разрешено использовать полученный IP-адрес.

Шаг 4: как только сервер получает DHCP REQUEST от клиента, он посылает в ответ сообщение DHCPACK о том, что теперь клиент может использовать IP-адрес, назначенный к нему. Клиент окончательно подключается к сети с параметрами, полученными от DHCP-сервера.

Формат кадра DHCP представлен на рисунке 2.6.2.

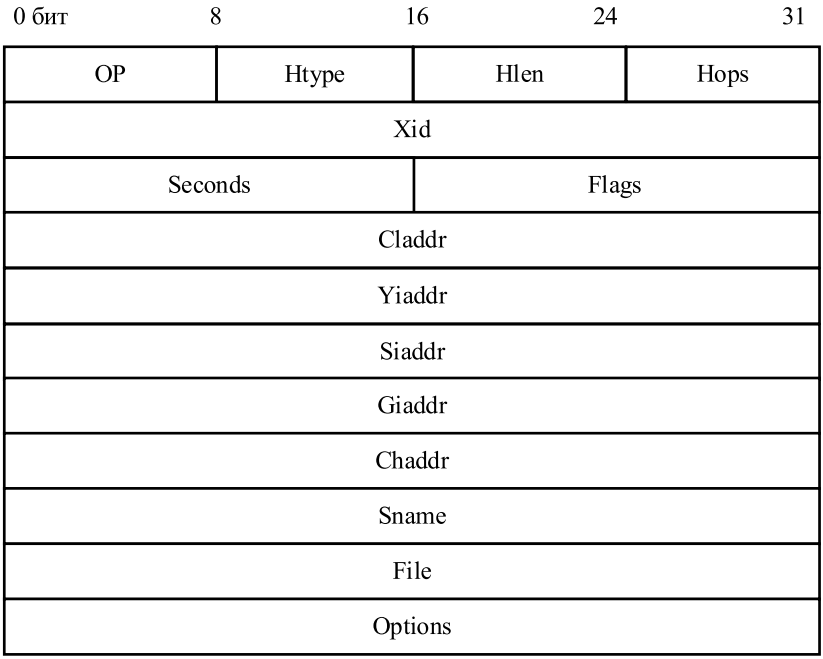

Рисунок 2.6.2 – Формат кадра DHCP

Поля кадра имеют следующие назначения:

- **OP (Operation Code)** – код операции, поле может принимать два значения: BOOTREQUEST (1, запрос от клиента к серверу) и BOOTREPLY (2, ответ от сервера к клиенту);
- **Htype (Hardware Type)** – поле определяет тип физического адреса, указанного в поле «Физический адрес клиента»;
- **Hlen (Hardware Address Length)** – длина физического адреса содержит число в байт, которые выделены под физический адрес клиента. Для MAC-адреса значение этого поля равно 6.
- **Hops** – количество транзитов, поле содержит количество промежуточных маршрутизаторов, через которые прошло сообщение. Клиент устанавливает это поле в 0;
- **Xid (Transaction ID)** – идентификатор транзакции позволяет соотнести последующие ответы с запросом в рамках одной DHCP-транзакции. Значение этого поля задается клиентом в начале процесса получения IP-адреса;
- **Seconds** – количество секунд, поле содержит время в секундах с момента начала процесса получения IP-адреса. Может не использоваться (в этом случае оно устанавливается в 0);
- **Flags** – поле флаги содержит флаги специальных параметров протокола DHCP. Старший бит определен как флаг BROADCAST, а остальные биты зарезервированы для будущего применения и должны быть равны 0. Флаг BROADCAST устанавливается в 1 если клиент требует широковещательного ответа;
- **Ciaddr (Client IP Address)** – поле IP-адрес клиента, заполняется только в том случае, если клиент уже имеет собственный IP-адрес (это возможно, если клиент выполняет процедуру обновления адреса по истечении срока аренды);
- **Yiaddr (Your IP Address)** – ваш IP-адрес, поле содержит IP-адрес, предлагаемый или уже назначенный сервером;
- **Siaddr (Server IP Address)** – поле IP-адрес сервера заполняется сервером при ответе на запрос;
- **Giaddr (Gateway IP Address)** – поле IP-адрес шлюза задает адрес агента-ретранслятора DHCP, которому сервер должен посылать ответы в случае, если клиент и сервер находятся в различных подсетях;
- **Chaddr (Client Hardware Address)** – поле физический адрес клиента, содержит MAC-адрес клианта;

- **Sname (Server Host Name)** – Необязательное поле имя сервера, содержит имя сервера в виде ASCIZ-строки;
- **File** – необязательное поле имя файла загрузки, содержит имя файла на сервере, используемое бездисковыми рабочими станциями при удаленной загрузке;
- **Options** – поле опции содержит различные дополнительные параметры конфигурации.

Также существуют дополнительные сообщения DHCP, которые могут передаваться при взаимодействии между клиентом и сервером:

DHCP NACK – сообщение, посылаемое сервером, которое запрещает использовать клиенту IP-адрес, который он запросил в сообщение DHCP REQUEST;

DHCPRELEASE – данное сообщение используется, когда клиент освобождает выданный ему IP- адрес;

DHCPRENEW – сообщение, посылаемое клиентом с просьбой обновить и продолжить аренду выданного сервером адреса.

DHCPINFORM – сообщение посылается серверу для получения локальных конфигурационных параметров, если клиент уже знает свой IP-адрес, например, если он настрое вручную.

DHCP-сервер может использовать два способа назначения IP-адресов устройствам в сети: фиксировано и динамически.

Фиксированный способ. В этом случае в конфигурационных файлах DHCP-сервера задается соответствие выдаваемого IP-адреса физическому MAC-адресу устройства. При выдаче IP-адреса, устройству будет выдан IP-адрес, заранее определенный в конфигурационном файле DHCP-сервера;

Динамический способ. В данном варианте назначение IP-адресов происходит из определенного пула (диапазона) адресов, специально выделенного на сервере для выдачи. При этом из пула могут быть исключены определенные IP-адреса, обычно это первые 10 IP-адресов подсети.

DHCP-сервер выделяет IP-адрес устройству на некоторое ограниченное время, называемое временем аренды (lease time), которое установлено в конфигурации сервера. По окончании времени аренды IP-адрес освобождается и DHCP-сервер может назначить его другому устройству. Клиент может продлить время аренды, обычно это происходит после истечения половины срока времени аренды, указанного клиенту при первичной выдаче IP-адреса в сообщении DHCP OFFER.

2.7. Трансляция сетевых адресов NAT

Трансляция сетевых адресов обеспечивает внутренним хостам частных сетей прозрачный доступ во внешнюю сеть. При трансляции сессии являются односторонними и направлены из частной сети. Трансляция сетевых адресов реализована в двух вариантах: NAT и NAPT. NAT обеспечивает преобразование только для адресов, а NAPT позволяет транслировать адреса IP и идентификаторы транспортного уровня (такие, как номера портов TCP/UDP или ICMP query ID).

Для выполнения трансляции могут использоваться только оконечные маршрутизаторы, как показано на рисунке 2.7.1.

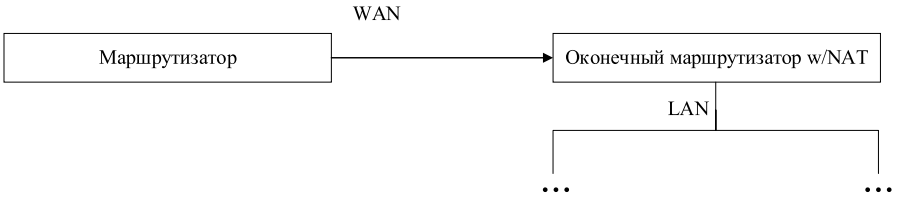

Рисунок 2.7.1 – Схема сети

2.7.1. Трансляция сетевых адресов NAT

NAT позволяет домену с набором приватных сетевых адресов взаимодействовать с внешними сетями, используя динамическое отображение набора приватных адресов на некое множество публичных адресов IP. Если число локальных узлов не превышает число имеющихся публичных адресов, каждому локальному адресу можно гарантированно поставить в соответствие публичный адрес. В противном случае число узлов, которые могут одновременно получить доступ во внешние сети, будет ограничено количеством публичных адресов. Отдельные локальные адреса могут статически отображаться на конкретные публичные адреса для обеспечения доступа к локальным хостам извне с использованием фиксированных адресов. Адреса внутри оконечного домена являются локальными для этого домена и некорректны за его пределами. Таким образом, эти адреса могут использоваться одновременно во множестве оконечных доменов.

В каждой точке выхода из оконечного домена во внешнюю сеть используется NAT. Если в домене используется несколько точек выхода, важное значение приобретает использование во всех таких точках одинаковых таблиц трансляции NAT.

В примере, показанном на рисунке 2.7.2, оконечные домены A и B используют блок приватных адресов класса A 10.0.0.0/8. Системе NAT домена A выделен блок адресов класса C 198.76.29.0/24, а в домене B — блок 198.76.28.0/24. Адреса блоков класса C являются уникальными в глобальном масштабе и другие устройства NAT не могут их использовать.

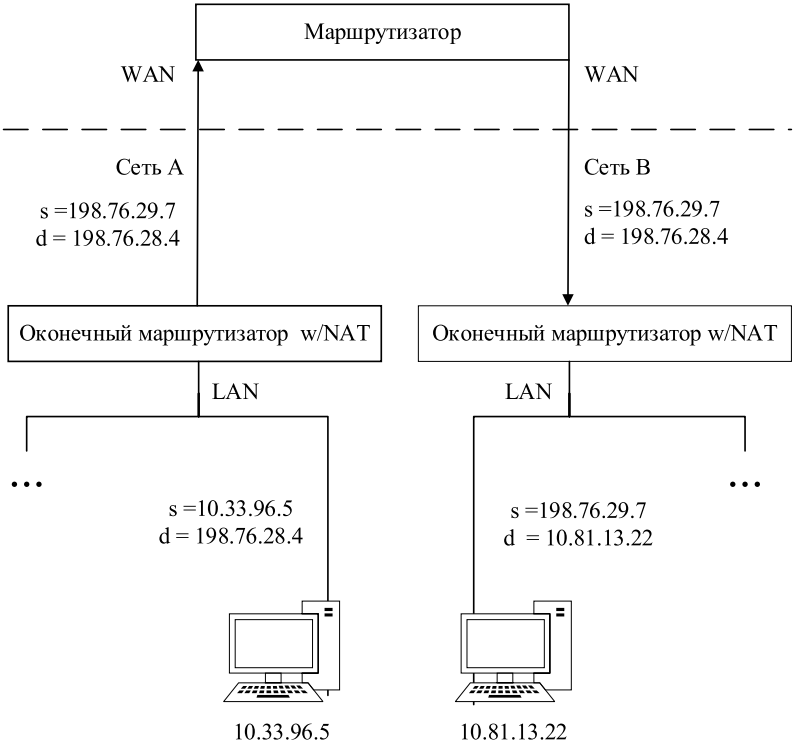

Рисунок 2.7.2 – Пример работы NAT

Когда хост домена A с адресом 10.33.96.5 хочет передать пакет хосту домена B с адресом 10.81.13.22, он использует в качестве адреса получателя уникальное в глобальном масштабе значение 198.76.28.4 и передает пакет своему основному маршрутизатору. Этот маршрутизатор имеет статический маршрут в сеть 198.76.0.0 и пакет пересылается в WAN-канал. Однако до того, как пакет будет передан, NAT преобразует адрес отправителя 10.33.96.5 в заголовке IP в уникальный адрес 198.76.29.7. Аналогично происходит преобразование адресов для пакетов IP, передаваемых в обратном направлении.

2.7.2. Трансляция сетевых адресов NAPT

В локальной сети, подключенной к сети Интернет маршрутизатору сети доступа присвоен уникальный адрес для интерфейса канала WAN, а узлы внутри сети используют приватные адреса, значимость которых ограничена локальной сетью. В этом случае узлы внутренней сети могут получить одновременный доступ во внешние сети с использованием единственного зарегистрированного адреса IP и трансляции NAPT. Этот вариант трансляции позволяет отображать пары типа (локальный адрес, локальный номер порта TU) в пары типа (зарегистрированный адрес, присвоенный номер порта TU).

Эта модель позволяет получить доступ во внешние сети с использованием единственного адреса IP, выделенного провайдером. Модель можно расширить для того, чтобы обеспечить доступ извне к локальному узлу за счет статического отображения локального узла на каждый номер порта TU для зарегистрированного адреса IP.

В показанном на рисунке 2.7.3, примере внутри оконечной сети А используется блок адресов класса А 10.0.0.0/8. WAN-интерфейсу граничного маршрутизатора сети провайдером присвоен IP-адрес 138.76.28.4.

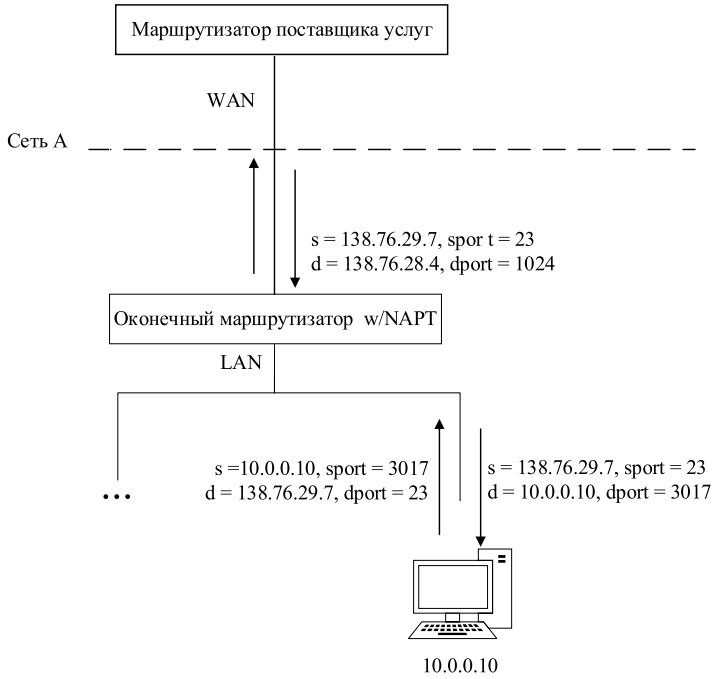

Рисунок 2.7.3 – Работа NAPT

Когда хост сети A с адресом 10.0.0.10 передает пакет telnet хосту 138.76.29.7, он указывает публичный адрес получателя 138.76.29.7 и отправляет пакет основному маршрутизатору. Маршрутизатор имеет статический маршрут в сеть 138.76.0.0/16 и пересылает пакет в канал WAN. Однако до пересылки пакета NAPT транслирует адрес отправителя 10.0.0.10 и номер порта TCP 3017 в заголовках IP и TCP, используя публичный адрес 138.76.28.4 и уникальное значение номера порта TCP (1024). Для обратных пакетов происходит похожее преобразование адреса и номера порта TCP в IP-адрес локального хоста и номер целевого порта TCP.

В описанном варианте поддерживаются только сеансы TCP/UDP, организованные из локальной сети.

В дополнение к сессиям TCP/UDP маршрутизатор NAPT может также обеспечивать мониторинг сообщений ICMP, за исключением типа REDIRECT. Запросы ICMP транслируются подобно пакетам TCP/UDP и поле идентификатора в заголовке сообщения ICMP будет уникально отображаться в идентификатор запроса для зарегистрированного адреса IP. Поле идентификатора в заголовках сообщений ICMP устанавливается отправителем и возвращается неизменным в отклике на запрос. Следовательно, пара (локальный адрес IP, локальный идентификатор ICMP) отображается в пару (публичный адрес IP, выделенное значение идентификатора ICMP) маршрутизатором NAPT и обеспечивает уникальную идентификацию всех типов сообщений ICMP от любого из локальных хостов. Изменение сообщений ICMP об ошибках рассматриваемое ниже, включает модификацию данных

Сессии, отличные от TCP, UDP и ICMP, не поддерживаются для локальных узлов, обслуживаемых маршрутизатором NAPT.

2.7.3. Изменения в заголовках IP, TCP, UDP и ICMP

Заголовки пакеты, относящиеся к сессиям NAT, подвергаются модификации.

При использовании NAT изменяется заголовок IP в каждом пакете. Модификация включает адрес IP (адрес отправителя для исходящих пакетов и адрес получателя для входящих) и контрольную сумму IP.

Для сессий TCP ([TCP]) и UDP ([UDP]) также требуется изменять контрольную сумму в заголовках TCP/UDP. Это связано с тем, что контрольная сумма TCP/UDP учитывает также псевдозаголовок, содержащий IP-адреса отправителя и получателя. Исключением являются случаи, когда контрольная сумма заголовка UDP имеет значение 0 — в этом случае поле контрольной суммы не меняется. Для пакетов ICMP Query ID не требуется вносить

изменений в заголовок ICMP, поскольку контрольная сумма в заголовке ICMP не учитывает адресов IP.

При использовании NAPT изменение заголовка IP похоже на случай NAT. Для сессий TCP/UDP изменяется также номер порта TU (порт отправителя для исходящих пакетов и порт получателя для входящих) в заголовке TCP/UDP. Заголовок ICMP в пакетах ICMP также требуется изменять для корректировки значения идентификатора запроса и контрольной суммы заголовка ICMP. Идентификатор запроса хоста внутренней сети в исходящих пакетах должен заменяться на присвоенный при трансляции идентификатор, а для входящих откликов должно выполняться обратное преобразование. Контрольная сумма заголовка ICMP должна корректироваться с учетом трансляции Query ID.

2.8. Протокол передачи команд и сообщений ICMP

Протокол ICMP (Internet Control Message Protocol – протокол межсетевых управляющих сообщений) является вспомогательным сетевым протоколом в стеке TCP/IP. Он предназначен для передачи транспортной и диагностической информации.

Протокол ICMP используется для транспортировки различной служебной информации, поэтому определена только общая структура заголовка ICMP-пакета (рисунок 2.8.1).

Type (Тип)	Code (Код)	Checksum (Контрольная сумма)
Unused (Разное)		
Internet Header + 64 bits of Original Data Datagram (IP заголовок, 8 байт данных)		

Рисунок 2.8.1 – Заголовок ICMP-пакета

Type (Тип) – однобайтовое поле, содержащее идентификатор типа ICMP-пакета. Возможные значения этого поля приведены в таблице 2.8.1.

Таблица 2.8.1 – Типы ICMP пакетов

Поле Type	Назначение
0	Ответ на запрос эха
3	Адресат не доступен
4	Подавление источника
5	Перенаправление
8	Запрос эха
11	Исчерпано время жизни
12	Ошибка в параметре
13	Запрос временной метки
14	Ответ на запрос временной метки

Code (Код) – однобайтовое поле, значение которого конкретизирует назначение ICMP-пакета определенного типа.

Checksum (Контрольная сумма) – 16-битовое поле, содержащее контрольную сумму, подсчитанную для всего ICMP-пакета целиком.

Unused (Разное) – четырехбайтовое поле, предназначенное для хранение разнообразной информации, специфичной для ICMP-пакетов определенного типа (например, номера в TCP-последовательности, IP-адреса).

Internet Header + 64 bits of Original Data Datagram (IP заголовок, 8 байт данных) – содержит заголовок IP-сегмента, который явился причиной появления данного ICMP-пакета, и первые 8 байт данных тела этого IP-пакета. Если ICMP-пакет есть результат проявления аномалии во взаимодействии TCP или UDP, то эти 8 байт будут представлять собой первые восемь байтов, соответственно, TCP или UDP заголовка, что дает возможность определить номера портов.

Для ICMP-пакетов некоторых типов это может содержать не начало IP-сегмента, а тестовые данные.

Источниками и обработчиками ICMP-пакетов могут быть IP, TCP и UDP модули.

Проблемы в доставке и обработке ICMP-пакетов никогда не приводят к порождению новых ICMP-пакетов, уведомляющих об этих проблемах. Сделано это с целью избежать возможных бесконечных циклов генерации ICMP-пакетов в сети.

ICMP-сообщения разделяются на две категории (рисунок 2.8.2).

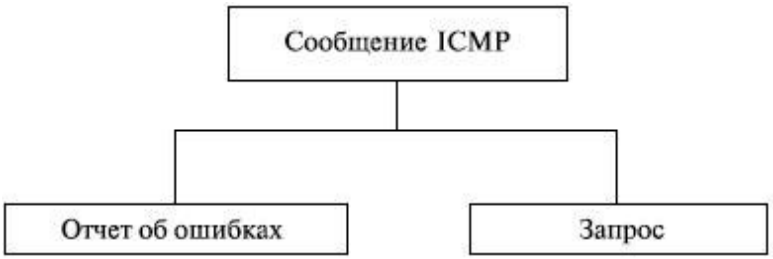

Рисунок 2.8.2 - Категории ICMP-сообщений

Некоторые типы ICMP-сообщений

- Адресат не доступен – ICMP-пакет такого типа генерируется в следующих случаях:
  - сеть, узел сети, протокол или порт не доступны;
  - в ходе продвижения по сети IP-пакета потребовалась его фрагментация, однако в заголовке пакета установлен запрещающий фрагментацию флаг DF;
  - маршрут, указанный в поле дополнительных данных IP-пакета, оказался недействительным.
- Подавление источника – такой тип генерируется в ситуациях, когда некоторый узел не имеет достаточно места в своих буферах для размещения интенсивно поступающих к нему данных, он может послать узлам-источникам ICMP-пакет данного типа (source quench). Узел-источник в ответ на такое уведомление обязан уменьшить темп передачи данных.
- Перенаправляющий ICMP-пакет посылается к источнику данных, когда узел-шлюз обнаруживает, что источник может направлять свои данные непосредственно к следующему шлюзу маршрута. Такой ICMP-пакет содержит в себе IP-адрес этого шлюза. Этот IP-адрес должен быть включен в таблицу маршрутизации на узле источнике данных.
- Эхо ICMP-пакет. Для реализации эха IP-модуль на узле A отправляет узлу B ICMP-сообщение типа “запрос эха”, содержащий в своём теле вместо IP-заголовка тестовые данный произвольной длины. Узел B, получив такой запрос, возвращает узлу A ICMP-пакет типа "ответ на запрос эха", содержащий те же данные, что и в запросе. Эхо-пакеты используются для проверки достижимости удаленных узлов сети и измерения времени прохождения данных.
- Исчерпано время жизни. ICMP-пакет данного типа посылается источнику IP-пакета в случае, если исчерпано время жизни (TTL) IP-

пакета или исчерпано допустимое время на сборку фрагментированного IP-пакета.

– Неверный параметр. Используя ICMP-пакет данного типа источник IP-пакета информируется о том, что данный пакет уничтожен, так как присутствует ошибка в одном из полей заголовка.

2.9. Маршрутизация

2.9.1. Основы процесса маршрутизации

Сетевой уровень модели OSI определяет правила доставки пакетов IP от устройства, создавшего пакет, на устройство его получателя. Этот процесс требует взаимодействия разных устройств на основе определенных правил.

Маршрутизация IP. Процесс перенаправления пакетов IP хостам и маршрутизаторам по локальным и глобальным сетям.

IP-адресация. Адреса, идентифицирующие хосты отправителя и получателя пакета. Правила адресации организуют адреса в группы, что помогает процессу маршрутизации.

IP-адресация и маршрутизация две взаимосвязанные темы, поскольку маршрутизация IP полагается на структуру и значение IP-адресов.

Протокол IP отвечает за маршрутизацию данных в форме пакетов IP от хоста отправителя к хосту получателя. Он не участвует в физической передаче данных, это задача более низких уровней модели OSI. Для этого логика сетевого уровня на хосте или маршрутизаторе должна передать пакет протоколу канального уровня, который, задействует физический уровень для осуществления передачи данных. Канальный уровень добавляет к пакету соответствующий заголовок и концевик, формируя отправляемый по физической среде фрейм.

Маршрутизаторы работают на сетевом уровне модели OSI, получая и перенаправляя IP-пакеты между сетями. Получив на интерфейсе фрейм канального уровня, маршрутизатор разбирает его до пакета сетевого уровня IP, анализирует IP-адрес получателя, и используя свою таблицу маршрутизации, перенаправляет пакет дальше.

Для передачи IP-пакетов, инкапсулированных во фреймы канального уровня, по физической линии связи, маршрутизаторы должны знать и уметь определять физические адреса (MAC-адреса) устройств назначения. Для этих целей все сетевые устройства используют протокол преобразования адресов ARP (Address Resolution Protocol). Протокол ARP по известному IP-адресу динамически определяет MAC-адрес получателя фрейма и формирует на устройстве ARP-таблицу соответствия IP-адреса и MAC-адреса получателя.

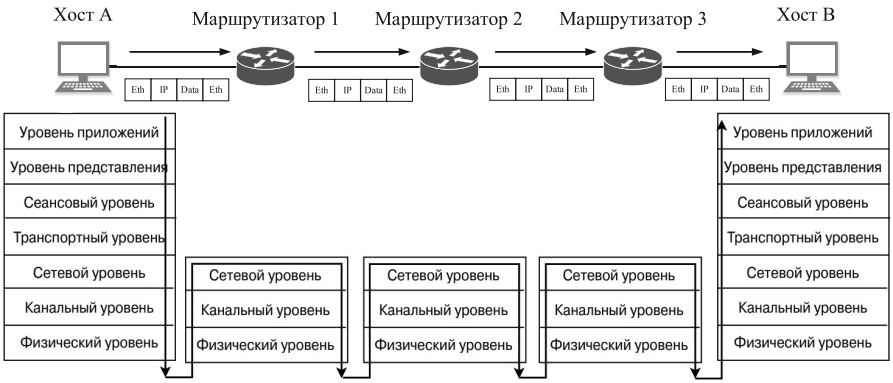

Рисунок 2.9.1 – Процесс маршрутизации в сети.

Важным звеном в работе любого маршрутизатора является его таблица маршрутизации. Анализируя маршруты в этой таблице, маршрутизаторы принимают решения куда направлять IP-пакеты для того, чтобы доставить их в сеть назначения. Таблица маршрутизации содержит основную информацию для работы маршрутизатора (рисунок 2.9.2).

Network	Gateway	Interface	Metric
172.16.0.0/24	10.10.3.1	Eth0	1
192.168.10.0/24	213.234.5.1	Eth1	5
10.1.3.0/30	Directly connected	Eth0	
213.234.5.0/30	Directly connected	Eth1	
10.10.1.0/30	Directly connected	Eth2	
Default gateway 10.10.1.1			

Рисунок 2.9.2 – Формат таблицы маршрутизации

Данные таблицы маршрутизации:

- **Network** – в данном столбце указывается адрес и маска сети назначения;
- **Gateway** – столбец шлюз указывает маршрутизатору куда должен быть отправлен пакет, чтобы достигнуть сети назначения;
- **Interface** – столбец интерфейс указывает через какой интерфейс доступен шлюз;
- **Metric** – метрика маршрута, чем меньше значение метрики маршрута в определенную сеть, тем более предпочтителен данный маршрут;

- **Directly connected** – напрямую подключенные сети, в таблице маршрутизации содержатся сети, которые напрямую подключены к маршрутизатору, также для них указан интерфейс подключения;
- **Default gateway** – статический маршрут по умолчанию (шлюз по умолчанию). Шлюз по умолчанию является маршрутом, который будет соответствовать IP-адресам всех пакетов. Когда маршрутизатор не находит соответствия между адресом получателя пакета и записью в таблице маршрутизации, он по умолчанию отправляет такие пакеты на специально заданный IP-адрес, являющийся шлюзом. Маршруты по умолчанию часто используется между граничным маршрутизатором компании и маршрутизатором Интернет-провайдера, перенаправляя на него все пакеты, исходящие из локальной сети предприятия.

Если у маршрутизатора есть несколько разных маршрутов до одной сети назначения, то маршрутизатор выбирает маршрут на основании метрик. Чем меньше метрика, тем больше маршрутизатор доверяет этому маршруту.

Статические маршруты имеют метрику, равную единице, что дает им самый высокий приоритет, так как статические маршруты конфигурируются вручную, а значит степень доверия к ним выше. Также маршрутизатор может автоматически изучать маршруты с помощью протоколов динамической маршрутизации, в этом случае маршрутизатор сам рассчитывает метрику для каждого изученного им маршрута.

У разных протоколов динамической маршрутизации метрика рассчитывается по-разному. Например, у протокола OSPF метрика рассчитывается на основании суммарной цены маршрута, которая в свою очередь зависит от ширины полосы пропускания интерфейсов и каналов связи на маршруте до сети назначения. А у протокола RIPv2 метрика рассчитывается исходя из числа транзитных участков между маршрутизатором и подсетью назначения.

2.9.2. Алгоритм работы маршрутизатора

Когда маршрутизатор получает фрейм из канала связи со своим адресом, он должен обработать его содержимое и отправить дальше. Для этого маршрутизатор применяет к фрейму канала связи следующую логику:

Шаг 1. Для проверки ошибок фрейма используется поле контрольной суммы фрейма (FCS) канала связи. Если есть ошибки, фрейм отбрасывается;

Шаг 2. Если пакет не был отброшен на предыдущем этапе, отбрасывается старый канальный заголовок и концевик и остается только пакет IP;

Шаг 3. IP-адрес отправителя пакета IP сопоставляется с таблицей маршрутизации и определяется маршрут, соответствующий этому адресу;

маршрут идентифицирует исходящий интерфейс маршрутизатора и IP-адрес маршрутизатора следующего перехода;

Шаг 4. Пакет IP инкапсулируется в новый канальный заголовок и концевик, подходящий для исходящего интерфейса, и фрейм отправляется.

Согласно этим этапам, каждый маршрутизатор перенаправляет пакет следующей области, заключив его во фрейм канала связи. Маршрутизаторы повторяют этот процесс, пока пакет не достигнет своего конечного получателя. Блок-схема алгоритма работы маршрутизатора приведена на рисунке 2.9.3.

Блок-схема алгоритма подробно показывает все этапы работы маршрутизатора:

Этап 1. Когда Ethernet фрейм поступает на маршрутизатор, сначала проверяется его контрольная сумма, для этого используется поле контрольной последовательности FCS (Frame Check Sequence). Получив фрейм, маршрутизатор выполняет собственное вычисление контрольной суммы для этого фрейма, сравнивает полученное значение с принятым значением поля FCS и, таким образом, определяет, не искажен ли полученный кадр. Если есть ошибки, фрейм отбрасывается.

Этап 2. Если фрейм не был отброшен на предыдущем этапе, то из него извлекается IP-пакет, отбрасывается старый канальный заголовок и концевик фрейма и остается только пакет IP. Маршрутизатор обрабатывает его и уменьшает время жизни IP-пакета TTL на единицу. Если поле TTL становится равным нулю, то пакет отбрасывается, а отправителю посылается ICMP-сообщение Time Exceeded (время превышено).

Этап 3. Если IP-пакет не был отброшен на предыдущем этапе, то он обрабатывается маршрутизатором дальше. Выполняется проверка, адресован ли пакет маршрутизатору или потребуется его дальнейшая маршрутизация на пути к пункту назначения. Пакеты, адресованные маршрутизатору в качестве IP-адреса получателя, содержат адрес одного из интерфейсов маршрутизатора. У таких пакетов удаляется заголовок IP, и они передаются на четвертый уровень. Если пакету предстоит маршрутизация, то IP-адрес получателя пакета сопоставляется с таблицей маршрутизации и определяется маршрут, который соответствует этому адресу. Маршрут идентифицирует исходящий интерфейс маршрутизатора и IP-адрес маршрутизатора следующего перехода;

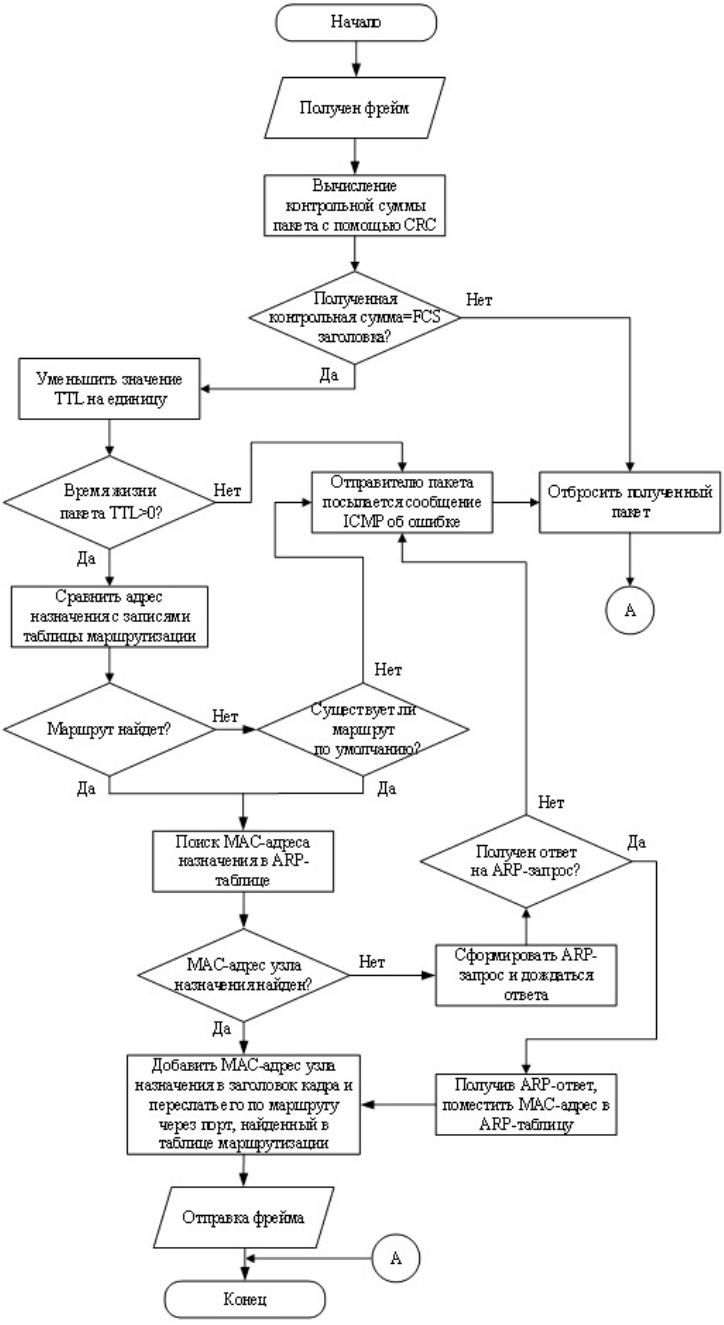

Рисунок 2.9.3 – Алгоритм работы маршрутизатора

Этап 4. Выбрав наилучший маршрут до сети назначения на основании метрики, маршрутизатор передает пакет на обработку канальному уровню. Если в таблице маршрутизации не оказалось маршрута до адреса назначения, маршрутизатор проверяет таблицу маршрутизации на наличие маршрута по умолчанию. Если маршрут по умолчанию также не сконфигурирован на устройстве, то IP-пакет отбрасывается, а отправителю пакета посылается ICMP-сообщение Host Unreachable (получатель недостижим);

Этап 5. Поступив на канальный уровень, пакет инкапсулируется в фрейм, добавляется заголовок и концевик фрейма. Далее маршрутизатор проверяет свою ARP-таблицу на наличие записи, устанавливающей соответствие между IP-адресом следующего узла, на который необходимо послать фрейм для достижения сети назначения, и его MAC-адресом. Если запись найдена, маршрутизатор добавляет MAC-адрес следующего узла в заголовок фрейма, вычисляет контрольную сумму полученного фрейма и отправляет его по каналу связи на MAC-адрес назначения.

Если запись для IP-адреса следующего узла назначения не найдена, то маршрутизатор формирует широковещательный ARP-запрос, в котором просит откликнуться устройство с заданным IP-адресом и сообщить ему о своем MAC-адресе в ARP-ответе. Получив ARP-ответ, маршрутизатор добавляет полученный MAC-адрес в заголовок фрейма, вычисляет контрольную сумму полученного фрейма и отправляет его на MAC-адрес назначения. В случае, если ARP-ответ получить не удается, маршрутизатор отбрасывает пакет, а отправителю пакета посылает ICMP-сообщение Host Unreachable (получатель недостижим).

2.10. Протоколы маршрутизации RIP, OSPF, BGP

Маршрутизация подразумевает два параллельных процесса: подготовку маршрутной таблицы и переадресацию дейтаграмм с помощью этой таблицы. Формирование маршрутной таблицы производится посредством протоколов маршрутизации или под воздействием инструкций администратора.

2.10.1. Внутренний протокол маршрутизации RIP

Протокол динамической маршрутизации RIP (RFC 1058) относится к категории протоколов маршрутизации внутреннего шлюза (Interior Gateway Protocol – IGP), которые предназначены для работы внутри автономной системы AS. Автономная система (AS) – это сеть под административным контролем одной организации. Сеть интернет провайдера является автономной системой.

Алгоритмы протоколов маршрутизации определяют то, как они решают задачи для изучения всех возможных маршрутов и выбора наилучшего, а также для реакции на изменения в работе сети (конвергенции). Конвергенция (сходимость) – процесс перестроения маршрутов, происходящий при изменении топологии сети, т.е. когда отказывает маршрутизатор или канал связи, либо наоборот – когда они восстанавливаются. Для этих целей протокол RIP использует дистанционно-векторный алгоритм (алгоритм Беллмана-Форда), выбирая наилучший маршрут к подсетям на основании самой низкой метрики маршрута.

Метрика в данном протоколе характеризуется вектором расстояния – числом транзитных участков до IP-узла назначения. Предполагается, что каждый маршрутизатор является отправной точкой нескольких маршрутов до сетей, с которыми он связан. Описания этих маршрутов хранится в таблице маршрутизации.

Таблица маршрутизации содержит следующие данные:

- Адрес сети пункта назначения;
- Метрику маршрута (от 1 до 15 – число транзитных участков до сети назначения);
- Следующий маршрутизатор (шлюз), на который нужно отправить пакет, чтобы достичь сети назначения.

Изначально, когда маршрутизатор RIP только начинает свою работу, он знает только о сетях, непосредственно подключенных к его интерфейсам. Маршрутизатор добавляет их в свою таблицу маршрутизации и присваивает им метрику, равную единице.

Каждый маршрутизатор RIP раз в 30 секунд отправляет обновления RIP соседним маршрутизаторам, с которыми соединен напрямую. В обновлениях RIP содержится полная копия таблицы маршрутизации. Когда маршрутизатор получает обновление RIP от соседа, он узнает из него о новых сетях, добавляет эти сети в свою таблицу маршрутизации и отправляет обновление другим маршрутизаторам, увеличивая счетчик транзитных участков (метрику) для каждой полученной им сети на единицу.

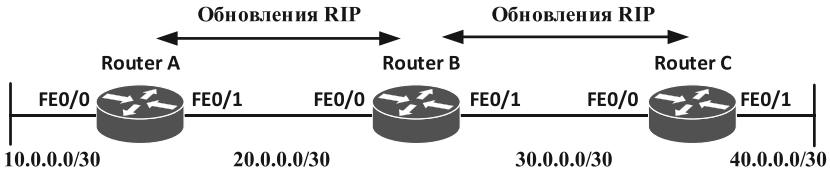

Таблица маршрутизации

Сеть	Метрика	Шлюз
10.0.0.0/30	1	FE0/0
20.0.0.0/30	1	FE0/1
30.0.0.0/30	2	Router B
40.0.0.0/30	3	Router B

Таблица маршрутизации

Сеть	Метрика	Шлюз
20.0.0.0/30	1	FE0/0
30.0.0.0/30	1	FE0/1
10.0.0.0/30	2	Router A
40.0.0.0/30	2	Router C

Таблица маршрутизации

Сеть	Метрика	Шлюз
30.0.0.0/30	1	FE0/0
40.0.0.0/30	1	FE0/1
20.0.0.0/30	2	Router B
10.0.0.0/30	3	Router B

Рисунок 2.10.1 – Процесс обмена маршрутной информацией

Сообщения протокола RIP инкапсулируются протоколом UDP, передача осуществляется через UDP-порт 520. Если между сетью отправителя и сетью назначения расположено три маршрутизатора RIP, то считается, что между ними 4 транзитных участка, и метрика до сети назначения будет иметь значение 4. Такой вид метрики не учитывает различий в пропускной способности или загруженности отдельных сегментов сети. Это является одним из недостатков протокола RIP, так как если до сети назначения существует два маршрута, то протокол RIP отправит трафик через маршрут с наименьшей метрикой, даже если пропускная способность этого канала слишком низкая.

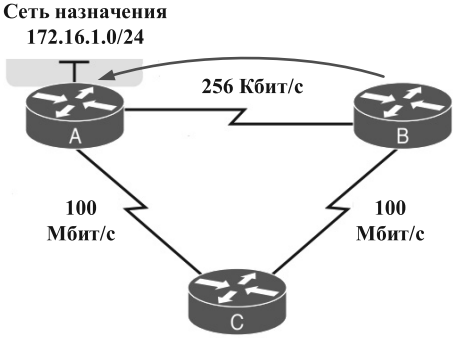

Таблица маршрутизации роутера B

Сеть	Метрика	Шлюз
172.16.1.0/24	2	Router A
172.16.1.0/24	3	Router C

Рисунок 2.10.2 – Выбор маршрута по протоколу RIP

В 1994 году вышла вторая версия протокола RIPv2 (RFC 2453), которая начала поддерживать групповые (multicast) сообщения и в которой появилась возможность работать с подсетями. Также появилась поддержка аутентификации сообщений протокола, это используется для повышения безопасности передачи RIP пакетов. Есть возможность шифровать аутентификационный код с помощью MD5.

Формат сообщения протокола RIPv2 представлен на рисунке 2.10.3.

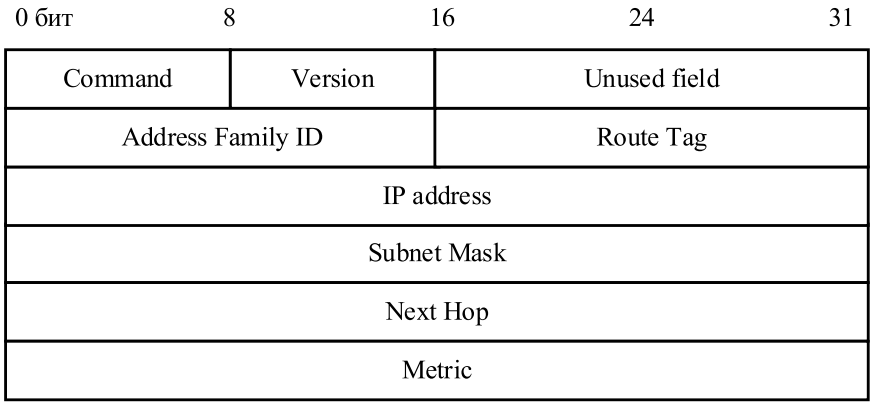

Рисунок 2.10.3 – Формат RIPv2-сообщения

Сообщение RIPv2 содержит следующие поля:

- **Command** – определитель, который идентифицирует тип команды. Имеет кодировку, как и в формате RIPv1-сообщения;
- **Version** – поле, содержащее версию протокола RIP (2 для протокола RIPv2);
- **Unused field** – неиспользуемое поле, заполняется нулями;
- **Address Family ID** – номер прикладного процесса маршрутизации, устанавливается значение 2 для IPv4. Единственным исключением является запрос полной таблицы маршрутизации маршрутизатора, в этом случае будет установлено значение ноль;
- **Route Tag** – предоставляет поле для маркировки внешних маршрутов или маршрутов, которые были перераспределены из другого протокола в протокол RIPv2;
- **IP address** – IPv4-адрес пункта назначения маршрута. Это может быть основной сетевой адрес, подсеть или маршрут к хосту;
- **Subnet Mask** – 32-битная маска подсети, идентифицирующая часть сети и подсети адреса ipv4;
- **Next Hop** – Определяет IP-адрес следующего маршрутизатора (шлюза), на который нужно отправить пакет, чтобы достичь сети назначения;
- **Metric** – метрика маршрута, число от 1 до 16.

Протокол RIPv2 каждые 30 секунд отправляют обновления RIP на групповой IP-адрес 224.0.0.9, используя UDP порт 520, содержащие полную

таблицу маршрутизации со всеми известными маршрутами соседним маршрутизаторам.

Из-за вероятности возникновения маршрутных петель, максимальная метрика маршрута имеет значение 15, т.е. максимальное число транзитных участков не должно превышать 15. При отказе маршрутов, маршрутизаторы анонсируют отказавший маршрут со специальным значением метрики – бесконечностью. Протокол RIP определяет бесконечность как значение метрики 16.

Для управления частотой рассылки обновлений RIP и улучшения времени конвергенции протокол RIP использует специальные таймеры:

- **Update timer** – таймер отправки маршрутных обновлений RIP (по умолчанию составляет 30 секунд). Каждые 30 секунд процесс RIP на маршрутизаторе рассылает сообщения другим маршрутизаторам RIP, содержащие полную таблицу маршрутизации;
- **Timeout timer** – таймер тайм-аута маршрута (по умолчанию составляет 180 секунд). По истечению таймера, маршрут, по которому не получены обновления в Update сообщениях помечается как не доступный и помечается метрикой 16, но сохраняется в таблице маршрутизации до тех пор, пока не истекает Flush таймер;
- **Flush timer** – таймер сброса маршрута (по умолчанию равен 240 секунд). По истечении таймера, недоступный маршрут окончательно удаляется из таблицы маршрутизации.

В 1997 году вышел протокол RIPng (RIP next generation, RFC 2080), RIP следующего поколения представляет собой полностью новый протокол, разработанный для работы в сетях IPv6. Поддержка IPv4 в нем полностью отсутствует. В RIPng применяются такие же таймеры, процедуры, типы сообщений и метрики, что и в RIPv2.

Для отправки Update-сообщений в качестве источника пакета, маршрутизаторы RIPng используют link-local адреса интерфейсов. Эти сообщения называются RIPng response и по умолчанию рассылаются каждые 30 секунд на специальный групповой IPv6-адрес FF02::9, предназначенный для всех маршрутизаторов, поддерживающих протокол RIPng. Сообщения RIPng response инкапсулируются в протокол UDP и используют порт 521.

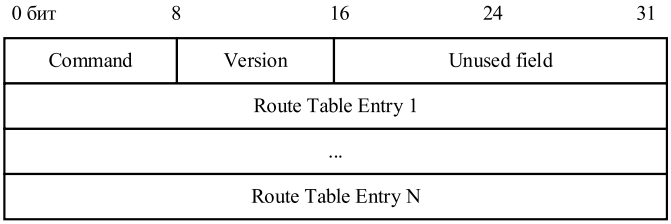

Рисунок 2.10.4 – Формат сообщения RIPng

Сообщение протокола RIPng содержит следующие поля:

- **Command** – поле команд. Значение команды 1 означает запрос на отправку всей или части таблицы маршрутизации, значение 2 означает ответное сообщение, содержащее полную или часть таблицы маршрутизации. Это сообщение может быть отправлено в ответ на сообщение с запросом или отправлено без запроса;
- **Version** – поле версии протокола, для RIPng значение поля всегда установлено в 1;
- **Unused field** – неиспользуемое поле, заполняется нулями;
- **Route Table Entry (RTE)** – остальная часть сообщения RIPng содержит одну или несколько записей таблицы маршрутизации (RTE).

Каждое поле RTE имеет следующий формат:

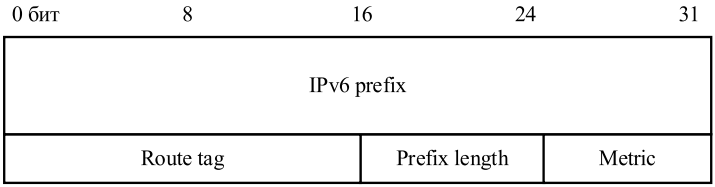

Рисунок 2.10.5 – Формат поля RTE сообщения RIPng

- **Prefix IPv6 –** 128-битный IPv6 префикс сети назначения;
- **Route tag –** поле используется аналогично тому, как используется 16-битное поле тега в RIPv2, для передачи атрибутов внешнего маршрута через домен RIP;
- **Prefix length** – поле, указывающее в битах маску IPv6 адреса в поле префикса IPv6;
- **Metric –** поле метрики указывает текущую метрику префикса от 1 до 15 включительно. Как и в RIPv2 метрика 16 считается недостижимой.

В современных сетях протокол RIP не применяется в качестве основного протокола маршрутизации, т. к. он уступает более современным протоколам, например, таким как OSPF. Небольшая метрика в 15 транзитных участков не

дает применять его в больших сетях. Главное преимущество протокола RIP заключается в простоте его конфигурирования.

Одна из проблем RIP это медленная сходимость, то есть изменения, произошедшие на одном из участков сети, распространяются очень медленно через остальную сеть. Основная проблема RIP – нестабильность, которая означает, что сеть, работающая по протоколу RIP, может стать нестабильной ввиду появления недостижимых или петлевых маршрутов. Ограничение участков в 15 увеличивает стабильность, но не решает всех проблем.

2.10.2. Протокол маршрутизации OSPF

Протокол динамической маршрутизации OSPF относится к категории протоколов маршрутизации внутреннего шлюза (Interior Gateway Protocol – IGP), которые предназначены для работы внутри автономной системы AS. Автономная система (AS) – это сеть под административным контролем одной организации, сеть каждого интернет провайдера является автономной системой.

Протокол OSPF формирует информацию о топологии сети, используя анонсы состояния канала LSA (Link State Advertisement), делая лавинную рассылку (flooding) анонсов LSA, OSPF доставляет информацию о топологии сети всем маршрутизаторам OSPF, чтобы у каждого из них была такая же информация о сети. Получая анонсы LSA, маршрутизаторы формируют у себя базу данных состояния каналов LSDB (Link State Date Base). Каждый анонс LSA – это структура данных, содержащая немного специфической информации о топологии сети. База LSDB – это просто коллекция всех анонсов LSA, полученных маршрутизатором.

Рисунок 2.10.6 дает представление о процессе лавинной рассылки, осуществляемой маршрутизатором R8 для передачи анонса LSA. Анонс маршрутизатора R8 описывает сам маршрутизатор и существующую подсеть 172.16.3.0/24.

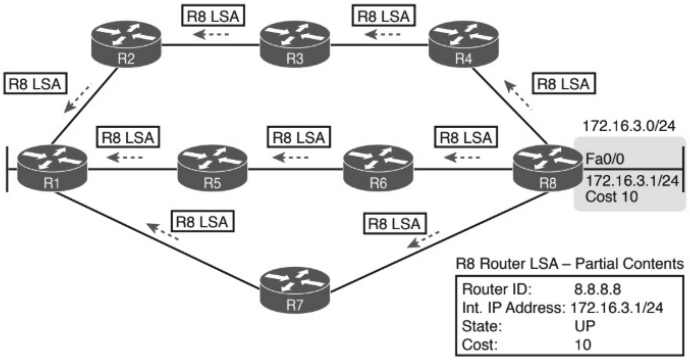

Рисунок 2.10.6 – Процесс лавинной рассылки анонсов LSA

Когда R8 публикует анонс LSA о себе, а другие маршрутизаторы перенаправляют анонс дальше, пока у всех не будет по экземпляру анонса. Процесс лавинной рассылки предотвращает циклическую передачу. Перед передачей анонса маршрутизатор запрашивает информацию у соседей, есть ли у них уже данные анонсы, если он есть, то передача отменяется.

В результате лавинной рассылки анонса о состоянии каналов на каждом маршрутизаторе будет одинаковый экземпляр базы LSDB, но в таблицу маршрутизации IP этот процесс наилучшие маршруты не добавит. Для получения маршрутов маршрутизаторы осуществляют математические вычисления, применяя к базе LSDB математический алгоритм кратчайшего пути Дейксты (Dijkstra Shortest Path First – SPF). Этот алгоритм анализирует базу LSDB и создает маршруты, добавляемы маршрутизаторами в таблицы маршрутизации IP. Маршруты включают адрес подсети и маску, исходящий интерфейс и IP-адрес следующего транзитного маршрутизатора.

Протокол OSPF использует концепцию соседских отношений. Соседские отношения позволяют соседним маршрутизаторам обмениваться базами данных LSDB. Соседи OSPF – это маршрутизаторы, находящиеся на одном канале связи. Для того, чтобы стать соседями, маршрутизаторы должны обменяться сообщениями OSPF Hello и согласиться стать соседями.

Пакеты Hello инкапсулируются заголовком IP с типом протокола 89 и посылаются на групповой IP-адрес 224.0.0.5, предназначенный для всех маршрутизаторов, работающих по протоколу OSPF. Hello пакеты также выступают в роли keepalive-пакетов для проверки доступности, и отправляются по умолчанию каждые 10 секунд.

После получения пакета Hello, и пройдя все промежуточные состояния установления соседских отношений, на последнем этапе маршрутизаторы

готовы обменяться друг с другом базами LSDB, чтобы достичь состояния полной синхронизации.

Сначала маршрутизаторы обмениваются списками анонсов LSA из своих баз данных. Каждый маршрутизатор проверяет, какие из анонсов LSA у него уже есть, а затем запрашивает у соседа только те анонсы, которых у него еще нет. Для этого используются разные типы сообщений OSPF, представленные в таблице 2.10.1.

Таблица 2.10.1 – Типы сообщений протокола OSPF

Тип	Название пакета	Описание
1	Hello	Обнаруживает соседей и строит соседство между ними.
2	DBD (Database Description)	Проверяет синхронизацию базы данных между маршрутизаторами
3	LSR (Link-State Request)	Запрашивает определенные записи состояния канала от маршрутизатора к маршрутизатору
4	LSU (Link-State Update)	Отправляет специально запрошенные записи о состоянии канала
5	LSAck (Link-State Acknowledgement)	Подтверждает другие типы пакетов

На рисунке 2.10.7 представлен формат заголовка протокола OSPF.

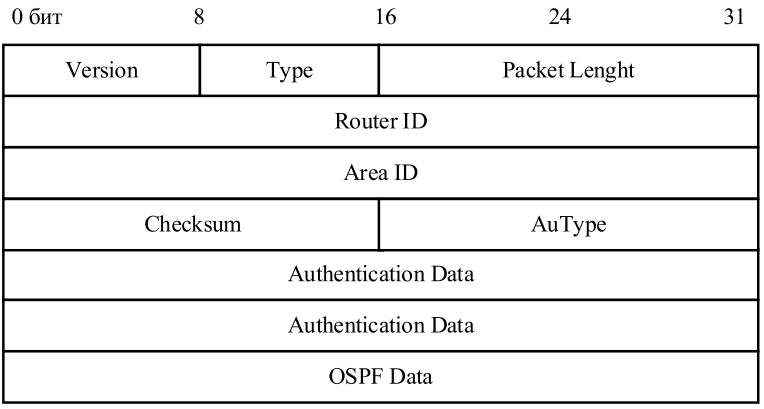

Рисунок 2.10.7 – Формат заголовка протокола OSPF

Обозначение полей заголовка OSPF:

- **Version** – версия протокола OSPF, для сетей IPv4 вторая версия OSPF;
- **Type** – тип пакета OSPF. Число 1 – пакет OSPF Hello, 2 – пакет DBD, 3 - пакет LSR, 4 - пакет LSU, 5 - пакет LSAck;

- **Packet length** – длина пакета OSPF в байтах, включает в себя заголовок;
- **Router ID** – идентификатор маршрутизатора, используется для уникального обозначения исходного маршрутизатора;
- **Area ID** – идентификатор области, в которой создан пакет;
- **Checksum** – Контрольная сумма, используется для проверки целостности пакета OSPF и для обнаружения ошибок при передаче;
- **AuType** – тип аутентификации, который используется между маршрутизаторами: 0 – аутентификация не используется, 1 – аутентификация открытым текстом, 2 – MD5-аутентификация;
- **Authentication data** – данные аутентификации содержат в себе пароли, использующиеся при аутентификации маршрутизаторов;
- **OSPF Data** – данные протокола OSPF.

В сетях OSPF со множественным доступом отношения соседства устанавливаются между всеми маршрутизаторами. Если бы все маршрутизаторы в состоянии соседства обменивались топологической информацией, это привело бы к рассылке большого количество копий LSA.

Для предотвращения проблемы рассылки копий LSA в сетях со множественным доступом используется концепция DR и BDR маршрутизаторов. Маршрутизаторами в сети выбираются выделенный маршрутизатор (DR) и запасной выделенный маршрутизатор (Backup Designated Router, BDR). Маршрутизатор DR играет ключевую роль в процессе обмена базами данных для таких сетей (рисунок 2.10.8).

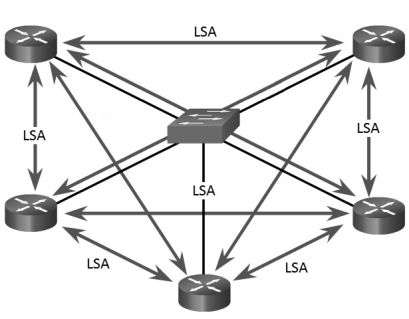

Рисунок 2.10.8 – Передача LSA-пакетов в сети без DR и BDR

Обмен базами данных осуществляется не между каждой парой маршрутизаторов, а между маршрутизатором DR и каждым другим маршрутизатором. Выделенный маршрутизатор гарантирует получение копии всех анонсов LSA другими маршрутизаторами. Протокол OPSF также использует резервный маршрутизатор BDR, поскольку выделенный

маршрутизатор DR столь важен для процесса обмена базами данных. Маршрутизатор BDR отслеживает состояние маршрутизатора DR, а при его отказе берет его роль на себя (рисунок 2.10.9).

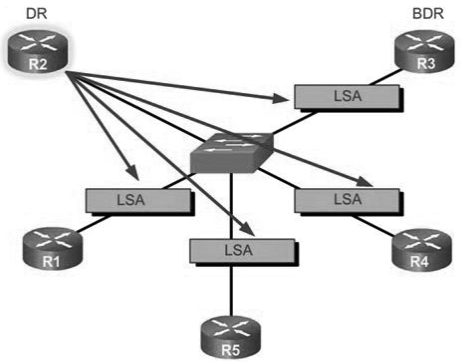

Рисунок 2.10.9 – Распространение LSA в сети с DR и BDR

Для определения лучшего маршрута в протоколе OSPF используется значение метрики. Протокол OSPF для определения метрики подсчитывает цену каждого интерфейса на всем маршруте до сети назначения, а также цену на основании ширины полосы пропускания канала связи и добавляет эту информацию в базу LSDB. Далее с помощью математического алгоритма Дейкстры, применяемого к базе LSDB, выбирает наилучший маршрут. Все пакеты, предназначенные DR и BDR, отправляются на адрес 224.0.0.6, а пакеты, предназначенные другим маршрутизаторам, отправляются на адрес 224.0.0.5.

Алгоритм Дейкстры вычисляет все маршруты для подсети назначения, т.е. все возможные маршруты от маршрутизатора до сети получателя. Если существует несколько маршрутов, то маршрутизатор сравнивает их метрики, выбирая маршрут с наилучшей (самой низкой) метрикой, и добавляет его к таблице маршрутизации.

Идентифицировав маршрут, протокол OSPF осуществляет вычисление следующим образом: суммируются стоимости всех исходящих интерфейсов OSPF на маршруте до сети назначения. В результате применения алгоритма Дейкстры к базе LSDB маршрутизатора, маршрутизатор R1 добавляет в таблицу маршрутизации наилучший вычисленный маршрут к подсети X.

На рисунке 2.10.10 представлен принцип суммирования стоимостей всех интерфейсов на пути маршрута.

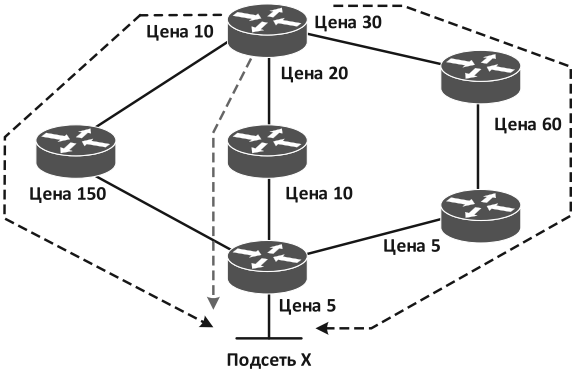

Рисунок 2.10.10 – Расчет метрики OSPF на основании суммарной цены маршрута

Работа сети OSPF организована с помощью зон (OSPF area). Разделение на зоны в протоколе OSPF позволяет снизить нагрузку на маршрутизаторы и оптимизировать работу сети. Если сеть слишком велика и находится целиком в одной области, то маршрутизатор должен будет хранить большую топологическую базу LSDB, занимающую много оперативной памяти. Каждая зона имеет свой уникальный идентификатор зоны. Среди зон внутри автономной системы существует специальная нулевая зона, называемая магистральной. Все не магистральные зоны должны быть соединены с магистральной зоной хотя бы через один граничный маршрутизатор (рисунок 2.10.11).

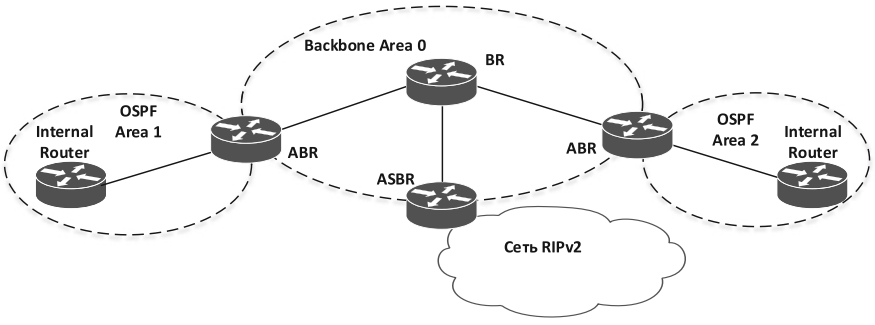

Рисунок 2.10.11 – Разделение на зоны сети OSPF

Маршрутизаторы внутри зоны содержат внутризоновую маршрутную информацию и разделены на следующие виды:

- **Area Border Router (ABR)** – граничный маршрутизатор зоны, соединяет зоны OSPF, суммируя информацию о зонах, является шлюзом для трафика, проходящего между зонами;
- **Backbone Router (BR)** – магистральный маршрутизатор, это маршрутизатор, находящийся внутри магистральной зоны. Магистральный маршрутизатор BR может также быть и пограничным маршрутизатором ABR.
- **Internal Router (IR)** – внутренний маршрутизатор зоны (не магистральной), у которого все интерфейсы находятся в одной зоне;
- **AS Boundary Router (ASBR)** – граничный маршрутизатор автономной системы AS. Обменивается трафиком с маршрутизаторами, находящимися в других автономных системах или с маршрутизаторами, работающими по другому протоколу маршрутизации. Граничный маршрутизатор автономной системы может находиться в любом месте автономной системы и быть внутренним, граничным или магистральным маршрутизатором.

Так как протокол OSPF работает с помощью алгоритма Дейкстры SFP, то даже изменение состояния интерфейса маршрутизатора запускает этот алгоритм. Обработка больших топологических баз алгоритмом Дейкстры занимает больше процессорного времени, а загрузка процессора растет экспоненциально при увеличении размера топологической базы. При разделении на зоны маршрутизатор будет просчитывать топологию только для своей зоны, также значительно уменьшится количество группового трафика, создаваемого пакетами OSPF Hello и анонсами LSA, рассылка которых будет ограничена границами зоны.

Проекты сетей с одной магистральной зоной OSPF хорошо подходят для небольших сетей, они позволяют избежать дополнительных сложностей и сделать сеть немного проще. В больших сетях использование нескольких зон улучшает работу протокола OSPF:

- Меньший размер базы LSDB в каждой зоне требует меньше памяти;
- Процессорам маршрутизаторов требуется меньше циклов на обработку меньших баз LSDB зон по алгоритму SPF, сокращаются дополнительные затраты процессора и улучшается время конвергенции;
- Изменения в сети (отказ и восстановление каналов связи) требуют вычислений алгоритма SPF только на тех соединенных с областью маршрутизаторах, на которых канал связи изменил состояние, что сокращает количество маршрутизаторов, на которых повторно запускается протокол SPF;

- Между областями можно передавать меньше информации в анонсах, сокращая ширину полосы пропускания, необходимую для передачи анонсов LSA.

Для работы в сетях IPv6 предназначен протокол OSPFv3 (RFC 5340), появившийся в 2008 году. OSPFv3 имеет некоторые различия в анонсах LSA по сравнению с OSPFv2. Для динамического поиска соседей маршрутизаторы OSPFv3 также используют пакеты OSPF Hello. Пакеты OSPFv3 отправляются с link-local адресов. Пакеты Hello посылаются на групповой IPv6-адрес FF02::5, предназначенный для всех маршрутизаторов, работающих по протоколу OSPFv3. Аутентификации в самом OSPFv3 нет, протокол использует аутентификацию IPv6.

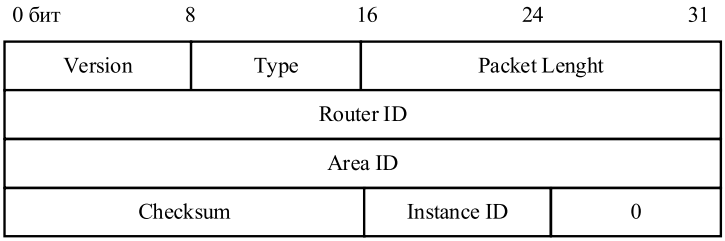

Рисунок 2.10.12 – Формат заголовка пакета OSPFv3

Все поля в OSPFv3 имеют такие же значения, как и у OSPFv2. Если у маршрутизатора запущено 2 и более процессов OSPF, то в OSPFv3 оба процесса могут совместно использовать один интерфейс, для этого необходимо поле идентификатора экземпляра процесса OSPF – Instance ID. Заголовок OSPFv3 включает в себя поле идентификатора экземпляра, чтобы идентифицировать пакеты OSPFv3, предназначенные для конкретного экземпляра процесса OSPFv3.

2.10.3. Внешний протокол маршрутизации BGP

Протокол маршрутизации граничного шлюза BGP (Border Gateway Protocol) относится к категории протоколов маршрутизации внешнего шлюза (Exterior Gateway Protocol – EGP), которые предназначены для маршрутизации трафика между автономными системами AS провайдеров и различных компаний, связывая их в единую сеть Интернет. На рисунке 2.10.13 предприятие анонсирует провайдеру свой открытый префикс, сеть класса C 192.0.2.0/24. Провайдер анонсирует далее этот префикс другим провайдерам в сети Интернет, чтобы все в Интернете узнали, как перенаправлять пакеты к этой сети.

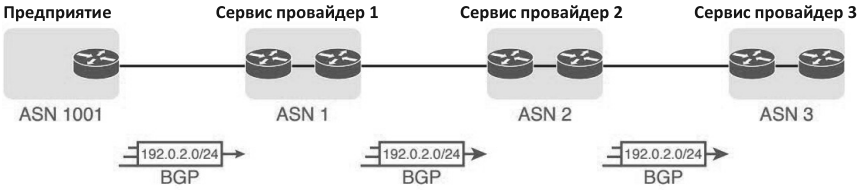

Рисунок 2.10.13 – Использование протокола BGP между автономными системами

В протоколе BGP анонсируемые блоки адресов называются префиксами, по причине того, что на практике протокол BGP редко анонсирует блоки адресов, такие же маленькие, как индивидуальные подсети.

Автономные системы AS играют важную роль в протоколе BGP. Автономная система (AS) – это сеть под административным контролем одной организации. Протокол BGP использует номера автономных систем ASN (AS Number) для многих целей, включая часть процесса выбора наилучшего маршрута, а также они используются механизмом предотвращения петлевых соединений.

Протокол BGP разделяется на внешний eBGP (external BGP) и внутренний iBGP (internal BGP). Чтобы протокол BGP работал в глобальном интернете, он должен сначала осуществить обмен префиксной информацией между автономными системами (внешний BGP). Но чтобы анонсировать префиксы по каналам связи в других частях сети провайдера ISP, он должен анонсировать эти префиксы другим маршрутизаторам в той же автономной системе (внутренний BGP), так как сеть провайдера может покрывать целый город или даже область и использует множество маршрутизаторов.

В основном протокол eBGP подразумевает анонсирование маршрутов между двумя разными номерами ASN. Протокол iBGP подразумевает анонсирование маршрутов к другим маршрутизаторам в той же автономной системе и использует несколько иные правила и подходы для анонсирования префиксов (рисунок 2.10.14).

Рисунок 2.10.14 – Использование протоколов eBGP и iBGP

Протокол BGP является дистанционно векторным протоколом. Отличие от обычных дистанционно векторных протоколов заключается в том, что вместо вектора расстояния используется вектор пути, содержащий номера всех автономных систем, через которые нужно пройти для достижения сети назначения. Для выбора наилучшего маршрута протокол BGP не использует единую концепцию метрик как в протоколах IGP. Вместо этого он использует атрибуты пути. Протокол BGP анонсирует каждый префикс наряду со списком различных атрибутов пути. Атрибуты пути – это различные факты о маршруте для достижения определенной подсети. Одним из важнейших атрибутов протокола BGP является атрибут пути AS Path. Атрибут AS Path – это передаваемый с маршрутами BGP атрибут пути, содержащий список номеров AS на маршруте. Процесс выбора наилучшего маршрута считает лучшим более короткий атрибут AS Path.

AS-path формируется следующим образом: когда маршрутизатор анонсирует маршрут своему внешнему соседу, он добавляет в список AS-path номер своей автономной системы AS. Маршрутизатор в соседней AS получает анонсируемый прификс и передает его в следующую автономную систему, добавляя в начало списка AS-path уже номер своей автономной системы. Маршрутизатор в тупиковой AS получает анонсируемый префикс и номера AS, через которые доступна эта сеть, и добавляет эту информацию в свою таблицу BGP (рисунок 2.10.15).

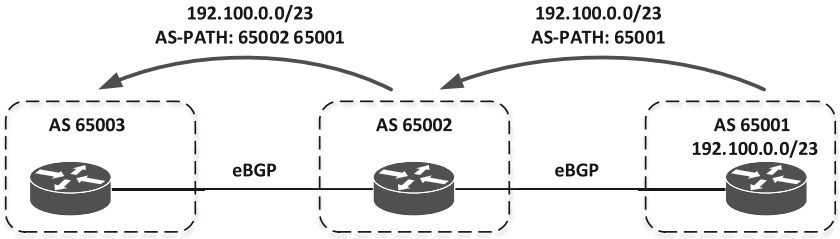

Рисунок 2.10.15 – Процесс анонсирования префиксов в протоколе BGP

Все сообщения BGP имеют следующий формат заголовка:

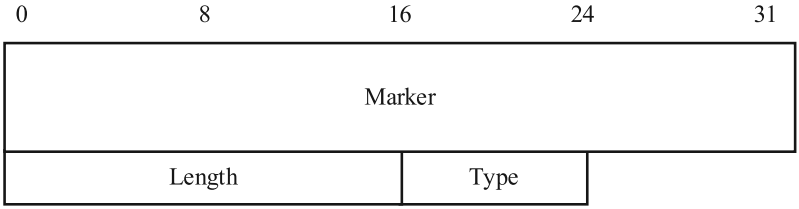

Рисунок 2.10.16 – Формат заголовка сообщения BGP

- **Marker** – поле маркера, которое включено в заголовок для совместимости. Размер поля 16 байт, маркер может использоваться для обнаружения потери синхронизации в работе BGP-партнеров;
- **Length** – поле длины имеет два октета и определяет общую длину сообщения в октетах, включая заголовок;
- **Type** – тип передаваемого сообщения: OPEN, UPDATE, NOTIFICATION или KEEPALIVE.

Протокол BGP использует концепцию соседских отношений, но не использует динамическое обнаружение и установление BGP сессии с соседом, так как маршрутизаторы должны быть сконфигурированы вручную. Для передачи своих сообщений между двумя партнерами BGP используется протокол TCP и порт 179. Когда настраивается протокол BGP, он открывает порт 179 и ожидает входящие запросы на установление соединения от других маршрутизаторов.

После того, как TCP-сессия установлена, маршрутизаторы BGP начинают обмен сообщениями OPEN. OPEN – первый тип сообщений BGP, они отсылаются только в самом начале BGP-сессии для согласования параметров. В нём передаются версия протокола, номер AS, Hold Timer и Router ID. Чтобы BGP-сессия установилась, должны соблюдаться следующие условия:

- Версии протокола должна быть одинаковой;
- Номера AS в сообщении OPEN должны совпадать с настройками на удалённой стороне;
- Router ID должны различаться.

Получив от соседа OPEN, маршрутизатор BGP отправляет свой OPEN, а также сообщение KEEPALIVE, информирующее о том, что OPEN от соседа получен – это сигнал переходить к следующему состоянию – ESTABLISHED. После всех шагов маршрутизаторы переходят в стабильное состояние ESTABLISHED, означающее, что запущена правильная версия BGP и все настройки идентичны.

После того как все состояния успешно пройдены, BGP-маршрутизаторы регулярно будут рассылать сообщения KEEPALIVE. Эти сообщения означают, что маршрутизатор находится в рабочем состоянии. Это происходит с истечением таймера Keepalive – по умолчанию 60 секунд.

Для обмена информацией протокол BGP использует сообщения об обновлении UPDATE. Как только партнерские отношения BGP установлены, партнеры BGP начинают посылать сообщения об обновлении, содержащие информацию о префиксе/длине и соответствующие атрибуты пути PA (Path Attribute).

Атрибуты пути бывают четырех видов:

- **Стандартными обязательными** (Well-known mandatory) – все маршрутизаторы BGP должны распознавать эти атрибуты. Данные атрибуты должны присутствовать во всех обновлениях Update;
- **Стандартными на усмотрение оператора** (Well-known discretionary) - все маршрутизаторы BGP должны распознавать эти атрибуты. Данные атрибуты могут присутствовать в обновлениях Update, но их присутствие не обязательно;
- **Опционными переходными** (Optional transitive) – данные атрибуты могут не распознаваться всеми реализациями BGP. Когда маршрутизатор BGP не распознает атрибут, он помечает обновление как частичное и передает его дальше партнерам BGP, сохраняя не распознанный атрибут;
- **Опционными непереходными** (Optional non-transitive) – данные атрибуты могут не распознаваться всеми реализациями BGP. Когда маршрутизатор BGP не распознает атрибут, то этот атрибут игнорируется и не передается дальше партнёрам BGP.

Существуют следующие разновидности атрибутов:

Атрибут пути ORIGIN – стандартный обязательный атрибут, который определяет происхождение маршрутной информации. Генерируется автономной системой, которая является источником маршрутной информации. Значение атрибута в этом случае может принимать следующие значения:

- **IGP** – задан вручную или получен по BGP;
- **EGP** – маршрут получен из устаревшего протокола EGP, который был полностью заменен протоколом BGP;
- **Incomplete** – информация достижимости сетевого уровня получена каким-то иным способом, чаще всего означает, что маршрут получен через редистрибьюцию.

Атрибут пути AS_PATH – стандартный обязательный атрибут, который описывает через какие автономные системы надо пройти, чтобы дойти до сети назначения. Номер AS добавляется при передаче обновления из одной AS eBGP-соседу в другой AS.

Атрибут пути NEXT_HOP – стандартный обязательный атрибут, определяющий IP-адрес eBGP маршрутизатора следующей AS для достижения сети назначения. Атрибут меняется при передаче префикса в другую AS.

Атрибут пути MULI_EXIT_DISC (MED) – опционный непереходной атрибут. Величина этого атрибута может применяться при выборе одного из нескольких путей к соседней автономной системе. Маршрутизаторы внутри соседней автономной системы используют этот атрибут, но, как только

обновление выходит за пределы AS, атрибут MED отбрасывается. Чем меньше значение атрибута, тем более предпочтительна точка входа в автономную систему.

Атрибут пути LOCAL_PREF – стандартный атрибут на усмотрение оператора. Он используется BGP-маршрутизатором, чтобы сообщить своим BGP-партнерам в своей собственной автономной системе степень предпочтения объявленного маршрута. Выбирается та точка выхода, у которой значение атрибута больше.

Атрибут пути ATOMIC_AGGREGATE – стандартный атрибут на усмотрение оператора, который применяется для информирования партнеров о выборе маршрута, обеспечивающего доступ к более широкому списку адресов.

Атрибут пути AGGREGATOR – опционный переходной атрибут. Атрибут содержит последний код автономной системы, который определяет агрегатный маршрут (ASN) и IP-адрес BGP-маршрутизатора (RID), который сформировал этот маршрут.

Формат сообщения BGP Update представлен на рисунке 2.10.17.

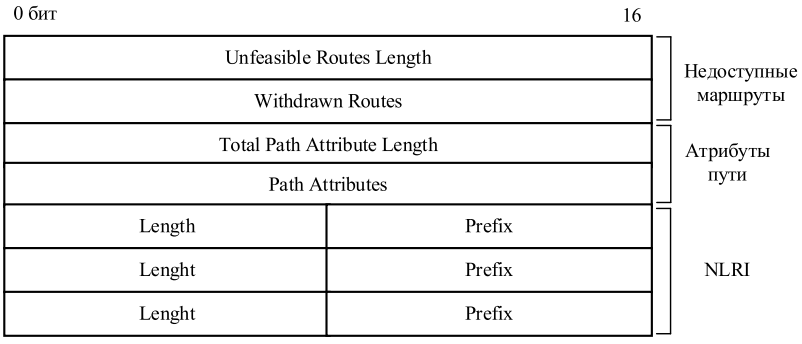

Рисунок 2.10.17 – Формат сообщения BGP Update

Поля сообщения имеют следующие значения:

- **Unfeasible Routes Length** – длина списка отмененных маршрутов. Если ни один маршрут не отменен, то длина писка равна 0;
- **Withdrawn Routes** – отмененные маршруты. Переменное поле, содержит список префиксов, которые стали недоступны. Если поле списка отмененных маршрутов равно нулю, то данное поле отсутствует;
- **Total Path Attribute Length** – полная длина списка атрибутов пути, нулевое значение указывает, что в сообщение об обновлении не включены атрибуты пути и NLRI;
- **Path Attributes –** поле переменной длины, в котором перечислены атрибуты пути, связанные с NLRI в следующем поле. Каждый атрибут

пути представляет собой переменной длины: (тип атрибута, длина атрибута, значение атрибута);

- **Network Layer Reachability Information (NLRI)** – поле переменной длины, которое содержит список IP префиксов, которые могут быть достигнуты по этому пути с помощью строки Length - Prefix (длина - префикс). Значение длины 0 указывает на префикс, который соответствует всем IP префиксам.

Сообщение BGP Update в анализаторе трафика Wireshark представлено на рисунке 2.10.18.

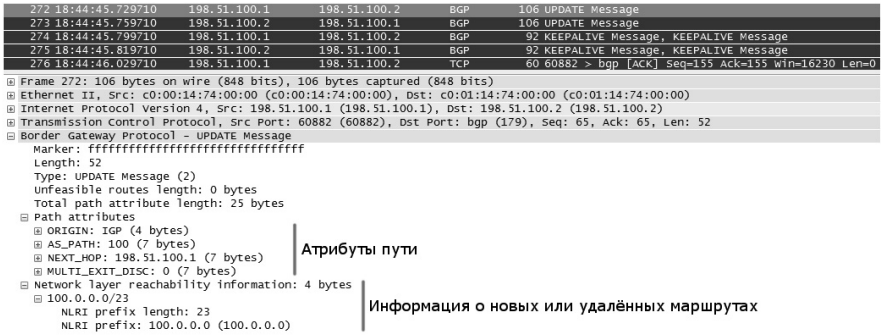

Рисунок 2.10.18 – Анализ трафика, содержащего сообщения BGP Update

Когда обнаружены какие-либо ошибки, то маршрутизаторами BGP отсылаются сообщения NOTIFICATION. BGP-сессия при этом немедленно разрывается.

Протокол BGP широко используется для маршрутизации трафика в глобальной сети Интернет, в настоящее время насчитывается около 817000 BGP маршрутов. Никакой другой протокол IGP не имеет таких механизмов фильтрации, распределения и балансировки трафика как BGP благодаря использованию специальных политик, основанных на картах маршрутов (Route Map).

2.11. Протокол управления передачей TCP

Протоколы транспортного уровня решают проблему правильной последовательности и гарантированной доставки сообщений. Основными протоколами этого уровня являются TCP (Transmission Control Protocol) и UDP (User Datagram Protocol).

Взаимодействие прикладных программ, использующих транспортные услуги протокола TCP или UDP, строится согласно модели "клиент-сервер", которая подразумевает, что одна программа (сервер) всегда пассивно ожидает

обращения к ней другой программы (клиента). Связь программы-клиента и сервера идентифицируется пятеркой:

1. Используемый транспортный протокол TCP или UDP;
2. IP-адрес сервера;
3. Номер порта сервера;
4. IP-адрес клиента;
5. Номер порта клиента.

2.11.1. Основные функции протокола TCP

Основная задача протокола TCP – обеспечение надежной передачи данных в сети. Его транспортный адрес в заголовке IP-сегмента равен 6.

Основные функции протокола TCP:

1. Реализует взаимодействие в режиме с установлением логического соединения;
2. Обеспечивает двунаправленную дуплексную связь;
3. Организует потоковый тип передачи данных;
4. Дает возможность пересылки части данных, как «экстренных»;
5. Для идентификации взаимодействующих сторон на транспортном уровне использует 16-битовые "номера портов";
6. Реализует принцип "скользящего окна" (sliding window) для повышения скорости передачи;
7. Поддерживает механизмы, обеспечивающие надежную передачу данных.

На рисунке 2.11.1 представлен формат заголовка TCP-пакета.

<table>
<tr><td colspan="2">Source Port
(Адрес порта источника)</td><td>Destination Port
(Адрес порта приёмника)</td></tr>
<tr><td colspan="3">Sequence Number
(Номер в последовательности)</td></tr>
<tr><td colspan="3">Acknowledgment Number
(Номер подтверждения)</td></tr>
<tr><td>Data Offset
(Смещение данных)</td><td>Reserved Control Bits
(Биты управления)</td><td>Window
(Размер окна)</td></tr>
<tr><td colspan="2">Checksum
(Контрольная сумма)</td><td>Urgent Pointer
(Указатель)</td></tr>
</table>

Options (Дополнительные данные заголовка)	Padding (Данные выравнивания)
Data (Данные)	

Рисунок 2.11.1 – Формат заголовка TCP-пакета

Source Port, Destination Port (Порт источника и порт приемника) – 16-битовые поля, содержащие номера портов, соответственно, источника и приёмника TCP-пакета. В таблице 2.11.1 дан список номеров портов для некоторых приложений.

Sequence Number (Номер в последовательности) – 32-битовое поле, содержимое которого определяет положение данных TCP-пакета внутри исходящего потока данных, существующего в рамках текущего логического соединения.

В момент установления логического соединения каждый из двух абонентов генерирует начальный номер для первого пакета в последовательности, основное требование к которому - не повторяться в промежутке времени, в течение которого TCP-пакет может находиться в сети. Абоненты обмениваются этими начальными номерами и подтверждают их получение. Во время отправления TCP-пакетов с данными, поле "номер в последовательности" содержит сумму начального номера и количества байт ранее переданных данных.

Acknowledgment Number (Номер подтверждения) – 32-битовое поле, содержимое которого определяет количество принятых данных из входящего потока к TCP-модулю, формирующему TCP-пакет.

Data Offset (Смещение данных) – 4-битовое поле, содержащее длину заголовка TCP-пакета в 32-битовых словах и используемое для определения начала расположения данных в TCP-пакете.

Флаг URG – 1 бит, установленное в 1 значение, которого означает, что TCP-пакет содержит важные данные. Обработке таких данных отдаётся наивысший приоритет.

Флаг ACK – 1бит, установленное в 1 значение, которого означает, что TCP-пакет содержит в поле «номер подтверждения» верные данные.

Флаг PSH – 1 бит, установленное в 1 значение, которого означает, что данные, содержащиеся в TCP-пакете, должны быть немедленно переданы прикладной программе, для которой они адресованы. Подтверждение для TCP-пакета, содержащего единичное значение во флаге PSH, означает, что и все предыдущие TCP-пакеты достигли адресата.

Флаг RST – 1 бит, устанавливается в 1 в TCP-пакете, отправляемом в ответ на получение неверного TCP-пакета. Также может означать запрос на переустановление логического соединения.

Флаг SYN – 1 бит, установленное в 1 значение, которого означает, что TCP-пакет представляет собой запрос на установление логического соединения. Получение пакета с установленным флагом SYN должно быть подтверждено принимающей стороной.

Флаг FIN – 1 бит, установленное в 1 значение, которого означает, что TCP-пакет представляет собой запрос на закрытие логического соединения и является признаком конца потока данных, передаваемых в этом направлении. Получение пакета с установленным флагом FIN должно быть подтверждено принимающей стороной.

Window (Размер окна) – 16-битовое поле, содержащее количество байт информации, которое может принять в свои внутренние буфера TCP-модуль, отправляющий партнеру данный TCP-пакет. Данное поле используется принимающим поток данных TCP-модулем для управления интенсивностью этого потока. Установив значение поля в 0, можно полностью остановить передачу данных, которая будет возобновлена, когда размер окна увеличится.

Checksum (Контрольная сумма) – 16-битовое поле, содержащее контрольную сумму, подсчитанную для TCP-заголовка, данных пакета и псевдозаголовка. Псевдозаголовок включает в себя ряд полей IP-заголовка и имеет показанную на рисунке 2.11.2 структуру.

Urgent Pointer (Указатель) – 16-битовое поле, содержащее указатель (в виде смещения) на первый байт в теле TCP-пакета, начинающий последовательность важных (urgent) данных.

Options (Дополнительные данные заголовка) – последовательность полей произвольной длины, описывающих необязательные данные заголовка. Протокол TCP определяет три типа дополнительных данных заголовка:

1. конец списка полей дополнительных данных;
2. пусто (No Operation);
3. максимальный размер пакета.

Дополнительные данные последнего типа посылаются в TCP-заголовке в момент установления логического соединения для выражения готовности TCP-модулем принимать пакеты длиннее 536 байтов.

Source Address (Адрес источника)		
Destination Address (Адрес приёмника)		
Zero (нули)	Protocol (Идентификатор протокола)	TCP Length (Длина TCP пакета)

Рисунок 2.11.2 – Схема псевдозаголовка TCP-пакета

Протокол TCP устанавливает обязательное получение подтверждения от принимающей стороной о правильности полученных данных.

В протоколе TCP используется принцип "скользящего окна" (sliding window), который заключается в том, что каждая сторона может отправлять партнеру максимум столько байт, сколько партнер указал в поле "размер окна" заголовка TCP-пакета, подтверждающего получение предыдущих данных.

Принцип "скользящего окна" обеспечивает "опережающую" посылку данных с "отложенным" их подтверждением. Следует отметить недостаток этого механизма: если в течение некоторого времени не будет получено "отсроченное" подтверждение ранее отправленного пакета, то отправляющий TCP-модуль будет вынужден повторить посылку всех TCP-пакетов, начиная с неподтвержденного. Размер окна, как правило, определяется объемом свободного места в буферах принимающего TCP-модуля.

2.11.2. TCP порты

Для доставки TCP-пакета конкретному приложению, используется уникальный идентификатор - номер порта, 16-битное число от 1 до 65535, указывающее, какой программе предназначается пакет.

TCP порты используют определенный порт программы для доставки данных, передаваемых с помощью протокола управления передачей (TCP). TCP порты являются более сложными и работают иначе, чем порты UDP. В то время как порт UDP работает как одиночная очередь сообщений и как точка входа для UDP-соединения, точкой входа для всех соединений TCP является уникальное соединение. Каждое соединение TCP однозначно идентифицируется двумя точками входа.

Каждый отдельный порт сервера TCP может предложить общий доступ к нескольким соединениям, потому что все TCP соединения идентифицируются двумя значениями: IP-адресом и TCP портом.

Все номера портов TCP, которые меньше чем 1024 - зарезервированы и зарегистрированы в Internet Assigned Numbers Authority (IANA).

Номера портов UDP и TCP не пересекаются.

TCP программы используют зарезервированные или хорошо известные номера портов, как показано в таблице 2.11.1.

Таблица 2.11.1 – Номера портов TCP

TCP номер порта	Описание
21	FTP
22	SSH
23	TELNET
53	DNS
80	HTTP
443	HTTPS

2.11.3. Установление соединения TCP

Процесс установления TCP-соединения показан на рисунке 2.11.3. Процесс, работающий на одном хосте, хочет установить соединение с другим процессом на другом хосте. Хост, который инициирует соединение называется «клиентом», а другой узел называется «сервером».

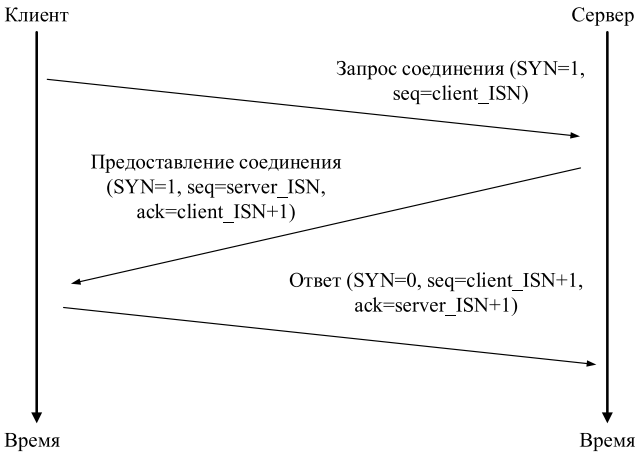

Рисунок 2.11.3 – Установление TCP соединения

Перед началом передачи каких-либо данных, согласно протоколу TCP, стороны должны установить соединение. Соединение устанавливается в три этапа.

Клиент отправляет SYN сегмент, указывая номер порта сервера, к которому клиент хочет подсоединиться, и исходный номер последовательности клиента (ISN).

Сервер отвечает своим сегментом SYN, содержащим исходный номер последовательности сервера. Сервер также подтверждает приход SYN клиента с использованием ACK (ISN + 1). На SYN используется один номер последовательности.

Клиент должен подтвердить приход SYN от сервера своим сегментов SYN, содержащий исходный номер последовательности клиента (ISN+1) и с использованием ACK (ISN+1). Бит SYN установлен в 0, так как соединение установлено.

После установления соединения TCP, эти два хоста могут передавать данные друг другу, так как TCP-соединение является полнодуплексным, они могут передавать данные одновременно.

2.11.4. Таймеры протокола TCP

Временные характеристики взаимодействия по протоколу TCP контролируют таймеры.

1.Таймер повторной передачи.

Таймер повторной передачи взводится значением RTO (Retransmission TimeOut - интервал до повторной передачи) в момент посылки TCP-пакета адресату. Если таймер окажется сброшенным в ноль до момента получения подтверждения пакета, то этот пакет должен быть послан вновь.

2. Таймер возобновления передачи.

В ходе взаимодействия двух TCP-модулей (A и B) вполне возможна следующая ситуация:

- TCP-модуль B уведомляет TCP-модуль A о невозможности приема от него данных, определяя размер окна равным 0;
- TCP-модуль A, имея данные для передачи, переходит в состояние ожидания от TCP-модуля B пакета с ненулевым размером окна;
- TCP-модуль B, у которого освободилось некоторое пространство в буферах, посылает модулю A TCP-пакет с ненулевым размером окна;
- адресованный модулю A пакет "теряется" по какой-либо причине и оба TCP-модуля переходят в состояние бесконечного ожидания.

Средством выхода из такого тупикового состояния и служит таймер возобновления передачи (persistence timer - "настойчивый" таймер). Он устанавливается в момент получения TCP-пакета с нулевым значением поля "Размер окна" в заголовке (типичное начальное значение для этого таймера - 5 секунд). Если до момента обнуления таймера не будет получено разрешение на возобновление передачи данных, то ожидающий разрешения TCP-модуль отправляет партнеру пакет, содержащий всего лишь 1 байт данных. По реакции

партнера, возвращающего пакет с нулевым/ненулевым значением размера окна, TCP-модуль продолжает ожидание или возобновляет посылку данных.

3. Таймер закрытия связи.

Протокол TCP предусматривает следующий простой прием предотвращения появления в сети TCP-пакетов, не имеющих адресатов. После закрытия логического соединения между партнерами, номера портов, использовавшихся в этом соединении, остаются еще некоторое время действительными. Это дает возможность, находящимся в сети TCP-пакетам добраться до места назначения, где они будут отброшены. Величина этого интервала равна удвоенному времени жизни IP-сегмента, обычно 30 секунд.

4. Таймеры поддержки соединения.

Каждый TCP-модуль, участвующий в логическом соединении, через фиксированный промежуток времени (keep-alive timer), равный обычно 45 секундам, периодически отправляет партнеру не содержащие данных TCP-пакеты и ждет их подтверждения. Каждое полученное подтверждение говорит об активном состоянии соединения. Если же в течении определенного интервала времени (idle timer), равного обычно 360 секудам, не будет получено ни одного подтверждения, то логическое соединение считается закрытым.

2.11.5. Алгоритмы повышения эффективности протокола TCP

2.11.5.1. Задержка подтверждения

Задержка отсылки подтверждения принятого пакета используется для сокращения числа TCP-пакетов, которыми обмениваются взаимодействующие стороны.

В ситуации без задержки TCP-модуль на стороне сервера, приняв пакет с данными и разместив их в своем буфере, сразу же отвечает подтверждающим пакетом, содержащим в своем заголовке и некоторый уменьшенный размер окна для приема последующих данных. Спустя некоторое время данные из буфера передаются серверной части прикладной программы. Освобождение места в буфере заставляет TCP-модуль отправлять партнеру на стороне клиента TCP-пакет с новым увеличенным размером окна. Тем временем прикладная программа, обработав полученные данные, передает результат TCP-модулю для отсылки его клиенту, для чего модуль формирует еще один пакет. Одна транзакция потребовала от TCP-модуля на стороне сервера посылки трех TCP-пакетов.

Введение же задержки при отсылке подтверждающего TCP-пакета позволяет в ряде случаев уменьшить количество пакетов с трех до одного, содержащего сразу подтверждение, новый размер окна и результирующие данные.

Для того, чтобы введение задержки сказывалось минимальным образом на приложения, предъявляющие жесткие требования к пропускной способности сети, задержка устанавливается нулевой при условии, что размер окна изменяется более чем на удвоенный максимальный размер TCP-пакета.

2.11.5.2. Исключение малых окон

Когда прикладная программа, использующая TCP-сервис, "выбирает" из буфера обслуживающего ее TCP-модуля пришедшие для нее данные малыми порциями. Это приводит к генерации TCP-модулем большого количества TCP-пакетов, содержащих в своих заголовках малую величину размера окна, что в свою очередь приводит к генерации на передающей данные стороне многих TCP-пакетов малого размера.

Для предотвращения данной ситуации используется следующий прием: TCP-пакет, информирующий посылающую данные сторону об увеличении размера окна, формируется только при выполнении одного из двух условий:

1. свободное место в буфере принимающего данные TCP-модуля увеличилось по крайней мере на четверть размера этого буфера;
2. свободное место увеличилось по крайней мере на максимальный размер TCP-пакета.

2.11.5.3. Исключение коротких TCP-пакетов

Для предотвращения отправки данных пакетами малого размера предлагается:

- самая первая порция данных отправляется TCP-модулем сразу же при поступлении "коротким TCP-пакетом;
- все последующие накапливаются в буфере TCP-модуля, пока их общий объем не составит максимального размера TCP-пакета или не будет получено подтверждение предыдущей посылки.

2.11.5.4. Алгоритм медленного старта

Необходимо согласование темпа передачи TCP-пакетов с возможностями их приема на узле-адресате. Задачу согласования решает алгоритм медленного старта, постепенно повышающий темп передачи данных от медленного до "оптимального", при котором нет повторных передач TCP-пакетов. Алгоритм использует так называемое "окно перегруженности" (congestion window), используемое на передающей стороне для определения максимального объема передаваемых данных вместо размера, получаемого от принимающей стороны в поле окна подтверждающего пакета.

Размер "окна перегруженности" определяется на передающей стороне путем постепенного его увеличения до момента появления повторных передач,

размер этого окна никогда не превышает размера окна на принимающей стороне. Однажды определенный размер "окна перегруженности" остается неизменным, пока вновь не появятся повторные передачи, однако периодически делаются попытки и увеличить этот размер.

2.12. Протокол пользовательских дейтаграмм UDP

Протокол UDP **(**User Datagram Protocol – протокол пользовательских дейтаграмм) определен в стандарте RFC768. UDP используется для быстрой, но ненадежной транспортировки данных между хостами.

2.12.1. Основные функции протокола UDP

UDP протокол работает без установления соединения, не гарантирует доставку и правильную последовательность дейтаграмм.

Приложения, работающие в реальном времени, используют UDP, длина заголовка которого составляет 8 байт (рисунок 2.12.1).

Source Port (Адрес порта источника)	Destination Port (Адрес порта назначения)
Length (Длина дейтаграммы)	Checksum (Контрольная сумма)
Data octets (Октеты данных)	

Рисунок 2.12.1 – Формат UDP дейтаграммы

Source Port (Адрес порта источника) – номер порта, который используется процессом, выполняющимся в хосте сервера. Он равен 16 битам это означает, что номер порта может быть в пределах от 0 до 65 535. Если хост источника — это клиент (клиент, посылающий зарос), номер порта в большинстве случаев – это кратковременный номер порта, затребованный процессом и выбранный работающим на хосте источника программным обеспечением UDP.

Destination Port (Адрес порта пункта назначения) – номер порта, используемый процессом в хосте пункта назначения. Он также имеет 16 бит длины. Если хост пункта назначения – это сервер (клиент, посылающий запрос), то номер порта в большинстве случаев хорошо известен. Если хост пункта назначения – это клиент (сервер, посылающий отклик), номер порта в большинстве случаев кратковременный. В этом случае сервер копирует кратковременный номер порта, он получен в пакете запроса.

Length (Длина) – поле размером 16 бит, которое определяет полную длину UDP дейтаграммы, включая заголовок.

Checksum (Контрольная сумма) – поле размером 16 бит, контрольная сумма заголовка, псевдозаголовка и данных дейтаграммы.

В протоколе UDP контрольная сумма включает в себя три части: псевдозаголовок, заголовок UDP и данные, которые поступили от прикладного уровня.

Псевдозаголовок – это часть заголовка IP-пакета, в котором дейтаграмма пользователя инкапсулирована и некоторые поля, заполненные нулями (рисунок 2.12.2).

Вычисление контрольной суммы на стороне передатчика происходит в 8 шагов:

1. Добавляется псевдозаголовок к пользовательской дейтаграмме UDP.
2. Происходит заполнение нулями поле контрольной суммы.
3. Все биты разделяются на 16-битовые слова. Если полное число байтов четное - один байт заполнения (все нули).
4. Это заполнение делается только с целью вычисления контрольной суммы и в дальнейшем будет удалено.
5. Сложение всех 16-битовых секций с использованием арифметики с дополнением единиц.
6. Дополнение результата (изменение нулей на единицы, а все единицы на нули). Это16-битовое число вставляет в поле контрольной суммы.
7. Удаление псевдозаголовока и всех дополнительных заполнений.
8. Передача UDP-сегмента к IP программному обеспечению для инкапсуляции.

<table>
<tr><td colspan="3">Source address
(Адрес источника)</td></tr>
<tr><td colspan="3">Destination address
(адрес приёмника)</td></tr>
<tr><td>Zero
(нули)</td><td>Protocol
(Идентификатор протокола)</td><td>UDP length
(Длина UDP дейтаграммы)</td></tr>
</table>

Рисунок 2.11.2 – Псевдозаголовок UDP дейтаграммы

В отличии от передатчика, вычисление контрольной суммы на стороне приёмника происходит за 6 шагов:

1. Добавляется псевдозаголовок к пользовательской дейтаграмме UDP.
2. Если надо, то дополняется заполнение.
3. Разделение всех битов на 16-битовые секции.

4. Сложение всех 16-битовых секций с использованием арифметики с дополнением единиц.
5. Дополнение результата.
6. Если результат равен нулю, отбрасывается псевдозаголовок и любые заполнения и принимает UDP-дейтаграмму. Если результат какой-либо другой, пользовательская дейтаграмма удаляется.

2.12.2. Порты протокола UDP

UDP порты обеспечивают возможность отправки и получения сообщений UDP. UDP порт функционирует как одиночная очередь сообщений для получения всех дейтаграмм, предназначенных для программы, указанной номером порта протокола.

Все номера портов UDP, которые меньше чем 1024 – зарезервированы и зарегистрированы в Internet Assigned Numbers Authority (IANA). Каждый порт UDP идентифицируется под зарезервированным или известным номером порта. В таблице 2.12.1 приведен список известных номеров портов UDP, которые используются стандартные программы.

Таблица 2.12.1 – Список номеров портов UDP

UDP номер порта	Описание
53	Система доменных имен (DNS)
69	Простой протокол передачи файлов (TFTP)
137	Служба имен NetBIOS
138	Служба дейтаграмм NetBIOS
161	Простой протокол сетевого управления (SNMP)
520	Протокол маршрутной информации (RIP)

2.13. Система доменных имён DNS

DNS (Domain Name System – глобальная система доменных имён) – это система, позволяющая клиенту получить информацию, связанную с запрашиваемым доменным именем. Доменное имя – идентификатор ресурса сети интернет, который выражен в удобочитаемом виде. Вся информация в сети интернет передаётся по IP-адресам ресурсов, выраженным в числовом виде. Самый распространённый запрос, обсуживаемый DNS, является получение IP-

адреса ресурса по его доменному имени. Помимо этого, DNS – глобальная распределённая база данных, хранящая сотни миллионов имён и связанных с ними ресурсов.

Система доменных имён описана в наборе протоколов, разработанных в IETF. Основными их них являются спецификации RFC1034 (Доменные имена – концепции и возможности) и RFC1035 (Доменные имена – реализация и спецификация). Также выпущено более 500 спецификаций, определяющих дополнительный функционал системы.

Всякий раз, когда пользователь набирает имя веб-сайта в своём браузере, он сталкивается с работой DNS. На сегодняшний день нормальное функционирование сети Интернет невозможно без корректно работающего DNS. Стоит отметить, что для осуществления связи между компьютерами в сети система доменных имён не требуется, так как вся передача происходит в соответствии с IP-протоколом, где все адреса уже должны быть представлены в числовом виде.

Система DNS является иерархичной и распределённой. В мире не существует единой базы данных, которая хранила бы информацию обо всех доменных именах и соответствующих им IP-адресах. DNS имеет “зоновую” архитектуру, в которой, как правило, два или более серверов отвечают за информацию о конкретном домене.

Такое распределение позволяет обеспечить уникальность имён, а также распределить нагрузку и ответственность за работу системы между администраторами отдельных доменов. Каждый администратор, независимо от других, отвечает за содержимое, производительность и бесперебойную работу обслуживаемой им зоны.

2.13.1. Архитектура DNS

Иерархическая структура DNS ярко выражена в доменных именах. Например, http://www.mtuci.ru. состоит из четырёх частей, разделённых точками. Всякое доменное имя заканчивается точкой, но, обычно, она опускается и явно не указывается. Эта последняя точка означает корневую зону DNS. В таблице 2.13.1 описаны все домены, участвующие в определении IP-адреса сайта http://www.mtuci.ru.

Таблица 2.13.1 – Перечень доменов и их описание

.	Корневой домен (зона), в которой хранится информация о серверах, обслуживающих поддомены верхнего уровня: ru, com, org и другие
ru	Домен ru, в котором содержится информация обо всех поддоменах, зарегистрированных в нём

mtuci	Домен mtuci, в котором содержится информация обо всех поддоменах и именах серверов, зарегистрированных в этом нём
web	Непосредственно имя веб-сервера и соответствующие ему IP-адреса

Архитектура DNS состоит из серверов трёх основных типов: авторитетные серверы, резолверы, резолверы-заглушки.

2.13.2. Авторитетные серверы

Серверы этого типа обслуживают определённые зоны. В ответ на запрос какой-либо информации могут предоставить запрашиваемую информацию, выдать отрицательный ответ, если данные отсутствуют или вернуть перенаправление (referral), указав на серверы, обслуживающие поддомены, содержащиеся в запрашиваемом имени.

Авторитетные серверы делятся на два типа:

- первичный сервер, который непосредственное обслуживает данные зоны;

- вторичный сервер, который зеркалирует данные зоны.

Первичный сервер обслуживает администратор зоны, а вторичные серверы могут обслуживаться другими операторами и компаниями, которые специализируются на оказании услуг такого вида.

2.13.3. Резолверы

Серверы этого типа, также называются итеративными-резолверами. Они обслуживают множество клиентов и выполняют за них работу по трансляции имен. Для повышения производительности кэшируют полученные ответы.

2.13.4. Резолверы-заглушки

Используются для преобразования запроса приложения в DNS-запрос с дальнейшей передачей его серверу для последующей обработки. Полученный DNS-ответ преобразуют в удобочитаемый вид для приложения, которое отправило запрос.

2.13.5. Работа DNS

Доменные имена очень удобные для указания ресурса, но непригодны для установления соединений, в которых требуется IP-адрес сервера. Поэтому браузер, после получения названия ресурса, должен отправить DNS-запрос, в котором будет запрашиваться информация об IP-адресе ресурса с указанным именем.

Приложение, которому необходимо получить IP-адрес (например, браузер), должно опросить зоновые базы данных, которые отвечают за компоненты полного доменного имени, чтобы определить адрес сервера, отвечающего за указанный ресурс, после чего отправить ему запрос и получить в ответе всю необходимую информацию. Данный процесс достаточно трудоёмкий, из-за чего приложение, для определения IP-адреса, обращается к итеративному резолверу, который сам проделывает все указанные операции и кеширует полученные ответы.

Если у резолвера в кэше есть необходимая информация, то она сразу же будет возвращена приложению, если же её нет, то последует череда запросов, к базам данным DNS.

Перво-наперво будет отправлен запрос к одному из тринадцати корневых серверов. В них содержится только информация о доменах верхнего уровня (для сайта www.mtuci.ru – доменом является ru). Корневой сервер в ответе вернёт адрес ответственного сервера, а от одного из серверов, обслуживающих зону ru, будет получен ответ об адресе сервера, обслуживающего домен www.mtuci.ru. Этот сервер и вернёт в ответе на запрос IP-адрес сайта.

На рисунке 2.13.1 отображен процесс трансляции имени в DNS.

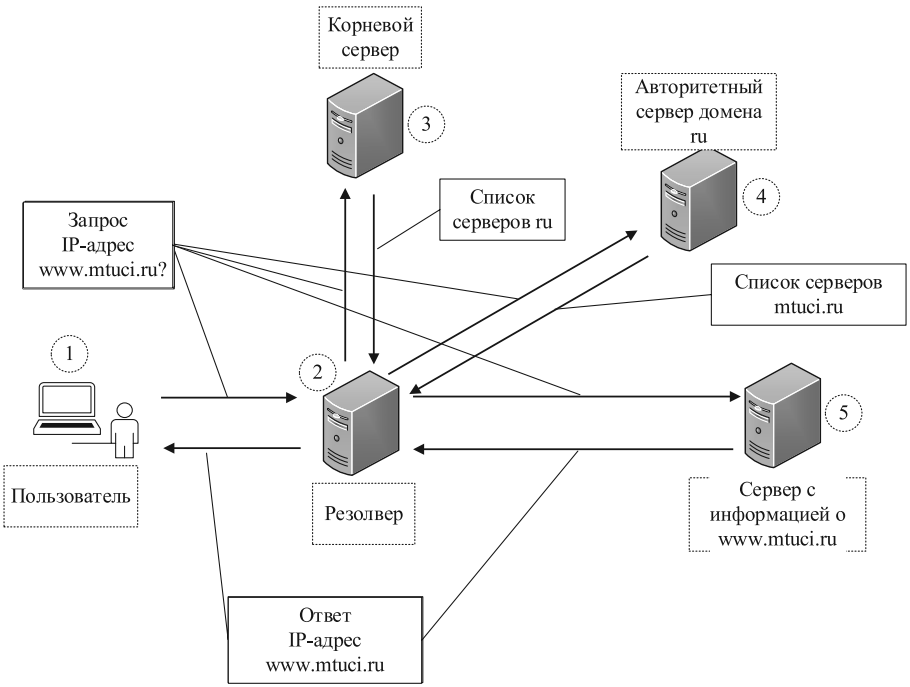

Рисунок 2.13.1 – Процесс трансляции имен в DNS

Из рисунка видно, что пользователь (1) хочет получить IP-адрес сайта www.mtuci.ru, для чего обращается к резолверу (2). У резолвера в кэше нет данных для быстрого ответа на запрос, из-за чего он поочерёдно опрашивает корневой сервер (3), авторитетный сервер домена ru (4) и сервер с информацией о необходимом ресурсе (5).

2.13.6. Зоны и записи

На уровне протокола используются файлы зоны, в которых описана информация о ресурсах. Любая DNS-запись имеет стандартизированный формат

```
владелец [класс] [TTL] тип данные
```

Владелец – относительное или полное имя записи. Если значение этого поля совпадает с именем зоны, то используется символ @, вместо указания полного имени зоны. Обязательное поле.

Класс – определяет класс, к которому принадлежит запись. Обычно, указывается только параметр IN, расшифровывается как “Internet” и говорит о том, запись принадлежит к классу интернет-ресурсов. Необязательное поле.

TTL –указывается время жизни конкретной записи в кэше других DNS-серверов. Для предотвращения кэширования необходимо указать 0. Необязательное поле.

Тип –определяет тип записи. Обязательное поле. В таблице 2.13.2 приведены основные типы записей, регулирующие их спецификации RFC и описание записей.

Данные – содержит в себе данные, которые определены форматом, определяемым типом записи. Обязательное поле.

Таблица 2.13.2 – Типы основных записей в файлах зоны DNS

Тип	RFC	Описание
A	RFC1035	IP-адрес четвёртой версии
AAAA	RFC3596	IP-адрес шестой версии
CNAME	RFC1035	Canonial Name (каноническое имя). Позволяет определить альтернативные имена (псевдонимы) хоста. Используется для единственной записи
DNAME	RFC6672	Non-Terminal DNS Name Redirection (нетерминальное перенаправление DNS-имён). Работает аналогично CNAME, но применимо к целой ветви записей

MX	RFC1035	Mail Exchanger (почтовый обмен). Определяет приоритет и имя почтового сервера, который обслуживает электронную почту для данной зоны
NS	RFC1035	Name Server (сервер имён). Определяет имя авторитетного сервера имен для зоны
PTR	RFC1035	Pointer (указатель). Определяет имя, соответствующее IP-адресу
SOA	RFC1035	Start of Authority (указание на авторитетность информации). Определяет имя зоны, контактный адрес электронный почты, частоту обновления и другие параметры зоны
SRV	RFC2872	Location of Server (расположение службы). Определяет дополнительные услуги, связанные с доменом
TXT	RFC1035	Descriptive text (описание). Используется для хранения общей информации о доменном имени, например, кто его размещает, контактное лицо, номера телефонов и др.

2.13.7. Запросы и ответы

Работа DNS происходит по схеме “запрос-ответ”. Для передачи используется протокол UDP, который не требует установления соединения, что уменьшает накладные расходы.

Запрос и ответ имеют одинаковую структуру, которая состоит из пяти разделов: заголовок, запрос, ответ, авторитет и дополнительная информация. В таблице 2.13.3 приведён формат запросов и ответов с описанием каждого из разделов.

Таблица 2.13.3 – Формат запросов и ответов с описанием разделов

Раздел	Описание
Заголовок	Всегда присутствует. Включает в себя поля, указывающие на разделы, которые присутствуют. Также указывает на то, является ли сообщение запросом или ответом
Запрос	Запрос к DNS
Ответ	Ответ на запрос
Авторитет	Содержит записи ресурсов, которые указывают на авторитетный сервер DNS, где следует продолжать поиск для получения ответа
Дополнительная информация	Содержит дополнительную информацию к запросу, но не является ответом

Запрос состоит только из заголовка и первой строки сообщения. Формат запроса описан в таблице 2.13.4. Ответ возвращает данные в соответствии с тем, какой тип записи требуется.

Таблица 2.13.4 – формат запроса

Элементы	Описание
QNAME	Доменное имя
QTYPE	Тип требуемой записи. Если указать ANY, то будут возвращены все записи, связанные с этим именем
QCLASS	Класс. Обычно указывается IN

2.13.8. Anycast

Для нормальной работы интернета устойчивость работы системы DNS имеет критическое значение. Для обслуживания одной зоны должно использовать минимум два сервера, находящиеся в различных сетях, что гарантирует работоспособность в случае отказа одного из компонентов.

Атаки “Отказ в обслуживании” DoS могут повлиять на устойчивость и производительность системы DNS. Для эффективного противостояния таким атакам необходимо иметь как можно более распределённую систему серверов.

Сложность увеличения числа обслуживающих серверов заключается в том, что число записей NS в зоне имеет пределы. Это связано с тем, что чем больше список серверов, тем больше размер ответа. Изначально максимально допустимым размером сообщения считалось 512 байт, что было связано с максимальным размером пакета UDP, который передаётся без фрагментации. Именно это послужило тому, что количество корневых серверов было ограничено 13. Ещё одной сложностью является то, что при увеличении числа записей уменьшается производительность системы.

Для решения этих сложностей и увеличения устойчивости к атакам DoS была разработана технология anycast (аникаст). Суть технологии заключается в том, что оператор анонсирует одну и ту же сеть (префикс IP и автономную систему) в различных частях интернета. Благодаря чему клиент может устанавливать связь с наиболее близкой в топологическом смысле сетью.

Описанная система может работать только по протоколу UDP, так как, если будет использоваться TCP, требующий установление соединения, то при каких-либо изменениях в топологии сети, кратчайший путь может быть изменён, из-за чего сеанс связи будет разорван.

2.13.9. DNSSEC

Помимо атаки типа “отказ в обслуживании” распространена атака с подменой и модификацией данных DNS. В процессе передачи данных DNS никак не защищает их от подмены и модификации.

Для борьбы с этой атакой было разработано расширение безопасности DNS, названные DNSSEC, описанное в стандартах RFC4033-4035 и RFC5155.

С помощью DNSSEC пользователь может убедиться в том, что полученные данные не были модифицированы в процессе передачи. Расширение основано на криптографии с использованием открытых ключей. Подлинность открытого ключа удостоверяет администратор родительской зоны. Для этого в записи DS публикуется хеш открытого ключа дочерней зоны, эта запись заверена цифровой подписью администратора этой родительской зоны. Ключ этого администратора заверен подписью администратора зона верхнего уровня. И так далее, пока не будет достигнута точка доверия, при которой подпись происходит ключом, которому пользователь абсолютно доверяет. Этим ключом является ключ корневой зоны.

Используются два типа ключей – KSK (Key Signing Key – ключ подписи ключей) и ZSK (Zone Signing Key – ключ подписи зоны). Ключ KSK является ключом долговременного пользования и является более устойчивым в криптографическом плане. При этом ключ ZSK служит для подписи записей самой зоны. Ключ ZSK подписывается ключом KSK.

2.13.10. Администрирование доменных имён верхнего уровня

Координацию корневого уровня DNS осуществляет ICANN (Internet Corporation for Assigned Names and Numbers) – частная некоммерческая компания. Задача координации заключается в двух частях: что может быть включено в качестве имени в корневую зону, а также как включить и обслуживать это имя.

В структуре ICANN существуют две организации поддержки – организация поддержки общих имён GNSO (Generic Names Supporting Organization) и организация поддержки национальных доменных имён ccNSO (Country Code Names Supporting Organization).

2.13.11. Корневой уровень DNS

Корневая зона содержит информация обо всех серверах, обслуживающих домены верхнего уровня:

- национальные (.ru, .рф);
- общего назначения (.com, .москва).

При получении запроса клиент получает ответ, на какие серверы DNS следует отправить последующий запрос для трансляции доменного имени.

Состав корневой зоны постоянно меняется. В среднем в зону вносится несколько изменений в неделю. К примеру, вносятся или удаляются сервера, обслуживающие домены верхнего уровня или добавляется новый домен верхнего уровня.

После проведения необходимых административных и технических процедур запрос на изменение подписывается цифровых ключом и направляется для авторизации аудитору. В настоящее время эту роль выполняет NTIA (National Telecommunications and Information Administration).

Затем изменения направляются в организацию, ответственную за редактирование и публикацию зоны в DNS. В настоящее время эту роль выполняет компания VeriSign. Компания публикует зону на скрытом мастер-сервере, после чего зона по протоколу TSIG, защищающего от модификации при передаче, распространяется на все корневые серверы.

2.13.12. Корневые серверы

Корневую зону обслуживают 13 серверов, которые называют корневыми. Имена корневых серверов начинаются с буквы латинского алфавита с общим окончанием – root-server.net. Например, b-root-server.net, чаще их называют по первой букве – B-сервер. Операторами, обслуживающими и отвечающими за эти сервера, являются различные организации. Список серверов и организаций, отвечающих за них представлен в таблице 2.13.5.

Таблица 2.13.5 – Корневые сервера и организации, обслуживающие их

Корневые серверы	Организация
A	VeriSign, Inc.
B	Information Sciences Institute
C	Cogent Communications
D	University of Maryland
E	NASE Ames Research Center
F	Internet Systems Consortium, Inc.
G	U.S. DOD Network Information Center
H	U.S. Army Research Lab
I	Netnod
J	VeriSign, Inc.
K	RIPE NCC
L	ICANN
M	WIDE Project

2.13.13. Глобализация корневой зоны DNS

Документ RFC920 (Требования к доменам), опубликованный в 1984 г., в дополнение к уже существующему домену .arpa, определил ещё пять доменов: .gov, .edu, .mil, .com и .org. В этом же документе были определены национальные домены, а также установлен их формат – двухбуквенный код таблицы ISO-3166. До введения RFC920 все хосты находились в домене .arpa.

До середины 90х годов корневая зона росла только за счёт национальных доменов, за исключением двух добавленных доменов - .int и .net.

В мае 1996 года был принят документ RFC1591 (Структура и делегирование в DNS), в котором были увеличены требования к созданию дополнительных доменов верхнего уровня.

Основной причиной введения новых требований являлась неудовлетворительная ситуация с международными доменами: .com, .org, .net. Регистрация имён и международных поддоменов, в то время, приняла глобальных характер, но любую регистрацию проводила одна организация Internic, что создало монополизацию. Регулирование было невозможно из-за международного характера проблема. Начали разрабатывать предложения по “открытию” рынка корневой зоны.

Первое предложение было разработано Джоном Постелом в 1996 году. Проект предусматривал создание нескольких комитетов, утверждающих образование новых доменов верхнего уровня. При этом Постел предлагал передать управление IANA (Internet Assigned Numbers Authority – Администрация адресного пространства интернет) организации ISOC (Internet Society, общество интернета), которая являлась основой для IETF. Предложение раскритиковали, так как оно понижало конкуренцию.

Другое предложение было разработано группой IAHC (International Ad Hoc Committee, международный специальный комитет). Оно содержало более сбалансированную модель управления, в сравнении с проектом Постела, но после опубликования предложений комитет прекратил своё существование. Многие из предложений комитета IAHC легли в основу деятельности созданной в 1998 году корпорации ICANN.

С 2000 года ICANN начало создание новых общих доменов верхнего уровня (gTLD – generic TLD): .aero, .biz, .museum, .pro и других.

В 2005 году GNSO начало рассмотрение вопроса о более масштабном создании общих доменов верхнего уровня. В результате были утверждены 19 рекомендаций по созданию новых доменов.

В июне 2011 года была запущена программа по созданию новых gTLD. Приём заявок был начат в 2012 году. Было получено 1930 заявок. На июнь 2014 году 272 заявки были реализованы в виде новых доменов верхнего уровня.

2.14. Протокол виртуального терминала TELNET

TELNET (TELetype NETwork – телетайпная сеть) – протокол удалённого доступа, использующийся для реализации двунаправленного текстового терминального интерфейса по сети. С помощью этого протокола клиент получает доступ к командной строке операционной системы на удалённом

хосте (сервере). Передача данных от пользователя происходит по протоколу TCP.

В спецификации RFC-854 определён виртуальный сетевой терминал NVT (Network Virtual Terminal), через который происходит общение между клиентом и сервером. Терминал представляет из себя специальную структуру данных, собственный алфавит, управляющие символы, а также порядок обмена управляющей информацией и данными, благодаря чему протокол абстрагируется от аппаратных особенностей и сам регулирует формат передаваемых данных.

При установлении соединения используется 23 порт, а сам процесс соединения регламентирован в спецификации, как и процесс обмена информацией.

Пример взаимодействия клиента и удалённого сервера изображен на рисунке 2.14.1. Протокол TELNET находится на прикладном уровне и не зависит от операционной системы (ОС), что делает его универсальным и независимым от типа ОС, но при этом снижается быстродействие.

При организации сеанса связи в реальных сетях у клиента и сервера могут существенно отличаться программно-аппаратные средства, из-за чего возникают конфликты, связанные с интерпретацией символьных кодов. Для решения этой проблемы используют NVT-интерфейс, который является неким преобразователем, изменяющим управляющие символы так, чтобы на удалённом сервере они были интерпретированы верно. В таблице 2.14.1 представлены символы управления, которые используются в NVT, а также их десятичное/шестнадцатеричное представление и назначение этих символов.

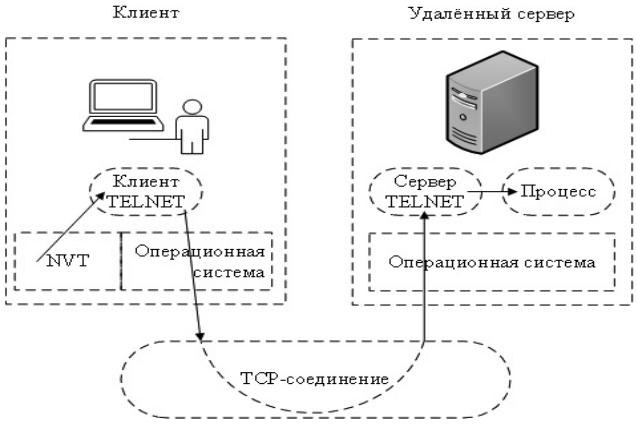

Рисунок 2.14.1 – Взаимодействие клиента и удалённого сервера с помощью TELNET

Таблица 2.14.1 – Символы управления, которые используются в NVT

Символ управления	Код (десят./ шест.)	Назначение
NUL	0/0x00	Пустая операция (null)
BEL	7/0x07	Звуковой сигнал(bell)
BS	8/0x08	Сдвиг на одну позицию назад (backspace)
HT	9/0x09	Горизонтальная табуляция (horizontal tab)
LF	10/0x0A	Перед строки (line feed)
VT	11/0x0B	Вертикальная табуляция (vertical tab)
FF	12/0x0C	Конец страницы (form feed)
CR	13/0x0D	Возврат каретки (carriage return)

Ещё одной функцией NVT-интерфейса является управление удалённым процессом. Это необходимо, так как у клиента и сервера могут использовать различные программно-аппаратные терминалы, из-за чего интерфейс использует собственные команды управления. При передаче команд всегда используются два кода, первый из которых – 255 в десятичном виде, что интерпретируется как начало управляющей команды (Interpret As Command). Список команд управления, их десятичных и шестнадцатеричный коды и назначение этих команд сведены в таблицу 2.14.2.

Таблица 2.14.2 – Список команд управления

Символы команд	Код (десят./ шест.)	Назначение
IAC	255/0xFF	Команда “Interpret As Command” (при наличии в данных байта со значением 255 передается дважды)
DON’T	254/0xFE	Запрет или запрос на установку параметров
DO	253/0xFD	Разрешение на установку параметров
WON’T	252/0xFC	Отказ в установке параметров
WILL	251/0xFB	Согласие на установку параметров
SB	250/0xFA	Начало согласования некоторых параметров (subnegotiation)
GA	249/0xF9	Продолжение передачи (go ahead)
EL	248/0xF8	Стирание предыдущей строки (erase line)
EC	247/0xF7	Стирание предыдущего символа (erase character)
AYT	246/0xF6	Идентификация сервера (are you there)
AO	245/0xF5	Прерывание вывода (abort output)
IP	244/0xF4	Прерывание процесса (interrupt process)

BRK	243/0xF3	Прерывание
DMARK	242/0xF2	Для передачи команды “SYNCH”. Команда “SYNCH” передается как “IAC+DMARK” с установкой в “*1*” бита “URGENT” (срочные данные) в заголовке сегмента TCP
NOP	241/0xF1	Пустая операция (no operation)
SE	240/0xF0	Завершение сеанса согласования параметров (subnegotiation end)
EOR	239/0xEF	Конец записи (end of record)

Важной особенность процедуры передачи команд управления является то, что все они сопровождаются командой SYNCH, очищающей буфер от обычных данных, из-за чего команды точно доводятся до сервера TELNET-протокола.

При инициализации соединения между клиентом и сервером происходит автоматическая установка необходимых внутренних и внешних параметров. При этом, в протоколе предусмотрена возможность установки дополнительных параметров для данного соединения, чтобы обеспечить более точного согласования параметров двух терминалов. Для этого используются специальные символы параметров. С их помощью можно настроить режим передачи данных, включить синхронизацию и так далее.

В таблице 2.14.3 представлены некоторые виды символов управления, их десятичных и шестнадцатеричный коды, спецификации RFC, в которых описано поведение протокола-TELNET при их использовании и краткое назначение этих символов.

Дополнительно к этому, механизм согласования параметров допускает использование различных версий TELNET-протокола у клиента и сервера.

Таблица 2.14.3 – Список символов управления

Символ параметра	Код (десят./ шест.)	RFC	Назначение
Transmit binary	0/0x00	856	Передача данных в двоичной форме
Echo	1/0x01	857	Эхо-пакет на принятые данные
Supress-Ga	3/0x03	858	Отмена команды “GA” после передачи данных
Status	5/0x05	859	Запрос параметров TELNET с удаленного терминала

Timing-Mark	6/0x06	860	Запрос вставки временных маркеров для синхронизации взаимодействующих процессов
Terminal-Type	24/0x18	884	Запрос типа терминала
End-Of-Record	25/0x19	885	Запрос на завершение передачи данных кодом “EOR”
Linemode	34/0x22	1116	Установка локального редактирования строк и по-строчной передачи на удаленном терминале

Протокол TELNET позволяет, находясь территориально далеко от удалённого сервера, управлять им. Но при передаче информации не используется шифрование, из-за чего любые логины и пароли передаются в явном виде, что позволяет злоумышленникам перехватить их и использовать в своих целях.

Использовать протокол TELNET в общедоступных целях небезопасно, так как может быть нарушена конфиденциальность передаваемой информации. В связи с чем был разработан протокол SSH (Secure Shell). Новый протокол обеспечивал всю функциональность TELNET, но при этом исправлял уже обнаруженные уязвимости и шифровал передаваемый трафик.

2.15. Протоколы передачи файлов FTP и TFTP

2.15.1. Протокол FTP

FTP (File Transfer Protocol – протокол передачи файлов) – протокол, использующийся для гарантированной передачи файлов за счёт подтверждения приёма-передачи. Для передачи файлов используется протокол TCP.

Изначально протокол FTP описан в спецификации RFC-959, а в дальнейшем дополнен в спецификациях RFC-2228 (Расширение безопасности FTP), RFC-2640 (Интернационализация FTP), RFC-3659 (Расширения для FTP), RFC-5797 (Реестр команд и расширений FTP), RFC-7151 (Команда HOST FTP для виртуальных хостов).

Протокол предоставляет:

- командный интерфейс для программного доступа к удалённым файлам;
- интерактивную оболочку, имеющую набор команд, позволяющих выполнять большой набор функций;
- преобразование данных и форматов хранимых данных;
- аутентификацию по имени пользователя и паролю.

Каждое FTP-соединение состоит из двух TCP-соединений:

- управляющего (control connection) – поддерживается от начала и до конца сеанса связи;
- доставки данных (data transfer connection) – устанавливается только на время процесса передачи данных.

Оба соединения поддерживаются двумя различными процессами в операционной системе. Управляющее соединение поддерживается на основе протокола TELNET с использованием NVT-интерфейса.

FTP-сервер работает на общепринятых портах: порт 21 для управляющего соединения и порт 20 для доставки данных. FTP-клиент может использовать любой заранее обговорённые порты для управления и доставки данных. Это различие позволяет поддерживать несколько FTP-соединений с сервером для клиента.

На рисунке 2.15.1 изображено взаимодействие между клиентом и сервером, а также указаны используемые порты.

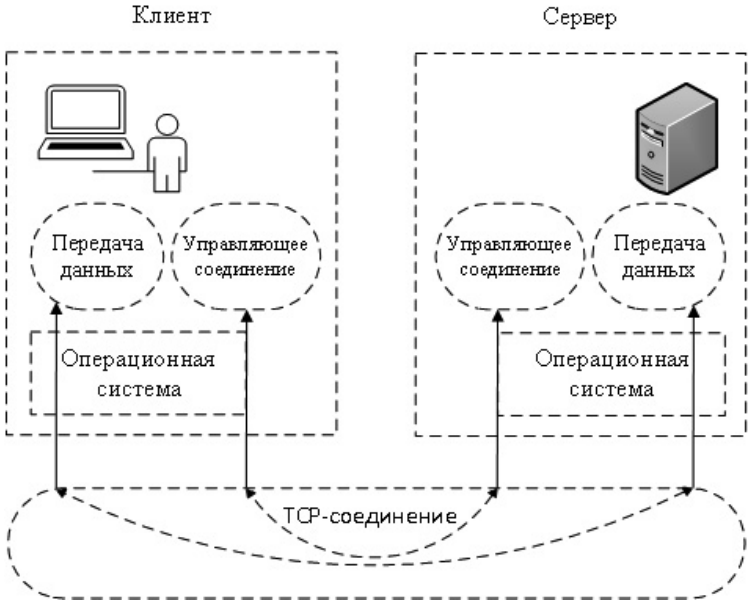

Рисунок 2.15.1 – Модель взаимодействия между клиентом и сервером

Процедурная составляющая FTP-протокола очень проста. Абонент посылает серверу текстовые команды, состоящие из имени команды и необходимых параметров. На полученные команды сервер отвечает текстовыми ответами, которые начинаются с кода состояния – трёх цифр, обозначающих

результат выполнения команды абонента и текста, раскрывающего значение кода.

Первая цифра означает исход выполнения команды. Вторая цифра определяет тип ошибки, если исход команды – ошибка выполнения. Третья цифра специфицирует ошибку.

В таблице 2.15.1 расшифрованы значения первой цифры кода состояния с пояснениями, а в таблице 2.15.2 – второй цифры кода состояния.

Таблица 2.15.1 – Значения первой цифры кода состояния

Цифра	Пояснение
1	Команда принята к выполнению, но ещё не завершена
2	Выполнение команды успешно завершено
3	Команда принята и ожидается какая-либо дополнительная команда
4	В данные момент команда не может быть выполнена
5	Невозможность выполнения команды

Таблица 2.15.2 – Значения второй цифры кода состояния

Цифра	Пояснение
0	Синтаксическая ошибка
1	Информационное сообщение
2	Сообщение относится к управляющему соединению/соединению доставки данных
3	Сообщения об аутентификации пользователя и его правах
4	Не используется
5	Состояние файловой системы

Список некоторых команд, использующихся в протоколе FTP, а также их назначение сведён в таблицу 2.15.3.

При интерактивном использовании FTP-протокола для пользователя это выглядит как работа в командной оболочке. При вызове FTP-протокола в командной строке появляется приглашение `FTP>`, после чего могут вводиться различные команды. Список некоторых команд, которые могут быть использованы в командной оболочке, а также пояснение к ним, приведён в таблице 2.15.4.

Таблица 2.15.3 – Список команд

Команда	Параметры	Назначение
ABOR		Прервать передачу файла
LIST		Возвращает список файлов каталога
CWD	path	Смена каталога

PASS	password	Передача пароля
PORT	n1, n2, n3, n4, n5, n6	Передаёт серверу IP-адрес клиента (первые четыре байта) и порт (последние два байта)
QUIT		Отключение от сервера
RETR	filename	Скачивание файла
STOR	filename	Закачка файла
SYST		Возвращает тип системы
TYPE	type	Установка типа передачи файла (бинарный/текстовый)
USER	username	Передача имени пользователя

Таблица 2.15.4 – Список команд в командной оболочке

Название команды	Назначение
dir, ls	Показывает содержимое каталога
mdir, mls	Показывает содержимое нескольких каталогов
cd	Осуществляет переход в другой (вложенный) каталог
cdup	Осуществляет переход в вышележащий каталог
lcd	Изменяет рабочий каталог на локальной машине
mkdir	Создает каталог на сервере
rmdir	Стирает каталог на сервере
delete	Удаляет файл на сервере
mdelete	Стирает несколько файлов на сервере
get	Копирует файл с сервера к клиенту
mget	Копирует несколько файлов с сервера к клиенту
put	Копирует файл клиента на сервер
mput	Копирует несколько файлов клиента на сервер
rename	Переименовывает файл на сервере
type	Устанавливает тип передаваемых данных
ascii	Устанавливает текстовый тип передаваемых данных
binary	Устанавливает двоичный тип передаваемых данных
close, disconnect	Завершает FTP-сеанс
bye	Завершает FTP-сеанс и выходит из оболочки

Некоторые FTP-серверы предоставляют возможность анонимного доступа к своим данным. Для этого при подключении указывается пользователь `anonymous`, а вместо пароля обычно вводится адрес электронной почты. Как правило такой доступ предоставляется для серверов с бесплатной общедоступной информацией.

2.15.2. Протокол TFTP

TFTP (Trivial File Transfer Protocol – простой протокол передачи файлов) – протокол, использующийся для передачи файлов по протоколу UDP без гарантии доставки. Протокол TFTP описан в спецификации RFC-1350.

Общение между клиентом и сервером происходит посредством передачи TFTP-пакетов. Существует шесть типов пакетов:

- RRQ (Read Request, 0x01) – запрос на чтение файла;
- WRQ (Write Request, 0x02) – запрос на запись файла;
- DATA (Data, 0x03) – передаваемые данные;
- ACK (Acknowledgment, 0x04) – подтверждение пакета;
- ERR (Error, 0x05) – ошибка;
- OACK (Option Acknowledgment, 0x06) – подтверждение опций.

WRQ и RRQ пакеты посылаются клиентом для начала передачи данных. Оба пакета имеют одинаковый формат, приведённый в таблице 2.15.5.

Таблица 2.15.5 – Формат WRQ и RRQ пакетов

Тип пакета	Имя файла	Конец строки	Режим передачи	Конец строки	Опции (не обязательно)
0x01/ 0x02	ASCII-строка	0x00	ASCII-строка	0x00	Формат опций в таблице 15.8

В протоколе TFTP присутствует два режима передачи: `netascii` – перед передачей файл перекодируется в формат ASCII и `octet` – файл передаётся без изменений.

Как только сервер получает пакет типа RRQ, незамедлительно начинается передача данных, при этом идентификатор будет равен единице. При получении пакета типа WRQ, сервер посылает пакет типа ACK с идентификатором 0.

Формат пакета подтверждения приведён в таблице 2.15.6.

Таблица 2.15.6 – Формат ACK-пакета

Тип пакета	Номер принятого пакета
0x04	2 байта

Пакет передачи данных имеет схожую структуру с пакетом подтверждения. В нём, вместо номера принятого пакета сообщается номер передаваемого пакета (2 байта) и данные (до 512 байтов).

Формат пакета передачи данных приведён в таблице 2.15.7.

Таблица 2.15.7 – Формат DATA-пакета

Тип пакета	Номер передаваемого пакета	Данные
0x03	2 байта	До 512 байт

В спецификации RFC-2347 был предусмотрен формат опций. Они указываются в конце RRQ и WRQ пакетов, если это необходимо. При этом добавлен новый тип пакетов – OACK, который используется для подтверждения сервером списка опций, полученным в пакете с перечисленными опциями.

Существует возможность указывать сразу несколько опций в одной передаваемом пакете, для этого они должны следовать друг за другом, а их порядок не важен.

Формат пакета опций приведён в таблице 2.15.8. В таблице 2.15.9 описан формат пакета OACK.

Таблица 2.15.8 – Формат опций

Опция	Конец строки	Значение	Конец строки
ASCII-строка	0x00	ASCII-строка	0x00

Таблица 2.15.9 – Формат OACK-пакета

Тип пакета	Опция	Конец строки	Значение	Конец строки
0x06	ASCII-строка	0x00	ASCII-строка	0x00

В одном OACK-пакете может передавать сразу несколько подтверждений полученных опций, тогда формат сообщения увеличивается на то количество опций, которое необходимо указать.

При возникновении ошибки сервер отправляет сообщения типа ERR. Формат сообщения представлен в таблице 2.15.10.

Таблица 2.15.10 – формат ERR-пакета

Тип пакета	Код ошибки	Описание ошибки	Конец строки
0x05	Указан в таблице 15.11	ASCII-строка	0x00

Изначально существовало 8 типов ошибок, но с вводом опций, появился девятый тип. Список ошибок и их описание приведено в таблице 2.15.11.

Суть TFTP-протокола заключается не только в передачи и приёме файлов, с помощью специальных пакетов, но и в восстановлении файла, переданного с помощью UDP-протокола.

При передаче RRQ и WRQ пакетов клиент и сервер устанавливают таймаут (timeout), после которого, если пакет не был принят, то его отправляют

вновь. После получения принимающая сторона отправляет код подтверждения принятого пакета.

Размер передаваемого пакета, стандартно равен 16 мегабайтам, но если клиент и сервер поддерживают расширения протокола, описанные в спецификациях RFC-2347 (Расширение опции TFTP) и RFC-2348 (Опция размера блока TFTP), то размер блока увеличивается до 4 гигабайт.

Таблица 2.15.11 – Коды ошибок и их описание

Код ошибки	Описание
0	Не определено, см. описание ошибки, если оно присутствует
1	Файл не найден
2	Доступ запрещён
3	Невозможно выделить место на диске
4	Некорректная TFTP-операция
5	Неправильный Transfer ID
6	Файл уже существует
7	Пользователь не существует
8	Неправильная опция

2.16. Протоколы электронной почты: SMTP, POP, IMAP

2.16.1. Протокол SMTP

Для передачи электронных почтовых сообщений по сети Интернет был специально разработан протокол почтовых систем SMTP (Simple Mail Transfer Protocol – простой протокол передачи почты). Основная спецификация, в которой описан протокол SMTP – RFC-821. В спецификации RFC-822 описан формат почтовых сообщений. Для передачи сообщений используется протокол TCP.

SMTP, как и некоторые другие протоколы, реализуется в виде двух частей – SMTP-клиента и SMTP-сервера. Клиент работает на стороне отправителя, а сервер на стороне получателя. При этом сервер должен постоянно находиться в режиме подключения к клиенту, ожидая поступление новых запросов.

Пример передачи сообщений с помощью протокола SMTP изображён на рисунке 2.16.1.

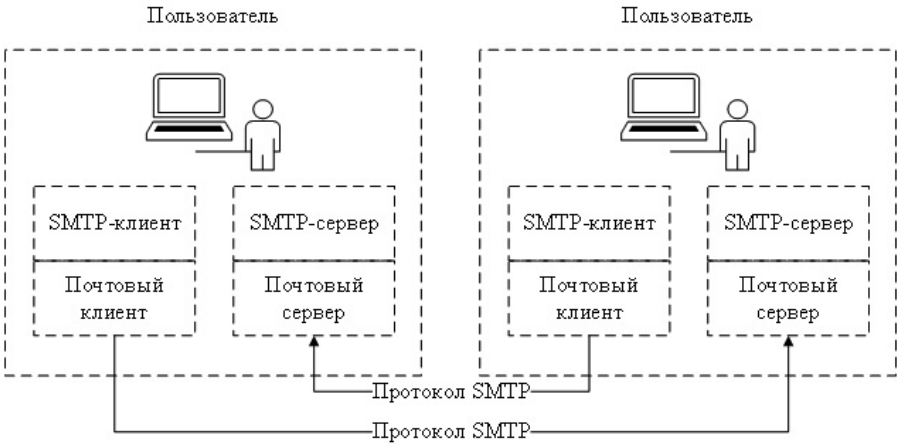

Рисунок 2.16.1 – Передача почтовых сообщений с помощью протокола SMTP

Логика протокола SMTP заключается в следующем:

- Клиент, с помощью графического интерфейса, отправляет запрос на установление TCP-соединения на 25 порт SMTP-сервера;
- В случае, если сервер готов обслуживать клиента, он отправляет свои идентификационные данные и DNS-имя. Если сервер не готов, то он уведомляет клиента соответствующим сообщением, после чего клиент вновь пытается установить соединение, отправляя новый запрос;
- Установив соединение, клиент передаёт серверу почтовые адреса отправителя и получателя. Если адреса корректны, то сервер даёт согласие на установления SMTP-соединения;
- Происходит передача сообщения;
- Если при получении тела сообщения не произошло ошибок, то сервер отправляет клиенту команду OK, которая обозначает, что сервер принял на себя ответственность по передаче сообщения получателю. Но это не гарантирует успешную доставку сообщения, так как она зависит не только от этого сервера;
- Если сервер не может доставить сообщение, то клиенту передаётся отчёт об ошибке и разрывается соединение;
- После передачи сообщения оба соединения (TCP и SMTP) разрываются, а сообщение сохраняется в буфере на сервере.

В рамках одного TCP-соединения клиент может передать несколько почтовых сообщений, каждый раз сообщая адреса отправителя и получателя.

В протоколе SMTP предусмотрены отрицательные и положительные уведомления о доставке почты. Однако положительные уведомления по

умолчанию не являются обязательными, из-за чего, они не передаются отправителю. Отрицательные же уведомления обязательны к отправке.

На рисунке 2.16.1 изображён пример взаимодействия двух пользователей, у которых на компьютерах установлены почтовые клиент и сервер. В настоящее время это взаимодействие используется крайне редко, а почта передаётся через выделенный почтовый сервер.

Причина, по которой схема непосредственного взаимодействия двух пользователей не нашла распространённого применения, заключается в том, что клиент и сервер должны быть в режиме постоянного подключения. Если подключения нет, то письма не доставляются получателю.

Для решения этой проблемы SMTP-сервер размещают удалённо от пользователей, оставляя в их распоряжении только SMTP-клиент.

На рисунке 2.16.2 изображён пример взаимодействия двух пользователей с выделенным сервером.

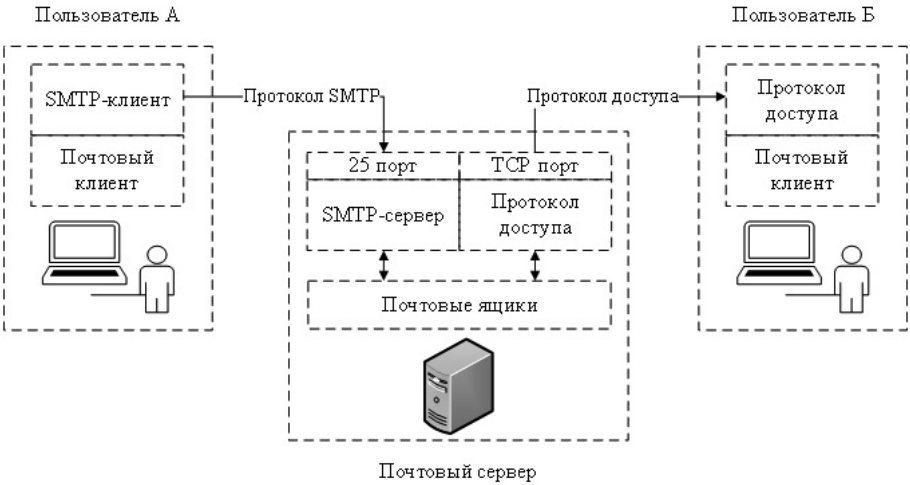

Рисунок 2.16.2 – Взаимодействие двух пользователей с выделенным сервером

В случае, когда используется отдельный SMTP-сервер, почтовые письма передаются на него точно так же, как и при непосредственном взаимодействии двух пользователей. После того, как письмо было загружено на сервер, оно попадает в почтовый буфер получателя.

Как только получатель письма запускает свою почтовую программу и инициирует проверку почты, его почтовый клиент запускает протокол доступа к почтовому серверу. Так как протокол SMTP используется только для загрузки

писем на сервер, то для получения письма с сервера должен использоваться какой-либо другой протокол.

2.16.2. Протоколы POP3 и IMAP

Специально для получения писем с выделенных серверов были разработаны два протокола – POP3 (Post Office Protocol v.3 – протокол почтового отделения версии 3) (RFC-1801) и IMAP (Internet Mail Access Protocol – протокол доступа к электронной почте Интернета) (RFC-3501).

Оба этих протокола ориентированы на приём данных по инициативе клиента. Почтовый клиент отправляет запрос на установление TCP-соединения с сервером на порт 110, если протокол POP3, или на 143, если протокол IMAP.

Протоколы поддерживают передачу отправителю уведомлений о том, что письмо доставлено, а также факт открытия сообщения, если отправитель запрашивает такую услугу. Обычно почтовый клиент не отправляет уведомления автоматически, а запрашивает у получателя подтверждение на такое действие.

Оба протокола поддерживают аутентификацию пользователей на основе их идентификаторов и паролей. Одна POP3 и IMAP имеют принципиальные различия в доступе к почтовому серверу.

Протокол POP3 скачивает все адресованные сообщения в память компьютера, на котором включен почтовый клиент. При этом с сервера удаляются все упоминания от считанной почты. Если клиент прочитал письмо на одном компьютере, то на другом эти письма не будут отображаться, так как они удалены с сервера.

При использовании протокола IMAP в память компьютера сохраняются копии писем и на сервере всегда хранится оригинал. Пользователь может отправить запрос на удаление письма с сервера с помощью почтового клиента.

Ещё одним отличием протоколов является то, что IMAP предоставляет возможность предварительного чтения заголовка письма до его полного скачивания с сервера.

2.16.3. Формат почтового сообщения

Согласно спецификации RFC-822 формат почтового сообщения состоит из трёх частей:

- Конверт (envelope) – используется программами доставки почтовых сообщений;
- Заголовок (header) – содержит служебную информацию для управления доставки и обработки сообщения;

- Тело сообщения (body) – непосредственно текст сообщения с возможными прикреплёнными данными.

В таблице 2.16.1 приведен состав заголовка почтового сообщения с пояснениями.

Таблица 2.16.1 – Поля, содержащиеся в заголовке почтового сообщения

Название поля	Описание
Date (дата)	Дата отправки сообщения
From (от)	Адрес, который отправитель указал в качестве исходящего
Sender (отправитель)	Реальный адрес отправителя
Subject (тема)	Тема сообщения
To (кому)	Адрес получателя
Cc (копия)	Адреса дополнительных получателей (может быть пустым)
In-Reply-To (ответить)	Адрес, по которому необходимо направлять ответ
Comment (комментарий)	Дополнительная информация к письму
X-Special-Action (доп. Информация)	Дополнительное поле пользователя. Его назначение не определено в спецификации
Message-ID (идентификатор сообщения)	Уникальный идентификатор сообщения

Изначально предполагалось, что тело сообщения может содержать только текст в формате ASCII, а передача нетекстовой информации не требовалась, из-за чего протоколы передачи почтовых сообщений, при попытке передачи нетекстовой информации, некорректно её обрабатывали. Для решения этой проблемы был разработал специальный протокол MIME (Multipurpose Internet Mail Extension – многоцелевое расширение почты Интернет) (RFC-2045), который преобразовывает нетекстовые данные к текстовому виду.

Ниже представлен пример заголовка и тело письма. Поля, начинающиеся с X, как было написано ранее, не определены в спецификации и используются по усмотрению почтового клиента. Поля Content-Transfer-Encoding и Content-Type относятся к формату тела письма. Поле Return-Path обозначает адрес, по которому должны быть доставлены уведомления об отказе.

```
From: Test Test <test.test@yandex.ru>
To: test-test@yandex.ru
Subject: Test
Date: Sun, 12 May 2019 17:48:34 +0300
Message-Id: <15632481557672514@sas2-22600713dea1.qloud
c.yandex.net>
Content-Transfer-Encoding: 8bit
Content-Type: text/html; charset=utf-8
Return-Path: test.test@yandex.ru
MIME-Version: 1.0
X-YandexSms-Digest: 1712972582882a797f5abc6747a315d3
X-Mailer: Yamail [ http://yandex.ru ] 5.0
<div>\xd0\xa2\xd0\xb5\xd1\x81\xd1\x82</div>
```

2.17. Гипертекстовый протокол HTTP

HTTP (HyperText Transfer Protocol – протокол доставки гипертекстовых сообщений) – это протокол прикладного уровня, с его помощью осуществляется обмен гипертекстовыми сообщениями в сети Интернет. HTTP-протокол реализует различные формы доступа, базирующиеся на URI-идентификации (Universal Resource Identifier – унифицированный идентификатор ресурса) в форме URL-ссылки/адреса (Uniform Resource Locator – унифицированный указатель ресурса) и универсальном способе именования информационных ресурсов URN (Universal Resource Name – унифицированное имя ресурса). В настоящее время используются три версии протокола: HTTP 1.0 (RFC-1945), HTTP 1.1 (RFC-2616), HTTP 2.0 (RFC-7540).

Во всех версиях протокола обмен сообщениями происходит по обычной схеме “запрос-ответ”. При этом все сообщения являются текстовыми, состоящими из обязательных и необязательных полей в кодировке ASCII.

Для передачи HTTP-сообщений используется протокол TCP. Все сообщения по умолчанию передаются по 80 порту. При этом используется два типа соединений:

- Кратковременное соединение – в течение одного TCP-соединения передаётся только один объект;

- Долговременное соединение – в течение одного TCP-соединения передаётся несколько объектов, а время существования соединения зависит от конфигурации веб-службы.

Долговременное соединение может использоваться двумя способами:

- последовательная передача запросов с простоями – после отправки запроса ожидается ответ и только после получения ответа отправляется новый запрос;

- конвейерная передача – последующие запросы посылаются до прибытия предыдущих ответов. Стандартно, параллельно могут передаваться от 5 до 10 запросов.

Использование кратковременных соединений приводит к увеличению передаваемого трафика, так как в протоколе TCP каждому соединению предшествует трёхэтапное установление этого соединения, что замедляет работу браузера. В HTTP версии 1.0 используются соединения только такого типа.

Для HTTP 1.1 стандартно применяются постоянные соединения в конвейерном режиме передачи. При этом соединение разрывается браузером или сервером с помощью отправки специального токена разрыва соединения в составе HTTP-пакета. Если пользователь неактивен, то соединение разрывается по тайм-ауту, чтобы не тратить ресурсы памяти.

Как отмечалось ранее, запросы и ответы протокола HTTP состоят из текстовых строк и имеют единую обобщенную структуру, которая состоит из трёх частей:

- обязательная стартовая строка;
- необязательный заголовок;
- необязательное тело сообщения.

Стартовая строка запроса имеет вид:

```
Метод URI HTTP/Версия
```

Метод (method) – последовательность из любых символов, кроме управляющих и разделителей, которая определяет операцию, необходимую осуществить ресурсом с указанным URI. Спецификация HTTP 1.1 и HTTP 2.0 не ограничивает количество различных методов, но для соответствия общим стандартам и для сохранения совместимости, как правило, используются лишь стандартные методы. Список стандартных методов, а также пояснения к ним приведены в таблице 2.17.1

Таблица 2.17.1 – Список стандартных методов с пояснениями

Метод	Пояснение к методу
GET	Используется для запроса содержимого указанного ресурса
HEAD	Используется так же, как и метод GET, но в ответе на него отсутствует тело сообщения

POST	Применяется для передачи пользовательских данных указанному ресурсу
PUT	Используется для загрузки содержимого запроса на указанный в запросе URI
PATCH	Аналогичен методу PUT, но применяется только к определённому фрагменту ресурса
DELETE	Применяется для удаления указанного объекта на сервере
OPTIONS	Используется для определения возможностей веб-сервера или для определения параметров соединения конкретного ресурса (отсутствует в версии 1.0)
TRACE	Возвращает полученный запрос с информацией о том, какую информацию добавили или изменили промежуточные серверы
CONNECT	Преобразует соединения запроса в прозрачный TCP/IP-тоннель

URI – путь до конкретного ресурса, над которым необходимо выполнить операцию, указанную в методе. Если запрос относится не к конкретному ресурсу, а к веб-серверу, то вместо URI указывается звёздочка (символ “*”). В таком случае запрос будет выглядеть следующим образом:

```
OPTIONS * HTTP 1.1
```

Версия (version) – пара разделённых точкой цифр, соответствующих версии протокола: 1.0/1.1/2.0.

Стартовая строка ответа имеет вид:

```
HTTP/Версия Код Пояснение
```

Версия (version) – аналогично версии в запросе.

Код состояния (Status Code) – код, состоящий из трёх цифр, определяющий результат совершения запроса. Первая цифра из трёх указывает на класс состояния. Всего существует пять классов состояния, которым соответствуют цифры от 1 до 5. В таблице 2.17.2 указаны коды состояний, их классы и назначение.

Таблица 2.17.2 – Коды и классы состояний с указанием их назначений

Код	Класс	Назначение
1xx	Информационный (Informational)	Информация о процессе передачи
2xx	Успех (Success)	Информация об успешном принятии и обработке запроса клиента

3xx	Перенаправление (Redirection)	Сообщает клиенту, что для успешного выполнения операции необходимо произвести запрос по другому URL-адресу
4xx	Ошибка клиента (Client Error)	Указание на ошибки со стороны клиента с пояснением в теле сообщения
5xx	Ошибка сервера (Server Error)	Информирование о неудачном выполнении операции из-за ошибки на сервере

Пояснение (Reason Phrase) – пояснение к коду ответа, которое предназначено для упрощения чтения ответа человеком.

Как говорилось ранее, после стартовой строки следуют заголовки, а также, если это необходимо, тело ответа.

Ниже приведён пример ответа сервера на запрос `GET http://mtuci.ru HTTP/1.1` (текст ответа опущен):

```
HTTP/1.1 200 OK
Date: Fri, 10 May 2019 14:12:25 GMT
Server: Apache
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Last-Modified: Fri, 10 May 2019 14:12:25 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-
check=0, pre-check=0
Pragma: no-cache
Set-Cookie: PHPSESSID=h773euhq8l2nf6vud1ra06p7f1; path=/
Transfer-Encoding: chunked
Content-Type: text/html
```

Тело ответа

Тело ответа следует через два переноса после последнего заголовка. Заголовки могут отличаться для различных браузеров.

Как видно из примера выше, запросы и ответы в протоколе HTTP передаются в незащищенном виде, что позволяет злоумышленникам перехватывать пакеты и производить атаку по середине, когда HTTP-пакеты доходят до злоумышленника и он модифицирует их. Для защиты конфиденциальности и целостности информации, передаваемой по протоколу HTTP, был разработан протокол HTTPS (RFC-2660) (HyperText Transfer Protocol Secure – безопасный протокол передачи гипертекста). Говорить

“протокол HTTPS” не корректно, так как под ним подразумевается работа двух протоколов – HTTP и SSL (RFC-6101).

При использовании протокола HTTPS сообщения запросов и ответов такие же, как и в протоколе HTTP, но при этом они шифруются на время передачи между клиентом и сервером, что позволяет обеспечить целостность и конфиденциальность передаваемой информации. Для передачи информации по HTTPS по умолчанию используется 443 порт.

2.18. Протокол безопасного доступа SSH

SSH (Secure Shell – безопасная оболочка) – сетевой протокол, предназначенный для безопасного удалённого управления операционной системой и туннелирования TCP-соединений.

В январе 2006 года рабочая группа IETF утвердила протокол в качестве интернет-стандарта. Подробно SSH описан в спецификациях RFC 4250-4256.

SSH был разработан для замены протоколов, которые так же позволяли удалённо управлять операционной системой, но при этом не шифровали передаваемый трафик, из-за чего целостность и конфиденциальность данных могла быть нарушена.

С помощью шифрования создаётся безопасное соединение для передачи информации. Канал связи при этом может быть не безопасным. Для передачи данных используется протокол TCP и, по умолчанию, используется 22 порт.

В обоих версиях поддерживаются различные алгоритмы аутентификации данных, такие как: RSA, DSA. Сеансовый ключ генерируется на основе алгоритма Диффи-Хеллмана. Передаваемые данные шифруются, как правило, симметричными алгоритмами.

Программная реализация SSH первой и второй версий делится на серверную и клиентскую. Соединение между сервером и клиентом, использующими разные версии протокола, невозможно из-за особенностей технической реализации.

2.18.1. Протокол SSH-1

Протокол первой версии был разработан в 1995 году для обеспечения к атакам прослушивания трафика. Для работы необходим SSH-сервер, который находится в состоянии прослушивания соединений от клиентских машин. Соединение инициализирует клиент.

Процесс установления соединения между сервером и клиентом изображён на рисунке 2.18.1 и состоит из следующих ключевых пунктов:

- На активный сервер поступает запрос от клиента на установление SSH-соединения и создание нового сеанса связи;

- Если сервер готов к открытию нового сеанса связи, отправляется запрос клиенту о том, какая версия протокола используется;
- Если версии совпали и обе стороны подтвердили готовность продолжить создание соединения, то сервер посылает клиенту постоянный публичный и временный ключи. Происходит передача сессионного ключа в соответствии с выбранным алгоритмом шифрования (RSA, DSA и др.);
- Зашифрованное соединение создано и готово к использованию.

Соединение будет существовать до момента разрыва клиентом или при невозможности подтверждения аутентификации клиентом и сервером.

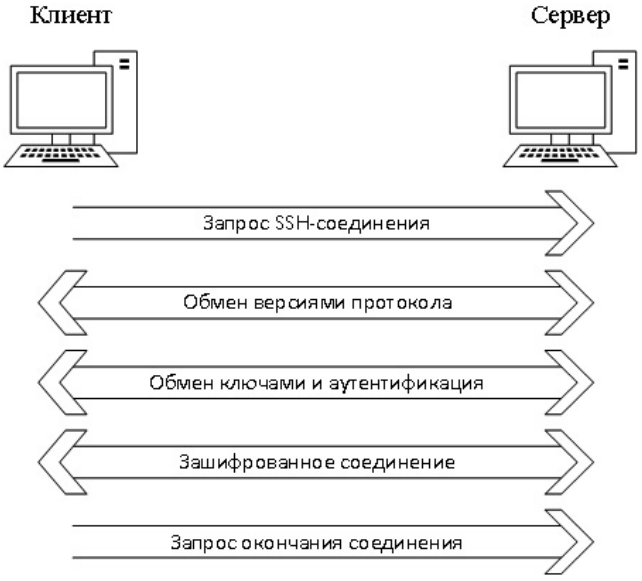

Рисунок 2.18.1 – Процесс установления соединения согласно протоколу SSH-1

2.18.2. Протокол SSH-2

Новая версия протокола выполняет те же функции, что и его предыдущая, но с рядом значительных отличий в реализации. Протокол SSH-2 состоит из трёх самодостаточных протоколов:

- Протокол транспортного уровня;
- Протокол соединения;
- Протокол аутентификации.

С помощью протокола транспортного уровня предоставляется возможность шифрования, сжатия и обеспечения целостности передаваемой информации. Этого протокола достаточно для установления защищенного SSH-

соединения без аутентификации клиентов. Такой вариант необходим, если сервер, например, предоставляет анонимный FTP доступ к информации.

Протокол соединения используется для установления многопоточных соединений по SSH-туннелю между клиентами и сервером, что позволяется снижать нагрузку.

Протокол аутентификации используется на сервере для проверки полномочий подключающихся клиентов.

2.18.3. Архитектура протокола SSH

В пакетах протокола SSH используются поля различных типов данных. Список типов данных, которые могут использоваться в пакетах приведён в таблице 2.18.1.

Каждый пакет имеет следующий формат:

```
uint32    packet_length
byte      padding_length
byte      payload
byte      random padding
byte      mac
```

packet_length – длина пакета в байтах.

padding_length – длина случайного заполнения.

payload – полезная нагрузка. Длина поля полезной нагрузки, вычисляется в соответствии с формулой: `packet_length - padding_length - 1`.

random padding – случайное заполнение, используемое для усложнения анализа трафика. Дополняет длину пакета до числа, кратного восьми. Заполняется случайными байтами. Минимальная длина – 4 байта, максимальная – 255 байт.

mac (Message Authentication Code) – код аутентификации сообщения. Если аутентификации нет, то поле должно содержать “none”. Обычно 4 байта.

Максимальная длина пакета составляет 35000 байт, из которых 32768 или меньше байт полезной нагрузки. Если необходимо передать большее количество информации, то, при подтверждении от клиента и сервера, эту величину можно повысить.

При передаче полезной нагрузки, существует возможность её сжатия, для уменьшения количества передаваемых данных. На текущий момент определены два вида сжатия:

- none – передача без сжатия;
- zlib – сжатие согласно алгоритму zlib.

Таблица 2.18.1 – Типы данных протокола SSH

byte	8 произвольных битов, может подразумевать под собой массив из n байтов
boolean	Логические значения. 0 – ложь, 1 – истина
uint32	32-битовое число без знака
uint64	64-битовое число без знака
string	Двоичная строка произвольной длины. Строки хранятся в переменной типа uint32. Если в строке находится текст, то для отображения пользователю используется кодировка UTF-8
mpint	Целые числа в формате дополнения до 2. Если число имеет отрицательное значение, то старший бит устанавливается в 1
name-list	Строка, состоящая из списка разделённых запятыми имён. Значения хранятся в виде uint32. В начале идёт длина списка, далее сам список, который может быть пустым

Если использовано сжатие, то шифрование происходит после выполнения процесса сжатия данных. Рекомендуется, чтобы метод сжатия был одинаков в обоих направлениях передачи данных SSH-сессии.

Пакеты SSH используют номера сообщений, которые обозначают, к какой компоненте они относятся. Номера сообщений и пояснения к ним сведены в таблице 2.18.2.

Таблица 2.18.2 – Номера сообщений SSH-пакетов

1-19	Базовые сообщения транспортного уровня. Например, 1-disconnect, 2-ignore, 4-debug.
20-29	Согласование алгоритма
30-49	Сообщения, связанные с обменом ключами
50-59	Базовые сообщения протокола аутентификации
60-79	Сообщения, связанные с методом аутентификации
80-89	Базовые сообщения протокола
90-127	Сообщения, связанные с каналом передачи данных
128-191	Используются для клиентских протоколов
192-255	Используются для локальных расширений

Для проверки целостности данных используется поле mac, состоящее из 32 битов. Значение генерируется псевдослучайным образом согласно алгоритму. После отправки 2^32 пакетов данных возможна утечка информации. Для предотвращения такой возможности, согласно спецификации RFC-4253, предлагается менять ключ шифрования после передачи каждого гигабайта данных.

Список литературы

Основная литература

1. Олифер В.Г. Компьютерные сети. Принципы, технологии, протоколы, 4-е издание. - М.: Питер, 2014. - 943с.
2. Кремер А.С., Мальянов С.А., Малюк А.А «Обеспечение доверия и безопасности при использовании ИКТ» Учебное пособие. - М.: ОГО АДЭ, 2017.

Дополнительная литература

3. Семенов Ю.А. Протоколы и алгоритмы маршрутизации в Интернет [Электронный ресурс]/ Семенов Ю.А.— Электрон. текстовые данные.— М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.— 998 с.— Режим доступа: http://www.iprbookshop.ru/62826.html.— ЭБС «IPRbooks
4. Берлин А.Н. Основные протоколы Интернет [Электронный ресурс]/ Берлин А.Н.— Электрон. текстовые данные.— М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.— 602 с.— Режим доступа: http://www.iprbookshop.ru/52181.— ЭБС «IPRbooks»
5. Information Sciences Institute University of Southern California. Internet Protocol - DARPA Internet Program Protocol Specification. RFC 791, 1981.– 45p. https://tools.ietf.org/html/rfc791
6. J. Postel. User Datagram Protocol. RFC768, 1980.– 3p. https://tools.ietf.org/html/rfc768
7. Information Sciences Institute University of Southern California. TRANSMISSION CONTROL PROTOCOL – DARPA Internet Program Protocol Specification. RFC793, 1981.– 85p. https://tools.ietf.org/html/rfc793
8. J. Postel. INTERNET CONTROL MESSAGE PROTOCOL – DARPA Internet Program Protocol Specification. RFC792, 1981.– 21p. https://tools.ietf.org/html/rfc792
9. Internet Engineering Task Force (IETF). Internet Protocol, Version 6 (IPv6) Specification. RFC8200, 2017.– 41p. https://tools.ietf.org/html/rfc8200
10. P. Mockapetris. DOMAIN NAMES - CONCEPTS AND FACILITIES. RFC1034, 1987. – 55 p. https://tools.ietf.org/html/rfc1034
11. P. Mockapetris. DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION. RFC1035, 1987. – 55p. https://tools.ietf.org/html/rfc1035
12. S. Thomson, et al.. DNS Extensions to Support IP Version 6. RFC3596, 2003. – 8p. https://tools.ietf.org/html/rfc3596

13. S. Rose & W. Wijngaards. DNAME Redirection in the DNS. RFC6672, 2012. – 22p. https://tools.ietf.org/html/rfc6672
14. Y. Bernet & R. Pabbati. Application and Sub Application Identity Policy Element for Use with RSVP. RFC2872, 2000. – 6p. https://tools.ietf.org/html/rfc2872
15. C. Partridge, T. Mendez & W. Milliken. Host Anycasting Service. RFC1546, 1993.– 9p. https://tools.ietf.org/html/rfc1546
16. T. Hardie. Distributing Authoritative Name Servers via Shared Unicast Addresses. RFC3258, 2002.– 11p. https://tools.ietf.org/html/rfc3258
17. K. Sollins. THE TFTP PROTOCOL (REVISION 2). RFC1350, 1992.– 11p. https://tools.ietf.org/html/rfc1350
18. G. Malkin & A. Harkin. TFTP Option Extension. RFC2347, 1998.– 7p. https://tools.ietf.org/html/rfc2347
19. G. Malkin & A. Harkin. TFTP Blocksize Option. RFC2348, 1998.– 5p. https://tools.ietf.org/html/rfc2348
20. David H. Crocker. STANDARD FOR THE FORMAT OF ARPA INTERNET TEXT MESSAGES. RFC822, 1982.– 47p. https://tools.ietf.org/html/rfc822
21. N. Freed & N. Borenstein. Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies. RFC2045, 1996.– 31p. https://tools.ietf.org/html/rfc2045
22. J. Klensin. Simple Mail Transfer Protocol. RFC2821, 2001.– 79p. https://tools.ietf.org/html/rfc2821
23. R. Fielding & J. Reschke. Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content. RFC7231, 2014.– 101p. https://tools.ietf.org/html/rfc7231
24. L. Dusseault & J. Snell. PATCH Method for HTTP. RFC5789, 2010.– 10p. https://tools.ietf.org/html/rfc5789
25. T. Ylonen & C. Lonvick. The Secure Shell (SSH) Protocol Architecture. RFC4251, 2006.– 30p. https://tools.ietf.org/html/rfc4251
26. K. Moriarty, et al. PKCS #1: RSA Cryptography Specifications Version 2.2. RFC8017, 2016.– 78p. https://tools.ietf.org/html/rfc8017
27. T. Pornin. Deterministic Usage of the Digital Signature Algorithm (DSA) and Elliptic Curve Digital Signature Algorithm (ECDSA). RFC6979, 2013.– 79p. https://tools.ietf.org/html/rfc6979
28. E. Rescorla. Diffie-Hellman Key Agreement Method. RFC2631, 1999.– 13p. https://tools.ietf.org/html/rfc2631
29. T. Ylonen & C. Lonvick. The Secure Shell (SSH) Transport Layer Protocol. RFC4253, 2006.– 32p. https://tools.ietf.org/html/rfc4253
30. T. Ylonen & C. Lonvick. The Secure Shell (SSH) Protocol Architecture. RFC4151, 2006.– 30p. https://tools.ietf.org/html/rfc4251

31. P. Deutsch & J-L. Gailly. ZLIB Compressed Data Format Specification version 3.3. RFC1950, 1996.– 11p. https://tools.ietf.org/html/rfc1950
32. P. Deutsch. DEFLATE Compressed Data Format Specification version 1.3. RFC1951, 1996.– 17p. https://tools.ietf.org/html/rfc1951
S. Lehtinen & C. Lonvick. The Secure Shell (SSH) Protocol Assigned Numbers. RFC4250, 2006.– 20p. https://tools.ietf.org/html

Подписано в печать 08.06.2019
Формат 60x90/16
Печать офсетная
Усл. печ. л. 12,5
Тираж 200, Заказ № 23871
ООО «Фабрика Офсетной Печати»
www.fop.ru